Gaming in Second Life via Scratch4SL: Engaging high school students in programming courses

Nikolaos Pellas, University of the Aegean

Available at: https://works.bepress.com/nikolaos-pellas/13/
The present paper below is an original authors’ working draft version that led to an article publication. A reference in APA settings to this work should always be done using the following citation:


This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.
Gaming in Second Life via Scratch4SL: Engaging high school students in programming courses

Abstract

While pedagogical and technological affordances of three-dimensional (3D) multi-user virtual worlds in various educational disciplines are largely well-known, a study about their effect on high school students’ engagement in introductory programming courses is still lacking. This case study presents students’ opinions about their participation in a 3D multi-user game-like environment, by harnessing Second Life in combination with the two-dimensional (2D) programming environment of Scratch4SL. Following a blended instructional format (face-to-face in a computer laboratory and supplementary online courses), fifty-six (56) students utilizing Scratch4SL participated in this study, with a view to reduce the “steep learning curve” created during their first-time entrance into Second Life. This study identifies Papert’s theory of Constructionism as a potentially appropriate theoretical foundation for the development of an instructional framework, in order to assist students to coordinate and manage learning materials with other teammates, using their computational thinking skills in collaborative problem-based programming tasks. The study findings based on a mixed-method research (a close-ended questionnaire and an open-ended interview) indicated the effectiveness of this “constructionist-oriented” instructional process for students’ engagement to acquire or empower social, cognitive, higher-order and computational thinking skills. Educational implications and recommendations for future research are also discussed.

Keywords: 3D environment, Computational thinking skills, Constructionism, Programming, Student engagement, Second Life, Scratch4SL
Introduction

Learning to think “computationally” has long been recognized as a significant thematic area in Computer Science. The contemporary research on computational thinking skills has focused on how well-designed instructional settings for programming courses can influence student engagement (Grover & Pea, 2013). There is broad agreement on the importance of learning and using computational thinking skills, from high school students (10-15 years old) to people who may never learn how to program using commercial languages (Lye & Koh, 2014; Koorsse et al., 2015). Deciding which computational thinking skills should be taught in a specific instructional format is also an important research challenge for computer scientists.

Introductory programming courses have specific computational rules that demand an understanding of basic computational concepts and require students’ ability to think, to communicate these concepts to others (students and teachers), and finally to propose a solution in problem-based settings. Following Wing’s (2006) statement about the significant role of using computational thinking skills in computer-assisted programming activities, it is necessary to recognize what circumstances contribute to problem-solving learning situations. She has also pointed out that the acquisition of these skills is of great importance for all people who want to solve problems, design systems and understand human behavior, by drawing on concepts which are fundamental to Computer science (Wing, 2006). This approach is adopted in introductory programming courses, since users need to employ higher order and computational thinking skills. More specifically, these are (Kim et al., 2012; Pellas, 2014): (a) deconstruction and comprehension of the main problem (logical reasoning of a process to solve and debug a problem); (b) analysis of the problem-based learning situation (the development of a proposed solution from the initial
concept to the project completion); and (c) synthesis of programming skills (the use of logical thinking skills for the synthesis of concurrent programming structures to complete programming structures, which can be executed simultaneously). These skills are fundamental for understanding the added value of using computational thinking in problem-based learning situations.

To teach computational thinking skills to a wide range of high school students who start to think about real-life’s problems logically, a digital programming environment that can engage them into motivating and user-friendly activities is generally required (Kalelioglu, 2015; Koorsse et al., 2015). To date, different digital-oriented environments for teaching computer science concepts and computational thinking skills in a school context have been utilized (Koorsse et al., 2015; Lye & Koh, 2014; Paliokas et al., 2011). However, visually-rich environments that are utilized by students who do not have previous programming experience to acquire and develop these skills are also observed. Kalelioglu (2015) has pointed out that the lack of operational sequence, the use of invalid programming language syntax and the poor utilization of problem solving strategies via digital-oriented environments can turn students’ attempts into a chaotic endeavor.

Recent studies (Koorsse et al., 2015; Lye et al., 2014; Pellás, 2014) have already noted some interesting features that may distract students’ engagement and participation in introductory programming courses. Some of the most crucial are as follows: a) students’ misconceptions on how they can better understand what they have learned with the purpose of tackling several programming concepts; and b) management responsibilities that students always have by using two dimensional (2D) or three dimensional (3D) visual environments. Students should handle learning materials of programming structures during pre-defined teaching school hours, and
after that they need to coordinate learning tasks to propose a solution to a problem within limited time; c) maintenance cost of a digital-oriented programming environment is sometimes prohibitive for state schools; and last but not least d) lack of an instructional design framework that should be followed in formal or informal settings (i.e. after-school programs). Constructivist-oriented learning theories can cause serious challenges regarding how students can more easily organize their actions, understand the learning objectives, and finally present their solutions in furtherance of achieving the best outcomes.

In the last ten years, many 3D visual or 2D text-based environments for programming courses that are free of charge and available for the development of students’ computational thinking have been used. In spite of utilizing visual programming environments, such as Scratch, Alice, Kodu or AgentSheets, users seemed not to become engaged with computer programming concepts from the outset with regard to create something meaningful. This situation initially creates a “steep learning curve” that cannot facilitate students’ engagement in problem-based tasks (Girvan et al., 2013). This “curve” also fails to assist novice programmers (high school students) with eliminating the obstacles of understanding how to use computational thinking skills. Due to a variety of programming activities that implemented through 2D or 3D visual environments, students started to participate in more complex problem-based learning situations, without a specific storyline or an instructional context. As a result, they needed to spend more time to customize and manipulate something, rather than learn how to program. Visual environments have not been designed to encourage the development of computational concepts and of course their transfer to more complicated tasks sometimes judged as not appropriate
(Howland & Good, 2015). This situation can negatively affect students’ engagement in introductory programming courses.

3D visual environments that use web-based transactions have successfully been introduced as a new dimension for programming skills acquisition. These environments are known as “3D multi-user virtual worlds”. Like other candidate learning platforms, 3D multi-user virtual worlds seemed to provide more meaningful and insightful aspects to (in-) formal instructional contexts, in contrast to 2D environments (Dalgaro & Lee, 2010; Pellias, 2014). Notable studies (Girvan et al., 2013; Pellias, 2014; Rico et al., 2011) have also described the positive aspects of learning experiences achieved by exploiting constructivist-oriented approaches. Students’ motivation and collaboration in highly engaging settings through game-like activities can be increased as well (Pellias, 2014). Taking advantage of 3D multi-user virtual worlds in introductory programming courses, users can improve their socio-cognitive and computational thinking skills through engaging in game-based learning tasks underpinned by Papert’s theory of Constructionism (Girvan et al., 2013; Pellias et al., 2013).

