INVESTIGATING THE USE OF VIRTUAL WORLDS TO TEACH BASICS OF PROGRAMMING TO CHILDREN: A MULTIPLE CASE STUDY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ BATTAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER EDUCATION AND INSTRUCTIONAL TECHNOLOGY

JANUARY 2018

Approval of thesis:

INVESTIGATING THE USE OF VIRTUAL WORLDS TO TEACH BASICS OF PROGRAMMING TO CHILDREN: A MULTIPLE CASE STUDY

submitted by **ALİ BATTAL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Computer Education and Instructional Technology, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ____________________

Prof. Dr. Soner Yıldırım
Head of Department, **Computer Edu. and Inst. Tech.** ____________________

Assist. Prof. Dr. S. Tuğba Tokel
Supervisor, **Computer Edu. and Inst. Tech., METU** ____________________

Examining Committee Members:

Prof. Dr. Yasemin Gülbahar Güven
Department of Informatics, Ankara University ____________________

Assist. Prof. Dr. S. Tuğba Tokel
Comp. Edu. and Ins. Tech., METU ____________________

Prof. Dr. Kürşat Çağıltay
Comp. Edu. and Ins. Tech., METU ____________________

Assist. Prof. Dr. Gül Tokdemir
Department of Comp. Engineering, Çankaya University ____________________

Assist. Prof. Dr. Cengiz Savaş Aşkun
Comp. Edu. and Ins. Tech., METU ____________________

Date: 05.01.2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname : Ali Battal

Signature :

ABSTRACT

INVESTIGATING THE USE OF VIRTUAL WORLDS TO TEACH BASICS OF PROGRAMMING TO CHILDREN: A MULTIPLE CASE STUDY

Battal, Ali

Ph.D., Department of Computer Education and Instructional Technology

Supervisor: Assist. Prof. Dr. S. Tuğba Tokel

January 2018, 322 pages

The main purpose of this study is to investigate the use of VWs in teaching basics of programming for children in different educational programs. More specifically, the current study aims to examine the perception of participants about the ease of use and perceived usefulness of VWs in programming education, the affordances and challenges of using virtual worlds, issues and strategies for the group study, design issues of different educational programs in VWs, factors that affect satisfaction, and avatar issues. In this context, this study was implemented in three different educational programs constituting the single cases of the study as; curricular, extra-curricular, and after-school programs.

The multiple case study was employed among the qualitative designs. In this context, data were collected mainly through interviews, observations and questionnaires from students and teachers in each case separately. Qualitative analysis and descriptive statistics were applied to data obtained for single case analysis of each case. Then cross-case analysis was employed in order to reveal the similarities and differences across the cases at the end of the study.

The results of the current study showed that students perceived VW as useful and used them for learning programming without major difficulties. Having fun, personal contributions, gaining experience on 3D, facilitating group study and motivation were the affordances of using VWs, whilst participants encountered challenges related to the 3D environment, equipment and infrastructure and tasks. Additionally, the results indicated the important issues and strategies for avatars, group study and the design of educational programs in the virtual learning environment. Students' satisfaction in each case was defined and factors affecting their satisfaction were addressed. Finally, similarities and differences across the cases were discussed based on the sub-research questions in the light of the literature. The findings of this current study might help to understand the use of VWs in programming education for children in three different educational programs, and also provide a basis for educators and other researchers in using VWs for the teaching programming.

Keywords: Programming for Children, Coding for Children, 3D Virtual Worlds, different educational programs.

ÖZ

SANAL DÜNYALARIN ÇOCUKLARA YÖNELİK PROGRAMLAMANIN TEMELLERİ ÖĞRETİMİNDE KULLANIMININ İNCELENMESİ : BİR ÇOKLU DURUM ÇALIŞMASI

Battal, Ali
Doktora, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. S. Tuğba Tokel
Ocak 2018, 322 sayfa

Bu çalışmanın temel amacı farklı eğitsel programlarda verilen çocuklara yönelik programlamanın temelleri öğretiminde sanal dünyaların kullanımının incelenmesidir. Bu çalışma daha açık bir ifade ile katılımcıların sanal dünyaların programlama eğitiminde kullanım kolaylığı ve yararlılığı ile ilgili algılarının belirlenmesi, programlama öğretiminde sağladığı olanaklar ve zorlukların belirlenmesi, grup çalışması ile ilgili durum ve stratejilerin, farklı eğitim programlarında sanal dünyaların kullanımının tasarımı ile durumların açığa çıkarılması, katılımcıların memnuniyetini etkileyen faktörlerin ve avatar ile ilgili durumların belirlenmesi amaçlarını taşır. Bu bağlamda, bu çalışma müfredata entegre, müfredat dışı ve okul sonrası program olmak üzere üç farklı eğitsel programda uygulanmıştır.

Bu çalışmada nitel araştırma desenlerinden çoklu durum çalışması yapılmıştır. Veriler her bir durum çalışmasında öğrenci ve öğretmenlerden görüşme, gözlem ve anket yoluyla toplanmıştır. Tekli durum çalışmalarının analizinde nitel analiz yöntemleri ve betimsel istatistikler kullanılırken, çoklu durumlar arasındaki farklılıkları ve benzerlikleri ortaya çıkarmak için durumlar arası analiz uygulanmıştır.

Çalışmanın sonuçları öğrencilerin programlama öğretiminde sanal dünyayı kullanımın kolay ve yararlı olduğu algısına ulaşmıştır. Eğlence katma, kişisel katkılar, 3B ortamda deneyim kazanma, grup çalışmasını artırma ve motivasyon sanal dünyaların sağladığı yararlar olarak bulunurken; katılımcılar 3B ortam, ekipman ve altyapı ile görevler konusunda bir takım zorluklar yaşamışlardır. Ayrıca, avatar, grup çalışması ve eğitsel programların tasarımı ile ilgili önemli durum ve stratejiler ile ilgili bulgular ortaya çıkarılmıştır. Her bir durumdaki öğrencilerin memnuniyet düzeyi ile memnuniyetlerini etkileyen faktörler belirlenmiştir. Çalışma sonunda, durumlar arasındaki benzerlik ve farklılıklar araştırmanın alt araştırma soruları doğrultusunda sunulmuş ve ilgili literatür ışığında tartışılmıştır. Çalışmanın bulguları, farklı eğitsel programlarda sanal dünyaların programlama eğitiminde kullanılmasının anlaşılmasına katkı sağlayabileceği gibi, sanal dünyayı programlama öğretiminde kullanmak isteyen eğitimci ve araştırmacılara temel oluşturabilir.

Anahtar kelimeler: Çocuklar için Programlama, Çocuklar için Kodlama, 3B Sanal Dünyalar, farklı eğitsel programlar

to my family

ACKNOWLEDGEMENTS

Completing a dissertation was a long journey consisting of many difficulties which was unbearable without the support and guidance of special people around me. Firstly, I would like to express my deep gratitude to my supervisor, Assist. Prof. Dr. Tuğba Tokel. She was always with me with her endless patience, support, encouragement, guidance and advice throughout the study. Besides, I would like express my sincere appreciations and thankfulness to the examination committee members, Prof. Dr. Kürşat Çağıltay, Prof. Dr. Yasemin Gülbahar Güven, Assist. Prof. Dr. Gül Tokdemir and Assist. Prof. Dr. Cengiz Savaş Aşkun for their valuable feedbacks, suggestion and comments.

I would like to thank to head of department, Prof. Dr. Soner Yıldırım for his support and advices. I am also very grateful to many friends and colleagues who supported me during this long process, especially Halil Kayaduman, Mustafa Şat, Tuğba Kamalı, Zafer Kadırhan, Yunus Alkış, and Ozan Raşit Yürüm.

I also would like to thank The Scientific and Technological Research Council of Turkey (TUBİTAK) for the support of scholarship during my PhD education. This study was also supported by Middle East Technical University (METU). I would like to thank to METU.

I would like to express my special gratitude to the management of educational programs for allowing conducting this study and their stuff for helping me during the implementation phase. Especially, I would like to thank to Ünsal Dilekçi, Merve Keleş, Burcu İnce, and Rukiye Altın for their support and efforts.

Finally, I would like to thank my parents, who always encourage me to push the limits of capabilities and life with their trust. Last but not least, I would like to express my greatest gratitude to my wife, Esra for her endless and unconditional love, support, understanding and patience during this long process. As a last word, my little son, Hasan Eymen, helped me to overcome the difficulties with his smile. I am sorry for the unallocated time in order to play with him more during this long process.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BoE: Board of Education (Talim Terbiye Kurulu)

GBS: Goal Based Sceneria

LSL: Linden Scripting Language

MoNE: Ministry of National Education

NPC: Non-Playable Character

OS: OpenSim

SDP: Sanal Dünyada Programlama (Programming in Virtual World)

SL: Second Life

S4OS: Scratch for OpenSim

S4SL: Scratch for Second Life

VW: Virtual World

CHAPTER 1

INTRODUCTION

There is a growing interest in teaching programming to children nowadays. Numerous scholars, worldwide initiatives and other stakeholders advocate the teaching basics of programming to children through the organization of different activities. The popular term, *coding* and the well-known term, *programming* actually refers to *computer programming*, which Brennan (2017) defined as "specifying instructions, or code, in specialized languages to control computer activity" (p. 123). Another scholar, Guzdial (2015) defined *programming* as the "process of writing programs" (p. 2). As it is understood from the definition, programming includes steps such as building a logic to solve a problem, writing programming expressions in a programming language, transforming them into machine language via compiling, testing and debugging (Papadopoulos & Tegos, 2012). Various programming languages and environments have been developed in order to remove or reduce some of these programming steps for novices, especially for children. For example, block-based programming tools remove the debugging process, which in fact would be unnecessary for novices to learn.

Various scholars have argued that everyone at any age should learn programming (Guzdial, 2015; Kafai & Burke, 2014). For example Maloney, Peppler, Kafai, Resnick, and Rusk (2008) mentioned that learning programming is an educational right of young people and that they should engage in programming in some way due to various reasons. Guzdial (2015) summarized the reasons for teaching programming to children. Firstly, children should be introduced to programming because of the requirements of today's computing intensive world and in preparation for the many computing-related future careers they will likely face in their future (Knobelsdorf & Vahrenhold, 2013). Secondly, children could understand the world better having learned programming since computing is now omnipresent in almost all aspects of life. Learning programming provides children the opportunity for developing certain skills. It was argued in previous studies that programming is an important means for developing higher order thinking skills (Özmen & Altun, 2014), problem solving and analytical thinking skills (Gomes & Mendes, 2007), and computational thinking skills (Akcaoglu, 2014; Grover & Pea, 2013; Kafai & Burke, 2013a). Finally, learning programming increases computational literacy and productivity. At this point, Brennan, (2017) simulated programming to producing a poem with written text and producing a documentary film with videos; arguing that children could produce many artefacts from games to websites through programming.

In parallel to today's rise of programming and based on the importance of programming, a comeback has been seen for programming to some extent in schools (Kafai & Burke, 2013; Robins, Rountree, & Rountree, 2003), as well as outside of formal school settings such as after-school programs (Fields, Kafai, & Giang, 2017; Krishnamurthi, 2017). Programming activities both in and out of the school setting have been organized worldwide to introduce programming to children. Many countries have been trying to transform their current ICT courses or to establish new computing related courses to teach programming to children (Gal-Ezer & Zur, 2013; Gujberova & Kalas, 2013; Kalelioğlu, 2015; Menekse, 2015; Tenenberg & McCartney, 2014; Yadav, Gretter, Hambrusch, & Sands, 2017). In Turkey, programming has moved into even ICT courses in both primary and secondary education through the advancements in the curriculum changes (Kalelioğlu & Gülbahar, 2014). In addition, a new "Computer Science" course is now offered to some high school students by the Turkish Ministry of National Education (MoNE) and it is planned to extend that course for most other students in the near future (MoNE-BoE, 2016). However, in spite of its increasing use and place within the curriculum, it has been reported that there are many students who have never even heard of programming. Maloney et al. (2008) studied with a group of young people aged 8-18 to offer programming activities in an after-school center and argued that 90% of the youth attending the center had “never been in a computer class during their entire K-12 schooling experience” (p. 4). Kafai and Burke (2013b) looked at the current use of computers in education from a different point of view and stated that, at the time of their research, curriculum did not support computational thinking and did not go beyond the teaching of word processing or how to create a PowerPoint presentation; both of which do not engage students to think more creatively or critically.

Previous research shows that besides its advantages for learners, programming is considered difficult for learners, especially for novices (Guzdial, 2004; Kelleher & Pausch, 2005; Saeli et al., 2011; Schulte & Carsten, 2013). It has been argued in the literature that learning programming could be very difficult for novices of all ages due to reasons such as understanding the problems, coming up with solutions, mental representation of abstract concepts, the rigid syntax and semantics of each programming language, arbitrary code with often confusing names, irrelevant activities to the teaching of programming, and a lack of support (Kelleher & Pausch, 2005; Maloney et al., 2008; Pears et al., 2007; Resnick, Maloney, Hernández et al., 2009). Such reasons could be overwhelming for beginners, discourage them and decrease their motivation towards learning programming. Papert (1980) stated that such difficulties could be overcome if the teaching of programming is supported by proper strategies and tools and, in accordance with this purpose, he developed the LOGO language. Since then, a number of strategies and tools have been developed by researchers after Papert’s LOGO programming language in order to simplify the mechanics of programming, and to provide support and motivation for beginners to learn programming (Kelleher & Pausch, 2005).

Virtual worlds (3D computer-based environments with multiple users) are one of the recent technological developments in the educational field (Esteves, Fonseca, Morgado, & Martins, 2008) that could be used as a tool for teaching programming to children. Dreher, Reiners, Dreher, and Dreher (2009) highlighted the importance of using virtual worlds in teaching information science education, including programming education, in terms of motivating students on computer-related courses. They argued that visualization of programming concepts in 3D environment, testing code in context and working collaboratively within the environment could be essential points in using virtual worlds for the teaching of programming.

1.1 Problem Statement

Technological advancements have made the use of VWs possible in different aspects of daily life in terms of access, usability and cost effectiveness (Dawley & Dede, 2014; Messinger et al., 2009; Warburton, 2009). Using VWs for educational purposes has since increased and a review of the literature shows that there has been a growing body of research on how to make use of VWs for educational purposes (Dalgarno & Lee, 2010; Dickey, 2005a; Hew & Cheung, 2010; Omale, Hung, Luetkehans, & Cooke-Plagwitz, 2009). Virtual worlds offer educators and students the ability to create rich and compelling 3D objects within the environment (Esteves, Fonseca, et al., 2008). Users can build objects and attach scripts to them in order to interact with other objects, the environment, or with other learners. Besides, it enables learners to study in groups and to share other learners' artifacts within the virtual world, subject to the necessary permissions being granted. Programming is essential to the construction of virtual artifacts (Girvan, Tangney, & Savage, 2013) because programming and the adding of scripts to 3D objects makes them "more meaningful by adding behaviors and interactivity which bring them to life" (Rosenbaum, 2008, p. 6).

Demographic research on the use of virtual worlds shows that the majority of users are aged 10 to 15 and use VWs for entertainment as well as for formal and informal learning opportunities (Merchant, 2017). In spite of the increasing use of VWs among teens and preteens, specific and verified research examining the use of VWs in diversified disciplines is limited (Kim, Lee, & Thomas, 2012; Tokel & Cevizci-Karataş, 2014). Besides, research investigating the use of VWs should reveal unique affordances when used in a specific area (Hew & Cheung, 2010). On the contrary, the educational use of virtual worlds is typically limited to the replication of traditional teaching approaches but in virtual environments (Winn, 2005) such as lecture theatres, or virtual university campuses. Thackray, Good, and Howland, (2010) argued that many studies just replicated traditional teaching scenarios in VWs, such as with texturing slide presentations on boards.

Some studies related to the use of VWs in programming education aimed to teach programming by using their own language such as Linden Scripting Language (LSL), whilst some used other programming environments such as Scratch for Second Life or Scratch for OpenSim (Vosinakis, Anastassakis, & Koutsabasis, 2016). Those Scratch-like programs actually translate the pseudocode of Scratch into LSL and make programming easier (Rosenbaum, 2008). Esteves, Fonseca, Morgado, and Martins, (2011) studied with university level students to understand how teaching and learning of computer programming could be developed in Second Life by using LSL, which was difficult for novice learners to handle when they were first introduced to programming. The findings of their study showed that the use of LSL was hard for even undergraduate students. It is therefore not a good choice to use LSL in teaching programming for students introducing for the first time (Dreher et al., 2009). Novice learners, especially children, should be introduced to programming with *low-floor* and *high-ceiling* tools, which they would be able to use intuitively and create a wide range of complex artefacts (Kafai & Burke, 2014). At this point, Girvan et al. (2013) proposed the use of S4SL in programming "SLurtles," a programmable turtle, in Second Life and investigated whether or not graduate students could use them effectively for constructing artifacts in SL by using S4SL.

VWs, with their characteristics and affordances when used for educational purposes could also be utilized for teaching programming to children. At this point, the use of virtual worlds could bring about new opportunities for children with regards to improving programming performance. Educators and researchers should investigate VWs in terms of overcoming learners' difficulties encountered during the learning of programming (Esteves et al., 2011; Esteves, Fonseca, et al., 2008). However, studies concerned with the use of VWs in programming education are limited and mostly conducted with high school, university and graduate students (Girvan et al., 2013; Hulsey, Pence, & Hodges, 2014; Pellas, 2014a; Seng & Edirisinghe, 2007). Besides, curricular and extra-curricular activities in schools and after-school programs outside of schools play an important role in programming education for children in terms of reaching as many students as possible (Kafai & Burke, 2014). Therefore, there is a need to understand the use of virtual worlds in programming education for children in different educational programs.

1.2 Purpose of the Study

The purpose of this current study is to investigate the use of virtual worlds in teaching basic of programming to children in different educational programs. In a broader context, the study aims to examine the perceptions of participants about the ease of use and perceived usefulness of VWs in programming education, as well as the affordances and challenges of using virtual worlds, issues and strategies related to group study and avatars, and factors affecting the satisfaction and designs of different educational programs in VWs. This study also aims to investigate the use of VWs in three different educational programs; curricular, extracurricular and after-school programs. The final purpose of this study is to reveal any similarities and differences between these cases.

This study aimed to build code on Scratch for OpenSim (S4OS) for completing tasks in the OpenSim (OS) virtual environment. The term, "Sanal Dünyada Programlama (SDP)", which means Programming in Virtual World, refers to the integration of S4OS with VW throughout this study. The topics intended to be taught in VW are limited to the basics of programming and the capabilities of S4OS (see Appendix A for the topics).

The main and sub-research questions of this study are as follows:

How could virtual worlds be utilized in programming education for children from different educational programs?

a. To what extent do participants perceive the ease of use and usefulness of SDP?
b. What are the affordances and challenges of using virtual worlds in programming education for children?
c. How does avatar representation affect the experience of participants?
d. How do the issues and strategies about group study in SDP affect the experience of participants?
e. What are the factors affecting participants' satisfaction in SDP?
f. What are the issues and strategies for the design of SDP?

1.3 Significance of the Study

Learning programming is essential for anyone at any age from primary school up to university level (Sauppé, Szafir, Huang, & Mutlu, 2015), and even from young working adults to the retired in accordance with the requirements of 21st century skills (Guzdial & Disalvo, 2013). Thus, many countries and schools have been trying to introduce concepts related to programming to children in different ways (Kafai & Burke, 2014). Firstly, they have been working on integrating programming into school curriculum by updating current courses as well as developing new standalone courses such as “Computer Science.” Secondly, they have been trying to implement extra-curricular activities into school settings such as establishing programs for software and game design projects. Such kinds of program are important since they are good examples of how programming could be contextualized in a program independently of curriculum. Lastly, after-school programs are offered to introduce learners to programming in different contexts. In addition to in-school and out-of-school activities, programming activities are organized worldwide in order to inform people about coding such as “Hour of Code,” “EU Code Week,” “Bebras,” and locally with “Georgia Computes,” and “KodlaManisa.”

Learning programming is generally considered difficult by learners of any age (Guzdial, 2004). Therefore, there is a need to use tools or environments as an aid to making programming easier to grasp (Gomes & Mendes, 2007). In their study, Kelleher and Pausch (2005) reviewed nearly 80 tools and categorized them according to their mechanical and motivational process benefits. Tools in the mechanical process category focused on making the mechanics of programming more manageable such as removing unnecessary syntax, designing languages closer to spoken language, introducing programming in visible context, and finding alternatives to typing programs. These kinds of tools allow learners to focus on the logic and underlying structures of programming rather than becoming overwhelmed with all the other issues associated with programming. Tools in the motivational process category were aimed at increasing learners’ motivation during learning by offering a social and motivating context through designing activities that draw learners’ attention such as moving cars, or the construction of objects. These kinds of tools are also important since programming is considered a solitary activity (Brennan, 2013; Rosenbaum, 2008).

Virtual worlds with their various features could be used as a tool in the teaching of programming. Previous studies used VWs in teaching programming to students from high schools though to postgraduate by using LSL, the language of VWs (Esteves et al., 2011; Hulsey et al., 2014; Seng & Edirisinghe, 2007). Dreher et al. (2009) advocated the use of virtual worlds in learning programming since they would allow direct visualization of the outcome, which could be seen animated virtually, and would provide learners with quick and concrete feedback. In addition, learners are able to test their code by applying it in a certain context and social environment. It is also possible to study in groups thanks to the multiuser capability of VWs, which might promote group or pair programming (Beck, 2000). Teaching programming in virtual worlds could motivate learners intrinsically and offer them a range of benefits when compared to teaching with traditional methods. Virtual worlds, as previously mentioned, could afford users certain benefits through the application of tools in both mechanical and motivational processes.

It was found that LSL was *high-floor* and therefore hard to understand by even learners at the undergraduate level (Esteves et al., 2011). Therefore, there is a need to employ a *low-floor* easy to use tool for completing tasks in VWs. At this point, S4SL and S4OS, which are Scratch-like programming tools, could be used for building code similar to the original Scratch. Previous research tried to integrate these programming tools into VWs for graduate learners (Girvan et al., 2013; Sajjanhar & Faulkner, 2014) and also for high school students (Pellas, 2014a; Pellas & Peroutseas, 2016; Pellas & Vosinakis, 2017).

Virtual worlds offer an exciting new environment for learners to engage in programming through the construction of meaningful 3D artefacts within group studies; something that is arguably difficult to achieve with other programming environments (Pellas & Vosinakis, 2017). However, the use of VWs for teaching programming is also limited (Pellas, 2014a; Seng & Edirisinghe, 2007) and scarce for children. The findings of this current study will help to understand the use of VWs in programming education using three different educational programs. The study also aims to reveal the similarities and differences across the three programs, and offer clear points about each program. In a broad manner, it also aims to shed light on the potential advantages and possible challenges associated with programming education through virtual worlds following Hew and Cheung's (2010) argument for further research to examine the unique affordances of VWs. Furthermore, the current study aims to reveal the acceptance of VWs, factors affecting satisfaction, group and avatar issues, and the main points in designing different educational programs. The results of this current study can also provide a basis for educators and other researchers in using VWs for the teaching basic of programming to children. In this way, it would be possible to investigate the application of a different programming tool, using VW in programming education; thereby realizing practical implications for the current study in being able to reach and educate more children in the area of programming. Lastly, it is expected that the current study will contribute to the body of literature on computing education research.

1.4 Definition of Terms

This section explains some of the more significant terms used throughout this study.

After-school Program: This is an informal learning environment held outside of school and removed from the formal settings of school-based learning (Shernoff & Silva, 2017). In the context of this current study, the term "After-school Program" refers to a programming course named "Üç Boyutlu Ortamda Temel Programlama Eğitimi" (Basics of Programming Education in 3D Virtual World) was offered at the Continuing Education Center of a public university in Turkey.

Curricular Program: This refers to a standalone course existing in the curriculum of a school setting. In the context of this current study, it corresponds to a compulsory ICT course for 4^{th} and 5^{th} grade students in Turkish schools.

Extra-curricular Program: This refers to activities occurring independent of the school curriculum. Such programs are usually promoted by schools and rarely the community (Fredricks, 2017). In the context of the current study, activities held within a school setting that are independent of the curriculum are referred to as extra-curricular programs.

Linden Scripting Language (LSL): This is the own programming language of SL and OS, which has C- style syntax and keywords (Esteves, Fonseca et al., 2008).

OpenSim (OS): This is a free, open-source, multi-user 3D application server upon which developers can create 3D virtual worlds and then customize them according to their needs.

S4SL: This is a Scratch-like programming tool which produces code in LSL for SL.

S4OS: This is a Scratch-like programming tool which produces code in LSL for OS.

Sanal Dünyada Programlama (SDP): This special term corresponds to 'Programming in Virtual World'. It is the name of the programming environment used throughout the current study, which is comprised of two components; a 3D learning environment that includes a number of programming activities, and the Scratch for OpenSim (S4OS) program used for building code to complete activities within the 3D environment. It therefore refers to the integration of S4OS within VWs.

Non-player Character (NPC): It is a computer-generated agent that can be in multiple forms which fulfill a pre-defined programmed activity continuously or triggered via user interaction (Kapp & O'Driscoll, 2010).

Virtual Worlds (VWs): These are 3D computer-based environments supporting multiple users represented as avatars, and in which users interact within the environment, communicate with others and take part in experiences via their avatars that are similar to those in a real-world context (Dieterle & Clarke, 2006).

1.5 Organization of the Study

In this current study, Chapter One presents the problem statement, purpose, research questions and significance of the study, as well as a definitions of terms. Chapter Two addresses 3D VWs, programming education, and relevant studies about the use of VWs in programming education. Chapter Three explains the detailed methodology of the study by providing information about the research design, pilot study, design and development of virtual environments, the selection of cases and participants, data collection methods and procedure, data analysis, the researcher's role, trustworthiness and limitations of the study. Chapter Four represents the results of single and cross-case analysis in line with the sub research questions. Chapter Five discusses the study's findings in light of the literature, and finally presents the implications and recommendations for further research.

CHAPTER 2

LITERATURE REVIEW

This chapter presents an in-depth literature review relevant to the current study. Specifically, 3D Virtual Worlds (VWs), their characteristics, affordances and challenges of using VWs for educational purposes are reviewed. Then, programming education for children are addressed. Thirdly, the importance of programming and tools for teaching programming are elaborated on. Finally, research about programming education in 3D virtual worlds is reviewed.

2.1 3D Virtual Worlds

Technological advances have affected many aspects of human life, particularly in business and industry. Indispensable prevalence of new technologies into human lives has also impacted the field of education, having expanded the imagination of how learning environments could also be (Dickey, 2005b). 3D virtual worlds are considered one of the new technologies. Advancements in video and audio technology, falling computer prices, increasing computing capacity and greater broadband have transformed 3D virtual environments to be more pervasive, useable, accessible and cost effective (Dawley & Dede, 2014; Messinger et al., 2009; Warburton, 2009). As a result of these advances, efforts to make use of these technologies for educational purposes has increased since the introduction of 3D virtual worlds. A review of the literature shows a growing body of research and an effort to make use of 3D virtual worlds for educational purposes (Dalgarno & Lee, 2010; Dawley & Dede, 2014; Dickey, 2005b; Hew & Cheung, 2010; Omale et al., 2009). However, there is still a lack of research in different areas of education. Therefore, research about VWs needs to be; (a) in more diversified disciplines (Tokel & Cevizci-Karataş, 2014), (b) more specific and verified (Kim et al., 2012), (c) addressing more advanced technological use of VWs (Dickey, 2005a), and (d) more meaningful to reveal their real potential affordances (Hew & Cheung, 2010).

The history of VWs goes back to the late 1970's in which Multi-User Dungeon (MUD) was first introduced (Bartle, 2009). In MUD, everything was text-based. For example, what characters did, saw and heard was controlled and reported via texts. In the late 1980's, MUDs were modified to "TinyMUD," which were introduced with small changes in MUD such as removing the weapons and monsters (Bruckman, 1997). Since then, with the help of advances in computing and networks, other terms such as Multi-User Virtual Environments (MUVE), Massively Multiplayer Online Role Playing Games (MMORPG), and Virtual Learning Environments (VLE) were coined to refer to 3D technologies (Dieterle & Clarke, 2006). There are some small differences among the terms (Tokel & Topu, 2016) although they are generally considered as one and used interchangeably in some studies (Kim et al., 2012; Omale et al., 2009). Throughout this current study, Virtual Worlds (VWs) is the preferred term applied. The significant difference between VWs and other terms is that there is a predefined storyline in MUVEs, while there is no story at the beginning in VWs and they could be repurposed based on the needs of the study (Warburton, 2009). Another difference between VWs and MUDs is that VWs yield more types of communication forms and building options for end users (Dickey, 2003).

There is no consensus on the definition of VW in the literature, since the terms could be used in different forms by different people in different times (Bell, 2008). Warburton (2009) argued that the reason behind the definition problem is multiple emerging forms of virtual environments. Academics, industry professionals and the media have defined it in numerous forms. Dickey (2005a) defined virtual worlds as "networked desktop virtual reality in which users move and interact in simulated 3D spaces" (p. 439). The most comprehensible definition belongs to Dieterle and Clarke (2006), who defined 3D virtual worlds as 3D computer-based environments with multi-users where participants are presented as avatars, a kind of graphical representation of themselves in which avatars interact within the environment, communicate with others and take part in experiences similar to those in a real-world context. In virtual worlds, users can apply their knowledge in 2D and 3D formats such as billboards, buildings, and interactive media (Richter & Dawley, 2010).

There are many open source and proprietary VW applications in use within educational settings (Warburton, 2009) such as ActiveWorlds, Second Life, OpenSim, Croquet, Project Wonderland, and Adobe Atmosphere (Dickey, 2005a; Hew & Cheung, 2010). The educational use of VWs could be listed as follows (Dieterle & Clarke, 2006; Messinger et al., 2009; Reisoğlu, Topu, Yılmaz, Karakuş Yılmaz, & Göktaş, 2017; Tokel & Cevizci-Karataş, 2014):

- Yielding online communities to train participants for professional development,
- Providing science-related activities and engaging in scientific inquiry,
- Promoting the understanding of students in different areas such as history, medical, mathematics, business, engineering and sports,
- Promoting social and moral development of participants,
- Providing an environment for teaching programming and language.

2.1.1 Characteristics of VWs

There are some common characteristics of VW, regardless of application and content. They are the immersive 3D environment, avatars and multiple users, and multiple communication forms (Dickey, 2005a, 2005b); interaction with objects and other avatars (Hew & Cheung, 2010); persistence of objects, immersive environment through the use of realistic 3D graphics (Dalgarno & Lee, 2010; Delwiche, 2006; Dieterle & Clarke, 2006); and object construction and manipulation (Dawley & Dede, 2014; Messinger et al., 2009). With their characteristics and affordances, VWs allow educators to incorporate a large variety of learning options and strategies into their teaching such as providing different types of feedback or roleplay (Dawley & Dede, 2014; Reisoğlu, 2014).

The features mentioned could be of "relative importance to a particular educational research" (Richter & Dawley, 2010, p. vi). In the following section, relevant characteristics of VWs concerned with the current study are elaborated on.

3D Object Construction and Manipulation

Users could create 3D geometric shapes such as cubes, spheres and prisms easily with the help of built-in features of VWs. Those objects are called as prims and it is possible to transform prims into new shapes by linking them (Kluge & Riley, 2008). Learners might also edit and move their own objects and other's objects if the necessary permission is defined. Other users can examine created objects and share them with their peers since the virtual world and objects are persistent (Girvan et al., 2013). Moreover, some virtual worlds allow users to import textures to apply them to the objects in terms of achieving a rich and realistic appearance. Virtual worlds provide educators and learners with the ability to create 3D objects (Delwiche, 2006; Esteves, Antunes, Fonseca, Morgado, & Martins, 2008). Users can create and edit objects, and attach scripts to those objects in order to assign behaviors in virtual worlds (Dalgarno & Lee, 2010).

The features of VWs related to construction mentioned here support the ideas of constructionism in which learning occurs when individuals construct knowledge structures while creating a public artifact (Ingram-Goble, 2013) and sharing it with others (Kafai & Resnick, 1996). Ackermann (2001) argued that knowledge is a personal experience that needs to be constructed during the learning process as it is "not a commodity to be transmitted, encoded, retained" (p. 7). Building options of VWs and enabling the creation and manipulation of objects via programming allows learners to create new meaningful and shareable artifacts, and to test their understanding in line with the ideas of constructionism (Hoyles, Noss, & Adamson, 2002, as cited in Girvan, Tangney, & Savage, 2013).

To summarize, VWs provide both educators and learners with the abilities to use them for exploring, constructing and manipulating virtual objects (Dalgarno, & Lee, 2010), which could be considered as the most promising aspect that distinguishes VWs from other applications (Kluge & Riley, 2008). Learners, having the ability to employ such features, can "compare and contrast their concepts with existing knowledge in graphical form" (Richter & Dawley, 2010, p. vi), which is especially useful for concepts that require a high degree of visual representation.

Avatar

Avatars are graphical representations of users in a 3D environment, enabling users to interact with other avatars and objects in the environment (Dickey, 2005b). Avatars might be human or non-human characters, with users offered many avatar options or they can choose to create their own. Users customize their avatars accordingly such as changing their avatars' clothing, height, skin color, hair, and eye color. Besides, avatars may also be accessorized with items such as a cowboy hat, glasses, and shoes. Avatars move around the environment in different modes (walk, run, fly and teleport), and they can even perform some gestures, facial expressions and emotional states such as showing happiness, or crying, typing, raising a hand, or applauding. Users interact in the 3D environment with their avatars via touching things, building something, doing sports, dancing, sitting on a chair, or playing a video on a presentation board (Messinger et al., 2009). Users perform the actions, interact with the environment, and communicate with other users via their avatars.

Representation through avatars in virtual worlds is important due to various reasons. Firstly, it increases the sense of immersion in the environment (Dalgarno & Lee, 2010). Secondly, users build identity and trust within the environment with the help of their avatars (Richter & Dawley, 2010; Tokel & Cevizci-Karataş, 2014). Thirdly, users feel themselves as part of the community and as though they are in a real place within the environment (Yee, Bailenson, Urbanek, Chang, & Merget, 2007). Finally, avatar representation increases communication, interaction and collaboration in the environment when compared to other mediums offering only text-based communication (Hew & Cheung, 2010). Similarly, Feldon and Kafai (2008) argued that displaying of avatars in the environment supports engagement and interaction.

Multiple Forms of Communication

VWs generally support multiple forms of communication and users communicate verbally either in voice or text format (Kim et al., 2012). Furthermore, text-based chat could involve both synchronous and asynchronous communication. Participants feel more involved rather than isolated from the community and obtain an immediate response with synchronous communication, whereas asynchronous communication allows the user to control the flow and thereby maintain a degree of flexibility over their communication (Petrakou, 2010).

VWs typically provide two types of text-based communication; public chat and private messaging (Dawley & Dede, 2014). Users prefer private chat in order to chat one-on-one such as between instructor and student, and prefer public chat when explaining an idea on a given topic publically in a virtual classroom (Dawley, 2009). Moreover, it is possible to communicate with users via the gestures and facial expressions of avatars in a nonverbal format (Tokel & Cevizci-Karataş, 2014). For example, a teacher could point out a student who had misunderstood instructions or misbehaved in the environment through nonverbal forms (Petrakou, 2010). Previous research shows that users usually prefer verbal communication, although VWs can encompass both verbal and nonverbal communication forms (Pita & Pedro, 2012).

To summarize, VWs offer many types of in-world communication, from group chat to private messaging, from notecards to conferences (Dawley, 2009). Communication in different forms enhances interaction among participants, their sense of immersion, likelihood of sharing information, and the building of social connections and communities of practice (Petrakou, 2010; Richter & Dawley, 2010).

Group Study

VWs can support group study in many aspects since they allow learners to perform tasks together rather than just communicate (Dalgarno & Lee, 2010; Duncan, Miller, & Jiang, 2012). It is possible to define groups in VWs and invite or add members to defined groups, and to create contact lists for synchronous or asynchronous communication (Richter & Dawley, 2010). Group members can construct 3D objects and each member of the group can manipulate objects owned by the group. Moreover, members of group can collaborate with each other and edit the same objects synchronously and asynchronously since the objects in virtual worlds are persistent (Girvan, Tangney, & Savage, 2013).

VWs facilitate group study and collaboration due to various reasons. Firstly, previous research indicates that different communication opportunities between learners facilitate collaboration and social negotiation (Dickey, 2005a), and VWs yield a wide variety of communication channels that are more similar to face-to-face communication (Kluge & Riley, 2008). Then, they present learners with resources and a work place to study in groups (Dickey, 2003). Finally, interaction among learners needs to be fostered for an effective group study. At this point, VWs help to increase interaction between learners (Petrakou, 2010).

Interaction

Interaction is an important contributor of meaningful learning in the online learning environment (Omale, 2010). There can be different types of interaction in online learning environments such as learner-learner, learner-content, or content-content (Moore, 1989). In VWs, interaction between users and objects could be in three possible combinations; person-person, person-object, and object-object (Antonacci & Modaress, 2008). The most visible is person-person which occurs when users study collaboratively together and share mutual points. Multiple users can exist in VWs, with each user represented by a recognizable avatar; therefore the person-person type of interaction is inevitable. Besides this, such type of interaction can be enhanced with the help of VWs' different communication forms (Petrakou, 2010). Rich interaction is possible among users such as through the exchange of objects, or messages (Messinger et al., 2009). This goes beyond the interaction of a traditional face-to-face relationship between learner and teacher, and also addresses a deficiency of the face-to-face relationship. In VWs, all learners have equal right to take part in activities, regardless of their social position or personal background (Kim et al., 2012). For example, VWs allow an extreme introvert to take part in any or all of the activities; whereas in the real world, they would most likely elect not to interact with their peers.

Another type of interaction is person-object. Users interact with objects in VWs all the time. Users can construct complex objects or undertake an experiment with the help of objects provided in VWs, and can thereby observe the consequences of their actions (Antonacci & Modaress, 2008). Interaction between objects and avatars can result in an educational experience for users (Gamage, Tretiakov, & Crump, 2011). The final type of interaction is object-object. It is possible to simulate processes via programming objects to interact with each other and then to observe the results (Antonacci & Modaress, 2008; Warburton, 2009).

Interaction is enriched in VWs, thanks to their extensive features. As Barab, Hay, Barnett, and Squire (2001) argued, “the more technology can get out of the way, the more actual interactivity might take place in the environment” (p. 136). However, technical challenges of VW and the adaptation period of users could be considered as obstacles to interaction (Petrakou, 2010), and such issues should be taken carefully into consideration.

Immersive and Persistence Environment

VWs are immersive and persistent environments, which distinguishes them from other types of computer application (Dalgarno & Lee, 2010). While an immersive environment relates to the realistic display of the environment in which users feel as though they are inside the environment, persistence is about feeling that the virtual world exists even when a user is not logged in. It was argued that user artefacts are also persistent in the VW environment, even when the user is not in the environment (Pellas & Peroutseas, 2017). Dede (2009) listed the potential educational advantages of immersive and persistence interfaces as; (a) to provide multiple perspectives, (b) to provide digital simulations and rich interactions, (c) to enhance transfer through the simulation of real world examples, and (d) to enhance participants’ engagement and learning.

2.1.2 Affordances and Challenges

VWs have affordances and challenges when used for educational purposes (Kluge & Riley, 2008; Petrakou, 2010; Samur, 2009), and it is therefore advisable for researchers and practitioners to ensure they keep themselves informed about such affordances and challenges. Previous studies listed the affordances of using VWs for educational as follows (Dalgarno & Lee, 2010; Duncan et al., 2012; Kluge & Riley, 2008; Topu et al., 2017; Warburton, 2009):

- VWs can support learning tasks that are impractical or impossible to implement in the real world. For example, content to be learned could be historical, too costly, not real, or not safe to practice in real life;
- VWs enable users to design interactive environments, yielding opportunities for interaction through elements in the environment;

- VWs can facilitate experiential learning and contextualization. In this way, they increase the potential for the transfer of what is learned to real life situations;
- VWs allow learners to view content from multiple perspectives and to enhance the spatial ability of learners through multiple dimensions;
- VWs support multiple users from different locations, backgrounds and culture. It is possible for learners to be exposed to authentic content and culture, and to an exchange of knowledge and culture between learners. VWs also support collaborative activities;
- VWs enable learners to construct objects and share them with others. Learners have the chance to learn by doing and making rather than being taught;
- VWs yield an immersive 3D environment and an augmented sense of presence for learners;
- VWs provide intrinsic motivation and engagement for learners as well as multifaceted feedback.

The complexity of immersive environments present technical and social challenges for educators in their integration within educational activities. Challenges of using VWs for educational purposes are listed as (Duncan et al., 2014; Dunleavy & Dede, 2014; Hew & Cheung, 2010; Omale et al., 2009; Warburton, 2009):

- VWs can potentially distract students from learning goals since they could get off-task and lose concentration in an immersive virtual environment; causing a lack of participation and/or inappropriate behavior;
- Technical problems with equipment, networks, and institutional firewalls are potential problem areas;
- Some skills in VWs such as navigation, creation of objects, or the manipulation of avatars and other issues could be daunting or overwhelming for some students;
- Identity construction in some public VWs could be difficult since students may introduce themselves differently. Building strong and accurate relationships might be problematic because of accountability issues;

- VWs require high performance capability computer hardware, especially needed are powerful graphics cards, high levels of RAM and broadband Internet connection for both the server and client side. Problems in any one of these could lead to a computer crash, performance lag and/or down time;
- Building trust, authenticity, accountability and scaffolding students is essential in group study work in VWs. Providing and maintaining group study could be difficult in certain situations. Additional asynchronous communication mechanisms may be needed in order to enhance group activities;
- Designing activities in VWs requires considerable time and skill; more so than designing activities in other platforms. Instructors need to be skilled in time management, design and the handling of technical issues;
- Costs for running a specific VW is another potential challenge, although some platforms are locally hosted and open sourced. Some applications require a premium level account in order to be able to design specific activities. Maintaining a virtual world platform might entail additional costs;
- Standardization among different VW applications and viewers could be a problem for both developers and users;
- Although VW is persistent, persistence only exists once an avatar was online in that environment.

2.2 Programming Education

In recent years, there has been growing worldwide interest towards teaching programming to some extent for all ages. Numerous scholars advocate that everyone needs to learn programming (Duncan et al., 2014; Guzdial, 2015; Guzdial & Disalvo, 2013; Kalelioğlu & Gülbahar, 2014; Resnick, Maloney, Hernández et al., 2009). For example, Guzdial and Disalvo (2013) argue that everyone from kids to working adults and even retired people should learn programming to some extent in order to gain a level of familiarity with the concepts of today’s fundamental issues. They associate knowledge of programming with reading, writing, and knowledge of arithmetic, since everyone uses them across all subject areas as well as in daily life. Similarly, Kalelioğlu and Gülbahar (2014) argued that everyone should learn programming concepts appropriate to age level because contemporary issues require them to possess some degree of programming skill. There have been many efforts to introduce concepts related to programming to all ages through integration into school curriculum, establishing worldwide programming activities, or organizing out of school activities. Some of the activities organized globally are “Hour of Code,” “EU Code Week,” “Bebras” and locally “MotherCoders,” “Black Girls Code” and “KodlaManisa.” All of these aim to promote programming for individuals at various ages. Previous studies have shown that people have become informed about programming from unanticipated sources such as from school setting to online courses, and even in art schools (Guzdial & Disalvo, 2013).

Different terms are used in the literature to refer to programming education such as computing education (Guzdial, 2015; Guzdial & Disalvo, 2013), coding education (Duncan et al., 2014), and introductory programming education (Powers et al., 2006). In addition, there are other terms derived from the growing interest towards programming such as Computational Thinking (Wing, 2006), and Computational Participation (Kafai & Burke, 2013b). The diversity of terms “similarly titled but seemingly different” have made the issue difficult to understand and confused the minds of students, parents, and administrators (Pears et al., 2007, p. 206). Although there could be some small differences and one term could be a subset of another (i.e. coding is a subset term of programming or computer science), their intersection point is the importance of programming education for everyone. There could be differences in the extent and purpose of education according to the level and age of learners. For example, Gujberova and Kalas (2013) argued that the aim of the professional programmer is not the same as primary and secondary school students who learn programming, and they coined another term *educational programming* in order to refer to the programming done by primary and secondary school students. The examination of programming in history proves that the definition is subject to change and new terms may emerge with advancements in technology, software, hardware and different usage of computers over time (Blackwell, 2002; Gujberova & Kalas, 2013).

Programming and *coding* are regularly considered to refer to the same thing and used synonymously (Duncan et al., 2014). However, in some studies, *coding* was used instead of *programming* due to fact that the term *coding* is mostly used by organizations that aim to teach programming to children and a “popular word” nowadays. *“Coding”* is defined as translating a designed program as a solution of a problem into programming expressions via typing, combining or entering them (Duncan et al., 2014). Programming could be defined as the "process of writing programs" (Guzdial, 2015, p. 2). By definition, the process of programming includes various steps such as building a logic to solve a problem, writing programming expressions in a certain computer input language, transforming those into machine language via compiling, and then testing and debugging (Papadopoulos & Tegos, 2012). During programming, those steps of instruction are defined for the computer to solve specific problems. In the current study, the term *programming* refers to the *basics of programming*.

Although learning programming is somewhat of a new trend nowadays, it was also popular from the late 1970's to the late 1980's with the advent of personnel computers (Gujberova & Kalas, 2013; Resnick, Maloney, Hernández et al., 2009). There were many attempts to introduce students to programming concepts at that time, but such efforts did not last for a long time and the purpose of computer has been changed until today. Due to the fact that only a relatively small number of people were interested in programming as a technical skill (Resnick, Maloney, Hernández et al., 2009), most efforts at teaching programming were aimed at preparing individuals for the software industry and the majority of research related to programming were conducted mostly with undergraduate students (Guzdial, 2015). In the following sections, firstly, studies in programming education for children are extensively addressed. Then, the importance of programming education and challenges of learning programming are presented. Then, issues related to how those challenges could be overcome are addressed, and finally, environments and tools developed for teaching programming are presented.

2.2.1 Programming Education for Children

Programming for children has gained worldwide interest in recent years. Many countries have been trying to transform their current ICT courses or to establish new courses in order to teach programming to children (Gal-Ezer & Zur, 2013; Gujberova & Kalas, 2013; Kalelioğlu, 2015; Menekse, 2015; Tenenberg & McCartney, 2014; Yadav, Gretter et al., 2017). Concepts related to programming, computational thinking and computer science are about to take place in K-12 curriculum. It has been reported that many countries such as Austria, Germany, Mongolia, Netherlands (Schulte & Carsten, 2013), Australia and Mexico (Yadav, Gretter et al., 2017), United States (Menekse, 2015), United Kingdom (Brown, Sentance, Crick, & Humphreys, 2014), Israel (Gal-Ezer & Stephenson, 2014) and Turkey (Kalelioğlu & Gülbahar, 2014) have been either trying or are planning to implement programming into their K-12 curriculum, since they have realized the importance of teaching programming at young ages (Gal-Ezer & Zur, 2013).

Previous studies argue that learning programming is essential for all ages (Sauppé et al., 2015) and that everyone needs to know programming to some extent (Guzdial, 2015; Kafai & Burke, 2014). With the growing interest towards programming education for children, some questions have arisen and discussions come into play; two of which are how to teach programming to students and how to broaden participation (Menekse, 2015). Others are around what age and in which phase of education (i.e. primary or secondary) students should learn programming (Duncan et al., 2012; Sauppé et al., 2015; Schulte & Carsten, 2013). Another is why students should learn programming at all. That is all members of society need to be informed about the importance of learning programming (Gal-Ezer & Zur, 2013).

Pedagogical aspects of learning programming comes into play in order to answer these questions. The effective pedagogy of learning programming is dependent on the phases of education and characteristics of learners (Waite, 2017). There are, of course, different scholarly views on these issues. Sayın (2017) investigated which topics should be taught to children at a specific age; arguing that it should be determined according to children's developmental psychology. It was reported that programming could be taught to children at the pre-school level by introducing the basics of algorithm, and the use of computer programs including programmable smart toys, games and so on. For students at the primary school level, topics such as problem solving, algorithm and programming should be taught including the use of information and communication technologies. For students at the high school level, advanced level programming topics such as robotics, web-based and mobile programming could be taught. Waite (2017) reviewed a number of studies related to teaching computer science in order to understand pedagogical underpinnings applied in those studies and she identified four different context for teaching computing as physical computing, game making, unplugged and cross-curricular activities, and she offered pair programming, problem based learning, peer support and apprenticeship for increasing students' engagement to learning programming.

Approaching this from a different perspective, Duncan et al. (2014) looked at the question of at what age a student should learn programming, arguing that developmental psychology, gender, and the relationship between programming and other disciplines could have an effect on what age a learner should learn programming topics. For example, if middle school girls have no natural passion for programming, then pair programming and collaborative activities could be designed. At the end of their study, Duncan et al. (2014) provided a set of levels, from 0 to 4, that correspond to age ranges in order to provide an idea of ability level and learning outcomes based on their heuristics. They also matched each level to specific programming tools according to their characteristics. However, Kalelioğlu and Keskinkılıç (2017) argued that children could learn the basics of programming without using a computer via unplugged activities, as well as block-based programming tools on computers according to their accessibility. For the more advanced levels, text-based and robotic programming tools could be preferred. Using programming in other disciplines such as mathematics or music is another method of teaching programming similar to Waite's (2017) cross-curricular activities. Lastly, activities for teaching programming could be categorized in three levels according to curriculum of CSTA (Duncan et all, 2014). It was suggested that topics in the first level (K–6) should focus on sequencing, and topics in the second level (grades 6–9) should focus on collaborative activities with enough support. Activities in the third level should focus on algorithmic problem solving by using the basics of programming.

There is "limited clear empirical evidence" to reveal the pedagogical aspects of teaching programming for children since studies focus on just coding and content, and conducted with older learners (Waite, 2017, p.41). More research needs to be conducted as Grover and Pea (2013) emphasized the importance of conducting research in this area in order to understand theoretical and practical advantages of programming for children. Especially educational researchers and computer scientists should collaborate on conducting research in order to fill the gap on this interdisciplinary field (Franklin, 2015). A review of the literature shows a lack of studies on programming education for children, with much of the research conducted at the higher education level (Grover & Pea, 2013; Gujberova & Kalas, 2013; Knobelsdorf & Vahrenhold, 2013; Schulte, Clear, Taherkhani, Busjahn, & Paterson, 2010), which might be due to being a relatively young field of research (Gal-Ezer & Zur, 2013).

2.2.2 Importance of Programming Education

The importance of programming education for children is multifaceted. There are different perspectives and scholars who argue that children should learn programming at an early age. They argue that children need to head towards programming and that they should develop a positive attitude about programming (Duncan et al., 2014). In this way, they would carry out the requirements of a computing intensive world and be better prepared for the future careers of the 21st century (Knobelsdorf & Vahrenhold, 2013). At this point it would be helpful to mention a growing demand for programming-related jobs (Robins et al., 2003) and teaching programming as a good opportunity for children to then follow it in their future life (Menekse, 2015). Guzdial and Fisher (2014) argued that learning programming informs children about the process of developing software and increases their awareness and understanding of the weaknesses of software.

In addition to children's future career options, programming is useful for children in order to teach and promote various thinking skills (Akcaoglu & Koehler, 2014). For example, Guzdial and Fisher (2014) argues that learning programming promotes logical thinking, whilst Grover and Pea (2013) claimed that it yields learners the basics of 21st century skills like critical thinking. As a pioneer advocator of programming education at early ages, Papert (1980) asserted that children could improve their thinking skills by analyzing problems, splitting problems into simpler forms to cope with, building code expression, and testing and debugging in order to find the solution to a problem situation. The value of learning programming is not limited to those aimed at a career in computing. Regardless of future potential career plans, children should learn programming to some extent in order to think critically and to express their creative ideas via programming (Resnick, Maloney, Monroy-Hernández et al., 2009). Children have the chance to express themselves via programming and programming is a popular method in which they could use it as a tool for thinking (Guzdial & Disalvo, 2013).

In the literature, it was asserted that learning programming brings some cognitive skills to children such as reasoning skills, planning skills, and problem-solving skills (Gomes & Mendes, 2007; Pears et al., 2007), as well as computational thinking skills (Grover & Pea, 2013). Previous research has shown that children who had previously learned programming outperformed in terms of cognitive skills when compared to those who had not (Akcaoğlu, 2013). Computational thinking is another important cognitive skill that programming plays an important role in promoting (Duncan et al., 2014; Grover & Pea, 2013; Kafai & Burke, 2013a; Menekse, 2015). Computational thinking was first popularized by Wing (2006) who defined it as "solving problems, designing systems, and understanding human behavior by drawing on the concepts" (p. 33) of computer science such as debugging, remixing or iteration. It was argued that not only those interested in computer science, but that everyone needs computational thinking skills (Herro, Gardner, & Boyer, 2015). People might use computational thinking skills in daily life (ISTE, 2015); however, an increase is necessary for all in the awareness of what computational thinking is and how best to use it. With the help of computational thinking, it is possible to overcome problems that could otherwise be difficult to solve at the beginning (Wing, 2006). Developing computational thinking is concerned with the learning of programming, but is not wholly limited to it (Kalelioğlu & Gülbahar, 2014; Yadav, Stephenson, & Hong, 2017).

The literature shows that learning programming promotes computational thinking skills (Boyer, Herro, & Gardner, 2014; Buffum et al., 2015; Kafai & Burke, 2013a; Kalelioğlu & Gülbahar, 2014). It was found that computational thinking skills of 5th grade students increased at the end of programming education with Scratch (Kalelioğlu & Gülbahar, 2014). Boyer et al. (2014) aimed to develop CT skills of students through a pilot study by developing mobile applications using App Inventor. At end of their study, they found that students had a positive attitude towards programming and realized that they could in fact achieve programming.

To summarize, Guzdial (2015) listed six reasons to teach programming to children. Firstly, learning programming increases the possibility of following a computer-related career since programming education at early ages can improve motivation to follow. Secondly, learning programming helps children to understand the world better since computing is omnipresent in today's world. It helps children to familiarize themselves with the foundational principles of interfaces and technologies that they will encounter in daily life. Thirdly, learning programming enhances computational thinking, which is helpful in transferring knowledge of computing to daily life. This approach might increase achievement in life. Other reasons are that learning programming increases computational literacy and productivity. People with the ability of reading and writing computer programs could potentially then go on to produce according to their needs rather being a consumer. Finally, everyone should have the chance to learn programming. Teaching programming at early ages could thereby broaden participation, regardless of gender or race or other demographic variables.

2.2.3 Tools for Teaching Programming for Children

Learners of programming come from a wide age range, from the primary school students to undergraduates in higher education (Guzdial, 2004; Kalelioğlu & Gülbahar, 2014). Although the age of learners and their purpose of learning programming could change from learner to learner, programming is a complex topic that is considered difficult to learn, especially for novices (Guzdial, 2004; Kelleher & Pausch, 2005; Saeli et al., 2011; Schulte & Carsten, 2013). There have been many studies that have investigated why programming is considered so difficult by students (Gomes & Mendes, 2007; Kurhila & Vihavainen, 2015; Özmen & Altun, 2014). In one such study, Esteves, Fonseca, Morgado, and Martins (2008) stated that novice learners have difficulty in understanding basic programming concepts due to the nature of traditional courses based on lecturing and programming syntax. In traditional courses, students usually learn programming theoretically by memorizing syntax and script without ever implementing them and not applying them in different contexts and problems (Esteves, Morgado, Martins, & Fonseca, 2006; Gülmez, 2009). Moreover, students cannot understand abstract concepts since they do not have real life equivalence and therefore cannot apply them to problem situations (Gomes & Mendes, 2007). Rather than learning programming syntax and semantics, students should learn how to solve problems and produce algorithms from the outset (Esteves, Fonseca et al., 2008). Semantics and syntax of programming languages preclude learners to focus on the important points of programming (Guzdial & Guo, 2014).

The literature explains the reasons why programming is considered hard for novices, including children, as follows (Blackwell, 2002; Maloney et al., 2008; Pears et al., 2007; Resnick, Maloney, Hernández et al., 2009):

- Most programming languages are inappropriate to introduce the basic concepts of programming to novices since they were not generally developed for educational purposes. Most are too difficult to use and understand since they require special concepts, features and notations. Syntax, nature and notational machine of programming languages is often complex for novices. Error and warning messages could be meaningless for novices;
- Activities designed (i.e. listing prime numbers or making simple line drawings) for teaching programming do not draw the attention of learners as they are not directly related to their interests;
- Learners lack motivation and engagement due to traditional and non-meaningful methodologies.
- Lack of guidance on supporting learners' deep understanding of programming concepts.

There are some misconceptions about learning programming derived from the reasons listed. Some of them are myths expressed by those who themselves lack understanding. Some misconceptions are that "programming is too hard for children," "programming is mostly done in isolated place from the community," "it is an asocial and competitive activity," "programming is mostly for boys and related to a profession," and "programming is a boring activity" (Gujberova & Kalas, 2013; Porter, Guzdial, McDowell, & Simon, 2013).

Many tools are designed to satisfy the needs of novices for the aforementioned reasons (Duncan et al., 2014; Pears et al., 2007; Sauppé et al., 2015). Their aim is to address the difficulties that students have during the learning process (Gomes & Mendes, 2007). Tools for children are often very creative, easy to use and also easy to adopt. Accordingly, students' success and motivation are aimed to be increased. What these tools actually do can be listed as visualization of abstract concepts, contextualization of programming, using games to motivate students (Esteves et al., 2008), using smaller segments of code to practice, and using graphical and narrative tools to eliminate syntactical errors (McWhorter, 2008).

There is a noticeable amount of study on such tools and their affects. Recent studies have shown that using tools in learning programming has an effect on programming achievement (Gülmez, 2009). One advantage of using such tools in teaching programming is that they could enable students to better understand abstract concepts (Esteves et al., 2006; Tekdal, 2013). Esteves et al. (2008) argue that visualization helps students understand the concepts of programming better since physical, spatial or visual representations are easier to retain, and that learners are provided with immediate feedback, with the help of visualization, so that learners try to find solutions to their own problems. Program visualization and animation tools could enable learners to see the execution of lines of program code by the application of graphical effects. Therefore, students better understand program code and how program execution works, which are the some of the most difficult concepts encountered during the learning process (Tekdal, 2013). Some tools also help students to increase their motivation since they add that element of fun to the learning process (Papadopoulos & Tegos, 2012).

Various learning strategies and approaches are employed in tools such as storytelling, visualization techniques, and pair programming (Salleh, Shukur, & Judi, 2013). Therefore, tool features and capabilities can vary, although they all have the same mission. In their study, Powers et al. (2006) categorized the tools as narrative, visualization, flow-model, and specialized output realization tools. Narrative tools such as Alice allow learners to create stories via programming. Visualization tools allow learners to create programs via drag and drop code blocks. Flow-model tools enable learners to create programs connecting the elements of tools to each other. Specialized output realization tools enable learners to take feedback in non-textual forms such as designing a concrete activity with a robot and giving commands to that robot.

To summarize, all of the developed tools starting from Logo are primarily aimed at making programming easy for students to understand, and to increase motivation although in each one, programming is presented in different formats (Bishop-Clark, Courte, & Howard, 2006; Ingram-Goble, 2013). Some of the important tools are briefly described as follows.

Scratch

Scratch is a programming tool used by young people to create a wide variety of projects from video games to interactive stories and newsletters, from science simulations to birthday cards, and many others beside. It was developed by a team working at the MIT Media Lab and the Scratch Website was first published in 2007. With its launch, it offers learners a community of practice to share and discuss their projects as well as source code of other projects for the purposes of remixing them (Resnick, Maloney, Hernández et al., 2009). According to statistics from the Scratch website, there are so many registered users and projects, and the community continues to grow all the time. Although Scratch was designed for ages 8 to 16; it has been used by those from K-12 to even universities as a first step towards programming. With growing interest, it has been translated into more than 40 languages and used in over 150 different countries (Scratch – About, 2017). Figure 2.1 illustrates a Scratch programming environment with some code blocks.

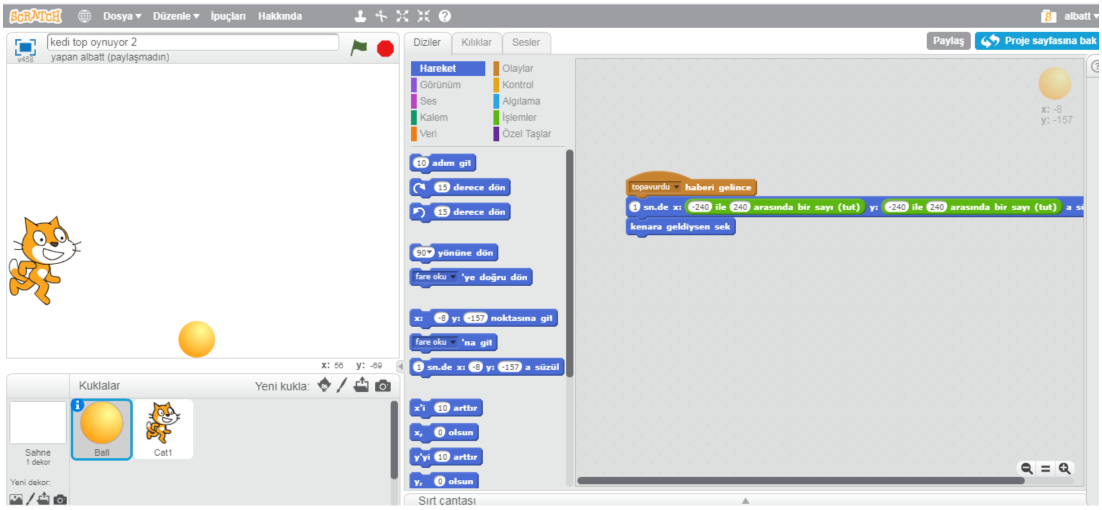


Figure 2.1 – Scratch programming environment

Scratch could be considered as a programming language that aims to improve the programming skills of young children with the help of its easy-to-use and attractive interface. It is so easy to create projects by integrating predefined code blocks that prevent learners from making syntax errors since the shapes of blocks are like a “jigsaw puzzle” that can only be integrated in a certain way. Users are able to integrate blocks in different sequences and combinations without worrying about the syntax and notation that are features of a traditional programming language (Resnick, Maloney, Hernández et al., 2009). As a result, the student’s code is free of syntax errors (Papadopoulos & Tegos, 2012). Users can program characters called “sprites” in Scratch on the screen called as “scenes.” They may also personalize their projects by adding background to their scene and new characters (sprites); and in this way, users can work on projects that are more meaningful to them. Scratch was developed with the aim of making programming more “tinkerable,” “meaningful” and more social, as well as being easier to use (Resnick, Maloney, Hernández et al., 2009).

With its wide acceptance among learners, Scratch was modified into different versions in order to add new features, or adapt it to other programming languages. For example, Scratch for Second Life (S4SL) was modified by a member of the Scratch team. It is used for creating scripts in Linden Scripting Language of Second Life by using the known code blocks of the original Scratch, and with a layout and logic similar to that of Scratch. Another Scratch modification is Scratch for OpenSim (S4OS), which is similar to S4SL. It creates scripts in order to program objects in OS by using the original Scratch software logic and interface.

Alice

Alice is a programming environment that enables learners to create animations, interactive stories and simple games in 3D. It was developed by the Stage 3 Research Group at the private Carnegie Mellon University in Pittsburgh, Pennsylvania. Students program built-in objects in a virtual world without the need to write source code (Kunkle, 2010). It enables students to build virtual worlds by dragging and dropping objects, methods, and control structures within an editor (Pausch, Dann, & Cooper, 2000). Creating virtual worlds in Alice is easy for students. *Figure 2.2* shows a screenshot from the Alice programming environment.

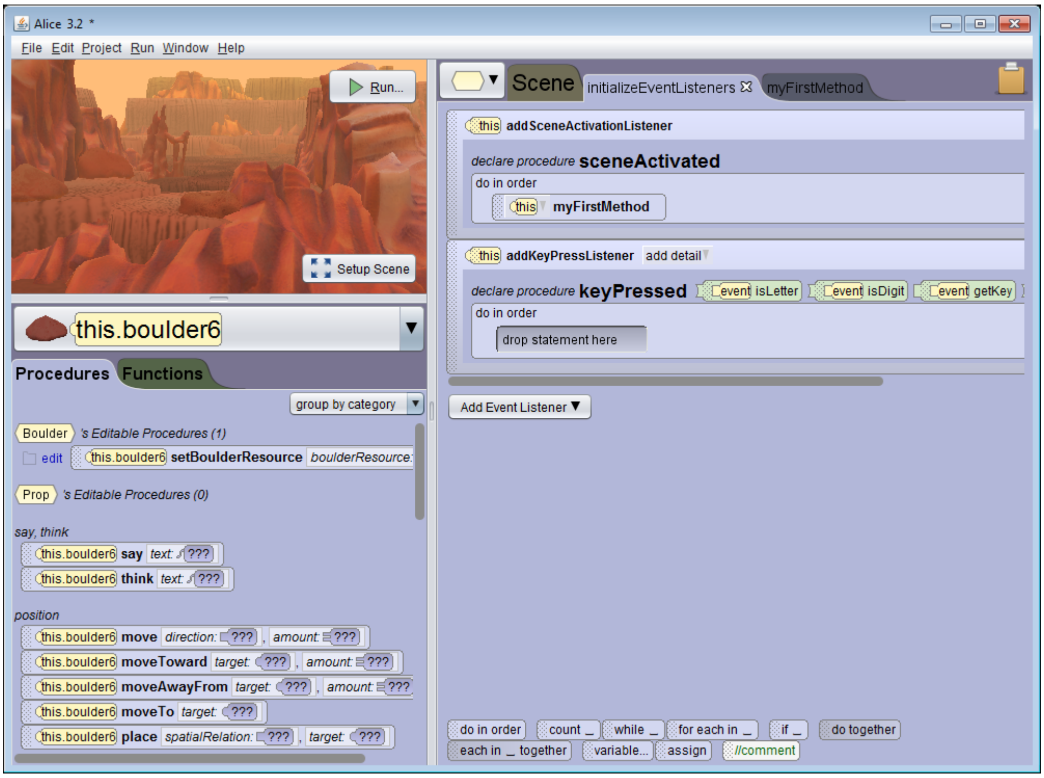


Figure 2.2 – Alice programming environment

Alice enables users to create animations, videos and interactive games in order to tell or share a story by only modifying objects and movements of a camera (Pellas & Vosinakis, 2017). It is generally used for meeting the needs of novice learners to learn object-oriented programming by controlling the appearance and/or behavior of objects in the environment via programming. The aim of Alice is to attract the interest of users from underrepresented groups such as women in order to draw their attention to programming and motivate others (Pausch et al., 2000). In Alice, users can create animated scenarios based on storytelling via integrating code blocks. Drag and drop code blocks prevent learners from making syntactic errors similar to Scratch.

Karel the Robot

Karel the Robot is a robot simulator that introduces students to the basics of programming using a language similar to Pascal. In Karel the Robot students navigate Karel and interact with other objects in its world via text-based instructions given to Karel. Students can then observe the movements of Karel, change them where needed. Students can build walls and add other robots in a 2D virtual environment. They can also analyze their programming via executing the instruction and following the behavior of the robot to see whether or not Karel functions as intended (Papadopoulos & Tegos, 2012).

Greenfoot

Greenfoot is a 2D interactive visual environment in which learners change the behavior and appearance of objects in a standard textual Java code, with some of the complexities removed (Kölling, 2010). The main aim of Greenfoot is to teach object-oriented programming in Java. It enables learners to experiment with real code in flexible scenarios, whilst providing visual feedback. Greenfoot makes it possible to understand object-oriented concepts in a simple way (Papadopoulos & Tegos, 2012). A screenshot from Greenfoot is shown in Figure 2.3.

Other Tools

There are many tools developed in order to help teach programming to learners (Gross & Powers, 2005; Kelleher & Pausch, 2005; Powers et al., 2006). Some other tools are robotic kits and tangible media such as Lego Mindstorms, Makey, and Arduino Kits. In contrast to the tools which confine learners to the constraints of a computer screen, these kits could be more meaningful for children since they are based on tangible, physical, real world objects which can be programmed to operate certain predefined functions.

Figure 2.3 – Greenfoot programming environment

Lego Mindstorm is the most widely known and used (Powers et al., 2006). *Figure 2.4* shows an example of Lego Mindstorm robot. The kit includes many pieces such as Lego bricks, a microcontroller, sensors, as well as motors and gears. The bricks can be joined in many combinations in order to accomplish a meaningful task with the help of actuators and sensors. For example, learners may build a fire truck that can detect and extinguish fires. Learners may be more motivated and engaged since they study using such hands-on activities and solve problems in context by testing their programs in a more concrete way through interaction with fellow students (Powers et al., 2006).

Figure 2.4 – A Lego Mindstorm robot

2.3 Research on Programming Education in 3D Virtual Worlds

Virtual worlds provide educators with the ability to create rich and compelling 3D context for students' learning as well as communication tools to support discussion and collaboration (Delwiche, 2006; Esteves, Fonseca et al., 2008). They allow learners to connect, interact and collaborate with each other simultaneously in the same space (Girvan, Tangney, & Savage, 2013). It is possible to edit the same objects, and attach scripts to those objects synchronously and asynchronously in VWs since the objects are persistent.

In spite of the affordances of VWs as new technological tools, their use for educational purposes could not go beyond the replication of real world examples (Winn, 2005) such as organizing lecture theatres or virtual university campuses. On the contrary, their affordances could be made use of for new educational opportunities (Girvan, Tangney, & Savage, 2010). At this point, the use of VWs could bring about new opportunities to students in regards to improving their programming performance. Educators could see VWs as a new way to overcome difficulties encountered by novice learners while teaching programming (Esteves et al., 2011; Esteves, Fonseca et al., 2008). Previous studies have argued that potential benefits of using VWs for programming education are (a) VWs could improve students' cognitive skills (computational and higher order thinking skills), (b) VWs could enhance reflective learning and facilitate the transfer of programming knowledge to real life examples, (c) VWs could yield a virtual community to study together and thereby improve the collaborative skills of students, (d) VWs could yield a wide range of context for different types of activities that could attract learners of any age, and (e) VWs could yield a place for learners to exhibit and share their artefacts (Kahn, 2007; Pellas & Peroutseas, 2016). In spite of these potential benefits, the use of VWs in teaching programming to novices has not yet been explored that much in the literature (Hulsey et al., 2014; Pellas, 2014a; Seng & Edirisinghe, 2007). In this section, studies related to use of VWs in programming education are addressed.

In one study by Esteves et al. (2008), the researchers studied with a group of undergraduate students working as pairs in SL in order to develop projects by using SL's built-in language, LSL. At the end of their study, they found that when the working area of the students was vast and scattered, instructors had difficulty in moving from one area to another within the environment. Additionally, it was difficult for instructors to follow the progress of students since the students were continuously working in the environment. Another problem was communication in the public area; when the students spoke in public, it was hard to figure out who was speaking. Other difficulties which students encountered were using the SL interface, understanding LSL, and interpreting errors while compiling the scripts. Findings of the study by Esteves et al. (2008) indicated that most students had a positive or neutral attitude towards the immersive nature of SL; however, a few students had a negative attitude since the environment was complex and inadequately serious, according to them.

In another study, Esteves, Fonseca, Morgado, and Martins (2009) conducted a four-phased action research in order to understand whether or not SL could be used in programming education for learners. At each phase they repeated the same study with different participant groups with enhancements based on results of the previous phase. Students completed activities such as building and programming a car, robot or dog to perform specific tasks. Students completed the tasks via programming in LSL of SL. At the end of the study, the researchers concluded that activities prepared in the environment should be as visual, interesting, meaningful and appropriate as possible, as well as communication within the environment should be public for general expressions and private for students' needs and to raise doubts about a specific topic.

In another study, Esteves et al. (2011) studied with students from diverse education levels. Students were required to undertake a project by using LSL of SL in pairs. At the end of the study, the researchers suggested that communication within the environment should be text-based and private rather than voice-based and public. This opportunity would enable students to obtain help at any time from their instructors. However, instructors had difficulty in giving immediate feedback to their students; whereas students would like to reach their teachers and get help as soon as possible. Students preferred the visual projects compared to nonvisual since such projects were more meaningful and engaging for them. In visual projects, the students built an object and added script to it within the environment. Students felt inspired with the immersive features of the environment and not isolated, as opposed to traditional programming courses without social contact. One significant problem experienced by some of the novice learners was in the use of interface of SL and LSL during programming.

Seng and Edirisinghe (2007) investigated the use of SL as a learning environment in computer science due to its inherent lack of use in those subjects. They provided simulated code segments to students and asked them to match simulations with correct code segments. They concluded that using SL increased student engagement, the effectiveness of peer teaching, and the attractiveness of the learning activities. The requirements of SL, the difficulty of LSL, the lack of a suitable SL compiler were all seen as disadvantages of using SL. In another study, Rico, Martínez-Muñoz, Alaman, Camacho, and Pulido (2011) prepared an introductory programming course using a platform named “VLeaF” for high school students in order to make programming concepts more attractive. The system was based on OpenSim and a web portal in which students could program 3D objects in LSL and find the necessary materials. Both students and teachers found the system easy to use and the authors reported that scripting within the VW increased their motivation, collaboration and level of cooperation. Students also liked scripting in VW because of the direct interaction with 3D objects and the ease of communication with other users. However, students expressed that they did not feel so free in the environment.

Hulsey et al. (2014) organized a weeklong camp named “Camp CyberGirls” in order to introduce the basic concepts of programming to 16 female high school students. In doing so, they prepared a virtual environment consisting of 10 tasks including modeling and scripting activities; for example, modeling and scripting a sliding door. Modeling and scripting were performed using the environment’s functions such as writing scripts in LSL. Results of the study indicated that although it was more complicated for students to complete tasks compared to other programming tools such as Alice or Scratch, students had the chance to perform a wide variety of learning activities. Moreover, studying and completing tasks in VW was a source of motivation for the students and they realized a high degree of satisfaction. They also reported that they would like to improve themselves for potential computer-related careers. Some students suggested that the camp hours could have been longer, whilst some reported that writing real code was in fact difficult for them.

Writing scripts in LSL is difficult for students, especially for novices. Therefore, Rosenbaum (2008) developed a Scratch-like tool known as S4SL, that could be used for creating scripts for LSL in a basic manner. S4SL was piloted with 10 inexperienced adult student users in order to discover the appropriateness and usability of S4SL in SL. The students developed virtual artefacts in the virtual world and added interactivity to those artefacts by programming them via S4SL. At the end of the pilot study, it was reported that the students were able to easily build objects and create scripts for them. The ease of being able to create scripts for objects made the students feel a sense of empowerment. It was reported that students were satisfied since it was very simple to create script for virtual objects via S4SL. However, technical challenges and unexpected situations caused some of the students to feel frustrated. Girvan (2014) argued that learners have the potential to engage in constructionist learning experiences with the help of virtual worlds as they provide learners with a blank canvas to fill in through their knowledge and programming skills. Virtual worlds also allow learners to share their artefacts with others, and in this way, it is possible for them to explore and test their understanding through collaborative construction.

Girvan, Tangney, and Savage (2010) proposed the use of "SLurtles" in Second Life or OpenSim, in order to teach geometry as well as abstract concepts of programming by collaboratively creating 3D objects within the environment. They designed a 3D object named "SLurtle" similar to Papert's turtle, which could be programmed with Scratch for Second Life in a 3D environment. Students programmed them to create artifacts such as a house, tree, piano, and a bowling alley by using S4SL. At the end of their study, the authors argued that SLurtles with S4SL could be used to lower the floor and widen the walls. Users could create a wide variety of shareable and consistent artefacts within the environment.

In another study by Girvan, Tangney, and Savage (2013), an exploratory case study was conducted with 24 graduate students with little or no programming experience. The participants studied in pairs for a period of four weeks and constructed interactive objects such as a playable piano. The aim of the study was to understand whether or not SLurtles could be used during the construction of shareable artefacts. The study's results showed that they were easy to use and powerfully expressive tools which enabled the learners to create a wide variety of artefacts within the environment in parallel with constructionist learning, even for novice learners. Moreover, SLurtles supported students' thinking processes, and provided them with visual feedback; making it possible to easily understand abstract concepts. They were also helpful in engaging students with the learning objectives since the students argued that studying with them was funny and the authors argued that students gained a high sense of satisfaction. On the other hand, S4SL limited the variety of artifacts created with SLurtles due to the limited complexity of code offered by S4SL. Transferring code from S4SL to the VW environment, creation of modeling 3D objects, generic skills in SL such as using camera control, avatar movement and communication tools were seen as difficult for some students at first. Finally, they reported that collaboration during programming SLurtle was problematic. Students tried to collaborate their code by adding the screenshot of the code on an object's texture in order to share it.

Sajjanhar and Faulkner (2014) also studied with 12 graduate students in order to investigate the use of SL in conjunction with S4SL for the teaching of basic computer programming concepts. At the end of their study, most of the students found SL easy to use and learn. Some of the students argued that learning programming in SL facilitated the learning process and the potential for instant feedback was deemed helpful. Apart from a few students, most appreciated the use of SL in learning basics of programming. A few students defined the VW as a distraction because of in-world entertainment.

In another study, Pellas (2014b) used OpenSim and Scratch for OpenSim (S4OS) to teach basic programming commands to high school students. They provided a *mind trap puzzle* to students in the VW and asked for them to solve it via S4OS. At the end of the study, the researchers concluded that students' motivation, engagement, collaboration with others and achievement had increased. S4OS in OpenSim provided students with a unique learning environment to create *objects-to-think-with* by using programming commands. They also reported drawbacks of using VW as students' misuse, distractions in the VW, difficulties of using and navigating through avatars in VW and the coordination of activities.

Pellas and Peroutseas (2016) conducted a mixed-method study with 56 high school students in order to understand how students engaged in an introductory programming course in SL. Students studied to create Greek letters in a 3D environment for a period of six weeks (eight hours laboratory time plus eight hours distance) with the help of S4SL. The study's results revealed that students' engagement, attention, and interest in programming via S4SL in the 3D environment were positively affected. Moreover, the authors argued that they found evidence of increased motivation, collaboration and achievement in programming concepts. The presence of a teacher and a source of feedback for the students enhanced their level of engagement. In another study, Pellas and Vosinakis (2017) conducted a study with 28 students (aged 14-15 years) in order to investigate the motivation of students using S4SL in a 3D environment. The students studied in pairs to complete tasks collaboratively. The study's results indicated that the use of VW and S4SL positively affected the students' motivation and participation.

Dreher, Reiners, Dreher, and Dreher (2009) complained that information science courses do not motivate students intrinsically and that they are far from the practical application of industry. They argued that "learning programming is not linked to real world relevance" (p. 213) although skills in programming are a key aspect of industry. Therefore, they stressed the importance of VWs that enable learners to create 3D objects and program them in VWs, which might be pedagogically useful. Moreover, visualization of programming concepts in a 3D environment, testing code in context and working collaboratively within the environment are the most prominent affordances of VWs in learning programming. Last but not least, learning programming in VWs could be more interesting and practical when compared to learning by traditional methods. On the other hand, the use of VWs could bring about some challenges since they require high capability computers, high level bandwidth and high usage quotas.

2.4 Summary and the Research Gaps

Programming for children has gained worldwide interest in recent years. It has been argued that learning programming is essential for all ages (Sauppé et al., 2015) and that everyone needs to know programming to some extent (Guzdial, 2015; Kafai & Burke, 2014). Therefore, there have been numerous efforts to introduce the basics of programming to children via designing curricular and extra-curricular activities in schools and after-school programs outside of schools (Kafai & Burke, 2014). In addition, in recent years many countries have been trying to transform their current ICT courses or establish new computing-related courses to teach programming to children (Gal-Ezer & Zur, 2013; Gujberova & Kalas, 2013; Kalelioğlu, 2015; Menekse, 2015; Tenenberg & McCartney, 2014; Yadav, Gretter, Hambrusch, & Sands, 2017).

Programming is a complex topic that is considered difficult to learn, especially for novices (Guzdial, 2004; Kelleher & Pausch, 2005; Saeli et al., 2011; Schulte & Carsten, 2013). In order to reduce the difficulties faced by students during the learning process and to increase their success and motivation (Gomes & Mendes, 2007), many tools and environments have been developed (Duncan et al., 2014; Pears et al., 2007; Sauppé et al., 2015). What these tools actually do can be listed as visualization of abstract concepts, contextualization of programming, using games to motivate students (Esteves et al., 2008), using smaller segments of code to practice, and using graphical and narrative tools to eliminate syntactical errors (McWhorter, 2008).

VWs, with their characteristics and affordances, could be utilized for teaching the basics of programming to children. At this point, the use of virtual worlds could bring about new opportunities for children with regards to improving their programming performance. Previous research investigated the use of VWs in terms of overcoming learners' difficulties encountered during the learning of programming (Esteves et al., 2011; Esteves, Fonseca, et al., 2008). However, studies concerned with the use of VWs in teaching programming have been limited and mostly conducted with participants at the high school, university and graduate levels (e.g., Girvan et al., 2013; Hulsey, Pence, & Hodges, 2014; Pellas, 2014a; Seng & Edirisinghe, 2007). Studies examining its use in different educational programs and comparing and contrasting the results have not been found in the literature; therefore, there is a need to understand the use of virtual worlds in teaching the basics of programming to children through different educational programs.

CHAPTER 3

METHODOLOGY

This chapter is devoted to the method in which several scientific steps and procedures were followed throughout this study. In this chapter, the reasons of how and why particular methods were chosen are broadly justified and explained. Firstly, the research design of the study and pilot studies are elaborated on; then, the design and development issues of 3D virtual learning environment are explained. Thirdly, the selection of cases and participants are presented. Finally, data collection methods and procedures, data analysis, the researcher's role and trustworthiness of the study are described.

3.1 Purpose of the Study and Research Questions

The purpose of this study is to understand the use of virtual worlds in teaching basics of programming for children as well as revealing the main points of using virtual worlds. In a broader context, this current study aims to examine the perceptions of participants about ease of use and perceived usefulness of VWs in programming education, the affordances and challenges of using virtual worlds, issues and strategies for the group study, design issues of different educational programs in VWs, factors that affect satisfaction, and avatar issues. This study also aims to investigate the use of VWs in three different educational programs; curricular, extra-curricular, and after-school programs. The final purpose of this study is to reveal the similarities and differences across the cases.

The main and sub-research questions of this study are as follows:

How could virtual worlds be utilized in programming education for children from different educational programs?

a. To what extent do participants perceive the ease of use and usefulness of SDP?
b. What are the affordances and challenges of using virtual worlds in programming education for children?
c. How does avatar representation affect the experience of participants?
d. How do the issues and strategies about group study in SDP affect the experience of participants?
e. What are the factors affecting participants' satisfaction in SDP?
f. What are the issues and strategies for the design of SDP?

3.2 Design of the Study

There are four ways of knowing something; experiencing via sensory, agreement with others, using logic, and using scientific methods (Fraenkel, Wallen, & Hyun, 2012). In order to reach the most reliable and accurate knowledge, scientific methods need to be followed. Research consists of systematic steps used to collect and analyze data in order to reach the required level of knowledge about a topic or issue (Creswell, 2012; Merriam, 2009). Quantitative and qualitative are two types of research. While quantitative studies investigate the relationships between variables by numerical attributes of the data, qualitative studies are concerned with the quality of relationships or situations from the viewpoint of participants by using words and narratives (Fraenkel et al., 2012). These differences are due to the fact that both are based on two different paradigms. That is, quantitative research is based on the philosophy of positivism, whilst qualitative research is based on the philosophy of postmodernism. Creswell (2012) suggests using quantitative research in order to provide broad explanations to a large number of people by assessing the impact of variables on an outcome, and suggests using qualitative research in order to provide explanations based on participant perspectives by obtaining detailed information from a few people or research sites.

Merriam (1998) defined qualitative research is an “umbrella concept covering several forms of inquiry that help us understand and explain the meaning of social phenomena with as little disruption of the natural setting as possible” (p. 5). She also argued that there were some other terms used by researchers that refer to qualitative research. The purpose of qualitative research is to explore in depth the central phenomenon of a study (Creswell, 2012). Although different terms have been used, the four common characteristics of qualitative research are that: (a) the focus is on process, meaning and understanding; (b) the researcher is the primary instrument of data collection and analysis; (c) the process is inductive; and, (d) the process is richly descriptive (Merriam, 2009).

Qualitative approach was chosen as the method for the current study in order to answer the research questions. Firstly, there is too little known about the use of virtual worlds in programming education for children, and therefore it needs to be explored in a more detailed way. Creswell (2007) stated that qualitative research is appropriate when there is too little information about the “phenomenon of study” (p. 16). Similarly, Bogdan and Biklen (2007) stated that it is possible to deeply explore a phenomenon in qualitative studies. Secondly, in order to deeply understand the use of virtual worlds in programming education, there is a need to learn more from the perspective of children through exploration in a real context and setting (Creswell, 2007). Lastly, the characteristics of qualitative studies correspond well to the approach followed in this study; for example, interview and observation were the main data collection instruments employed.

After matching the methodological approach to the research problems, the type of qualitative research was then decided from the many forms of qualitative research. Case study research, one of the qualitative research designs, was adopted for the current study. Before rationalizing this selection, it could be helpful to consider some of the characteristics of case study in more detail. Yin (1994) defined case study as “an empirical inquiry that investigates a contemporary phenomenon within its real life context especially when the boundaries between the phenomenon and context are not clearly evident” (p. 13). In addition, Stake (1995) defined case study as the use of a single case or multiple cases with boundaries in which a researcher provides an in-depth understanding of the cases or a comparison of several cases by gathering multiple sources of data. Baxter and Jack (2008) argued that Stake (1995) and Yin (1994) are two proponents of case study and that the mutual point in the two definitions is the existence of boundaries or a requirement of placing boundaries on a case. Creswell (2007) also pointed out the importance of bounded context and Merriam (1998) cautioned that if a phenomenon studied could not be intrinsically bound, then that study cannot be a legitimate case study.

Case study research could be preferred as a research design; when the form of research questions are how and why format, when the researcher's control is limited or absent, or when the focus is on a contemporary phenomenon (Yin, 2003). Case study provides much detailed information about the phenomenon being studied in its context and offers a more complete picture of what happened and why (Neale, Thapa, & Boyce, 2006). Merriam (2009) added that case study is useful when studying educational innovations, evaluating programs, and informing on policy.

Due to several legitimate reasons, case study is considered as appropriate for the research design of the current study. Firstly, the main points of using virtual worlds in programming education for children are not yet clearly known. There is therefore a clear need for more in-depth understanding of the issues in order to reveal the main points with regards to VW usage in programming education. Such a phenomenon under investigation can be studied in significant depth through the application of case studies (Merriam, 1998). Secondly, the focus of the current study is on a contemporary phenomenon, virtual worlds, which is one of today's educational innovations (Kafai, 2010), and the use of VWs in the provision of programming education for children needs to be explored in a broad manner. Thirdly, the use of virtual worlds needs to be investigated in real context without manipulation. Lastly, the cases were bounded as curricular, extra-curricular and after-school programs.

Case studies vary according to their type, intent of analysis and size of bounded case, such as an individual or multiple case (Creswell, 2007; Merriam, 2009; Stake, 2005). Researchers need to decide on the type of case study most convenient and pertinent for the particular study in hand (Creswell, 2007; Baxter & Jack, 2008). Multiple case is one of the case study designs. It requires collecting and analyzing more than one single case (Creswell, 2007; Yin, 2003). There are different names used in the literature referring to multiple case studies such as collective case studies, multicase or multisite studies (Merriam, 1998). In this current study, multiple case is the preferred term used. The number of studies adopting this design has increased in recent years, especially in the study of school-based educational innovations (Yin, 2003). In the multiple case study, researchers aim to show different perspectives on the phenomenon by analyzing multiple cases (Creswell, 2007). It enables researchers to analyze and explore differences and similarities both within and between cases (Baxter & Jack, 2008). Yin (2003) also stated that multiple case study uses the “logic of replication” and it either “(a) predicts similar results (a literal replication) or (b) predicts contrasting results but for predictable reasons (a theoretical replication)” (p. 47).

Multiple case study design has its own advantages and disadvantages (Yin, 2003). It could simply take up too much time and effort to conduct a multiple case study when the number of cases are considerable (Baxter & Jack, 2008). On the other hand, it increases the precision, validity and stability of findings and enhances generalizability of findings when compared to single cases since multiple cases increase the degree of variation across the cases (Merriam, 1998). Yin (2003) stated that the results of multiple case design are also more compelling and robust and added that if resources are available, multiple case design is often a good choice since it increases the overall quality of the research. Keeping in mind these advantages and disadvantages, multiple case design was chosen to be conducted in the current study, with three cases from different educational programs selected in order to answer the research questions more accurately. *Figure 3.1* illustrates the design of this particular multiple case study.

With increasing worldwide interest, *programming education* has become part of the curricular activities in some schools as well as an extra-curricular activity organized both within schools and after-school programs that operate outside of schools. Students can therefore learn programming in three different educational programs: curricular, extra-curricular and after-school programs. With the help of this current study, it is aimed to better understand the use of virtual worlds in programming education across these three different types of educational program. Moreover, the similarities and differences between and within the cases will be revealed.

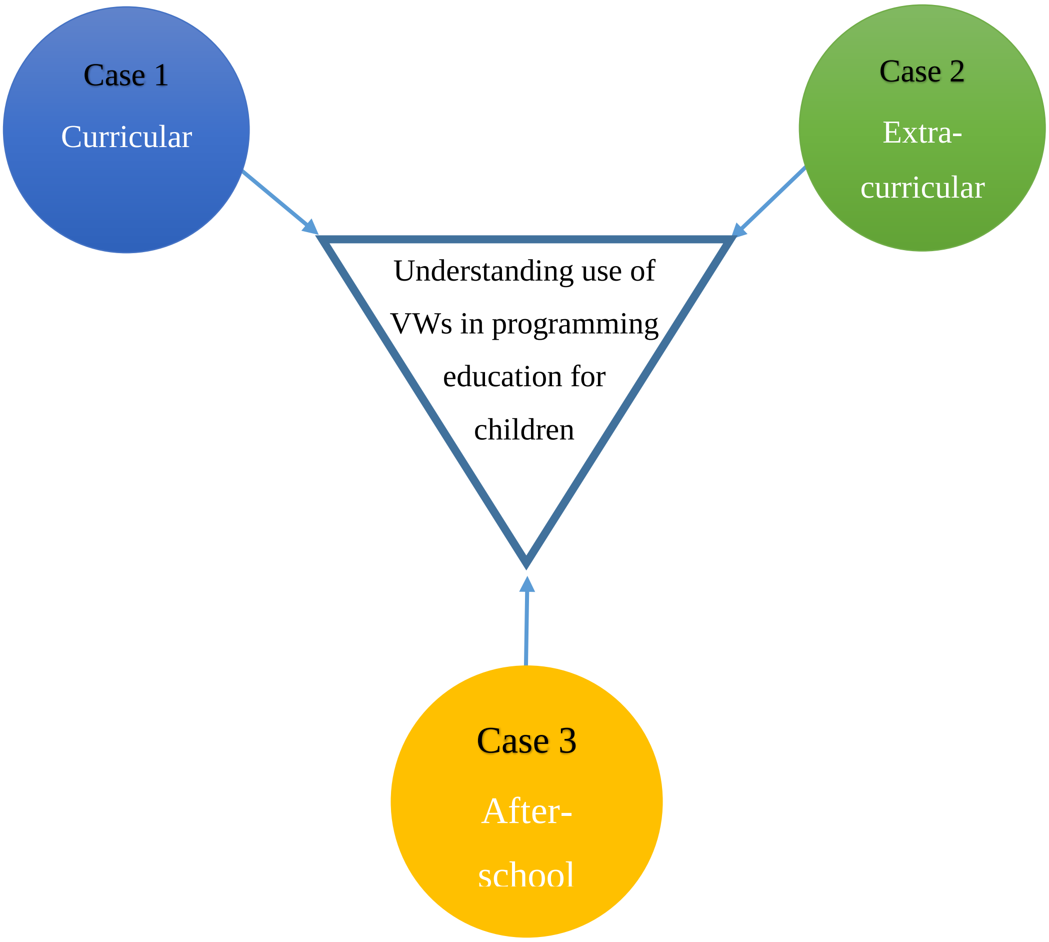


Figure 3.1 – Multiple case study design (adopted from Creswell, 2007)

Figure 3.2 represents the overall research design of the study. Detailed information about each part of the figure are discussed in the following chapters.

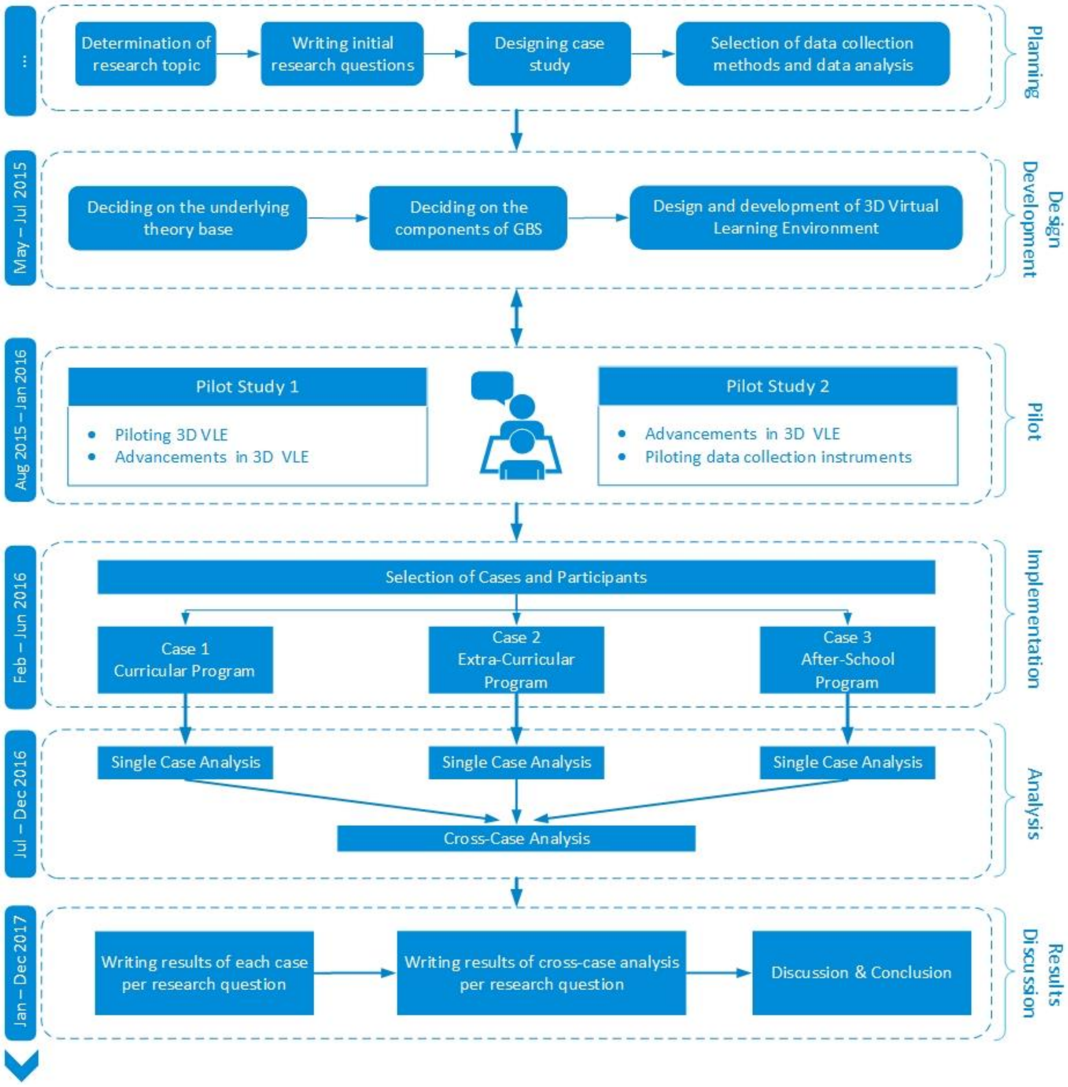


Figure 3.2 – Overall research design and implementation of the study

3.3 Pilot Study

Pilot study is a small-scale trial in which proposed procedures are conducted and tested in order to detect problems, if any, and to refine the processes before the main study starts (Fraenkel et al., 2012). Conducting a pilot study before the main study enables a researcher to be aware of potential problems and thereby protects the researcher from entering the "field" in a "blind" mode (Sampson, 2004, p. 398). Conducting a pilot case also enables a researcher to take the necessary precautions to mitigate risks from potential problems that may occur throughout the research study. Pilot study is more than testing the data collection methods of a study (Merriam, 2009). There are particular advantages of conducting a pilot study such as it allows for the refinement of research instruments, interview questions, data collection plans and procedures, and helps to clarify the appropriateness of the research design, and to assess the degree of observer bias (Merriam, 2009; Sampson, 2004; Yin, 2003).

Pilot cases should be selected based on their convenience, access and geographical proximity to the researcher (Creswell, 2007; Yin, 2003). The researcher of this current study decided to conduct two pilot cases before the main research. In the first pilot, technical and infrastructural issues such as server and client-side capability, and the appropriateness of the programs to be used by the participants were tested in order to provide information about the logistics of the study's application and to "observe the technology in action" Yin (2003, p. 110).

The first pilot case was applied within a voluntary organization called CoderDojo; an international agency that aims to teach programming to youth. The pilot study lasted for a period of about six weeks during August and September, 2015. Each Sunday, between six to eight students voluntarily participated in a 90 minute activity. Among the participants of this first pilot, six of them regularly participated in weekly activities. The participant students' grade level was between Grade 5 and Grade 7, their age ranged from 10 to 13 years, they were split 50/50 in terms of their gender, and all were unfamiliar with programming prior to the application of the pilot.

For this first pilot case, a dedicated OpenSim server was established in the CEIT department of the researcher's university. Students brought their own laptops to the activity and the researcher helped them with their activities. The virtual environment was designed in such a way to allow all of the students to study in the same region. They were all connected to the same server through their personal computers using the organization's infrastructure.

Based on the results of the first pilot case, the researcher took some precautionary measures in reaction to problems that occurred during the first pilot case. The researcher decided to use local servers that did not require an Internet connection rather than a dedicated server. One local server per two or four clients were planned to be established for the main study. In order to do so, a special server package called "Sim on a Stick" that includes all necessary programs to run an OpenSim server on a local computer, was planned to be used for the following reasons:

- A dedicated OpenSim server was not found to be suited to the activities in plan as the dedicated OpenSim server was inadequate at times. Code created by the students were not running as fast as they should, even though only six to eight students were worked on the same server. Some lag was experienced when running the students' code;
- It was not possible to connect to the dedicated server due to the organization's infrastructure; most schools facilities were not in a good condition and their Internet connection was known to be generally slow. More significantly, there was known Internet censorship imposed at some schools;
- On occasions, problems could occur at a server side that would be difficult to overcome with an offsite server. Problems occurring on a single dedicated server would affect all students in the class;
- When all students were in the same region in the virtual world on a dedicated server, they did not concentrate on specific tasks since they were performing off-task activities such as pushing their avatars in the same building during the pilot study.

The use of personal computers to connect to a virtual environment caused some problems as they had diverse technical capabilities and were generally inappropriate to run the viewer. The personal computers either had older version software installed or their hardware performance or capability was inadequate. Moreover, maintaining the personal computers could be problematic since they were diverse, and could not necessarily be freely controlled as they were personal items belonging to the students. Since attendance to the CoderDojo organization was voluntary, some participants might not attend all of the classes, and for that reason, a voluntary organization was not selected for the main case study.

In addition, the virtual environment, viewer and S4OS were tested. Some adjustments to the virtual environment were applied as a result. Some objects (i.e. caterpillar) most captured the students' attention and were either removed or left aside from the environment. Students also liked to travel within the environment by flying. However, whilst flying they tended to miss the task locations, so flying was limited in the second region. Signboards on the virtual buildings and billboards were enhanced and their numbers increased.

Based on the results of the first pilot case, the second was designed accordingly and conducted in a private school located in Ankara during the 2015-2016 academic year's fall semester. Implementation at the school took place in the "Game Programming" club, with 1.5 lesson hours per week over an eight week period. The school had a well-established computer laboratory consisting of 24 computers. Before the study, all the necessary software programs were installed and tested on the computers, which were set up with one local server per two clients. One female and 19 male students from the 5th grade aged 10-11 years participated in the second pilot case.

Based on lessons learned from the second pilot case; the following enhancements and precautionary measures were applied for the main study's application:

- Since the number of servers exceeded one and it took time to prepare the connection of the servers, all servers were checked to ensure they were ready to be connected before the lesson began. Backup servers were prepared to cover for any unforeseen server-based issues;
- Data collection instruments, interview questions and questionnaires were all tested. Ambiguity and missing points were detected and changes applied according to the participants' feedback;
- The whole study and activities were practiced and their appropriateness was analyzed. At the end of the second pilot study, it was realized that some tasks from the first and second region were deemed to be too difficult for the students' level and that they took too much time to complete. Based on the ideas of a teacher, such tasks were eliminated from the study. The required time to complete individual activities and whole topics were identified;
- Finally, the researcher was able to take note of important implementation issues. During the pilot study and interviews held with students, the researcher realized that the students were able to use the virtual environment and S4OS, and that they had no major problems while transferring their code to the virtual

world. Additionally, it was possible for the researcher to test whether or not the study could be conducted with a high number of participants.

3.4 3D Virtual World

A 3D virtual learning environment was designed and developed based on a platform called OpenSim. OpenSim is a free, open-source, multi-user 3D application server on which 3D virtual worlds can be created. It allows developers to customize and develop their virtual worlds based on their specific needs. Among the other virtual world server applications, OpenSim was chosen because of its characteristics. Firstly, since the participants of the study are children, the designed virtual world must be appropriate for children in that they must feel safe and secure. OpenSim server was customized by the researcher and with private access restriction, only the study's participants, as permitted users, could gain access to the virtual world. The content of the server was also assured to be appropriate for children, having been specifically designed by the researcher based on the needs of children. Secondly, it was free and customizable according to the needs of the study (Rico et al., 2011). For example, all participants should have the right to build objects and to program those objects. Building objects and getting virtual space is limited in commercial virtual worlds. However, the researcher was able to arrange such features and add additional functionality in the OpenSim server to the desired level. Lastly, due to the nature of the current study, the servers could be reached from a local network without requiring Internet connection. This was also possible on the OpenSim server.

Sim on a Stick (SoaS), including all the necessary programs for creating a 3D virtual world, was used in the current study. SoaS is an all-in-one server package. It contains an OpenSim server, an Apache web server, and a MySQL database server in order to create a portable and standalone server. Thanks to this package, installation and configuration of all necessary components of running a proper 3D virtual world server could be predefined and easily distributed. Figure 3.3 shows the architecture of SoaS. A brief explanation of the SoaS components are given following the figure.

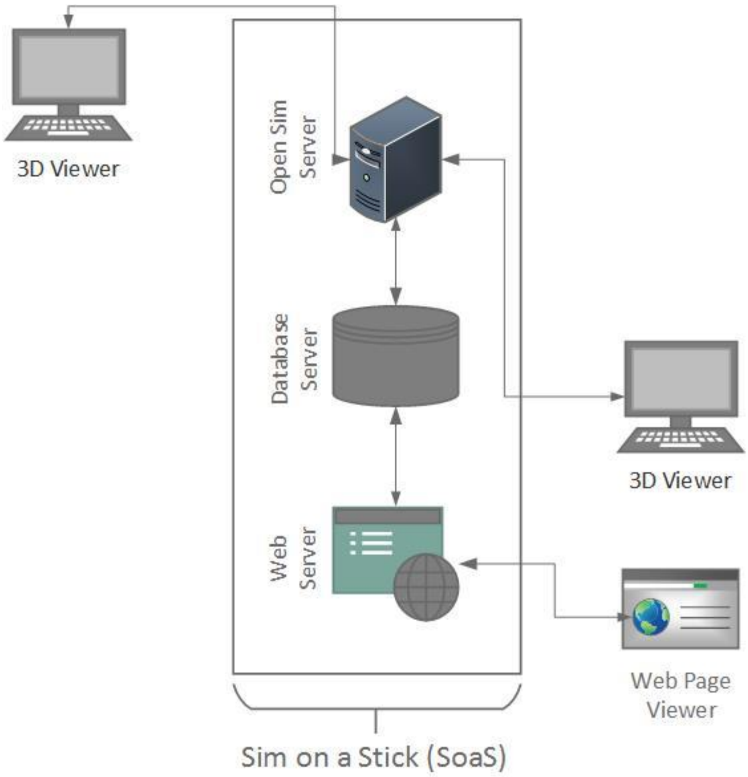


Figure 3.3 – Architecture of Sim on a Stick

Database Server: User Logon IDs, passwords and all other information are stored on this server. It is responsible for authenticating users, maintaining and providing information about users when needed. In SoaS, a MySQL database server was used.

Web Server: Users' information could be reached and managed via a web page, which could be accessed via a web address. *Figure 3.4* shows a web page layout of SDP. For example, users' login name and passwords were created via this server. In SoaS, an Apache web server was used.

OpenSim Server: This server is the main server on which virtual worlds are created. Users connect to this server with their user name and password via a 3D viewer installed on their computer.

Figure 3.4 – Web page layout of SDP

3D Viewer: This is used for connecting, navigating, and building objects in a 3D virtual world and could therefore be considered as an Internet browser (i.e. Google Chrome). There are various 3D viewers such as Imprudence, Firefox or Singularity. In this current study, the Imprudence viewer was used due to its extensive support available for LSL (Choudhury & Banerjee, 2012), stability, and Turkish language support. Most of the terms on the interface of Imprudence are in Turkish. *Figure 3.5* shows a screenshot of the Imprudence viewer. There are many collapsible panels on the Imprudence interface such as Communication, Movement, Camera Control, Inventory, and Avatar Appearance. The most used panels are explained as follows:

Inventory: This is one of the most used panels on the Imprudence viewer. All items belonging to a specific user are stored inside the inventory, such as items worn, objects, and script files. When an object is taken from the environment, it is stored in the inventory. All of the entities belonging to users are organized in folders within the inventory. For example, when a user wants to add an object such as a desk to the 3D environment, they must first find it in the inventory among the folders, and then drag and drop it into the 3D environment.

Building and Editing: The creation of primitives are performed via this panel. Students use the panel during the creating of 3D robots from scratch and editing their current objects' properties such as size, position, or color. Students can also edit, configure and add scripts to their 3D objects from this panel.

Figure 3.5 – Screenshot of Imprudence viewer

Movement Control: The movement of an avatar can be performed by using the functions on this panel, as well as using the keyboard hotkeys. An avatar's different modes of movement (i.e. run, fly) are changed from this panel.

Communication: Users can communicate with each other in different forms. Although OpenSim supports text-based or voice-based chat, students generally used text-based chat. Moreover, it is possible to communicate in two types in OpenSim; public and private chat. Any user can see/hear public chat and join in, which provides the user with a certain level of distance. Private chat can be performed between or among certain users once they have been added to their friends list.

Maps: Users can see an aerial view of the whole virtual world by using this panel. Users can also teleport their avatars to a certain location by double-clicking to a point on the map.

Avatar Appearance: Users can customize their avatar's appearance by using this panel. For example, they can change their avatar's height, weight, skin color, or hair type and color from the functionality in this panel.

3.5 Design and Development Phase of SDP (Sanal Dünyada Programlama)

Sanal Dünyada Programlama (SDP), means Programming in Virtual World, refers to integration of S4OS with VW. The design and development phase of 3D learning environment were undertaken by the researcher with the help of a PhD student at CEIT and an ICT teacher who works at a private school and has taught programming to children for four years. Before deciding on the activities for a 3D VW, during the design phase, topics to be covered to teach basics of programming were specified by inspecting academic studies and educational textbooks prepared for teaching children programming. See Appendix A for the topics and sub-topics covered by this study. The activities were designed based on an instructional theory, which is comprehensively explained in the following section. In the last part of this section, the components of SDP, Scratch for OpenSim and 3D virtual learning environment are presented.

3.5.1 Underlying Theory: Goal Based Scenario (GBS)

3D learning environment was designed based on the principles of the Goal-Based Scenario approach. Gustafsson and Branch (2002) identified instructional design models as providing guidelines by putting the instruction into small steps for effective instruction. Instructors and teachers should follow the steps of the model as it explains how to practice the instructional design process. The nature of learner characteristics, learning environment, and the ability and background of the instructional designer and educator helps to identify which instructional design model is the most appropriate to use (Şendağ & Başer, 2013). The reasons for selecting GBS for the current study are explained as follows:

- Programming education could be seen as boring for some students. However, GBS includes activities capable of increasing learners' motivation by presenting topics in a realistic context and providing learners a role within a scenario to accomplish tasks (Schank, Berman, & Macpherson, 1999). Moreover, it is essential in GBS to draw the learners' attention to the lesson (Schank & Kass, 1996);
- Learning programming is a complex matter, especially for the novices, hence learning objectives need to be defined very clearly. Kilic and Yildirim (2012) argued that GBS is appropriate for the teaching of complex learning skills and Schank and Kass (1996) contend that activities in GBS include scenarios that each present a concrete mission to teach a set of defined learning objectives;
- GBS is emphasized as "a model that students learn how-to rather than know-that" (Schank et al., 1999, p. 165). Thanks to GBS, students learn how to use the basic concepts of programming rather than learning useless or superfluous factual information. The current study aims to teach students programming through relevant tasks. The main purpose of GBS is, therefore, to enable students to make use of their knowledge and skills in a real-life context (Gülbahar, Avcı, & Ergün, 2012);
- GBS is appropriate for any domain, skills and for any student at any age (Schank, Fano, Bell, & Jona, 1994). It is also appropriate for both formal and informal learning situations (Kolodner, 1994);
- Computers play an important role in the successful implementation of GBS since they are well-suited and often necessary for the appropriate GBS application (Schank et al., 1994). Therefore, using a computer is considered essential for the effective implementation of GBS in practice (Kılıç & Yıldırım, 2012). In the current study, the use of computers in the 3D environment also increases the appropriateness of GBS;
- Students can study either individual or in small groups within a computer-simulated environment (Schank et al., 1999). In the current study, the students were encouraged to work in groups in VW.

Consequently, as can be understood from the aforementioned reasons, GBS corresponds well to the instructional design model adopted in the current study. The activities were therefore designed and prepared in accordance with GBS as one of the most promising instructional design models for instruction based on educational software (Kılıç, 2009). The seven components of GBS were adopted for the current study by examining rare examples showing how to implement GBS to an educational course (Gülbahar et al., 2012; Hsu & Moore, 2010; Kılıç, 2009), which are elaborated on in the next section.

Seven Components of GBS

GBS consists of seven components (Schank et al., 1999), which are learning goals, mission, role, cover story, scenario operation, resources, and feedback. A brief explanation and adaptation to the current study are as follows:

- *Learning Goals:* Target skills need to be defined as learning goals. The researcher worked with experts on defining the learning goals for the current study. The learning goals were determined after examining studies related to programming education for children. The main and sub-topics are shown in Appendix A;
- *Mission:* This is defined as realistic, motivational and meaningful for children to follow. In the current study, the students were told that there is a town in the virtual world called "Sorunlu Kasaba," (Problematical Town) which is known to have some problematic issues. The problems need to be solved one-by-one in order for the town's residents to become happy. The students' mission was to solve each of the problems within their pairs;
- *Cover Story*: This is the background story that is the driving force for students to achieve the mission, and therefore needs to be motivational and interesting. In this current study, a rationale for solving the town's problems was provided to the students. Students then worked with their peers in the town within the scope of this story. The cover story for "Sorunlu Kasaba" can be seen in Appendix B;
- *Role*: This is something realistic and exciting for the student within the cover story. In the current study, the role of the students was to work in the town along with their teammate as if they were builders. Each team of two students was responsible for completing a number of tasks in a separate town;

- *Scenario Operations*: These include all of the activities that students must perform in order to accomplish their mission. Both the researcher and the experts prepared tasks on the first island for the purposes of orientation, and then 12 tasks per student were prepared for the second island that helped students acquire the learning goals of the study. Table 3.1 provides information about each task that the students needed to solve in order to complete their mission and thereby attain their learning objectives;
- *Resources*: It is essential to provide resources to students while trying to achieve the goal of the mission. Educational materials related to programming were given to students via boards and video clips, as well as hardcopy task cards. In addition, the researcher and the teacher were on hand to provide resources for students in need;
- *Feedback*: This component "allows learners to properly index information as it is given" (Schank et al., 1999, p. 178). Feedback could be given in three formats in GBS: consequences of action, coaches, and domain experts. Consequences of action is seen whenever the student makes a mistake. The second type of feedback is via coaches who follow a student's progress and provide feedback to them as and when needed. The last type is using the domain expert, from which students can obtain feedback in terms of how an expert solves a problem. In this current study, feedback was generally given through coaches and domain experts. The researcher, teacher and sometimes peers were all sources of real-time feedback. Interaction with an instructor or a peer enables students to learn skills that have been defined for a scenario (Schank et al., 1999). A video clip was also prepared for each task in order to help students obtain immediate feedback that showed the steps an expert followed in order to complete the task. Students can watch them whenever they are stuck on a certain task, at any time and without any feeling of embarrassment (Schank, 2002).

Table 3.1 – *Tasks on the town and scenario operations*

Task Name	Definition of Task	Similarity	Topics
Read the story	Read the cover story and put on your helmet.	Same	-
Build a bridge on the river	The bridge over the Yeşilırmak River has been partially destroyed due to natural disaster. You need to rebuild the bridge for the people living in the town.	Same	• Algorithm and basics of programming
Build the wall	You need to build the walls of a shelter inside the garden for a newly arriving pet.	Similar	• Algorithm and basics of programming
Build a revolving door	You need to build a revolving door at the market. When the avatar collides the door, the door should revolve. The door should stop revolving when the collision of avatar ends.	Same	• Event handler
Build a staircase	You need to rebuild the fire damaged staircase. The staircase should have … steps and each step should be ... meters.	Similar	• Event handler • Loop
Build an automatic door	You need to build an automatic door for the building. When the avatar reaches within two meters proximity of the door, the door should open automatically. The door should close when the avatar's proximity to the door exceeds two meters.	Similar	• Forever loop • Conditional statements • Boolean logic
Move the turtle to its home across the river	A turtle managed to escape from jail and is sheltering in a rotating box. Take the turtle by touching the box. Code the turtle so that the turtle accompanies you to its home across the river.	Same	• Forever loop • Conditional statements • Boolean logic

Table 3.1 – *Tasks on the town and scenario operations (cont'd)*

Task Name	Definition of Task	Similarity	Topics
Build a counter for café / ice-cream shop	The owner of the shop wants to know how many customers visited the shop and asks for your help. You need to record the number of visitors and then show the number.	Similar	• Variable • Change and view the value of variables
Smart mail counter	A family want a smart mailbox for their home with the following characteristics. Each time the avatar touches the mailbox, the number of letters in the mailbox should increase. The mailbox can only hold a maximum of 20 letters. When there are fewer than 20 letters in the mailbox, return the message, "The mailbox has space." When there are equal or more than 20 letters, return the message, "The mailbox is full, please empty it."	Similar	• Variable • Conditional statements • Coordination and synchronization between objects
Move the heavy box	You need to help an elderly couples on this task. Since the couples had difficulty in moving the heavy box, you should program the box to move both sides when the related button is touched.	Similar	• Coordination and synchronization between objects
Build a letter game	You need to build a letter game for the grandchild of the house owner. A cube has a letter, from A to F, on each face. The cube should randomly rotate when touched.	Similar	• Random numbers • Loop
Revolve a funfair carrousel	You are asked to help revolve a funfair carrousel. Firstly, the carrousel should stop and it should start revolving e around when touched by an avatar, and it should stop when touched again. This should continue in this order.	Similar	• Variable • Conditional statements

3.5.2 Components of SDP

SDP was comprised of two components; 3D learning environment including a number of programming activities, and the Scratch for OpenSim (S4OS) program that is used for building code to complete activities in the 3D environment. The following sections provide further detail, firstly for S4OS and then the 3D learning environment.

Scratch for OpenSim (S4OS)

This program is used for creating code in the "Linden Scripting Language" (LSL), which is the original programming language of OpenSim. It is a modification of the original Scratch software that was modified by Rosenbaum (2008), a member of the MIT Lifelong Kindergarten team.

In S4OS, users can build code by dragging and dropping known code blocks of the original Scratch program onto the 3D object being constructed. Unlike Scratch, S4OS does not produce any output of built code itself, but it is only used for creates LSL code based on the code blocks by the user. After building the code on this program, users then need to click the "Sanal Dünya Kodunu üret" (Generate Virtual World Code) button in order to translate the pseudocode of Scratch into LSL. After the translation process, learners attach the LSL code to an object or robots offered to them within the 3D environment in order to see the output of their code. Transferring code from S4OS to the virtual world is very easy and can be achieved simply through double-clicks by the user. Pellas (2014) argued that it could therefore be used easily by learners from primary school through to university level in order to create a wide range of 3D virtual artefacts in OpenSim. In a study by Pellas (2015), S4OS was used by novice learners to add behaviors and interactivity to 3D objects, and thereby create 3D artifacts within an OpenSim environment.

The user language of the software's interface was English, but the researcher translated much of the interface into Turkish after obtaining the necessary permissions from the software's author. The S4OS interface is very similar to Scratch, except for stage and sprites. A screenshot of the S4OS programming environment is shown in *Figure 3.6*. As seen in *Figure 3.6*, the interface consists of four parts. There are eight code categories (top-left), with the relevant code blocks for each chosen category listed on the left. The area in the right-middle is used for dragging and dropping code blocks. Lastly, there is a button used for translating the code into LSL, labeled “Sanal Dünya Kodunu üret” (Generate Virtual World Code), and another button for translating the code from LSL to Scratch, labeled “kodu yapıştır” (paste code), which are both positioned at the top-right of the screen.

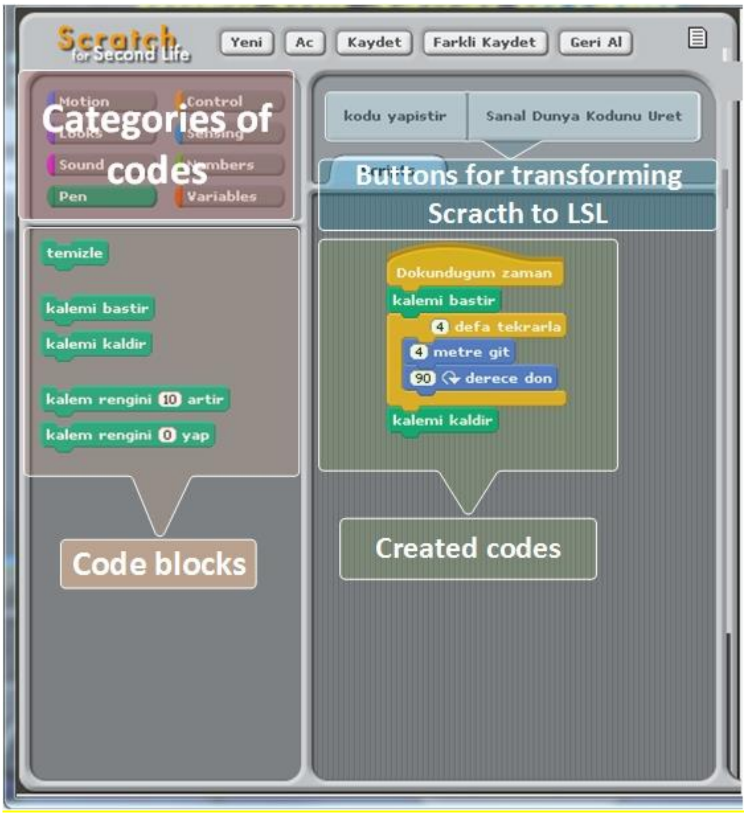


Figure 3.6 – S4OS programming environment

3D Learning Environment

The virtual learning environment was developed and built by the researcher in collaboration with experts, in line with the underlying theory, between June and August of 2015. It was composed of two regions, named islands in OpenSim. The first island is for the orientation of students to the virtual world. The aim of this island is to introduce students to the generic skills of the VW including navigation within the environment, the creation of basic 3D robots and shapes, and the attachment of code required in order to keep them progressing on their chosen routes. *Figure 3.7* shows an overview of this first island. The students completed the activities themselves on this island. This island consists of five areas:

- *Welcome Center* is the main area for welcoming users. All of the students started off here when they first logged in and they also reach to other areas via this center.
- *Avatar Center* consists of many options for avatars and outfits. It is here that students chose their avatars and select the appropriate clothing for their avatar.
- *Social Area and Cafe* includes posters and videos of famous speeches related to the importance of programming. Students can tour around this area at will.
- *Robot Training Center* consists of buildings and various routes. Students coded their 3D robots in order to keep them moving on the chosen routes for each building.
- *3D Geometric Shape Creation Center* is used for creating 3D shapes via programming, such as creating a triangle, or a square.

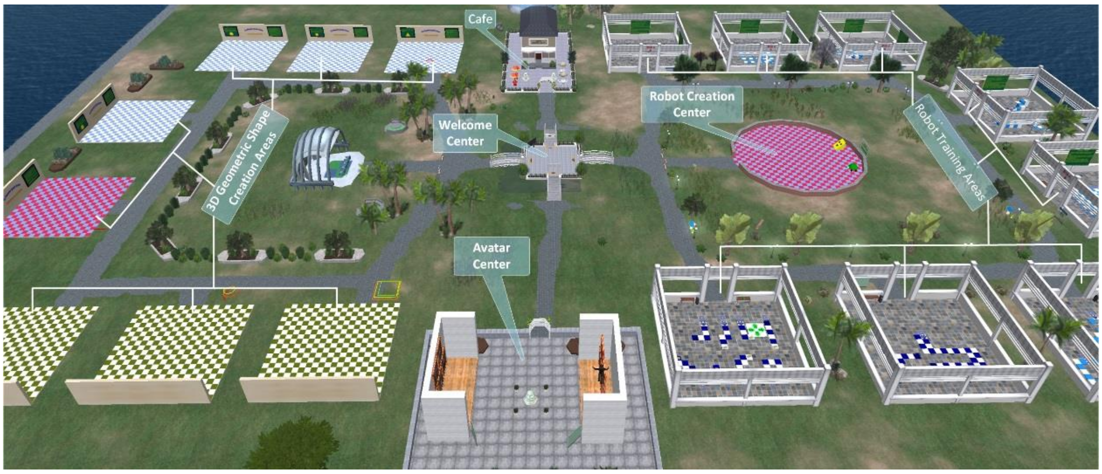


Figure 3.7 – Overview of the first island

The second or main island is called "Sorunlu Kasaba," and was designed based on the theory of GBS. Students worked together on this island in pairs. There were a total of 24 tasks for each team and therefore each team member had 12 tasks to complete on the island. Each task was numbered from 1 to 12 and colored as either red or blue. A color was assigned to each student at the start of the study and they were tasked with completing all of the tasks with their corresponding assigned color. Each task was designed for the achievement of a specific learning goal. Students were required to complete each task one by one along with their teammate. While engaging in their tasks, teammates could discuss their assigned tasks and thereby get help and support from each other. Group study was promoted and encouraged throughout the study. Information about tasks on this island are detailed in Table 3.1, whilst *Figure 3.8* shows an overview of the island. There were some numerical differences in the code of red and blue colored tasks.

Figure 3.8 – Overview of the second island

3.6 Selection of Cases and Participants

Main cases of a study need to be identified properly (Merriam, 1998; Stake, 2005; Yin, 2003). Two levels of sampling are followed in multiple case studies (Merriam, 1998), with cases selected in the first level and participants selected for each case in the second level. Purposive sampling technique was employed for the current study because it enabled the researcher to specifically select individuals intentionally and to better understand the central phenomenon (Creswell, 2012). Purposeful sampling is used when the researcher “wants to discover, understand and gain insight and therefore must select a sample from which the most can be learned” (Merriam, 1998, p. 61) and when the researcher wants to reach information-rich cases (Patton, 1990). At this point, the researcher selects the most appropriate cases which provide the most available data (Stake, 2005).

Building a rationale or criteria for purposeful sampling strategy is the next step (Creswell, 2007; Merriam, 1998; Stake, 2005; Yin, 2003). In the current study, the primary criteria was to understand the use of virtual worlds in programming education offered to children in different settings. Previous studies showed that introductory level programming education for children has been offered (Kafai & Burke, 2014) in three educational programs; curricular, extra-curricular and after-school programs. Therefore, the cases selected for the current study were based on these three educational program types. There are also other criteria related to the settings of cases that can influence the selection of cases as well as the implementation process (Miles & Huberman, 1994). Selected cases should also meet the following conditions for the effective implementation of the study:

- Laboratory infrastructure needs to be in good condition. All computers should be running properly, and it is better to have a robust local network connection among the computers (Crellin, Duke-Williams, Chandler, & Collinson, 2009; Dreher et al., 2009);
- Technical capabilities of the computers need to be sufficient to run 3D viewers since they have some high capability hardware requirements (Choudhury & Banerjee, 2012; Cooper, Carroll, Liu, Franklin, & Chelberg, 2009);
- Administrators and teachers should agree on the use of this system for their courses.

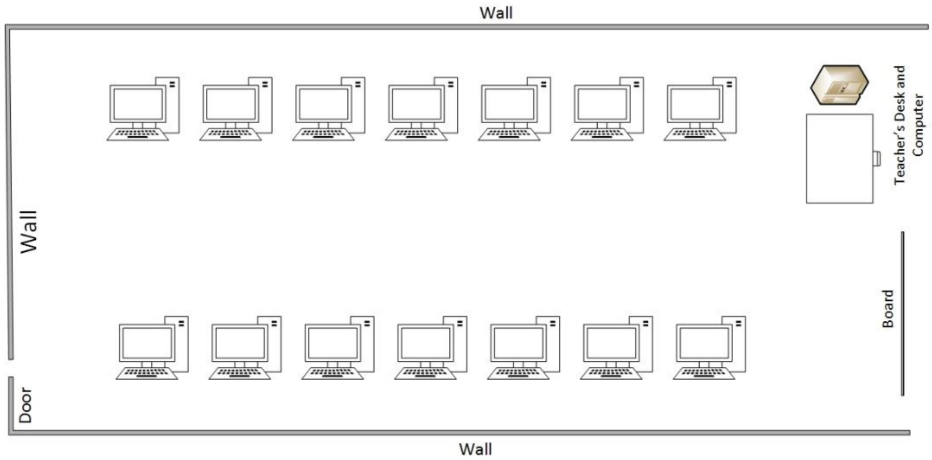
Three cases that met these criteria were then defined. Each case corresponds to three different educational programs as well as settings. The participants of the cases were generally 5th grade students, aged 10 to 12 years. Detailed information about the participants is provided in the next chapter, whilst information about each case is detailed in the following section.

3.6.1 Case-1: Curricular Program

The first case was a curricular program in which programming education was adopted from the current curriculum of an existing course in a school setting. This case took place in a private school located in Ankara. The management and the ICT teacher of the school granted permission to conduct this study in one of the sections of their school. There was only one 5th grade section in the school and it was selected. The selected section, named 5-A, consisted of 12 students. A consent form was given to the students' parents informing them about the study. All parents agreed to allow their children to participate in this study.

Programming education was adopted in the ICT course for this case. The ICT course was compulsory for all 4th and 5th grades in the school. This course did not have a specific curriculum defined by the Ministry of National Education. The aim of the ICT course was to enable students to use information and communication tools in an effective and creative way, but considering the appropriate ethical issues (MoNE-BoE, 2012). It was stated in the ICT curriculum of MoE-TEB (2012) that teachers were free to decide on contemporary topics, including coding education. The course lasted for two lesson hours per week at the 5th grade level.

Lectures took place in the school computer laboratory with 16 computers. All necessary programs were installed and tested on the computers in advance. The implementation phase for this case lasted for a period of eight weeks during the spring semester of the 2015-2016 academic year. Table 3.2 presents the weekly activities performed in this case. The first part of the study (activities on the first island) was conducted in the school's laboratory, whilst the second part (activities on the second island) was conducted in the CEIT laboratory at the researcher's university. The reason to move the study from the school to the department laboratory was that the school network infrastructure proved to be inadequate which had resulted in students not being able to study in pairs. Therefore, the students and their teacher were invited to complete the second part of the study at the CEIT department's laboratory. *Figure 3.9* illustrates the layout of the computer laboratories at the school and the CEIT department.



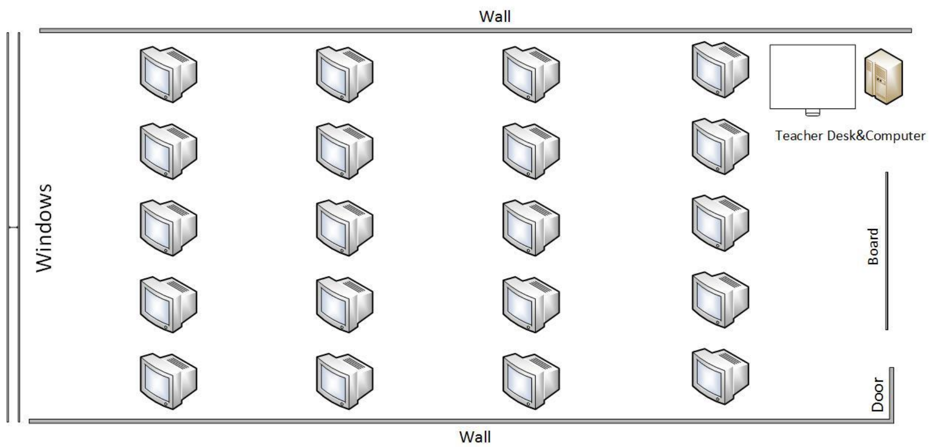


Figure 3.9 – Computer laboratory layout: school (top) , CEIT department (bottom)

The teacher for this case was relatively new to teaching, having recently graduated from the Computer Education and Instructional Technology department of a private university in Turkey. She had two years prior teaching experience and had been working at the participating school for one year. She appeared to be a hard working teacher and was trying to learn about the contemporary issues related to ICT. The teacher was also receptive to innovation and seemed quite willing to try new tools in her classes. She was teaching basics of programming to the students of this school at a club; however, she had not managed to integrate programming education into the ICT curriculum at the school. After the school management approved this study could be conducted in the school's ICT course, the teacher agreed to collaborate with the researcher in the implementing of the study in her classes. She assisted the researcher in terms of classroom management, organization of the lesson and study, as well as on technical issues and providing feedback to her students.

Table 3.2 – *Weekly activities of Case-1*

Week	Lesson Hour (40 min)	Activity
1	2	Training on generic skills, customizing avatar, , and introducing S4OS
2	2	Robots trained to follow routes
3	2	3D object construction
4	2	3D object construction
5	3	Meeting on the second island, starting to complete tasks
6	3	Task completion
7	–	Interviews conducted
8	–	Interviews conducted

3.6.2 Case-2: Extra-Curricular Program

The second case was an extra-curricular program held in a private school located in Ankara. Extra-curricular activities are usually voluntary, unlike the normal curricular activities. They can be in diverse contexts and have some rules in terms of participation schedule and meeting hours (Fredricks, 2017). In terms of programming education, extra-curricular activities present a good opportunity for students to be introduced to programming due to the lack of computing courses in the set curriculum (Wyffels, Martens, & Lemmens, 2014). This program was applied in a club that had been established in the private school to teach programming. The name of the club was "Game Programming" and its purpose was to enable students to realize that they could create games similar to daily life and thereby to increase students' awareness regarding the use of computers in creating programs and games through programming. Attending the club was optional for students and only students from the 5^{th} and 6^{th} grades participated in the club voluntarily.

Club participants met for one and half lesson hours per week throughout the semester. The number of students attending the club was 22, with all but two parents allowing their children to participate in the study. Implementation in this case lasted for 10 weeks during the spring semester of the 2015-2016 academic year. Table 3.3 shows the weekly activities for Case-2. Courses were conducted in the school laboratory with 24 computers. *Figure 3.10* illustrates the layout of the school's computer laboratory.