Despite the general acceptance of 3D multi-user virtual worlds in different educational tasks, the “steep learning curve” during students’ first-time entry has become the most crucial parameter that might hinder their participation and engagement. Unquestionably, many instructional technologists (Young et al., 2012; Xu et al., 2011) have already applied their educational scenarios in Second Life. It is a 3D interactive/persistent multi-user virtual world, in which users need to rent a virtual island (grid) with a minimal financial cost. Dickey (2005) has already noticed that the built-in tools of 3D multi-user virtual worlds can create a high-floor hurdle (steep learning curve) that high school students should overcome. The utilization of 2D tools
combined with constructionist learning approaches is proposed as solution to lowering these barriers and facilitating their engagement. Second Life provides a “high-floor” environment for introductory programming courses at high school level and this may negatively impact their first-time entry. Therefore, “low threshold” programming tools are needed in order to engage students easier in “wide walls/high floor” activities via Second Life (Girvan et al., 2013). Also notable are students’ obstacles for the development and enhancement of their computational thinking skills during the in-world tasks with their peers. Scratch4SL (Scratch for Second Life) is a free programming tool that can be used with Second Life. It provides an easy way for users to integrate new behaviors into virtual objects (primitives) and predict their interactions inside Second Life. Scratch4SL has generally low computational requirements (low-floor). The programming commands produced can directly affect a 2D “low-threshold” digital environment (Rosenbaum, 2008). On the one hand, this characteristic can become really useful for the reduction of the “steep learning curve” that is created when students are involved in complex learning tasks via a 3D multi-user virtual world. On the other, the inherent characteristics (adaptability, high representational fidelity and flexibility) of Second Life can assist instructional technologists and educators to create a 3D multi-user (low-cost) game-like environment, without additional financial cost.

Besides the above findings, a study to exemplify high school students’ engagement factors affecting their positive or negative perceptions of introductory programming courses is still lacking from the international literature. It is necessary for students firstly to learn how to think “computationally” and secondly to propose a solution in logical steps, before the execution of any programming concept. To cope with the dilemma of reducing the “steep learning curve”, it is crucial to engage
students on how to use a 3D programming game-like environment, and then to present how to figure out and then propose a solution.

Hence, the main hypothesis was whether the combination of Second Life and Scratch4SL for the creation of a 3D multi-user game-like environment can engage students to participate in introductory programming courses. The research question raised is focused on: How can a 3D multi-user game-like environment contribute to high school student engagement during their participation in introductory programming courses?

The purpose of this paper is to investigate the extent to which Scratch4SL can be used by high school students for creating programming scripts and for examining improvements in the expression of computational rules or concepts, by using this programming tool in a 3D multi-user game-like environment. This study identifies Papert’s theory of Constructionism as a potentially appropriate pedagogy for the development of an instructional framework and students’ assistance to coordinate and manage learning materials with other peers, in several programming tasks. Finally, this study presents findings of a descriptive case study carried out with fifty-six (n=56) high school students.

**Background**

**The use of a “constructionist-oriented” instructional framework**

Guided by Papert’s theory of Constructionism, a learning process can become more effective when students are actively involved in a process of constructing interactive and realistic objects (Pellas, 2014). Each student needs to create their own rules, and this learning process occurs as insightful for the assimilation of new experiences (Papert & Harel, 1991). Thus, in parts of all these experiences, students are engaged by manufacturing activities with specific meaning. In this way,
Constructionism as a theoretical foundation is connected with experimental learning, and users’ advanced psychomotor skills in high-fidelity representations can be enhanced more easily via 3D visual environments (Kommers, 2003). Li et al. (2013) have mentioned that 3D game-like learning environments influenced by Constructionism are able to facilitate students’ activities and foster in-depth learning strategies. The same authors have noticed that Constructionism can help students who have low programming background to participate in several courses and even more easily become accustomed to a game-like environment. A 3D “game-like” environment is a simulation-based learning environment circumscribed by a prototype game with playable characteristics inspired by serious games or game-based environments.

The contemporary literature (Li et al., 2013; Pellas, 2014) has suggested Constructionism as a theoretical foundation that can become really meaningful in empowering instructional game-like learning settings, in order to: (a) engage students in a digital-oriented environment, in which they can easily study in playful and experimental learning activities, and (b) facilitate exploratory or practice-based tasks, in which students actively organize their materials, knowing their obligations and reflecting upon their ideas to complete tasks. This study hypothesizes that Constructionism as a theoretical foundation in an instructional framework can enhance students’ learning experiences in game-like settings.

**Instructional benefits of using 3D multi-user virtual worlds for programming courses**

Previous studies (Brennan & Resnick, 2013; Kafai & Burke, 2013) have pointed out the useful perspectives for students’ engagement in a digital environment
not only to learn actively, but also to collaborate with their peers to exchange ideas and communicate so as to provide a solution in more complex learning activities. For instance, Zhang et al. (2014) has proved that in Storytelling Alice, students learned fundamental programming concepts in the context of creating animated movies and video games.

Nevertheless, some research studies (Carbonaro et al., 2010; Gross & Kelleher, 2010) have proven the effect of teaching programming with the purpose of increasing students’ higher order thinking, such as computational thinking, critical thinking, creative thinking or problem solving skills. Other instructional technologists (Shadiev et al., 2014; Voogt et al., 2015) have focused their attention on how effectively a learning process can become and assist a student to start thinking “computationally” in collaborative instructional settings. Harlow and Leak (2014) have considered as particularly important not only the role of collaboration among users in programming courses, but also the utilization of an instructional approach to become more effective than the conventional (in-class) that is based on teacher’s lectures. Fields et al. (2014) have found that asynchronous forms of communication using Scratch 2.0 can positively affect the understanding of basic programming concepts. In contrast, Zuckerman et al. (2009) have found that the online platform of Scratch (2.0) did not meet the needs of users for communication, inasmuch as collaboration among students using only asynchronous communication tools have not satisfied and engaged them to participate extensively. In this notion, a need for utilizing a 3D multi-user virtual world in different instructional format has been raised for two reasons: firstly, either due to the use of a visual (game-like) and persistent environment owing to understand basic programming structures better, or secondly due to the use of various a-/synchronous communication and built-in tools which are
freely available.

Various research findings have shown that 3D multi-user virtual worlds have the potential to enhance a learning process for programming, with the following potential benefits (Pellas, 2014; Rico et al., 2013): (a) they can encourage reflective learning of all group members, (b) they can improve students’ cognitive skills (analysis, evaluation and creation) or higher-order thinking skills in synchronous blended or online instructional formats and (c) they can promote the change from superficial to deeper learning of the new knowledge and facilitate the acquisition of socio-cognitive skills for all students, according to their previous experiences through collaborative practice-based tasks. The interactivity and social forms of modeling visually-rich artifacts in 3D multi-user virtual worlds can furnish the instructional design of collaborative and interactive learning activities, considering the appropriate pedagogical underpinnings.

One of the most well-established learning theories that has been extensively used in 3D multi-user virtual worlds is Papert’s theory of Constructionism (Girvan et al., 2013; Pellas et al., 2013). Constructionism can provide the meaningful correspondence to the view that individuals have, in order to think how to produce something as a proof of learning. However, a lack of description on how a well-defined learning process can affect students’ engagement and participation using Constructionism as a theoretical foundation of an instructional design framework has not been implemented for introductory programming courses.

Table 1 briefly summarizes how learning conceptions influence both the instruction and game-like characteristics of an educational virtual grid using Second Life and Scratch4SL. The analysis of tasks was inspired by previous studies (Girvan, 2014; Kafai, 2008; Kommers, 2003; Li et al., 2013), which offered crucial points of
view about Constructionism as a theoretical foundation for well-designed instructional contexts.