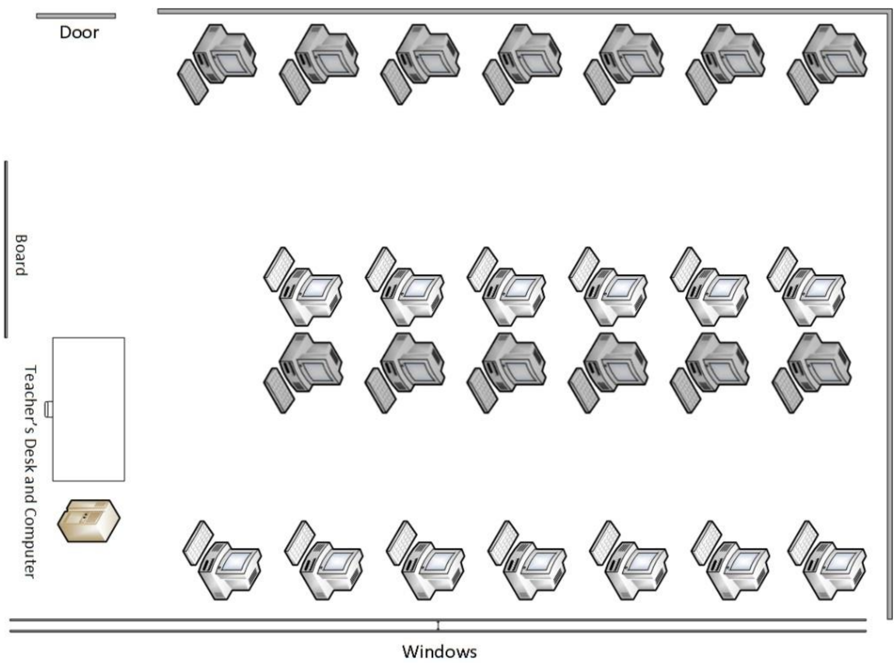


Figure 3.10 – Computer laboratory layout of Case-2

The teacher for this case was an experienced practitioner who had been teaching programming for about two years. She had obtained both a Bachelor's and Master's degree from the CEIT department of a public university, and had two years public school and 12 years private school teaching experience. She was working at a private school at the time of this study. The teacher was open to innovations and tried to integrate programming languages like Python into her courses. She has been teaching programming at an intense level way for two years in ICT courses as well as in student clubs, and was experienced in programming tools such as Scratch, Lightbot, Kodu

Game, and Small Basic. The teacher worked closely with the researcher in the computer laboratory and helped the researcher during the implementation of activities in many ways such as classroom management, resolving technical issues, providing feedback to students, and general organization of the classroom.

Table 3.3 – *Weekly activities of Case-2*

Week	Lesson Hour (60 min)	Activity
1	1.5	Training on generic skills and introducing S4OS
2	1.5	Customizing avatars
3	1.5	Robots trained to follow routes
4	1.5	Robots trained to follow routes; 3D object construction
5	1.5	3D object construction
6	1.5	Meeting on the second island; defining colors and roles
7	1.5	Task completion
8	1.5	Task completion
9	1.5	Task completion
10	1.5	Task completion
11	–	Interviews conducted
12	–	Interviews conducted

3.6.3 Case-3: After-School Program

The third case was an after-school program held in an informal learning environment occurring outside of the school and removed from the formal settings of school-based learning (Shernoff & Silva, 2017). This case was conducted on a course offered at the Continuing Education Center of a public university, which offers numerous courses across various subject areas. A computing-related course called "Üç Boyutlu Ortamda Temel Programlama Eğitimi" (Basics of Programming Education in 3D Virtual World) was offered at the Center, and students aged 11 to 13 were invited to enroll. The aim of the course was to teach basics of programming in a 3D virtual world. Announcement of the course was achieved via the Center.

Ten students enrolled to the course at the start. However, two students then left due to their heavy study and exam schedule after just the first week. Therefore, eight students completed the course and received participation certificates. Students were from different schools and ages. All parents were informed about the study and their consent taken in advance. The course lasted for a period of five weeks during May and June of 2016, with one class lasting three lesson hours per week. The weekly activities for this case are shown in Table 3.4. The classes took place in a laboratory consisting of 24 computers. *Figure 3.11* illustrates the layout of the laboratory.

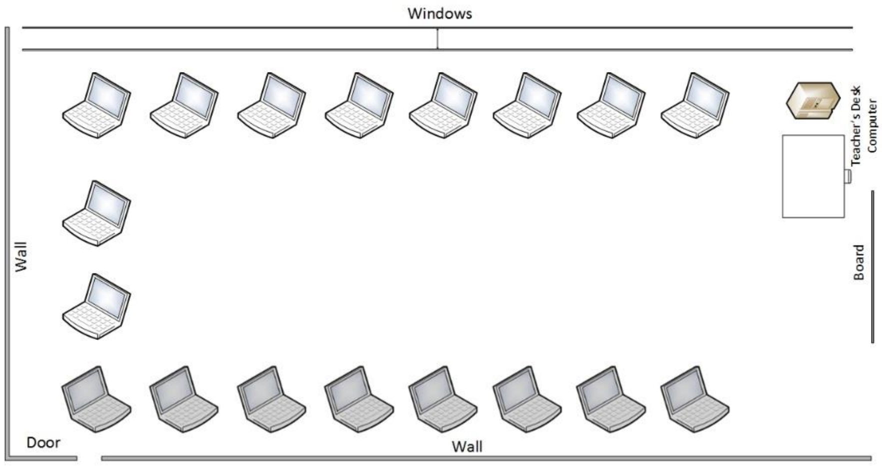


Figure 3.11 – Computer laboratory layout of Case-3

Table 3.4 – *Weekly activities of Case-3*

Week	Lesson Hour (45 min)	Activity
1	3	generic skills training and customizing avatar, and introducing S4OS
2	3	Robots trained to follow routes
3	3	3D object construction, meeting on the second island, defining colors and roles
4	3	Task completion
5	3	Task completion
6	–	Interviews conducted

The researcher along with a university instructor conducted the implementation of this third case, and worked together throughout the implementation phase in order to conduct the study in the most efficient way. The instructor had a Bachelor’s, Master’s and a Doctoral degree in the field of Instructional Technology, and has been giving graduate courses about the use of virtual world in educational settings. The researcher holds a Bachelor’s degree and is currently completing a Doctoral degree at CEIT. He has also been a Research Assistant for a period of eight years and assisted on the delivery of many courses including “Programming Languages” and “Teaching Practice.” Moreover, he has one and half year former teaching experience at a public primary school. The researcher has also taken graduate level courses about the use of VW for educational purposes.

3.7 Detailed Explanation of Implementation Phase of the Course

Although the case settings differed on points such as different weekly lesson hours and different number of weeks for the completion of implementation phases, the same guidelines were followed in all three cases. In this section, the implementation phase is explained in detail for practitioners and teachers who would like to implement a similar study with similar settings to the current study. Moreover, lesson plans of each session is presented in Appendix N.

Before commencing a session in each case, the researcher prepared all the necessary programs and ensured that they were stable and running properly. This precautionary action was taken to limit extra time being required for unanticipated computer-related issues during the lessons. The implementation phase was held in two stages, based on the activities on the two islands of the 3D environment. Figure 3.12 illustrates the overall activities on the implementation phase. The two-staged implementation is explained as follows.

3.7.1 First stage of the implementation

Activities on the first island are aimed at the orientation of students to the virtual world, in order to introduce them to the generic VW skills including navigation within the environment, creation of basic 3D robots and shapes, and the attachment of code to the objects created. Students completed the activities on this island individually.

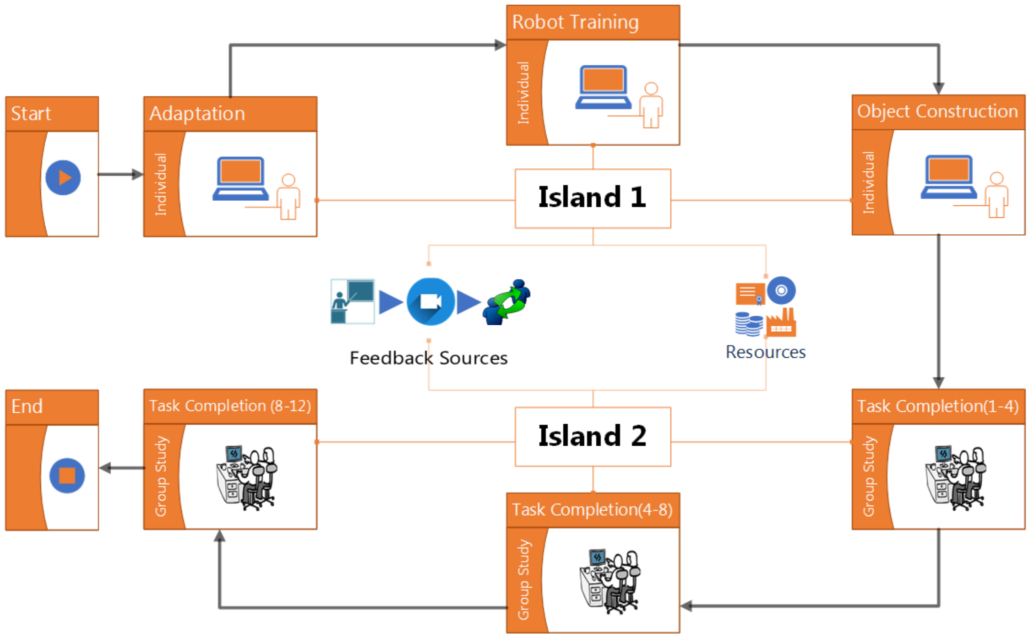


Figure 3.12 – Overview of the activities in the implementation phase

In the first sessions of this stage, students were introduced with programming, and its importance. In addition, Scratch for OpenSim and its purpose of use were introduced to the students, especially for those unfamiliar with it. A broad explanation of how to build code, its underlying structure, the categories of code and the places of code blocks were provided. Then, they were informed about their login names and passwords in order to be able to sign in to the virtual environment. Besides, students were told about the generic skills in the 3D environment such as movement of avatars, camera control, navigation, and the use of functions on the interface of the viewer. This stage was especially important for those students who did not play 3D games similar to the one used in the current study. The students were then informed about how to customize their avatar (e.g., accessories, clothing) since avatars had a default appearance when first entering the environment. Appropriate time was allocated and adequate support provided to the students during this process. At the end of this session, the students were permitted to tour areas on this island such as the café and social area.

In the next session, a 3D object, named “robot” was created with students. Students were told how to edit and change the features of the object such as its location, color, and size. At end of this session, sample code which makes a 3D object move was built with the students and they were shown how to transform the pseudocode into LSL, OpenSim’s own programming language, and how to add transformed code to the script file of the object. Students were warned that from time to time they may need to rebuild the code and redo certain steps if their code did not run as anticipated.

In the other session, the students completed activities inside the four buildings. Each building has different routes on which students trained their robots to follow the routes via programming. That is, they first created a 3D robot and then coded it to keep it moving on a certain route inside each building. Before undertaking the activities in each building, an activity sheet containing an illustration of the routes for the building was distributed to the students in order for them to be able to follow the activities (see an activity sheets in Appendix K).

In the last session on this island, the students created 3D geometric shapes such as a triangle, square, regular pentagon and hexagon. To do this, a copy of the robot was given to the students in the 3D environment, and they were asked to code the robot to create 3D shapes. Detailed information about the shapes was presented on the boards in the 3D environment and a sheet containing information about the shape was distributed before each shape (see activity sheets in Appendix L).

3.7.2 Second stage of the implementation

Students teleported to the second island after completing the activities on the first island. This second island was called “Sorunlu Kasaba,” and was designed based on the theory of GBS. Students worked in pairs together on this island. Thus, members of pairs were first assigned. Students were generally paired with the person sitting next to them according to their wishes. Coaches did not assign group members unless a problem between the members of a pair occurred. Each pair worked in a separate town in line with the cover story belonging to the island (see Appendix B for the cover story). The cover story was presented to the students in the first task as a notecard in the 3D environment. Then, the mission of the students on the island and their role was explained as they would work in the town along with their teammate as builders. Each team was responsible for solving the problems of the town by completing a number of assigned tasks. For this stage, group study with a peer was preferred due to reasons that (a) programming has a poor image as a solitary activity performed by socially “awkward” people (Brennan, 2013; Rosenbaum, 2008); (b) it should be done as a communal practice in a community (Kafai & Burke, 2014); (c) previous studies have shown that studying in a group with peers has an effect on the success, motivation, reflection, enjoyment, retention, confidence, and assistance (Esteves et al., 2011; Guzdial et al., 1996; Hanks, 2008; Hanks et al., 2011; Liebenberg et al., 2012).

There were a total of 24 tasks for each team, and therefore each team member had 12 tasks to complete. Each task was numbered from 1 to 12 and colored as either red or blue. A color was assigned to each student at the start of the study and they were tasked with completing all of the tasks with their corresponding assigned color. Most of the red and blue tasks with the same number has some differences, while four of them were completely the same. The differences between the tasks were their stories and codes. However, there were only numerical differences on their code, such as building a seven-laddered staircase as the blue task, or a ten-laddered staircase as the red task.

Students started to complete tasks; the first of which was actually explaining the cover story and putting on the red or blue helmet according to their assigned color. The other tasks had different meaningful story for students in line with the cover story. A hardcopy of the task cards containing detailed information about each task such as the instructions, warnings, and a checklist of how it was completed were prepared (see two example task cards in Appendix M). Then, all task cards was distributed to the students as hard copy in a colored binder. Students read those cards before the tasks and referred to them whenever necessary. They checked the appropriate boxes on their cards after completing each task.

Tasks were designed from simple through to advanced level. Each programming concept was aimed at being taught to the students via tasks in a cumulative way. That is, a new programming concept was taught in a subsequent task by using concepts learned in previous tasks. Therefore, the final task requires students to use most of the code blocks they had learned, and can therefore said to be the most complicated task. Students were required to complete the tasks one by one along with their teammate.

While engaging in their tasks, teammates could discuss their assigned tasks and thereby get help and support each other. Before starting a specific task, students were advised to read the information on their task card, and then the teachers explained the tasks to the students in terms of clarifying any missing points. The students then studied together with their teammate for a while. Necessary materials for completing the tasks were given to the students in 3D environment. During this time, the teacher and researcher as coaches followed the students' progress and supported them by providing feedback about their activities whenever necessary. They also managed the classroom environment. After some time, the researcher explained how to complete the tasks by demonstration in cases where any student was unable to complete them. All students passed on to the next task together, which enabled the teachers to follow the ones completed the tasks the fastest or the ones skipping tasks. The teachers explained each programming concept and corresponding code block on S4OS for the first use, and in their next use, they gave some ideas and clues to the students. Students were first asked to study each task along with their peers.

Coaches, video clips, boards in the 3D environment and tasks cards were all resources available to the students. In the resources area of the 3D environment, code blocks on S4OS were explained on the boards. In the expert videos, how an expert could complete each task was explained. They contained information about how a task should be completed, the steps to follow, and the code to build. An expert video was prepared for each of the tasks and offered to the students as a source of instant and easy feedback.

3.8 Data Collection Methods

The purpose of the study, problem and sample selected determines the type of data and data collection methods (Merriam, 1998). In qualitative studies, data are collected from participants via different forms (Creswell, 2012; Merriam, 1998; Stake, 2006; Yin, 2003). Creswell (2012) defined different kinds of data forms as observations, interviews, documents, audiovisual materials and so on. Yin (2003) also proposed six different types of data forms; documents, archival records, interviews, direct observations, participant-observations, and physical artifacts. Similarly, Merriam (1998) listed common forms of data as interviews, observation and documents.

Intensively investigating a phenomenon requires breadth and depth of data collection, as well as multiple forms of data in qualitative case studies (Yin, 2003).

Three data collection forms were used in the current study. The two main data collection tools were semi-structured interviews and observation forms, with supportive data from a questionnaire ensuring the phenomenon was sufficiently understood in depth (Merriam, 1998). Additionally, screenshots of virtual artifacts created within the VW environment were also used as a form of data collection. Detailed information about each data collection form are given in the following sections.

3.8.1 Interviews

In a case study, interview is one of the most important data sources since most case studies are about human affairs and behavioral events (Yin, 2003). Conducting interviews enables the researcher to elicit participant's thoughts and insights about the phenomenon being studied (Patton, 1990). Through participant interviews, a researcher can reveal the experiences, attitudes, ideas, intention, and perceptions of participants (Yıldırım & Şimşek, 2013). It is possible to reach important insights of participants via interviews (Yin, 2009) since many such issues cannot simply be observed (Merriam, 1998). The semi-structured interview is the most common interview type and includes specific open-ended questions followed by probing questions (Merriam, 2009). In the current study, semi-structured interviews were conducted with the participant students as well as the teachers. Two separate semi-structured interview protocols were developed after examining the related literature. One was applied to the students (see Appendix G) and the other was applied to the teachers (see Appendix H).

In order to finalize the interview protocols, a three-step method was employed. Firstly, the interview protocols were applied as part of the second pilot study. After the pilot study, some questions were consequently revised in terms of their comprehensivity and some questions were also added to the protocols. For example, questions related to demographic information of participants were added to the two interview protocols. In a second step, the interview protocols were examined by five experts in terms of language, clarity, potential for misunderstandings or ambiguities. The experts consisted of one faculty member and three PhD students at CEIT, in addition to one teacher of Turkish language. Based on the experts' feedback, some questions were enhanced and refined with respect to their clarity and grammar. In the final step, the "think aloud" method was conducted with a student who was not a participant of the main study, but had a similar background to the participants. The final version interview protocols were then formed following completion of the review processes mentioned.

The students' interview protocol (see Appendix G) was comprised of 12 main questions and several sub-questions. While the first two questions related to the students demographic information and their programming experience, the other 10 questions were concerned with one of the sub-research questions of the current study. On the other hand, the teachers' interview protocol (see Appendix H) consisted of seven main questions and several sub-questions. The first question related to the teachers demographic information and their experience in teaching programming. The other questions and their sub-questions were aimed at gathering the teachers' ideas in line with the sub-research questions of the study.

3.8.2 Observations

Observation enables researchers to directly understand the behavior of participants in their actual settings (Creswell, 2007; Yıldırım & Şimşek, 2013). As a method, observation was used as complementary to the interviews in order that any differences between what interviewees said in their interviews and how they actually behaved in the real setting could be identified (Yıldırım & Şimşek, 2013). Yin (2009) suggested that interviews need to be corroborated with information from other sources such as through observation. Observation can be a helpful and invaluable source of data, especially when a new technology is being studied within a case study since it reveals potential problems and enables the researcher to understand the actual use of the new technology being studied.

The role of the researcher could change in the observation process depending on their level of comfort and rapport with the participants. Additionally, it depends on how best to reach and collect data (Creswell, 2012). The researcher can be either a direct observer outside of the study or act as a participant-observer from within the study (Yin, 2009), or play a role somewhere between these continuum (Merriam, 1998).

As to the other main data collection form, observation, the researcher observed the students in their real and natural context and took notes about their behaviors and events. The researcher took the role of participant observer in the study. In participant observation, the researcher is not passive, but is also a member of the group inside the setting (Creswell, 2007; Fraenkel et al., 2012; Yin, 2009). In order not to miss anything, observations need to be either noted or video/audio recorded (Merriam, 2009). The management of the schools in the cases did not permit the recording of the classroom environment, so instead an observation form (see Appendix I) was utilized and field notes were regularly taken.

3.8.3 Questionnaires

Questionnaire is another way to learn information from a study's participants (Creswell, 2012). Merriam (1998) placed questionnaires into documents as a data collection form and defined these kinds of forms as "researcher generated documents." She defined questionnaires as "documents prepared by the researcher … [in order] to learn more about the situation, person, or event being investigated" (Merriam, 1998, p. 119). Quantitative data collected by way of questionnaires can be used to support qualitative data collected through interviews (Merriam, 2009). Contrary to common belief, quantitative data can also be collected and integrated within case studies (Merriam, 2009; Woodside, 2010; Yin, 1981) in order to increase the credibility of the findings by employing multiple data sources. Therefore, questionnaires were used as a data collection form in the current study.

Two questionnaires were employed in the study; the first being a demographic questionnaire, which was developed by the researcher based on the purpose of study. The second questionnaire consisted of three scales adopted from previous studies (Davis, 1989; Chou & Liu, 2005). The adopted scales were written in English, and were therefore translated into Turkish by the researcher and an expert with prior translation experience. After the translation, draft versions of the questionnaires were sent to a Turkish teacher to check for clarity, grammatical errors, and for the appropriateness of the language used considering the age group of the current study's participants. Final versions of the questionnaires (see Appendix J) were achieved after revising the draft based on the feedback received. Details of the questionnaires and scales are provided in the following sections.

Demographic Questionnaire: This included 12 questions related to the participants' demographic information such as their age, gender, grade level, and the frequency of their Internet usage, playing games and programming experience. Some of the questions were adopted from the study of Bakar-Çörez (2011).

Perceived Ease of Use (PEU) Scale: This scale was adopted from Davis (1989). It consisted of four Likert-type items ranging from (1) *completely disagree* to (5) *completely agree*. The scale was used for obtaining participants' perceived ease of use related to programs used in the current study. Scores obtained were presented descriptively, with no statistical analysis applied.

Perceived Usefulness (PU) Scale: This scale was adopted from Davis (1989). It consisted of six Likert-type items ranging from (1) *completely disagree* to (5) *completely agree*. The perceived usefulness of using a virtual world in programming education was measured via the application of this scale. Scores obtained were presented descriptively, with no statistical analysis applied.

Satisfaction Scale: This scale was adopted from Chou and Liu (2005). It consisted of four Likert-type items ranging from (1) *completely disagree* to (5) *completely disagree*. This scale is used for measuring the satisfaction level of students about programming education in a virtual world. Scores were presented descriptively, with no statistical analysis applied.

3.8.4 Other Data Collection Form

Artefacts created by the students were used as a complementary data source to the three main data collection forms. Screenshots of some of the students' artifacts were taken at the end of activities. For example, colored stairs of student artefacts were collected and stored on a portable data drive.

3.9 Data Collection Procedure

The necessary permissions were granted before starting to collect data from the participants. Firstly, the research proposal and data collection instruments were submitted to the Institutional Review Board (IRB) of the university. The METU Ethics Committee examined the documents and sanctioned the study (see Appendix C). Then, the researcher applied to the Ministry of National Education (MoNE) and gained their permission to conduct the study in the schools (see Appendix D). Thirdly, the teachers and school management were informed about the details of the study, the data collection forms to be applied and the 3D virtual world. After they also gave their approval to conduct the study in their schools, the students and parents were then informed about the study as a final step. Separate consent forms were signed by both students (see Appendix F) and parents (see Appendix E). Permission was taken from all the students and their parents, with the exception of the parents of two students in Case-2. Students whose parents did not permit their children to participate in the study were excluded from the study, although they still participated in the sessions. Similar data collection procedures were applied in each case, and the procedures are elaborated on in the following parts of this section.

Data were collected throughout the study, namely during and after the implementation. In addition, during the sessions, the students were observed. Interviews and questionnaires were employed at the end of the implementation. Data collection procedures for the three cases are summarized in Table 3.5. As can be seen from Table 3.5, although the total number of weeks for the completion of implementation phases differ for each cases due to the different weekly lesson hours, the total lesson hours for completing the implementation were mostly the same. Before commencing a session, the researcher prepared all the necessary programs and ensured that they were stable and running properly. This precautionary action was taken to ensure no extra time was required for unanticipated computer-related issues during the lessons. In addition, a hardcopy of the task cards explaining each of the activities for the current session were distributed to the students. The students were observed during the sessions. In this study, the researcher and the teachers for each case completed the observation forms with as much detail as possible for each student. This approach was applied since multiple observers increase the reliability of evidence obtained via observations (Yin, 2003). Additionally, the researcher collected and saved screenshots of some students' artifacts at the end of the activities.

Table 3.5 – *Data collection procedures*

Cases	Participants	Implementation	Forms of Data Collection
Case-1 Curricular Program	6 female, 6 males	Spring semester, 2015-2016, 4 weeks X 2 lesson hours plus 2 weeks X 3 lesson hours	• Student questionnaires at the end of implementation • Interviews with students and teacher at the end of implementation • Observation during sessions
Case-2 Extra-curricular Program	2 female, 18 males	Spring semester, 2015-2016, 10 weeks X 1.5 lesson hours	• Student questionnaires at the end of implementation • Interviews with students and teacher at the end of implementation • Observation during sessions
Case-3 After-school Program	2 female, 6 males	May–June, 2016, 5 weeks X 3 lesson hours	• Student questionnaires at the end of implementation • Interviews with students at the end of implementation • Researcher field notes for all sessions • Observation during sessions

Interviews were the primary data source of the current study. They were conducted as soon as the implementation was completed in all three cases; with all being completed within a period of two weeks. All of the interviews were conducted by the researcher to ensure consistency of the data collection procedure. Both the students and their parents were informed about the interview and their permission obtained in order to conduct and record the interviews. Then, the semi-structured interviews were conducted face-to-face and audio-recorded based on the developed interview protocol. The number of interviewees conducted in the cases were seven for Case-1, then 12 for

Case-2, and finally six for Case-3. The length of the interviews varied between 15 and 20 minutes. Interviews with the teachers of both Case-1 and Case-2 were conducted after the implementation phase completed and were audio-recorded after obtaining their permission. The length of the teacher interviews was approximately 30 minutes each. There was no responsible teacher to interview for Case-3; instead, the researcher's opinions were included for Case-3 based on the field notes taken by the researcher. Demographic and perception questionnaires were applied together to the students at the end of the implementation phase of each case.

3.10 Data Analysis

Qualitative data analysis is the process of transforming data into meaningful explanations (Creswell, 2012; Merriam, 1998). Since each qualitative study is unique and the analyst is the fundamental actor, the "procedures and processes" followed in qualitative analysis need as much elaboration as possible (Patton, 2002, p. 434). The analyst decides on the data analysis procedures and methods based on the data collected (Yıldırım & Şimşek, 2013; Yin, 2003). Although scholars have provided many different descriptions and used many different terms for the processes of data analysis in their published studies, the general process consists of three steps (Creswell, 2007). The first step includes the preparation and organization of data to be analyzed. In the second step, the data is reduced and condensed into themes and sub-themes by means of a special process called "coding." In the final step, the themes and sub-themes that emerged are presented through figures, tables and narrative text. Creswell (2012) extended these processes into six steps, which were followed during the analysis phase of this current study.

Firstly, data collected through the forms were prepared for analysis. All the interviews records and completed observation forms were transcribed verbatim. Transcriptions were read one more time by listening to the audio records and examining the observation forms in order to ensure that they were a correct match. The analyst needs to have an understanding of what data has been collected (Patton, 2002). Then, all the information must be organized in a way so that it is easily accessible and "retrievable" (Merriam, 1998, p. 194). In the current study, a Computer Assisted Qualitative Data Analysis Software (CAQDAS) was utilized to analyze the qualitative data. Yin (2009) calls such organized materials as *case study database*, whereas Patton (2002) refers to the comprehensive resource package which brings together and organizes enormous data as the *case record*. Detailed information about CAQDAS are addressed in the following part.

As the next step, the coding processes was applied. Miles, Huberman, and Saldana, (2014) described code as labels assigned to a segment of information. Those labels are then used to categorize similar segments of information. At this point, the researcher needs to use an effective coding strategy that is appropriate to the data (Yıldırım & Şimşek, 2013) and to develop meaningful and manageable coding schema (Patton, 2002). Although there are different strategies or techniques in the coding process, the mutual point is the importance of themes and the need to describe them in an organized way (Yıldırım & Şimşek, 2013). In order to achieve these, data were read several times by the researcher and then in doing so, the code emerged and evolved throughout the analysis. After the data were reduced into a code list, the themes and sub-themes were created based on the similarity of code aggregating in parallel with the research question of the study. Analysis of each single case (within-case analysis) were finalized by the steps previously mentioned. For detecting the reliability of the study, intercoder agreement strategy was applied, which is described in more detail in the Intercoder Agreement section of this chapter.

Cross-case analysis is another type of analysis that starts right after the completion of within-case analysis (Merriam, 1998). While within-case analysis enables the researcher to understand and explain each case separately (Miles et al., 2014), cross-cross analysis enables the researcher to spot similarities and differences across the cases (Stake, 2005). After the analysis of each single case was completed, cross-case analysis was applied in order to compare and contrast the emerging themes and sub-themes across the cases. Yin (2009a) suggested using a table in order to present data from each case in a separate column in order to best exhibit the researcher's understanding of what differences and similarities were seen among the cases. Therefore, in the next chapter the findings revealed at the end of each within-case analysis are reported for each sub-research question, and then the cross-case analysis findings present the overall findings in tabular format, consecutively. Lastly, issues related to the credibility of the findings are addressed in the final section of this chapter to follow the last step of Creswell (2012).

In the qualitative data analysis phase, a Computer Assisted Qualitative Data Analysis Software (Nvivo 11 Pro) was used in order to facilitate the analysis process. Using the software was helpful in terms of storing data, coding, retrieving code segments (Patton, 2002) and organizing massive amounts of data considering the difficulty in management of the analysis process (Merriam, 2009). Qualitative software provides an organized file system that enables researchers to examine data closely and easily and to visualize the relationships between code and themes (Merriam, 2009). In addition to the qualitative software, SPSS version 24 was used to deal with the descriptive statistics of scores obtained via the completed questionnaires.

3.11 Researcher's Role and Bias

The role of the researcher is crucial in research as the researcher is the central instrument in qualitative studies (Patton, 2002; Yıldırım & Şimşek, 2013). Information about the researcher including personal and professional data as well as biases should be provided in the study (Creswell, 2012; Merriam, 2009). Therefore, the role of the researcher in the current study has been addressed in this section.

Firstly, the researcher was the facilitator as well as the participant-observer. He was responsible for conducting the study himself within the classroom. In Case-1 and Cade 2, there was also a teacher who helped the researcher in managing the classroom and helping the students. In Case-3 there were no school teacher present, however a university instructor was also in the classroom in all sessions and helped the researcher in Case-3 in a similar way to the teachers in the other two cases.

Secondly, the researcher was a former ICT teacher in a public primary school for one and a half years; hence he was quite familiar with the classroom environment and therefore comfortable dealing with the students. Moreover, he has worked as a teaching and research assistant at the Computer Education and Instructional Technology (CEIT) department of the university and has a Bachelor's degree from a CEIT department at a public university in Turkey. After working for a few years as a teacher, the researcher started work at a public university as a research assistant and enrolled to an integrated PhD program at the CEIT department of a public university in Turkey.

Thirdly, the researcher had taken several graduate courses related to quantitative and qualitative studies in education, instructional technology, and instructional material development. Moreover, he took a graduate course about the use of VWs in education. The researcher also has experience in studies related to use of VWs for educational purposes. In addition, the researcher had designed and developed the VW used in this current study and maintained the system.

Finally, the researcher's main aim was to shed light on the use of VWs in programming education for children. Although he kept in mind this aim throughout the study, his background and experience in this field could be seen as a source of prejudice; something the researcher was well aware of throughout the study. Multiple sources of data and different perspectives and procedures such as audit trail and debriefing were used in order to prevent instances of researcher-related bias. The researcher tried to reveal the potential use of VWs and wanted to find how they could be best utilized for educational purposes in different settings.

3.12 Trustworthiness

Multiple perspectives exist on the literature of scholars about validity and ensuring validity in qualitative studies; however, the general consensus is that "qualitative inquirers need to demonstrate that their studies are credible" (Creswell & Miller, 2000, p. 124). Different terms have been used for referring to the validity of qualitative studies as equivalents of the quantitative approaches (Creswell, 2012). In the current study, Lincoln and Guba's (1985) view of trustworthiness was preferred while referring to that issue. All researchers are expected to assess the accuracy of their findings with the application of appropriate strategies and the reporting of them (Yıldırım & Şimşek, 2013), since each researcher needs to persuade the wider academic community that their findings are indeed trustworthy (Merriam, 2009). Various strategies could be used for assessing the accuracy of findings (Creswell, 2007; Merriam, 1998; Yıldırım & Şimşek, 2013). Detailed information about the strategies employed in the current study are given in the following sections.

3.12.1 Triangulation

Triangulation is confirming and corroborating evidence by using different individuals (e.g. a principal and a student), multiple sources of data, and multiple investigators. (Creswell, 2012; Merriam, 1998). Triangulation increases the credibility of the study and findings (Yıldırım & Şimşek, 2013). In the current study, this strategy was employed via different methods. First of all, different data collection forms were used such as interviews, observation and other forms; and the different data sources complemented each other. Secondly, interviews were conducted with both the students and the teachers. Different individuals yielded multiple perspectives about the phenomenon under investigation. Finally, investigator triangulation was employed by using different investigators in the observation process and through applying intercoder agreement strategy, which will be detailed in the following part. Merriam (2009b) argued that investigator triangulation could be employed when multiple investigators collect and analyze data.

3.12.2 Prolonged Engagement

Prolonged engagement is met when the researcher(s) stayed with the research for a long period of time (Creswell & Miller, 2000) and when the researcher has had considerable interaction with the data sources (Yıldırım & Şimşek, 2013). It is useful for building trust and rapport with participants, and for locating gatekeepers that can permit access to participants and settings. By such means as these, participants may feel comforted by the researcher familiarity and subject knowledge and thereby disclose more information. The researcher participated in all sessions of each case as a facilitator throughout the implementation. The researcher actively sought out interaction with both the students and the teachers in each case for this purpose.

3.12.3 Peer Debriefing

Also known as *peer review*, the act of *peer debriefing* occurs when someone from outside of the study, but who is familiar with the research process, reviews the overall research and data collection process (Creswell & Miller, 2000). The external reviewer could well be a peer and their role is to check the accuracy of the employed research design, the data collection methods, data analysis strategies and the reporting of the findings through the lens of devil’s advocate, and then provides feedback to the researcher (Creswell, 2007; Yıldırım & Şimşek, 2013). This strategy was employed by the researcher via consulting a peer of the researcher who was also a PhD student and familiar with the research process followed. The researcher consulted the peer and sought feedback with regards to the process of study on a number of occasions. Moreover, Merriam (2009) argued that each graduate student has the opportunity to benefit from this built-in strategy via their dissertation committee. Therefore, the researcher was also able to discuss the research process and receive valuable feedback from his advisor and dissertation committee periodically.

3.12.4 Thick Description

This procedure is met when the researcher describes the setting, the participants of the study, and the themes of the qualitative study (Creswell, 2007; Creswell & Miller, 2000) with as much detailed information as possible. Thick description enables the transferability of findings to other studies which have similar context and participant backgrounds. With such detailed information about the study, the results are more likely to make sense to readers in terms of applicability of the current study to similar studies (Yıldırım & Şimşek, 2013). Quotes from the participant interviews or observation forms are mostly used to fulfil this procedure (Merriam, 2009; Yıldırım & Şimşek, 2013). In the current study, thick description was complied with as a procedure by describing the setting of each case and providing detailed information about the participants and each of the cases. In addition, direct quotations of the participants were used while presenting the findings.

3.12.5 Reflexivity

Reflexivity, also known as *researcher's position*, is explaining the researcher's position, bias, assumption and disposition related to the study being conducted (Merriam, 2009). It is inevitable in qualitative studies that the researcher's features can affect the study being conducted (Yıldırım & Şimşek, 2013). Therefore, the researcher needs to clarify their experience, biases and assumptions at the outset of the study (Creswell, 2007). These clarifications facilitate the understanding of readers in terms of how the researcher came up with the findings of the study being presented to the reader. Those issues were mentioned in the current study in the section on the researcher's role in this chapter.

3.12.6 Intercoder Agreement

Reliability refers to the consistency of the results should the study be replicated either by the same researcher or a third party. Although this process is applicable in quantitative studies, it is "problematic in the social sciences simply because human behavior is never static" (Merriam, 2009, p. 221). In qualitative studies, dependability is taken into consideration instead of reliability and the researcher is expected to complete the analysis appropriate to the data being collected (Yıldırım & Şimşek, 2013). Therefore, it was suggested that the researcher investigate whether or not the results are consistent with the data, rather than considering whether or not the same results would be obtained if the same study was replicated (Merriam, 1998). One of the strategies employed for detecting the consistency of results compared to the collected data is to employ intercoder agreement. In this procedure, multiple coders analyze the same data in order to investigate the stability of the results between the coders (Creswell, 2007). Multiple coders try to find a level of agreement on the name of the code, themes, and sub-themes, as well as the segments of code.

In the current study, two intercoders analyzed some parts of the data in addition to the researcher. Both were PhD students from the Computer Education and Instructional Technology department and familiar with qualitative studies and analysis techniques. Firstly, the researcher explained the purpose of the current study, the research question, the data collection procedure and the overall research design of the study in a detailed way in order to inform the intercoders about the study. Then, the researcher analyzed one of the student interviews with each coder separately in order to inform them about the themes and sub-themes developed by the researcher. After that, each intercoder independently analyzed a different student interview. Lastly, the researcher's and the intercoder's coded data were compared and contrasted in order to find similarities and differences between the coded data. The researcher and the intercoders discussed any differences and tried to form a consensus about them. If successfully negotiated, any necessary updates were applied to the code and themes.

Differences among the coders could be considered as normal and natural to some extent. Various scholars have provided formulae as to how to calculate an intercoder reliability score and provided some minimum values for this score to be deemed acceptable. In the current study, the interrater agreement formula (see *Figure 3.13*) and the expected value of Miles and Huberman (1994) were used. Calculated values for intercoder agreement according to this formula were 86% and 88% for the first and second interrater, respectively. They were both above the minimum acceptable value of 80% (Miles & Huberman, 1994).

3.12.7 Audit Trail

Audit trail procedure can be conducted when the researcher broadly explains how the data were collected and the analysis conducted. It is actually explaining "how decisions were made throughout" the study in a detailed way (Merriam, 2009, p. 223). She also argued that the methodology part of dissertations should contain detailed information about how the study was conducted. Therefore, an audit trail could be said to be established within the current chapter of the study.

$$\text{Reliability} = \frac{\text{Number of agreements}}{\text{Number of agreements} + \text{Number of disagreements}}$$

Figure 3.13 – Interrater agreement formula

3.13 Limitations of the Study

As with any research, there are some limitations to the current study. Firstly, Scratch for OpenSim (S4OS) was used to build code to complete the activities in the 3D virtual world. The activities were designed for children and were limited to the basics of programming and the capabilities of S4OS. Secondly, using two separate programs might have hindered revealing the main points of VWs, although it was not stated by the study's participants. Thirdly, participants of each case were in the same physical environment during the implementation phase. Therefore, the participants interacted with each other in both the virtual and the real world. Feldon and Kafai (2008) argued that findings need to be interpreted cautiously in such situations, which are considered as unique to the study of virtual worlds. Throughout the current study, interaction in both worlds was inevitable, and this could have affected the experience of the participants in the virtual environment.

Purposive sampling was employed and some criteria were defined since the use of VW requires high capability computer hardware. The selection of cases and participants could also be mentioned as a limitation of the study; with only cases meeting set criteria selected, and the participants were typically self-selected. Therefore, the obtained data could have been affected by this issue. Due to the small sample size across the cases and the nature of case study, the findings of the current the current study cannot be generalized, but it would be possible to transfer the outcomes of the current study to a similar context to some extent. Another point is that multiple forms of data from multiple sites were gathered throughout the study; however, it was not possible to conduct an interview with a teacher for Case-3 since the researcher and faculty were responsible for that course. Instead of a teacher's interview, the researcher's field notes were utilized to corroborate the findings of Case-3. Lastly, data were collected through interviews and questionnaires based on the fact that participants responded to them fully and honestly.

CHAPTER 4

RESULTS

In this chapter, the findings of each case are presented under each sub-research question. After presenting the results of single cases, the results of cross-case analysis are presented. First, the demographic information of participants are considered, followed by the students' and teachers' perceptions about the ease of use and perceived usefulness of SDP. Thirdly, the results of affordances and challenges of using VWs in programming education for children are presented. Then, issues and strategies for avatar and group study are addressed, followed by factors affecting satisfaction and then issues and strategies for the design of SDP in different educational programs. Lastly, the results of cross case analysis are presented under the related sections of the chapter.

4.1 Demographics of Participants

This study was conducted with three different cases. Detailed descriptive information about the participants of each case is presented in this section before giving the results of each sub-research question. Participants' demographic information were collected via a questionnaire. In this section, distribution of students' gender, age, grade level are presented. Moreover, their distribution of having a home computer and Internet connection, places where they connected to the Internet, their purpose of using the Internet, weekly Internet usage hours, computer/video games playing duration are all given in the following part. Lastly, the number of students playing computer/video games and students who had programming and 3D VW experience are provided.

4.1.1 Gender, age, and grade

Case-1, curricular program, was conducted in a private school located in Ankara in the scope of an ICT lesson. Case-2, extra-curricular program, was conducted in a private school located in Ankara in the scope of a club named “Game Programming.” Case-3, after-school program, was conducted at a Continuous Education Center of a university located in Ankara, in the scope of an after-school program. The number of students and their gender distribution are presented in Table 4.1. The number of students in Case-1 were 12, consisting of six females and six males. In Case-2, there were two female and 18 male students with a total of 20. The number of students in Case-3 is eight, consisting of two females and six males.

Table 4.1 – *Gender*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
Female	6	50.00	2	10.00	2	25.00
Male	6	50.00	18	90.00	6	75.00
Total	12	100.00	20	100.00	8	100.00

Participants’ age distribution is presented in Table 4.2. Students in Case-1 were between the ages of 10 and 12 years, with a mean age of 10.83 years. Participants of Case-2 were also aged between 10 and 13 with a mean age of 11.2 years. Participants of Case-3 were aged between 10 and 12 years, with mean age of 11 years. It is noted that the mean scores of each case are very close to each other.

Table 4.2 – *Age*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
10	3	25.00	1	5.00	2	25.00
11	8	66.7	15	75.00	4	50.00
12	1	8.30	3	15.00	2	25.00
13	-	-	1	5.00	-	
Mean	10.83		11.20		11.00	

Table 4.3 shows the distribution of participants according to their school grade level. As can be seen from Table 4.3, all of the participants in Case-1 were 5^{th} graders. In Case-2 there were 17 (85%) 5^{th} graders and only three (15%) 6^{th} graders. Most of the participants (62.5%) in Case-3 were 5^{th} graders, and only one (12.5%) 6^{th} grader and two (25%) 4^{th} graders.

Table 4.3 – *Grade level*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
4^{th} grade	-	-	-	-	2	25.00
5^{th} grade	12	100.00	17	85.00	5	62.50
6^{th} grade	-	-	3	15.00	1	12.50

4.1.2 Having home computer and Internet connection at home

It was investigated whether students had computers and Internet connection at their home. As presented in Table 4.4, most of the participants (n = 11, 91.7%) have home computers as well as Internet connection in Case-1. All participants (n = 20, 100%) of Case-2 have both computers to use and Internet connection at home. In Case-3, except for one student, all of the others (n = 7, 87.5%) have home computers, but they all (n = 8, 100%) have Internet connection at home.

Table 4.4 – *Having home computer and Internet connection*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
Having Home Computer	11	91.70	20	100.00	7	87.50
Having Internet Connection	11	91.70	20	100.00	8	100.00

4.1.3 Internet connection places

Places where students connect to the Internet were also investigated. Table 4.5 shows the number and percentages of students according to the places where they connect to the Internet. Results showed that participants of all cases mostly connected to the Internet at home. While there was one student (8.3%) in Case-1 and one student (5%)

in Case-2 who connected to the Internet both at home and at an Internet cafe, there were none in Case-3. On the other hand, the number of students who connected to the Internet at home and school was three (15%) in Case-2 and two (25%) in Case-3; but there were none in Case-1 due to the fact that the students' school has no Internet connection. Lastly, only one (8.3%) student in Case-1 stated that he connected to the Internet in another place.

Table 4.5 – *Internet connection places*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
Home	11	91.70	16	80.00	6	75.00
Home & Internet Cafe	1	8.30	1	5.00	-	-
Home & School	-	-	3	15.00	2	25.00
Other places	1	8.30	-	-	-	-

4.1.4 Purpose of using the Internet

Students' purpose of using the Internet were investigated in order to learn about their major activities during Internet usage. As can be seen in Table 4.6, the students' primary purpose of using the Internet was to do research and prepare homework with a frequency of 12 (100%) for Case-1, 15 (75%) for Case-2, and eight (100%) Case-3. The other major activity, for all three cases, was to do recreational activities such as watching movies, films and listening to music. Playing games on the Internet was another activity performed by the students with a frequency of 10 (83.3%) for Case-1, 16 (80%) for Case-2, and seven (87.5%) for Case-3. Moreover, while the students of Case-2 (n = 16, 80%) and Case-3 (n = 7, 87.5%) mostly connected to the Internet in order to communicate (e.g., via social networks, mail), the students from Case-1 (n = 6, 50%) ranked it as the least. Only one student (5%) from Case-2 stated that he used the Internet in order to undertake hacking activities.

Table 4.6 – *Purpose of Internet usage*

	Curricular		Extra-Curricular		After-School	
	n	***%***	***n***	***%***	***n***	***%***
Research & homework	12	100	15	75.00	8	100
Recreational (films, music)	11	91.70	17	85.00	7	87.50
Playing games	10	83.30	10	50.00	6	75.00
Communication (e.g., social network, mail)	6	50.00	16	80.00	7	87.50
Other: Hacking			1	5.00		

4.1.5 Weekly Internet usage hours

Students' weekly Internet usage hours were investigated and are presented in Table 4.7. A high percentage of students in Case-1 (*n* = 4, 33.3%) and Case-2 (*n* = 7, 35%) used the Internet less than three hours a week. In Case-3, a high percentage (*n* = 5, 62.5%) of students used the Internet for three to five hours a week. Students who used the Internet for more than seven hours was moderately high for Case-1 (*n* = 3, 25%) and Case-2 (*n* = 6, 30%), but there was a low percentage for the students of Case-3 (*n* = 1, 12.5%).

Table 4.7 – *Weekly Internet usage hours*

	Curricular		Extra-Curricular		After-School	
	n	***%***	***n***	***%***	***n***	***%***
< 3 hours	4	33.30	7	35.00	1	12.50
3-5 hours	3	25.00	6	30.00	5	62.50
6-7 hours	2	16.70	1	5.00	1	12.50
> 7 hours	3	25.00	6	30.00	1	12.50

4.1.6 Experience in games, 3D VWs and programming

It was investigated whether or not students played computer/video games, or had any prior experience in 3D VW, or programming. The results are presented in Table 4.8. There was only one student who did not play computer or video games in each case. The majority of the students played computer/video games. Students' experience in

3D VWs was also investigated. While most of the students in Case-2 (n = 14, 70%) and Case-3 (n = 7, 87.5%) stated that they had 3D VW experience, there were only two students (16.7%) in Case-1. Lastly, students were asked whether they had learned or used any programming tool or language in the past. According to the students' percentage, it was high in Case-3 (n = 3, 37.5%), less in Case-2 (n = 4, 20%), and the least in Case-1 (n = 1, 8.3%).

Table 4.8 – *Having experience in gaming, 3D VW and programming*

	Curricular		**Extra-Curricular**		**After-School**	
	n	***%***	***n***	***%***	***n***	***%***
Playing games	11	91.70	19	85.00	7	87.50
3D VWs	2	16.70	14	70.00	7	87.50
Programing	1	8.30	4	20.00	3	37.50

4.1.7 Game playing experience

Students' gaming experience was also investigated since it may affect the use of VW in the current study. Table 4.9 shows that most of the students in all three cases have been playing computer games more than three years. While three (25%) students in Case-1 and seven (35%) students in Case-2 have been playing games for about 1-3 years, there was only two (25%) students in Case-3. The number who had been playing computer games for about one year was one (5%) in Case-2 and two (25%) in Case-3.

Table 4.9 – *Game playing experience*

	Curricular		**Extra-Curricular**		**After-School**	
	f	***%***	***n***	***%***	***n***	***%***
< 1 year	-	-	1	5.00	2	25.00
1-3 years	3	25.00	7	35.00	2	25.00
> 3 years	8	66.70	11	55.00	3	37.50

4.2 Sub RQ – Perceptions about SDP

Participants' perception about SDP were gathered via questionnaires. The use of SDP was comprised of two different programs. One was the viewer for connecting to and navigating within 3D virtual world, and the other was for building the S4OS code. Imprudence viewer, among various other viewers available on the market, was used in the current study due to its mostly Turkish interface and stability throughout this study. Moreover, students used another software, Scratch for OpenSim, a modification of Scratch, in order to program the objects in their 3D VW. In this section, students' perception are addressed based on the results of interviews and questionnaires. Firstly, students' perceptions about the ease of use of SDP are addressed. Then, students' perception about perceived usefulness of SDP are presented. In addition to students' perception related to interface, the teachers' perception are also presented in the related sections.

4.2.1 Case-1: Curricular Program

Perceived ease of use of SDP

Students' perceived ease of use of SDP was measured via the questionnaire. Descriptive results of the perceived ease of use questionnaire are given in Table 4.10. The questionnaire consists of four, five-point, Likert-type items. The number of participants in this case is eight. The overall mean of perceived ease of use score is low ($M = 3.90$) for this case when compared to the other cases. Students scored three items less than four points, except for Item-3. It could be said that they found the use of virtual worlds as easy. Students' and teachers ideas are in line with the questionnaire results. Students' mostly used the interface of both software without any major difficulties. However, since some of them were using them for the first time, they experienced some problems until they became accustomed to their usage over time. For example, one of the students stated that she experienced problems while working with pop-up windows inside the viewer in terms of finding the correct buttons and code.

> *CS1-STUDENT7- [When] you hide something, I always click on that, I click on here and it doesn't work. After that I get distracted for a while and I focus on it again, when you stop hiding it, I put it there. That was a bit of the problem for me.*
>
> *CS1-STUDENT7- Gizliyorsunuz, ben orayı hep tıklıyorum, şuraya basıyorum olmuyor. Ondan sonra kafam gidip geliyor, çekince koyuyorum. O biraz sıkıntı oluyordu bende.*

Similar to this student, most other students, especially those who had no previous 3D gaming experience, started to use the viewer easily after an initial adaptation period. In addition to students' perception, the teacher's ideas about the interface support the students' perception. One student without any gaming experience stated that;

> *CS1-STUDENT6- On this part I had some problems when I first came across that. However, [on the second part] I did not have much trouble. The ones there were easier.*
>
> *CS1-STUDENT6- Burada ben ilk karşıma çıktığında birazcık fazla zorlandım. Ancak [ikinci kısımda] o kadar fazla zorlanmadım. Oradakiler daha kolaydı.*

Table 4.10 – *Perceived ease of use results of Case-1*

	M	***SD***
1. Learning to use SDP was easy for me.	3.75	1.29
2. I find SDP easy to use	3.75	1.29
3. My interaction with SDP was clear and understandable.	4.17	0.72
4. It was easy for me to become skillful at using SDP	3.92	1.08
Total	**3.90**	

The interface of the viewer was mostly translated into Turkish; however, there were some English terms used on the interface and also some Turkish terms that were not meaningful for the students such as "Envanter," which means "Inventory" in English. the teacher of this case mentioned this issue as;

> *INTERVIEWER- Do you think the interface of this environment and messages were comprehensible for the students?*
>
> *CS1-TEACHER- Well, some of the menus on the upper part of the interface were in English, the lower part was in Turkish. ... She might have had problems on that part. Also, for example they should have clicked on the 'Save to Inventory' as written in the warning, but at first, most of them did not see that. But when you showed them the 'Save' button, they did it.*
>
> *INTERVIEWER- Bu ortamın ara yüzü ve mesajlar öğrenciler için anlaşılır mıydı sizce?*
>
> *CS1-TEACHER- Şimdi prograramın üst bölümü menülerin bir kısmı İngilizceydi, alt taraf Türkçeydi. ... O kısımda zorlanmış olabilir. Bir*

de mesela gelen uyarıda Envanterine sakla demeleri gerekiyordu, ama çoğu görmedi ilk başta. Ama sonra siz sakla butonunu gösterdiğiniz zaman yaptılar.

Ease of use of S4OS. Scratch for OpenSim (S4OS) is a Scratch-like program consisting of code blocks located in different colored groups. Students in this case mostly used S4OS easily in order to program objects, regardless of their previous experience. Besides, the students mostly transferred the code they had created in S4OS to the 3D environment easily. For example, CS1-STUDENT1 argued that;

> *CS1-STUDENT1- You just produce the virtual world code, delete [the previously existing one], and copy and paste, and you're done with it*

> *CS1-STUDENT1- Sanal dünya kodunu üretiyordun, [var olanı] siliyordun, kopyala yapıştır yapıyordun bitiyordu.*

The major problems that students confronted while using this software were their lack of knowledge on how to combine proper code blocks correctly and confused code blocks such as which one went on the right or the left. One student complained while working on decimal points on this software due to a lack of knowledge about decimal numbers; "I had problems in Scratch while dealing with decimal numbers. I was confused about those numbers in mathematics."

3D Environment and Navigation. Students were observed during the implementation phase. According to the observation forms and the interviews, they mostly navigated through the environment and completed the tasks easily. However, they experienced problems when their avatar was jammed between objects or where their avatar went outside the borders of the region. For example, CS1-STUDENT5 mentioned this problem;

> *CS1-STUDENT5- ... You know there was a bridge when I teleported into the house, CS1-STUDENT3- I and most of us fell down under the bridge. I mean we barely got out of there.*

> *CS1-STUDENT5- ... eve ışınlandığım zaman köprü vardı ya, ... CS1-STUDENT3, ben birçoğumuz o köprünün altına düştük. Yani çıkamadık neredeyse.*

Inventory and Positioning Objects. Getting objects from the inventory and positioning them in the environment were a major problem for the students. They had some problems due to the complexity of the inventory. All the things which belong to the students and are taken are saved to the inventory. In time, the inventory could contain many items from avatar skeletons to items of clothing, from notecards to other objects. It could be a problem sometimes for the students to find the right object and position it in the environment. One of the students stated that; “I had a problem in finding from the inventory.”

Another issue students had problems with was to position objects in the 3D environment. Students were required to drag and drop the objects from the inventory onto the right place in the 3D environment. However, students generally double-clicked on objects; in doing this, the objects came to their avatar’s hand and this was a problem for most of the students at first.

Perceived usefulness of SDP

Teaching the basics of programming is the primary focus of the implementation for all three cases in this study. Learning programming is generally considered difficult by learners of any age (Guzdial, 2004; Kelleher & Pausch, 2005). Thus, the perceived usefulness of SDP was measured via a questionnaire consisting of six, five-point, Likert-type items. Descriptive results of the perceived usefulness questionnaire are given in Table 4.11. As can be seen from Table 4.11, the overall mean value of the items on the questionnaire was quite high (M = 4.14). Besides, apart from Item-5, the mean of the other items were generally higher than four points. Only Item-5, using VW made it easier to communicate with the instructor, was lower than four points (M = 3.50).

Interview results support the questionnaire results. They confirmed that using 3D VWs facilitated the learning of programming. Most of the students (n = 4) in this case stated that they had learned the basics of programming owing to the use of the VW. For example, CS1-STUDENT1 stated that using VW facilitated his learning;

> *CS1-STUDENT1- We are learning to code. [Thanks to Scratch] we can learn it in an easier way now. We have learned what we are supposed to learn in an easier way.*

CS1-STUDENT1- Kodlamayı öğreniyoruz, [Scratch sayesinde] öğrenmemiz kolaylaştı. Öğrenmemiz gereken şeyleri daha kolay öğrendik.

Table 4.11 – *Perceived usefulness results of Case-1*

	M	***SD***
1. Using SDP as a tool for learning increased my academic performance	4.33	0.49
2. SDP allowed me to progress at my own pace.	4.17	0.84
3. Using SDP enhanced the effectiveness of my learning	4.08	0.90
4. Using SDP makes it easier for me to understand the lecture.	4.08	0.90
5. Using SDP makes it easier for me to communicate with the instructor.	3.50	1.51
6. Overall, SDP was useful in supporting my learning.	4.67	0.49
Total	**4.14**	

4.2.2 Case-2: Extra-Curricular Program

Perceived ease of use of SDP

Descriptive results of the perceived ease of use questionnaire are given in Table 4.12. The number of participants in this case is 20. The overall mean value of perceived ease of use scores is high (M = 4.09) for this case. Students scored the four items generally above four points except for the item related to interaction in the environment. It could be said from the results that they found the use of virtual worlds as easy and simple.

Table 4.12 – *Perceived ease of use results of Case-2*

	M	***SD***
1. Learning to use SDP was easy for me.	4.10	0.79
2. I find SDP easy to use	4.35	0.75
3. My interaction with SDP was clear and understandable.	3.90	1.45
4. It was easy for me to become skillful at using SDP	4.00	0.79
Total	**4.09**	

The qualitative results supported that students found the use of SDP software to be easy. The students mostly used the interface of both software without any major difficulties. However, they complained that they could not understand the function of some items on the interface such as “arazi sat, satın al, HM gönder” (sell land, buy, send DM). Since they could not understand them, they hesitated when using those functions. At this point, one student stated that the most frequently used functions should be made easier and easily accessible on the interface.

In addition to the students’ perception, the teacher’s perception about the interface was also taken. The teacher’s ideas were in line with their students’ perception about the ease of use of SDP. She stated that the students learned to use the software in just a short period of time and without much effort since they were accustomed to using these kinds of environment. Besides, she also added that students were engaged with the 3D environment and that they went beyond the scope of the objectives of the lesson from time to time due to the immersive features of the 3D environment. She suggested that texts and graphics on the buttons should be more age-appropriate to the students;

> *INTERVIEWER- Do you think the students were having difficulty while they were learning the environment and virtual world?*
>
> *CS2-TEACHER- I think they did not have any difficulties because these students… Since they are quite used to this kind of environments, they explore them quickly, but they get lost in them easily, too. I mean, when they say ‘Let me look at this part, let me do this and that, I want to change the clothes and the hairstyle of the character…’ and such, they might be getting away from their main goal. However, they are very fast at discovering the environment, at using it as well.*
>
> *INTERVIEWER- Do you think the interface and the messages of the virtual world viewer of this environment are understandable for the students?*
>
> *CS2-TEACHER- They were understandable. I only thought that the texts could have been made more engaging, that is, more interesting for the students. The simplest example for this is that the fonts could be changed. Apart from these, the system has an interface that is easy to learn and quite understandable.*
>
> *INTERVIEWER- Sizce öğrenciler ortamı ve sanal dünyayı kullanırken zorluk yaşıyor muydu?*

> *CS2-TEACHER- Bence hiç zorluk yaşamadılar. Çünkü bu çocuklar ... bu tarz ortamlara çok alışkın oldukları için çabuk keşfediyorlar, sadece çabuk da kayboluyorlar. Yani dur bir şuraya bakayım, öyle yapayım, işte karakterin kıyafetini değiştireyim, saçını değiştireyim falan derken amaçtan sapıyor olabilirler ama ortamı keşfetmek konusunda bence gayet hızlılar, kullanmak konusunda da hızlılar.*
>
> *INTERVIEWER- Peki sizce bu ortamın sanal dünyaları görüntüleyicisinin ara yüzü ve mesajları öğrenciler için anlaşılır mıydı?*
>
> *CS2-TEACHER- Anlaşılır. Sadece metinler daha sempatik, yani daha çocuk seviyesine uygun olabilir diye düşündüm ara yüz sadece öğrencinin daha ilgisini çekecek hale getirilebilir. Buna en basit örnek; yazı tipi bile değiştirilebilir. Bunun dışında öğrenmesi kolay ve gayet anlaşılır bir ara yüzü var sistemin.*

Ease of use of S4OS. Students in this case mostly found the use of S4OS easier in time when building the code for their tasks. The most significant problems students confronted while using this software was not being able to find the code or the appropriate names for defining variables. They had difficulty in creating code files for the objects, using code blocks related to logical operators, and moving 3D objects in the X, Y and Z axes. Transferring code from S4OS to the 3D environment was stated as easy by most of the students. CS2-STUDENT2 highlighted this by saying; "Transferring was easy, I delete the code on the current file, then just did copy and paste of the new code."

3D Environment and Navigation. Jamming of avatars in the 3D environment was also a major issue for this case. Avatars of users jammed into walls, underbrush and so on from time to time. Going outside the border of the region was another issue that students faced. For example, CS2-STUDENT10 mentioned that his avatar jammed into walls and kept going outside the region and then he lost control of his avatar;

> *CS2-STUDENT10- The other parts in which I've had difficulty in… When I was walking, I got stuck in some of the walls. And I also had some trouble in this too; I always used to walk by the sea, when I got in the sea, I was done for, I used to lose control and had to teleport.*
>
> *CS2-STUDENT10- Başka zorlandığım yerler… Yani bazı duvarların falan içine sıkıştım yürürken. Bir de şeyde zorlandım, hep deniz kenarında yürüyordum, denize girdiğim anda gidiyordu, kontrolü kaybediyordum, ışıklanmak zorunda kalıyordum.*

There were some textual information on the boards and panels in the environment. They could take a long time to load sometimes due to lags and bandwidth issues. Reading them could be difficult on these occasions. Students mostly navigated through the environment and completed the tasks easily according to the observation forms and interviews. Only those with no prior experience of computer or video games had any difficulty when they started out. One of the students highlighted this as;

> *CS2-STUDENT8- For example, the first time I was starting up the avatar, I thought we were supposed to do it with the letters [on the keyboard]. For instance it could be like 'f' to go fast and 's' to go slow, I thought about this but then I figured out the arrows [direction keys].*
>
> *CS2-STUDENT8- Avatarı mesela ilk hareket ettirirken, ben [klavyedeki] harflerle yapacağımızı sanıyordum. Mesela 'h' hızlı, 'y' yavaş gibi öyle olabilirdi belki, onu düşündüm sonra aklıma geldi o ok işaretleri [yön tuşları].*

Inventory and Positioning Objects. Getting objects from the inventory and positioning them in the 3D environment was a major issue for some of the students in this case. Taking objects from the inventory, finding them and transferring them to the 3D environment were other problems they experienced. CS2-STUDENT5 expressed this as "I had problem in taking something from the inventory, I could not understand it." Another issue that students had problems with was to position the objects in the 3D environment. Rotating, moving, and deciding on the direction of the objects were the most frequently stated issues that the students were confronted with. For example, CS2-STUDENT5 highlighted this problem as;

> *CS2-STUDENT5- It is a bit hard to place things onto wherever you want. You know changing the place of something, since I can't frame it properly while doing that and you can't frame it for each centimeter, [you fail at changing its place.]*
>
> *CS2-STUDENT5- [Bir şeyleri] yerleştirmek biraz zor oluyor istediğin yere. Hani böyle yerini değiştirmek var ya, onu yaparken tam ayarlayamadığın, için santimi santimine ayarlayamadığın için gidiyor.*

Perceived usefulness of SDP

The perceived usefulness of SDP was also measured via a questionnaire for this case. Descriptive results of the perceived usefulness questionnaire are given in Table 4.13.

As can be seen, the overall mean value of items on the questionnaire was moderate (M = 3.91) for this case when compared to the other cases. Although the means of Item-1 and Item-6 were higher than four points, the others were lower than four points. The lowest mean belongs to Item-2 (M = 3.65), using VW allowed students to progress at their own pace.

Table 4.13 – *Perceived usefulness results of Case-2*

	M	***SD***
1. Using SDP as a tool for learning increased my academic performance	4.05	0.76
2. SDP allowed me to progress at my own pace.	3.65	0.99
3. Using SDP enhanced the effectiveness of my learning	3.80	0.89
4. Using SDP makes it easier for me to understand the lecture.	3.85	0.88
5. Using SDP makes it easier for me to communicate with the instructor.	3.85	1.09
6. Overall, SDP was useful in supporting my learning.	4.25	1.25
Total	**3.91**	

In contrast to the moderately low scores in the questionnaire, the qualitative results of the interview analysis indicated that most of the students (n = 8) in this case highlighted that they learned programming easier owing to the use of VWs. CS2-STUDENT7 commented about this as “I learned programming better and how to use computers in a more meaningful manner.” Similar to CS2-STUDENT7, another student stated that he learned the logic of programming and added that the tasks and stories helped him learn the logic of programming better.

In addition to the students’ ideas about facilitating the learning of programming, the teacher’s ideas were also taken with regard to this issue. She addressed another aspect of using VWs. She argued that learning objectives of the lesson were well-defined for both students and teachers and it was possible to understand what had been learned at the end of the tasks;

CS2-TEACHER- …The most important advantage of virtual world is that in each lesson, task planning is neatly arranged. And we can come to a profound conclusion such as 'We have learned the sensing code blocks or motion code blocks today.' at the end of the lesson and when the task has been completed.

CS2-TEACHER- …Sanal dünyanın en önemli avantajı her dersin çok net görev planlaması yapılmış olması. Dersin sonunda görev tamamlandığında da Scratch'te biz bugün işte algılama komutlarını öğrendik, hareket komutlarını öğrendik diye net bir sonuca varıyoruz.

4.2.3 Case-3: After-School Program

Perceived ease of use of SDP

The number of participants in this case is eight. Descriptive results of the perceived ease of use questionnaire are given in Table 4.14. The overall mean of the perceived ease of use scores is high (M = 4.19) for this case. Students of this case scored all items higher than four points, and therefore, their scores are the highest when compared to the other cases. These results could be interpreted as students in this case found the software easy to use. In this case, the students mostly had previous 3D gaming experience, and were therefore familiar with these kinds of environments and were easily accustomed to using the viewer.

Table 4.14 – *Perceived ease of use results of Case-3*

	M	***SD***
1. Learning to use SDP was easy for me.	4.13	1.13
2. I find SDP easy to use	4.25	0.71
3. My interaction with SDP was clear and understandable.	4.25	0.89
4. It was easy for me to become skillful at using SDP	4.13	0.99
Total	**4.19**	

On the other hand, one of the students did not have any prior programming experience and was unfamiliar with block-based programming tools. He argued that he first needed to learn the logic of S4OS and how to construct code from the beginning. The same student also had problems with some of the English terms that were not translated into Turkish in S4OS;

> *CS3-STUDENT1- I think the logic of Scratch should have been taught because I could not understand how to use it the first time I saw Scratch… There were some English texts and I was having difficulties about that.*
>
> *CS3-STUDENT1- Scratch'in mantığının öğretilmesi gerekirdi bence. Çünkü Scratch'i ben ilk gördüğümde nasıl kullanacağım anlayamamıştım. …Scratch'te İngilizce bazı şeyler yazıyordu, ben onda zorlanıyordum.*

Ease of use of S4OS. Understanding the function of code blocks was mostly easy for the students since it is written on the blocks such as "move 1 meter." Transferring the code from S4OS to the 3D environment was found easy by the students as well. Majors problems experienced with the use of this software were being unable to combine code blocks, use Turkish characters in variable names, and find code blocks and numbers with decimal points.