Table 1: How the view of knowledge based on Constructionism influences the view of instruction and 3D game-like characteristics

<table>
<thead>
<tr>
<th>Gaining knowledge based on Constructionism</th>
<th>Instructional settings</th>
<th>Game-like characteristics in 3D multi-user virtual environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Transmission of the learning material</td>
<td>Design artifacts as products of knowledge</td>
<td>Learning by doing something using digital tools and media sources</td>
</tr>
<tr>
<td>2. Cognitive statement of a person’s schemas and provision of his/her socio-cognitive skill usage</td>
<td>Setting main learning tasks that aimed at changing an individual’s cognitive schemas</td>
<td>Having winning states which can give to each student an ego gratification</td>
</tr>
<tr>
<td>3. Interaction of each student with the learning environment</td>
<td>Utilize tools or artifacts to create meaningful objects based on users’ knowledge</td>
<td>Interactive activities in game-like settings rely heavily on primary digital-oriented sources of data and manipulative 3D visual artifacts</td>
</tr>
<tr>
<td>4. Using personal meanings based on students’ interactions</td>
<td>Follow specific rules and problem-based strategies to enhance students flexibility to create their own constructs</td>
<td>Tasks in game-like environments should have conflict/competition/challenge/opposition in problem-based tasks that completed after team members’ feedback.</td>
</tr>
<tr>
<td>5. Enculturation of group members’ ways to present what they understand</td>
<td>Collaboration and participation in team-based activities in contemplation of using computational thinking skills to present their final product</td>
<td>Assessment of student learning is interwoven with the instructional process and it occurs through teacher’s observations based on students’ tasks</td>
</tr>
</tbody>
</table>

In this attempt, the 3D multi-user virtual world of Second Life can support an instructional framework underpinned by Constructionism, and collaborative activities should foster the development of students’ higher order thinking and computational skills (Girvan et al., 2013; Pellas, 2014).

3D multi-user game-like environment design

A game-based environment has various playable elements, which can engage students in challenging learning activities. Inevitably, “gaming” is not only socially fascinating and challenging, but also relaxing for users (Hoffman & Nadelson, 2010). Moreno-Ger et al. (2008) have asserted that educational game design needs to involve the requirements of “integration with online education,” “adaptation” and “assessment.” Thus, challenge and social interaction among users are necessary to an educational game.
While some researchers (Bai et al., 2012; Connolly et al., 2012) have revealed the benefits of game-based learning in digital-oriented environments, others have asserted an opposite view. The negative results are normally associated with learning activities that might be hindered, due to the lack of “transferring motivational engagement” from gaming to educational conditions (Hoffman & Nadelson, 2010). Similarly, digital games are resources, which are difficult to be utilized in education for the sake of understanding differences in the educational and gaming contexts (Sancho et al., 2009). As a result, the effectiveness of a game-like environment remains a challenging issue and it is certainly essential to realize how to take advantage of it for introductory programming courses.

Secondary education curricula for Computer Science courses at a global rate have recently posed as main learning objectives the importance of developing or using alternative platforms for computational thinking acquisition. To achieve this, a first step is the creation of geometrical shapes in Logo-like programming environments that can be proposed on the part of high school students learning how to think logically, then to configure something practically, and thereafter to present their solutions to a problem, by using sequential or repetitive programming structures (for example to create circles, arcs or normal shapes).

In the present study, an activity in which the main learning objective was the creation of 24 letters of the Greek alphabet in a 3D multi-user game-like environment was chosen. This programming process can provide some advantages, which are the following: a) 24 Greek letters were distributed to all students in the class; b) letters of the Greek alphabet that consisted of a mixture of shapes (circle, straight line etc.) were taught and have already been known by all students; c) there were different levels of complexity that required a combination of logical thinking skills on the part
of students that should be connected with the appropriate geometric shapes (for example the Greek letters such as "Τ" or "Γ" need two lines to be connected and thence students needed not only to think how they should vertically or horizontally illustrate them, but also to define the appropriate angle to start writing with the stylus in Second Life); and d) it was necessary to use the repeat structure for curved design and to display each letter in several learning contexts.

In an attempt to handle the same structures (artifacts), students should consider different solutions for the same problems based on their computational thinking skills using the Logo-like programming language of Scratch4SL. For this reason, it was decided that learning materials should be designed in 3D geometric complex shapes and finally synthesized in concurrent programming constructs. Based on the learning objectives of programming lessons titled "Programming with innovative learning platforms" the activity was "Designing, prototyping and writing the twenty-four (24) capital letters of the Greek alphabet" provided to all participants. Students needed to learn how to program and successfully propose solutions by creating Greek words in collaborative problem solving activities. A 2-hour preliminary lesson was dedicated to the description of the entire learning process, the disclosure of all students’ accounts and their guidance to install the main client viewer of Second Life and Scratch4SL folder on their computers.

The levels of difficulty varied according to a “fading scaffolding” process carried out for the construction of letters (each letter had its own particular inclined lines, circles or parallel lines). It was more appropriate to separate them into two categories: (i) simple (Γ, Δ, E, H, I, Λ, Ξ, Ο, Π, Τ, Υ, Χ) and (ii) complex (Α, Β, Ζ, Θ, Κ, Μ, Ν, Ρ, Σ, Φ, Ψ, Ω). Each student should create a letter from each category. Between letters of the same category, there was a degree of difficulty that should be
executed. This process was scheduled to have an added complexity. It was assumed that the visually-rich and concurrent object-oriented learning tasks could lead to the reinforcement of students’ programming thinking skills.

The first three class meetings were devoted to familiarizing students with Scratch4SL, in order to bring out their higher order thinking skills for communication and creativity, by drawing basic geometric shapes and shifting with the structures that a stylus of Scratch4SL can provide. Its conjunction with Second Life supports visually-rich (high-ceiling) activities, and by using Scratch4SL, students can obtain a wide range of interactive applications to enhance their technological literacy and produce a more interactive content (wide wall). Unlike Scratch (the 2D visual programming environment), Scratch4SL does not generate Java outputs, but it translates what someone drags and drops from the Scratch block-based pallet into Linden Scripting Language (LSL) that is the core for programming primitives or artifacts in Second Life. This procedure can help them to organize/coordinate their teams and enhance the sense of co-presence to be provided with the successful achievement of common objectives. Learning how to structure recurrence of letters with curves (e.g. Y and I) also became necessary. The next two courses were devoted to the construction of the remaining 23 letters. Below, Figure 1 illustrates the uppercase letter “Γ” using an equilateral triangle.
In-world (Second Life) creation of letters which did not contain curves were used for the sequence of programming structures with the following commands: a) “when I receive A” (to start the sub-program creation of A), b) “pen up/ pen down” (to the stylus that moves in order to write or not), c) “move x meters” (to move the stylus at distance x) and d) “turn right/left x degrees” (to turn the stylus at x degrees), while the creation of rest repeat structure “repeat x times” was also needed. Some indicative examples are as follows: i) for the letter "E" students should bring lines in parallel and perpendicular turn everything at 90° degrees, ii) for the letter "A" students should turn with acute and obtuse angles of the stylus, and iii) for the letter "O" students should use the repetition programming structure and twist a small acute angle many times to be turned 360° degrees. It is worth noting that the small number of orders did not increase the complexity of learning. The computational thinking skills that were used for the creation of Greek words are related with the logical sequence of thinking, understanding, debugging and capturing of the proposed solutions. Thus, students focused on the use of their personal socio-cognitive thinking skills to collaborate together, exchange ideas and present realistic solutions to write an entire
Greek word. However, students found it very difficult to handle problematic situations well enough to provide a logical solution in the first three courses.

A collection of awards was included for this 3D multi-user game-like environment. Beyond the basic winner prize, a set of other awards that determined the final winners in the case of a tie was presented to all participants in Second Life. The awards were announced as follows: (i) “write the code correctly”: it was given to the group that has correctly written all programming codes, (ii) “deconstruction of the problem in right settings”: it was given to the group that had discovered the largest number of questions to solve the problem, even if finally they had not answered them correctly and (iii) “best short time spent to assemble and program interactive artifacts”: it was given to the group that assembled and programmed correctly almost all the artifacts without destroying the letters of other members. This was awarded from two letters, one simple (e.g. “Π”) and another complex (e.g. “Φ”) for each student to create an imaginary chain, where each one had the same letter to the next and the other same as the previous one. Students preferred the communication and collaboration with others via Second Life, rather than the conventional instructional formats (teacher’s lectures). The last lesson was essential for understanding students’ interactions in collaborative activities. It was dedicated to the integration of the individual structures in a total of all the letters in order to write a Greek word («ΓΕΙΑ», «ΕΛΑ», «ΦΩΤΙΑ»).

Figure 2 depicts the alignment between Scratch4SL and Second Life for the creation of a 3D game-like environment. This figure illustrates students’ management responsibilities to coordinate their teams during the learning process, following the pedagogical principles of Constructionism. One suggestive example of this 3D game-like environment is the creation and presentation of the Greek word, e.g. “ΓΕΙΑ”.
Two students should create one letter each time in Second Life. Collaboration in problem solving tasks in this 3D multi-user game-like environment promoted by students’ reflections through interactive activities, by using verbal (VoIP) or non-verbal (gestures or IM) communication tools. Second Life enables synchronous collaboration among avatars in a common virtual environment to edit the same primitive and share the same code while programming it with Scratch4SL. All users (students and teacher) had the opportunity to access and leave in-world objects or private messages to the other members (group of 2 avatars for each team). If someone provided a solution on how to program or collide a primitive successfully, s/he
should be logged in all his/her messages to see all the objects left in Second Life by others, when the proper permissions were set.

**Research methodology**

**Overview**

This case study aimed at measuring the promotion of students’ engagement and participation in a 3D multi-user game-like environment using a mixed-method research. This type of research method is defined as a mix of both quantitative and qualitative methods in a single study, in which data should be collected concurrently (Creswell et al., 2003). It requires both a legitimate and standalone design. The reasons of this combination for the present study are presented below (Jick, 1979):

a) to bring the strengths of research forms in order to validate results of a case study;

b) to enrich the study results in ways that one form of data does not allow;

and lastly

c) to design “mixed” data that can be combined with different, but complementary data, which may be neglected or lacked by a single one method.

The present research followed recommendations of Abbas (2010) who has highlighted that small-scale research projects in 3D multi-user virtual worlds can offer an added value to the educational perspectives. This study was designed to investigate the student engagement and collaboration in Second Life and Scratch4SL for the implementation of a collaborative problem-based learning approach delivered in blended instructional format (class supported in a computer laboratory and supplementary online). This 3D multi-user game-like environment supported students so as to understand during their participation adequately:
i. how to think computationally, before the basic programming structures are executed concurrently in Second Life via Scratch4SL;

ii. how to use Scratch4SL and Second Life capabilities for the implementation of problem-based activities in programming courses; and

iii. how to collaborate with other peers in problem-solving learning tasks via a 3D multi-user game-like environment.

Participants

Fifty-six (56) students of a Greek public high school participated in the present study voluntarily, after the signed permission of their parents in the second trimester of 2014, from three different classes. The mean age of all students was between 14-15 years old for males and females (SD=1.14). While 57% of males and less than 13% of females had previous experience with massive multi-user online virtual worlds (MMORPGs), they had no previous experience with Second Life. Nonetheless, all had already utilized 2D Scratch environment for animation-based programming concepts. Forty-five (45) are already members of the Scratch online community. All participants had different socio-cognitive backgrounds in programming courses and grades in Computer Science courses. Moreover, they had no extensive practical experience in the execution of programming structures through other innovative learning platforms, but only with Scratch. Indeed, they had only two weeks before becoming involved in both environments for understanding the initial conditions of their exposure via Scratch4SL and Second Life. Some important demographic characteristics that should be noted are as follows: a) 28 students had from low to mediate grades (13-16/20) and only 28 of them had from mediate to high grades (17-20/20), b) 24 students had daily visited Web-based platforms or transactions, 20 of them only monthly and 2 had not
visited any Web source. Kodu and Storytelling Alice were two visual programming environments that students had the opportunity to learn how to code. However, they mentioned several difficulties (36/56). Table 2 below presents students’ demographic data.

Table 2: Students’ demographic characteristics

<table>
<thead>
<tr>
<th>Measures</th>
<th>Frequency</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>33</td>
<td>64</td>
</tr>
<tr>
<td>Female</td>
<td>23</td>
<td>36</td>
</tr>
<tr>
<td>Age (years old)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-13</td>
<td>38</td>
<td>50</td>
</tr>
<tr>
<td>14-15</td>
<td>18</td>
<td>50</td>
</tr>
<tr>
<td>Member of Scratch online community</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>45</td>
<td>70</td>
</tr>
<tr>
<td>No</td>
<td>11</td>
<td>30</td>
</tr>
<tr>
<td>Total</td>
<td>56</td>
<td>100</td>
</tr>
</tbody>
</table>

**Instruments and data collection tools**

Some studies have provided several validated questionnaires, such as: a) Fredricks et al. (2004) who described factors related to student engagement that are measurable at conventional settings (school, classroom, and individual levels) and b) Appleton et al. (2006) who focused on different aspects of student engagement, which are related to academic (time on task), behavioral (classroom participation), cognitive (strategizing), and psychological (sense of belonging) settings.

Wiebe et al. (2014) have stressed that is an urgent need to re-conceptualize a new instrument for understanding the term of user engagement in digital game-based environments. Indeed, more emphasis should be given to a new instrument that encompasses student behavior in both physical and virtual environments. The psychological states also need to be varied in two measurement approaches: a) on the behavioral side, to characterize overt behavior in education (Admiraal et al., 2011) and
b) in the context of computer-based activities to direct observation of activity 
(Lehmann et al., 2012). Besides the validity of the above questionnaire, there is a 
growing academic body of literature (Grafsgaard et al., 2014; O'Brien& Cairns, 2015) 
that were finally used to measure users’ engagement in game-based instructional 
settings.