3D Environment and Navigation. Students in this case were more experienced in 3D worlds such as Minecraft than the other students. They expressed their perception about the 3D environment by comparing it with their past experience. They were impressed when they first saw the 3D VW environment. For example, CS3-STUDENT5 found the 3D environment similar to Minecraft. Results indicated that few students in this case had problems with their avatar's getting jammed or other navigation problems.

Inventory and Positioning Objects. Positioning objects from the inventory to the 3D environment was a major issue for this case, as well. The axes of the 3D environment (X, Y and Z), determining the object's movement direction, and locating the objects into the proper place on the ground were all problems that confronted the students. CS3-STUDENT4 highlighted the positioning problems as;

> *CS3-STUDENT4- About robots… Sometimes robots went into the ground. And it was hard to place [the robot] on the ground. I experienced this sort of problems.*
>
> *CS3-STUDENT4- Robotlarla ilgili… Robot bazen yerin içine giriyordu. Bir de [robotu] yerin tam üstüne getirmek zordu. Öyle zorluklarla karşılaştım..*

Perceived usefulness of SDP

Perceived usefulness of SDP was measured via the questionnaire consisting of six, five-point, Likert-type items in this case, as well. Descriptive results of the perceived usefulness questionnaire are given in Table 4.15. The overall mean of items on the questionnaire was quite high (*M* = 4.13), which was similar to Case-1. It can be seen from Table 4.15 that except for Item-4 and Item-5, the mean of the other items were generally higher than four points. Only Item-4 (*M* = 3.88), using VW made it easier for students to understand the lecture, and Item-5 (*M* = 3.88), using VW made it easier to communicate with the instructor, were moderately lower than four points.

Table 4.15 – *Perceived usefulness results of Case-3*

	M	***SD***
1. Using SDP as a tool for learning increased my academic performance	4.50	0.54
2. SDP allowed me to progress at my own pace.	4.13	0.64
3. Using SDP enhanced the effectiveness of my learning	4.13	0.84
4. Using SDP makes it easier for me to understand the lecture.	3.88	0.84
5. Using SDP makes it easier for me to communicate with the instructor.	3.88	0.99
6. Overall, SDP was useful in supporting my learning.	4.25	0.89
Total	**4.13**	

Qualitative results of the interviews showed that the students perceived using VW as a facilitator of their learning. Out of the eight total students in this case, four of them mentioned that 3D VWs facilitated learning programming to some extent. For example, one student argued that activities in the VW helped him learn programming better. Similar ideas were stated by the other students. Besides, one of the students in this case highlighted another important point; arguing that using the VW accelerated her learning. She also added that what she learned on the course helped her with activities in her ICT lessons.

CS3-STUDENT3- I can say that this has accelerated my learning process. In computer lessons we learn to write code on code.org more often. It's really helped me on that.

CS3-STUDENT3- Öğrenmemi hızlandırdı diyebilirim. Biz bilgisayar derslerindecode.org'tan kod yazmayı öğreniyoruz daha çok. Orada mesela çok yardımcı oldu bana.

4.2.4 Cross-Case Analysis

The results of the analysis taken across all of the cases are provided in this section. The similarities and differences of the cases are presented in the following parts. Firstly, the overall perception of students and teachers across the cases was elaborated on. Results for the perceived ease of use across the cases are presented in Table 4.16. The mean scores of Item-1, Item-2, and Item-4 were less than four points for Case-1. Students of Case-1 were unfamiliar with the 3D virtual environment and its use at the beginning since they mostly had not played 3D games when compared to the students of the other cases. Therefore, using the viewer may have been difficult at the beginning for the students of this case. However, in time they became accustomed to using the viewer. The mean scores of Item-1, Item-2, and Item-4 in Case-2 were more than four points, except for Item-3. Some terms were either in English or their translation to Turkish seemed meaningless for the students to understand, which was stated by a few of the students in Case-2. That could be the reason why Item-3, which is about clearness and comprehensivity of the interaction in 3D environment, was scored below four points in Case-2. The mean scores of all items in Case-3 were generally higher than four points. As can be seen from Table 4.16, the overall highest mean scores belong to the students of Case-3, Case-2, and Case-1, respectively.

S4OS was another program used in the scope of SDP. Students mostly used this software to build code and transfer it to the 3D environment without experiencing any major difficulties. However, they did experience some problems while using this software. Finding the code blocks, not being able to combine them and confusing numbers with decimal points were problems experienced by students of Case-1 and Case-3. Defining variables with appropriate names was a problem experienced by students in Case-2 and Case-3. However, creating code files for the objects, using logical operators, and code blocks related to moving the objects in the X, Y and Z axes were difficulties faced only by students of Case-2.

The 3D environment and navigation in the environment was another emerging sub-theme about the ease of use of SDP. Most of the students generally became accustomed to using the programs after an adaptation period and they were then able to navigate in the 3D environment without much difficulty in all three case. However, they did encounter some difficulties on occasion. The difficulties faced mutually across all three cases were avatar's jamming between objects and avatar's going outside of the region. Those were the most significant problems for students in all the cases. Slow loading of textuals on boards was also a problem stated by the students of Case-2.

Table 4.16 – *Perceived ease of use results across the cases*

	Case-1		**Case-2**		**Case-3**	
	M	***SD***	***M***	***SD***	***M***	***SD***
1. Learning to use SDP was easy for me.	3.75	1.29	4.10	0.79	4.13	1.13
2. I find SDP easy to use	3.75	1.29	4.35	0.75	4.25	0.71
3. My interaction with SDP was clear and understandable.	4.17	0.72	3.90	1.45	4.25	0.89
4. It was easy for me to become skillful at using SDP	3.92	1.08	4.00	0.79	4.13	0.99
Total	**3.90**		**4.09**		**4.19**	

Inventory and positioning of objects from the inventory onto the 3D environment was other emerging sub-theme about the ease of use of SDP. The problems which students experienced about the inventory were similar to some extent among the cases. Complexity of the inventory, finding objects in the inventory, taking objects to the inventory, positioning them from the inventory to the 3D environment were the most significant problems faced by the students of all three cases. For example, the students all faced the problem of getting objects to their avatar's hand when placing them to the ground in the 3D environment since they were double-clicking on the objects in the inventory rather than dragging-and-dropping the object from the inventory to the ground.

Perceived usefulness of SDP was investigated with the help of a questionnaire and interviews administrated with the students. Table 4.17 presents the perceived usefulness results of the questionnaire across all three cases. As can be seen from Table

4.17, the overall mean value of perceived usefulness was quite high for Case-1 (*M* = 4.14) and Case-3 (*M* = 4.13), but it was moderate in Case-2 (*M* = 3.91). The results show that students in Case-2 did not benefit from the VW in terms of learning at their own pace as much as students in the other cases. This might be due to a higher number of students in Case-2 as well as the group members themselves. Group members could affect each other in terms of completing tasks at the same time. When the mean of the items were examined across the cases, it was found in all three cases that the mean of Item-5, using VW made it easier to communicate with the instructor, was lower than four points. This was not surprising because students in all three cases were able to speak face-to-face easily with their teacher.

Qualitative results of the interviews proved that using the VW facilitated the students' learning of programming. This is a remarkable result because students generally perceive the learning of programming as being difficult. The results indicated that most of the students across all three cases thought that using the VW facilitated learning the basics of programming. Besides, the teacher of Case-2 defined the learning objectives of the study as well-planned for both students and teachers.

Table 4.17 – *Perceived usefulness results across the cases*

	Case-1		**Case-2**		**Case-3**	
	M	***SD***	***M***	***SD***	***M***	***SD***
1. Using SDP as a tool for learning increased my academic performance	4.33	0.49	4.05	0.76	4.50	0.54
2. SDP allowed me to progress at my own pace.	4.17	0.84	3.65	0.99	4.13	0.64
3. Using SDP enhanced the effectiveness of my learning	4.08	0.90	3.80	0.89	4.13	0.84
4. Using SDP makes it easier for me to understand the lecture.	4.08	0.90	3.85	0.88	3.88	0.84
5. Using SDP makes it easier for me to communicate with the instructor.	3.50	1.51	3.85	1.09	3.88	0.99
6. Overall, SDP was useful in supporting my learning.	4.67	0.49	4.25	1.25	4.25	0.89
Total	**4.14**		**3.91**		**4.13**	

4.3 Sub RQ – Affordances of SDP

VWs have some affordances when used for educational purposes (Duncan et al., 2012; Warburton, 2009)**.** Using virtual worlds in programming education can bring about some affordances too. The results revealed seven sub-themes for affordance, which are elaborated on in the following parts of this section.

4.3.1 Case-1: Curricular Program

Having Fun

Having fun is an important drive for students to like programming and to increase their level of participation. Of the total students in this case, four of them stated that they had fun during the activities and liked programming thanks to the 3D VW. For example, CS1-STUDENT5 explained this as;

> *CS1-STUDENT5- Sir, the subjects were quite fun. For example if there was 105 tasks instead of 15 tasks, that'd be better.*
>
> *CS1-STUDENT5- Hocam, konular çok eğlenceliydi. Mesela 15 görev yerine 105 tane görev olsa daha güzel olurdu.*

The teacher's ideas are line with the students for this sub-theme. She stated that the lessons were enjoyable for the students and that students were active. She stated that the students were wondering what was coming up next in the following lessons and asking questions about the following activities. She also added that using the VW not only increased the participation of some students, but that it also increased the level of students' interest towards the lesson in an appreciable way;

> *INTERVIEWER- Do you think virtual worlds have raised the students' interest in programming?*
>
> *CS1-TEACHER- It has aroused their interest in programming. But in the virtual environment. Also it might have only raised it for Scratch but [a] kid told me in the club that when s/he uses virtual world with Scratch, s/he is more interested…For example the kids were wondering the next tasks when they were done with the previous tasks. They asked questions such as 'How can it be done?' They were always curious.*
>
> *INTERVIEWER- Sanal dünyalar sizce öğrencilerin programlamaya ilgisini artırmış mıdır?*

CS1-TEACHER- Programlamaya ilgisini artırdı. Ama o sanal ortamda artırdı. Diğer taraftan sadece Scratch olarak da artırmış olabilir ama [bir] çocuk Scratch'le sanal ortamı kullandığında daha ilgili olduğunu bugün bana söyledi yani kulüpte... Mesela her bir görevden sonra diğer görevleri merak etti çocuklar. Nasıl yapılabilir gibi sorular sordular. Hep bir merak içindeydiler.

Personal Contribution

Another emerging sub-theme was personal contribution that was stated by the students as an affordance. Four students in this case thought of transferring the knowledge of programming that they had acquired during the lesson into their future life in some way. For example, both CS1-STUDENT2 and CS1-STUDENT6 thought that they could transfer the knowledge into real life and hoped to improve upon their knowledge;

CS1-STUDENT2- When we grow older, we might work on programming this way because half of the world is trying to learn to do programming now.

CS1-STUDENT6- Sir, we've learned that we can improve in coding in the future by learning it like this.

CS1-STUDENT2- Büyüyünce böyle programlamada çalışabiliriz. Çünkü dünyanın artık yarısı programlama öğrenmeye çalışıyor.

CS1-STUDENT6- Hocam, ileriki hayatımızda kodlamayı bu şekilde öğrenerekten daha çok fazla geliştirebileceğimizi öğrendik.

Gaining 3D Experience

The environment was 3D and also immersive. While few students in this case were familiar with such an environment, the others were not. Only one student in this case mentioned this experience as an affordance. He stated that he gained experience with the 3D environment. The teacher of this case also pointed out this theme; stating that it was a good experience for the students to be an avatar (character) and to interact with each other in the 3D environment;

CS1-TEACHER- The environment is three-dimensional… That is, there is a whole environment there, a world, the students being their own characters there… The kids pushing each other in that environment, I mean as characters, as avatars [has contributed to their experience.].

> *CS1-TEACHER- Ortam 3 boyutlu... Yani orada tamamen bir ortam var, bir dünya var, orada [öğrencilerin] kendi karakteri olması… Çocukların bulunduğu ortamda birbirlerini iteklemesi, yani karakter olarak, avatar olarak [onlara deneyim kazandırmıştır].*

Facilitating Group Working

Students paired-up with a friend and the aim was to complete tasks together. They were able to see and interact with each other in the 3D environment. Students' ideas were taken about group working through the interview questions. The results revealed that the VW enabled students to work together and facilitated group working. Out of all of the students, four of them argued that 3D VW enabled them to work together. For example, CS1-STUDENT1 highlighted that the VW let them work together and facilitated group working;

> *CS1-STUDENT1- [It's taught us] collaboration. [in virtual world] We were doing it with our friends so it was collaboration.*

> *CS1-STUDENT1- İşbirliğini [öğretti]. [sanal dünyada] Arkadaşımızla yapıyorduk onun için işbirliği oluyordu.*

Motivation

The effect of using VW on student motivation was asked to the teachers of the cases. The teacher of this case thought that using VW had a positive effect on student motivation towards the learning of programming. She stated that students would like to use Scratch in 3D VW rather than just using the original Scratch;

> *INTERVIEWER- How do you think virtual world has affected the students' motivation in terms of the issues about programming?*

> *CS1-TEACHER- I have asked the kids just now. They said 'Using Scratch in virtual world is more fun.' One of the students said that learning about programming in virtual world, that is using Scratch in virtual world, is more fun. I mean a student of mine said that it's increased his/her motivation.*

> *INTERVIEWER- Sizce sanal dünyanın öğrencilerin programlama konularına motivasyonu etkisi nasıl oldu?*

> *CS1-TEACHER- Biraz önce sordum çocuklara. 'Sanal dünya içerisinde Scratch kullanmak daha eğlenceli.' dediler. Çocuklardan biri sanal dünyada programlama öğrenmesinin, yani Scratch'i sanal*

> *dünyada kullanmanın daha eğlenceli olduğunu söyledi. Yani daha motivasyonunu artırdığını bir öğrencim söyledi.*

4.3.2 Case-2: Extra-Curricular Program

Having Fun

Some students (n = 4) stated that they had fun during the activities. They mentioned that they liked programming activities in the 3D environment as well as they had fun during the activities since the activities were both fun and adventurous. For example, CS2-STUDENT2 thought that the contribution of VWs was to teach programming in a fun way and to add some sense of adventure to their learning. He also argued that learning programming in the virtual world was more creative when compared to learning with the original Scratch application, because Scratch could become boring after a while due to its limited number of activities;

> *INTERVIEWER- What are the benefits of this environment, in your opinion?*
>
> *CS2-STUDENT2- I think the benefits [of it] are both teaching us about programming through fun, and also putting some adventure in it because I mean [for instance] there was a city with problems, we were fixing the bridge etc., we were saving the turtles… There was a story here. This world was huge and more creative.*
>
> *INTERVIEWER- Sizce bu ortamın size faydaları neler?*
>
> *CS2-STUDENT2- Bence bize faydaları hem eğlendirerek programlama öğretmek, hem de birazcık içine macera katmak. Çünkü yani şey sorunlu şehir falan vardı, işte köprüyü yapıyorduk falan, kaplumbağaları kurtarıyorduk… burada hikaye vardı. Bu dünya çok büyüktü ve daha yaratıcıydı.*

Two of the students argued that VW was more enjoyable when compared to other tools, such as original Scratch or Python. Using Scratch with VW was more enjoyable since VW was an entire world and consisted of creative stories.

The teacher's ideas were also taken for this sub-theme. Her ideas were in line with those of her students. She argued that lessons were fun and the students participated actively in order to achieve the mission of each lesson. She stated that using VW definitely increased the level of participation and students' awareness as well as their interest towards the lesson when compared to previous years;

INTERVIEWER- How do you think were the lessons going? Fun, boring…

CS2-TEACHER- I think they are quite enjoyable, [they are] the lessons in which the rate of participation is high, and the lessons in which students are definitely participating in order to succeed. [The student] enjoys it, but also wants to carry out the tasks well, I mean [s/he] wants to complete it.

INTERVIEWER- And do you think it's raised the students' rate of participation during the lessons?

CS2-TEACHER- I think it definitely has. I was the teacher of this club last year as well. There is a big difference between the point that we reached with my students last year and the point we've reached with this year's students. With the help of the studies we've done with this three-dimensional virtual environment, even the students' level of awareness is so different that the way they verbally express [things] has changed.

INTERVIEWER- Sizce dersler nasıl geçiyordu? Zevkli, sıkıcı…

CS2-TEACHER- Bence çok zevkli geçiyor, katılım oranının yüksek olduğu dersler, öğrencinin mutlaka başarıya ulaşmak isteyerek katıldığı dersler bir de. [Öğrenci] zevk alıyor, ama bir de görevi başarmak istiyor, yani onu sonlandırmak istiyor.

INTERVIEWER- Peki sizce derse katılımını falan artırdı mı öğrencilerin?

CS2-TEACHER- Bence kesinlikle artırdı. Ben geçen sene de bu kulübün öğretmeniydim. Geçen sene öğrencilerle geldiğimiz noktayla bu yıl bu 3 boyutlu sanal ortamda yaptığımız çalışmalar sonucunda öğrencilerin geldiği nokta, farkındalık seviyeleri bile o kadar farklı ki, yani sözel olarak ifade ediş şekli bile değişti.

Personal Contribution

Students of this case mentioned about some personal contributions as an affordance. The most-cited personal contribution as an affordance in this case was to gain experience in game programming. Of all the students in this case, two argued that they gained experience in game programming to some extent. For example, CS2-STUDENT5 explained this affordance as, "I gained insight about game programming, I learned it to some extent." Other personal contributions were to transfer the knowledge of programming to real life, to realize the mission of computing in a meaningful way, and to enhance their creativity and imagination. Each of these sub-themes were stated by only one student. For example, CS2-STUDENT6 argued that VWs enhanced his creativity and imagination, while CS2-STUDENT3 stated that he learned how to use computers in a more meaningful way;

> *CS2-STUDENT6- It has improved my creativity and my imagination.*
>
> *CS2-STUDENT3- I've improved myself in computer usage, I've learned to use [a] computer multi-functionally.*
>
> *CS2-STUDENT6- Yaratıcılığımı, hayal gücümü geliştirdi.*
>
> *CS2-STUDENT3- Bilgisayar kullanımımı geliştirdim, daha amaçlı kullanmayı öğrendim bilgisayarı.*

Gaining 3D Experience

Students of this case were more familiar with 3D environments than the students of the other cases since it was their own choice to join the game programming club. Therefore, they mentioned about this theme more. Three of the students stated that they gained experience in 3D and virtual reality. For example, CS2-STUDENT6 argued that he learned how to create 3D objects and resize them in a 3D environment;

> *CS2-STUDENT6- ... It has also contributed to, how can I put this, the experience of doing something on virtual environment because it is the three-dimensional [system] experience which has taught us how three-dimensional things are created, what they are like, how to maximize them, how to expand them, how to minimize them.*
>
> *CS2-STUDENT6- ... Bir de şeyi kazandırdı böyle bir, nasıl desem, sanal ortamda bir şeyi yapma tecrübesi. Çünkü 3 boyutlu şeylerin nasıl yaratıldığını, nasıl olduğunu, nasıl daha büyütüldüğünü, nasıl genişletildiğini, nasıl küçültüldüğünü bize daha çok 3 boyut tecrübesi verdi.*

Another student in this case pointed out that the 3D environment forced him to think in another dimension, 3D, when compared to two-dimensional (2D) tools such as with the original Scratch. He added that it was a different experience for him and it increased his spatial ability;

> *CS2-STUDENT2- At first we thought programming to be in two-dimensional way in the 4th grade because we were thinking it was always two-dimensional on Scratch, we'd never imagined it three-dimensionally. This virtual world has enabled us to think in a three-*

dimensional way. You have to think in a three-dimensional way since you make the character and for instance the stairs in 3D as well.

CS2-STUDENT2- Başta programlamayı 4. sınıfta böyle çok 2 boyutlu falan düşünüyorduk. Çünkü Scratch'te falan hep 2 boyutlu düşünüyorduk, hiç 3 boyutlu düşünemiyorduk. Bu sanal dünya daha çok 3 boyutlu düşünebilmemizi sağladı. Çünkü hem karakteri 3 boyutlu, hem de örneğin merdiven yapıyorsun 3 boyutlu düşünmen gerek.

Facilitating Group Working

Students worked in pairs in this case, as well. The results indicated that VW facilitated group working as an affordance. Out all the students in this case, half of them (n = 5) argued that the VW enabled them to work together and facilitated group working as they worked together more with the help of VW. One of students explained this issue very well. CS2-STUDENT2 mentioned that before learning programming, first, VW taught them how to do things together;

CS2-STUDENT2- [The virtual world] has taught us about team spirit, I mean collaboration. And also about programming.

CS2-STUDENT2- [Sanal dünya] önce takım arkadaşlığını, yani birliği öğretti, nasıl birlikte bir şey çözebileceğimizi öğretti. Sonra bir de programlamayı öğretti.

Motivation

The teacher of this case argued that using VW had a positive effect on the students' motivation towards learning programming, as well. Students assumed that they would have fun at the beginning of the lesson due to immersive features of the 3D environment. Therefore, she stated that this feeling of the students positively affected their motivation;

INTERVIEWER- How do you think virtual worlds have influenced the students' motivation about the issues of programming?

CS2-TEACHER- I think that they have been affected in a very positive way because most importantly, the fact that a student is entering a three-dimensional environment gives the impression to the student that s/he will have more fun. I mean s/he enters the lesson with that motivation. It doesn't matter whether the lesson is fun or not but when s/he runs into a three-dimensional environment, when s/he moves a character or an avatar, s/he views this concept more differently.

> *INTERVIEWER- Sizce sanal dünyalar öğrencilerin programlama konularına yönelik motivasyonuna etkisi nasıl oldu?*
>
> *CS2-TEACHER- Kesinlikle çok pozitif yönde etkilendiklerini düşünüyorum. Çünkü her şeyden önce öğrencinin, yani 3 boyutlu bir ortama giriş yapıyor olması belki biraz daha böyle eğleneceği şeyi veriyor çocukta. Yani öyle bir motiveyle giriyor derse. Dersin eğlenceli olup olmaması önemli değil ama, 3 boyutlu bir ortam çıkınca karşısına, bir karaktere hareket verince, bir avatarı olunca çok daha bambaşka bakıyor olaya.*

Other Affordances

There were some students in this case who had some prior programming experience. Those pre-experienced learners had the opportunity to use or see at least one programming tool or language prior to this course. The teacher of this case also had experience in the teaching of programming with other tools. Those students and teachers were asked to define the outstanding affordances of VW. The teacher of this case pointed to an outstanding affordance that was missing from the original Scratch. It was possible to see real LSL code after transforming the code built on S4OS; which was something different to the original Scratch. She mentioned that students realized that real code lay behind the ready-to-use puzzle blocks of Scratch. This was helpful for students to realize and investigate the real code of a programming language. Similar to the ideas expressed by the teacher, one student who also realized about the underlying code highlighted this by saying, “I was inspecting the generated code after the transformation. We wrote the real code there.”

> *CS2-TEACHER- ... After building the code on Scratch, they were translated into code and student actually realizes the fact that s/he gives the computer a code in a way the computer would understand it. This is the difference of the virtual world. But what does a kid do when using original Scratch? S/he actually controls the cat with an interface. In fact, there is a working programming language behind that interface. That is, in the normal version of Scratch, when s/he tells the cat to take 10 steps, s/he in fact writes an algorithm for Scratch, doesn’t s/he? I mean, s/he creates some kind of an sequence. S/he does not write that code blocks on her/his own, I do not think so. But what does [the student] do in the virtual world? After building the code blocks, s/he translates them in to her/his own language, using that [LSL] language. Therefore, I think the virtual world is more advantageous at this point. The kid takes this more seriously. In the other one, s/he thinks that there is a cartoon series.*

CS2-TEACHER- ... Scratch'te bloklar oluşturulduktan sonra koda çevriliyordu ve öğrenci aslında arka planda bilgisayara bilgisayarın anlayacağı bir makine diliyle kod verdiğini fark ediyor. Sanal dünyanın farkı bu. Ama normal Scratch kullanırken ne yapıyor çocuk? Aslında kediyi bir ara yüzle kontrol ediyor. O ara yüzün arkasında çalışan bir programlama dili var aslında. Kediye 10 adım git derken yani bir algoritma yazıyor aslında Scratch'e, kendi normal Scratch'te öyle değil mi? Yani bir sıralama veriyor. O komutu kendi yazmıyor, ben öyle düşünüyorum. Ama sanal dünyada ne yapıyordu? Komutları dizdikten sonra onu [kendi] diline çeviriyordu tekrar, o [LSL] dilini kullanıyordu. Dolayısıyla burada sanal dünyanın avantajlı olduğunu düşünüyorum. Çocuk daha ciddiyetle bakıyor olaya. Ötekinde bir çizgi dizi varmış gibi düşünüyor.

4.3.3 Case-3: After-School Program

Having Fun

The students of Case-3 expressed their feelings about the lesson almost matching the students in the other cases. Three of the students mentioned having had fun and they liked programming thanks to VW. One of the students highlighted an important point with regard to the like/dislike of programming. She stated that she did not like programming and doing such activities on Scratch before participating in the VW lesson. However, she stated that she now likes programming and she would like to do more of the activities on her own on Scratch after participating in the lesson. After she found a taste for programming, she realized how to make use of computers in different ways. She has a passion for programming now and wants to build her own objects;

INTERVIEWER- So, have the things you have experienced here made you like programming?

CS3-STUDENT6- Yes. Actually, I did not like it at first, I mean when they put Scratch in front of me, I would push the computer away. But now, I beg to try opening Scratch.

INTERVIEWER- Peki programlamayı sevmeni sağladı mı burada yaşadığın şeyler?

CS3-STUDENT6- Evet. Aslında, başta pek sevmiyordum, neredeyse Scratch'i önüme koysalardı böyle bilgisayarı iteklemeyi deniyordum. Ama şimdi Scratch'i ben kendim yalvararak açmayı deniyorum.

Personal Contribution

The most stated affordance were the transfer of knowledge of programming into real life and to gain some experience in game programming. Each of these sub-themes were mentioned by two students. CS3-STUDENT6 highlighted this affordance as she could transfer what she learned into her future life and she hopes to make use of it;

> *CS3-STUDENT6- I will be able to do many more things on it such as building something differently, or that I can fix my own games, maybe I can be rich in the future when I grow up with that thing by editing programs.*

> *CS3-STUDENT6- …daha çok şey yapabileceğim, yani kendi oyunlarımı düzeltebilmem, belki gelecekte büyüyünce belki o şeyle belki programları düzelterek zengin olabilirim.*

Another personal contribution was to gain experience in game programming. Two of the students argued that what they learned throughout the study could contribute to game programming. CS3-STUDENT5 hopes to create his own games in the future with the help of what he learned. Another personal contribution stated by one student was to realize the mission of computers in a more meaningful way. CS3-STUDENT6 argued that she realized another important point of computers; having, before the course, only used her computer in order to do homework or watch videos, but through the help of the VW, she wants to create her own artefacts by using code blocks on Scratch;

> *CS3-STUDENT6- I've been taking programming more seriously. At first I was thinking that a computer was just something to watch videos and do my homework.. But then my brother showed Scratch and that sort of things, I opened it one or two times then, but after Scratch [used with VW] has made programming more fun, I am going to make my own artefacts by building options in virtual world*

> *CS3-STUDENT6- programlamaya doğru daha çok önemsedim. Önceden bilgisayarın sadece video izlemek ve ödevlerimi yapmak için kullanıyordum. Ama sonra ağabeyim Scratch ve benzeri şeyleri bana gösterince o sıralarda 1-2 kere açtım, ama [sanal dünyada] kullandığımız Scratch programlamayı eğlenceli hale getirince sanal dünyada inşa etme seçeneklerini kullanarak kendi eşyalarımı yapacağım.*

Gaining 3D Experience

Out of all the students in this case, two of them mentioned gaining experience on 3D. One student argued that, “I had experience on such environments before, but they did not move. The experience here was more pleasing because they are moving.” The other student mentioned that it was possible to learn what to do and how to move objects in a 3D environment. He stated this as;

> *CS3-STUDENT3- It has taught me what I can do on that virtual world, which things I can move.*
>
> *CS3-STUDENT3- O sanal dünyayı neleri yapabileceğimi öğretti, neleri hareket ettirebileceğimi öğretti.*

Facilitating Group Working

Facilitating group working and enabling students to work together was the emerging theme in this case, as well. Most of students (n = 4) in this case thought that the VW, with its unique features, enabled them to work together and facilitated group working. For example, one of them stated that his team peer helped him, and that they completed most of the tasks together in the 3D environment;

> *CS3-STUDENT4- I did most of the tasks with my friend [in the virtual world]. ... [therefore] I managed better things in the tasks, I have learned more about programming*
>
> *CS3-STUDENT4- Çoğu görevleri arkadaşımla [sanal dünyada] birlikte yapmıştık. .... [Böylelikle] görevlerde daha iyi şeyleri başarabildim, daha iyi öğrendim programlamayı.*

4.3.4 Cross-Case Analysis

Affordances revealed were varied across the cases. Emerging codes were categorized under six sub-themes and their frequencies across the cases presented in Table 4.18. Having fun was the first affordance to emerge from the results. Most of the students across all three cases argued that they had fun in the activities and liked programming owing to VWs. Teachers of the cases argued that the activities were fun and increased the interest and participation of students towards the lesson.

Personal contribution was another emerging sub-theme as an affordance. The only emerging code in all three cases was to use the transfer of programming knowledge into real life. At least one student from each of the three cases mentioned this contribution. While students from Case-2 and Case-3 listed gaining experience in game programming and realizing the mission of computers in a meaningful way as a personal affordance, there was no emerging codes for Case-1. On the other hand, the code related to enhancing creativity and imagination only emerged in Case-2. As to the other emerging sub-theme, gaining 3D experience, at least one student from each case mentioned this. However, the number of students was a bit high for Case-2 since their club was primarily focused on game programming and were therefore more willing to do the activities in 3D.

Table 4.18 – *Frequencies of Affordances across the Cases*

	Case-1		***Case-2***		***Case-3***	
	n	***%***	***n***	***%***	***n***	***%***
Having fun during the lesson	4	57	4	40	3	50
Personal Contribution						
Transferring programming knowledge into real life	4	57	1	10	2	33
Experience in game programming	-	-	2	20	2	33
Realizing the real mission of computers	-	-	1	10	1	17
Enhancing creativity and imagination	-	-	1	10	-	-
Gaining experience on 3D	1	14	3	30	2	33
Facilitating group working	4	57	5	50	4	67

Facilitating group working is the other affordance of VW that emerged at the end of the analysis. Most of the students in all three cases argued that VW enabled them to work together and that it facilitated group working. Motivation was the other affordance stated by the teachers of both Case-1 and Case-2. The teachers of all of the three cases argued that using VW had a positive effect on the students' motivation. Lastly, students and teachers of each case were asked to list the affordance of using VW with Scratch when compared to other programming tools they had used. The affordance only mentioned by the participants of Case-2 was to see real code being generated. Since the students in this case were more aware of programming and the teacher more experienced in teaching programming, they realized the transformed code, and also recognized the importance of real code experience.

4.4 Sub RQ – Challenges of SDP

Challenges of SDP were investigated based on the interview and observation analysis of both students and teachers. Students had used SDP for about 15 hours in total and in that time they faced some challenges. Those challenges are elaborated on in this section. Firstly, challenges that students encountered related to 3D environment are given. Then, the challenges related to equipment and infrastructure are addressed. Lastly, challenges related to tasks related to the 3D environment are elaborated on. The categories that emerged under these sub-themes are also addressed in the following parts of this section.

4.4.1 Case-1: Curricular Program

3D environment

Students faced challenges related to the 3D environment during their usage. Avatar-related problems and misuse of the environment were among the challenges that students of this case encountered. Avatars could sometimes load slowly or disappear. In such situations, the students were unable to see their avatars in the environment. One student in this case experienced this problem with his avatar. CS1-STUDENT5 highlighted this problem as, “My avatar disappeared suddenly from time to time, I could not customize it.” Another problem was due to misuse of the environment. Students sometimes could misuse the environment by trying different things and this could subsequently affect other students due to slowing the server down. One student encountered such a challenge since she tried many things on her computer and she stated that her virtual world crashed and it was then not possible to move in the 3D environment;

> *CS1-STUDENT7- I tried doing different things on our own computer when we were not using it. I combined everything together and a really interesting thing came out. My virtual world has collapsed, I mean one could not play games on it anymore.*

> *CS1-STUDENT7- Kendi bilgisayarımızda değişik şeyleri denedim kullanmadığımız zamanlarda. Hepsini birbirine kattım, çok değişik bir şey oldu. Sanal dünyam iflas etti, yani oyun oynanılmıyordu artık.*

Equipment and infrastructure

Equipment and infrastructure of the setting are important issues while using a VW, because the use of VWs requires high capability computers and a robust network connection among the computers (Purbrick & Lentczner, 2007). Students of this case mentioned about challenges related to equipment and infrastructure in four categories. The most mentioned challenge was computer freeze or an abrupt shutdown. Most of the students (n = 6) in this case faced this problem and mentioned it as a problem because the computers were largely old. They could freeze or abruptly shut down. CS1-STUDENT3 highlighted this challenge as, "My computer froze, then I had to use the next computer."

Electricity and inadequate lighting in the school setting were also challenges mentioned each by student. In the first part of the study, there was a power cut for about two hours and it was therefore not possible to do any activities that day. One student defined this problem as a challenge. The computer laboratory of the school was located on a lower floor and inadequate lighting in the laboratory was stated by one student as a problem.

Tasks

Reaching the place in the 3D environment for tasks, grasping them and their difficulty level also emerged as categories for this sub-theme. The first challenge stated by two of the students was to correctly reach the place where the tasks were to be undertaken. Students reported that waymarks on the ground were unclear or invisible due to the fact that textures were sometimes loading slowly. CS1-STUDENT7 commented on this problem as, "Waymarks on the ground were not clear; they looked blurred." Besides, one student reported that she was unable to notice those marks since they were on the ground.

Difficulty of tasks was another emerging sub-theme. Two of the students in this case mentioned that a few of the tasks were difficult and that they had difficulty in completing them. The other challenge stated by one student was about grasping the tasks. In the first part of the study, students were tasked with programming their robots in order to move them from the home location to the finish location along different paths. One of the students highlighted that the tasks in the first part were meaningless for her and she tried to do them by chance;

> *CS1-STUDENT6- [The things on the first part] were not very good. I mean I was trying to make things up and take wild guesses, depending on the possibility that they would be correct. ... There were some squared things like this. You had given us some cards. ... I was having a bit more difficulty while moving our robot.*

> *CS1-STUDENT6- [İlk kısımdakiler] pek iyi değildi. Yani ne bileyim kafadan ata ata yapmaya çalışıyordum belki tutarsa diye. ... orada kare kare böyle şeyler vardı. Siz de bize bazı kartlar vermiştiniz. ... robotumuzu götürmek için birazcık fazla zorlanıyordum.*

Moreover, the teacher's ideas were also taken about this sub-theme. She pointed out an important challenge related to the tasks' difficulty level. She argued that working with numbers with decimals could be hard for some students to grasp. She added that they might not know those numbers if they had not been previously informed;

> *CS1-TEACHER- ... For example when you got closer to the door on there… We wrote numbers such as 0,1. I wondered how a student could notice this without knowing them. When they were approaching the door, when it was opening, I wondered how a student could figure it out on his/her own. I mean it is a bit hard, I could not have figured it out there…*

> *CS1-TEACHER- ... Şimdi mesela orada kapıya yaklaştığımızda… 0,1 gibi sayılar yazdık. Bunu öğrenci bilmeden nasıl fark edebilir ben bunu merak ettim. Kapıya yaklaşınca kapının açılması o sayıyı nasıl kendi bilebilir bunu merak ettim. Yani o biraz hani [zordu],ben bilemezdim yani orada…*

4.4.2 Case-2: Extra-Curricular Program

3D environment

Students of this case encountered challenges more than the students of the other cases. Some students were unable to watch the help videos, take objects in the 3D environment, control the camera, or find the environment as it was left. In addition, avatar-related problems and misuse of the environment were the other challenges encountered by some of the students. Not being able to watch help videos were the most-cited challenge seen in this case. Out of the total number of students, five of them mentioned about this challenge. This might be due to fact that there were a high number of students in this case, and it was therefore not possible for the teacher to respond to the need for help of all students. Therefore, students preferred to watch help videos, and videos were not always openable due to various reasons. Opening videos was a problematic issue in the VW since it requires more capability from the computers and equally more bandwidth. For example, CS2-STUDENT1 argued that they were unable to watch some of the help videos;

> *CS2-STUDENT1- Through the end of the videos, I was not able to play the videos, I could not.*
>
> *CS2-STUDENT1- Videoların sonlara doğru filan videoları açamamaya başladım ben, açamadım.*

The other most-cited challenge was being unable to take objects and use them in the 3D environment. Of all the students, two of them argued that they were unable to take the necessary objects to complete the tasks. CS2-STUDENT9 commented about this problem as, “My box did not work, but my friend’s box was working, so I took the necessary materials from his material box.” Avatar-related problems was another challenge mentioned by two of the students. They mentioned that their avatar transformed to the default avatar when they teleported to another region. CS2-STUDENT2 highlighted this problem as, “When I went to another world [region], my customized avatar reverted back to the default appearance.”

Not being able to control the camera, changing structure of the environment in subsequent weeks and misuse of the environment were other challenges stated by one student. For example, CS2-STUDENT6 complained that it was not possible to find the environment as he had left it in the previous lesson. He highlighted this problem as, “Each week, some place in my VW would disappear, which I had no idea about.” This may be due to the fact that some students would misuse the environment on occasions, or could accidentally delete part of the environment by mistake. Misusing the environment was a challenge stated by one student, with CS2-STUDENT6 arguing that he built enormous objects that may affect the other objects in the environment;

> *CS2-STUDENT6- I found a bug [in the game]. You know that we can create giant things there, with those, I made a base camp on the sea. I made a huge area, I was walking above it.*

CS2-STUDENT6- Ben [oyunda] bug buldum. hani böyle dev şeyler yaratabiliyoruz ya, ben onlarla denizin üstünde bir üs kurdum. Böyle kocaman bir alan kurdum, üstünü geziniyordum.

Equipment and infrastructure

Challenges in this sub-theme could be categorized under three categories for this case. The most-cited challenge (n = 5) in this case was about network issues. The server was slowing down sometimes, and students could wait for the server for some time in order to log on or to respond to a request when the network connection was running slow. At those times, students faced challenges when they tried to log on to the OpenSim server. For example, CS2-STUDENT5 commented that; "It was too slow to log in sometimes."

Another challenge was about computer's freezing or unexpectedly shutting down. Of the total students in this case, three of them mentioned this challenge. CS2-STUDENT9 highlighted this problem as, "I encountered the problem of computer freeze the most." The last challenge mentioned by one student of this case was problems with the headset. There were headsets plugged into each computer in the school setting so that students were able to use them during the class hours. One of the students mentioned about a problem with his headset;

CS2-STUDENT5- You know… On the phone, the man talks, you cannot hear him.

CS2-STUDENT5- Kulaklıkta … adam konuşuyor ya, yani onu duyamıyorsun.

Tasks

Emerging codes could be categorized under three categories in this case: reaching the location of the tasks, grasping the tasks, and the difficulty level of the tasks. Reaching the location of the tasks was the most mentioned challenge (n = 4) by the students in this case. There were colored waymarks like footprints on the ground to help users navigate to the location of the tasks. Students needed to follow those markings in order to reach the location of the tasks to complete them. However, they sometimes confused the marks and could not reach the location. Some of the students reported that they did not notice the marks since they were located on the ground. Therefore, they might have skipped passed the location of the tasks. Some students mentioned this challenge as;

CS2-STUDENT5- I had difficulty finding the place of some of them because what shows the place of that thing is on the ground. Since it is on the ground and you cannot bend down, you cannot see it.

CS2-STUDENT5- Bazılarının yerini bulmakta sıkıntı çektim. Çünkü onun yerini gösteren şey yerde. Yerde olunca eğilemediğin için de göremiyorsun.

Grasping the tasks was seen as another type of challenge for three of the students. It was hard for the students to grasp what and how to perform the tasks for the first time. Sometimes, their mind was confused; however, after a while, they understood the task. CS2-STUDENT1 highlighted this problem as, "I had difficulty at first, and then I started to understand them when I examined the papers for a while." The last challenge stated by two of the students of this case was the difficulty level of the tasks. They argued that the level of tasks sometimes could be very difficult and that it was not possible to do them by thinking them over;

CS2-STUDENT7 - For example, you think "How can it be possible?" But sometimes it could be so hard that you cannot find it by thinking about it, and you think of these things etc.

CS2-STUDENT7 - Mesela düşünüyorsun bu nasıl olabilir. Ama bazen düşününce bulunulmayacak kadar zor olabiliyordu, işte şey düşünüyorsun işte böyle falan.

4.4.3 Case-3: After-School Program

3D environment

Students of this case encountered three kinds of challenge related to the 3D environment. Camera control, watching help videos, and changing structure of the environment were the challenges defined by different students in this case. For example, CS3-STUDENT1 complained that, "Codes on the object were removed which I could not understand, so I saved the code again." Not being able to watch help videos was another challenge encountered by one student, with CS3-STUDENT6 complaining that it was not possible to watch help videos. The other challenge mentioned was about control camera, with CS3-STUDENT3 highlighting this problem as, "I was unable to see anything when I was driving the car."

Equipment and infrastructure

Challenges related to this sub-theme stated by the students of this case could be listed as; computer freeze, shutdown, and network issues. Four of the students mentioned about the computer freeze or shutdown problem. Although the computers were high capabilities in this case, they were sometimes still inadequate to respond to the requests of the students. CS3-STUDENT3 mentioned this problem as, “The computer froze sometimes.”

The other challenge stated by three of the students was about the network. Connectivity problems with the network occurred due to various reasons in this case. Network connection was sometimes either off or slow. During these times, the students could not logon to the OpenSim server and had to wait for a while. CS3-STUDENT6 commented about this as, “Network connection seemed to be off sometimes, but it was on again later.”

Tasks

Results revealed only one type of challenge in this case, which was the grasping of the tasks. Two of the students encountered difficulty in grasping the tasks in this case. There were some terms such as specifying the coordinate axes (X, Y, or Z) in the tasks. When students did not understand these terms, they were unable to grasp the tasks. For example, CS3-STUDENT6 claimed that the story of one task made no sense to her and that she therefore had difficulty in grasping it;

> *CS3-STUDENT6- Just one of them [the stories] seemed a bit illogical to me. About how the carrousel would turn. The story of the carrousel felt a bit different.*
>
> *CS3-STUDENT6- Sadece bir tanesi [hikaye] biraz mantıksız geldi. Atlıkarıncanın nasıl döneceğiyle ilgili. Atlıkarıncanın biraz da olsa hikayesi farklı geldi*

4.4.4 Cross-Case Analysis

Challenges encountered were varied across the cases. Emerging codes were categorized under three sub-themes. Challenges related to the 3D environment was the first sub-theme; which consisted of six categories that varied in each case. Frequency of challenges across the cases are presented in Table 4.19. As can be seen from

Table 4.19, there were not so much mutual challenges for this category encountered across all three cases. There could be various factors causing the challenges. However, they may not have emerged when the conditions were appropriate or they may not be realized and perceived as a challenge by the students. For example, students of Case-1 mostly preferred not to watch help videos, preferring to seek help from their teacher; thus, they did not encounter such a challenge. Challenges in this category were generally dependent on the context and setting and stated by a student, and so it may not be accurate to attempt to compare them across the three cases.

Table 4.19 – *Frequencies of Challenges across the Cases*

	Case-1		**Case-2**		**Case-3**	
	n	***%***	***n***	***%***	***n***	***%***
3D environment						
Watching help videos	-	-	5	50	1	17
Controlling the camera	-	-	1	10	1	17
Changing structure of the environment	-	-	1	10	1	17
Avatar-related problems	1	14	2	20	-	-
Misuse of the environment	1	14	1	10	-	-
Taking objects	-	-	2	20	-	-
Equipment and Infrastructure						
Computer freeze / shutdown	6	86	3	30	4	67
Network issues	-	-	5	50	3	50
Other issues	2	29	1	10	-	-
Tasks						
Grasping the tasks	1	14	3	30	2	33
Reaching the location of the tasks	3	43	4	40	-	-
Difficulty of tasks	2	29	2	20	-	-

Equipment and infrastructure in the setting varied for each case. For example, the equipment and infrastructure at the school used in Case-1 was significantly poor. There was no Internet connection at the school, the computers were old and the computer laboratory was located in a basement of the school. Because of these reasons, students were invited to complete the second part of the study in the laboratory of the university's CEIT department. Although the equipment and infrastructure of the other cases were better organized, there were still similar challenges faced in those cases. In fact, the challenges related to the equipment and infrastructure were inevitable regardless of the case. Computers freezing or shutting down unexpectedly was the mutual challenge faced across all three cases. It was not possible to maintain an OpenSim server as well as having ready to use computers continuously. The important point was to somehow overcome the challenges in somehow. Electricity, lighting and others challenges were particular to the settings of each case.

The last sub-theme related to the tasks. At least one student from each case experienced challenges regarding the tasks since they could just not grasp them. Some tasks, their stories or terms used in the tasks did not make sense to some of the students, and those students therefore had difficulty in grasping the tasks. Therefore, tasks should be ensured that they are comprehensible to all students. The other challenges stated by the students of Case-1 and Case-2 were to reach the location of tasks and the difficulty level of the tasks. Some students in these two cases were poorly motivated in tackling the tasks. When left unattended, they would move around the environment randomly. They might have ignored or bypassed the hallmarks or might have not paid attention as much as the others had done. The teacher of Case-2 highlighted this problem, claiming that the students wandered around the environment and would explore it, but the teacher needed to also follow the timetable. However, it was hard for the teacher due to the high number of students and their diversified abilities as computer users. Students are heterogeneous; while some of them could finish activities on time, others might need considerable help or fall behind the activities completed by the other students. It was difficult for the teacher to keep all of the students in scope of the topic sometimes due to these reasons. On the other hand, students of this last case were less in number that the other cases, and were more motivated. It was also possible for the teacher to keep the lower number of students within the scope of the lesson;

> *INTERVIEWER- What are the negative sides of virtual worlds?*
>
> *CS2-TEACHER- … Negative sides are like this; yes the kids spend too much time on exploring new features in the environment or on exploring the environment on their own. I think there is a disadvantage*

here; I wish we could work with smaller groups and the readiness level of the students were the same because our group consists of 24 students. And 24 out of 24 students are not capable of [using] computers at the same level. Some of them explore things too fast and go on with the task, finish the task and get bored or they want to do something different. And some of them are left behind, they cannot understand the purpose, they get stuck on different things until the teacher helps them out. Therefore, yes, this might be the most important disadvantage of the virtual world. I think that this disintegration would decrease and it would reduce the negative sides of it, if smaller groups with students with similar knowledge levels created and if the teacher could reach the students more quickly.

INTERVIEWER- Sanal ortamların olumsuz yönleri sizce neler?

CS2-TEACHER- ... Olumsuz da şöyle; evet çocuklar ortamdaki farklı özellikleri keşfetmek için ya da ortamı kendi başına bir keşfetmek için çok fazla zaman kaybediyor. Burada dezavantaj bence şöyle de bir şey var; daha küçük gruplarla çalışılsa ve öğrencilerin hazır bulunuşluk seviyesi aynı olsa. Çünkü bizim grubumuz 24 öğrenciden oluşuyor[ve bu]24 öğrencinin 24'ü de aynı seviyede bilgisayar karşısında yetkinliğe sahip değiller. Kimisi çok hızlı keşfediyor ve göreve geçiyor, görevi bitiriyor sıkılıyor ya da başka şeyler yapmak istiyor. Kimisi de geri kalıyor, olayı anlayamıyor, öğretmen ona yardımcı olabilene kadar ki bölümde başka şeylere takılıyor. Dolayısıyla, evet, bu sanal dünyanın en önemli dezavantaj olabilir. Daha küçük grupla daha birbirine yakın bilgi seviyesine sahip öğrencilerle çalışılırsa, öğretmen daha hızlı öğrencilere ulaşabilirse bu dağılma azalacaktır diye düşünüyorum, olumsuz yönünü azaltacaktır diye düşünüyorum.

4.5 Sub RQ – Avatar Issues

Avatar is the graphical representation of users in the 3D environment. Learners interact with others and travel in the environment through their avatars (Yee, 2006). There were some avatar options for the students to choose from, as well as a variety of clothing options for their avatars to wear. Students were free to choose among the available avatar and clothing options. In this section, firstly, the students' feelings about the avatar representation and what kind of changes they made to their own avatars are elaborated on. Secondly, the purpose of customization in avatars is addressed. Thirdly, the most- and the least-liked aspects about the avatars are presented. Finally, the suggestions of students about the avatars are addressed.

4.5.1 Case-1: Curricular Program

Avatar Representation and Changes in Avatar

Students were represented by a default avatar when they first logged into the 3D VW environment. However, they were also then free to change their avatars among a number of options. They could change their overall avatar, as well as make certain changes to their look, such as the shape of their body and body parts, color of their skin and so on. Moreover, there were many clothing options for students to choose for their avatars, from jackets and trousers to shoes and hats. Students were asked whether or not they felt that they were well-represented by their avatars in the 3D environment. Interview results indicated that the most of the students reported that they felt this way. There was no-one in this case who argued the opposite. Differences among the avatars in the 3D environment stated by one of the students was the main factor behind this feeling.

It was investigated what kind of changes in avatars were applied by the students. Qualitative analysis of the interviews indicated that the students mostly changed their avatars' clothing, hair type, skin color and accessories. For example, CS1-STUDENT3 listed the changes of his avatar as, "I changed the skin color ... I put on a cowboy hat ... I wore a jacket."

Purpose of customization

Participants changed their avatars' clothing and features from time to time as just mentioned. The purpose of those changes were investigated through interview questions. Results indicated that most of the participants (n = 4) in this case changed their avatars in order to differentiate from those of the other students. Namely, they would wanted to be unique and wanted others to thereby recognize their avatar. Some of them (n = 3) changed their avatars' feature and clothing in order to simulate their avatars to resemble themselves. CS1-STUDENT7 wanted her avatar to resemble herself by "inserting a copy of her portrait to the avatar's face." The last reason for changing avatars was that two students disliked the clothing of the default avatar, and therefore they changed their avatar's clothing. Of these two female students, one of them argued of her avatar's default clothing that, "her clothing was awful."

Most- and least-liked things about avatars

The students' most-liked and least-liked things about their avatars were investigated via the interview forms. Emerging categories were the humanoid features of avatars, movement of avatars in the 3D environment, and having different options to customize the avatars. Of the total students, five of them argued that they liked the movement of the avatar most. Students could move in the 3D environment by running, flying, or teleporting. They mentioned that they liked all three types of movement of the avatar in different styles. The other most-liked aspect about avatars was the humanoid features. Four students stated that they liked the things about the avatars that made them look human-like, such as avatar's sitting, touching, taking something to its hand, or typing like a human. The last thing stated by one student as the most-liked feature was being able to dress the avatar as they desired from among the many options.

The least-liked things about the avatar were also investigated. The emerging categories were style and the slow movement of the avatar. Two students complained that avatars were too slow while walking somewhere in the 3D environment. Other disliked aspects were the walking style of avatars. CS1-STUDENT2 stated that she disliked the walking style of her avatar;

> *CS1-STUDENT2- You touch something, the avatar draws circles at that moment while walking. I don't like this.*
>
> *CS1-STUDENT2- Böyle bir şeye dokunuyorsunuz, orada böyle yuvarlak yuvarlak gidiyor hemen. Bu hoşuma gitmiyor.*

Suggestions about avatars

Students in this case suggested two things about their avatar. The most mentioned suggestion was that avatars could have supernatural power such as the power of Hulk. For example, CS1-STUDENT3 suggested about this as;

> *CS1-STUDENT3- Sir, I think it would be better if there were super modes on avatars. If they were not only like normal people. For example, if [only] they had the power of Hulk or the speed of Flash.*
>
> *CS1-STUDENT3- Hocam, bence şey olsaydı daha güzel olurdu, böyle süper modlar olsaydı avatarda. Tek böyle normal insan gibi olmasaydı. Mesela Hulk'un gücü filan olsaydı. Flash'ın hızlılığı olsaydı.*

Another suggestion related to avatars by a student was that avatars could do routine things just the same as humans in everyday life such as cooking, eating, holding something, or laying on a beach. CS1-STUDENT2 expressed her suggestion as avatars could do routine tasks such as “cooking, or setting the table.”

4.5.2 Case-2: Extra-Curricular Program

Avatar Representation and Changes in Avatar

It was asked whether the students felt that they were well-represented by their avatars in the 3D environment. Most of the students in this case felt that their avatars were their 3D representation in the virtual environment. As a foundation for this feeling, some of them mentioned about being able to customize their avatars. In this way, their avatars were differentiated to others in the environment and for some, they even carried some of their own features. For example, one of the students stated;

> *INTERVIEWER- And do you think your avatar really represents you in the virtual world?*
>
> *CS2-STUDENT2- Yes. It both represents me and I have attributes, I have some attributes and such, so I think it represents me.*
>
> *INTERVIEWER- Peki avatarının seni gerçekten sanal dünyada temsil ettiğini düşünüyor musun?*
>
> *CS2-STUDENT2- Evet. Hem temsil ediyor, hem de benim özelliklerim var, birkaç özelliğim falan vardı, o yüzden bence beni temsil ettiğini düşünüyorum.*

In contrast to these students, two other students argued that they did not feel they were represented by their avatars. For example, one argued that he did not think his avatar represented himself in the virtual environment due to his avatar’s appearance as his own appearance and his avatar’s were quite different. He explained his feeling about his avatar as, “I do not think that it represents me. Because my appearance is not as comical as my avatar’s appearance.”

There were some changes made to the avatars in this case. Students reported that they mostly changed their avatar’s clothing, body type, hair, skin color and accessories of their avatars. While some students were changing their avatars a great deal, a few of the other students (n = 2) ignored the appearance of their avatars. They argued that they changed the appearance of their avatar at the beginning, and then never changed them again. CS2-STUDENT3 said that he ignored the appearance of his avatar;

> *CS2-STUDENT3- I did not care how it looked at all. Mine was bald… Some of them has made really very different things. When it comes to appearance, they add some hairstyles, I did not care at all. I just chose one, that is all.*
>
> *CS2-STUDENT3- Ben hiç görünüşünü umursamadım. Keldi benimki zaten… Bazıları baya baya değişik şeyler yaptı. Görünüm deyince böyle saç ekliyorlar, ben hiç umursamadım. Bir tanesini seçtim o kadar.*

Purpose of customization

The purpose of customization was investigated via the interview questions. The results indicated four reasons for the changes made to the avatars. The most-cited reasons argued by the six students were to simulate avatars that resembled themselves and their dislike of the initial clothing of their avatar. The students in this case wanted to simulate their avatars to resemble themselves as much as possible. One of them mentioned this as, "I tried to simulate my avatar to look like me; however, I couldn't achieve it all that much." Another stated about the clothing as, "I didn't like the clothing, and changed it to some nicer clothing."

The other reason for changing avatars stated by three students was for the purposes of differentiation. Students would like their avatars to be recognized in the 3D environment from those of other students. Thus, they customized their avatars. For example, CS2-STUDENT2 argued that he changed his avatar in order for his avatar to not be confused with others' avatars;

> *CS2-STUDENT2- So that it would not be confused with that of anybody else, or my friend's.*
>
> *CS2-STUDENT2- Hem böyle örneğin başkalarıyla karışmaması için, [veya] arkadaşımla karışmaması için.*

The last reason stated by two students was to look funny. Students wanted to attract their friends' attention sometimes, and for this they tried to make the shape and feature of their avatars more funny.

Most- and least-liked things about avatars

Students' most- and least-liked things about their avatars were investigated. The most liked things were having different options, movement, humanoid features and building. Most of the students (n = 8) in this case stated that what they liked the most was having many options to customize their avatars. They really liked to choose the best fitting thing for their avatar among the many options available to them. For example, while one of the students mentioned that it was possible to change the appearance as they desired, and another stated that it was possible to choose one item of clothing among many options and thereby arrange the features of their avatars;

> *CS2-STUDENT5- You can give whatever clothes and whatever look you want to give.*
>
> *CS2-STUDENT7- The features that I like on my avatar, you can wear lots of clothes, you can grow in height, and you can add facial effects, I like these.*
>
> *CS2-STUDENT5- İstediğin kıyafeti [ve] istediğin görünüşü verebiliyorsun.*
>
> *CS2-STUDENT7- Avatarımın beğendiğim özellikleri, şimdi bir sürü kıyafet giyebiliyordunuz, boy uzatabiliyorsunuz, ondan sonra yüz efekti ekleyebiliyorsunuz, hoşuma gitti bunlar.*

The other most-liked feature stated by the six students was the movement of their avatar in different modes in the 3D environment such as flying or teleporting. Another most-liked aspect stated by three of the students was the humanoid features of the avatars. Students stated that they appreciated the human-like features of their avatars such as being able to take an object in their hand, sitting down, typing and so on.

The least-liked things about the avatar were style, the slow movement of the avatar, and not being able to customize the avatar as much as desired. Four of the students stated that they least-liked the style of the avatar. For example, one of the students described the face of the avatar as horrible upon closer inspection. Inability to customize the avatar as desired was another dislike stated by three of the students. Although there were many options for customizing the avatars, some students would like to wear other things apart from those offered to them such as glasses or even a superman cloak. One of the students mentioned this as;

> *INTERVIEWER- Are there any features that you do not like on your avatar?*
>
> *CS2-STUDENT8- When you click on the appearance part, sometimes not everything is on there. We could not set it the way we liked it. [Once] we wanted to add glasses [on the avatar] for example, but I couldn't find it.*
>
> *INTERVIEWER- Beğenmediğin özelliği var mı avatarının?*
>
> *CS2-STUDENT8- Görünüme girdiğimizde her şeyi olmuyor her zaman. Istediğimiz gibi tamamen ayarlayamıyorduk. Mesela orada diyelim ki gözlük ekleyeceğiz, onu mesela ben bulamamıştım.*

The last disliked thing about avatars were their slow movement. Two students complained about this issue.

Suggestions about avatars

There were three suggestions related to avatars in this case. The most-cited suggestion was about the avatar's ability to do routine tasks similar to real life. Half of the students in this case suggested that their avatar could do things similar to real life, such as bending and holding on to something, sleeping and so on. CS2-STUDENT9 highlighted his suggestion as, "It would be better if the avatar were able to do routine tasks of daily life such as sleeping, drinking, and eating." Another suggestion from two students in this case differed to Case-1, which was limiting the changes in avatar to some extent and punishing those who did something wrong. CS2-STUDENT5 suggested that avatar customization should be limited and in this way, the virtual environment could be safer;

> *CS2-STUDENT5- But also you could have created a safer environment. For example some [of the students] would make their character chubbier etc., I think you could limit them.*
>
> *CS2-STUDENT5- Ama bir de daha güvenlikli bir ortam yapabilirdiniz. Mesela bazıları karakterini daha tombul yapıyordu falan onları bence sınırlayabilirdinizi.*

The last suggestion was about the different modes of movement for the avatars. One student suggested that there could be different speed modes while navigating through the environment such as slow, moderate and fast options.

4.5.3 Case-3: After-School Program

Avatar Representation and Changes in Avatar

Students were asked whether they felt that they were well-represented by their avatars in the 3D VW environment. Most of the students in this case reported that they felt this way. Some of the students mentioned that they felt like this since their avatars carried some of their features. CS3-STUDENT4 expressed his feeling as, "I felt like that, because it is so similar to me, I tried to make it resemble me." In this case, there was only one student who argued an opposing view.

Changes made by the students to their avatars were also investigated. Some changed some of the features and clothing of their avatars. The results indicated that they mostly changed their clothing, hair color and style and body type. For example, one of them changed the shape of his nose. Another students said, "Clothing, body type and length of skeleton – I changed them several times."

Purpose of customization

The reasons for making changes in their avatar were investigated in this case as well. The results revealed three reason types for making changes. Half of the students (n = 3) changed their avatar either to resemble themselves or for the purposes of differentiation. They wanted for their avatars to have some special features similar to themselves such as the same hairstyle or same hair color. They also customized their avatars in order to be unique and therefore to be known by others. One of the students mentioned that he changed his avatar for simulating himself and for his avatar to be recognized among others;

> *CS3-STUDENT4- I've designed my avatar specially for myself. I did it so that it would resemble me and I would not confuse it [with others].*
>
> *CS3-STUDENT4- Avatarımın kendime özel, kendim için özel olarak bana göre olması için ayarladım. Bana benzemesi için, karıştırmamam için ayarladım.*

The last reason stated by one student was to look funny. CS3-STUDENT2 expressed that "We changed our avatars to look funny; we were changing their shape to look fat, and simulating their eyes to the eyes of Spiderman."

Most- and least-liked things about avatars

Students' most-liked and least-liked things were investigated for this case, as well. The emerging categories for the most-liked things were having different options to customize and the movement of avatars. Most of the students in this case (n = 5) argued that they liked having so many clothing options for their avatars the most. Another most-liked feature stated by two of the students was the movement of the avatar in its different modes.

Emerging categories for the least-liked things about avatars were being unable to customize it as desired, and the slow movement and style of the avatars. It was difficult for a few of the students to change the avatar and its clothing. For example, one of them mentioned that the avatar's hair color could change when something wrong was done in the program. Slow movement and the appearance of looking cross-eyed were the other least-liked things stated by one of the students.

Suggestions about avatars

There were two suggestions made about the avatar from this case. The first suggestion was that avatars could be made to perform routine tasks. One student suggested that it should be possible for an avatar to sleep and wake up, or sing a song. The other suggestion stated by one student was that there could be more clothing options for dressing avatars.

4.5.4 Cross-Case Analysis

Cross–case analysis of the three cases revealed some differences and similarities among the cases. The emerging sub-themes and codes in regards to avatar issues, and their frequencies are presented in Table 4.20, and elaborated on in this section.

First emerging sub-theme was about representation by an avatar and the changes that could be made to avatars. According to the results, most of the students in each case felt that they were represented by their avatars in the 3D VW environment. However, a few students from Case-2 and Case-3 argued the opposite to this feeling. The argument of those students were similar to those who felt that they were represented with avatars. Students argued that customizing appearance, differentiation from other avatars in 3D environment, and the fact that avatars carried some of the features of the students were the main factors behind this feeling. The features changed for avatars by the students were similar to some extent across the cases. Students from all three cases changed the clothing and hair of their avatars. However, only a few of the students of Case-2 and Case-3 changed the avatar's body type, while only one student in Case-1 and Case-2 changed the skin color and accessories.

Table 4.20 – *Frequencies of Avatar Issues across the Cases*

	Case-1		**Case-2**		**Case-3**	
	n	***%***	***n***	***%***	***n***	***%***
Purpose of Customization						
Simulating to resemble themselves	3	43	6	60	3	50
Differentiating from others	4	57	3	30	3	50
Disliking avatar clothing	2	29	6	60	-	-
To look funny		-	2	20	1	17
Most-Liked Things about Avatar						
Different options to customize	1	14	8	80	5	83
Movement (running, flying, teleporting)	5	71	6	60	2	33
Humanoid features	4	57	3	30	-	-
Least-Liked Things about Avatar						
Style / appearance	2	29	4	40	1	17
Slow movement	2	29	2	20	1	17
Inability to customize as desired	-	-	3	30	2	33

The purpose of customization of avatars was the other issue investigated. The mutual reasons for changes to avatars across the three cases were to simulate avatars in order to resemble themselves, and for the purposes of differentiation from other avatars. The only reason not mentioned by the students of Case-1 was to be funny. This might be due to the reason that the settings in Case-1 were more formal than the other two cases. Although the students of Case-3 changed their avatars' clothing due to a dislike of the appearance, it was not mentioned as a reason to apply changes to the avatar.

The most- and least-liked things about avatars were the other issue investigated. Emerging categories were mostly similar among the cases. Movement of the avatars in different modes and having different options to customize an avatar were stated by the students of all three cases as the most-liked thing about them. Only the humanoid features of avatars were not stated by the students of Case-3, although it was stated by the students of both Case-1 and Case-2 as the most-liked aspect. It seems that students in Case-3 did not care about the human-like features of the avatars. Style and the slow movement of avatars were stated by the students of all three cases as the least-liked aspect. Some students of both Case-2 and Case-3 reported that they disliked not being able to customize their avatars as much as they desired. No students from Case-1 stated the same issue though. Students in both Case-2 and Case-3 would like to have had more options for customizing their avatars than the students of Case-1. In addition, the students of Case-1 seemed to be glad of the options to customize their avatars.

Students' suggestions about the avatars were similar in terms of the avatar's ability to perform routine tasks that were similar to real life. Half of the students suggested this issue in Case-2, while only one student suggested this in the other cases. Different from both Case-1 and Case-3, the students of Case-2 suggested limiting changes in avatars and having multiple movement modes (i.e. slow, fast) for avatars. On the other hand, only the students of Case-1 suggested that avatars could have additional supernatural powers, such as the power of the Hulk or the speed of the Phantom. The only suggestion made by students in Case-3 that were different from the other cases was more extensive clothing options for avatars. Some students, especially females, cared about the appearance of their avatars more than the others did.

4.6 Sub RQ – Group Issues and Strategies

Studying with peers is an important issue in programming education for children (Resnick & Siegel, 2016). In the current study, the students were paired with a friend who was generally sitting next to them. Pairs worked in separate regions in the virtual world and needed to complete their own assigned tasks. Students were encouraged to study in groups by using the opportunities of both the virtual and the physical environment. Only one student in Case-2 completed the tasks fully alone. In this section, the students' preferences to study with a pair or alone and their ideas related to the number of members in groups are addressed. Then, the forms of help available between group members and the rapport and problems among group members is elaborated on. Lastly, the issue of how to select group members and the similarity of tasks between group members is addressed in the last part of this section.

4.6.1 Case-1: Curricular Program

Group Study

Students were paired up in order to study with and receive help from each other. The preferences of the students and the reasons behind their preference were sought as to whether they preferred to study alone or with a peer. It was also investigated about the potential number of members in each group. In this case, there was only one student who preferred to study alone rather than in a group. This student argued that studying in a group was useless and he would prefer not to get help from anyone;

> *INTERVIEWER- Why do you want to work alone?*
>
> *CS1-STUDENT3- Not because I do not like my teammate, I like doing it without getting help, working on things on my own more. And it is not any better when we have a teammate.*
>
> *INTERVIEWER- Sen neden yalnız çalışmak istiyorsun?*
>
> *CS1-STUDENT3- Takım arkadaşımı sevmediğimden değil, ben böyle yardım almadan yapmayı, kendim uğraşmayı daha çok seviyorum. Bir de takım arkadaşımız olunca bir işe yaramıyor ki.*

Other students (n = 5) generally preferred to study within a group. However, one student preferred to study both alone and in a group, arguing that this situation never bothered him. Similar to this student's idea, the teacher of this case argued that both were helpful according to a student's own preference. It should be the students' choice to study in either a group or alone. She argued that students who preferred to study alone could be negatively affected when forced to study in groups. On the contrary, she thought that the performance of students who preferred studying in a group increased;

> *CS1-TEACHER- Both of them have been beneficial… There were the ones who wanted to be a group and who did not. I think it has had a negative effect on the ones who did not want to be a group when they were made one. But still they completed their tasks in the virtual world… I believe the students who wanted to make a group have given a better performance in virtual environment.*

CS1-TEACHER- İkisi de faydalı oldu… Şimdi grup olmak isteyenler vardı, istemeyenler vardı. İstemeyenlerin grup olması onları olumsuz etkiledi diye düşünüyorum. Ama sanal ortamda yine görevleri yerine getirdiler… Grup olmak isteyen öğrenciler de sanal ortamda daha iyi bir performans sergiledi diye düşünüyorum.

The reasons behind students' preference for studying in a group were that studying in a group were more enjoyable and cooperative. Students received help on the 3D environment from each other, and it was easier to therefore perform the tasks. More information about the forms of help is presented in the next part.

There were two students in each group in the current study. Students' preferences about the number of group members were investigated. Of the total students in this case, four of them stated that it would be more than two such as three, four, or six. In contrast to the students, the teacher of this case stated that two-person groups were the most appropriate because it would be hard to draw attention of students to the lesson when there were more than two students working in the virtual environment;

INTERVIEWER- What is your opinion on the number of students in groups?

CS1-TEACHER- I think 2 [students in a group] is fine. When there are more than that, I feel that the kids get distracted. I mean that when there are 3 people, it goes as 'I have hit you,' or they become more focused on wandering on the environment or writing and sending messages to each other. They start doing stuff apart from the task. But if there were 2 people [in the group], I think that they would directly complete the tasks.

INTERVIEWER- Gruptaki kişi sayısı konusunda ne düşünüyorsunuz?

CS1-TEACHER- Bence 2 iyi yani. Daha fazla olursa çocukların dikkati dağılıyor gibi düşünüyorum. Yani 3 kişi olsa mesela işte sana çarptım gibi, yani direkt ortamda gezinme odaklı oluyor ya da ona işte yazma, mesaj yazma farklı oluyor. Görev dışı faaliyetler oluyor. Ama 2 kişi olsa direkt görevi yerine getirme olur diye düşünüyorum.

Forms of Help

Group members helped each other in many ways. Forms of help between teammates were investigated via the interview form. Results revealed that they could be categorized in three ways. The first form was to help each other related to tasks. Four of the students stated that they received help from their teammates in terms of how to perform tasks in the environment. For example, one of the students helped his teammate with how to take cars since his friend was unable to do that, and CS1-STUDENT6 described the form of help in this category as, "Your friend helps you when you could not do the task."

Another category was to help each other related to issues with the 3D environment. Two of the students argued that they received help from their teammate with such issues. While one of them helped his friend with how to take an object to his inventory, another student helped her friend since she was confused with the direction of objects in the 3D environment.

The last sub-theme was to help each other in terms of how to create code in S4OS. Some of the students had difficulty in creating code in S4OS, and their teammates were helping them with this. One student argued that he helped his teammate in terms of how to create code on S4OS and showed the code he had built in helping his teammate;

> *CS1-STUDENT1- When my friend could not do something, if I could do it, I would show him/her how to do it and the code. We would do it according to the task he/she was supposed to complete.*
>
> *CS1-STUDENT1- Arkadaşım bir şeyi yapamıyordu, nasıl yapacağını, yapamadığı yerlerde bazı yerleri ben yapabildiysem o kodları gösteriyordum. Onun yapacağı göreve göre yapıyorduk.*

Rapport and Teammate Problems

Having a good rapport with teammate is a general requirement when working on group studies. Students were asked whether or not they had rapport with their teammate. The problems they had were also investigated via the interview questions. Of the total students, only four reported on this issue. The results indicated that three students had good rapport with their teammate, while only one student reported that he had a rapport with his teammate other than when doing a task. Problematic issues were the potential to damage a peer's task in which they used the same names in their code for special situations such as broadcasting names. This caused interference for the students' code. CS1-STUDENT3 explained this problematic issue as, "When I touch my own button, my friend's box was moving."

The major problem stated by four students was about completing tasks in a different duration. While some students were completing the tasks fast, others were completing them much slower. This was generally a problem when students studying in different speed modes were paired. For example, CS1-STUDENT7 explained the problem she had with her teammate as;

> *CS1-STUDENT7- I have experienced [the same thing] with İrem. ... because she was supposed to wait for me as a teammate. Otherwise we would not be able to complete any tasks. She was doing it a bit faster. But I was doing it slowly so that I could understand it, ... She was going too fast.*
>
> *CS1-STUDENT7- Ben de İrem'le yaşamıştım. ... Çünkü beni beklemesi gerekiyordu takım arkadaşı olarak. Yoksa hiçbir görevi bitiremezdik. O biraz hızlı yapıyordu. Ama ben yavaş yapıyordum ki anlayım, ... O çok hızlı gidiyordu.*

Pair Assignment

Defining group members is an important strategy for the effectiveness of a group study and for the members to study with a good level of rapport. Participants' ideas were sought via the interview questions on this issue. The results indicated two categories of preference for this sub-theme; according to gender and students' wishes. According to students' wishes was the most-cited preference. Of the total students, two stated that they could define their own group members according to their wishes. CS1-STUDENT3 commented on this as, "It would be better to be matched with the one I want." Another strategy mentioned by one of the students was that it could be according to gender of group members, namely that girls should be grouped with girls and boys should be grouped with boys.

Similarity of Tasks between Group Members

There were 24 tasks completed by groups working together. Although the four tasks were exactly the same, others were differed at some point. For example, in one of the tasks, while one of the peers was programming an object to make a seven-stepped ladder, the other teammate was programming an object to make a ten-stepped ladder. Students' ideas were sought for this issue on whether or not the tasks should be completely the same or different in some way. The participants of this case thought that the tasks could be same and different to the same ratio. While half of the students

(n = 3) stated that tasks should be same for teammates, the others (n = 3) stated that they should be different for teammates. For example, CS1-STUDENT6 commented that, “They should be the same because my teammate will help me when I can’t do that task.” Another participant stated that, “Tasks should be different. If they are the same, during doing my task, I could see the code of my friend while he was doing it; perhaps I will try to cheat from him. However, when they were different, we will each try to do them at the same time.” One of the participants stated another important issue; CS1-STUDENT7 thought that they should be same at some points, but should be different at some points;

> *CS1-STUDENT7- I think they should have been same when it was the right time, and sometimes different, I mean we could do the different [tasks] during the easy parts, and same tasks in the difficult parts.*
>
> *CS1-STUDENT7- Yeri geldiğinde aynı olması gerekiyordu bence, yeri geldiğinde de farklı olsaydı, yani kolay yerlerde farklı yapsaydık, zor yerlerde aynı olsaydı.*

4.6.2 Case-2: Extra-Curricular Program

Group Study

Only one student in this case studied alone, rather than in a pair. When his preference was sought, he argued that he would simply prefer to study alone. The rationale behind his preference was that dealing with others could be difficult at times if he studied with a peer such as some with undesirable behaviors. He also added that completing tasks was faster when he was alone;

> *INTERVIEWER- Were you always alone?*
>
> *CS2-STUDENT8- Uh-huh.*
>
> *INTERVIEWER- So I will ask my question like this; would you prefer working as a team or working alone?*
>
> *CS2-STUDENT8- Alone, because in teams something unexpected can happen, we may not get along with others and for example we can go faster when we are alone. When you are waiting for your teammate to do [something], it might take you too long to go on with the next task. We can be faster when we are alone.*
>
> *INTERVIEWER- Sen yalnız mıydın hep?*

CS2-STUDENT8- Hı-hı.

INTERVIEWER- Peki sorum şöyle soruyorum; takımla mı çalışmayı tercih ederdin, yoksa yalnız olarak mı çalışmayı mı?

CS2-STUDENT8- Yalnız. Çünkü bazen takımda istemeyeceğiniz şeyler olabiliyor, başkalarıyla anlaşamayabiliyoruz ve mesela tek kişi olduğunuzda biraz daha hızlı gidebiliriz. Diğer takım arkadaşımızın yapmasını beklerken diğer göreve geçmeniz uzun sürebilir. Tek kişi olunca daha hızlı yapabiliyoruz.

Only one student mentioned that he would prefer both situations. She stated that both situations were good for herself, whilst most of the students ($n = 8$) generally preferred to study in groups. They discussed things with each other in order to complete the tasks and other topics. One of them learned how to be more social with his friends thanks to VWs. Students understood tasks better and completed the tasks easier when studying in a group. Moreover, studying in a group enabled the students to learn from their peers. The teacher of this case also emphasized the importance of group study since it was easier for students to learn from each other;

INTERVIEWER- Do you think it is better to be a team or should it be individual?

CS2-TEACHER- It is good that we are [working as] teams because I believe peer learning is significant especially during the usage of information technology tools. It is easier for them to learn this sort of things from each other. That is, sometimes they feel shy about asking the teacher about something or the teacher may not reach everyone, 24 people in crowded groups like ours, at the same time. In these cases group work helps us a lot.

INTERVIEWER- Sizce takım olması iyi mi, yoksa bireysel mi olmalı?

CS2-TEACHER- Takım olması iyi.. Çünkü akran öğretimi bence özellikle bilişim teknoloji araçlarının kullanımı sırasında akran öğretiminin önemli olduğunu düşünüyorum. Birbirlerinden öğrenmeleri daha kolay böyle şeyleri. Yani, bazen öğretmene sormak konusunda çekiniyorlar ya da öğretmen bizim gibi kalabalık gruplarda 24 kişiye aynı anda işte ulaşamayabiliyor. Böyle durumlarda grup çalışmasının çok faydasını görüyoruz.

Possible number of members in a group was asked to the students. While six students and the teacher of this case stated that the number of students in groups would ideally be two, two other students stated it should be more than two. The ones who preferred two-membered groups defended their decision saying more than two would be too crowded and chaotic, more difficult to control and communicate with each group member. Those who stated it should be more than two could not define a logical reason to their decision. For example, CS2-STUDENT2 commented on the two-membered group preference as;

> *INTERVIEWER- You were 2 people [students] in the group. What do you think about this?*
>
> *CS2-STUDENT2- I think 2 was the best of all because it was easier to communicate with him/her. If there were 3 people, you would be supposed to talk to both of the people and that would be a bit difficult. So I think 2 people were fine.*
>
> *INTERVIEWER- Peki grupta 2 kişiydiniz. Bu konuda ne diyorsun?*
>
> *CS2-STUDENT2- Bence 2 en iyisiydi. Çünkü onunla daha kolay iletişim oldu, 3 kişi olsa hem ona, hem ona anlatmak zorunda olurdun, o biraz zor olurdu. O yüzden 2 kişi bence iyiydi.*

Forms of Help

Forms of help in this case could be categorized under three sub-themes; help related to tasks, help related to the 3D environment, and help related to S4OS. The first was about helping each other related to tasks. Most of the participants (n = 7) argued that they helped each other with issues. For example, CS2-STUDENT1 declared that she received help from her teammate on how to complete tasks since she was absent during the first week of the club;

> *CS2-STUDENT1- I was a bit late in the beginning. So I did not understand anything about the first task. Then teammate helped me and because he said 'You are going to click on this box and do this and that,' I started not to have problems after that.*
>
> *CS2-STUDENT1- Ben ilk başta biraz geç kalmıştım. O yüzden 1. görevden hiçbir şey anlamadım. Sonra takım arkadaşım bana yardım etti, şu kutuya tıklayıp işte şunları yapacaksın falan dediği için sonra zorlanmamaya başladım.*

Other emerging sub-themes in this case was help related to the 3D environment itself. Half of the students (n = 5) argued that they received help with such issues. Help was generally about getting objects, opening help videos, bringing the teammate to the right location and finding the location of tasks. For example, CS2-STUDENT1 helped her friend by teleporting him next to her when he was lost in the 3D environment;

> *CS2-STUDENT1- For example once, my friend had got lost in the ocean. When I told him/her 'Come towards me' by clicking on his/her avatar, S/he came towards me.*
>
> *CS2-STUDENT1- Mesela arkadaşım bir keresinde okyanusun içinde kaybolmuştu. Ben onun avatarının üstüne tıklayarak benim yanıma gel falan dediğim için benim yanıma gelmişti.*

The last sub-theme was to help peers with their code and S4OS. Of the total participants, three of them argued that they helped each other with these issues. They were generally showing each other code blocks, the place of the code on S4OS and giving ideas about the code. CS2-STUDENT5 highlighted this form of help as, "When I could not find the accurate code on S4OS, my friend showed me the code and suggested where to place the code on some occasions."

Rapport and teammate problems

Most of the students (n = 9) reported that they had good rapport with their teammates throughout the study. Only one student argued that he did not have good rapport with his teammate sometimes. CS2-STUDENT4 highlighted the problem with his teammate as;

> *INTERVIEWER- Did you work with your friends in a harmonious way, did you have problems?*
>
> *CS2-STUDENT4- I worked in an harmonious way. I did not have many problems. But sometimes s/he made problems for me.*
>
> *INTERVIEWER- For example?*
>
> *CS2-STUDENT4- For example during the task s/he clicked on my item by accident, we programmed them at the same time, so it became a problem for his/her own item.*
>
> *INTERVIEWER- Arkadaşınızla uyumlu çalışabildiniz mi, sorun yaşadınız mı?*
>
> *CS2-STUDENT4- Ben çalıştım. Çok sorun yaşamadım. Ama bazen onun bana sorun yaşattığı oldu.*
>
> *INTERVIEWER- Ne gibi?*

> *CS2-STUDENT4- Mesela görevi yaparken o da yanlışlıkla benim malzememe tıkladı, aynı anda programladık kendi malzemesi yerine sorun oldu o da.*

Another potential problem stated by one student was the possibility of causing problems when working in a group. The student had worked alone throughout the study in the 3D environment. Surprisingly, he thought that there would be the potential for disagreements between members of the group in certain situations; adding that it would be possible to complete the tasks at your own pace without waiting for the friend to complete;

> *CS2-STUDENT8- Sometimes things that you do not want to happen can happen in teams, for example we might not get along with others, we say things such as 'We will do this there, we will do that there' or for instance one says 'You should turn right,' the other one says 'You should turn left.' If we cannot agree on that part, there might be a problem. And for instance when you are working alone, you can go a bit faster. It might take you long to move on to the next task. We can do it faster when we work individually.*

> *CS2-STUDENT8- bazen takımda istemeyeceğiniz şeyler olabiliyor, başkalarıyla anlaşamayabiliyoruz mesela orada şöyle yapacağız, böyle yapacağız, mesela diğeri sağa dönmelisin, diğeri de sola dönmelisin, orada anlaşamazsak belki sorun olabilir. Bir de mesela tek kişi olduğunuzda biraz daha hızlı gidebiliriz. Diğer takım arkadaşımızın yapmasını beklerken diğer göreve geçmeniz uzun sürebilir. Tek kişi olunca daha hızlı yapabiliyoruz.*

Pair Assignment

Participants of this case explained their ideas about the preferences of choosing group members. Emerging categories for this sub-theme were that students would like to be grouped according to their wishes, or gender. Of the total participants, most of them (n = 7) argued that defining group members should be according to the group member's wishes. They usually preferred to be grouped with someone they either recognized or would like to be with. For example, CS2-STUDENT2 commented about his teammate as, "Being able to choose my teammate was better for me because he was both my best friend and sat next to me."

Another important strategy about pair assignment stated by one of the female students was that team members could be assigned according to gender. She mentioned that she would like to be grouped with a female although most of the class consisted of males.

The teacher of this case mentioned another important strategy in defining group members. She explained that students should be grouped according to level of self-confidence in computer usage. In this way, groups would consist of one member with high and one member with low confidence with computers; making it possible for one to encourage the other;

> *INTERVIEWER- And how do you think you should determine the members of the groups?*
>
> *CS2-TEACHER- Instead of 2 students that have the same level, we should put 2 students one of whom is a bit better and one of whom is shy about certain things together. I mean I do not call this situation as 'successful / unsuccessful,' in fact that shy kid might be successful as well. But s/he might not have enough self-confidence to show it. So s/he must be matched with a student that can encourage or motivate this kid. In programming, more precisely in front of the computer, it is very important for a kid to feel secure. Because some of the kids are afraid of the machines, they think 'I will do something wrong' or 'If I click on here, this will happen'... I think a braver student must be sitting next to that kid.*
>
> *INTERVIEWER- Peki grup üyelerini belirleme nasıl olmalı sizce?*
>
> *CS2-TEACHER- Yani grup üyelerini belirleme 2 tane aynı düzeyi bilen yerine, biraz daha iyiyle belli konularda çekingen davranmayı tercih eden çocuğu bir araya getirmek şeklinde olmalı. Yani başarılı-başarısız demiyorum buna, aslında o çekingen davranan çocuk da başarılı olabilir belki. Ama onu göstermek konusunda çok yeteri güveni yoktur kendine. O yüzden onu hareketlendirecek, ona güven verecek çocukla eşleştirmek lazım. Programlamada, yani daha doğrusu bilgisayar başında çocuğun kendini güvende hissetmesi çok önemli. Korkuyor çünkü bazı çocuklar makineden, yani yanlış bir şey yapacağım, şimdi buraya basarsam şu olur… Onun yanına cesaretli bir çocuğu oturtmak gerekiyor diye düşünüyorum.*

Similarity of Tasks between Group Members

Most of the participants (*n* = 7) emphasized that the tasks should be different. For example, CS2-STUDENT9 highlighted that he learned other things due to the differences between tasks while helping his friend. In another case, he would perhaps not learn anything new other than what he learned during his own task;

> *CS2-STUDENT9- I think it is better that [they] were different. For example one of our friends could not do it, I did it, then I taught him/her*

[how it should be done] by using my own knowledge. When they are different, I learn a few things, too.

CS2-STUDENT9- Bence farklı olması daha iyiydi. Çünkü mesela işte arkadaşımız yapamıyordu, ben yapmıştım, sonra kendi bilgilerimden yola çıkarak ona öğretiyordum. Farklı olunca ben de öğreniyorum mesela birkaç şey.

In contrast to these students, some (n = 3) stated that tasks should be the same for each person. On this issue, for example, CS2-STUDENT1 highlighted her rationale as, "It would better when the tasks are the same because then we are in the same group." Moreover, two of the participants in this case pointed out an issue related to the tasks of pairs. Although two different users theoretically could code the same object in the 3D environment, it was not possible to do that practically. CS2-STUDENT5 pointed out this issue as he would like to code the same object with his peer rather than programming separate objects;

CS2-STUDENT5- When there are 2 people [in the team] there is no problem, [but] they both should code the same object together because when someone puts something on there, only the owner can code it. For example only the people that the owner lets should be able to code. For instance I looked on there, there were something like the owner was permitting however it was very complicated… I could not understand it.

CS2-STUDENT5- 2 kişi sorun olmuyor da, ikisi de aynı objeyi kodlayabilsin Çünkü birisi bir şey koyduğunda onu sadece sahibi olan kişi kodlayabiliyor. Mesela herkes kodlayabilsin ama, sahibinin izin verebildiği kişiler. Orada ben baktım mesela, böyle sahibinin izin verme bir şeyleri vardı da böyle çok karışıktı… Ben onu anlayamadım.

4.6.3 Case-3: After-School Program

Group Study

The ideas of students in this case about group study were investigated, as well. In this case, only one student would prefer to study alone. CS3-STUDENT5 stated that, "It causes too much of a clash among team members because I generally don't like the ideas of everyone." Most of the students (n = 5) would preferred to study along with their peers. The students thought that it was easier and more enjoyable when they studied with their peers. For example, CS3-STUDENT2 argued that he completed the tasks with his teammate both in an easier and more fun way.

Their preferences about the number of group members were investigated. Of the total students, three stated that the number of group members should be two. These students listed the reasons of their choice as it could be confusing and more chaotic when the number was exceeded two. Only two of the students stated that it should be more than two.

Forms of Help

The emerging sub-themes of this case were similar to those that emerged in the other cases, but with different frequencies. The first form of help was about the tasks, as stated by two of the participants who helped each other for most of the tasks. The second form of help was about the issues associated with the 3D environment, which was stated by two of the participants. Help related to the 3D environment was about the use of function in the environment such as how to teleport and how to place objects.

The last form of help was with regard to the code and S4OS, which was stated by two of the students, and focused on how to build code blocks on S4OS, and how to configure a code block. For example, CS3-STUDENT4 received help from his teammate on how to build code blocks since he had difficulty building them himself;

> *CS3-STUDENT4- S/he helped me with programming on Scratch. For example s/he helped me put the boxes in the right places. I could not do some of them but I was able to, when s/he helped me.*
>
> *CS3-STUDENT4- O Scratch'te programlama işine yardım etti. Mesela o kutucukları doğru yere koymama yardım etti. Bazılarını yapamıyordum ama, o yardım edince yaptım.*

Rapport and teammate problems

Most of the students ($n = 4$) reported that they had good rapport with their teammates. Although the students of this case did not know each other before the course, it did not take them much time to build up a good rapport with each other. CS3-STUDENT1 explained this situation as;

> *INTERVIEWER- And, did you get on well with your teammate?*
>
> *CS3-STUDENT1- I tried to get to know him/her at first, and then s/he made a little joke to me, then we became very good friends.*

INTERVIEWER- Peki takım arkadaşınla anlaşabildin mi?

CS3-STUDENT1- En başta onu tanımak için uğraştım da, sonradan küçük bir şaka yaptı bana, sonra acayip iyi arkadaş olduk.

Two of the students reported that they did not have a good rapport with their teammate in some situations. They experienced some problems between teammates such as damaging the completed task of their peer or distracting the teammate sometimes so that they could not focus on their tasks. Among them, distracting the peer was seen as the most problematic. Drawing students' attention toward the lesson was hard sometimes due to such kinds of student behaviors. Other problems stated by two of the students was damaging things by mistake. For example, CS3-STUDENT6 addressed this problem as, "She damaged what I had already done while she was trying to help me."

Pair Assignment

Issues and strategies stated by the students in this case about defining group members were similar to those in the other two cases. Half of the participants (n = 3) stated that they could be grouped according to their wishes. One of the participants stated on this issue that he would prefer to be grouped with someone of his choice in order to have a better rapport with them and therefore have better cooperation. He argued that it would be possible to experience problems with an unwanted team member;

INTERVIEWER- Would you like to determine your teammate by yourself?

CS3-STUDENT2- Yes I would; because if you do not determine the [teammate], maybe s/he would not like his/her friend, and we could have such problems since [one of the parties] is not a sharing person. However if we choose the [teammates], they might get along better in the virtual world and they can work together comfortably; something like that could happen.

INTERVIEWER- Takım arkadaşını kendin belirlemek ister miydin?

CS3-STUDENT2- Evet, belirlemek isterdim. Çünkü zaten siz belirlemezseniz şöyle bir şey olur; belki sevmez arkadaşını, sonra belki böyle paylaşımcı olmadığınız için belki böyle sıkıntılar yaşayabiliriz. Ama biz seçersek belki böyle iyi anlaşabilirler sanal dünyada, rahat rahat gidebilirler beraber çalışarak, böyle olabilir.

Another important strategy stated by the two participants was that defining group members should be done according to gender. Although it was stated by one female and one male student, their argumentation regarding this issue were similar. They stated that they had a good rapport with their friends of the same gender. The last issue, that was stated by one of the participants, was that defining group members should be done according to the level of the students. He argued that it could be done according to age, grade level or point of interest.

Similarity of Tasks between Group Members

Only three of the students expressed their opinions on this issue. They argued that tasks should be different for various reasons. For example, one of them expressed that a small difference between the tasks of peers added a kind of style to the tasks; however, all tasks should seem to be of the same difficulty level. Another stated that if they were all the same, we could simply cheat from each other since we sit next to each other;

> *CS3-STUDENT5- I think it was good. All of them seemed to have the same level of difficulty. But it adjusted its own style; I mean it created its own style. For example, one was a coffee shop and the other one was a supermarket.*
>
> *CS3-STUDENT1- I think that was good, because if they were exactly the same, we would copy each other's stuff as we sat next to each other, it wouldn't be that good.*
>
> *CS3-STUDENT5- Bence iyiydi. Hepsi aynı zorlukta gibiydi. Ama böyle kendi stilini katıyordu, olayın içine. Yani kendi tarzı oluyordu. Mesela bir tanesi kahve dükkanı, diğeri market gibi.*
>
> *CS3-STUDENT1- Bence iyiydi. Çünkü tıpatıp aynı olsa birbirimizden kopya çekerek yapardık. Bir de ikimiz yan yana oturduğumuz için o güzel olmazdı.*

4.6.4 Cross-Case Analysis

Students were encouraged to study in groups by studying alongside a peer throughout this study. Issues and strategies related to group study were investigated in each case. In this section, the results of the cross-case analysis are presented. Table 4.21 summarizes the frequencies of the group issues and strategies across the three cases of the study.

Table 4.21 – *Frequencies of Group Issues and Strategies across the Cases*

	Case-1		**Case-2**		**Case-3**	
	n	***%***	***n***	***%***	***n***	***%***
Group Study						
Prefer to study alone	1	14	1	10	1	17
Prefer to study alone or in a group	1	14	1	10	-	-
Prefer to study in a group	5	71	8	80	5	83
Two members only	-	-	6	60	3	50
More than two (3 to 6)	4	57	2	20	2	33
Forms of Help among Group Members						
About tasks	4	57	7	70	2	33
About the 3D environment	2	29	5	50	2	33
About code and S4OS	3	43	3	30	2	33
Pair Assignment						
According to wishes	2	29	7	70	3	50
According to gender	1	14	1	10	2	33
According to level	-	-	-	-	1	17

Students' preferences about group study were similar across the cases. Most of the students would prefer to study within some sort of group. Only one student from each case would prefer studying alone. However, there were some small differences about the preferences of the number of group members across the cases. Although most of the students in Case-1 argued that groups should consist of more than two members, most of the students in the other two cases generally preferred two-membered groups. This may be due to the fact that the students of Case-1 knew each other for more than three or four years prior to the course. For example, one of the students in Case-1 argued that all the girls and boys in the class could form two large groups. Emerging forms of help between the peers were also similar across the three cases. However, it seems that students of Case-2 received help from their peers more than others did when the total frequencies are taken into consideration (see Table 4.21). Students mostly received help from their peers on issues relating to tasks, the 3D environment and the code itself.

Rapport with teammates and problems between group members were another issue investigated. Most of the students from all three cases generally had a good rapport, except for in certain situations. Some problems were also encountered between group members. The common problem faced across the three cases was down to damage caused to the teammate's code. Speed of the teammate was the most-cited problem in Case-1, and distracting teammates the most-cited in Case-3.

Pair assignment is another important strategy in group studies. Students' preferences across the cases were the same to some extent. Students from each case mentioned that it could be arranged according to team members' wishes or the gender. No students in Case-1 mentioned that it should be according to level of students. The teacher of Case-2 and one student in Case-3 argued that defining group members could be achieved according to the level of the students. Similarity of tasks between group members was another issue that needed to be taken into consideration. Some students from all three cases preferred that tasks should be different between peers. On the other hand, a few students from both Case-1 and Case-2 argued that tasks should be the same, although no students from Case-3 reported the same.

4.7 Sub RQ – Satisfaction

Satisfaction could affect students' approach to lesson in a positive or negative way. Participants' satisfaction in the current study was measured via the application of a questionnaire. In addition, qualitative data were gathered via interview questions in order to determine the factors affecting student satisfaction. In this section, firstly, the status of satisfaction level is presented based on the descriptive analysis of the quantitative data, and the results are also supported with qualitative data. Then, the factors according to increasing and decreasing satisfaction of the participants are addressed, respectively.

4.7.1 Case-1: Curricular Program

Current Status of Satisfaction

The satisfaction questionnaire was adopted from Chou and Liu (2005), and consisted of four items. Descriptive results of the questionnaire are given in Table 4.22. As can be seen, the students' scores were generally high for each item. The total mean scores

(M = 4.35) were moderately high. This could be interpreted as students having been generally satisfied from the study.

The qualitative results supported the quantitative results. However, in this case, there were some factors of decreasing students' satisfaction scores from the outset of the study since students experienced some problems. Those problems were overcome during the second part of the study. One of the students commented about this, "I did not expect something like this, but I am satisfied at the end. However, I was not satisfied at the beginning."

Table 4.22 – *Satisfaction questionnaire results of Case-1*

	M	***SD***
I was satisfied with SDP learning experience.	4.33	0.78
I think this SDP benefit my learning achievement.	4.17	1.19
I was satisfied with SDP.	4.50	0.52
I was satisfied with the overall learning effectiveness.	4.42	0.51
Total	**4.35**	

Factors increasing satisfaction

Factors causing increase in students' satisfaction level were investigated through interview questions. Emerging factors based on the results were group study, object construction, tasks, off-task activities, 3D environment, story of tasks and touring in the 3D environment. They are elaborated on more as follows.

The most-cited factor increasing the satisfaction of the students in this case was group study. Most of the students (n = 6) argued that being together in groups and doing tasks together was a satisfying factor. Students really liked being a member of a group and to do the tasks together. CS1-STUDENT2 commented on group study as, "Being together was the thing I liked the most."

The other increasing factor stated by five of the students was regarding the tasks. Students mentioned that they liked the tasks that needed to be completed. Another factor was object construction, as stated by two students. The students liked to build 3D objects such as bridges, turtles, and walls of a shelter, and to code them. Other factors stated by one student were off-task activities, the 3D environment and tour of the environment. Students did not like to build objects related to tasks, but also they did like off-task activities. Moreover, traveling in the 3D environment as they wished was seen as a satisfying factor for one student.

Factors decreasing satisfaction

Some factors revealed from the interview responses led to a decrease in the students' satisfaction. Those were technical problems, difficulty of task, avatar-related problems, difficulty of use and rearranging the code. They are elaborated on more as follows.

Students studied alone in the first part of the study, and then studied alongside a teammate in the second part. Studying alone was the most-cited (n = 3) factor decreasing satisfaction of the students. Being alone was considered as boring and undesired for the students. For example, CS1-STUDENT2 expressed her feeling about studying alone as it being rather boring. Other decreasing factor was the technical problems encountered by two of the students. Technical problems were inevitable; however, they need to be limited to a minimal level. In this case, the students experienced some technical difficulties in the first part of the study due to poor computer hardware issues and weaknesses related to the infrastructure. One of the students highlighted that the technical problems he had faced decreased his satisfaction;

> *INTERVIEWER- … Did any difficulty you had with computer affect your satisfaction?*
>
> *CS1-STUDENT3- Sometimes I was upset because I couldn't. Because once, it just skipped… My computer was shut down and I couldn't complete the tasks. And I am also stressed out when it slows down. So all these affected my satisfaction badly.*
>
> *INTERVIEWER- … Yaşadığınız bilgisayarla ilgili zorluklar sizin memnuniyetinizi etkiledi mi?*
>
> *CS1-STUDENT3- Bazen yapamadım diye üzüldüğüm oldu. Boş geçti çünkü bir keresinde… Bilgisayarım kapanmıştı, yarısında kalmıştı görevler. Kasınca da sinir oluyorum. Bunlar da bizim memnuniyetinizi azalttı.*

Other factors were difficulty level of tasks and difficulty of use. Two students mentioned that difficulty of some tasks were above their level of capability. Difficulty of use was the other factor decreasing satisfaction. Of the total students, two claimed that it was difficult to use the programs and that this led to them to disliking the VW experience. The most problematic structure of the program was the inventory. CS1-STUDENT7 mentioned that she did not like to use the VW due to difficulty using its inventory and the complex structure;

> *INTERVIEWER- So was there anything that you didn't appreciate?*
>
> *CS1-STUDENT7- The inventory was a bit complicated. It was difficult to move it from left to right. And I also failed at drag and dropping them [from inventory]. They were the difficulties that I faced.*
>
> *INTERVIEWER- Peki sevmediğiniz şeyler var mıydı?*
>
> *CS1-STUDENT7- Biraz envanter çok karışıktı. envanter biraz sağa-sola o biraz zor geliyordu bana. Bir de benim yapamadığım şey alıp koyamamak… [Bunlar] biraz sıkıntı oluyordu bende.*

Other factor was about the avatar that was stated by two of the students. It was not possible to move avatars when they became jammed in some parts of the 3D environment. Two students defined this situation as annoying. The last factor stated by one of the students was the need to rearrange code when they were wrong. CS1-STUDENT7 described this situation as frustrating; explaining that she tried to rebuild the code when she got it wrong, however, in this situation; she became bored and thought about just giving up;

> *CS1-STUDENT7- ... Once you build the wrong code, you try to do it again but this time it gets boring, you think "Should I do it again, or not…"*
>
> *CS1-STUDENT7- ... Kodları yanlış yazınca zaten bir daha yapmaya çalışıyorsun ama bu sefer de sıkılıyorsun. Yapayım mı, yapmayım mı diye…*

4.7.2 Case-2: Extra-Curricular Program

Current Status of Satisfaction

The current satisfaction scores of participants were measured via the satisfaction questionnaire in this case, as well. Table 4.23 shows the descriptive results of the questionnaire for each item. As can be seen, the overall mean score (*M* = 3.99) was moderately high. Only two items relating to VW benefits learning achievement (*M* = 3.70) and satisfaction of overall learning effectiveness (*M* = 3.75) were lower when compared to the other two items.

Table 4.23 – *Satisfaction questionnaire results of Case-2*

	M	***SD***
I was satisfied with SDP learning experience.	4.40	0.68
I think this SDP benefit my learning achievement.	3.70	0.80
I was satisfied with SDP.	4.10	0.85
I was satisfied with the overall learning effectiveness.	3.75	0.91
Total	**3.99**	

Qualitative results indicated that the expectation of most of the students were met by the end of the study, except for only two students argued the opposite. For example, CS2-STUDENT5 argued that the study met his expectations more than he thought; because he thought he would learn programming in the club from a simple programming tool such as Lightbot;

> *INTERVIEWER- So do you think this training met your expectations about programming?*
>
> *CS2-STUDENT5- Yes but I wasn't expecting it to be like a virtual reality.*
>
> *INTERVIEWER- So you were expecting something simpler?*
>
> *CS2-STUDENT5- Exactly. I thought they would be simpler things like Lightbot for example.*
>
> *INTERVIEWER- Peki bu eğitim senin beklentilerini karşıladı mı programlama konusunda?*
>
> *CS2-STUDENT5- Karşıladı da benim beklentim böyle sanal gerçeklik gibi olacağını düşünmemiştim.*
>
> *INTERVIEWER- Daha mı basit düşündün yoksa?*
>
> *CS2-STUDENT5- Aynen. Daha basit şeyler olacağını düşünmüştüm. Lightbot gibi mesela.*

Among the non-satisfied students, CS2-STUDENT1 highlighted that this study was expected to have been easier, however, it was not as expected;

> *INTERVIEWER- What was your expectation from this club for example? What did you expect to learn?*
>
> *CS2-STUDENT1- I thought we would learn things such as programming because I know it from the games. We would learn such things so I supposed it would be easier. But it's not, I mean, it's hard.*
>
> *INTERVIEWER- Mesela beklentin neydi bu kulübe gelirken? Ne öğrenmeyi düşünüyordun?*
>
> *CS2-STUDENT1- Yine böyle programlama gibi şeyler öğreneceğimizi düşünüyordum. Çünkü oyunlardan biliyorum. Öyle şeyler öğreniriz kolay olur diye düşünüyordum ama, öyle olmadı yani, zormuş.*

Factors increasing satisfaction

The factors increasing the satisfaction of students in this case were similar to those of the first case. There was only one emerging factor that was different from the first case, which was about the story of tasks. Other factors emerged in this case too.

The most-cited factor increasing the satisfaction of students in this case were object construction, 3D environment and off-task activities, which were each stated by six of the students. For example, CS2-STUDENT2 highlighted his greatest liked as, "It is possible to build and program what you want." Off-task activities and the 3D environment were the other emerging factors. Students liked to do extra things that were not related to the tasks they had been assigned in their free time. CS2-STUDENT2 and his teammate liked to resize the objects they constructed as an off-task activity in their free time after having finished their tasks and while waiting for the others to complete their own tasks;

> *CS2-STUDENT2- But when my teammate and I were waiting for others to finish after we were done with ours, we were creating more complicated stuff like houses with the things we used for other tasks, which we liked more.*
>
> *CS2-STUDENT2- Ama biz takım arkadaşım ile o zaman görevler bittikten sonra diğer kişilerin bitirmesini beklerken böyle başka görevlerde kullandığımız maddeleri büyütüp evler gibi karmaşalar yaratmayı daha çok beğendik.*

Group study and tasks were the other emerging factors, with group study cited by five of the students, and tasks cited by four of the students. Similar to students in Case-1, students liked to be in a group and defined the group study as having been enjoyable. For example, CS2-STUDENT2 defined this issue as studying in a group and being able to receive help were enjoyable and fun. Stories about each task was another emerging factor, which is different from Case-1. The teacher of this case and three of the students mentioned this factor. The students stated that stories about the tasks provided them with a mission, a reason to complete the tasks and the stories were creative, funny, and exciting. The teacher of this case also argued that the stories were helpful since they drew the students' attention to the tasks and it was a satisfying factor for the students to complete the tasks;

> *INTERVIEWER- And Madam, every task had a story. What do you think about those stories?*
>
> *CS2-TEACHER- This is what we always do also in our lessons. I mean it definitely should have a story or a scenario and the kid should concentrate on that scenario so that s/he would feel enthusiastic while completing the tasks. I think it's satisfactory.*
>
> *INTERVIEWER- Bir de Hocam her görevin bir hikayesi vardı. Bu hikayeler hakkında ne düşünüyorsunuz?*
>
> *CS2-TEACHER- Bu bizim dersimizde de hep yaptığımız bir şeydir. Yani mutlaka bir hikayesi olmalı, bir senaryosu olmalı ve çocuk o senaryonun içine girebilmeli ki görevi tamamlamak için şevk duysun. Onu da başarılı buluyorum.*

The last factor that emerged was touring in the 3D environment, which was mentioned by two of the students who liked to travel in the 3D environment by way of walking, flying, and driving cars.

Factors decreasing satisfaction

The factors decreasing the satisfaction of the students in this case were quite similar to those in Case-1. A less than realistic environment and difficulty of use were the two most-cited factors decreasing satisfaction. Four of the students found the environment less than realistic and the graphics of the objects were defined as low. CS2-STUDENT5 commented on this issue as, "It seems that the 3D environment was less than realistic and the graphics of the 3D objects were low." Difficulty of use was cited by four of the students as another factor. The inventory and arranging the pop-up screen were stated as complex and also less liked. It was mentioned that it could be easier to use and thereby simpler to cope with. Being left to work alone in the 3D environment, technical problems encountered and the difficulty of the tasks were stated by three of the students as the least-liked. CS2-STUDENT3 mentioned about the difficulty of tasks; adding that he would be more satisfied if they could have been made easier. Technical problems encountered were considered as annoying situations for some of the students. CS2-STUDENT7 pointed to a technical problem as, "I did not like it when the computer froze." Issues related to the avatars such as getting jammed in the environment and not being able to change accessories were other factors stated by two of the students. Being alone was another factor that also decreased satisfaction. Students generally studied with their peers; however, they were alone in some situations. One of the students mentioned about studying alone as;

> *CS2-STUDENT6- As I said, it was boring when we were on our own, we couldn't do anything. Go there, put it, make this, turn it, and the task is over. But now that we have our friends, we go to the task, and do it together. I mean we help each other.*
>
> *CS2-STUDENT6- Şimdi tek başımıza söylediğim gibi çok sıkıcıydı, hiçbir şey yapamıyorduk. Git, koy, yap, çevir, görev bitti. Ama arkadaşım olunca şimdi biz göreve gidiyoruz, birlikte ikimiz yapıyoruz, birbirimize yardım ediyoruz.*

4.7.3 Case-3: After-School Program

Current Status of Satisfaction

Descriptive results of the satisfaction questionnaire were similar to those of Case-1, and are presented in Table 4.24. As can be seen, the students ranked each item higher than four points. The overall mean (*M* : 4.28) was quite high compared to Case-2. Moreover, the mean of each item was higher than four points.

The qualitative results revealed from the interviews supported the descriptive results. Most of the students mentioned that the overall lesson met their expectation and that they enjoyed the study. CS3-STUDENT3 assumed that the lesson would similar to a traditional lesson in which PowerPoint presentations were used. On the contrary, she argued that it was more than what she had expected;

INTERVIEWER- And did this training meet your expectations?

CS3-STUDENT3- Yes, it did. It was even better than I'd expected. I thought we would do it after learning some more things from the presentations. I did not know there would be games or something like that.

INTERVIEWER- Bu eğitim senin beklentilerini karşıladı mı peki?

CS3-STUDENT3- Evet, karşıladı. Hatta beklediğimden daha güzel geldi. Ben böyle sadece slayttan falan bir şeyler öğrenip öyle yapacağız zannediyordum. Oyun falan olduğunu bilmiyordum.

Table 4.24 – *Satisfaction questionnaire results of Case-3*

	M	***SD***
I was satisfied with SDP learning experience.	4.38	0.74
I think this SDP benefit my learning achievement.	4.38	0.92
I was satisfied with SDP.	4.13	0.83
I was satisfied with the overall learning effectiveness.	4.25	0.89
Total	**4.28**	

Factors increasing satisfaction

Results showed that the emerging factors for this case were similar to those ones of Case-2, but with different frequencies. The major increasing factor in the satisfaction of the students (n = 6) was object construction. All of the students in this case mentioned this issue as a satisfying factor. CS3-STUDENT6 expressed her feeling as, "I most liked being able to build objects, which does not happen so fast in real life." The other most satisfied factor was group study, as stated by five of the students. In this case, the students liked being in a group as much as the students in the other cases. They expressed their feelings about group study as more enjoyable and fun. Off-task activities were the third major factor, which was stated by four of the students. In this case, the students liked to do off-task activities such as playing in the 3D environment with their friends, chatting in the breaks and so on; similar to the students of the other cases.

The other factors were tasks, 3D environment and story of tasks, which were each stated by three students. The students stated that they liked the tasks and their stories as well as the 3D environment. For example, one of the students, CS3-STUDENT4, mentioned that the 3D environment was fascinating and had a well-planned story and that they all had a good rapport with each other. The last factor stated was touring the 3D world. Traveling by car was stated by one of the student as the most enjoyable;

> *CS3-STUDENT4- The story was well-matched with the island too. The island was planned very well, the story and everything were perfect. Actually the tasks, story and the island matched together very well.*
>
> *CS3-STUDENT4- Hikaye uyumluydu ve adaya da uyumluydu. Ada çok güzel planlanmıştı, hikaye falan hepsi muhteşemdi. Görevler açıkçası adaya da, yani üçü de birbirine benziyordu, uyumluydu.*

Factors decreasing satisfaction

The results showed that technical problems, studying alone, a less than realistic 3D environment, the difficulty of the tasks and avatar-related problems were all factors decreasing the satisfaction of the students in this case. Technical problems were the most-cited, as mentioned by four of the students. For example, one of the students mentioned this as;

> *INTERVIEWER- And what was the thing that you did not like in this environment?*
>
> *CS3-STUDENT3- [computer] froze once in a while, it was continuing on its own and it was not very good.*
>
> *INTERVIEWER- Peki beğenmediğin ne vardı bu ortamda?*
>
> *CS3-STUDENT3- Arada bir donuyordu, böyle kendi kendine takılıyordu o çok güzel değildi.*

The other factors were only stated by one of the student in this case. While one stated that being alone would be boring for him, another found the use of the VW hard. These were stated as annoying factors which caused them to be less satisfied. Moreover, one of the students compared the VW with the real world and commented that the VW was less than realistic. CS3-STUDENT6 explained this issue with an example, arguing that although it was possible to understand the direction of a ball in the real world, it was not possible to realize this in the VW; adding that this was so frustrating for her;

CS3-STUDENT6- ...For example if I [had] a ball in real life, I would know where it would go. However [in virtual world] you have to change the direction of the ball all the time because you do not know the direction [that the ball would go to]. It really upset me.

CS3-STUDENT6- ...Mesela topun yüzeyini gerçek hayatta yaptırsaydım fark edersin, top nereye giderse gitsin anlarsın, ama [sanal dünyada] buradaki topun yönünü bilmediğin için sırayla bir sürü yön değiştiriyorsun. Bu çok canımı sıktı.

4.7.4 Cross-Case Analysis

The students' satisfaction level was measured via a questionnaire in each of the three cases. The students' overall mean satisfaction scores for each case were just higher than four points. The descriptive results of the quantitative data shows that the most satisfied students were from Case-1 (M = 4.35), Case-3 (M = 4.28) and Case-2 (M = 3.99), respectively. Students of Case-1 were not so satisfied in the first part of study due to the technical and infrastructural problems that they experienced. However, those problems were overcome in the second part of the study, which could be why their scores were higher than others in the end. It seems that the least satisfied students were those from Case-2. This might due to fact that they were in a more crowded class and it was impossible therefore for the teacher to deal with all the students to the same extent as the other cases. Moreover, some of them had high expectations from the club. Using VW in the club was unknown to the students before joining the club and it might therefore not have met their expectations as a whole. Students of Case-3 knew that the VW would be used to teach the basics of programming, having been informed and volunteered to take part in the course. This may explain why their satisfaction scores were above four points, almost to the level of the students of Case-1.

Increasing factors in the satisfaction of the students were determined in all three cases. Apart from the story about the tasks, other factors emerged across all three cases at different ratios (see Table 4.25). Story of the tasks was not stated as an increasing factor by the participants of Case-1. This may be due to the students in that case either disliking or having ignored the stories. The most satisfied factor stated by the students across all cases was group study. Building and programming objects together, off-task activities and the 3D environment were stated as the most satisfying factors for both

Case-2 and Case-3, but they were not found so satisfying by the students of Case-1. However, Case-1 was applied as a more formal class exercise than the other cases. Due to timetabling problems in this case, students did not have so much free time to do off-task activities and explore the 3D environment on their own.

As to the factors that decreased the students' satisfaction levels; technical problems, studying alone, difficulty of the tasks, and avatar-related problems were all factors stated by some of the students across all cases (see Table 4.25) that decreased their satisfaction. However, a less than realistic 3D environment was not found as a decreasing factor for the students from Case-1, but was found to be the most decreasing factor in Case-2 and less so in Case-3. This implies that while the students of Case-2 had a high expectation about the reality of the 3D environment, it appears that the students from Case-1 had no concerns about this issue. On the other hand, the students from Case-3 found the system easy to use (see the section on the perceived ease of use results) and no student from this case not mentioned any difficulty of use as a decreasing factor.

Table 4.25 – *Frequencies of Factors Affecting Satisfaction across the Cases*

	Case-1		**Case-2**		**Case-3**	
	n	***%***	***n***	***%***	***n***	***%***
Factors Increasing Satisfaction						
Group study	6	86	5	50	5	83
Object construction	2	29	6	60	6	100
Having tasks	5	71	4	40	3	50
Off-task activities	1	14	6	60	4	67
3D environment	1	14	6	60	3	50
Story of tasks	-	-	3	30	3	50
Tour in the 3D environment	1	14	2	20	1	17
Factors Decreasing Satisfaction						
Technical problems	2	29	3	30	4	67
Studying alone	3	43	3	30	1	17
Difficulty of tasks	2	29	3	30	1	17
Avatar-related problems	2	29	2	20	1	17
Less than realistic environment	-	-	4	40	1	17
Difficulty of use	2	29	4	40	-	-

4.8 Sub RQ – Issues and Strategies for the Design of Educational Programs

The educational design of the cases were different from each other due to the nature of each case. The total course hours was 15 lesson hours across all three cases; however, the weekly course hours varied. In this section, the issues and strategies about the design of the educational programs are elaborated on. The results revealed four different issues and strategies. Firstly, the weekly course hours for each case is addressed; secondly, the number and difficulty level of the tasks is investigated; and thirdly, the issues and strategies regarding feedback are presented. Lastly, the issues related to the instructional materials that were presented to the students in the form of task cards and informational presentation tools are addressed.

4.8.1 Case-1: Curricular Program

Weekly Course Hours

In the first part of this case, the weekly course hours were two lesson hours (45 minutes each) for a period of four weeks. However, it was not possible to do the lesson each week due to various reasons such as an electricity power cut in one week. In the second part of the study, the students were invited to the university's CEIT department twice in order to complete the remainder of the study. The course hours were three lesson hours (45 minutes each) in the second part of the study for this case. The teacher's and students' ideas about the weekly course hours were taken via interview questions. The teacher of this case stated that the course hours were sufficient; however, the first part of the study was less than efficient due to infrastructural issues with the school's computer laboratory. Therefore, the course hours may have seemed to be inadequate. The results of the student interviews indicated that the course hours were insufficient, and most of the participants ($n = 5$) found the course hours were too low, having stated that it could have been more.

Number and Difficulty Level of Tasks

Students' ideas related to the number and difficulty level of the tasks were taken via interview questions. Two students and the teacher of this case indicated that the number of tasks was sufficient. For example, the teacher highlighted this issue as;

INTERVIEWER- …Do you think the number of tasks was sufficient to teach the topic and the basics of programming?

CS1-TEACHER – I think it was. …I believe the tasks went on from the easy levels to more difficult levels gradually. If you are asking about whether the number was enough, yes, I think they are.

INTERVIEWER- …Görev sayısı sizce konuyu, programlama temellerini öğretmek için yeterli miydi?

CS1-TEACHER- Bence yeterli. … Bence basitten karmaşığa doğru gidersek kolaydan zora doğru gitti diye düşünüyorum yani görevlerin. Sayı olarak yeterli mi diye soruyorsanız, yani bence yeterli.

However, four of the students indicated that the number of tasks could have been more, although they were unable to explain the reason behind their preference. They mostly asserted that the tasks were funny and there could have been more. One of the students argued that it would be better if there were more tasks since the subjects were enjoyable.

The difficulty level of the tasks was investigated as to whether or not they were perceived as being difficult. Three of the students' and the teacher's responses to the interview questions indicated that the tasks were neither hard nor easy. For example, CS1-STUDENT6 explained this as the tasks were moderately difficult, meaning neither hard nor easy. Besides, there were some hard tasks as well as easy ones. The teacher of this case stated that the difficulty level of the tasks was interrelated to the level of the students' interest and their mathematical intelligence. Those who were more interested and who had higher levels of mathematical intelligence completed the tasks more easily and to a better standard. She added that the difficulty level of the tasks was appropriate;

INTERVIEWER- Do you think the difficulty of the tasks were matching with the students' capability?

CS1-TEACHER- I think it changes from student to student. There were the ones who were curious and more interested. I believe that the students with a little more mathematical intelligence do better in virtual worlds. But in general the difficulty was suitable for children.

INTERVIEWER- Sizce görevlerin zorluğu öğrencilerin yapabileceği düzeyde miydi?

> *CS1-TEACHER- Bence öğrenciye bağlı. Merak eden, daha ilgili olan öğrenciler vardı. Biraz daha ilgisiz olan öğrenciler de var. Biraz matematik zekası fazla olan çocukların sanal dünyanın içerisinde daha iyi yaptığını düşünüyorum ben. Ama genel olarak görevlerin zorluğu uygundu.*

Feedback

Feedback has vital importance in educational programs since it informs the learners about their actions. In the current study, feedback was aimed to be given via different platforms. Teachers in the physical environment and videos in the 3D environment were the main sources of feedback for the students. Moreover, the students sometimes preferred to take feedback from their peers. Participants' preferences on the source of feedback and their suggestions about those were investigated and are presented in this section.

In the first case, the participants mostly preferred feedback from their teachers and from videos in the 3D environment. Of the total students, four of them preferred feedback from their teacher and five of them argued that they preferred feedback from the videos. For example, CS1-STUDENT6 argued that she got help from the teacher when she had difficulty and that the teacher managed to help her in those situations. In contrast to CS1-STUDENT6, CS1-STUDENT1 preferred to take feedback from the videos and in that way he had learned how to build code on S4OS and how to complete the tasks;

> *CS1-STUDENT1- There were help boxes, you prepared a video and put it in those. That was instructive. You also told us how we were supposed to do it. For example at first we did not know what kind of code we were going to write here, whether we were supposed to write them on keyboards or on Scratch, we did not know. I learned it thanks to the videos.*

> *CS1-STUDENT1- Bilgi kutucukları vardı, oraya video hazırlayıp atmışsınız. O bilgilendiriyordu. Hem de o yaparken orada anlatmıştınız nasıl yapacağımızı. Mesela ilk başta burada nasıl kodlar yazacağımızı bilmiyorduk. Kodları direkt klavye üzerinden mi yazacağız, yoksa Scratch'ten mi yapacağız bilmiyorduk. Bunu videolardan öğrendim.*

The students' and the teacher's suggestions were sought with regard to the feedback sources via interview questions. Emerging suggestions were mostly related to the teacher and video as feedback sources. One of the students suggested that the presence of the teacher in the virtual environment could be increased. In this way, the teacher could demonstrate actions to the students and they in turn could see the teacher's actions. There were some suggestions about the videos too. One of the students complained about the duration of the videos. He inclined that the videos were too long and that they could have been summarized. Three of the students suggested that the quality of the videos could have been better and that a video should be prepared for each task in the 3D environment.

In addition, there were other suggestions mentioned by one of the participants. One of those suggestions was about informing the participants when they had completed the task. This was a point seen as missing from the 3D environment. CS1-STUDENT3 expressed this as, "I would like to see a message telling you that you had completed the task." Another suggestion by one student was that there should be virtual characters guiding the students when they experienced difficulties. The last suggestion stated by the teacher of the case was that interactive help could have been offered to the students so that they could give instructions to the students gradually.

Task Cards and Information Presentation Tools

Some instructional materials were distributed to the students such as task cards as hard copies, and posters displayed on billboards in the 3D environment. Students' ideas related to the instructional materials were investigated via the interviews. The results indicated that the task cards were seen as helpful by the students. The task cards informed the students in terms of which task to complete and how to complete it. Most of the students (n = 7) and the teacher agreed that they were helpful. Only one student reported that there were missing or incorrect points on the task cards. He suggested that those points needed to be rectified. Two of the students suggested that the task cards could contain the code of the tasks. In this way, they stated that they would be able to compare their own code with the correct code and thereby receive help when they needed it.

Some information was presented on billboards in the 3D environment. The students' ideas about these informational tools were sought out. Only one student in this case indicated that they were actually helpful. Their suggestions were also gathered. One of the students suggested that they could include videos rather than static posters. He added that either a TV broadcast or a live stream of the teacher could be presented on boards in the 3D environment;

CS1-STUDENT5- Sir, it would be better if there were televisions instead of boards. You give us the code and we do it by showing them.

CS1-STUDENT5- Hocam, levha yerine televizyon olsaydı daha güzel olurdu ya da Hocam siz çıkıyorsunuz kodları gösterip yapıyoruz.

4.8.2 Case-2: Extra-Curricular Program

Weekly Course Hours

The course hours were 1.5 lesson hours (60 minutes each) in length for this case, and lasted for a period of 10 weeks. Interview results revealed two opinions. Half of the participants (*n* = 5) found that the course hours were enough and ideal. For example, one of them mentioned that it was sufficient in total for the completion of the tasks considering the number of tasks in total.

However, in contrast the other half of the students and the teacher found the course hours to be insufficient, stating that there could have been more. The teacher of this case argued that there could be an extra half lesson hour each week. Namely, she argued that course could be two lesson hours with each lasting a total of 80 minutes. She explained why she saw a need for the extra course hours as;

INTERVIEWER- How do you think were the weekly course hours?

CS2-TEACHER- In some cases they were not enough. I mean I wished that we had had 30 more minutes. As soon as the students gather their speed and do their tasks, the lesson finishes. It could have been better if they were 80 minutes as two blocks on the condition that it would not be too often.[it would be better in terms of reinforcement of the knowledge and feedback] if we had 20 more minutes in some of the tasks because each time we tried to reach every kid and give feedback. We could use 20 more minutes due to these reasons.

INTERVIEWER- Sizce haftalık ders saatleri nasıldı?

CS2-TEACHER- Bazı durumlarda yetmedi. Yani bir 30 dakikamız daha olsaydı diye düşündüm. Yani tam böyle çocuklar bir ivme kazanıyorlar, görevi yapacaklar o sırada ders bitiyor. Çok hani üst üste olmamak koşuluyla işte bu belki 2 ders saati üst üste, yani 80 *dakika falan olsa daha iyi olabilirdi. 60 dakika yerine bir 20 dakikamız daha olsa bazı görevlerde, bazı derslerde anlatılan bilginin de oturması açısından, geri bildirim açısından çocuklara çünkü her defasında hemen hemen her çocuğa ulaşıp geri bildirim vermeye çalıştık. O açıdan belki bir 20 dakikaya daha ihtiyacımız olabilirdi.*

Number and Difficulty Level of Tasks

The participants were asked whether or not the number of tasks were sufficient. Only two of the students argued that it could have been more. Six of the students' and the teacher's responses indicated that they were sufficient; mentioning that they were sufficient for learning the subject. The teacher of this case also argued that the number of tasks were sufficient to enable students to create things on their own in the 3D environment;

INTERVIEWER- Do you think the number of the tasks were enough in order to teach the subjects, that is the basic concepts of programming?

CS2-TEACHER- Absolutely yes, they were enough. I mean a student who finishes the last task knows the basic programming topics and is ready enough to create something. It is up to his/her creativity after this point.

INTERVIEWER- Sizce görevlerin sayısı konuları, hani programlamanın temel kavramlarını öğretmek için yeter miydi?

CS2-TEACHER- Kesinlikle yeterdi, evet, yeterdi. Yani son görevi bitiren bir öğrenci temel olarak programlama konularını bilip artık bir şeyler oluşturmaya başlaması konusunda çocuğu hazır hale getirdi. Bundan sonrası onun ne kadar yaratıcılık kattığına kalıyor.

The students' responses about the difficulty level of the tasks indicated that it was moderate. Most of the students argued that the tasks were moderate and at the appropriate level in general. For example, CS2-STUDENT2 stated that the level of tasks were moderate, and that except for the last task, they were neither hard nor easy. The teacher of the case also argued that the tasks were not so hard. She mentioned that there were some tasks which took more time due to the students' level of readiness. Overall, they were at an appropriate level for the students to complete them;

INTERVIEWER- Another question; do you think these tasks are in a level that the students could achieve them?

CS2-TEACHER- I think yes. I don't think there is a specific task that students had problems with in general… Of course some of the tasks might have taken more time than the others. I think… That was because all the students did not have the same background but they were able to handle them to a large extent.

INTERVIEWER- Diğer sorum da; sizce bu görevler öğrencilerin yapabileceği düzeyde miydi?

CS2-TEACHER- Bence evet. Genel olarak çocukların çok da zorlandığı bir görev olduğunu da düşünmüyorum…Tabii ki bazı görevler diğerinden daha fazla zaman almış olabilir sadece. Bence… O şeyden kaynaklandı. Her öğrencinin aynı … hazır bulunuşluk seviyesinde olmamasından kaynaklandı ama, büyük oranda bence şeydi, yapabildiler.

Feedback

In this case, the students received feedback from their teachers, videos and also from their peers. Different to the other cases, the students also received feedback from their peers. Of the total students, seven argued that they received feedback from their teachers, eight students received feedback from the videos, and three students received feedback from their peers. The students found that the videos were helpful in terms of understanding and informing them on some issues that they could not do. Three of the students and the teacher of this case mentioned that the students received feedback from their peers. The teacher of this case argued that the students preferred not to follow the guidelines or watch the videos as they easily became bored. Instead of this, she added that they would prefer to seek help from their peers; which was quick and easy for the students;

CS2-TEACHER- I think the students do not read, inquire or watch things such as helping videos or other sources since they get bored. … When there is a lesson involved in it, the kid is not interested in that part. They prefer learning from each other, after an observation of years, [I can say that they think as follows]: 'You know this game better, I know it less than you, so I prefer learning it from you.' Not because of the video or anything else, they get bored because they think they are going to learn from their friends in a better way.

CS2-TEACHER- Sıkıldıkları için öğrenciler yardım videosu gibi ya da işte böyle bir kılavuz kaynak gibi şeyleri çok fazla okumuyorlar, incelemiyorlar, izlemiyorlar diye düşünüyorum. … İşin içinde ders olunca istediğin kadar video hazırla çocuk oraya çok takılmıyor. Birbirlerinden öğrenmeyi daha çok tercih ediyorlar, bak bu noktada *ben yılların verdiği bir gözlem, yani 'Sen oyunu iyi biliyorsun, ben de daha az biliyorum, senden öğrenmeyi tercih ediyorum ben.' Video falan değil yani, sıkılıyor, çünkü 'Daha çabuk öğreneceğim arkadaşımdan.' diye düşünüyorlar.*

Interview results indicated that there were five types of suggestions; most of which were about the videos. Four of the students made suggestions about this issue. Two of them complained that the duration of the videos were too long and took too much time to load; suggesting that they could be made shorter. The other student suggested that important points in the videos should be highlighted. Another important suggestion stated by four of the students was for virtual characters in the 3D environment to guide the learners. Those characters would help by interacting with the learners. For example, one of the students expressed his ideas about this as, "A human-like character should be in the virtual environment, and it could give hints when it was touched." Giving some hints to the learners was another suggestion stated by three of the students. For example, CS2-STUDENT10 suggested that there could be some hints when they could not complete the task after a certain length of time. The other suggestion which was made by the teacher of this case was to inform learners when they completed tasks. The last suggestion was about teacher feedback as stated by one of the students. He inclined that teachers were in a hurry and added that they could slow the lesson pace down a bit.

Task Cards and Information Presentation Tools

The interview results revealed that most of the students (n = 6) and the teacher of this case found the task cards helpful. For example, one of them inclined that they were helpful because he learned what to do from the task cards. The teacher also argued that they assigned the students responsibility for completing all of the tasks. Moreover, she added that students also followed their completed and to-be-completed tasks from the task cards;

INTERVIEWER- What do you think about the task cards?

CS2-TEACHER- …The task cards were fine. The kids were able to follow them. Even in the last lesson the students were still after the task cards. I mean it was good for the kids that they could follow from there. … Also they contributed to their feeling of responsibility and maybe a *kind of seriousness. They realized that they were being followed on there as well.*

INTERVIEWER- Görev kartları hakkında ne düşünüyorsunuz?

CS2-TEACHER- ...İyiydi görev kartları. Takip edebildi çocuk, hatta bugün son derste bile hala öğrenci görev kartlarının peşindeydi. Yani oradan takip etmesi onun çoğu noktada işine geldi. ... Bir de sorumluluk kattı. Belki işe bir ciddiyet de kattı. ... Takip edildiğini de oradan fark etti.

Students' suggestions about the tasks cards were also sought. Two of them suggested that the task cards do not need to contain the task stories. Another student suggested that they needed to be more brief. The last suggestion stated by a student was that task cards should also contain the code of the task.

The interview results indicated that the informational presentation tools were helpful to some extent, with three of the students stating that they were helpful. In contrast, two of the students indicated that there was no need for the information presentation tools since they did not attract their attention; stating that they could either be removed altogether or be made more attractive in order to attract attention. One of the students suggested they could contain more visuals rather than text. The last suggestion was both related to the task cards and the information presentation tools. CS2-STUDENT8, suggested that the information on task cards should be on the boards in the 3D environment, which could then inform others in the 3D environment about the tasks completed;

CS2-STUDENT8- I think you could have put the texts in the files on a sign on there. For example there are red and blue boxes, and a box with a question mark on it. If there were explanations next to it, we could look at it there and do [our tasks] easily. [Also] When the task was over, we could click on there and it would be visible to everyone. For example Someone would have written 'The task has been completed.' to the teacher now.

CS2-STUDENT8- Bence bir de o dosyada yazanları orada bir tabelaya koyabilirdiniz. Mesela orada kırmızı ve mavinin kutuları var, bir de soru işareti kutusu var. Onun yanında da açıklama olabilir 2 grup için, oraya bakıp daha kolay yapabilirdik. [Ayrıca] görevi bitirdiğimizde oraya tıklardık, herkese açık olurdu. Mesela birisi şimdi 'Görevi tamamladı.' yazardı öğretmene.

4.8.3 Case-3: After-School Program

Weekly Course Hours

The course hours were three lesson hours (45 minutes each), lasting for a period of five weeks in this case. Four of the students found that the course hours were sufficient. One of them emphasized that, "It was satisfactory because the number of lessons and breaks were sufficient." The weekly course hours seemed to be more appropriate in this case when compared to the other two cases. Only one student argued in favor of four lesson hours per week.

Number and Difficulty Level of Tasks

Students' ideas related to the number of tasks were investigated in this case too. Only two of the students found the number of tasks to be sufficient in this case. The other four students argued that the total number of tasks was insufficient and that there could have been more. For example, CS3-STUDENT3 commented that, "There could be 20 tasks and more course hours; then it would be more fun."