For the aforementioned reasons, the User Engagement Scale (Wiebe et al. 
2014) appeared to be the most appropriate tool for the collection of this quantitative 
data. It consisted of 31 close-ended Likert scale questions (from 1=strongly disagree to 
5=strongly agree) with the following six subscales: focused attention (FA), felt 
involvement (FI), novelty (NO), endurability (EN), aesthetics (AE) and perceived 
usability (PU). The entire questionnaire was needed to be adapted for the purposes of 
this study. All questions which had the word “website” were changed to the words 
“3D multi-user game-like environment”. For example, in the EN subscale when it was 
asked ‘‘Playing the game on this website was worthwhile,’’ it would be adapted 
according to the needs of the present study as ‘‘Playing on this 3D multi-user game-
like environment was worthwhile’’. The reliability results were in slightly acceptable 
level (Cronbach’s alpha, $a=.795$).

The data were gathered in order to be encoded into the multi-dimensional 
directions of this case study, according to students’ opinions. The statistical program 
SPSS (ver. 22) was used for setting up the quantitative data. Also, Nvivo (ver. 10) was 
utilized as a means to aggregate students’ answers from the interview (qualitative 
analysis). The students’ interviews were meticulously conducted with the extent to 
strengthen the reliability and validity of this study’s findings. The data were obtained 
through this process to clarify the research questions collected from the interview in an
open-ended questionnaire to encode students’ experiences based on their interactions as avatars in collaborative problem solving tasks.

**Treatment**

The contribution of this treatment comes from the anticipated outcomes that were expected from the conjunction of Second Life and Scratch4SL. Some of the most important factors that influenced the creation of a game-like environment in Second Life are the following: a) a low-cost 3D persistent workflow (i.e. a workflow that still exists even when users log out from it and the changes that they have made are permanent), and this “workflow” can assist the implementation of different instructional formats (blended/online); b) the technological infrastructure of Second Life can provide real-time feedback on users’ interactions to create and syntax multiple codes in visually-rich problem-based learning settings; c) the a-/synchronous communication tools and realistic aesthetics of artifacts or objects in a 3D virtual grid that can aid users to be engaged in realistic situations; and lastly d) flexibility and adaptability are unique characteristics of a 3D virtual grid for users to create a learning platform, according to their needs or demands (sense of adaptability).

This study had a twofold purpose: firstly, to familiarize all students in using concurrent programming languages through the creation of artifacts in a 3D multi-user game-like environment; and secondly, to present an easier way of assisting students to be enrolled in programming courses collaboratively, using programming structures of 2D Scratch4SL environment and thereafter by combining the code to create through the “stylus” (pen) in favor of creating 3D artifacts (Greek letters) in Second Life. The virtual world of Second Life is one of the most appropriate 3D multi-user virtual worlds for educational needs. This process enable students to
develop and produce alternative solutions in problem-based learning conditions and strengthen further their technological literacy skills.

The study took 6 weeks and 18 hours of exercise in blended settings (8 hours in the schools' computer laboratories and another 10 in supplementary online courses at the discretion of the Computer Science teacher). The teaching of concurrent programming constructs took place in the computer lab (2 teaching-hours), but students also attended online supplementary sessions (maximum of 2 teaching-hours in each meeting). The teacher attended all courses, and he accessed all learning materials during the online and face-to-face instructional settings.

Apart from the difficulty of creating 24 Greek letters, additionally important was the creation of words which could be achieved in parallel (concurrent programming), using sequence or repetition programming structures provided in Scratch4SL. To achieve their learning objectives, students had to collaborate and set some common rules that letters had the same size for both their height and the distance from each other. Then, attention was given to the design of the letter so that the stylus was placed in the same height and in the same direction (i.e. always upward). Only authorized users participated in the educational process. It was also necessary to address security issues that allowed the access only to all those users who participated in these courses in order to avoid misbehaviors from other unknown avatars. Everyone who was enrolled after the end of the first course was also geographically distributed with her/his laptops from others. Meanwhile, each team consisted of two students needed to be separated side-by-side (in pairs) at computers in the laboratory for the initial courses with the intention of learning collaboratively.

The main principles of a teaching and learning process based on a Constructionism framework in Table 3 are analyzed.
Table 3: The alignment of learning activities based on Constructionism

<table>
<thead>
<tr>
<th>Guidance for students’ participation in the learning process</th>
<th>An instructional design framework based on Constructionism to support main principles of students’ computational thinking skills</th>
<th>Guidelines for describing a 3D game-like programming environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st phase of the teaching process:</td>
<td>(A) Degradation and comprehension of the main problem (logical reasoning of a process to solve and debug a problem)</td>
<td>a. Motivating and providing fading scaffolding activities under the teacher’s guidance.</td>
</tr>
<tr>
<td>1a. The teacher’s presentation for the problem-based situation in which students are engaged.</td>
<td>i) Constructing visual artifacts as key elements of knowledge acquisition.</td>
<td>b. Understanding the spatial horizons in which students try to adopt with.</td>
</tr>
<tr>
<td>1b. Main goal settings of the learning process and the problematic view of programming thinking skills are presented.</td>
<td>ii) Incentives to engage students in learning activities to recognize a new way of thinking and then constructing concurrent programming structures.</td>
<td>c. Organizing and coordinating well-established programming structures based on students’ needs and their previous experience.</td>
</tr>
<tr>
<td>1c. Analysis of students’ needs and demands should be elucidated.</td>
<td>iii) Proposals of students’ efforts and aspects for Second Life and Scratch4SL.</td>
<td></td>
</tr>
<tr>
<td>1d. The conjunction of design content analysis according to students’ needs is proposed.</td>
<td>(B) Analysis of a problem-based learning situation (development of a proposed solution from the initial concept to the project completion)</td>
<td></td>
</tr>
<tr>
<td>2nd phase of the teaching process:</td>
<td>i) Discerning the combination of low requirements for the management and use of Scratch4SL’s stylus (pen).</td>
<td>a. Triggering students so that prepare a participatory – collaborative (in pairs) activity with a gradual difficulty in order to co-construct complex artifacts.</td>
</tr>
<tr>
<td>2a. Basic challenges should be created via 3D visual artifacts</td>
<td>ii) Sharing users’ pallet with multiple programming structures.</td>
<td>b. Implementation phase.</td>
</tr>
<tr>
<td>2b. Initial evaluation should be provided according to students’ needs.</td>
<td>iii) Providing functional requirements related to the team coordination level.</td>
<td>c. Constructing an innovative knowledge field for learning concurrent programming structures.</td>
</tr>
<tr>
<td>2c. Designing the delivery of the educational experience.</td>
<td>(C) Synthesis of programming skills (persistent motivation for the synthesis of concurrent programming based on sequential programming constructs commands which can be executed simultaneously)</td>
<td></td>
</tr>
<tr>
<td>3rd phase of the teaching process:</td>
<td>i) Constructing the main simulation collaboratively.</td>
<td>a. Getting functional requirements related to teacher’s feedback and support.</td>
</tr>
<tr>
<td>3a. The implementation of interactive perspectives via constructing learning materials in a collaborative climate is announced.</td>
<td>ii) Launching and engaging with artifacts a large audience related to the appropriate experimental methods.</td>
<td>b. Final assessment of artifacts and presentation of the visually-rich prototypes.</td>
</tr>
<tr>
<td>3b. The coordination of pilot projects that implemented with inquiry-based activities should be executed.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The potential creation and transformation of the programming process was created by each student using only “copy-paste” processes through Scratch4SL scripts combined with Linden Scripting Language (LSL) script in each 3D visual object of Second Life. This programming language allows the structure of “objects-to-think-
with” so that program and watch students the process of computational programming constructs concurrently.