As to the difficulty level of the tasks, most of the students agreed that the level of tasks were appropriate for them to complete the tasks. They stated that except for a few easy and difficult tasks, most were generally of a moderate level. For example, CS3-STUDENT6 thought that they were not so easy for her at the beginning of the course. However, she then said that they became getting easy in time;

> *INTERVIEWER- And what do you think about the difficulty level of the tasks?*
>
> *CS3-STUDENT6- Some of the tasks are too hard, some of them are too easy. In some of them I got some help but then I did them by myself. They have got more difficult gradually. In fact I did not think that they would be that easy when I first came here. ... But they turned out to be easier even just a bit.*
>
> *INTERVIEWER- Peki zorluğu konusunda ne düşünüyorsun görevlerin?*
>
> *CS3-STUDENT6- Bazı görevler çok zor, bazı görevler çok kolay, bazılarında yardım aldım ama sonra kendim de yaptım. Gitgide zorlaştı. Aslında ilk geldiğimde o kadar kolay olacağını düşünmüyordum. ... Ama az da olsa kolay çıktı.*

Feedback

Students in this case usually received feedback from two sources; the videos and the teacher. Of the total students, five of them preferred to receive feedback from the videos and four of them stated that they received their feedback from the teacher. CS3-STUDENT1 highlighted about the videos as, “They were so helpful for me because I could not attend the course one week, but I could complete the tasks after receiving help from the videos.”

The results revealed four types of suggestions received for this case, with each having been stated by one participant. The first was about teacher feedback, with one student suggesting that teachers could also be seen in the virtual world and then help them in the environment in terms of how to do things, which would be instant and easy;

> *CS3-STUDENT1- For instance, it could have been like this: our teachers could log in all of our worlds and they would help us immediately by showing what we could do. Then, they would enter the world of a friend who would be having trouble and it would have been more comfortable.*
>
> *CS3-STUDENT1- Mesela şöyle olabilirdi; öğretmenlerimizin bilgisayarı hepimizin dünyasına bağlanabilirdi, öyle hemen yardım ederdi bizim nasıl yapacağımızı gösterip. Sonra çıkıp yardım isteyen arkadaşımızın dünyasına girerdi, daha rahat olurdu hem de.*

Another suggestion was about watching the videos since one student inclined that she was unable to load the videos. Other suggestions were to guide learners through the use of virtual characters, and to give hints when needed.

Task Cards and Information Presentation Tools

Students’ ideas were also taken in this case with regard to the task cards and the informational presentation tools. All of the students (n = 6) found that the task cards were helpful. For example, CS3-STUDENT4 expressed that the “[task cards] informed us about which task to do and how to do them. They were giving hints to us.” However, there was only one suggestion stated by one student about the task cards in this case. CS3-STUDENT6 found some of the information on the task cards as unnecessary since she had already know it. She suggested that they could be more brief;

CS3-STUDENT6- There were some necessary parts and unnecessary parts. For example, it writes there 'Square has four sides,' but, I know that, there is no need to that.

CS3-STUDENT6- Gerekli yerleri de vardı, gereksiz yerleri de vardı. Mesela karenin dört kenarı var yazıyordu, onu biliyordum, o yüzden yazmasına gerek yoktu.

As to the participants' ideas about the informational presentation tools, they were positive. Two of the students found them to be helpful. Moreover, one student suggested that they could be more visible and readable since it was sometimes hard to recognize the texts from a distance in the 3D environment;

INTERVIEWER- How were the boards in that environment?

CS3-STUDENT3- They were noticeable. But [I wish] they would be more readable. The texts on the boards were a bit too blurry. We had to get too close [to make them out].

INTERVIEWER- Ortamdaki panolar nasıldı?

CS3-STUDENT3- Fark edilebiliyordu. Ama daha net okunabilirdi sanki. Panolar da biraz bulanık gösteriyordu. İyice yaklaşmak gerekiyordu.

4.8.4 Cross-Case Analysis

There were some similarities and differences on the issues and strategies in the educational programs of each case, and Table 4.26 presents the frequencies across the three cases. The first emerging sub-theme was weekly course hours. As can be seen from Table 4.26, the students' and teachers' ideas about the weekly course hours varied across the cases. This was quite normal since the weekly course hours also varied across the cases, although the total hours for the course were almost the same. For example, most of the students in Case-3 found that the three lesson hours per day to be sufficient, but the students and teachers of Case-2 found their one and half-hours per week to be inadequate. Case-2 was conducted in a club, meeting for one and a half lesson hours for a period of 10 weeks. However, the tasks were left half-finished since the weekly course hours of the club was deemed to be insufficient. On the other hand, students of Case-1 found their two lesson hours per week for the first part of the study and three lesson hours for the two weeks of the second part of the study to also be insufficient.

As to the total number of tasks in the current study, some of the students in Case-1 and Case-3 found them to be insufficient; however, most students in Case-2 and the teachers across all cases found the number of tasks to be sufficient. Those arguing that the number of tasks were sufficient considered the duration of the academic semester and the topics to be covered. The others who argued that the number of tasks were insufficient mostly considered that the lesson was enjoyable and could last longer by including some extra tasks. Apart from a few students, most students and the teachers of all cases mentioned that the difficulty level of the tasks was neither too hard or too easy, agreeing that they were generally moderate.

Table 4.26 – *Frequencies of Issues and Strategies for Educational Programs across the Cases*

	Case-1		**Case-2**		**Case-3**	
	n	***%***	***n***	***%***	***n***	***%***
Weekly Course Hours						
Sufficient	-	-	5	50	4	67
Insufficient	5	71	5	50	1	17
Number of Tasks						
Sufficient	2	29	6	60	2	33
Insufficient	4	57	2	20	4	67
Feedback Source Preference						
Expert videos	5	71	8	80	5	83
Teacher	4	57	7	70	4	67
Peer		-	3	30	-	-

Feedback sources and suggestions related to those sources were the other issues and strategy noted for this theme. The mutual feedback sources for the three cases were videos and the teachers. Students mostly preferred getting feedback from the videos, followed by getting feedback from their teachers (see Table 4.26). The only difference in the feedback source was seen for Case-2. Some students in Case-2 preferred to receive feedback from their peers as well as from other sources. This might be due to the high number of participants in Case-2. The teacher and videos might have been seen as inadequate from time to time, and the students therefore preferred taking feedback from their peers. There were some suggestions made by the students about the feedback sources. At least one student from each case suggested issues about the videos, the teacher and the virtual characters (NPCs). One student from Case-1 and the teacher of Case-2 suggested informing learners when they had completed a task. Giving hints about the tasks was suggested by three of the students from Case-2 and one from Case-3.

The students and teachers from all three cases usually found the tasks cards and information presentation tools in the 3D environment as helpful. Some students highlighted that there were some missing points on the task cards and made suggestions that the task cards should be brief and contain the code of the task. Then, it was stated that the information presentation tools should be made to be more attractive and clear to read.

4.9 Summary of the Results

The main purpose of this study is to investigate the use of virtual worlds in teaching the basics of programming to children across different educational programs. The findings of the study are presented in this chapter in line with the sub-research questions. In order to summarize the overall findings, the emerged themes, sub-themes, and categories are illustrated in Figure 4.1.

The first sub-research question was about the perception of participants on the ease of use and perceived usefulness of SDP. Quantitative data showed that students found the use of SDP as easy in all cases with a mean score of 3.90, 4.09 and 4.19 respectively. Qualitative findings revealed that students experienced some minor difficulties especially in the adaptation period of the 3D environment, navigation, inventory and locating 3D objects. Another investigated issue was the perceived usefulness, with descriptive statistics of the perceived usefulness questionnaire showing the overall mean value of perceived usefulness as 4.14 for Case-1, and 4.13 for Case-3 and 3.19 for Case-2. Qualitative results of the interviews supported the quantitative results and it was found that using the VW facilitated the students' learning of programming.

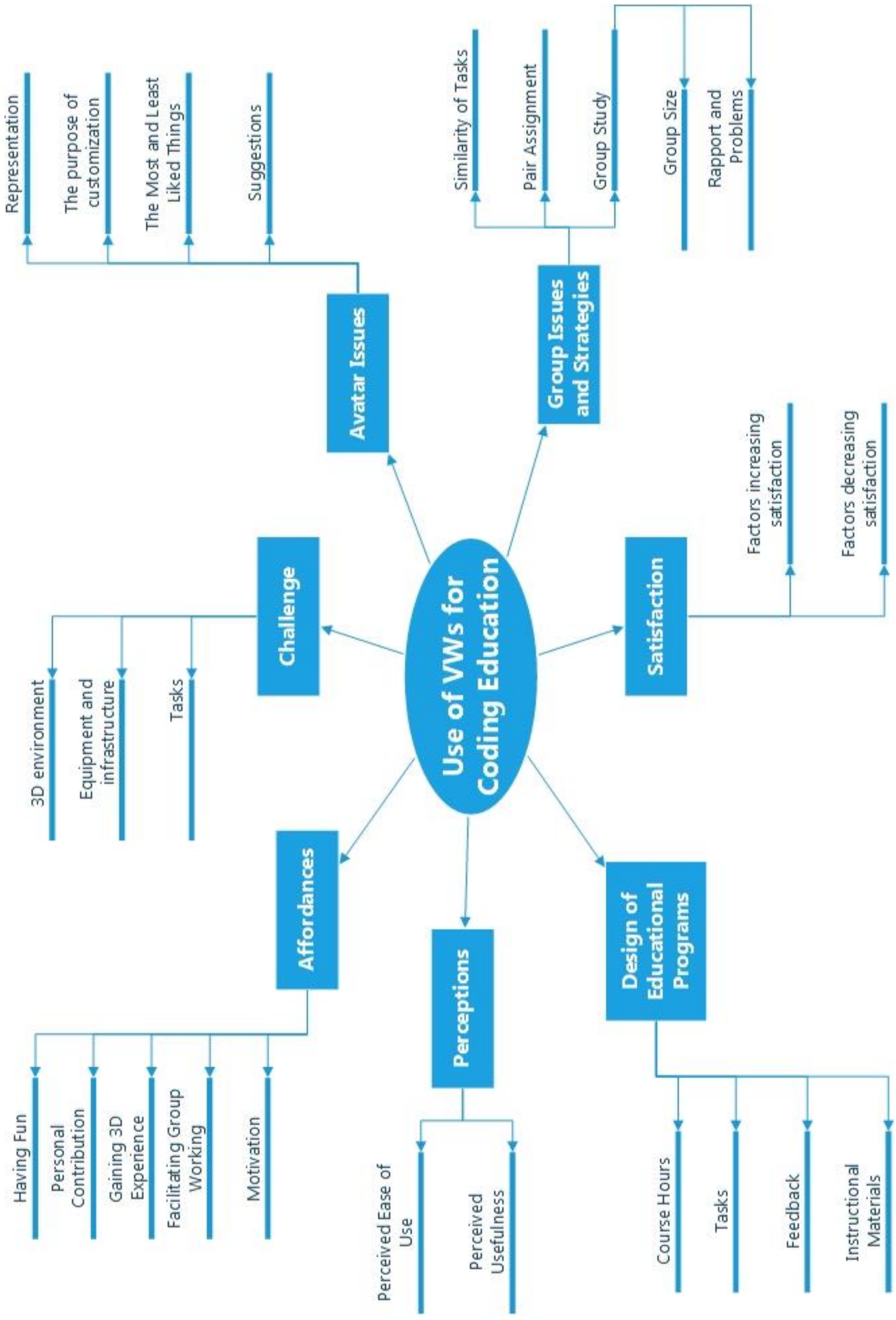


Figure 4.1 – Overall findings: Emerged themes, sub-themes and categories

The second sub-research question was about the affordances and challenges of using VW to teach the basics of programming. Results indicated that having fun during the learning process, personal contribution, gaining experience on 3D, facilitated group working and motivation were the emerged affordances in different proportions across the cases. The affordance only mentioned by the participants of Case-2 was to see real code being generated behind the pseudo-code of Scratch for OpenSim. On the other side, challenges related to the 3D environment, equipment and infrastructure and tasks were confronted by the students across the cases with different frequencies. Actually, challenges were specific to the participants and settings of the cases and some of them were inevitable in VW studies.

Other investigated issues and strategies were about the avatars. Most of the students in all three cases considered their avatars as a graphical representation of themselves in the virtual environment, and they customized their avatars for the purpose of (a) differentiation, (b) simulation to themselves, (c) looking funny and (d) disliking the clothes and default appearance of their avatars. Students' most-liked things about their avatars were multiple options, different movement modes and human-like features of the avatars. However, some of them considered the options as limited and stated them as their least-liked feature. Overall, the slow movement and style of the avatars were stated as the least-liked features of avatars. The suggestions of some students across the cases were that students would like to simulate real world behaviors to their avatars, have multiple options for customization and apply the characteristics of cartoon character to their avatar.

Another sub-research question was about group study in 3D environment. Apart from a few students in each case, most preferences about group study showed that the students preferred to study in groups. The ideal group size in the VW environment for the current study was found to be two students in each group. Moreover, qualitative results indicated that the students in all three cases generally had a good rapport within their pairs, except for in some situations. Another investigated issue was about pair assignment. Students' preferences about group study showed that they would like to be paired with someone according to their wishes, gender and level. Similarity of tasks was another important issue in group study. Results indicated that there should be some differences between the tasks of pairs.

Another sub-research question was about the satisfaction level of the participants and determining which factors affect their satisfaction across the cases. The descriptive results of the quantitative data shows that the most satisfied students were those from Case-1 (M = 4.35), Case-3 (M = 4.28) and Case-2 (M = 3.99), respectively. Qualitative findings revealed the factors increasing and decreasing satisfaction of the students. Apart from the story about the tasks, group study, object construction, having tasks, off-task activities, 3D environment and tour in the environment emerged as an increasing factor across all three cases at different ratios. However, story of the tasks was not stated as an increasing factor by the participants of Case-1. On the other side, technical problems, studying alone, difficulty of the tasks, and avatar-related problems were the factors stated by some of the students across all cases as decreasing factors. However, a less than realistic 3D environment was not found to be a decreasing factor for the students from Case-1, but was found to be the most decreasing factor in Case-2 and less so in Case-3. Finally, none of the students from Case-3 mentioned difficulty of use as a decreasing factor, while a few students from Case-1 and Case-2 found difficulty of use a decreasing factor.

The final sub-research question was about the design of educational programs across the cases. The first emerging sub-theme was weekly course hours. The students' and teachers' ideas about the weekly course hours varied across the cases. This was quite normal since the weekly course hours also varied across the cases, although the total hours for the course were almost the same. Cross-case analysis results showed that one and a half and two lesson hours were deemed insufficient, and that three lesson hour sessions with necessary comfort breaks should be arranged each week. Another investigated issue was about the tasks. The teachers and students found the tasks to be adequate to teach the basics of programming to their students. However, some students would like to have extra task activities. Arranging the difficulty level of activities is other issue. Apart from a few students, most students and the teachers of all cases mentioned that the difficulty level of the tasks was neither too hard or too easy, agreeing that they were generally moderate. The mutual feedback sources for the three cases were videos and the teachers. Students mostly preferred obtaining feedback from the videos, followed by feedback from their teachers. The only difference in the feedback source was seen for Case-2. The last issue investigated was about the resources. The students and teachers from all three cases usually found the task cards and information presentation tools in the 3D environment as helpful. They also offered some suggestions about them.

CHAPTER 5

DISCUSSION AND CONCLUSION

The results of the study were presented in line with the sub-research questions of the study in the previous chapter. In this current chapter, the findings of each sub-research question are discussed in light of the literature in the same order as presented in the previous chapter. This chapter actually is organized as per the findings illustrated in Figure 4.1 provided in the last of section of the previous chapter. Firstly, the perception of students with regard to SDP is presented. Then, the affordances and challenges of SDP in programming education are discussed. Thirdly, avatar- and group-related issues are elaborated on; then, satisfaction, and issues and strategies for the design of educational programs are discussed in light of the relevant literature. Implications of the findings and recommendations for further research are then addressed in the final section.

5.1 Perceptions about SDP

5.1.1 Perceived ease of use of SDP

Quantitative results related to the perception of students about the ease of use of SDP indicated that students of Case-2 and Case-3 used the programs without many difficulties. Students of Case-1 did seem to experience some difficulties, which could have stemmed from the fact that they did not play 3D games as much as the students from the other two cases. Previous research suggests that expertise in 3D gaming is an enabler for adapting the abilities to the 3D environment such as the movement of an avatar and interacting within the environment (Crellin et al., 2009).

The mean score of the item related to clearness and comprehensivity of the interaction with the programs was lower than four points in Case-2. Qualitative results indicated the reason for this could be that some terms on the interface of the programs were either in English or their translation to Turkish could have been meaningless to the students. These issues could cause some persistent uncertainty in students' minds if they did not have a clear understanding (Esteves et al., 2009). The adaptation process for using the 3D environment and S4OS could take some time. For example, the inventory, positioning objects in the 3D environment and building code on S4OS and transferring them to the 3D environment were difficult for the first time for most students across all three cases. Students need some time to understand the concepts; in time, they can gradually become accustomed to using the programs (Pellas, 2014a). After the necessary adaptation process, students could make use of the affordances of the environment in a real manner and feel a sense of empowerment (Rosenbaum, 2008) by creating meaningful artefacts (Girvan et al., 2013).

Students at any age wonder about the 3D environment and they try to push the limits of the environment. In the current study, regardless of cases, avatars of some of the students became jammed between objects and they also tried to go outside of the region. These difficulties occurred due to students' misuse of the environment which could have been prevented via limiting and reducing the 3D environment according to level of the students (Rico et al., 2011). Both students and teachers found the interface of the programs easy to use in general, which is parallel to the findings of similar studies conducted in similar 3D environments with K-12 students (Pellas, 2014b;

Pellas & Peroutseas, 2016; Rico et al., 2011) and students in higher education (Esteves et al., 2009; Girvan et al., 2013; Rosenbaum, 2008; Sajjanhar & Faulkner, 2014).

5.1.2 Perceived usefulness of SDP

Acceptance of a new technology as a learning platform is an important issue (Tokel & Topu, 2016). The results of the perceived usefulness questionnaire showed that students in all cases perceived VW as useful. Only the item related to perceiving VW as an enabler to communicate with the instructor scored low in all cases. This was not surprising because face-to-face interaction was the most preferred and occurred communication form throughout the courses in all cases. Besides, the item related to learning at one's own pace was moderately low in Case-2. A possible explanation for this might be that some students could not complete the activities because of their teammate. Previous research supports this explanation as students have diverse experience levels and they completed the tasks at their own pace at different times (Esteves et al., 2009). Thus, some of the pairs were unable to complete the tasks at the same time, and students needed to wait for each other, which seems to have occurred mostly in Case-2.

Qualitative results of the current study also support the questionnaire results. According to analysis of the interviews, using VW in programming education facilitated the learning process. This finding is supported by previous studies showing that students' learning in programming education is facilitated when 3D environments such as ALICE (Bishop-Clark et al., 2006), SL (Girvan et al., 2013; Hulsey et al., 2014; Jakoš & Verber, 2017; Seng & Edirisinghe, 2007), and OpenSim (Pellas, 2014b; Pellas & Vosinakis, 2017; Rico et al., 2011) are used. These findings suggest that VW could be used in teaching programming as a tool, which aims to reduce the difficulties of learning programming (Duncan et al., 2014; Kelleher & Pausch, 2005; Pears et al., 2007; Sauppé et al., 2015).

Here, there is a need to address the question of whether or not media affects learning, which has been an ongoing debate over the past decade between two scholars, Kozma and Clark. In this debate, while Clark (1983) acutely argued that media has no effect on learning, Kozma's (1991) approach to this contradiction was moderate and he argued that media might have an effect on the effectiveness of the method applied in an instructional design. Therefore, in another of his studies, Kozma (1994) argued that there is no need to distinguish whether media or method affects learning as suggested by Clark (1983); that is they could complement each other. Each new medium has special characteristics and attributes, which help present the information in a different way as each learner has specific characteristics to make use of each different medium (Kozma, 1991). For example, in teaching basics of programming, different programming environments and tools have been developed in order to reduce the difficulties of programming according to the level of the user or learner (Powers et al., 2006). The question of whether media affects learning should be avoided, rather it should be investigated how media could be used for educational purposes and how it could be best served for this purpose (Yazıcı & Kültür, 2013), which were the primary purposes of the current study.

Different features and components of VWs could be a driving force to facilitate learning. The literature review shows that visualization of programming code for children is an important issue in programming education, because code is an abstract concept and otherwise meaningless for children (Esteves, Antunes et al., 2008; Sajjanhar & Faulkner, 2014; Salleh et al., 2013). With the features of VWs providing visualization and visual feedback, they could facilitate students' learning process in programming education (Esteves et al., 2009; Girvan et al., 2013). In this way, VWs could create opportunities for learners to reflect on their learning process (Brennan, 2013) and thereby help students to understand the most difficult parts of the learning process (Tekdal, 2013).

Research has shown that pure visualization is insufficient to facilitate learning effectively (Naps et al., 2003). Learners need to construct and manipulate meaningful and shareable objects. At this point, VWs could provide learners with a wide range of compelling context for different types of activities that could attract students to the learning of programming (Delwiche, 2006; Esteves, Fonseca et al., 2008; Kahn, 2007; Pellas & Peroutseas, 2016). Contextualization is another important component in learning programming, which could be facilitated when students apply and transfer the knowledge of programming into real life problem situations rather than perceiving programming as just a computer-related activity (Esteves et al., 2006; Gomes & Mendes, 2007; Gülmez, 2009; Lukkarinen & Sorva, 2016). Since VWs could enable contextualization and increase the possibility of transferring what is learned into real life situations (Dalgarno & Lee, 2010; Duncan et al., 2012; Kahn, 2007; Kluge & Riley, 2008; Pellas & Peroutseas, 2016; Topu et al., 2017; Warburton, 2009), they might contribute to facilitate the learning of programming. Features of VWs such as an immersive 3D environment, enhanced interaction, engagement and facilitation of group study could be other contributing factors in facilitating the learning of programming (Esteves et al., 2009; Pellas & Peroutseas, 2016)

5.2 Affordances and Challenges of SDP

5.2.1 Affordances

Using VW in teaching basics of programming for children and revealing the potential affordances is one of the primary purpose of the current study. Previous studies have noted that VWs have many affordances when used for educational purposes (Dalgarno & Lee, 2010; Duncan et al., 2012; Kluge & Riley, 2008; Richter & Dawley, 2010; Topu et al., 2017; Warburton, 2009). Results of the current study have both consistent and inconsistent findings when compared to the literature. The affordances of the VW in programming education explored in the current study are interconnected with each other. The themes that were drawn from this study included having fun, personal contribution, gaining experience with 3D, facilitating group working, motivation. In the following part, the results are discussed in light of the literature.

Having fun

Results indicated having fun as one of the affordances of using VW in programming education. This result is consistent with previous studies showing that students had fun in the learning process when VWs were used in different domains such as geography education (Tüzün, Yilmaz-Soylu, Karakuş, Inal, & Kizilkaya, 2009), science education (Bakar-Çörez, 2011; Dieterle & Clarke, 2006) and programming education (Bishop-Clark et al., 2006; Crellin et al., 2009; Esteves et al., 2011; Pellas & Peroutseas, 2016). In addition, it has been indicated in previous studies that students created additional activities just for fun and that they would like to share them with their friends when programming education was integrated into VWs (Esteves et al., 2011; Girvan et al., 2013).

Having fun plays an important role in programming education also. A large body of research shows that programming lessons face some of the highest dropout rates (Mason et al., 2012; Robins et al., 2003) since learners can face difficulties in the learning of programming (Guzdial, 2004; Kelleher & Pausch, 2005; Saeli et al., 2011; Schulte & Carsten, 2013). The main reasons why students consider programming so difficult are highlighted in the literature as either students having been introduced to programming with a bad experience (Esteves et al., 2011), or that the activities used for the teaching of programming did not make sense to the students and were not related to their areas of interest (Blackwell, 2002; Maloney et al., 2008; Pears et al., 2007; Resnick, Maloney, Hernández et al., 2009). As a result, some students are therefore likely to find programming a rather boring activity (Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2014). At this point, having fun might increase the likelihood of children liking programming. In this way, it could be possible for students to develop a passion to learn programming (Resnick & Siegel, 2016), and thereby increase their participation in programming education.

Personal contribution

Personal contribution is another emerging affordance which has four sub-themes. Transferring the knowledge of programming into real life was one such sub-theme. Students thought that they could apply the knowledge acquired in their future life and that the knowledge could be base for them. The results of the current study are consistent with the study of Hulsey et al. (2014) in which novice female middle school students wished to improve their computing career prospects following programming education in VW. In this way, students would meet the requirements of a computing intensive world and be better prepared for their future careers (Knobelsdorf & Vahrenhold, 2013). Furthermore, they would be motivated to follow computed-related careers (Guzdial, 2015; Menekse, 2015). Interestingly, although it emerged as an affordance in all three cases of the current study, the percentage of students in Case-1 perceiving personal contribution as an affordance was higher than the others. There are several possible explanations for this. Case-1 was curricular and the students mostly consisted of those who were introduced to programming for the first time in this study. Students in Case-1 generally thought that they could use the knowledge acquired in computing-related jobs in their future. Students in other cases were more aware and had some prior knowledge of programming.

Another affordance was to learn game programming. Results across the cases showed that students of Case-2 and Case-3 defined this sub-theme as an affordance while no students in Case-1 mentioned it. The purpose of students in Case-2 and Case-3 in joining the current study was more to the point, which could be a possible explanation why no students in Case-1 mentioned the learning of game programming as an affordance. Based on this finding, students should be led to topics that they are most interested in such as game programming in teaching basics of programming as a way to increase their engagement in the learning process (Pellas & Peroutseas, 2016). In the literature, it was suggested that practicing the knowledge of computing in real-life situations and real-life problems could help students to learn better rather than just accruing knowledge (Esteves et al., 2009).

Other investigated affordances were realization of the mission of computers in a meaningful way and to enhance creativity and imagination. One student from Case-2 and also from Case-3 argued that they started to use computers in a more meaningful way since they previously only considered computers as a machine upon which to watch videos or play games before the current study. This interesting result supports the study of Resnick (2012), who argued that learning programming enables children to use computers to design, create, and express themselves in a more meaningful way. Enhancing creativity and imagination is in line with the literature. It was reported that creating artefacts in SL and programming them increased students' imagination (Crellin et al., 2009) and that programming could help students to develop some important skills (Akcaoglu & Koehler, 2014).

Gaining experience in 3D

Gaining 3D experience was considered as an affordance in each case. At least one student in all three cases argued that they gained experience in the use of 3D. The literature review showed that VWs with their built-in features provide learners with the ability to experience and build within an immersive 3D environment (Dawley & Dede, 2014; Messinger et al., 2009). Hulsey et al. (2014) argued that VW enables learners to experience a wide variety of experiences on 3D modeling and programming that other approaches and tools could not afford. For example, it was reported that creating 3D objects in Alice was difficult despite the fact that it was a 3D programming tool (Rodger et al., 2010). On the other hand, the easy to use built-in modeling capabilities of the VW provides novice learners with an environment in which to engage in and experience 3D (Cooper et al., 2009; Girvan et al., 2013). In terms of student experiences, 3D VWs coordinate a system of creating, resizing, rotating and integrating 3D objects, in addition to texture-mapping, which are invaluable experiences for learners.

Facilitating group working

In the current study, the students studied alongside a peer and completed tasks in 3D virtual environment together. Results indicate that facilitating group working was considered as another affordance of VW by most of the students across all three cases. The results of the current study are therefore in line with previous studies, which show that VW supports and facilitates group study with its distinctive characteristics (Choudhury & Banerjee, 2012; Dalgarno & Lee, 2010; Dickey, 2003; Duncan et al., 2012; Vosinakis et al., 2016). This emerged affordance plays a vital role in programming education because studying with a peer in programming education is strongly encouraged by many studies in the literature (Bishop-Clark et al., 2006; Buffum et al., 2015; Esteves et al., 2009; Guzdial, 2015; Hanks, Fitzgerald, McCauley, Murphy, & Zander, 2011; Kafai & Burke, 2014; Liebenberg, Mentz, & Breed, 2012). It should be promoted to teach programming in a community where students can reflect on their communal practice and that children should learn collaborative working along with their learning of programming (Kafai & Burke, 2014). Issues and strategies about group study is discussed in the following parts broadly.

Motivation

Motivation was stated as an affordance of VWs in programming education by the teachers of cases. They argued that using VW positively affects student motivation towards programming. Motivation is an important component for programming education since lack of motivation causes students to give up learning programming and to think that programming is difficult (Gomes & Mendes, 2007; Kurhila & Vihavainen, 2015; Papadopoulos & Tegos, 2012; Selby, 2015). In fact, many programming tools aims to motivate students (Bishop-Clark et al., 2006; Ingram-Goble, 2013). The findings of current study are in line with previous studies which show that using 3D tools for teaching programming has a positive effect on students' motivation such as Alice (Pausch et al., 2000), OpenSim (Pellas, 2014b; Rico et al., 2011), and SL (Hulsey et al., 2014; Pellas & Peroutseas, 2016; Pellas & Vosinakis, 2017).

The literature review shows that factors affecting motivation for learning programming in 3D environment were multifaceted. It was found in previous studies that features of VWs such as the immersive 3D environment (Dreher et al., 2009), contextualization and transferring what was learned into real life (Esteves et al., 2009), adding fun to the learning process (Papadopoulos & Tegos, 2012), constructing things by programming (Hulsey et al., 2014), receiving immediate feedback in 3D format (Rico et al., 2011), and the game-based learning environment (Cooper et al., 2009) all have an effect on learner motivation. The qualitative results of the teacher interviews indicated that the use of a compelling 3D environment, the presence of peers in the virtual environment, and having fun were all sources of motivation for the students. These findings should be enriched and need to be corroborated by asking the ideas of students in other studies.

Other Affordances

Writing code in a professional programming language is difficult for novice students. On the other hand, it could be helpful for students to realize the real code behind the pseudocode of a block-based programming tool such as Scratch as the learners' expertise level increases. S4OS transforms the pseudocode into LSL of OpenSim. Students were able to see the code while transferring to the virtual environment. One of the students realized about the transformed code and stated that he was investigating it. The teacher in Case-2 also argued that seeing the code could facilitate student understanding of the real code that lies behind the pseudocode. There were different opinions of scholars on this issue found in the literature. Some scholars criticized block-based programming tools since they do not require learners to type actual code. For example, Hulsey et al. (2014) argues that students should write real code and see their syntax errors in text-based programming languages. On the other hand, Brennan

(2017) argues that those issues would not be necessary for novices, although they could be considered as the essence of programming experience. From the pragmatist view, learners should be informed about text-based languages as their expertise increases, which was also mentioned by the teacher of Case-2 in the current study.

5.2.2 Challenges

Challenges are inevitable when virtual environments are used for educational purposes due to their technical requirements (Dawley, 2009; Richter & Dawley, 2010), as seen in the current study. The intent and functionality of the technology, learner needs, ability and characteristics, and their interplay might have an effect on the challenges that could be come across (Dickey, 2011). Therefore, some of the challenges seen in the current study are in line with the literature, while some are specific to the participants and context of the current study. Revealing potential challenges and taking necessary precautions is essential because such challenges could impede on the seamless experience of the learners (de Freitas, Rebolledo-Mendez, Liarokapis, Magoulas, & Poulovassilis, 2010). In the following parts of this section, the emerging challenges are discussed in light of the available literature.

3D Environment

The first emerging challenge related to the 3D environment. There was no mutual challenges seen across in all three cases. For example, students of Case-1 mostly preferred not to watch the help videos, preferring instead to seek the help of their teachers. As a result, they did not encounter challenges related to the watching of the videos. Therefore, some challenges may not be obvious in some cases or perceived as a challenge by the students unless they came into contact with certain functionality. Challenges in this sub-theme were generally dependent on the ability of the learners. Leaners need some generic VW skills such as being able to control the camera, take objects, or customize their avatar. Previous studies have shown that some students even at the graduate level might experience difficulties in certain 3D VW-related skills (Girvan et al., 2013). It was reported that acquiring the relevant skills could be quite complex for the less technically skilled students (Crellin et al., 2009). Such students need to be taken into consideration and extra effort applied in order to help them in the adaptation period.

Besides, findings showed that some students negatively impacted their 3D environment such as building huge objects, or somehow damaging the creation of their peers, or misusing the objects. These kinds of improper user behavior were also reported in other studies too (Crellin et al., 2009; Pellas, 2014a); although they are reported as rare in the literature, which is similar to the findings of the current study. Students should be informed about the appropriate use of VWs and the results of their misuse in order to prevent such behaviors. Besides, some additional rules could be defined in order to prevent instances of undesirable user-oriented behaviors (Jakoš & Verber, 2017). Watching videos was another perceived challenge revealed in the current study. Watching videos in VWs was also reported as a problematic issue due to server and client-side lags the study of Choudhury and Banerjee (2012). The lags related to performance and server and client resources sometimes resulted in missing textures or frames on videos.

Interestingly, no students perceived the transferring of scripts from S4OS to the virtual environment as a challenge. This finding is inconsistent with the study of Girvan et al. (2013) in which it was found that using two separate programs could adversely affect students' experiences and engagement.

Equipment and Infrastructure

The complexity of the virtual environments brought about some technical challenges related to equipment and the infrastructure for both educators and the researcher whilst integrating them into the educational activities. Challenges related to computers and the network used were the most stated in the current study. It is a known fact that virtual environments require a high bandwidth connection (Crellin et al., 2009; Dreher et al., 2009). In some studies, it was suggested that multiple dedicated servers should be used rather than a single server in order to address issues of restrictive bandwidth (de Freitas et al., 2010; Dreher et al., 2009). In the current study, even though bandwidth problems dependent on the server side were minimized by using a special distribution of an all-in-one OpenSim server package, some problems were experienced in both Case-2 and Case-3 due to the network infrastructure of those settings.

Computer-related challenges emerged in all three cases. Previous studies indicated that VWs require computer hardware with a high level of performance, especially required are high-end graphic cards and high RAM levels for both server and client side (Choudhury & Banerjee, 2012; Cooper et al., 2009). Problems with these areas could lead to the crashing of computer hardware and/or software, as well as system lag and down time (Duncan et al., 2012), the slow loading of avatars and textures, or simply through user log-in problems (Choudhury & Banerjee, 2012). In the current study, the students were able to create multiple scripts containing any code in order to try the limits of the virtual environment such as inadvertently adding infinite loops to their code. Previous studies showed that such user behaviors could result in performance problems and server-side lag; and as a results, all residents logged into the server were affected (Choudhury & Banerjee, 2012). Other challenges were specific to cases such as inadequate light or power outage as seen in Case-1 and the absence of working headphones in Case-2. In their study, Dunleavy and Dede (2014) argued that challenges related to student safety and privacy could also be possible in VW-based studies. However, this was not mentioned as a challenge in the current study because the use of dedicated servers avoided the issues.

To summarize, the challenges seen in this sub-theme were generally out of the researcher's full control, and whilst the researcher could make certain improvements or take precautionary measures, it is not possible to prevent all tentative challenges.

Tasks

Students faced certain challenges related to their tasks. Grasping what a task was all about was the single emerged sub-theme that was seen across all three cases as a challenge. One possible explanation might be that some tasks, their stories and terms used may not make sense to some of the students. Based on this finding, tasks should be written that are comprehensible and meaningful for all students, as corroborated by the ideas of Resnick and Siegel (2016), who argued that meaningful tasks are an important driver to the developing of a passion towards programming in children. The difficulty level of some tasks were stated as a challenge by some students in Case-1 and Case-2. In her study, Brennan (2013) argued that children have "desires for challenging things to do" in programming activities (p. 81). On the other hand, Brennan also mentioned that all children became stuck in at least one of their programming activities and that they felt that those tasks were too difficult. Accordingly, it can be inferred that designing challenging activities may cause children to sometimes get into difficulties. Besides, reasons behind the status of getting stuck were stated as reluctance to invest time, negative feelings and loss of interest (Brennan, 2013). In the current study, insufficient evidence was found to mention why students became stuck.

Reaching the place or location of tasks in the 3D VW was the last challenge stated by students of both Case-1 and Case-2, which is in line with the results of Esteves et al. (2008). Tasks were scattered across a huge island in the VW, and there were three colored footprints showing the direction of tasks on the ground. Red and blue footprints brought the students to the red and blue tasks, respectively, whereas green footprints brought them to mutual landmarks. Students could ignore or confuse the meaning of the hallmarks and footprints. Additionally, there could be other factors affecting this issue, regardless of the signs given in the VW environment. Some students in Case-1 and Case-2 were less motivated to do the tasks when compared to the students in Case-3. On the other hand, students of Case-3 were less in number, and more motivated. When some of the students were left unattended, they moved around the environment randomly at will, and thereby could miss the correct location of the tasks. Similar findings were mentioned in the study of Cooper, Carroll, and Liu (2009), in which they suggested some methods in order to enforce students to follow specific paths such as placing invisible walls surrounding the paths in the 3D environment. According to the qualitative results of the current study, it was suggested that navigation through tasks could be achieved task by task or automatically; that is, when the learner completes a task, they navigate to the next task automatically.

5.3 Avatar Issues

Users develop an online identity with the help of avatars representing them in the 3D environment (Dalgarno & Lee, 2010). VWs offer a wide range of avatar types from perfect human-like clones to bizarre non-human fantasy characters (Crellin et al., 2009). Representation by avatar in the 3D environment is important in terms of many aspects such as sense of immersion (Dalgarno & Lee, 2010), identity and trust (Richter

& Dawley, 2010; Tokel & Cevizci-Karataş, 2014), interaction and collaboration (Hew & Cheung, 2010) and engagement (Feldon & Kafai, 2008).

Most of the students in all three cases considered their avatars as a graphical representation of themselves in the virtual environment, which accords with the findings of Feldon and Kafai (2008). Students built code for completing the tasks and interacted with others and objects with the help of their avatars. All students had the same avatars when they first logged in to the virtual environment. To increase their sense of representation, they were free to choose an avatar and customize its appearance at the beginning of the study and again later on in their free time. Previous research reports that students allocate a considerable amount of time and effort in the selection and customization of their avatars (Crellin et al., 2009; Kafai, 2008; Yee, 2006) which is consistent with findings of the current study. Feldon and Kafai (2008) investigated avatar-related activities of nearly 600 children aged 8-18 on Whyville based on multiple forms of data. The participants came from different settings, from a science class at a private school to after-school programs. Feldon and Kafai found that all of the participants changed their avatars at least once and at different ratios, and that avatar-related activities comprised one third of all activities in the virtual environment. One third of participants changed everything about their avatar's look. Others changed parts of their avatar except for their entire face, such as hair or clothing. In the current study, while the body type, skin color and accessories were rarely changed or not reported as changed by the students of all cases, the clothing and hair were reported as having been changed by students of all cases, which is similar to the findings of Feldon and Kafai (2008).

5.3.1 Purpose of customization

The customization of avatars is really important for students in order to feel represented with a unique identity (Hulsey et al., 2014) in the virtual environment, which has an effect on students' learning outcomes (Feldon & Kafai, 2008), participation (Kafai, 2008) , and their interaction with others (Messinger et al., 2009). Thus, it is better to provide multiple options for avatar appearance such as different types of clothing, and avatar designs (Bakar-Çörez, 2011). For example, there are thousands of avatar parts offered to Whyville users (Kafai, 2010). Although there were less in the current study, multiple options for avatars and their clothing were offered to students in order that they could customize their avatars.

Reasons for the customization of avatars have convergent points across the cases. The mutual reasons were for simulating avatars to resemble themselves and for differentiation. Actually, these two reasons are highly interrelated. When the changes were applied in order to simulate avatars to resemble themselves, then, differentiation was achieved anyway, and the opposite is also true. Therefore, it could be argued as all students' primary purpose of customization were for either differentiation or simulating resemblance to themselves. This finding is in line with previous studies in which participants make their avatar look similar to themselves (Cooper et al., 2009). In addition, Messinger et al. (2009) argued that very few participants simulated their avatar to resemble themselves very closely, or to be completely different from their appearance. They added that participants seem to make some features of their avatar more attractive than their real-life appearance. It was also observed that most of the students tended to behave in this way in the current study.

Other reasons were to look funny or due to their dislike of the clothes given to the default avatar. A few students from Case-2 and Case-3 customized their avatars in order to look funny. Previous research indicated that some students would like to draw other user's attention in the 3D environment and that they could use their appearance as a tool for accomplishing this (Feldon & Kafai, 2008). For example, in their study, Cooper et al. (2009) reported that some participants created some really crazy outfits for their avatars. Surprisingly, only the students of Case-1 did not mention this rationale, which may be due to the setting of Case-1 as more formal than in the other cases. Disliking the clothes of the default avatar was the last reason given. A possible explanation of this rationale could be that teens care about their avatar's appearance for others to admire or so not to be criticized themselves (Feldon & Kafai, 2008). Lastly, although the students of Case-3 changed their avatars' clothing, it was not provided as a reason for the changes made to their avatars.

5.3.2 Suggestions

Students' suggestions about the avatars were investigated. Students in all three cases simulated their real-world behaviors to their avatar in the virtual world. This preference is consistent with the study of Duncan, Miller, and Jiang (2012), in which it was stated that a more realistic environment was usually preferred. A virtual environment that was close to real life could contribute to the students' learning in terms of increasing their sense of immersiveness (Kim et al., 2012; Tokel & Cevizci-Karataş, 2014) and presence (Witmer & Singer, 1998). Therefore, students' suggestion about this issue should be taken into consideration while designing virtual worlds.

Customizing avatars as desired is another important issue as previously mentioned. It would be better to provide multiple options for avatar appearance such as different types of avatars, body parts, clothing, and accessories. Limiting changes to avatars to some extent was also another suggestion made. In the current study, a few unexpected or undesirable situations occurred when students misbehaved due to a lack of authority imposed in the environment. In the literature, it was suggested that rather than applying limitations all of the time and for all, it could be helpful to apply restrictions for a limited time and to those who misuse the 3D environment (Crellin et al., 2009). The last suggestion was about being able to apply the characteristics of a cartoon character such as having the power of Hulk, or Superman's cloak, which is in line with the findings of Kafai (2008), who found that most students attached something they liked in real life to their avatars such as Matrix glass. These kinds of attachment could be offered to students in VWs, as well.

5.3.3 Most- and least-liked things

VWs could be considered as a stage where avatars represent the users (Kafai, Fields, & Cook, 2010). Thus, it could be important to reveal what students liked most and least about their avatars. Results indicated that while avatar and clothing options were stated by most of the students in all cases as the most-liked, they were considered limited by some students in Case-2 and Case-3. These findings indicated that some students had high expectations about avatar design, and they considered the available choices as limited. Similar to this finding, avatar customization in SL was stated as one of the least-liked aspects (de Freitas et al., 2010). Results surprisingly showed that the type of movement of the avatar (walk, run and teleport) was stated by most of the students in all cases as the most-liked feature of the avatars, while the slow movement of avatars was stated by a few of the students in all cases as the least-liked thing about avatars. On occasions, there were some lag experienced with the avatar's movement and appearance due to technical issues of the current study. Unexpected situations could be behind seemingly contradicting results, as de Freitas et al. (2010) argued "technical issues did significantly impede the users' seamless experience" (p. 80).

Human-like features of avatars were stated by the students of Case-1 and Case-2 as the most-liked feature, whereas no students in Case-3 stated it as the most-liked. It can therefore be inferred that students in Case-3 did not care about the features of their avatars. Some students in all three cases, however, disliked some issues regarding the style of the avatar, such as walking style or cross-eyed appearance. This mostly depended on the preferences of the users. The important point to be considered here is that users should be able to customize their avatar as desired.

5.4 Group Issues and Strategies

Although the format of group study could be changed, previous studies have shown that studying in a group with peers has an effect on the success, motivation, reflection, enjoyment, retention, confidence, and assistance, as well as bringing about some problems (Esteves et al., 2011; Guzdial et al., 1996; Hanks, 2008; Hanks et al., 2011; Liebenberg et al., 2012). The literature review shows that VWs with their distinctive features allow learners to study in groups (Dalgarno & Lee, 2010; Duncan et al., 2012; Richter & Dawley, 2010). Issues and strategies about group study in VW found in the current study are discussed in light of the literature in the following part.

5.4.1 Group study

Programming has a bad public image as a solitary activity performed by socially "awkward" people (Brennan, 2013; Rosenbaum, 2008). In contrast to this image, it should be seen as a communal practice (Kafai & Burke, 2014) as in the current study, where most of the students studied with one of their peers in the virtual environment. Students' preferences about group study showed that most of them preferred to study in groups. These findings are in line with the previous studies (e.g. Hanks, 2008; Hanks et al., 2011; Liebenberg et al., 2012), in which most of the students enjoy and prefer studying with a peer.

As to students who would like to study alone, only one student was found in each case. Similar to this finding, Liebenberg et al. (2012) reported that a small number of students in their study preferred to study alone, but no reason for this was stated. Findings of the current study suggest that they could be concerned with the students' characteristics and settings. Enforcing students to study with peers might result in some negative issues. Therefore, it is better to take students' preferences into consideration as whether to study alone or within a group.

There is sufficient evidence in the literature that shows that group study has a positive effect on students' learning and satisfaction (Sajjanhar & Faulkner, 2014), reflection (Esteves et al., 2011), success and confidence (Hanks, 2008), engagement to perform activities (Pellas & Peroutseas, 2016), and enjoyment (Buffum et al., 2015) in programming education. Accordingly, it is seen as an effective approach for teaching programming (Guzdial et al., 1996). It should be reported that the method employed might have an effect on group study. However, the use of VW with its distinctive features holds opportunities in terms of sustaining and enhancing group study too (Dreher et al., 2009; Vosinakis et al., 2016). The important point here is that the method was in accord with the media adopted for the current study because the instructional method had a relationship in designing a learning environment with a specific medium. This is based on Kozma's (1994) argument that interaction between method and media affects learning and other issues.

The most notable form of group study investigated in the computing literature is pair programming (Buffum et al., 2015). In pair programming, two peers study together to create solutions to problems by sharing a mutual computer (Bishop-Clark et al., 2006). In the current study, the students also studied in pairs in a similar way to pair programming. However, the procedures of pair programming were not exactly followed by the current study. For example, the students were not separated as "observer" or "driver," as in pair programming. On the other hand, there were some similar points between them, such as studying in pairs and sharing the responsibility of completing a number of tasks together within the 3D environment. In the current study, two students were responsible for solving a number of similar tasks together by using the features of the VW. Results indicated that students in all cases helped each other on issues concerned with tasks, 3D environment and code. When the amount of helping across the cases was investigated, students of case -2 seems to help each other most. These results could be due to fact that this case consisted of highest number of students.

Group size

Preferences and argumentations of most of the students in Case-2 and Case-3, and all teachers' ideas show that the ideal group size in the VW environment for the current study was two students. However, most of the students in Case-1 argued that it could be more than two, although they were not able to provide a logical reason for their preference. It is difficult to explain this result, but it might be related to the fact that they have known each other for more than four years and perhaps therefore would like to be in the same group. Argumentation of a student in Case-1 supports this; who stated that all the girls and boys in the class could be in two separate and large groups based on gender. In a previous study by Üçgül and Çağıltay (2014), they argued that deciding on the optimum number of students in a group study is related to various factors such as defining a responsibility for each member or the activities designed. The finding of the current study is consistent with the previous research which inclines struggling with crowded groups could be harder for the facilitator to follow their students' progress since the students could become bored in large group studies (Üçgül, 2012) and distracted easily by off-task activities in the virtual environment (Cooper et al., 2009; Pellas, 2014b; Sajjanhar & Faulkner, 2014).

Rapport and problems

Results indicated that the students in all three cases generally had a good rapport with their pairs, except for in some situations. Damaging the activities of the fellow peer was the most common problem encountered in this study. While working in the virtual environment together, students sometimes damaged the activities of their fellow peer by mistake. This problematic and undesired issue was also mentioned in the study of Hulsey et al. (2014), a study which aimed to teach introductory programming to middle school participants in a similar context to the current study. Based on this finding, students should be forewarned about this potential issue.

Distracting the teammate from their own tasks was another problematic issue reported, as stated by the students of Case-3. Previous research argues that distraction from the learning objectives is more probable in a 3D environment when compared to other mediums (Crellin et al., 2009). Another problematic issue was that the peers in Case-1 completed the tasks in different speeds, which could well hinder the effectiveness of a group study. The underlying reasons for this situation are linked to the pair assignment, which is discussed in the next part. Lastly, there is insufficient evidence to argue why the last two problematic issues occurred in one specific case and not in the others.

5.4.2 Pair assignment

Assigning well-matched group members is another fundamental issue associated with group studies (Buffum et al., 2016). Students in all three cases were generally paired-up with other students according to own their wishes. Results indicated that there were no examples of considerable disparity between the peer pairs, except for the pace of work issue mentioned in Case-1. Previous research shows that matching a slow performing student with a fast performing student resulted in frustration in terms of completing activities (Liebenberg et al., 2012) and team dynamics, which is in line with the findings of Case-1. Different strategies about pair assignment were suggested in previous studies such as assignment according to wishes (Buffum et al., 2015), by personality traits, similar levels of experience, and skill (Buffum et al., 2016; Hanks et al., 2011) and dedication (Liebenberg et al., 2012).

Results revealed similar preferences about the optimal groups across the three cases. Most of the students preferred to be paired with someone of their choice, and a few students preferred to be paired with a student of the same gender. Lastly, the teacher of Case-2 and one student in Case-3 argued that defining group members could be achieved according to the level of the student. Previous research suggests that the level of the students could be taken into consideration, preferably while assigning pairs since similarity between the level of pairs is mostly related to pair compatibility (Hanks et al., 2011). The findings of the current study also suggests that students' wishes as well as their gender for some situations should be taken into consideration with regard to this issue. The last suggestion was that the instructor/teacher should be open to reconsidering team profiles and to be ready and willing to change the members of incompatible pairs (Hanks et al., 2011).

5.4.3 Similarity of tasks

Another important issue was whether or not the tasks of the pairs should be identical. Students' preferences indicated that tasks assigned to grouped pairs should have some differences rather than be exactly the same. When the tasks are exactly the same, some members of the groups might not want to complete the tasks or they might just cheat and copy their team mate's code in order to complete the same task. This may result in an imbalanced group study, in which one pair completes and dominates the tasks while the other misleads and lacks the necessary learning experiences (Buffum et al., 2015). Previous research argues that granting pair the responsibility for the tasks is essential for an effective group study (Üçgül, 2012).

5.5 Satisfaction

Results of the students' satisfaction questionnaire across the three cases were almost all in excess of four points; with only the mean score in Case-2 slightly lower, but quite acceptable. One possible explanation for this might be that the class was more crowded and it was therefore more difficult for the teacher to deal with all the students. Another possible explanation might be that some of the students had high expectations from the club and using the VW might not have satisfied their expectations as a whole. Similar to this situation, the findings of Case-1 corroborate the second explanation because the students in Case-1 had low expectations from the current study and their satisfaction level was the highest. The overall results of each single case are consistent with previous studies, showing that the use of VW with its distinctive features has a positive effect on students' satisfaction in general (Hew & Cheung, 2010), and in programming education (Buffum et al., 2015; Girvan et al., 2013; Hulsey et al., 2014; Pellas & Kazanidis, 2014). Factors affecting the students' satisfaction are discussed in the following part. While designing VWs for programming education, these issues should be kept in mind.

5.5.1 Factors increasing satisfaction

Factors affecting students' satisfaction were revealed in the current study. Group study, object construction, tasks, their stories, off-task activities, the 3D environment and touring the environment were found as factors that increased the satisfaction level of the students. Group study and tasks were the most-cited in all three cases. Findings of the literature review are also in line with the findings of the current study. These results validate the findings of previous studies, arguing that having a partner and tasks that draw the attention of learners are factors associated with increasing the satisfaction of leaners (Bishop-Clark et al., 2006; Buffum et al., 2015; Crellin et al., 2009; Girvan et al., 2013; Rico et al., 2011).

There were no major differences among the factors except for the story of the tasks across the cases. The story of the tasks was not found as a factor in Case-1. Object construction, off-task activities, and the 3D environment were not mentioned as increasing factors by the students of Case-1 as much as in the other two cases. Differences across the cases may therefore be due to timetabling issues associated with Case-1, where students did not have so much free time to do any off-task activities or explore the 3D environment at will. Results of the single cases are therefore in line with the literature review, showing that generic features of VWs such as the ability to move around freely (Hew & Cheung, 2010), being able to create 3D objects and code them without difficulties (Girvan et al., 2013), and the existence of fun activities (Esteves et al., 2009) made the learning process enjoyable for the students. Additionally, "Working with peers on something meaningful is usually significantly more engaging and fun" (Berland, 2017, p. 140).

5.5.2 Factors decreasing satisfaction

Technical problems, studying alone, difficulty of tasks, and avatar-related problems were the least-liked factors stated by some students across all three cases and caused a decrease in the students' satisfaction. Previous studies found that technical problems were the biggest obstacle to student satisfaction in the use of VWs for educational purposes (Dawley & Dede, 2014; Hew & Cheung, 2010; Rosenbaum, 2008). The literature review shows that challenging tasks could lead students to Paper's (n.d.) "hard fun"; however, it might also lead them to frustration and getting stuck when the difficulty of tasks increased beyond their ability or comfort level (Brennan, 2013). The difficulty of tasks emerged as a factor that decreased satisfaction in the current study, which seems to be concerned with the latter argumentation of Brennan (2013). This finding suggests that there should be adequate support available for the students in order to prevent this situation from occurring.

Difficulty of use was also stated as a decreasing factor for student satisfaction. Some complex structures of the VW were mentioned in the first research question that could be the cause of this situation. Surprisingly, a less than realistic environment was not found as a factor decreasing satisfaction in Case-1, but it was found as the most decreasing factor in Case-2 and less so in Case-3. This shows that while students of Case-2 had high expectation about the realism of the 3D environment, students of Case-1 did not have any concerns about this issue.

5.6 Issues and Strategies for the Design of Educational Programs

The current study was implemented in three different educational programs. There were some similarities seen among the programs as well as some differences stemming from differences among the context and settings of the programs. Issues and strategies found in each educational program are discussed in light of the literature in this section.

5.6.1 Course hours

The settings of each case were different from each other. Case-1 and Case-2 were conducted in a school setting and were more formal than Case-3 which was conducted in an out-of-school setting as an after-school program. Although the total number of hours were the same, the weekly course hours were two lesson hours for Case-1, one and a half lesson hours for Case-2, and three lesson hours for Case-3.

Most of the students in Case-3 only stated that the weekly course hours were adequate. When the weekly course hours were insufficient, tasks could be left half-finished and students faced doing them all over again the following week. Moreover, arranging and maintaining the necessary programs ready for the course each week was a struggle since the VWs required extensive technical support (Choudhury & Banerjee, 2012). These cross-case analysis results may help to understand that one and a half and two lesson hours were insufficient, and it would be advisable to arrange three lesson hour sessions with necessary comfort breaks each week.

One interesting finding was that students of Case-1 found two lesson hours per week for the first part of the study and three lesson hours for the second part of the study to be insufficient. Students of Case-1 wanted to spend more time working in the VWs, which might be concerned with the fact that they liked learning programming in VWs and were more satisfied by it. This inference is in line with the findings of Hulsey et al. (2014) in which students asked for extra time to spend in the VW due to similar reasons. Another reason could be the fact that they liked the idea of learning programming instead of learning traditional ICT topics.

5.6.2 Tasks

The subject needs to be delivered to students in appropriate forms in accordance with the underlying instructional theory, GBS. Each programming concept was aimed to teach students via tasks in a cumulative way; thus, the number of tasks is important in order to present the content. As to the number of tasks covered in the VW environment, the teachers found them to be adequate to teach the basics of programming to their students. However, some of the students argued that the number of tasks was insufficient and that they would like to have additional tasks. The results indicated that at least one student in each case would like extra activities. This finding suggests that preparing additional task activities could be helpful for students, especially for those who are able to complete their activities ahead of their peers.

Tasks were designed from simple through to advanced. Except for a few tasks, their difficulty level were generally stated as moderate by most students and their teachers in all three cases. Arranging the difficulty level of activities is an important issue. As mentioned in the related part of “Factors decreasing satisfaction” in this chapter, challenging activities to some extent could lead students to have fun in line with Papert’s “hard fun” concept (Esteves et al., 2009). On the other hand, over-difficult activities could lead to students becoming stuck on an activity (Brennan, 2013). The latter situation could leave students feeling alone about how and what to do next, and lead them to abandon the task(s) or even the course as a result, unless the necessary support is not immediately obvious to them. The issue of feedback is discussed in the next section.

Each task had an interesting story relevant to the learning objectives for the students. In the story of the tasks, some of the characters were narrated such as a café owner, However, they were absent in the 3D environment, and most of the students suggested that those characters should be represented in the 3D environment too. In order to achieve this, it was suggested to use agents or NPCs which fulfill a pre-defined programmed activity continuously or triggered via an interaction (Kapp & O'Driscoll, 2010; Tüzün & Özdinç, 2016). Previous studies found the existence of NPCs and quality interaction between them and the learners would motivate the learners towards to the content (Dede, Clarke, Ketelhut, Nelson, & Bowman, 2005; Veletsianos, Heller, Overmyer, & Procter, 2010).

The final suggestion was to show an indicator of the completed tasks on the screen. Previous research suggests the use of a Heads-up Display (HUD) object, which is a 2D object located on one part of the viewer screen as a fixed bar graph which guides and keeps tracking information on the learners (Cooper et al., 2009). These issues should also be taken into consideration.

5.6.3 Feedback

Feedback is an essential component of any educational activity, and deemed helpful for students since it leads them from the actual level to a desired level (Ramaprasad, 1983). In parallel with GBS, the current study provided feedback to the students through coaches and expert stories. Teachers, the researcher and sometimes the peers were sources of real-time feedback. Moreover, a video clip showing the steps how an expert would perform each task was presented to the students in order to provide them with a form of instant feedback. Only the first type of feedback according to GBS, consequences of actions, was ambiguous for students in the current study.

Visualization tools enable learners to see the consequences of their coding in different formats (Jakos & Verber, 2016). Students could then see the consequences of their code in a visually rich and animated format in 3D VWs. Various scholars have reported similar findings that students were able to receive an obvious feedback by following the behavior of the object coded in the VW (Dreher et al., 2009; Esteves et al., 2011; Pellas, 2014b). Although this could be assumed as a first kind of feedback for learners in GBS, there were some missing points that also need to be addressed. Firstly, the findings of the current study indicated that it took some time for learners to realize that this indeed a form of feedback. Also, the students were not informed emotionally at the end of their actions in the virtual environment, and they were just expected to understand the consequences of action.

Most of the students' preferences for a source of feedback were generally their teachers and the expert videos. Only the students of Case-2 preferred to take feedback from their peers, which differed from the students in the other two cases. Physical existence of the teacher and the researcher as coaches made feedback possible for the students at any time. Some of the students in all three cases headed towards the other feedback source, which was the expert videos available to view in the 3D environment. This was for various reasons. Firstly, the coaches failed to satisfy all the students' instant feedback requests, especially when the course was particularly crowded. It was found in previous research that students could experience delays in receiving feedback, which is meaningless to them when the teacher is the only available source of feedback (Esteves et al., 2011). Secondly, it was reported in the literature that some students might not prefer to receive feedback from the teachers due to a feeling of embarrassment (Schank, 2002). Finally, the expert's videos ease of implementation and dispersion could affect the students' preference. The only difference among the cases in terms of feedback source was that some students in Case-2 preferred taking feedback from their peers. On occasion, the teacher and videos might be inadequate due to the high number of participants in the case. Besides, the findings of the single case analysis indicated that taking feedback from peers was easy and instant for the students.

Suggestions of students were about the use of NPCs as a feedback source, enhancements in video clips, and the existence of teachers in the virtual environment. Previous studies showed that NPCs could be used for giving real-time feedback in a virtual environment (Hew & Cheung, 2010; Holmes, 2007). In his study, Holmes (2007) developed an expert agent giving instructions to students in ActiveWorlds and it was found that the learners' understanding of procedural instructions were facilitated via these agents. This finding suggests that the idea of using agents as experts could be adapted to the application used in the current study. It could be seen as better than the expert videos since interaction in the video clips was limited. Another suggestion was that video clips could be made shorter and more to the point. It was found that the virtual presence of the teacher helped the students when they became stuck or had a problem (Sajjanhar & Faulkner, 2014). Therefore, the teacher should stay online for longer periods in the virtual environment as suggested by some of the students in the current study. The final suggestion made was to give students hints within the 3D environment about tasks. This suggestion should be taken into consideration because just informing learners about their actions and nothing more could lead to student frustration (Hsu, 2009).

5.6.4 Presentation of instructional materials

One of the components of GBS was to provide resources to students. Instructional materials related to activities were delivered to the students on boards as posters or videos within the virtual 3D environment, as well as being provided with hardcopy task cards. The results showed that some of the students did not view the boards very much. One possible explanation for this result might be that the boards in the virtual environment sometimes loaded slowly and they also did not attract the attention of the students. It can therefore be suggested that the boards should be made more attractive and load faster.

Another instructional material was the hardcopy task cards, which were distributed to each student before the lessons. They contained information about tasks. The findings showed that the task cards were helpful in terms of following the assigned tasks to be completed. Some of the students suggested that they could be made more brief and that they could also contain the code of each task. In parallel to this suggestion, a repository of code and resources could be used, as proposed by Esteves et al. (2009). These suggestions are important in that Gülbahar, Avcı, and Ergün (2012) argued that resources in GBS should be well-organized and easily accessible.

5.7 Implications of the Findings

The current study aimed to investigate the use of VWs in teaching basics of programming for children in different educational programs and find the similarities and differences between the programs. The following implications can be offered for educators and researchers based on the findings of the current study:

- This study showed that VWs with a *low floor* programming tool could be used in teaching basics of programming to children in different educational programs. Children used such programs without any major difficulty. The multiuser nature of VWs, code visualization in 3D format, contextualization of what was learned in real life, and providing multifaceted feedback allow children to better understand programming.
- The findings of this study found important affordances that traditional approaches lack in teaching programming such as having fun, motivation, and facilitating group study, and specific affordances to VWs such as gaining experience on 3D and game programming when VW was used in programming education.
- Encountered challenges could impede on the students' experience. Especially computing requirements for both server and client side are important in terms of sustaining the virtual learning environment. The use of VWs requires additional time and effort to make best use of them. Other issues happen out of the researcher's control, so practitioners and researchers should be pre-informed and take necessary precautions before any implementation.
- Findings showed how particular educational programs could be better designed, and how issues and strategies concerned with activities, course hours, and feedback affect each program. Educators and researchers alike should consider and take note of them.
- Factors increasing and decreasing satisfaction were revealed for each program, which are important for practitioners aiming to teach programming in VWs.
- Children as well as parents might assume VW as a 3D game played just for fun and children could easily be distracted in the pervasive 3D environment. Practitioners need to be aware of such risks.

- Issues and strategies for group study in VW were defined for each program. Studying with peers is important to teach basics of programming for children, and these points could provide a basis to better understanding the use of VWs in group studies. Besides, the multiuser nature of VWs can bring about new opportunities for pair programming.
- Participants' expressed that they liked being represented by avatars. However, children like to significantly customize their avatars. They would also like some activities to be made easier, such as taking objects, and using the inventory. Practitioners could provide many more options for avatar enhancement by users.
- Similarities and differences among educational programs are important for educators, school management and other stakeholders wanting to teach programming to children. This study provides important points for each educational programs that educators and other stakeholders could benefit from when considering using VW in their programming education.

5.8 Recommendations for Further Research

In this section, recommendations for further research are addressed based on the results and limitations of the study:

- Each case of this study was conducted in a physical environment, with students participating in virtual learning activities within a computer laboratory. They interacted with each other in both the real world and the virtual world. Real world interaction could affect children's virtual world interaction and this may cause some lack of understanding of VW in programming. For example, face-to-face interaction was the most preferred and occurred form of communication throughout in all three cases, whereas other communication channels were barely used. Therefore, similar studies could be conducted with participants physically separated by location by investigating the dynamics of the virtual environment.
- This study aims to teach the basics of programming to children. Further studies could investigate the use of VWs in teaching different levels of programming such as advanced level of programming with different activities. Furthermore,

instead of S4OS, other programming tools that yield the ability of creating complex code could be used, with a wider range of complex activities. For advanced level of programming, VWs' own language could be used rather than using a low-floor programming tool. It might then be possible for researchers to understand whether only VW could be used in programming education.

- Multiple dedicated servers were used in the current study. However, only one dedicated server could be used by eliminating bandwidth and server problems. In this way, more students could participate in a study at their own convenience. Server records such as avatar activities and chat records could be employed as a valuable data source for further research studies. For example, distracting points in the virtual environment could be easily detected which currently distract children from concentrating on learning activities in the 3D environment. Findings could even be corroborated with server records in further research.
- The OpenSim application was used in the current study as a VW. Other virtual world applications with different features could be investigated; for example, some VW applications allow users to share screens, which could be important in group study and pair programming. Further research could be conducted with applications with such features to investigate their effects on programming education.
- Further research can investigate the components of group study, and thereby bring about new opportunities for pair programming.
- NPCs and the interaction between them and users might be investigated as a source of feedback in similar studies. Factors affecting the effectiveness of using NPCs in programming education might also be studied in future research.
- Similar study that are fully online can be conducted with large number participants whom were selected via other sampling methods. Generalization can then be applied by employing interferential statistics to data obtained with valid instruments.
- Some affordances of using VW in programming education explored in the current study are interconnected. Similar studies could be conducted with different participants and the relationship among them explored.

- Unguided activities should be done fully online, and the existence and presence of the teacher kept at a minimal level in future studies. Time and places to play in the 3D environment could be increased to some extent since students should be free to build creative artefacts by playing with the available features of the environment and then to code them. When students are free to experiment playfully and then work on meaningful projects (Brennan, 2013), programming education would be more meaningful rather than following the traditional aims of teaching programming such as considering it solely as a means to getting a technical job (Resnick & Siegel, 2016).

REFERENCES

Ackermann, E. K. (2001). Piaget's Constructivism, Papert's Constructionism: What's the difference? *Future of Learning Group Publication*, *5*(3).

Akcaoglu, M. (2013). *Cogntive and motivational impacts of learning game design on middle school children*. *(Unpublished Doctoral Dissertation)*. Michigan State University, Michigan.

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development*, *62*(5), 583–600. https://doi.org/10.1007/s11423-014-9347-4

Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers & Education*, *75*(June), 72–81. https://doi.org/10.1016/j.compedu.2014.02.003

Antonacci, D. M., & Modaress, N. (2008). Envisioning the Educational Possibilities of User-Created Virtual Worlds. *AACE Journal*, *16*(2), 115–126.