**Data collection**

Before students try to understand how to program and create a Greek word using programming artifacts (“objects-to-think-with”), students should first recognize and then propose a solution. Computational thinking skills that students should afford before and after the execution of concurrent programming concept are the following: (a) the degradation and comprehension of the main problem, (b) the analysis of the problem-based learning situation, and (c) the synthesis of programming skills. As for the latter skills, concurrent programming based on sequential commands in parallel via a 3D game-like environment was necessary for the creation of Greek letters and then for the creation of an entire word. Students need to have systematic thinking, regarding the guidelines that should be linked with programming structures in order to execute all of them at the same time. The construction of the main user-oriented instrumentation of this study aimed at measuring the first-impact of their experiences. Specifically, another qualitative analysis for significant themes, like students’ motivation or the empowerment of computational thinking skills in collaborative problem-based tasks, before or after their intervention is also amplified. In Table 4, students’ perceptions and opinions are presented.

*Table 4: Criteria and indicators of the quantitative and qualitative analysis related to the contribution of Scratch4SL and Second Life on students’ engagement*

<table>
<thead>
<tr>
<th>Criteria of evaluation</th>
<th>Indicators</th>
<th>Implementation of activities in Second Life via Scratch4SL</th>
<th>Indicative teaching process – The analysis of students' socio-cognitive and programming skills</th>
<th>The alignment of user engagement scale with the initial evaluation criteria</th>
<th>Data collection</th>
<th>Data analysis process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degradation and Deliberation to Activity 1 (4)</td>
<td>i. Navigate and discover the 3D</td>
<td>1. Aesthetics</td>
<td>i. Reading</td>
<td>Qualitative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(B) Analysis of the problem-based learning situation (the development of a proposed solution from the initial concept to the project completion)</td>
<td>Deliberation to clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(AE) Game-like environment in order to understand better the communication and built-in tools that should be used via Scratch4SL. ii. Use positioning commands for the creation of an interactive motion.</td>
<td>Deliberation to clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(B) Analysis of the problem-based learning situation (the development of a proposed solution from the initial concept to the project completion)</td>
<td>Deliberation to clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(AE) Game-like environment in order to understand better the communication and built-in tools that should be used via Scratch4SL. ii. Use positioning commands for the creation of an interactive motion.</td>
<td>Deliberation to clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(AE) Game-like environment in order to understand better the communication and built-in tools that should be used via Scratch4SL. ii. Use positioning commands for the creation of an interactive motion.</td>
<td>Deliberation to clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C) Synthesi</td>
<td>Deliberation to Activity 4 (2 teaching hours of exercise in blended instructional formats): Providing to the main teacher what information can be gained from Second Life.</td>
<td>Activity 3 (6 teaching hours of exercise in blended instructional formats): After understanding how to use both Scratch4SL and Second Life, all students should try to understand the main problem that titled: “Writing Greek words with other using Scratch4SL”. Each one must try to write correct and program a Greek letter. The scaffolding process divided in way to program a Greek word, firstly with an easy to create letter via Scratch4SL, such as “Λ” and secondly to program a more complex one, e.g. “Α” or “Ε” to complete a Greek word like “ΕΛΑ”.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
clarify the team’s common conviction: i) by creating artifacts (Greek letters), ii) by communicating verbal via VoIP or non-verbal via gestures or IM, chat text in order to exchange ideas.

teaching hours of exercise in blended instructional formats: according to each student experience, as teams separately all should provide their experience and compare Scratch4SL with other programming environments that they have used.

previous environments with Scratch4SL with the respect to their functional characteristics

ii. Select the appropriate tools and transactions to solve a problem-based situation.

Endurability (EN)

2. Perceived usability (PU)

and organizing brainstorming via text chat and VoIP

ii. Videotaping the session of the blended and supplementary online course delivery methods

| Qualitative analysis: (a) Codifying chat message and (b) Receiving observations notes of the main researcher |
|____________|____________|____________|
| Mean (M), and Standard deviation (SD) |

The data were collected from the qualitative analysis of the students’ chat text messages, in which they exchanged opinions or proposed solutions to the problem-based learning situation. The data referred to the comprehension of common fractional concepts, which were also collected though a two-stage assessment (pre-test and post-test interview sessions), before and after their learning intervention in a 3D multi-user game-like environment. In addition, there was systematic observation by the researcher during all video-recorded interviews.

Results and Discussion

According to students’ answers for the user engagement scale (quantitative data collection), it was proved that their attention, persistent involvement and interest to utilize Scratch4SL and Second Life in a 3D game-like environment were at high levels. Students’ answers in main questionnaire adapted from Wiebe et al. (2014) have given some important aspects for the learning process. These are presented below:
First, for the FA scale, students seemed to be engaged in the entire process ($M=4.01$, $SD=2.41$), without losing their interest ($M=2.01$, $SD=1.99$) during the execution of their actions ($M=4.21$, $SD=3.14$). The 3D game-like environment allowed not only their incentives to think and solve a problem ($M=4.21$, $SD=1.41$) collaboratively, but also with Computer Science teacher’s feedback, they have not lost their track on it ($M=1.77$, $SD=2.11$).

Second, for the FI scale according to students’ answers, the learning experience was generally favorable ($M=4.81$, $SD=3.01$) and fun ($M=4.55$, $SD=2.77$).

Third, for the PU scale, the utilization of the environment was not frustrating and many students had the chance to communicate verbally or non-verbally with others ($M=4.21$, $SD=2.92$). This process allowed them to exchange ideas or opinions on how to work on a problem in pursuance of solving it collaboratively, without being discouraged or mentally distracted ($M=4.11$, $SD=2.88$).

Fourth, for the EN scale according to students’ answers the schedule of utilizing Scratch4SL and Second Life was very attractive for the implementation of students’ ideas, as it was associated with their computational thinking skills usage ($M=1.87$, $SD=1.77$) in game-like settings that were completed quite well ($M=3.78$, $SD=1.74$). Also, the rewarding game experience helped them to study in better learning settings ($M=3.68$, $SD=1.78$).

Fifth, for the AE scale according to students’ answers the 3D multi-user game-like environment was aesthetically appealing ($M=4.21$, $SD=2.91$) and Second Life helped them optically and acoustically to get prompt feedback ($M=3.45$, $SD=1.43$). Graphics and images of Scratch4SL assisted students to get involved more easily with programming structures, in which they first of all needed to think algorithmically how to execute something and then program “objects-to-think-with” in Second Life. They
also should predict the behavior of the stylus from Scratch4SL block-based pallet and create Greek words in Second Life ($M=4.51, SD=2.71$).

Finally, for the NO scale of students have shown their curiosity ($M=3.91, SD=1.94$) and they found the entire learning experience really interesting ($M=4.39, SD=3.89$). Hence, students’ engagement in introductory programming courses was increased.

In Table 4, students’ answers (qualitative research method) based on a brief description is presented through an open-ended interview questionnaire. After finishing their activities students were asked to complete a web-based questionnaire, in which they freely expressed their thoughts. Table 5 encodes the following issues.

<table>
<thead>
<tr>
<th>Basic criteria</th>
<th>Significant themes</th>
<th>Students’ opinions about the learning process before Scratch4SL and Second Life usage</th>
<th>Students’ opinions about the learning process after Scratch4SL and Second Life usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Degradation and comprehension of the main problem (logical reasoning of a process to solve a problem)</td>
<td>1. Motivating students attention to get involved in a 3D multi-user game like environment</td>
<td>&quot;Programming courses are quite interesting but I cannot carry them out when they associated with programming constructs easily, because the instructor cannot be closer to me or my team. Thus programming environments that have been utilized inside the class seemed to be boring.&quot;</td>
<td>&quot;The course was conducted in collaborative contexts and it was provoked my interest and my personal motivation to acquire the programming knowledge.&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(b) &quot;I believe that Programming is a good lesson, because it presents a new way of thinking programming concepts which are really helpful and insightful. The utilization of different is the most interesting part in programming courses.&quot;</td>
<td>(b) &quot;Finally, it was a pleasant lesson that enabled me to product my script by co-constructing programming structures in collaborative contexts.&quot;</td>
</tr>
<tr>
<td></td>
<td>2. Empowering students programming thinking skills in problem-solving situations</td>
<td>&quot;I wish to have more freedom to move or navigate in a programming environment and help others that cannot alone configure anything. The instructor cannot always help us.&quot;</td>
<td>&quot;I believe that cooperation in the 3D virtual environments and in a common place with my teammates was the key parameter to our success and thus we performed algorithmic structures. The direct feedback from the&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(b) &quot;I do not like to&quot;</td>
<td></td>
</tr>
</tbody>
</table>
| 3. Trying to use computational thinking skills to syntax programming structures | (a) "Usually everyone was getting bored in class and get tired because it has interesting activities. The programming course is generally so difficult for me!".  
(b) "Sometimes I do not understand a lot of things about Computer Science courses. Especially the lesson programming constructs seemed hard and software Micro-worlds did not give me the enthusiasm to conquer the programming thinking skills".  
(c) "In our conversation with other peer, we have discussed and addressed basic problems to understand programming constructs." |
|---|---|
| (B) Analysis of a problem-based learning situation (the development of a proposed solution from the initial concept to the project completion) | (a) "Something that really helped me to understand the lessons were that the instructor was followed a fading scaffolding process, while he/she was always with us in the computer lab and let our personal initiatives in order to write the code.".  
(b) "It was an interesting process, but without the instructor's guidance it would be more difficult.".  
(c) "Gradually I have learned some basic programming structures (if... Else if... then repeat process) helped me to analyze and predict via algorithmic thinking artifacts movements and graphic pen styles." |
| 4. Improving peer relations and collaborations | (a) "Usually I don’t study together with my peers, as it was not given to us various opportunities to share our personal experiences with others.".  
(b) "It’s so difficult to work with others because sometimes I think I have the right to talk only me and not the others.".  
(a) "Despite the fact that I am not good in using new technologies and therefore to manipulate algorithmic structures, I found that the Scratch4SL seemed more accessible to my needs and additionally it seemed that my friends could easily understand programming structures too.".  
(b) "I am very pleased that I wish my team took part in this project and learned more easily programming structures based on the utilities that scratch4SL and Second Life can replicate." |
| (C) Synthesis of programming skills (the use of logical thinking skills) | (a) "Changing one script numerous times by using 2D Scratch was very " |
| 5. Implementation of students’ ideas via Scratch4SL and Second Life | (a) "I found both visual and acoustical feedback really helpful" |
The above interview analysis indicated that the use of Second Life positively influenced students’ engagement in programming. However, this cannot be guaranteed for all cases. Among these, there were also positive signs that should include some further fundamental points of view: (a) the increasing motivation, the degree of engagement and collaboration among students, and (b) the increasing level of students’ achievements.

According to the interview’s observations, the benefits observed are really important. Some of them are as follows:

- Students finally seemed to start thinking critically and logically in order to execute their programming concepts that were divided into a series of complex tasks. They started to provide their ideas visually, by utilizing synchronous or asynchronous communication tools in collaborative problem-solving situations.
The syntax of concurrent programming concepts via Scratch4SL based on a set of colored blocks has enhanced students’ programming experiences in Second Life. The 2D environment reduced the “steep learning curve” that novice programmers needed to overcome during their first entrance in Second Life.

The construction of other “objects-to-think-with” in Second Life via Scratch allowed students to be engaged and event-driven in programming executions. The financial cost was minimal, beyond the rent of a secured virtual grid for this study, and students’ engagement was at high levels.

Some interesting findings revealed from this study are the below:

(a) Enhancement of students’ computational thinking skills in problem-based activities: Second Life and Scratch4SL contributed to the entire instructional approach of programming. The study findings revealed positive acceptance for the construction of “objects-to-think-with” through Scratch4SL or a-/synchronous communication forms through Second Life, regardless of students’ previous socio-cognitive background. These two environments have been provided as worthwhile innovations and have reinforced students’ high-order and computational thinking skills. It is proven that students seek to understand sequential processes and to think “computationally” in complex programming concepts via a 3D multi-user game-like environment.

(b) Sociability and cooperativeness: All students were satisfied about their communication with their peers and with the Computer Science teacher in collaborative problem-based processes. This seemed to be pretty remarkable and insightful for their engagement and participation.
(c) The knowledge on how to use computational thinking skills: Students’
engagement, collaboration and active participation have affected more positively the
understanding of basic programming concepts.

A general observation is that the practice of concurrent programming
languages which are visually observable can also be implemented in Second Life via
Scratch4SL. This was the main reason of using these environments for students’
persistent engagement in programming courses. Furthermore, it seemed to encourage
high school students to continue their study in blended instructional format in a 3D
user-friendly/ game-like environment, in which each user “learned by playing.” In
these circumstances, the use of Scratch4SL and Second Life outside the aid of
technological literacy can be also useful for students’ achievements and empower their
socio-cognitive skills, such as creativity, analytical-synthetic thinking, and their
efficient collaboration.

Two objectives are generally related as significant and extent in the
contribution of the entire study. The first is related to students’ cognitive background
for understanding basic concepts, based on problematic views that appeared in
programming courses, and which should be taken under consideration from Computer
Science teachers. This distinction is focused on the central concern of how students
can recognize the added value of basic programming concepts through the
implementation of a proposed solution that has been taken collaboratively. Secondly,
the training session in well-organized instructional settings helped students to solve
problem-based learning situations.

It is also significant to mention that this study comes into contrast with Olive’s
(1991) study which considered that high school students cannot benefit from Logo
programming in learning geometry. Scratch-based game programming tools, in
association with the Computer Science teacher’s feedback, have acted as very important for students to test and sometimes extend the conceptual understanding of concurrent programming structures via Second Life. In fact, students as designers can substantiate the meaning of abstract concepts and can test their reasoning with different 3D visually-rich objects for a series of programming tasks.