Bakar-Çörez, A. (2011). *The Perception and Experiences of Students and Teachers in Formal and Informal Learning Settings That Uses Muves : Quest Atlantis Case*. *(Unpublished Doctoral Dissertation)*. Middle East Technical University, Ankara.

Barab, S. a., Hay, K. E., Barnett, M., & Squire, K. (2001). Constructing Virtual Worlds: Tracing the Historical Development of Learner Practices. *Cognition and Instruction*, *19*(1), 47–94. https://doi.org/10.1207/S1532690XCI1901_2

Bartle, R. A. (2009). From MUDs to MMORPGs: The History of Virtual Worlds. In *International Handbook of Internet Research* (pp. 23–39). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-9789-8_2

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The Qualitative Report*, *13*(4), 544–559.

Beck, K. (2000). *Extreme programming explained : embrace change*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. Retrieved from https://dl.acm.org/citation.cfm?id=318762

Bell, M. W. (2008). Toward a Definition of "Virtual Worlds." *Journal of Virtual Worlds Research*, *1*(1), 1–5. https://doi.org/10.4101/jvwr.v1i1.283

Berland, M. (2017). Constructionist Learning. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 139–141). Thousand Oaks, California: SAGE Publications, Inc. https://doi.org/10.4135/9781483385198.n59

Bishop-Clark, C., Courte, J., & Howard, E. V. (2006). Programming in Pairs with Alice to Improve Confidence, Enjoyment, and Achievement. *Journal of Educational Computing Research*, *34*(2), 213–228. https://doi.org/10.2190/CFKF-UGGC-JG1Q-7T40

Blackwell, A. F. (2002). What is Programming? In *14th workshop of the Psychology of Programming Interest Group* (pp. 204–218). MIT Press. Retrieved from www.ppig.org

Bogdan, R. C., & Biklen, S. K. (2007). *Qualitative research for education: An introduction to theory and methods*. Boston, MA, MA: Allyn and Bacon.

Boyer, D. M., Herro, D., & Gardner, C. M. (2014). Developing computational thinkers: A working example for using app programming with middle school students. In *American Educational Research Association (AERA) Conference*. Philadelphia, PA,.

Brennan, K. (2017). Computer Programming. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 123–126). Thousand Oaks, California: SAGE Publications, Inc. https://doi.org/10.4135/9781483385198.n55

Brennan, K. (2013). *Best of Both Worlds: Issues of Structure and Agency in Computational Creation, in and Out of School*. *(Unpublished Doctoral Dissertation)*. Massachusetts Institute of Technology,Cambridge, MA.

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart:The Resurgence of Computer Science in UK Schools. *ACM Transactions on Computing Education*, *14*(2), 1–22. https://doi.org/10.1145/2602484

Bruckman, A. S. (1997). *MOOSE Crossing : Construction , Community, and Learning in a Networked Virtual World for Kids*. *Unpublished Doctoral Dissertation*. Massachusetts Institute of Technology, Cambridge, MA.

Buffum, P. S., Frankosky, M., Boyer, K. E., Wiebe, E., Mott, B., & Lester, J. (2015). Leveraging collaboration to improve gender equity in a game-based learning environment for middle school computer science. In *2015 Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (pp. 1–8). IEEE. https://doi.org/10.1109/RESPECT.2015.7296496

Buffum, P. S., Frankosky, M., Carolina, N., Elizabeth, K., Wiebe, E. N., Mott, B. W., & Lester, J. C. (2016). Collaboration and Gender Equity in Game-Based Learning for Middle School Computer Science. *Computing in Science & Engineering*, (April 2016), 18–28. https://doi.org/10.1109/MCSE.2016.37

Choudhury, J., & Banerjee, I. (2012). Programming in Virtual Worlds for Educational Requirements: LSL Scripting and Environment Development Challenges.

International Journal of Computer Science Research and Application, *2*(3), 30–41.

Clark, R. E. (1983). Reconsidering research on learning from media. *Review of Educational Research*, *53*(4), 445–459.

Cooper, T., Carroll, S., Liu, C., Franklin, T., & Chelberg, D. (2009). Using the virtual world of second life to create educational games for real world middle school science classrooms. In G. Siemens & C. Fulford (Eds.), *Proceedings of world conference on educational multimedia, hyper- media and telecommunications '09* (pp. 2124–2133). Chesapeake, VA: AACE.

Crellin, J., Duke-Williams, E., Chandler, J., & Collinson, T. (2009). Virtual worlds in computing education. *Computer Science Education*, *19*(4), 315–334. https://doi.org/10.1080/08993400903384950

Creswell, J. W. (2007). *Qualitative Inquiry and Research Designs: Choosing Among Five Approaches* (2nd ed.). Thousand Oaks, CA: Sage Publications.

Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. *Educational Research* (4th ed., Vol. 4). Boston, MA: Pearson. https://doi.org/10.1017/CBO9781107415324.004

Creswell, J. W., & Miller, D. L. (2000). Determining Validity in Qualitative Inquiry. *Theory Into Practice*, *25*(3), 124–130. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2474845&tool=pmcentrez&rendertype=abstract

Dalgarno, B., & Lee, M. J. W. (2010). What are the learning affordances of 3-D virtual environments? *British Journal of Educational Technology*, *41*(1), 10–32. https://doi.org/10.1111/j.1467-8535.2009.01038.x

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, *13*(3), 319. https://doi.org/10.2307/249008

Dawley, L. (2009). Social network knowledge construction: emerging virtual world pedagogy. *On the Horizon*, *17*(2), 109–121. https://doi.org/10.1108/10748120910965494

Dawley, L., & Dede, C. (2014). Situated Learning in Virtual Worlds and Immersive Simulations. In J. M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop (Eds.), *Handbook of Research on Educational Communications and Technology* (pp. 723–734). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-3185-5_58

de Freitas, S., Rebolledo-Mendez, G., Liarokapis, F., Magoulas, G., & Poulovassilis, A. (2010). Learning as immersive experiences: Using the four-dimensional framework for designing and evaluating immersive learning experiences in a virtual world. *British Journal of Educational Technology*, *41*(1), 69–85.

https://doi.org/10.1111/j.1467-8535.2009.01024.x

Dede, C. (2009). Immersive Interfaces for Engagement and Learning. *Science*, *323*(5910), 66–69. https://doi.org/10.1126/science.1167311

Dede, C., Clarke, J., Ketelhut, D., Nelson, B., & Bowman, C. (2005). Fostering motivation, learning, and transfer in multi-user virtual environments. In *American Education Research Association Conference*. Montreal, Canada, Canada.

Delwiche, A. (2006). Massively multiplayer online games (MMOs) in the new media classroom. *Educational Technology & Society*, *9*(3), 160–172.

Dickey, M. D. (2003). Teaching in 3D: Pedagogical Affordances and Constraints of 3D Virtual Worlds for Synchronous Distance Learning. *Distance Education*, *24*(1), 105–121. https://doi.org/10.1080/01587910303047

Dickey, M. D. (2005a). Brave new (interactive) worlds: A review of the design affordances and constraints of two 3D virtual worlds as interactive learning environments. *Interactive Learning Environments*, *13*(1–2), 121–137. https://doi.org/10.1080/10494820500173714

Dickey, M. D. (2005b). Three-dimensional virtual worlds and distance learning: two case studies of Active Worlds as a medium for distance education. *British Journal of Educational Technology*, *36*(3), 439–451.

Dieterle, E., & Clarke, J. (2006). Multi-User Virtual Environments for Teaching and Learning. In M. Pagani (Ed.), *Encyclopedia of Multimedia Technology and Networking* (2nd Editio, Vol. 2, pp. 1033–1041). Hershey, PA: Idea Group, Inc. Retrieved from http://muve.gse.harvard.edu/rivercityproject

Dreher, C., Reiners, T., Dreher, N., & Dreher, H. (2009). Virtual Worlds as a Context Suited for Information Systems Education : Discussion of Pedagogical Experience and Curriculum Design with Reference to Second Life. *Journal of Information Systems Education*, *20*(2), 211–225.

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? *Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14*, 60–69. https://doi.org/10.1145/2670757.2670774

Duncan, I., Miller, A., & Jiang, S. (2012). A taxonomy of virtual worlds usage in education. *British Journal of Educational Technology*, *43*(6), 949–964. https://doi.org/10.1111/j.1467-8535.2011.01263.x

Esteves, M., Antunes, R., Fonseca, B., Morgado, L., & Martins, P. (2008). Using Second Life in Programming's Communities of Practice. In R. O. Briggs, P. Antunes, G. J. de Vreede, & A. S. Read (Eds.), *Groupware: Design, Implementation, and Use. CRIWG 2008. Lecture Notes in Computer Science* (pp. 99–106). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-92831-7_9

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2008). Contextualization of programming learning: A virtual environment study. *38th Annual Frontiers in Education Conference*. Saratoga Springs, NY. https://doi.org/10.1109/FIE.2008.4720544

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2009). Using Second Life for Problem Based Learning in Computer Science Programming. *Journal of Virtual Worlds Research*, *2*(1), 23. https://doi.org/10.4101/jvwr.v2i1.419

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*, *42*(4), 624–637. https://doi.org/10.1111/j.1467-8535.2010.01056.x

Esteves, M., Morgado, L., Martins, P., & Fonseca, B. (2006). The use of Collaborative Virtual Environments to provide student ’ s contextualisation in programming. In *IV International Conference On Multimedia And Information And Communication Technologies In Education* (p. 283). Seville, Spain.

Feldon, D. F., & Kafai, Y. (2008). Mixed methods for mixed reality: understanding users’ avatar activities in virtual worlds. *Educational Technology Research and Development*, *56*(5–6), 575–593. https://doi.org/10.1007/s11423-007-9081-2

Fields, D. A., Kafai, Y., & Giang, M. T. (2017). Youth Computational Participation in the Wildd: Understanding Experience and Equity in Participating and Programming in the Online Scratch Community. *ACM Transactions on Computing Education*, *17*(3), 1–22. https://doi.org/10.1145/3123815

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education* (8th ed.). New York, NY: McGraw-Hill.

Franklin, D. (2015). Putting the computer science in computing education research. Communications of the ACM, 58(2), 34–36. http://doi.org/10.1145/2700376

Fredricks, J. A. (2017). Extracurricular Activities. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 270–272). Thousand Oaks, California: SAGE Publications, Inc. https://doi.org/10.4135/9781483385198.n106

Gal-Ezer, J., & Stephenson, C. (2014). A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. *ACM Transactions on Computing Education*, *14*(2), 1–18. https://doi.org/10.1145/2602483

Gal-Ezer, J., & Zur, E. (2013). What (else) should CS educators know? - Revisited. In *8th Workshop in Primary and Secondary Computing Education(WiPSE ’13)* (pp. 83–86). New York, NY, USA: ACM. https://doi.org/10.1145/2532748.2532757

Gamage, V., Tretiakov, A., & Crump, B. (2011). Teacher perceptions of learning affordances of multi-user virtual environments. *Computers & Education*, *57*(2011), 2406–2413. https://doi.org/10.1016/j.compedu.2011.06.015

Girvan, C. (2014). Constructionism , Creativity and Virtual Worlds. In *3rd International Constructionism Conference 2014* (pp. 367–377). Vienna, Austria.

Girvan, C., Tangney, B., & Savage, T. (2010). SLurtle Soup : a conceptual mash up of constructionist ideas and virtual worlds. In *Constructionism 2010* (pp. 1–10). Paris: France.

Girvan, C., Tangney, B., & Savage, T. (2013). SLurtles: Supporting constructionist learning in Second Life. *Computers & Education*, *61*, 115–132. https://doi.org/10.1016/j.compedu.2012.08.005

Gomes, A., & Mendes, A. J. N. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education* (pp. 1–5). Coimbra, Portugal. Retrieved from http://ineer.org/Events/ICEE2007/papers/411.pdf

Gross, P., & Powers, K. (2005). Evaluating assessments of novice programming environments. In *Proceedings of the 2005 international workshop on Computing education research - ICER '05* (pp. 99–110). New York, USA: ACM Press. https://doi.org/10.1145/1089786.1089796

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Gujberova, M., & Kalas, I. (2013). Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education on - WiPSE '13* (pp. 108–117). New York, New York, USA: ACM Press. https://doi.org/10.1145/2532748.2532750

Gustafson, J., & Branch, R. M. (2002). What is instructional design. In R. A. Reiser & J. A. Dempsey (Eds.), *Trends and issues in instructional design and technology*. Upper Saddle River , New Jersey: Merrill/Prentice Hall.

Guzdial, M. (2004). Programming Environments for Novices. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 127–154). Lisse, The Netherlands: aylor & Francis.

Guzdial, M. (2015). Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*, *8*(6), 1–165. https://doi.org/10.2200/S00684ED1V01Y201511HCI033

Guzdial, M., & DiSalvo, B. (2013). Computing Education: Beyond the Classroom. *Computer*, *46*(9), 30–31. https://doi.org/10.1109/MC.2013.306

Guzdial, M., & Fisher, L. M. (2014). Teach the teachers, and contribute to humanity. *Communications of the ACM*, *57*(11), 10–11. https://doi.org/10.1145/2667107

Guzdial, M., & Guo, P. (2014). The difficulty of teaching programming languages, and the benefits of hands-on learning. *Communications of the ACM*, *57*(7), 10–

11. https://doi.org/10.1145/2617658

Guzdial, M., Kolodner, J., Hmelo, C., Narayanan, H., Carlson, D., Rappin, N., … Newstetter, W. (1996). Computer support for learning through complex problem solving. *Communications of the ACM, 39*(4), 43–45. https://doi.org/10.1145/227210.227600

Gülbahar, Y., Avcı, Ü., & Ergün, E. (2012). Learning by Doing: An Example Application of “Goal-Based Scenario” Approach. *Education and Science, 37*(165), 274–291.

Gülmez, I. (2009). *Effects of using visualization tools in programming instruction on student success and motivation. (Unpublished Master Thesis)*. Marmara University, Istanbul.

Hanks, B. (2008). Problems encountered by novice pair programmers. *Journal on Educational Resources in Computing*, *7*(4), 1–13. https://doi.org/10.1145/1316450.1316452

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: a literature review. *Computer Science Education, 21*(2), 135–173. https://doi.org/10.1080/08993408.2011.579808

Herro, D., Gardner, C. M., & Boyer, D. M. (2015). Perceptions of Coding with MIT App Inventor: Pathways for their Future. *Journal for Computing Teachers,* (Winter), 30–40.

Hew, K. F., & Cheung, W. S. (2010). Use of three-dimensional (3-D) immersive virtual worlds in K-12 and higher education settings: A review of the research. *British Journal of Educational Technology*, *41*(1), 33–55. https://doi.org/10.1111/j.1467-8535.2008.00900.x

Holmes, J. (2007). Designing agents to support learning by explaining. *Computers & Education*, *48*(4), 523–547. https://doi.org/10.1016/j.compedu.2005.02.007

Hsu, C.-Y. (2009). *Formative Research on an Instructional Design Model for the Design of Computer Simulation for Teaching Statistical Concepts (Unpublished Doctoral Dissertation)*. Ohio University, Ohio.

Hsu, C.-Y., & Moore, D. R. (2010). An Example Implementation of Schank’s Goal-Based Scenarios. *TechTrends*, *54*(1), 58–61. https://doi.org/10.1007/s11528-009-0364-9

Hulsey, C., Pence, T. B., & Hodges, L. F. (2014). Camp CyberGirls: Using a Virtual World to Introduce Computing Concepts to Middle School Girls. In *Proceedings of the 45th ACM technical symposium on Computer science education - SIGCSE ’14* (pp. 331–336). New York, USA: ACM Press. https://doi.org/10.1145/2538862.2538881

Ingram-Goble, A. (2013). *Playable stories: making programming and 3d role-playing* *game Design personally and socially relevant*. *(Unpublished Doctoral Dissertation)*. Indiana University, Indianapolis.

ISTE. (2015). Computational thinking for all. Retrieved September 28, 2015, from https://www.iste.org/explore/articleDetail?articleid=152&category=Solutions&article=Computational-thinking-for-all

Jakoš, F., & Verber, D. (2017). Learning Basic Programing Skills With Educational Games:A Case of Primary Schools in Slovenia. *Journal of Educational Computing Research*, *55*(5), 673–698. https://doi.org/10.1177/0735633116680219

Kafai, Y. (2008). Gender Play in a Tween Gaming Club. In Y. Kafai, C. Heeter, J. Denner, & J. Sun (Eds.), *Beyond Barbie and Mortal Kombat: New perspectives on gender and gaming* (pp. 111–124). Cambridge, MA: MIT Press. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.469.4171&rep=rep1&type=pdf

Kafai, Y. (2010). World of Whyville: An Introduction to Tween Virtual Life. *Games and Culture*, *5*(1), 3–22. https://doi.org/10.1177/1555412009351264

Kafai, Y., & Burke, Q. (2013a). Computer programming goes back to school: learning programming introduces students to solving problems, designing applications, and making connections online. *Phi Delta Kappan*, *95*(1).

Kafai, Y., & Burke, Q. (2013b). The Social Turn in K-12 Programming : Moving from Computational Thinking to Computational Participation. In *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13* (p. 603). New York, New York, USA: ACM Press. https://doi.org/10.1145/2445196.2445373

Kafai, Y., & Burke, Q. (2014). *Connected code : why children need to learn programming*. MIT Press.

Kafai, Y., Fields, D. A., & Cook, M. S. (2010). Your Second Selves: Avatar Designs and Identity Play. *Games and Culture (Special Issue)*, *5*(1), 23–42.

Kafai, Y., & Resnick, M. (1996). *Constructionism in Practice: Designing, Thinking, and Learning in A Digital World*. Lawrence Erlbaum Associates, Inc.

Kahn, K. (2007). Should LOGO keep going FORWARD 1? *Informatics in Education*, *6*(2), 307–320.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, *52*, 200–210. https://doi.org/10.1016/j.chb.2015.05.047

Kalelioğlu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills. *Informatics in Education*, *13*(1), 33–50.

Kalelioğlu, F. & Keskinkılıç, F. (2017). Bilgisayar Bilimi Eğitimi için Öğretim Yöntemleri. In Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (1. Baskı, pp. 155–182). Ankara.

Kapp, K. M., & O'Driscoll, T. (2010). *Learning in 3D: Adding a new dimension to enterprise learning and collaboration*. San Francisco, CA: Pfeiffer.

Kelleher, C., & Pausch, R. (2005). Lowering the Barriers to Programming : a survey of programming environments and languages for novice programmers. *ACM Computing Surveys*, *37*(2), 83–137. https://doi.org/10.1145/1089733.1089734

Kılıç, E. (2009). *The effects of cognitive load in learning from goal based scenario designed multimedia learning environment for learners having different working memory capacities*. *(Unpublished Doctoral Dissertation)*. Middle East Technical University, Ankara.

Kılıç, E., & Yıldırım, Z. (2012). Cognitive Load and Goal Based Scenario Centered 3D Multimedia Learning Environment: Learners' Motivation, Satisfaction and Mental Effort. *Journal of Educational Computing Research*, *47*(3), 329–349. https://doi.org/10.2190/EC.47.3.e

Kim, S. H., Lee, J., & Thomas, M. K. (2012). Between Purpose and Method : A Review of Educational Research on 3D Virtual Worlds. *Journal of Virtual Worlds Research*, *5*(1), 1–18.

Kluge, S., & Riley, L. (2008). Teaching in Virtual Worlds: Opportunities and Challenges. In *Issues in Informing Science and Information Technology* (Vol. 5). Retrieved from http://proceedings.informingscience.org/InSITE2008/IISITv5p127-135Kluge459.pdf

Knobelsdorf, M., & Vahrenhold, J. (2013). Addressing the Full Range of Students: Challenges in K-12 Computer Science Education. *Computer*, *46*(9), 32–37. https://doi.org/10.1109/MC.2013.263

Kolodner, J. L. (1994). Introduction. *Journal of the Learning Sciences*, *3*(4), 303–304. https://doi.org/10.1207/s15327809jls0304_1

Kozma, R. B. (1991). Learning with media. *Review of Educational Research*, *61*(2), 179–211.

Kozma, R. B. (1994). Will media influence learning? Reframing the debate. *Educational Technology Research and Development*, *42*(2), 7–19.

Kölling, M. (2010). The Greenfoot Programming Environment. *ACM Transactions on Computing Education*, *10*(4), 1–21. https://doi.org/10.1145/1868358.1868361

Krishnamurthi, A. (2017). After-School STEM Programming. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 26–29). Thousand Oaks, California: SAGE Publications, Inc.

https://doi.org/10.4135/9781483385198.n15

Kunkle, W. M. (2010). *The Impact of Different Teaching Approaches and Languages on Student Learning of Introductory Programming Concepts*. *(Unpublished Doctoral Dissertation)*. Drexel University, Philadelphia.

Kurhila, J., & Vihavainen, A. (2015). A Purposeful MOOC to Alleviate Insufficient CS Education in Finnish Schools. *ACM Transactions on Computing Education, 15*(2), 1–18. https://doi.org/10.1145/2716314

Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education, 22*(3), 219–236. https://doi.org/10.1080/08993408.2012.713180

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA, CA: Sage Publications.

Lukkarinen, A., & Sorva, J. (2016). Classifying the tools of contextualized programming education and forms of media computation. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research - Koli Calling '16* (pp. 51–60). New York, New York, USA: ACM Press. https://doi.org/10.1145/2999541.2999551

Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08)* (pp. 367–371). Portland, Oregon. https://doi.org/10.1145/1352135.1352260

Mason, R., Cooper, G., & Raadt, M. De. (2012). Trends in Introductory Programming Courses in Australian Universities – Languages , Environments and Pedagogy. In *14th Australasian Computing Education Conference* (pp. 33–42). Darlinghurst, Australia.

McWhorter, W. I. (2008). *The effectiveness of using lego mindstorms robotics activities to influence self-regulated learning ın a university introductory computer programming course*. *(Unpublished Doctoral Dissertation)*. University of North Texas, Texas.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education, 3408*(August), 1–26. https://doi.org/10.1080/08993408.2015.1111645

Merchant, G. (2017). Virtual Worlds. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 809–810). Thousand Oaks, California: SAGE Publications, Inc. https://doi.org/10.4135/9781483385198.n307

Merriam, S. B. (1998). *Qualitative research and case study applications in education*. San Francisco, CA: Jossey-Bass Publishers.

Merriam, S. B. (2009). *Qualitative research: A guide to design and implementation*. San Francisco, CA: Jossey-Bass.

Messinger, P. R., Stroulia, E., Lyons, K., Bone, M., Niu, R. H., Smirnov, K., & Perelgut, S. (2009). Virtual worlds — past, present, and future: New directions in social computing. *Decision Support Systems*, *47*(3), 204–228. https://doi.org/10.1016/j.dss.2009.02.014

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Thousand Oaks, CA: Sage Publications.

Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). Thousand Oaks, CA: SAGE Publications.

MoNE-BoE. (2012). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı*. *T.C. Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığı*. Retrieved from http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196

MoNE-BoE. (2016). *Bilgisayar Bilimi Dersi Öğretim Program*. *T.C. Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığı*. Retrieved from http://www.eba.gov.tr/ekitap?icerik-id=4370

Moore, M. G. (1989). Three Types of Interaction. *American Journal of Distance Education*, *3*(2). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=C049510C7A4D7ED11C7C4CE92D2A7DFB?doi=10.1.1.491.4800&rep=rep1&type=pdf

Naps, T. L., Rodger, S., Velázquez-Iturbide, J. Á., Rößling, G., Almstrum, V., Dann, W., … McNally, M. (2003). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, *35*(2), 131. https://doi.org/10.1145/782941.782998

Neale, P., Thapa, S., & Boyce, C. (2006). Preparing a case study : A guide for designing and conducting a case study for evaluation input. *Pathfinder International Tool Series*, (May), 1–12.

Omale, N. (2010). *Exploring the use of 3-D multi-user virtual environments for online problem-based learning. (Unpublished Doctoral Dissertation).* Northen Illinois University, Illinois.

Omale, N., Hung, W. C., Luetkehans, L., & Cooke-Plagwitz, J. (2009). Learning in 3-D multiuser virtual environments: Exploring the use of unique 3-D attributes for online problem-based learning. *British Journal of Educational Technology, 40*(3), 480–495. https://doi.org/10.1111/j.1467-8535.2009.00941.x

Özmen, B., & Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles Üniversite Öğrencilerinin Programlama Deneyimleri: Güçlükler ve Engeller. *Turkish Online Journal of Qualitative Inquiry*, *5*(3), 9–27. https://doi.org/10.17569/tojqi.20328

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice Programming Environments. Scratch & App Inventor: a first comparison. In *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments - IDEE ’14* (pp. 1–7). New York, USA: ACM Press. https://doi.org/10.1145/2643604.2643613

Papadopoulos, Y., & Tegos, S. (2012). Using Microworlds to Introduce Programming to Novices. In *16th Panhellenic Conference on Informatics* (pp. 180–185). Piraeus, Greece. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6377388

Papert, S. (n.d.). Hard fun. Retrieved December 1, 2017, from http://www.papert.org/articles/HardFun.html

Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas*. New York, NY, USA: Basic Books. Retrieved from https://dl.acm.org/citation.cfm?id=1095592

Patton, M. Q. (1990). *Qualitative evaluation and research methods, 2nd ed. Qualitative evaluation and research methods, 2nd ed.* BOOK, Thousand Oaks, CA, US: Sage Publications.

Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). Thousand Oaks, CA: Sage Publications.

Pausch, R., Dann, W., & Cooper, S. (2000). Alice : a 3-D Tool for Introductory Programming Concepts. *Journal of Computing Sciences in Colleges*, *15*(5), 107–116.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., … Paterson, J. (2007). A survey of literature on the teaching of introductory programming. In *Working group reports on ITiCSE on Innovation and technology in computer science education - ITiCSE-WGR ’07* (Vol. 39, p. 204). New York, USA: ACM Press. https://doi.org/10.1145/1345443.1345441

Pellas, N. (2014a). The development of a virtual learning platform for teaching concurrent program- ming languages in Secondary education: The use of Open Sim and Scratch4OS. *Journal of E-Learning and Knowledge Society*, *10*(1), 129–143.

Pellas, N. (2014b). The influence of computer self-efficacy, metacognitive self-regulation and self-esteem on student engagement in online learning programs: Evidence from the virtual world of Second Life. *Computers in Human Behavior*, *35*(January 2016), 157–170. https://doi.org/10.1016/j.chb.2014.02.048

Pellas, N. (2015). Conceptual foundations for interactive programming activities with the conjunction of Scratch4OS & Open Sim. In M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (Third Ed, pp. 1–12). Hershey, PA: IGI Global. https://doi.org/10.4018/978-1-4666-5888-2

Pellas, N., & Kazanidis, I. (2014). Online and hybrid university-level courses with the utilization of Second Life: Investigating the factors that predict student choice in Second Life supported online and hybrid university-level courses. *Computers in Human Behavior*, *40*, 31–43. https://doi.org/10.1016/j.chb.2014.07.047

Pellas, N., & Peroutseas, E. (2016). Gaming in Second Life via Scratch4SL:Engaging High School Students in Programming Courses. *Journal of Educational Computing Research*, *54*(1), 108–143. https://doi.org/10.1177/0735633115612785

Pellas, N., & Peroutseas, E. (2017). Leveraging Scratch4SL and Second Life to motivate high school students' participation in introductory programming courses: findings from a case study. *New Review of Hypermedia and Multimedia*, *23*(1), 51–79. https://doi.org/10.1080/13614568.2016.1152314

Pellas, N., & Vosinakis, S. (2017). How can a simulation game support the development of computational problem- solving strategies ? In *IEEE Global Engineering Education Conference (EDUCON 2017)*. Athens: Greece.

Petrakou, A. (2010). Interacting through avatars: Virtual worlds as a context for online education. *Computers & Education*, *54*(2010), 1020–1027. https://doi.org/10.1016/j.compedu.2009.10.007

Pita, S., & Pedro, L. (2012). Verbal and Non-Verbal Communication in Second Life. In N. Zagalo, L. Morgado, & A. Boa-Ventura (Eds.), *Virtual Worlds and Metaverse Platforms: New Communication and Identity Paradigms* (pp. 100–116). Hershey, PA: IGI Global. https://doi.org/10.4018/978-1-60960-854-5.ch007

Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in Introductory Programming: What Works? *Communications of the ACM*, *56*(8), 25–28. https://doi.org/10.1145/2492007

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K. J., Proulx, V., & Carlisle, M. (2006). Tools for Teaching Introductory Programming: What Works? *SIGCSE Bull.*, *38*(1), 560–561. https://doi.org/10.1145/1124706.1121514

Purbrick, J., & Lentczner, M. (2007). Second life:The World's Biggest Programming Environment. In *22nd ACM SIGPLAN conference on Object oriented programming systems and applications companion - OOPSLA '07* (p. 720). Montréal, Québec: ACM Press. https://doi.org/10.1145/1297846.1297849

Ramaprasad, A. (1983). On the definition of feedback. *Behavioral Science*, *28*(1), 4–13. https://doi.org/10.1002/bs.3830280103

Reisoğlu, I., Topu, B., Yılmaz, R., Karakuş Yılmaz, T., & Göktaş, Y. (2017). 3D virtual learning environments in education: a meta-review. *Asia Pacific Education Review*, *18*(1), 81–100. https://doi.org/10.1007/s12564-016-9467-0

Reisoğlu, İ. (2014). *3B Sanal Öğrenme Ortamlarinda Öğretimsel , Sosyal ve Bilişsel Buradalık*. *(Unpublished Doctoral Dissertation)*. Atatürk University, Erzurum.

Resnick, M. (2012). Reviving Papert ' s Dream. *Educational Technology*, *52*(4), 42–46.

Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for Everyone. *Communications of the ACM*, *52*(11), 60–67. Retrieved from http://web.media.mit.edu/~mres/scratch/scratch-cacm.pdf

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Resnick, M., & Siegel, D. (2016). A Different Approach to Coding: How kids are making and remaking themselves from Scratch. Retrieved April 7, 2017, from https://brightreads.com/a-different-approach-to-coding-d679b06d83a#.yjbcreqlw

Richter, J., & Dawley, L. (2010). Creating Context for educational research in Virtual Worlds: an invitation to Dialogue. *International Journal of Gaming and Computer-Mediated Simulations*, *2*(1), i–xi. Retrieved from https://pdfs.semanticscholar.org/aa75/a9305032aa68e2d6d5539b7bedcf64273946.pdf

Rico, M., Martínez-Muñoz, G., Alaman, X., Camacho, D., & Pulido, E. (2011). A programming experience of high school students in a virtual world platform. *International Journal of Engineering Education*, *27*(1), 1–9.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, *13*(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Rodger, S. H., Bashford, M., Dyck, L., Hayes, J., Liang, L., Nelson, D., & Qin, H. (2010). Enhancing K-12 Education With Alice Programming Adventures. In *Proceedings of the 15th Conference on Innovation and Technology in Computer Science Education - ITiCSE '10* (p. 234). Bilkent, Ankara. https://doi.org/10.1145/1822090.1822156

Rosenbaum, E. (2008). Scratch for Second Life. In S. V. Harrell (Ed.), *Proceedings of the International Conference on Learning Sciences – ICLS 2008* (pp. 383–391). Utrecht, The Netherlands. Retrieved from http://dl.acm.org/citation.cfm?id=1600042

Saeli, M., Perrenet, J., Jochems, W. M. G., Zwaneveld, B., Nederland, O. U., & Centrum, R. D. M. (2011). Teaching Programming in Secondary School : A Pedagogical Content Knowledge Perspective. *Informatics in Education*, *10*(1), 73–88. https://doi.org/10.1145/2016911.2016943

Sajjanhar, A., & Faulkner, J. (2014). Exploring second life as a learning environment for computer programming. *Creative Education*, *5*(1), 53–62. https://doi.org/10.4236/ce.2014.51010

Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social and Behavioral Sciences*, *103*(2013), 127–135. https://doi.org/10.1016/j.sbspro.2013.10.317

Sampson, H. (2004). Navigating the waves: the usefulness of a pilot in qualitative research. *Qualitative Research*, *4*(3), 383–402. https://doi.org/10.1177/1468794104047236

Samur, Y. (2009). Can we use second life for language learning and instruction? *Dil Dergisi*, (145), 086–094. https://doi.org/10.1501/Dilder_0000000117

Sauppé, A., Szafir, D., Huang, C.-M., & Mutlu, B. (2015). From 9 to 90: Engaging Learners of All Ages. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education - SIGCSE '15* (pp. 575–580). Kansas City, MO, USA: ACM Press. https://doi.org/10.1145/2676723.2677248

Sayın, Z. (2017). Bilgisayar Bilimi Eğitimi Kapsamı. In Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (1. Baskı, pp. 133–154). Ankara.

Schank, R. C. (2002). Every curriculum tells a story. *International Journal of Cognition and Technology*, *1*(1), 169–182. https://doi.org/10.1075/ijct.1.1.11sch

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by Doing. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models*. London: Lawrence Erlbaum.

Schank, R. C., Fano, A., Bell, B., & Jona, M. (1994). The Design of Goal-Based Scenarios. *Journal of the Learning Sciences*, *3*(4), 305–345. https://doi.org/10.1207/s15327809jls0304_2

Schank, R. C., & Kass, A. (1996). A Goal-Based Scenario for High School Students. *Communications of the ACM*, *39*(4), 1–2.

Schulte, C., & Carsten. (2013). Reflections on the role of programming in primary and secondary computing education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education on - WiPSE '13* (pp. 17–24). Aarhus, Denmark: ACM Press. https://doi.org/10.1145/2532748.2532754

Schulte, C., Clear, T., Taherkhani, A., Busjahn, T., & Paterson, J. H. (2010). An introduction to program comprehension for computer science educators. *Proceedings of the 2010 ITiCSE Working Group Reports on Working Group Reports - ITiCSE-WGR '10*, 65. https://doi.org/10.1145/1971681.1971687

Scratch - About. (2017). Retrieved October 10, 2017, from https://scratch.mit.edu/about/

Selby, C. C. (2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE '15* (pp. 80–87). London, United Kingdom: ACM Press. https://doi.org/10.1145/2818314.2818315

Seng, K. L. J., & Edirisinghe, E. M. N. S. (2007). Teaching computer science using Second Life as a learning environment. In *Ascilite* (pp. 583–586).

Shernoff, D., & Silva, R. (2017). Motivation in Out-of-School Learning. In K. Peppler (Ed.), *The SAGE Encyclopedia of Out-of-School Learning* (Vol. 1, pp. 501–505). Thousand Oaks, California: SAGE Publications, Inc. https://doi.org/10.4135/9781483385198.n196

Shih-Wei Chou, & Chien-Hung Liu. (2005). Learning Effectiveness in Web-Based Technology-Mediated Virtual Learning Environment. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (pp. 1–10). IEEE. https://doi.org/10.1109/HICSS.2005.385

Stake, R. E. (1995). *The art of case study research*. Thousand Oaks, CA: Sage Publications.

Stake, R. E. (2005). Qualitative Case Studies. In N. K. Denzin & Y. S. Lincoln (Eds.), *The SAGE handbook of qualitative research* (3rd ed., Vol. 3rd, pp. 443–466). Thousand Oaks, CA: SAGE Publications. Retrieved from http://www.loc.gov/catdir/toc/ecip053/2004026085.html

Stake, R. E. (2006). *Multiple Case Study Analysis*. New York: The Guilford Press.

Şendağ, S., & Başer, V. (2013). Öğretim Teorileri ve Öğretim Teknolojileri. In K. Çağıltay & Y. Göktaş (Eds.), *Öğretim Teknolojilerinin Temelleri: Teoriler, Araştırmalar ve Eğilimler*. Ankara: Pegem Akademi.

Tekdal, M. (2013). The effect of an example-based dynamic program visualization environment on students' programming skills. *Educational Technology and Society*, *16*(3), 400–410.

Tenenberg, J., & McCartney, R. (2014). Editorial: Computing Education in (K-12) Schools from a Cross-National Perspective. *ACM Transactions on Computing Education*, *14*(2), 1–3. https://doi.org/10.1145/2602481

Thackray, L., Good, J., & Howland, K. (2010). Learning and Teaching in Virtual Worlds: Boundaries, Challenges and Opportunities. In A. Peachey, J. Gillen, D. Livingstone, & S. Smith-Robbins (Eds.), *Researching Learning in Virtual Worlds* (pp. 139–158). London: Springer London. https://doi.org/10.1007/978-1-84996-047-2_8

Tokel, S. T., & Cevizci-Karataş, E. (2014). Three-dimensional virtual worlds: Research trends and future directions. *Mersin University Journal of the Faculty of Education*, *10*(1), 1–12.

Tokel, S. T., & Topu, F. B. (2016). Üç boyutlu sanal dünyalar. In K. Çağıltay & Y. Göktaş (Eds.), *Öğretim Teknolojilerinin Temelleri: Teoriler, Araştırmalar, Eğilimler* (2nd ed., pp. 825–844). Ankara: Pegem Akademi.

Tüzün, H., & Özdinç, F. (2016). The effects of 3D multi-user virtual environments on freshmen university students' conceptual and spatial learning and presence in departmental orientation. *Computers & Education*, *94*, 228–240. https://doi.org/10.1016/j.compedu.2015.12.005

Tüzün, H., Yilmaz-Soylu, M., Karakuş, T., Inal, Y., & Kizilkaya, G. (2009). The effects of computer games on primary school students' achievement and motivation in geography learning. *Computers and Education*, *52*, 68–77. https://doi.org/10.1016/j.compedu.2008.06.008

Üçgül, M. (2012). *Design and Development Issues for Educational Robotics Training Camps*. *(Unpublished Doctoral Dissertation)*. Middle East Technical University, Ankara.

Üçgül, M., & Çağıltay, K. (2014). Design and development issues for educational robotics training camps. *International Journal of Technology and Design Education*, *24*(2), 203–222. https://doi.org/10.1007/s10798-013-9253-9

Veletsianos, G., Heller, R., Overmyer, S., & Procter, M. (2010). Conversational agents in virtual worlds: Bridging disciplines. *British Journal of Educational Technology*, *41*(1), 123–140. https://doi.org/10.1111/j.1467-8535.2009.01027.x

Vosinakis, S., Anastassakis, G., & Koutsabasis, P. (2016). Teaching and learning logic programming in virtual worlds using interactive microworld representations. *British Journal of Educational Technology*. https://doi.org/10.1111/bjet.12531

Warburton, S. (2009). Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *British Journal of Educational Technology*, *40*(3), 414–426. https://doi.org/10.1111/j.1467-8535.2009.00952.x

Waite, J. (2017). Pedagogy in teaching Computer Science in schools : A Literature Review. The Royal Society. Retrieved from https://royalsociety.org/~/media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf

Wing, J. M. (2006). Computational thinking. *Comunications of the ACM*, *49*(3), 1–3.

Winn, W. (2005). What we have learned about VR and learning and what we still need to study. *In Proceedings of Virtual Reality International Conference*: Laval, France.

Witmer, B. G., & Singer, M. J. (1998). Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, *7*(3), 225–240. https://doi.org/10.1162/105474698565686

Woodside, A. G. (2010). *Case study research: Theory, methods, practice*. Bingley, UK: Emerald Group Publishing.

Wyffels, F., Martens, B., & Lemmens, S. (2014). Starting from Scratch: Experimenting with Computer Science in Flemish Secondary Education. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14* (pp. 12–15). New York, New York, USA: ACM Press. https://doi.org/10.1145/2670757.2670763

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2017). Expanding computer science education in schools : Understanding teacher experiences and challenges. *Computer Science Education*, *3408*(January), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55–62. https://doi.org/10.1145/2994591

Yazıcı, C., & Kültür, C. (2013). Medya mı Yöntem mi? Bitmeyen Tartışma. In K. Çağıltay & Y. GÖKTAŞ (Eds.), *Öğretim Teknolojilerinin Temelleri: Teoriler, Araştırmalar, Eğilimler* (1st Ed., pp. 41–57). Ankara: Pegem Akademi.

Yee, N. (2006). The Psychology of Massively Multi-User Online Role-Playing Games: Motivations, Emotional Investment, Relationships and Problematic Usage. In R. Schroeder & A. Axelsson (Eds.), *Avatars at work and play: Interaction in shared virtual environments* (pp. 187–207). London: Springer-Verlag.

Yıldırım, A., & Şimşek, H. (2013). *Sosyal bilimlerde nitel araştırma yöntemleri* (9th ed.). Ankara: Seçkin Yayıncılık.

Yin, R. K. (1981). The case study crisis: Some answers. *Administrative Science Quarterly*, *26*(1), 58–65. https://doi.org/10.2307/2392599

Yin, R. K. (1994). *Case Study Research* (2nd Editio). Thousand Oaks, CA, CA: SAGE Publications.

Yin, R. K. (2003). *Case study research: Design and methods* (Fourth Ed.). Thousand Oaks, CA: SAGE Publications.

APPENDIX A

LEARNING GOALS

- **Algoritma ve programlamanın temel kavramları:** Problemin çözümü için yapılması gereken temel aşamalar hakkında bilgi, algoritma kavramının çocuklara öğretilmesi
    - ♦ Problemin tanımı
    - ♦ Çözüm yolunun tespiti
    - ♦ Algoritma (akış şeması) hakkında bilgi ve hazırlanması
    - ♦ Akış şemalarının görsel olarak gösterilmesi
- **Döngüler(for, while, do while):** Programlamanın temel komutlarından olan birden fazla tekrar edilmesi istenilen kod veya kod grupları için kullanılması gereken kod bloğu,
    - ♦ … defa tekrarla : bir kod bloğunu verilen sayı kadar tekrar halinde çalıştırır.
    - ♦ Sürekli tekrarla: verilen kod bloğunu sürekli olarak çalıştırır.
- **Kontrol Yapıları (if, ve else if):** Bulunulan duruma uygun olarak, program akışı içerisinde uygun koşul cümleleri kullanmak. Programlamada temel olarak kullanılan programın akışını bir koşula göre değiştiren if (eğer)…... else ve if (eğer) ….. else if kod bloklarnın kullanımı
- **Mantıksal Sınama:** Bir durumu başka bir durumla eşitlik, büyüklük veya küçüklük yönünden karşılaştırma ve sonrasında bir sonuç oluşturma.
    - ♦ a>b , a<b, a=b
    - ♦ a ve b, a ya da b

- ♦ a küçük değildir b
- **Rastgele Sayılar:** Program akışı içerisinde ihtiyaç duyulması halinde rastgele olarak belirlenen aralıkta sayı üretme ve bunu gerekli yerlerde kullanma.
    - ♦ 1 ile 10 arasında bir sayı tut
- **Kullanıcı Etkileşimi:** Oluşturulan kodların kullanıcının yaptığı bir hareket sayesinde çalıştırılmasının başlatılması veya tetiklenmesi, programın kullanıcı ile etkileşimi
    - ♦ Bir objeye dokunulması durumu
    - ♦ Avatarın bir objeye çarpması (collision) durumu
    - ♦ Objenin oluşturulması (created) durumu
    - ♦ Kullanıcının klavyeden bir şey yazması
- **Değişken kavramı:** Değişken, kullanım amacı, ve şekli
    - ♦ Değişken oluşturma, isim verme ve değer atama
    - ♦ Değişkenler ile ilgili matematiksel ve diğer işlemler,
    - ♦ Değişkenin değerini değiştirmek
- **Objeler Arası İletişim ve Senkranizasyon:** Ortamdaki objelerin birbirleriyle iletişimi ve bir birlerine komut (parametre) göndermeleri.
    - ♦ Bir zil objesine basılınca kapının (bir diğer objenin hareket etmesi) açılması gibi
- **Programı Sınama ve hataları giderme:** yazılan kodları test etme ve varsa hataları düzeltip tekrar çalıştırma.

APPENDIX B

COVER STORY

Merhaba,

Bu kasabada bir takım sorunlar olmuş ve aksaklıklar meydana gelmiştir. Örneğin Yeşilırmak üzerindeki köprünün yıkılması ve yeniden inşa edilmesi, bazı evlerin merdiven ve bahçe duvarına ihtiyaç duyması gibi sorunlar, ya da bir kafenin sayaca ihtiyaç duyması aksaklıklar vardır.

Bu kasabada; takım arkadaşın ile birlikte her birinize toplam 12 tane olmak üzere 24 tane tamamlanması gereken görev bulunmaktadır. Her bir görev için gerekli malzemeler ve bilgiler, ortamda dönen renkli kutular aracılığıyla sana verilecektir. Bu kutuya dokunduğunda görevi tamamlamak için ihtiyacın olan her şey envanterine gelecektir. Ayrıca sana verilen görev kartlarından görevler hakkında bilgi ve uyarılara ulaşabilirsin.

Senden istenilen bu görevleri takım arkadaşın ile birlikte tamamlamaktır. Bu sırada, arkadaşın ile yardımlaşabilir, soru sorabilir ve her türlü fikir alışverişinde bulunabilirsin.

Unutma, bu kasabanın geleceği takım arkadaşın ile birlikte sana emanet…

APPENDIX C

PERMISSION OF METU-ETHICAL COMMITTEE

UYGULAMALI ETİK ARAŞTIRMA MERKEZİ
APPLIED ETHICS RESEARCH CENTER

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

Sayı: 28620816 / 361

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 22 91
F: +90 312 210 79 59
ueam@metu.edu.tr
www.ueam.metu.edu.tr

09 EYLÜL 2015

Gönderilen: Yrd. Doç. Dr. S.Tuğba TOKEL

Bilgisayar ve Öğretim Teknolojileri Eğitimi

Gönderen: Prof. Dr. Canan SÜMER

İnsan Araştırmaları Komisyonu Başkanı

İlgi: Etik Onayı

Danışmanlığını yapmış olduğunuz Bilgisayar ve Öğretim Teknolojileri Eğitimi Araş. Gör. Ali BATTAL 'ın "3D Sanal Dünyaların Çocuklara Yönelik Programlamaya Giriş Eğitiminde Etkili Bir Şekilde Kullanılması: Bir Durum Çalışması" isimli araştırması " İnsan Araştırmaları Komisyonu tarafından uygun görülerek gerekli onay 01.09.2015 -31.08.2016 tarihleri arasında geçerli olmak üzere verilmiştir.

Bilgilerinize saygılarımla sunarım.

Prof. Dr. Canan SÜMER

Uygulamalı Etik Araştırma Merkezi

İnsan Araştırmaları Komisyonu Başkanı

Prof. Dr. Meliha ALTUNIŞIK
Etik Komitesi Üyesi

Prof. Dr. Aydan BALAMİR
Etik Komitesi Üyesi

Prof. Dr. Mehmet UTKU
Etik Komitesi Üyesi

Prof. Dr. Ayhan SOL
Etik Komitesi Üyesi

APPENDIX D

PERMISSION OF MINISTRY OF NATIONAL EDUCATION

T.C.
ANKARA VALİLİĞİ
Milli Eğitim Müdürlüğü

ÖĞRENCİ İŞLERİ
DAİRESİ BAŞKANLIĞI
Ev. Arş. Md. Saat :

Sayı : 14588481-605.99-E.10463716

16.10.2015

Konu: Araştırma izni

ORTA DOĞU TEKNİK ÜNİVERSİTESİNE
(Öğrenci İşleri Daire Başkanlığı)

İlgi: a) MEB Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 2012/13 nolu Genelgesi.
b) 30/09/2015 tarihli ve 9708 sayılı yazınız.

Üniversiteniz Doktora Öğrencisi Ali BATTAL' ın **"3D sanal dünyaların çocuklara yönelik programlamaya giriş eğitiminde etkili bir şekilde kullanılması: Bir durum çalışması"** başlıklı tezi kapsamında çalışma yapma talebi Müdürlüğümüzce uygun görülmüş ve araştırmanın yapılacağı İlçe Milli Eğitim Müdürlüğüne bilgi verilmiştir.

Uygulama formunun (8 sayfa) araştırmacı tarafından uygulama yapılacak sayıda çoğaltılması ve çalışmanın bitiminde bir örneğinin (cd ortamında) Müdürlüğümüz Strateji Geliştirme (1) Şubesine gönderilmesini arz ederim.

Ali GÜNGÖR
Müdür a.
Şube Müdürü

19.10.2015 – 15414

Güvenli Elektronik İmzalı
Aslı İle Aynıdır.
16.10.2015

Atatürk Blv. 06648 Kızılay/ANKARA
Elektronik Ağ: www.meb.gov.tr
e-posta: adsoyad@meb.gov.tr

Ayrıntılı bilgi için: Ad SOYAD Ünvan
Tel: (0 312) XXX XX XX
Faks: (0 312) XXX XX XX

Bu evrak güvenli elektronik imza ile imzalanmıştır. http://evraksorgu.meb.gov.tr adresinden f80f-0110-3215-8c8b-e3d0 kodu ile teyit edilebilir.

APPENDIX E

PARENT CONSENT FORM

Sayın Veli,

Bu çalışma ODTÜ - Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü öğretim üyelerinden Yrd. Doç. Dr. S. Tuğba TOKEL danışmanlığında, Arş. Gör. Ali BATTAL tarafından yürütülmektedir. Bu form çocuğunuz ile yapılması planlanan araştırma koşulları hakkında sizi bilgilendirmek için hazırlanmıştır. Bu çalışmanın amacı, son zamanların popüler konusu olan programlama eğitiminin üç boyutlu sanal dünyalarda etkin olarak yapılması ve bu eğitime etki eden sanal dünya ile ilgili faktörlerin incelenmesi olarak belirlenmiştir. Bununla birlikte ortamın programlama eğitimine sağladığı olanaklar ve ortam ile ilgili sınırlıklar bu çalışma kapsamında incelenecektir.

Bu çalışma boyunca çocuğunuzun üç boyutlu sanal ortamda araştırmacı tarafından programlama öğretimi için hazırlanan görevlerin bir kısmını bireysel, bir kısmını ise takım arkadaşı ile birlikte tamamlaması beklenmektedir. Çalışma sonunda yukarıda belirtilen amaçlar doğrultusunda çocuğunuz ile bir görüşme yapılması planlanmaktadır. Çocuğunuzun bu görüşme sırasında bizimle paylaşacağı görüşler son derece önemli olduğu ve en ufak bir görüş kaybına sebebiyet vermemek için ses kaydı yapılması planlanmaktadır. Bu kayıtlar sadece araştırmacı tarafından incelenecektir. Çalışmaya katılmak tamamen gönüllülük esasına dayalıdır Çocuğunuz arzu ettiği takdirde hiç bir yaptırıma maruz kalmadan katılımdan vazgeçme hakkına sahiptir.

Bu çalışma hakkında daha fazla bilgi almak, her türlü soru ve/veya yorumlarınız için Arş. Gör. Ali BATTAL ile 0 (312) 210 4183 nolu telefondan veya albattal@metu.edu.tr e-posta adresinden iletişime geçebilirsiniz.

Lütfen bu araştırmaya katılmak konusundaki tercihinizi aşağıdaki seçeneklerden size en uygun gelenin altına imzanızı atarak belirtiniz.

A) Velisi bulunduğum ....’nın bu araştırmaya katılımcı olmasına izin veriyorum. Çocuğumun çalışmayı istediği zaman yarıda kesip bırakabileceğini biliyorum ve verdiği bilgilerin bilimsel amaçlı olarak kullanılmasını kabul ediyorum.

Velinin Adı-Soyadı ....

İmza ....

B) Velisi bulunduğum ....’nın bu çalışmaya katılmasını kabul etmiyorum ve katılımcı olmasına izin vermiyorum.

Velinin Adı-Soyadı ....

İmza ....

APPENDIX F

STUDENT CONSENT FORM

Bu çalışma ODTÜ Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü öğretim üyelerinden Yrd. Doç. Dr. S. Tuğba TOKEL danışmanlığında, Arş. Gör. Ali BATTAL tarafından yürütülmektedir. Bu form seni araştırma koşulları hakkında bilgilendirmek için hazırlanmıştır. Bu çalışmanın amacı, üç boyutlu sanal dünyaların programlama eğitiminde kullanılması ve bu ortamın etkin olarak kullanılmasına etki eden faktörlerin incelenmesi olarak belirlenmiştir. Bununla birlikte 3B ortamın programlama eğitiminde sağladığı olanaklar ve ortam ile ilgili sınırlıklar bu çalışma kapsamında incelenecektir.

Bu çalışma boyunca senden üç boyutlu sanal ortamda verilen görevlerin bir kısmını bireysel bir kısmını ise grup arkadaşların ile birlikte tamamlaman beklenmektedir. Bunun dışında çalışma sonunda ortamın etkililiğini ve yukarıda belirtilen diğer amaçları saptamak için seninle bir görüşme yapılması planlanmaktadır. Bu çalışmaya katılmak tamamen gönüllülük esasına dayalıdır. Araştırma süresince yapılan görüşmeler, gözlemler ve diğer bilgiler tamamen gizli tutulacaktır. Ayrıca toplanan verilere sadece araştırmacılar tarafından bu araştırma kapsamında değerlendirilecektir. Araştırma süresince yapılan etkinlikler seni hiçbir şekilde rahatsız etmeyecektir, ancak yine de program sürecinde bir nedenle rahatsızlık duyarsan, istediğin zaman araştırmadan çıkabilirsin.

Bu çalışma hakkında daha fazla bilgi almak, her türlü soru ve yorumlarınız için Arş. Gör. Ali BATTAL ile 0 (312) 210 4183 nolu telefondan veya albattal@metu.edu.tr e-posta adresinden ulaşabilirsin.

Yukarıdaki bilgileri okudum ve bu çalışmaya tamamen gönüllü olarak katılıyorum. Verdiğim bilgilerin bilimsel amaçlı olarak kullanılmasını kabul ediyorum.

Adı- Soyadı :

Tarih : …. / …. / ………

İmza :

APPENDIX G

INTERVIEW PROTOCOL FOR STUDENTS

ÖĞRENCİ GÖRÜŞME FORMU

Yer **:** ………………….……………..

Tarih **:** ….. / ….. / …..

Görüşülen Kişi **:** ………………….……………..

Merhaba,

Öncelikle bu çalışmaya katıldığın ve sanal ortamda sana verilen görevleri özverili bir şekilde tamamladığın için teşekkür etmek istiyorum. 3B Sanal ortamda programlama eğitimini tamamladık. Şimdi sizinle aldığınız eğitim ile ilgili biraz konuşmak istiyorum. Soracağım soruların hiç birinin doğru ya da yanlış cevabı yoktur, ben sadece senin bu konuyla ilgili ne düşündüğünü merak ediyorum.

Soracağım sorular kesinlikle sizin bilginizi ölçme amaçlı değildir. Bu nedenle sorduğum sorulara içtenlikle cevap verebilirsin. Görüşme sırasında sorularda, anlamadığın noktalar olursa sen de sorabilirsin.
Başlamadan önce bu söylediklerimle ilgili belirtmek istediğin bir düşünce ya da sormak istediğin bir soru var mı? İzin verirsen görüşmeyi kaydetmek istiyorum. Bu görüşme yaklaşık 30-45 dakika sürecektir. Görüşme sırasında ara vermek isterseniz bana söylemeniz yeterli. Görüşmeye başlayabilir miyiz?

SORULAR

1. İlk başta seni tanımak istiyorum.
    a. Evinizde bilgisayar ve Internet var mı?
    b. Bilgisayar veya video oyunu oynuyor musun?
        i. Evet ise
            1. Ne tür oyunlar oynarsın? (3 boyutlu sanal oyunlar GTA)
            2. Ne kadar süredir oynuyorsun?
            3. Ne sıklıkta oynarsın?
            4. Bu oyunları seçmenin bir sebebi var mı?
        ii. Hayır ise
            1. Oyun oynamamanın özel bir sebebi var mı? Açıklayabilir misin?
    c. Oyun programlama kulübünü neden seçtin? (Case-2 öğrencileri için)
2. Daha önce programlama öğrenimi konusunda bir aktiviteye katıldın mı?
    a. Evet ise
        i. Nerede tanıştın? Ne tarz aktivitelerde bulundun?
        ii. Hangi araçları/yazılımları kullandın?
        iii. Ne tarz programlar hazırladın? (bir önceki soruya verdiği cevaba göre düzenlenecek, Örnek Scratch ise mesela kedi karakterine neler yaptırdın?)
        iv. Neden programlama öğrenmek istedin?
        v. Programlama öğrenimi sırasında zevk alıyor muydun? Yoksa bir süre sonra sıkıcı mı geliyor sana?
    b. Hayır ise
        i. Programlama ile ilgili bir fikrin var mıydı? Neler biliyordun?
3. Şimdi de sanal dünyalarda programlama ortamında geçirdiğin zaman ile ilgili sormak istiyorum.

a. Sanal ortama ilk girdiğinde neler hissettin/ ne düşündün? (binalar, objeler, avatarın olması)

b. Ortamda görevleri yaparken eğlendin mi?

c. Sanal ortamın sana olan faydaları nelerdir? Örneğin, *programlama öğrenmene fayda sağladı mı? Programlamayı sevmeni sağladı mı? Deneyim?*

  i. Örneklendirebilir misin?

d. Ortamın en çok sevdiğin özelliklerden bahseder misin?

  i. Örneklendirebilir misin?

e. Ortamda sevmediğin şeyler var mıydı?

  i. Örneklendirebilir misin?

4. Bu çalışmanın ilk kısmında yalnız çalışırken, ikinci kısımda Sorunlu Kasabanın sorunlarını çözmek için takım arkadaşın ile birlikte çeşitli görevleri tamamladın. Şimdi de sizinle bununla ilgili konuşmak istiyorum.

  a. Sizce yalnız çalışmak mı yoksa takım ile birlikte çalışmak mı daha iyi idi? Nedenleri ile birlikte açıklayabilir misin?

  b. Takım arkadaşların ile ne derece anlaştınız?

    i. Benzer görevleriniz oldu mu?

    ii. Görevleri tamamlarken takım arkadaşın ile yardımlaştın mı? Ne tür bir yardımlaşma oldu? Açıklayabilir misin?

  c. Görevleri tamamlarken arkadaşından kaynaklanan bir problem ile karşılaştın mı?

    i. Neler yaşadığın ile ilgili örnek/örnekler verebilir misin?

    ii. Bu sorun (veya sorunları) nasıl çözdün?

  d. Takım arkadaşın ile birlikte çalışmak seni motive etti mi? *Örneğin, iş bölümü, seni görevleri yapma konusunda motive etme gibi.*

5. Sorunlu Kasabadaki sorunları (görevleri) takım arkadaşın ile birlikte çalışarak çözmeye çalıştın. Şimdiki sorularım birlikte görev yapmak ile ilgili olacak.

  a. Ortamda takım arkadaşın ile nasıl bir iletişim sağladın?

    i. Yaşadığın zorluklar nelerdi?

    ii. Ne tür özellikler olsaydı daha rahat sohbet edebilirdin?

  b. Sanal ortamda programlama öğrenmeniz için neler yapıldı? Örneğin,

i. Öğretmen neler yaptı?
ii. Ortamda programlama konularını öğrenmenize yardımcı olan şeyler nelerdi? Nelerden yararlandınız?
iii. Neler olsaydı daha iyi öğrenirdiniz?

6. Genel olarak sanal ortamda seni temsil eden bir avatar vardı. Şimdiki soracakların bu avatar ile ilgili olacak.
   a. Avatarın ile ilgili bir değişiklik yaptın mı? Neden yaptın/yapmadın?
   b. Bu avatarın seni gerçekten sanal dünyada temsil ettiğini düşünüyor musun?
   c. Avatarın en çok hangi özelliğini beğendin / hangilerini beğenmedin?
   d. Avatarının gerçek hayatta benzediğini düşündüğün noktalar neler?
7. Şimdi de sanal ortamın kullanımı sırasında yaşadığın zorluklar ile ilgili sormak istiyorum.
   a. Ortamı rahat kullanabildin mi? Kullanırken zorluk yaşadın mı? [yaşadığın zorluklardan bahseder misin?]
      i. Örneklendirebilir misin?
   b. Scratch programını kullanırken ve kodları oluştururken zorluklar yaşadın mı?
      i. Ne tür zorluklar yaşadın? / Örneklendirebilir misin?
   c. Scratchde oluşturduğun kodları sanal ortama aktarım sırasında zorluklar yaşadın mı?
   d. Scratch programında yapmak isteyip de yapamadığın şeyler oldu mu?
   e. Kodların çalışmadığı veya beklediğin gibi çalışmadığı durumlar oldu mu? Bu durumda neler yaptın?
   f. Sanal ortamda bilgisayar veya Internet bağlantısıyla ilgili sorunlar yaşadın mı?
   g. Yaşadığın zorluklar memnuniyetini nasıl etkiledi?
8. Bu dersi biz bu şekilde sanal ortamda yaptık, sizce bu ders nasıl olmalıydı? Örneğin,
   a. Görevler:
      i. Görev sayısı?
      ii. Görevlerin zorluğu?
      iii. Görevleri anlama? Görevleri ayırt etme? Görevlere ulaşma?

iv. Takım arkadaşın ile görevlerin farklı olması konusunda ne düşünüyorsunuz?

b. Grup:

   i. Bireysel / Grup? Neden?
   ii. Gruptaki kişi sayısı?
   iii. Grup üyelerini belirleme nasıl olmalı? Neden?

c. Eğitim:

   i. Programların kullanımı için önceden ayrıca bir eğitim verilmeli mi?
      1. Örneğin Scratch programının nasıl kullanılacağı eğitimi
   ii. Ders saatleri yeterli mi? Kaç saat olmalıydı?
   iii. Ders saatleri dışında da sanal ortamdaki görevleri tamamlama olmalı mıydı?
   iv. Tamamen sınıf ortamında / Tamamen sanal ortamda?

d. Öğrenme Ortamı / Öğrenme Teorisi:

   i. Rol:
      1. Sanal ortamda sana verilen rolü beğendin mi? Neden
      2. Daha farklı bir rolde olmak ister miydin? Hangi rolü seçerdin?
   ii. Hikaye ve Misyon: sanal ortamda Sorunlu bir kasaba ve senin bu kasabada takım arkadaşın ile birlikte bir misyonun vardı.
      1. Bununla ilgili düşüncelerin neler?
      2. Başka bir misyon veya başka bir kasaba hikayesi ister miydin?
   iii. Öğrenme materyalleri:
      1. Sanal ortamda levha, video, pano gibi araçları inceledin mi?
      2. Bu araçlar daha fazla olmalı mıydı?
      3. Görev kartları/dosyaları hakkında ne düşünüyorsunuz?
   iv. Geri dönüt: Her görev için bir yardım kutucuğu vardı.
      1. Yardım aldın mı / yardım videoları izledin mi (veya hangi sıklıkta izledin)?

        2. Görevler ile ilgili yardımı başka nasıl aldın?
        3. Verilen yardımdan farklı olarak sen nasıl bir yardım almak isterdin?
    e. Sanal ortamın farklı olmasını istediğin yönler oldu mu? Ortamın tasarımı ile ilgili şöyle olsa daha iyi olurdu dediğin noktalar neler?
9. Bu eğitim sırasında programlamaya karşı düşüncelerin değişti mi?
    a. İlk başta programlama ile ilgili ne düşünüyordun?
    b. Şimdi ne düşünüyorsun?
    c. Bu düşüncelerinizin değişmesinde <u>neler</u> etkili oldu?
10. Sanal ortamda programlama konusunda kendini nasıl görüyorsun? Örneğin,
    a. Eğitim öncesinde / sonrasında nasıldın?
11. Aldığın bu eğitim programlama konusunda beklentilerini ne düzeyde karşıladı?
12. Tekrar bu tarz sanal ortamlarda başka bir eğitim/ders almak ister misin?

Sorularım bitti. Senin sormak istediğin bir şey yoksa görüşmeyi bitirebiliriz. Görüşmeye katıldığın ve görüşlerini benimle paylaştığın için teşekkür ederim.

APPENDIX H

INTERVIEW PROTOCOL FOR TEACHERS

ÖĞRETMEN GÖRÜŞME FORMU

Yer **:** ………………….…………….

Tarih **:** ….. / ….. / …..

Görüşülen Kişi **:** ………………….…………….

Merhaba,

Öncelikle bu çalışmaya katıldığınız ve sanal ortamda verilen eğitimde bana yardım ettiğiniz için teşekkür etmek istiyorum. 3B Sanal ortamda programlama eğitimini tamamladık. Şimdi sizinle birlikte verdiğimiz bu eğitim ile ilgili biraz konuşmak istiyorum.

Soracağım sorular kesinlikle sizin bilginizi ölçme amaçlı değildir. Bu nedenle sorduğum sorulara içtenlikle cevap verebilirsiniz. Görüşme sırasında sorularda, anlamadığınız noktalar olursa sorabilirsiniz.
Başlamadan önce bu söylediklerimle ilgili belirtmek istediğiniz bir düşünce ya da sormak istediğiniz bir soru var mı? İzin verirseniz görüşmeyi kaydetmek istiyorum. Bu görüşme yaklaşık 45-50 dakika sürecektir. Görüşme sırasında ara vermek isterseniz bana söylemeniz yeterli. Görüşmeye başlayabilir miyiz?

SORULAR

1. İlk başta sizi tanımak istiyorum.
    a. Ne kadar süredir bu okulda çalışıyorsunuz?
    b. Ne kadar süredir bilgisayar derslerine giriyorsunuz?
    c. Daha önce herhangi bir programlama dilini çocuklara öğrettiniz mi?
        i. Programlama dilini kullanma amacınız neydi?
        ii. Hangi araçları ve programlama dillerini öğrettiniz?
        iii. Bu dilleri öğretirken zorluklar yaşadınız/yaşıyor musunuz?
2. Sizce sanal dünyaların öğrencilerin programlamaya yönelik motivasyonuna etkisi nasıl oldu? Pozitif, negatif ya da etkilemedi?
    a. Neden etkilemedi?
    b. Negatif ise neden böyle düşünüyorsunuz? Size göre bu ne gibi faktörler etki etmiştir?
    c. Pozitif ise sanal dünyalarının kullanımının öğrencilerin motivasyonunu artırdığının göstergeleri nelerdir?
    Sanal dünyaların aşağıdakileri artırdığını gözlemlediniz mi?
        i. Derse katılım
        ii. Derse/konuya ilgi
        iii. Derslerin zevkli geçmesi
        iv. Derse olan memnuniyet
        v. Derse/konuya çalışma isteği
3. Sanal dünyaların çocukların programlama öğrenmesi konusundaki yararlığı hakkında ne düşünüyorsunuz?
    a. Olumsuz ise neden yararlı olmadığını düşünüyorsunuz?
        i. Yetersiz özellik mi vardı?
        ii. Ne tür geliştirmeler yapılabilir?
    b. Olumlu ise neden yararlı olduğunu düşünüyorsunuz?
4. Bu soruyu öğrencilerinize gözlemlerinize dayanarak cevaplayabilirsiniz.
    a. Sizce öğrenciler sanal dünyaları kullanırken zorluk yaşadın mı? [yaşadıkları zorluklardan bahsede bilir misiniz?]

b. Öğrencilerin sanal dünyaların ve Scratch programının kullanımını öğrenmesi kolay oldu mu?
    i. Örneklendirebilir misin?
c. Sanal dünyaların ara yüzü ve mesajları öğrenciler için anlaşılır mıydı?
d. Sanal dünyadaki görevler öğrencilerin yapabileceği düzeyde miydi?