On the other side, Kelleher and Pausch (2005) have pointed out that by understanding programming structures and by utilizing an easy-to-use 3D visual environment, students not only can easily try to acquire a logical thinking to solve a problem, but also try to understand easily their management responsibilities to handle learning materials and express their solutions with other peers. Scratch4SL has low-threshold hardware requirements and as a result assisted high school students to exemplify a problem more easily and to propose a solution in collaborative instructional settings. Scratch combined with Second Life acquires high-impact (wide walls) capabilities, and a wide range of applications can empower students’ technological literacy, by starting to prototype several visually-rich components and by producing an interactive learning content.

By studying students’ activities from Scratch4SL pallet to Second Life artifacts, it can be generally developed a definition of computational thinking that involves three key dimensions: (a) computational concepts, (b) computational practice-based tasks, and (c) computational outcomes. The main questionnaire and interviews have been instrumental in helping computer Science educators and teachers understand the longitudinal development of novice programmers, with participation and project portfolios spanning weeks to several months. Workshops weekly or monthly have been an important context for understanding the practices of the creator-in-action.
The added value of present study for educational practice provided the following benefits:

- The exploitation of a 3D (low-cost) multi-user game-like environment for the implementation of programming courses created in Second Life and combined with Scratch4SL, in the direction of eliminating the “steep learning curve” for introductory programming courses.

- The development of an instructional framework based on Constructionism as a theoretical foundation for the development and implementation of programming courses in game-like settings.

- A constructivist-oriented instructional framework with regard to amplifying students’ interactions via a 3D interactive game-like environment for the implementation of future-driven tasks that should be seriously considered by Computer science teachers.

- Scratch4SL can be effectively used as a “low-floor” programming tool that can be successfully combined with Second Life. Students’ engagement was increased and as a result the construction of “wide walls/high floor” artifacts by using their programming thinking skills succeeded at high levels.

- The instructional framework based on Constructionism reinforced students’ computational and problem-solving thinking skills, especially when they needed to manage “high ceiling/wide walls” learning materials.

This study findings further stated that instructional design frameworks based on Constructionism are prescriptive guidelines that indicated on “how to do” something meaningful in introductory programming courses via a 3D multi-user virtual environment. Moreover, the “constructionism-oriented” instructional framework proposed may also add much to the literature, due to its prescriptive and
complementary nature about the creation of game-like learning environments. The study might also be unique in this sense, since it is separated from a series of descriptive studies (Carbonaro et al., 2010; Harlow & Leak, 2014; Lye & Koh, 2014) that provided knowledge of ‘what is’, but not on “how to do”.

Finally, there is the apparent and urgent need for the introduction of an instructional design framework that will help and guide instructional designers for the efficient use of games and simulations in programming courses for high school students, and more precisely to create game-like learning environments. Additionally, while there exists hardly any study that bears a resemblance to the researchers’ intentions, the findings of this study can possibly contribute, if not illuminate to some extent, the path along with the creation of game-like learning environments using Second Life and Scratch4SL.

Conclusion

Scratch4SL and Second Life in a 3D game-like environment have created a noticeable contribution to introductory programming courses, in which students are engaged at high levels. Although Chang (2014) has found that by integrating only Scratch into K-12 computer courses, students cannot be engaged so easily with object-oriented programming concepts, this study proved that collaborative tasks have been implemented very well based on students’ answers. The study findings were consistent with the findings of with previous works (Hwang et al., 2012; Shadiev et al. 2014), which have identified the positive effect of collaboration for students’ participation to exchange ideas for better learning achievements in well-designed instructional design settings.
Students were positively surprised with supplementary (online) collaborative schedules outside the regular school program. The teacher’s feedback during all courses in Second Life was another significant point of view that should be noted. The students’ answers revealed their positive acceptance on how instruction in a 3D game-like environment assisted them to engage with innovative and interactive learning situations. While some of them had some difficulties during their first introduction, the vast majority found the learning process really interesting compared with the conventional 2D Scratch environment. Also, they did not find the supplementary online instructional format via Second Life boring.

According to the qualitative and quantitative analyses, this study has positively contributed to students’ engagement and participation in programming courses. Some of the most positive points of view that must be underlined are as follows:

(a) A significant increase in students’ engagement to complete tasks in collaborative problem-solving programming tasks, and

(b) Students’ learning outcomes indicated that the appropriate synthesis of programming structures can be completed correctly without syntax errors, regarding the completion of several tasks from the Scratch4SL stylus to Second Life artifacts.

Educational implications of this study that should be underlined are as follows:

(a) the diversity of students’ self-presentations using computational thinking skills and then executing structures, which were optically and acoustically observed in Second Life. This learning process offered each student a 3D embodied representation (virtual alter-ego), in which they felt, like being in a virtual space where they participated with other users,

(b) the freedom of creativity in a 3D game-like environment. All students had their own domain to create attractive conditions for thinking “computationally,” then
to execute and finally to implement visually-rich structures by communicating and collaborating with others,

(c) the combination of Second Life and Scratch4SL encouraged students to co-develop an innovative knowledge domain that may allow their meetings with peers in face-to-face and online instructional settings, outside the specific teaching hours, facilitating them to become "seekers of knowledge,"

(d) the 3D multi-user game-like environment and the feedback from all users (teacher and students), are considered to be equally important for their participation and persistent engagement.

The present study provides various educational benefits in using 3D multi-user virtual worlds for the implementation of collaborative problem-based scenarios in introductory programming courses, contrary to previous studies (Kelleher & Pausch, 2005; Scaffidi et al., 2012; Zuckerman et al. 2009) which have utilized other 2D or 3D visual environments (e.g., AgentSheets, Alice or Scratch). Firstly, is the collaborative climate among users created by using a/synchronous communication tools in a common 3D virtual grid of Second Life and Scratch4SL, which is a well-known palette for them and secondly the Computer Science teacher’s feedback/assistance to all students who participated at the same time, in their effort either to understand their mistakes or to produce ideas visually in Second Life. Finally, the combination of Second Life and Scratch4SL greatly enhanced students’ engagement, socialization and collaboration.

Limitations

Despite the positive attitudes based on students’ opinions, some of the most significant limitations are:
(a) The small size of the sample (n=56) with the same age, but with different socio-cognitive background in programming courses,

(b) The presentation of qualitative and quantitative data through interviews cannot be easily generalized, due to the small sample of students.

**Future research**

A future study could include a statistical analysis of quantitative data from a comparison between an experimental and a control group of students. A comparative study could expose some weaknesses or strong arguments concerning whether Computer Science teachers should include Scratch4SL and Second Life in their courses as candidate learning platforms. Such a study would measure learning effectiveness in both platforms by following Constructionism.

**References**


Symposium on Computer Science Education, SIGCSE ’13 (pp. 463–464).
New York, NY, USA: ACM.


Bogazici University, Istanbul, Turkey.


Kafai, B. & Burke, Q. (2013). The social turn in K-12 programming: moving from computational thinking to computational participation. *In SIGCSE 2013 Proceedings* (pp. 603-608). USA.