5. Bu dersi yaklaşık 8 hafta boyunca bu şekilde sanal ortamda yaptık, sizce bu ders nasıl olmalıydı?
   Örneğin,
   a. Görevler:
      i. Görev sayısı?
      ii. Görevlerin zorluğu?
      iii. Görevleri anlama? Görevleri ayırt etme? Görevlere ulaşma?
      iv. Öğrencilerin takım arkadaşı ile görevlerin farklı olması konusunda ne düşünüyorsunuz?
   b. Grup çalışması:
      i. Bireysel / Grup? Neden?
      ii. Gruptaki kişi sayısı?
      iii. Grup üyelerini belirleme nasıl olmalı? Neden?
   c. Eğitim:
      i. Programların kullanımı için önceden ayrıca bir eğitim verilmeli miydi?
         1. Örneğin Scratch programının nasıl kullanılacağı eğitimi
      ii. Ders saatleri yeterli mi? Kaç saat olmalıydı?
      iii. Ders saatleri dışında da sanal ortamdaki görevleri tamamlama olmalı mıydı?
   d. Öğrenme Ortamı / Öğrenme Teorisi:
      i. Rol:
         1. Sanal ortamda öğrencilerin rolü hakkında ne düşünüyorsunuz? Neden
         2. Daha farklı bir rol olabilir miydi?

ii. Hikaye ve Misyon: sanal ortamda Sorunlu bir kasabanın olması ve öğrencilerin bu kasabada takım arkadaşı ile birlikte bir misyonu vardı.
    1. Bununla ilgili düşünceleriniz neler?
    2. Başka bir misyon veya başka bir kasaba hikayesi önerir misiniz?

iii. Öğrenme materyalleri:
    1. Sanal ortamda levha, video, pano gibi araçlar ile ilgili görüşleriniz neler?
    2. Bu araçlar daha fazla olmalı mıydı?
    3. Görev kartları/dosyaları hakkında ne düşünüyorsunuz?

iv. Geri dönüt: Her görev için bir yardım kutucuğu vardı.
    1. Bununla ilgili düşünceleriniz neler?
    2. Görevler ile ilgili yardım başka nasıl olabilirdi?

e. Genel olarak sanal ortamın farklı olmasını düşündüğünüz yönler oldu mu?

i. Çocuklar için anlaması zor olan, gereksiz olduğunu düşündüğünü şeyler var mıydı?

6. Sanal dünyaların programlama öğretiminde kullanılması ve geleceği hakkındaki görüş ve önerileriniz nelerdir?
7. Sanal dünyaları kullanımının avantaj ve dezavantajları nelerdir?

   (diğer programlama dili öğreticileri ile karşılaştırırsanız sizce artıları ve eksileri neler olabilir? Bu soruyu daha önce kullandığınız araçlar ile ilgili deneyimlerinize dayanarak cevap verebilirsiniz. *Örneğin* Scratch programı)

   a. Artıları nelerdir?
   b. Eksileri nelerdir?

Sorularım bitti. Sizin sormak istediğiniz bir şey yoksa görüşmeyi bitirebiliriz. Görüşmeye katıldığınız ve görüşlerinizi benimle paylaştığınız için teşekkür ederim.

APPENDIX I

OBSERVATION FORM

Öğrenci Gözlem Formu		
Gözlemci		
Ortam		
Tarih		
Kategori	**Durum**	**Açıklama**
Görevler	Görevler öğrenci tarafından anlaşılıyor mu?	
	Görevi tamamlarken görev kartından faydalandı mı?	
	Görevi tamamlamak / çözmek için çaba sarf ediyor mu?	
	Görev dışı faaliyetlerde bulunuyor mu?	
	Görevleri anlamakta zorluk çekiyor mu?	
	Görev yardım seçeneğinden yardım aldı mı?	
	Görevi tamamlarken yaşadığı zorluklar?	
	Diğer varsa belirtiniz	
Grup Çalışması	Görevleri tamamlamak için grup arkadaşından yardım alıyor mu?	
	Sınıf ortamındaki diğer kişilerden yardım alma? (size soru sorması, ya da başka bir arkadaşına soru sorması gibi)	
	Takım arkadaşıyla yardımlaşma konusunda ortamın imkanlarını kullanıyor mu? (mesaj gönderme, genel sohbete yazma gibi)	
	Grup arkadaşı ile uyumlu çalışabiliyor mu? Birlikte hareket ediyor mu?	
	Adadaki sorunları çözümlemek için grup arkadaşı ile tartışıyor mu?	

<table>
<tr><th colspan="3">Öğrenci Gözlem Formu</th></tr>
<tr><td></td><td>Diğer varsa belirtiniz</td><td></td></tr>
<tr><td rowspan="5">Sorunlar ve Aksaklıklar</td><td>Öğrenci görevi tamamlarken sorun yaşadı mı?</td><td></td></tr>
<tr><td>Teknik aksaklıklar oldu mu?</td><td></td></tr>
<tr><td>Yaşadığı sorunları nasıl çözdü?</td><td></td></tr>
<tr><td>Programları kullanma sırasında yaşadığı zorluklar?</td><td></td></tr>
<tr><td>Diğer varsa belirtiniz</td><td></td></tr>
<tr><td rowspan="7">3B Ortam ve Avatar</td><td>Görevler öğrenci tarafından anlaşılıyor mu?</td><td></td></tr>
<tr><td>Görevler ile ilgili ortamdaki dokümanları, kaynakları inceledi mi?</td><td></td></tr>
<tr><td>Bir görevi tamamladığında diğerine rahatlıkla ulaşabiliyor mu?</td><td></td></tr>
<tr><td>Sanal ortamda rahat hareket ediyor mu?</td><td></td></tr>
<tr><td>Avatarı ile sorun yaşıyor mu?</td><td></td></tr>
<tr><td>Diğer varsa belirtiniz</td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>Genel Gözlemler</td><td colspan="2"></td></tr>
</table>

APPENDIX J

DEMOGRAPHIC AND PERCEPTION QUESTIONNAIRE

Sevgili çocuklar, bu ankette sanal dünyalarda aldığınız programlama eğitimi ile ilgili deneyimleriniz ve görüşlerinize ait bazı bilgiler istenmektedir. Lütfen her soruyu dikkatli bir şekilde okuyup size en uygun olanını işaretleyiniz. Ankete verdiğiniz cevaplar sadece bu araştırma kapsamında kullanılacaktır.

Şimdiden çalışmaya katıldığınız ve anketi doldurduğunuz için teşekkür ederim.

A) Genel Bilgiler
1. Cinsiyetiniz: ☐ Erkek ☐ Kız
2. Yaşınız : …………………………..
3. Okulunuz: …………………………………….. Sınıfınız: ……….
4. Evinizde bilgisayar var mı? ☐ Var ☐ Yok
5. Evinizde İnternet bağlantısı var mı? ☐ Var ☐ Yok
6. İnternete genellikle nereden erişiyorsun? (Birden fazla seçenek işaretleyebilirsiniz) ☐ Ev ☐ Okul ☐ İnternet Kafe ☐ Diğer (lütfen belirtiniz) ………………

A) Genel Bilgiler
7. İnterneti genellikle ne amaçla kullanıyorsun? (Birden fazla seçenek işaretleyebilirsiniz) ☐ Ödev hazırlama, araştırma yapma ☐ İletişim (sosyal ağlar, mail) ☐ Oyun oynama ☐ Eğlence (Müzik, film, vb) ☐ Diğer varsa (Belirtiniz) …………………………………
8. Haftada ortalama kaç saat Internet kullanıyorsunuz? ☐ Hiç ☐ 3 Saatten az ☐ 3-5 Saat ☐ 6-7 Saat ☐ 7 Saatten fazla
9. Bilgisayar veya video oyunu oynuyor musunuz? ☐ Evet ☐ Hayır Cevabınız Evet ise en çok oynadığınız 3 oyunun isimleri nelerdir? ………………………………………………………… …………………………………………………………. ………………………………………………………….
10. Ne kadar süredir bilgisayar veya video oyunları oynuyorsunuz? ☐ Hiç ☐ 1 yıldan az ☐ 1-3 yıldır ☐ 3 yıldan daha fazla
11. 3B ortamlar ile ilgili deneyiminiz var mı? ☐ Evet ☐ Hayır Cevabınız Evet ise hangi ortamlarda deneyiminiz var? ………………………………………………………… …………………………………………………………. ………………………………………………………….
12. Programlama ile ilgili deneyiminiz var mı? ☐ Var ☐ Yok Var ise kullandığınız programlama araçları veya dillerinden hangisi hakkında bilginiz var? (Birden fazla seçenek işaretleyebilirsiniz)

A) Genel Bilgiler

☐ Scratch ☐ NXT Logo Mindstorm

☐ Code.org ☐ Karel

☐ Microsoft Small Basic ☐ Lightbot

☐ Diğer varsa (lütfen belirtiniz) ……………………………………

B) Algı ve Memnuniyet Anketi

Aşağıda Sanal Dünyalarda programlama eğitimi ilgili verilen ifadelere ne ölçüde katıldığınızı, size en uygun seçeneği işaretleyerek belirtiniz.

İFADELER	Kesinlikle katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle katılıyorum
	1	2	3	4	5
1. Sanal dünyayı kullanmayı öğrenmek benim için kolaydı.	◯	◯	◯	◯	◯
2. Sanal dünya ortamının kullanımını kolay buluyorum.	◯	◯	◯	◯	◯
3. Sanal dünya ortamında etkileşimi açık ve anlaşılır buluyorum.	◯	◯	◯	◯	◯
4. Sanal dünya kullanımında beceri kazanmak benim için kolaydı.	◯	◯	◯	◯	◯

B) Algı ve Memnuniyet Anketi

Aşağıda Sanal Dünyalarda programlama eğitimi ilgili verilen ifadelere ne ölçüde katıldığınızı, size en uygun seçeneği işaretleyerek belirtiniz.

İFADELER	Kesinlikle katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle katılıyorum
	1	2	3	4	5
5. Sanal dünyayı bir öğrenme aracı olarak kullanmak dersteki başarımı arttırdı.	○	○	○	○	○
6. Sanal dünya, kendi öğrenme hızımda ilerlememi sağladı.	○	○	○	○	○
7. Sanal dünyayı kullanmak öğrenmemin daha etkili olmasını sağladı.	○	○	○	○	○
8. Sanal dünyayı kullanmak dersi anlamamı kolaylaştırdı.	○	○	○	○	○
9. Sanal dünyayı kullanmak öğretmen ile iletişim kurmamı kolaylaştırdı.	○	○	○	○	○
10. Sanal dünya, programlama öğrenmeme yardımcı oldu.	○	○	○	○	○
11. Sanal dünyadaki öğrenme deneyimlerimden memnunum.	○	○	○	○	○
12. Sanal dünyanın öğrenmeme katkı sağladığını düşünüyorum.	○	○	○	○	○
13. Genel olarak öğrenmenin etkililiğinden memnundum.	○	○	○	○	○
14. Sanal dünyadaki öğrenme etkililiğinden genel olarak memnunum.	○	○	○	○	○

APPENDIX K

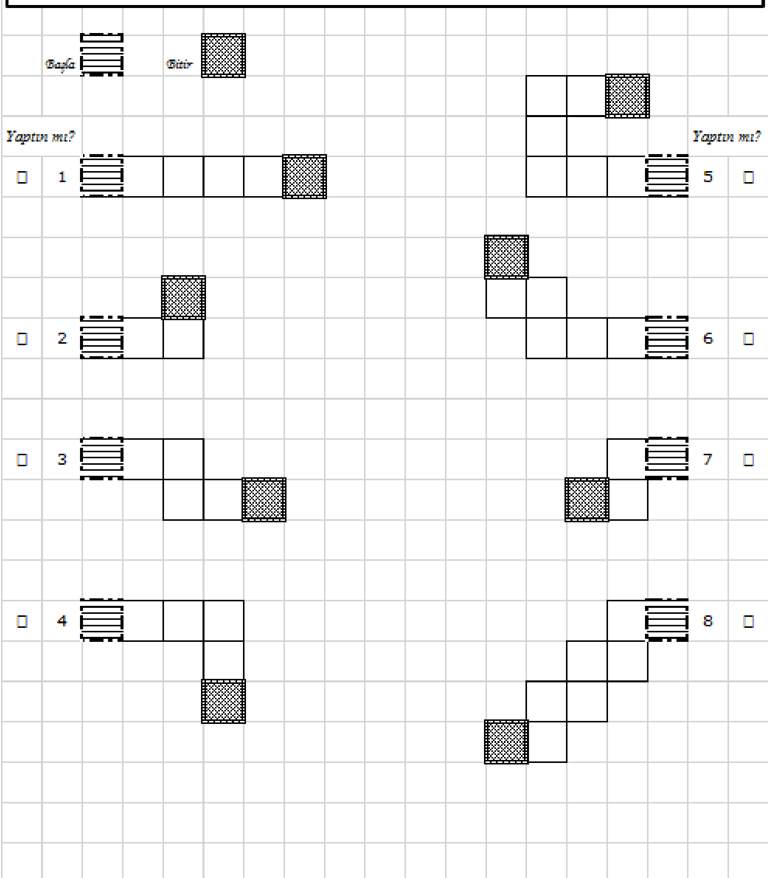
ACTIVITY SHEETS FOR ROBOT TRANINING

Adı-Soyadı:...................................... Robot Eğitimi

Bina 1'deki Görevler

Görevin binadaki numaralı yollarda robotunu sırasıyla belirtilen yolları izleyecek şekilde programlamaktır. Bunu yapmak için;

- Robotunu tam ev resmine gelecek şekilde ve yönünün gitmesini istediğin yönde olmasından emin ol.
- Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin gibi çalışmıyorsa ise robotunu tekrar ev sekmesine getir, ve kodları düzenleyerek tekrar dene.
- Bu işlemleri doğru sonuca ulaşana kadar tekrar edebilirsin.

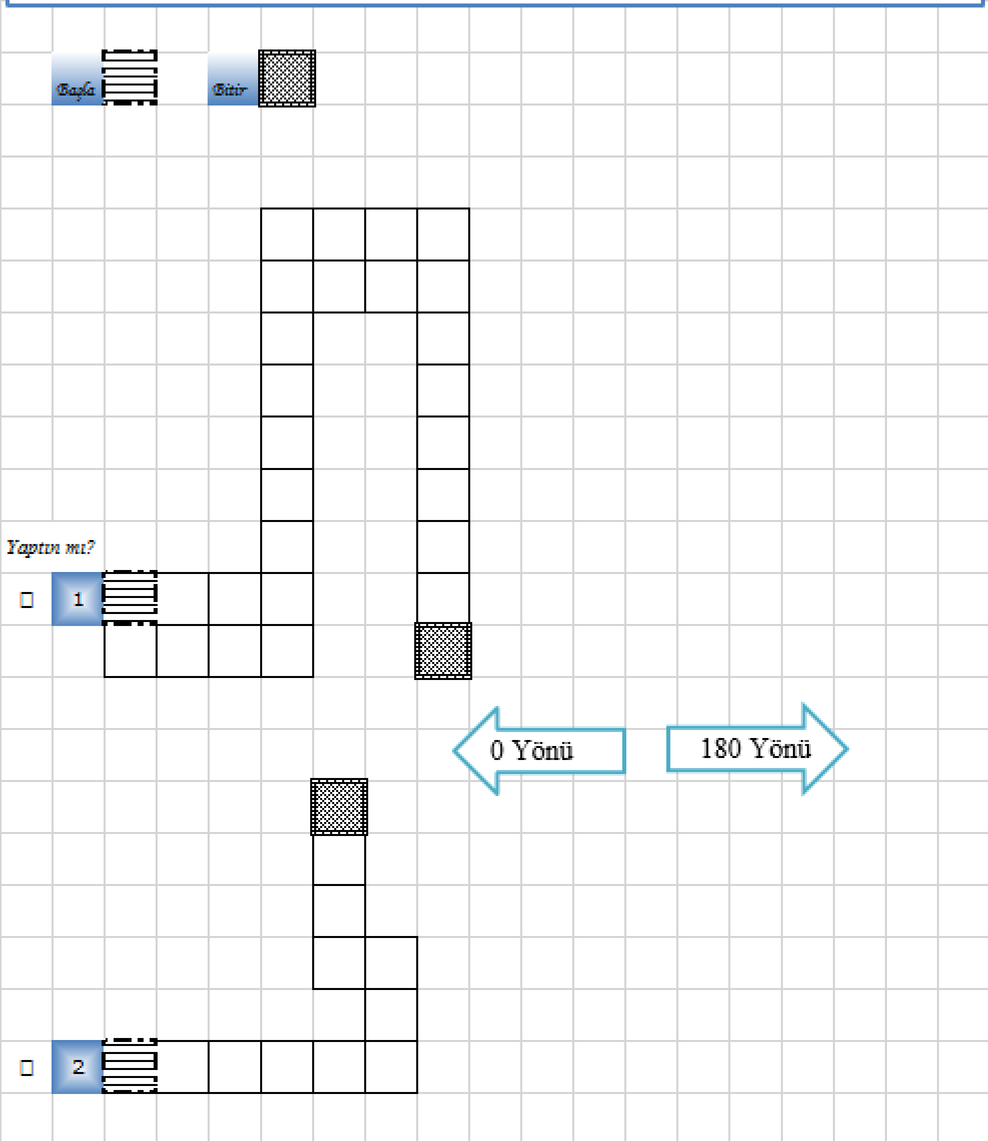


Sanal Dünyalarda Programlama Eğitimi ODTÜ - BÖTE Bina 1

Bina 2'deki Görevler

Görevin binadaki numaralı yollarda robotunu sırasıyla belirtilen yolları izleyecek şekilde programlamaktır. Bunu yapmak için;

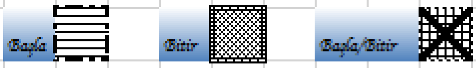
- Robotunu tam ev resmine gelecek şekilde ve yönünün gitmesini istediğin yönde olmasından emin ol.
- Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin gibi çalışmıyorsa ise robotunu tekrar ev sekmesine getir, ve kodları düzenleyerek tekrar dene.
- Bu işlemleri doğru sonuca ulaşana kadar tekrar edebilirsin.

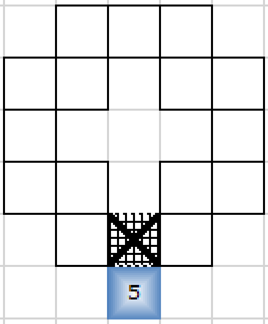


Bina 3'deki Görevler

Görevin binadaki numaralı yollarda robotunu sırasıyla belirtilen yolları izleyecek şekilde programlamaktır. Bunu yapmak için;

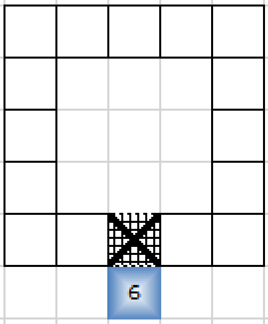
- Robotunu tam ev resmine gelecek şekilde ve yönünün gitmesini istediğin yönde olmasından emin ol.
- Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin gibi çalışmıyorsa ise robotunu tekrar ev sekmesine getir, ve kodları düzenleyerek tekrar dene.
- Bu işlemleri doğru sonuca ulaşana kadar tekrar edebilirsin.





☐

Yaptın mı?



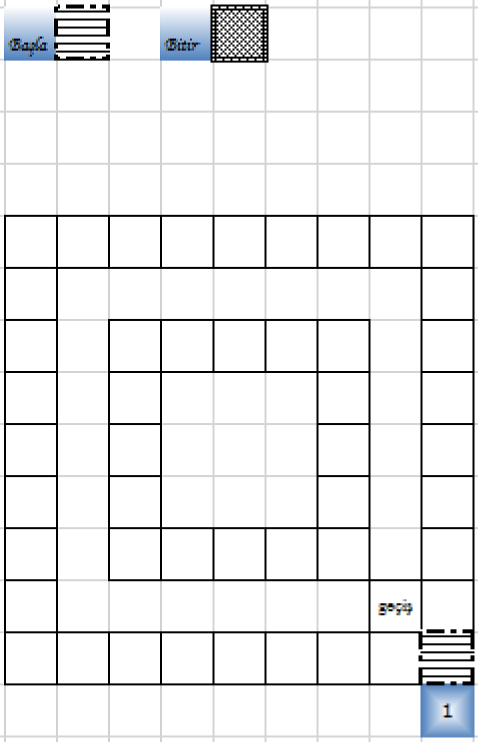
☐

Yaptın mı?

Bina 4'deki Görevler

Görevin robotunu aşağıdaki belirtilen yolu izleyecek şekilde programlamaktır. Yalnız bu binada diğer görevlerden farklı olarak iç içe iki kare vardır, ve iki kare arasında sadece bir geçiş vardır. Görevini tamamlarken şunlara dikkat et;

- Robotunu tam ev resmine gelecek şekilde ve yönünün gitmesini istediğin yönde olmasından emin ol.
- Dıştaki karenin kenar uzunluğu 9 metre içtekinin ise 5 metre olup iki kare arasında geçiş sadece gösterilen şekilde olmaktadır.
- Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin gibi çalışmıyorsa ise robotunu tekrar ev sekmesine getir, ve kodları düzenleyerek tekrar dene.



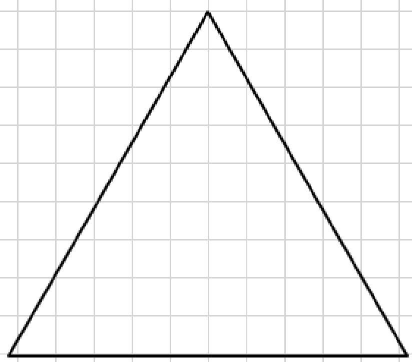
APPENDIX L

ACTIVITY SHEETS FOR 3D OBJECT CONSTRUCTION

ODTÜ - BÖTE 3D Şekil Oluşturma

Eşkenar Üçgen Oluşturma

- ✓ Eşkenar üçgen kenar uzunlukları birbirine eşit 3 kenardan oluşur.
- ✓ İç açıları toplamı 180 derece ve her bir iç açısı 60 derecedir.
- ✓ Dış açılarıda birbirine eşit olup her biri 120'şer derecedir.
- ✓ Kalem aracı ile 3 boyutlu üçgen çizmek için kalem aracını kullandıktan sonra robotu eşit uzunluklarda üçgenin dış açısı kadar döndürerek 3 tane kenar çizmek yeterlidir.
- ✓ Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin şekil oluşmuyorsa robotunu tekrar programlama ve dene. Bunu istediğin kadar yapabilirsin.
- ✓ Aşağıda üçgen ve özellikleri gösterilmiştir.

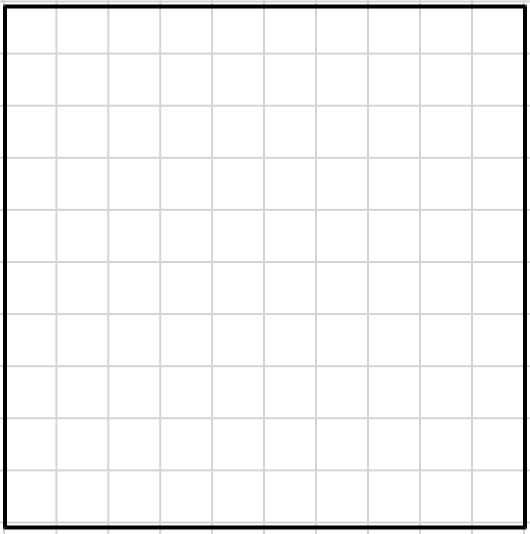


- ❖ Şekil çizen robotu envanterden çizim alanına yerleştiriniz. Eğer daha önce almadı iseniz robotun bir kopyasını masadan envanterine sağ tıklayıp 'ileri' ve 'kopyasını al' diyerek alabilirsin.
- ❖ Çalışma alanını temizlemek için şekillere sağ tıklayıp klavyeden 'delete' tuşuna basabilirsin.

Sanal Dünyalarda Programlama Eğitimi Şekil 1

Kare Oluşturma

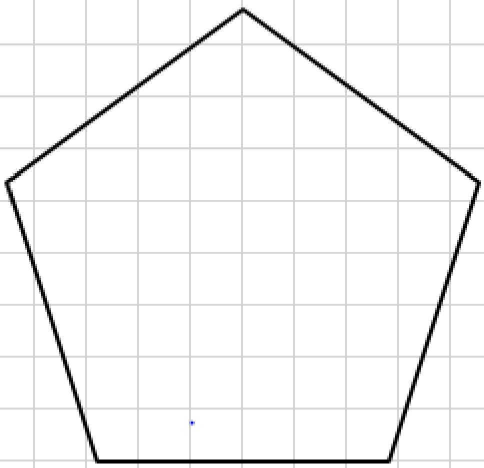
- ✓ Kare kenar uzunlukları birbirine eşit 4 kenardan oluşur.
- ✓ İç açıları toplamı 360 derece ve her bir iç açısı 90 derecedir.
- ✓ Dış açılarıda birbirine eşit olup her biri 90'ar derecedir.
- ✓ Kalem aracı ile 3 boyutlu kare çizmek için kalem aracını kullandıktan sonra robotu eşit uzunluklarda karenin dış açısı kadar döndürerek 4 tane kenar çizmek yeterlidir.
- ✓ Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin şekil oluşmuyorsa robotunu tekrar programlama ve dene. Bunu istediğin kadar yapabilirsin.
- ✓ Aşağıda kare ve özellikleri gösterilmiştir.



- ❖ Şekil çizen robotu envanterden çizim alanına yerleştiriniz. Eğer daha önce almadı iseniz robotun bir kopyasını masadan envanterine sağ tıklayıp 'ileri' ve 'kopyasını al' diyerek alabilirsin.
- ❖ Çalışma alanını temizlemek için şekillere sağ tıklayıp klavyeden 'delete' tuşuna basabilirsin.

Beşgen Oluşturma

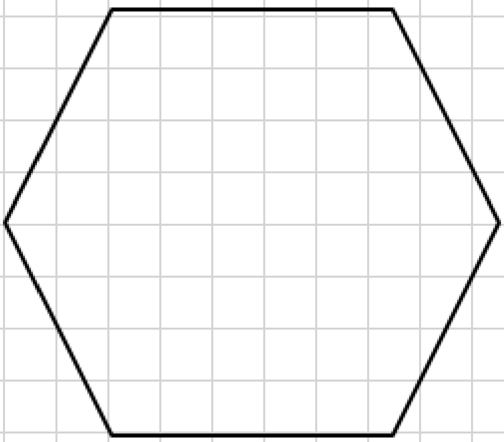
- ✓ Beşgen uzunlukları birbirine eşit 5 kenardan oluşur.
- ✓ İç açıları her biri 108 derecedir.
- ✓ Dış açılarıda birbirine eşit olup her biri 180 den iç açılarının çıkarımı ile bulunabilir.
- ✓ Kalem aracı ile 3 boyutlu kare çizmek için kalem aracını kullandıktan sonra robotu eşit uzunluklarda beşgenin dış açısı kadar döndürerek 5 tane kenar çizmek yeterlidir.
- ✓ Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin şekil oluşmuyorsa robotunu tekrar programlama ve dene. Bunu istediğin kadar yapabilirsin.
- ✓ Aşağıda beşgen ve özellikleri gösterilmiştir.



- ❖ Şekil çizen robotu envanterden çizim alanına yerleştiriniz. Eğer daha önce almadı iseniz robotun bir kopyasını masadan envanterine sağ tıklayıp 'ileri' ve 'kopyasını al' diyerek alabilirsin.
- ❖ Çalışma alanını temizlemek için şekillere sağ tıklayıp klavyeden 'delete' tuşuna basabilirsin.

Altıgen Oluşturma

- ✓ Altıgen uzunlukları birbirine eşit 6 kenardan oluşur.
- ✓ İç açıları her biri 120 derecedir.
- ✓ Dış açılarıda birbirine eşit olup her biri 60'şar derecedir.
- ✓ Kalem aracı ile 3 boyutlu kare çizmek için kalem aracını kullandıktan sonra robotu eşit uzunluklarda altıgenin dış açısı kadar döndürerek 6 tane kenar çizmek yeterlidir.
- ✓ Robotunu Scracth programı ile programla ve çalışmasını dene, istediğin şekil oluşmuyorsa robotunu tekrar programlama ve dene. Bunu istediğin kadar yapabilirsin.
- ✓ Aşağıda altıgen ve özellikleri gösterilmiştir.



- ❖ Şekil çizen robotu envanterden çizim alanına yerleştiriniz. Eğer daha önce almadı iseniz robotun bir kopyasını masadan envanterine sağ tıklayıp 'ileri' ve 'kopyasını al' diyerek alabilirsin.
- ❖ Çalışma alanını temizlemek için şekillere sağ tıklayıp klavyeden 'delete' tuşuna basabilirsin.

APPENDIX M

TASK CARDS

Sorunlu Kasaba Sorumlu Vatandaş

Kırmızı

Görev Kartları

Adı Soyadı: ………………………………………

Görev No: 1

Görev Adı: Kasaba hikayesini oku ve kaskını tak

Görev Tanımı: Arkadaşın ile birlikte dönen kutucuklara dokunarak kasaba hikayesini oku ve görevlerini anla.

Uyarılar:

- Kaskını takmayı unutma.
- Her zaman kendi rengindeki görevleri tamamla!

Görev No: 2

Görev Adı: Nehir üzerine köprü oluşturma

Görev Tanımı: Yeşilırmak üzerindeki köprü doğal afet sonucu yıkılmıştır. Bu köprünün kasaba halkının kullanabilmesi için tekrar inşa edilmesi gerekiyor. Oluşturman gereken köprünün uzunluğu 15 metre.

Yönergeler: Kutuya dokunarak robotun bir kopyasını Envanterine al, Envanterden robotu belirtilen noktaya yerleştir ve sonra köprü oluşturan robotu programla. 15m uzunluğundaki köprüyü oluşturmak için kalem aracını kullandıktan sonra belirtilen uzunluk kadar robotunu ilerlet. Kalem rengini değiştirerek köprü rengini ayarlayabilirsin.

Uyarılar:

- Bazen robot düzgün çalışmıyor olabilir, bu durumda robotu tekrar yerine getir ve tekrar çalıştır.

Görevi nasıl ve ne kadar tamamladın?

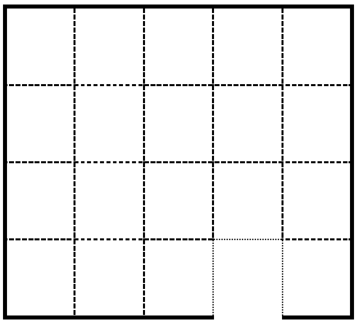
Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 3

Görev Adı: Tavuk kümesi için duvar oluşturma

Görev Tanımı: Eve alınan yeni tavuklar için bahçe içerisine tavuk kümesi duvarlarını oluştur.

Yönergeler: Dönen kutuya dokunarak duvar oluşturan robotun bir kopyasını al. Sonra robotu belirtilen alana yerleştir, yönünü ayarla ve aşağıdaki plan çerçevesinde kümes duvarını oluştur. Bunu yaparken giriş için 1 metrelik bir kapı boşluğu bırakmayı unutma. Duvar oluşturmak için kalemi bastır komutunu kullan ve kapı boşluklarında ise kalemi kaldır komutunu kullanmalısın.



Uyarılar:

- Kapı için boşluk bırakmayı unutma.
- Bazen robot düzgün çalışmıyor olabilir, bu durumda robotu tekrar yerine getir ve tekrar çalıştır.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 4

Görev Adı: Dönen kapı

Görev Tanımı: Kapısı olmayan bakkal dükkanı için avatarın çarpması durumunda otomatik olarak dönmeye başlayan bir kapı yapman isteniyor. Avatarın kapıya çarptığında kapının bir miktar dönmesi, çarpma bitince ise durması gerekiyor.

Yönergeler: İlk önce kapı için gerekli malzemeyi kutuya dokunarak al, sonra envanterden döner kapıyı bulup yerine yerleştir.

Şimdi kodlamaya başlayabilirsin. Kodlarken avatarın kapıya her çarpması durumunda kapı belli bir miktar dönsün. Böylelikle dönen bir kapı oluşturmuş oluyorsun. Kapının, avatarın çarpması bittikten sonra durması lazım.

Uyarılar:

- Kapının yönünü değiştirme.
- Kapıya çarpma komutu ekle.

Görevi nasıl ve ne kadar tamamladın?

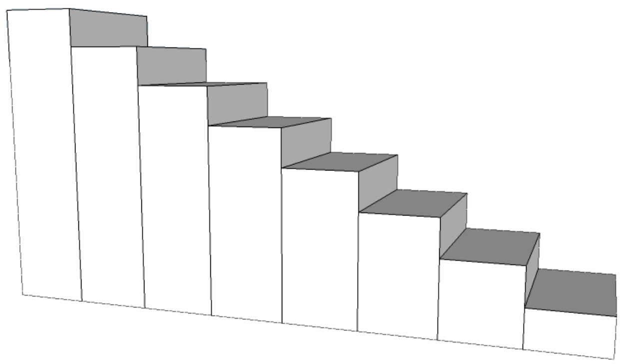
Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 5

Görev Adı: Merdiven oluşturma

Görev Tanımı: Yangın sonucu hasar gören kırmızı evin basketbol sahası tarafındaki girişi için merdivenlerini tekrar oluşturman gerekiyor. Oluşturman gereken merdivenin basamak sayısı 7 ve her bir basamağının uzunluğu 1.75 m'dir. Evin girişinin zeminden yüksekliği ise 3.5 m'dir.

Yönergeler: Kutuya dokunarak merdiven oluşturan robotun bir kopyasını al. Robotu belirtilen noktaya yerleştir. Sonra merdivenin basamak sayısı kadar (7) devam eden döngü kurman gerekiyor. Her basamak oluşumundan sonra robotu bir önceki yerine getir ve sonra 0.5 metre kadar yukarı ve biraz eve doğru gidecek şekilde ayarla.



Uyarılar:

- Robotu her bir basamağı oluştururken kalemi bastır komutundan sonra 1.75 metre ilerlet.
- Basamağı oluşturduktan sonra geri dönüşlerde kalemi kaldırmayı unutma.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 6

Görev Adı: Otomatik açılan kapı

Görev Tanımı: Kapısı olmayan bu bina için otomatik açılan bir kapı yapman isteniyor. Avatarın kapıya 2 metre yaklaştığında kapının otomatik olarak açılması, uzaklaştığında ise tekrar kapanması gerekiyor.

Yönergeler: İlk önce kapı için gerekli malzemeyi kutuya dokunarak al, sonra envanterden kapıyı bulup alandaki yere yerleştir. Şimdi kodlamaya başlayabilirsin. Kodlarken kapının kapalı halini oluşturulduğum zaman kod bloğu içerisinde başlangıç konumum yap diyerek hafızaya alabilirsin, böylelikle kapı hareket ettiğinde (açıldığında) tekrar eski yerine gelebilsin. Sonra sürekli tekrarla içinde avatar kapıya belirli bir metre yaklaştığında kapı ileri 3 metre gitmeli (kapı açıldı), 8 saniye sonra tekrar yerine gelecek (kapı kapalı) şekilde ayarla. (başlangıç konumama git)

Uyarılar:

- Kapının yönünü değiştirme.
- Kapının konumunu ilk başta 'oluşturulduğum zaman' kod bloğu içerisinde hafızaya al.
- Kapı açıldıktan sonra kapanması için bir süre kodların çalışmasını beklet.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No:	7
Görev Adı:	Kaplumbağayı nehrin karşısındaki yuvasına taşı
Görev Tanımı:	Kapalı barınaktan kurtulmayı başaran bir kaplumbağa dönen kutu içerisine sığınmıştır. Bu kaplumbağayı kutuya dokunarak al ve sana eşlik edecek şekilde programlayarak nehrin karşısındaki yuvasına annesinin yanına bırak.
Yönergeler:	Kutuya dokunarak barınaktan kaplumbağayı al, sonra envanterine gelen kaplumbağayı barınağın yanına boş bir yere yerleştir. Kodlarken kaplumbağanın yönünün her zaman sana doğru olmasını sağla, sonra kaplumbağa ile arandaki mesafeyi sürekli kontrol et, eğer 2 metreden fazla ise kaplumbağayı 1 metre ileri doğru hareket ettir.
Uyarılar:	• Kaplumbağanın yönünün sana doğru olmasına dikkat et. • Arandaki mesafeyi sürekli kontrol et. 2 metreden fazla ise sana doğru gelmesini sağla. • Kaplumbağayı yuvasına bıraktıktan sonra seni takip eden kod dosyasını silmelisin.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No:	8
Görev Adı:	Kafe için sayaç
Görev Tanımı:	Mustafa Bey kafesine gelen toplam kişi sayısını merak ediyor ve bunun için senden yardım istiyor. Bu görevde yapman gereken kafeye gelen müşteri sayısını bulmaktır. Bunun için kapı önüne koyacağın bir nesneye avatarın her çarpması durumunda tanımlayacağın değişkenin değerini bir artırarak yapabilirsin. Değişken değerini nesne üzerinde göstermeyi unutma.
Yönergeler:	Bu görev için yapman gereken dükkan kapısı önüne paspas boyutunda bir şekil oluşturmaktır. Bu şekli tam kapının önüne gelecek şekilde yerleştir ki avatarlar gelirken çarpsın. Ancak çok fazla yükseğe yerleştirme bu durumda avatarlar giremez. Daha sonra kapıdan gelip nesneye çarpan müşteri sayısını tutmak için bir değişken tanımla ve bu değişken değerini avatarın her çarpması durumunda bir artır. Sonra değişken değerini kapıya koyduğun engel (şekil) üzerinde göster.
Uyarılar:	• Nesneyi ne çok yükseğe ne de çok alçağa yerleştir. Avatarların kafeye girerken nesneyi çarpmalarını sağla. • Avatarın nesneye her çarpması durumunda değişken değerini bir artır. • Değişken değerini nesne üzerinde göster.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 9

Görev Adı: Posta kutusu sayacı

Görev Tanımı: Çakır ailesinin evine akıllı posta kutusu eklenmesi gerekiyor ve bunun için senden yardım istiyorlar. Akıllı posta kutusunun özellikleri şöyle olacak. Avatar posta kutusuna her dokunduğunda posta kutusundaki mektup sayısını bir artıracaksın. Yalnız posta kutusu en fazla 20 mektup alabiliyor. Bu nedenle kutu içerisinde 20'den az mektup varken üzerinde 'Yer var' yazması, 20'den çok olduğu zaman ise üzerinde 'Posta kutum doldu. Boşaltmanız gerekiyor' diye uyarması gerekiyor. Ayrıca birde posta kutusunu sıfırlamak için bir sıfırlama butonu koyman gerekiyor. Avatar bu butona dokunduğunda posta kutusundaki mektup sayısının sıfırlanması gerekiyor.

Yönergeler: Gerekli malzemeleri kutuya dokunarak al. Posta kutusunu ve sıfırlama düğmesini ev sınırları içerisinde boş bir alana yerleştir.

Sonra posta kutusundaki mektup sayısı için bir değişken tanımla. Posta kutusunu avatar kutuya her dokunduğunda değişken değerini bir artacak şekilde kodla. Sonra mektup sayısı 20'den küçük olması veya fazla olması durumunu gerekli kodlar ile kontrol ettirerek duruma göre ilgili mesajları üzerinde yaz.

Çöp kutusundaki mektupları sıfırlamak için butona dokunulduğunda bir mesaj yayınlat ve kutu bu mesajı aldığında değişken değerini sıfıra eşitlemen gerekiyor.

Uyarılar:

- Oluşturulduğum zaman kod bloğu içerisinde değişken değerini sıfırla.
- Çöp kutusunu ve sıfırlama butonunu ayrı ayrı kodlaman gerekiyor.
- Çöp kutusuna dokunulduğunda değişken değerini bir artır ve içerisindeki mektup sayısını kontrol et.
- Sıfırlama butonuna dokunulduğunda çöp kutusuna mesaj gönder.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 10

Görev Adı: Ağır kutuyu hareket ettirme

Görev Tanımı: Bu görevde evin yaşlı çiftine yardım etmen gerekiyor. Çok fazla ağır olan kırmızı kutuyu yaşlı çift taşıyamadığı için bu kutuyu 0 ve 180 yönlerine hareket ettirmek için programlaman gerekiyor. Görev için ihtiyacın olan malzemeler döner kutuda. Malzemeleri aldıktan sonra boş bir alana ağır kutuyu ve taşıma butonlarını yerleştir. Ok tuşlarına her dokunulduğunda kutu 0.5 metre hareket etmeli.

Yönergeler: Gerekli malzemeler aldıktan sonra boş bir alana ağır kutuyu ve 2 tane ok butonunu yerleştir. Sonra ok tuşlarının yönünü 0 ve 180 yönüne bakacak şekilde ayarla. Sonra hangi ok tuşuna dokunulursa ağır kutuya o yönde gitmesi için mesaj gönder, ve kutu mesajı aldıktan sonra mesaj göre kutuyu yönlendir ve 0.5 metre hareket etmesi için gerekli kodu yaz.

Uyarılar:

- 0 ve 180 yönleri için 2 ayrı ok butonu yerleştir ve butonların yönlerini ayarla.
- Avatar hangi ok butonuna dokunursa ağır kutuyu o yöne dönecek ve 0.5 metre gidecek şekilde ayarla.
- Nesneler arasındaki iletişimi mesaj göndererek sağla (… haberi yayınla) ve al (when I receive … komut blokları ile)

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 11

Görev Adı: Harf oyunu

Görev Tanımı: Bu görevde ev sahibinin torunları için bir harf oyunu yapman gerekiyor. Bu harf oyununda yüzeylerinde harfler bulunan bir küp var. Görevin bu küpü torunların oynaması için programlamak. Oyun şu şekilde. Küpe, çocuklar her dokunduklarında farklı bir yüzü üste gelmeli ki çocuklar gelecek harfi tahmin etme oyunu oynayabilsinler.

Yönergeler: Gerekli malzemeyi aldıktan sonra envanterinden küpü masanın üstüne yerleştir. Sonra rastgele 1 ile 40 arasında seçilecek sayı kadar döngü kurarak küpü döngünün her adımında x, y ve z eksenlerinde 90'ar derece dönecek şekilde programla.

Uyarılar:

- Döngü için rastgele sayı üret ve döngüyü bu sayı kadar tekrarla.
- Küpü 3 eksende de döndür ve döndürürken döngünün her adımında 90'ar derece dönecek şekilde 3 eksende de döndür.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

Görev No: 12

Görev Adı: Lunaparktaki atlıkarıncayı döndür

Görev Tanımı: Bu lunaparkta eksik olan bir şey var: **atlıkarınca**. Bu görevde lunapark içerisine bir atlı karınca yerleştirmen isteniyor. Atlıkarınca ait bilgiler şu şekilde:

İlk başta atlı karıncanın sabit durması ve dokunulunca dönmeye başlaması gerekiyor.

Atlıkarınca sana hazır olarak verilecek, envanterine aldıktan sonra boş bir yere yerleştir ve gerekli kodu yaz.

Yalnız atlıkarıncanın ikinci dokunmada durması, bir sonraki dokunmada ise tekrar dönmesi lazım.

Yönergeler: Gerekli malzemeyi aldıktan sonra envanterinden atlı karıncayı lunaparkta boş bir yere yerleştir ve sürekli tekrarla kod bloğu içerisinde bir miktar dönecek şekilde ayarla.

Bir sonraki dokunmada durması için ise bir değişken tanımlayıp o değişkenin değerine göre döndürmeye başlatıp durdurabilirsin.

Ayrıca değişken değerini her dokunmada sırasıyla değiştirmen gerekiyor.

Uyarılar:

- Atlıkarıncanın ilk başta durması için oluşturulduğum kod bloğu içerisinde değişken değerini dönmemesi için ayarla.
- Döndürürken bir süre bekletmek için saniye komutunu kullan.

Görevi nasıl ve ne kadar tamamladın?

Yapmadım	Kendim yaptım	Takım arkadaşım ile birlikte yaptım	İkimizde uğraştık fakat yapamadık
☐	☐	☐	☐

APPENDIX N

LESSON PLAN

In this section, lesson plans of a 15-hours course are presented session by session. There should be three sessions on a day which lasted for 45 minutes with a 15-minutes break. In the first six sessions of the course, students study in the first island individually, and in the last six sessions, they should study with one of their peers as pairs in the second island. Each group will study in a separate island and they will try to solve the problems of this island by completing the tasks assigned them. A repository of resources would be presented to students via a web site. Students should be informed about this web site and promoted to share their comments on the related posts.

First Day / Session 1

Topic: Introducing programming and S4OS

Learning Objectives: By participating this session, students will be able to:

- define what programming is
- explain the importance of programming
- give examples about the use of programming
- use S4OS

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Presentation about programming and S4OS, videos about programming, and projector.

Activities:

- Present what is programming and the importance of programming
- Explain why programming is important to learn, give examples about the use of programming in daily life
- Watch the video on www.youtube.com/embed/nKIu9yen5nc
- Explain the S4OS and how to use it such as building code and transforming it

Feedback and Assessment:

- Ask students to give example use of programming in their daily life and their experience if any.

Wrap-Up the Session

- Summarize the activity
- Remind students that they should visit the social area in 3D environment which contains posters and videos about programming

First Day / Session 2

Topic: First login to 3D environment and training on generic skills

Learning Objectives: By participating this session, students will be able to:

- use the Imprudence viewer
- perform basic skills in 3D environment

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, and projector.

Activities:

- Distribute students' login names and passwords, demonstrate how to login to 3D environment via the viewer and first login to 3D environment
- Present the important panels and their use on Imprudence viewer
- Present the overview of the first island and its areas
- Train students on generic skills in 3D environment such as movement, camera control, communication, building basic objects, and inventory

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Ask students about their first impression about the 3D environment.

Wrap-Up the Session

- Summarize the use of important functions on the viewer.
- Remind students that they should keep in mind their login names, passwords
- Remind the rules of 3D environment

First Day / Session 3

Topic: Customizing avatars

Learning Objectives: By participating this session, students will be able to:

- customize their avatars

Target Learners: 4th, 5th and 6th graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, and projector.

Activities:

- Demonstrate how to customize the avatars
- Go to Avatar Center and choose one of the avatars and some outfit
- Show students how to change the appearance of their avatars and the outfits
- Show how to move in different modes and how to apply gestures on their avatars

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Ask students to explain the importance of avatar appearance for themselves
- Ask student to take a photo of their avatar and share it on the web site.

Wrap-Up the Session

- Summarize the activity
- Remind students that they should customize their avatar in their free time

Second Day / Session 4

Topic: Building the first robot and train it to follow routes

Learning Objectives: By participating this session, students will be able to:

- build the first 3D robot and edit it
- build code to move the robot on the routes via S4OS
- attach script to the robot by transferring code built via S4OS
- use the movement and event handler codes on S4OS
    - move … meters
    - turn …. degrees
    - when I am touched

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

<u>Teaching-Learning</u>

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector and activity sheet for Building 1 in Appendix K per student.

Activities:

- Demonstrate how to build 3D robot and edit its features
- Go to Robot Creation Center and build a simple robot
- Go to Building 1 on the Robot Training Center, distribute the sheet for the activities in Building 1 to each student and explain what they will do in this activity.
- Complete the first and second activity together with students
- Firstly, explain the activity and then allocate some time for students to think about the code to do the activities, and complete the activities for students on their own. Finally, demonstrate how to complete the activities.

Feedback and Assessment:

- Watch students and help anyone encountering any problem

- Ask students to rebuild their code if their robots would not follow the routes on the activities
- Ask students to mark the activities they completed on their sheets and collect them at the end of the day to follow students' progress

Wrap-Up the Session

- Summarize the activities, codes and their function
- Remind students about the use of S4OS and its appropriate use

Second Day / Session 5

Topic: Train the robot to follow the routes on Building 2, 3 and 4

Learning Objectives: By participating this session, students will be able to:

- build code to move the robot on the routes via S4OS
- attach script to the robot by transferring code built via S4OS
- use the movement and event handler codes on S4OS
    - move … meters
    - turn …. degrees
    - when I am touched

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector and activity sheet for Building 2,3 and 4 in Appendix K per student.

Activities:

- Go to Building 2 on the Robot Training Center, distribute the sheet for the activities in Building 2 to each student. Firstly, explain what they will do in this activity and then allocate some time for students to think about the code to do the activities, and complete the activities for students on their own. Finally, demonstrate how to complete the activities.
- Go to Building 3 on the Robot Training Center, distribute the sheet for the activities in Building 3 to each student. Firstly, explain what they will do in this activity and then allocate some time for students to think about the code to do the activities, and complete the activities for students on their own. Finally, demonstrate how to complete the activities.
- Go to Building 4 on the Robot Training Center, distribute the sheet for the activity in this building to each student. Firstly, explain what they will do in this activity and then allocate some time for students to think about the code to do the activity, and complete for students on their own. Finally, demonstrate how to complete the activity.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Ask students to rebuild their code if their robots would not follow the routes on the activities
- Ask students to mark the activities they completed on their sheets and collect them at the end of the day to follow students' progress

Wrap-Up the Session

- Summarize the activities, codes and their function
- Remind students about the use of S4OS and its appropriate use

<u>Second Day / Session 6</u>

Topic: 3D object construction via programming

Learning Objectives: By participating this session, students will be able to:

- build code to create basic 3D objects via S4OS
- attach script to the given robot by transferring code built via S4OS
- use pen, event handler and loop codes on S4OS
    - pen down
    - pen up
    - change pen color by ….
    - when I am touched
    - repeat …

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector and activity sheets in Appendix L per student.

Activities:

- Go to 3D Geometric Shape Creation Center and take a copy of the robot which will be used for creating shapes. Locate the robot to the related area on this center. Distribute the sheet for the activities in this session to each student.
- Create the first shape, triangle, together with students, and demonstrate how to use pen code to create 3D shapes.
- Then, explain the other shapes, which are square, regular pentagon and hexagon, and their features to the students.
- Allocate some time for students to think about the code to create other shapes, and create them for students on their own. Finally, demonstrate how to create the shapes at the end.
- Introduce students the loops in programming, and create one of the shapes by using repeat code block on S4OS. Allow students to realize how and why to use repeat code.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Ask students to rebuild their code if shapes would not be created as desired.
- Ask students to take a snapshot of their artefacts and promote them to share those artefacts on web site.

Wrap-Up the Session

- Summarize the codes and their function
- Remind students about the use of repeat code block which makes the process of building code easier and briefer.

Third Day / Session 7

Topic: Introducing the second island, defining the roles and colors

Learning Objectives: By participating this session, students will be able to:

- learn about their roles, assigned colors and team members
- recognize the second island and what to do on this island

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and cover story sheet in Appendix B per group.

Activities:

- Teleport students to the second island and describe it.

- Define the groups and assign two members to each group. Then assign a color, blue or red to each members of the group.
- Distribute cover story of the town to each group, and explain what is expected from students to do as a group on this island.
- Explain the mission of the students on the island and their role as they would work in the town along with their teammate as builders. Each team is responsible for solving the problems of the town by completing a number of assigned tasks on this island.
- Explain that there is a total of 24 tasks for each team on this island, and each team member have 12 tasks to complete. Each task is numbered from 1 to 12 and colored as either red or blue. Group members are tasked with completing all of the tasks with their corresponding assigned color in accord with the teammate.
- Explain the overall tasks and their location to the students.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Ask students about their first impression about the tasks, roles, mission and second island

Wrap-Up the Session

- Summarize the tasks on this island
- Remind students that they will study in groups while completing the tasks on this island and therefore they could get help from each other.

<u>Third Day / Session 8</u>

Topic: Completing the tasks 1-3

- Task 1: Read the story
- Task 2: Build a bridge on the river
- Task 3: Build the wall

Learning Objectives: By participating this session, students will be able to:

- build an algorithm for a basic programming task

Target Learners: 4th, 5th and 6th graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and task cards numbered 1, 2 and 3 in Appendix M for each student.

Activities:

- Distribute the colored binders according to assigned colors of students and ask them to write their names to the binder.
- In the first task, ask students to take the blue or red helmet according to their assigned color in 3D environment and put on it to their avatars. Remind them to follow the colored waymarks which bring them to their assigned tasks in turn.
- Then, ask students to go to location of Task 2. Explain this task as “The bridge over the Yeşilırmak River has been partially destroyed due to natural disaster. You need to rebuild the bridge for the people living in the town.”
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
    - Allocate some time for students to understand the task and think about how to do that with their team members.
    - Demonstrate how to take the necessary materials for completing this task since this is the first time for them.
    - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask

students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

- Ask students to go to location of Task 3. Explain this task as "You need to build the walls of a shelter inside the garden for a newly arriving pet."
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
    - Allocate some time for students to understand the task and think about how to do that with their team members.
    - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day
- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the tasks, codes for completing them and their functions
- Promote the group study

<u>Third Day / Session 9</u>

Topic: Completing the tasks 4-5

- Task 4: Build a revolving door
- Task 5: Build a staircase

Learning Objectives: By participating this session, students will be able to:

- use pen and event handler code
    - pen down
    - pen up
    - change pen color by ….
    - when I am touched
- use loop code
    - repeat …

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

<u>Teaching-Learning</u>

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and tasks cards numbered 4 and 5 in Appendix M for each students.

Activities:

- Distribute the colored binders of each student.
- Ask students to go to location of Task 4. Explain this task as "You need to build a revolving door at the market. When the avatar collides the door, the door should revolve. The door should stop revolving when the collision of avatar ends."
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.

- Allocate some time for students to understand the task and think about how to do that with their team members.
    - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.
- Ask students to go to location of Task 5. Explain this task as "You need to rebuild the fire damaged staircase. The staircase should have … steps and each step should be ... meters."
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
    - Allocate some time for students to understand the task and think about how to do that with their team members.
    - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day
- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the tasks, codes for completing them and their functions
- Promote the group study

<u>Fourth Day / Session 10</u>

Topic: Completing the tasks 6-7

- Task 6: Build an automatic door
- Task 7: Move the turtle to its home across the river

Learning Objectives: By participating this session, students will be able to:

- use the forever loop, conditional statements and boolean logic
    - forever
    - if …
    - if … else ….

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

<u>Teaching-Learning</u>

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and tasks cards numbered 6 and 7 in Appendix M for each students.

Activities:

- Distribute the colored binders of each student.
- Ask students to go to location of Task 6. Explain this task as "You need to build an automatic door for the building. When the avatar reaches within two meters proximity of the door, the door should open automatically. The door should close when the avatar's proximity to the door exceeds two meters."

- For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
  - Allocate some time for students to understand the task and think about how to do that with their team members.
  - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.
- Ask students to go to location of Task 7. Explain this task as “A turtle managed to escape from jail and is sheltering in a rotating box. Take the turtle by touching the box. Code the turtle so that the turtle accompanies you to its home across the river.”
  - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
  - Allocate some time for students to understand the task and think about how to do that with their team members.
  - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day

- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the tasks, codes for completing them and their functions
- Promote the group study

Fourth Day / Session 11

Topic: Completing the tasks 8-9

- Task 8: Build a counter for cafe
- Task 9: Smart mail counter

Learning Objectives: By participating this session, students will be able to:

- use variable, change and view the value of variables, conditional statements and coordination and synchronization between objects
    - Make a variable
        - change … by ….
        - view variable ….
        - set … to ….
    - if … else ….
    - broadcast and when I receive ….

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and tasks cards numbered 8 and 9 in Appendix M for each students.

Activities:

- Distribute the colored binders of each student.
- Ask students to go to location of Task 8. Explain this task as “The owner of the shop wants to know how many customers visited the shop and asks for your help. You need to record the number of visitors and then show the number.”
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
    - Allocate some time for students to understand the task and think about how to do that with their team members.
    - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.
- Ask students to go to location of Task 9. Explain this task as “A family want a smart mailbox for their home with the following characteristics. Each time the avatar touches the mailbox, the number of letters in the mailbox should increase. The mailbox can only hold a maximum of 20 letters. When there are fewer than 20 letters in the mailbox, return the message, <The mailbox has space> When there are equal or more than 20 letters, return the message, <The mailbox is full, please empty it> .”
    - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
    - Allocate some time for students to understand the task and think about how to do that with their team members.

- Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day
- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the tasks, codes for completing them and their functions
- Promote the group study

<u>Fourth Day / Session 12</u>

Topic: Completing the tasks 10-11

- Task 10: Move the heavy box
- Task 11: Build a letter game

Learning Objectives: By participating this session, students will be able to:

- use random numbers, loop, coordination and synchronization between objects codes
    - pick random … to …
    - broadcast and when I receive ….

- repat …

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and tasks cards numbered 10 and 11 in Appendix M for each students.

Activities:

- Distribute the colored binders of each student.
- Ask students to go to location of Task 10. Explain this task as “You need to help an elderly couples on this task. Since the couples had difficulty in moving the heavy box, you should program the box to move both sides when the related button is touched.”
  - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
  - Allocate some time for students to understand the task and think about how to do that with their team members.
  - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.
- Ask students to go to location of Task 11. Explain this task as “You need to build a letter game for the grandchild of the house owner. A cube has a letter, from A to F, on each face. The cube should randomly rotate when touched.”

- For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
- Allocate some time for students to understand the task and think about how to do that with their team members.
- Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day
- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the tasks, codes for completing them and their functions
- Promote the group study

<u>Last Day / Session 13</u>

Topic: Completing the task 12

- Task 12: Revolve a funfair carrousel

Learning Objectives: By participating this session, students will be able to:

- use variable, conditional statements codes and change variable values
  - set a variable
  - change the value of variable
  - rotate the object according to value of the variable

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

Teaching-Learning

Materials and Resources: Computer for each student and teacher which are ready to login 3D environment, projector, and task card numbered 12 in Appendix M for each students.

Activities:

- Distribute the colored binders of each student.
- Ask students to go to location of Task 12. Explain this task as "You are asked to help revolve a funfair carrousel. Firstly, the carrousel should stop and it should start revolving e around when touched by an avatar, and it should stop when touched again. This should continue in this order."
  - For more information about this task, allocate some time for students to read more information about the name, definition, instruction and warnings about this tasks on the relevant card in the binder.
  - Allocate some time for students to understand the task and think about how to do that with their team members.
  - Ask students to build codes and complete the task on their own by getting help with their teammate for a while. After a certain time, ask students how they completed it and you could want one of the students to explain the codes. At the end of the task, ensure that everyone could complete the tasks correctly.

Feedback and Assessment:

- Watch students and help anyone encountering any problem
- Explain students that they could watch the Expert Videos on 3D environment for viewing the steps of relevant tasks, and they could also get feedback from their peers.
- Ask students to put on marks on their task cards that fits best for them. Collect the task cards at the end of the day
- Ask students to take a snapshot of their artefacts at the end of the tasks and save them to a portable hard disk

Wrap-Up the Session

- Summarize the task, codes for completing it and their functions
- Promote the group study

<u>Last Day / Session 14</u>

Topic: Summary of the overall course

Learning Objectives: By participating this session, students will be able to:

- remember the overall tasks, and basic concepts of programming learned throughout the course

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

<u>Teaching-Learning</u>

Materials and Resources: Presentation about the overall course, basics of programming and S4OS, and projector.

Activities:

- Present the basics of programming, codes on S4OS, and their example use in the previous sessions
- Ask students how to transfer what was learned into real life situations

Feedback and Assessment:

- Ask students to give example use of programming concepts in different situations.

Wrap-Up the Session

- Mention about the importance of programming in today's life

<u>Last Day / Session 15</u>

Topic: Ideas of students in programming for the future

Learning Objectives: By participating this session, students will be able to:

- express their future ideas and expectations at programming
- get their certificate based on their attendance to overall sessions

Target Learners: 4^{th}, 5^{th} and 6^{th} graders (learners first meeting with programming)

Duration: 45 minutes

Number of students: 8-16

<u>Teaching-Learning</u>

Materials and Resources: Certificate of attendance based on the students' attendance to overall sessions.

Activities:

- Ask students about their ideas and expectations about programming after the course
- Ask students about how this course would be enhanced better
- Explain the use of programming in different situations, such as robot and mobile programming
- Try to develop a passion for students to learn programming in advanced levels in different environments and platforms
- Distribute the students' certificate of attendance

Feedback and Assessment:

- Gather the task cards of students and investigate the marks

Wrap-Up the Session

- Remind students that they should learn programming in different levels in their future life via different platforms

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	: Battal, Ali
Nationality	: Turkish (TC)
Phone	: +90 312 210 75 26
Email	: albatt@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
BS	Selçuk University, Computer Education and Instructional Technology (Hons)	2007
High School	Karatay S.D.M.P. Anadolu High School, Konya	2003

WORK EXPERIENCE

Year	Place	Enrollment
2010 -	METU - Computer Education and Instructional Technology	Research Assistant
2009 - 2010	Trakya University – Computer Education and Instructional Technology	Research Assistant
2007 - 2009	Necatiye İlköğretim Okulu, Edirne	ICT Teacher Ministry of National Education

FOREIGN LANGUAGE

Advanced English

PUBLICATIONS

ARTICLES

Battal, A., Çağıltay, K. (2015). Effects of Usage Frequency and Department on Usability Issues with the Online Examination System (OSSI). *Mersin University Faculty of Education Journal*, 11 (3). p. 752-763. DOI: 10.17860/efd.19683

Ugras, G., Uzun, A. M., **Battal**, A., & Kayaduman, H. (2016). An Examination of Instructional High School Chemistry Videos on EBA Portal in terms of Nine Events of Instruction. *Participatory Educational Research*, 3(1), 66–78.

Pala, F. K., Arslan, H., **Battal**, A., & Özdinç, F. (2017). Teachers' Opinions on the Usability of Interactive Board Software: Antropi Teach. *SDU International Journal of Educational Studies*, 4(1), 1-11

CONFERENCE PROCEEDINGS

Çuhadar, C. & **Battal**, A. (2010). Perspectives of the academic staff regarding the instructional use of information and communication technologies. *4th International Computer & Instructional Technologies Symposium*, Konya, Turkey.

Battal, A., Taşdelen A. & Tümer A. (2010). An assessment and evaluation model for a web based teaching: OSSI. *4th International Computer & Instructional Technologies Symposium*, Konya, Turkey.

Battal, A., & Kayaduman, H. (2013). Effectiveness of instructional videos on the eba portal from the perspective of teachers and students: A Case Study. *7th International Computer & Instructional Technologies Symposium*. Erzurum, Turkey.

Battal, A., & Tokel, T. (2013). Comparing two open source virtual worlds in terms of educational purposes:OpenSim And Open Wonderland. *7th International Computer & Instructional Technologies Symposium*. Erzurum, Turkey.

Pala, F. K., Arslan H., **Battal**, A., & Özdinç, F.. (2015). Perspectives of teachers towards to usability of interactive white board software, ANTROPI. *3th International Instructional Technologies and Teacher Education Symposium*, Trabzon, Turkey.

Battal, A., & Tokel, T. (2016). A new tool in programming education for kids: Virtual Worlds. 10*th International Computer & Instructional Technologies Symposium*. Rize, Turkey.

Battal, A., & Tokel, T. (2017). Investigating the effects of using virtual worlds in introductory programming education on students' problem solving skills. *IVth International Eurasian Educational Research Congress*. Denizli, Turkey.

Kadırhan, Z., Gül, A., & **Battal**, A., & Tokel, T. (2017). An investigation of programming education from the perspective of ICT teachers. *IVth International Eurasian Educational Research Congress*. Denizli, Turkey.

CONFERENCE PRESENTATIONS

Battal, A., & Çağıltay, K. (2014). The Usability Study of an Online Examination System (OSSI) Paper presented at the annual meeting of the Association for Educational Communications and Technology (AECT): Jacksonville, FL.

Kayaduman, H., Uzun, A. H., **Battal**, A., Ugras, G. (2014). Examination of Instructional High School Chemistry Videos on EBA Portal in terms of Nine Events of Instruction. Paper presented at the annual meeting of the Association for Educational Communications and Technology (AECT): Jacksonville, FL.

RESEARCH PROJECTS

“Information and Comunication Technologies in teaching Foreign Language Teachers: Blog Based Approach”, TÜBAP(Trakya University, Scientific Research Center), Researcher in Project, Project No:2009-158, 2009-2011.

“Effective use of virtual worlds in teaching introductory programming for children: a case study”, ODTU – BAP – LTP (METU - Scientific Research Center, Graduate Dissertation Project), Researcher in Project, Project No: BAP-07-02-2014-007-546.

PANEL DISCUSSIONS

Invited speaker at the panel related to cyber-crimes and cyber-bullying for kids, (2016, April), 19th Annual Internet Week Activities, METU – Development Foundation (DF) Schools, Ankara.

TEACHING EXPERIENCE (TEACHING ASSISTANTSHIP)

CEIT 111, Information Technology in Education I , CEIT, METU, 2011 - 2012

CEIT 313, Use of Operating Systems , CEIT, METU, 2011 - 2012

CEIT 390, Database Management Systems , CEIT, METU, 2011- 2012

CEIT 436, Project Management and Development , CEIT, METU, 2011 - 2012

CEIT 411, School Experience , CEIT, METU, 2012 - 2013

CEIT 321, Foundations of Distance Education , CEIT, METU, 2012 - 2013

CEIT 112, Information Technology in Education II, CEIT, METU, 2012 - 2013

CEIT 207, Design and Use of Instructional Material , CEIT, METU, 2013 - 2014

CEIT 210, Programming Languages I, CEIT, METU, 2014 – 2015

CEIT 380, Computer Education Teaching Methods I, CEIT, METU, 2014 – 2015

CEIT 100, Computer Applications in Education, ESE, METU, 2014 – 2015

CEIT 382, Computer Education Teaching Methods II, CEIT, METU, 2016 – 2017