Leveraging Scratch4SL and Second Life to motivate high school students’ participation in introductory programming courses: Findings from a case study

Nikolaos Pellas, University of the Aegean

Available at: https://works.bepress.com/nikolaos-pellas/18/
Leveraging Scratch4SL and Second Life to motivate high school students’ participation in introductory programming courses: Findings from a case study

Abstract

Students in secondary education strive hard enough to understand basic programming concepts. With all that is known regarding the benefits of programming, little is the published evidence showing how high school students can learn basic programming concepts following innovative instructional formats correctly with the respect to gain/enhance their computational thinking skills. This distinction has caused lack of their motivation and interest in Computer Science courses. This case study presents the opinions of twenty-eight (n=28) high school students who participated voluntarily in a 3D game-like environment created in Second Life. This environment was combined with the 2D programming environment of Scratch4SL for the implementation of programming concepts (i.e. sequence and concurrent programming commands) in blended instructional format. An instructional framework based on Papert’s theory of Constructionism to assist students on how to coordinate or manage better the learning material in collaborative practice-based learning activities is also proposed. By conducting a mixed-method research, before and after finishing several learning tasks, students’ participation in focus group (qualitative data) and their motivation based on their experiences (quantitative data) are measured. Findings indicated that is meaningful an instructional design framework based on Constructionism for acquiring or empowering students’ social, cognitive, higher order and computational thinking skills. Educational implications and recommendations for future research are also discussed.

Keywords: Constructionism; Programming; Scratch4SL; Secondary education; Second Life; Virtual prototyping.
Secondary education curricula (high or middle school level) in all around the world have already recognized Computer Science and specifically programming courses as very important for students’ social-cognitive development. Computer programming requires the use of students’ computational thinking and problem solving skills throughout a persistent practice. A major problem with introductory programming courses is primarily the result of the cognitive complexity that many high school students face, while learning with difficulties even the most basic computer programming skills, as Pellas (2014) has analysed. Gaining or enhancing programming skills by thinking “computationally” is challenging, due to its cognitive complexity. Several studies (Ala-Mutka, 2007; Esteves & Mendes, 2004; Gomes & Mendes, 2008) have also noted that learning and understanding basic programming concepts in introductory programming courses is a really difficult task, because students need to acquire or exploit a wide range of skills, such as higher order thinking skills to deconstruct a problem, propose a solution, and a logical synthesis or analysis of ideas based on the respecting rules, beyond knowing only how to syntax a code correctly. Mei et al. (2004) have pointed out students’ low ability to create/develop their own code in problem-based learning situations effectively. Students need to start thinking the details to syntax a code on the one hand, but on the other many of them fail to acknowledge how to use a specific programming language as a presupposition factor to solve a problem.

Referring to the affordances of computational representations for the expression of powerful ideas, Papert (1996) was the first theorist who referred the term “computational thinking”. This term embraces the logical and analytical-critical thinking skills that each human at any time need to gain knowledge using computers. Computational thinking skills are meaningful to high school students in favour of searching solutions to problem-solving situations rising from different learning disciplines (e.g. Mathematics, Physics etc.). A general
reflection that instructors of Secondary education have to consider, is how to motivate
students in a way to learn not only how to subdivide a problem in smaller pieces (abstraction),
but also thinking to solve it with specific rules, purposes and auxiliary data that is used so as
to propose a solution. These key components seemed to be very crucial for someone who
wants to think logically. This process can be generally associated in introductory
programming courses. More specifically, the use of a digital-oriented programming
environment can help students to solve similar real’s life problem-based situations. Depending
on why computational thinking skills are important to be learned and what are the learning
goals that all students need to gain, Computer Science instructors should focus their
investigation not only to an appropriate digital learning environment, but also to the
instructional format that these “skills” can be delivered. Therefore, Computer Science
instructors and instructional technologists, in high school introductory programing courses
need to deal with a twofold problem. Firstly, from a pedagogical-instructional perspective, to
recognize all students the cognitive value of learning how to gain/use computational thinking
skills and how to program correctly. In introductory programming courses, students’
disengagement due to a lack of higher order thinking skills acquisition using digital-oriented
environment is also observed (Lye & Koh, 2014). Preliminarily, because they cannot
understand how to solve a problem in a logical way, and next because they do not realize how
to construct a solution using basic programming constructs (i.e. loops, sequential
programming commands or if statements). The results of this process are negative for those
students who cannot utilize programming environments’ requirements sufficiently, neither on
their own, nor with other peers collaboratively. Furthermore, they should seek to understand
their management responsibilities to handle learning materials, following a well-defined
instructional framework (Pellas, 2014). Problem-solving activities in programming
environments are not easily achieved for the acquisition of fundamental programming
concepts during the conventional lecture-based teaching approaches. The cognitive value of participating in programming courses can assist students to produce or increase their cognitive thinking skills, firstly by understanding the problematic situation, in which they are involved, and secondly by recognizing how to configure the main problem into smaller pieces correctly. To this notion, three are the main objectives that appeared as significant (Lye & Koh, 2014; Su et al., 2015; Wu et al., 2014): (a) the understanding of students’ socio-cognitive background aiming to find a solution in a problem and then to use basic programming commands with a central concern to recognize the situation that need to be solved using their computational thinking skills, (b) the instructional process that should be followed. Specifically, a “constructivist-oriented” instructional framework can help students to use problem-solving skills, based on the deconstruction of the main problem in smaller pieces, and (c) the proposal of solutions in practice-based tasks using tools of a (visual) programming environment, although that in other times, lack of engaging tasks for students’ motivation and participation is also highlighted.

Secondly, from a technological-functional perspective, digital-oriented programming environments assist students to overcome these problems. However, regarding the utilization of visual programming environments, previous studies (Lye & Koh, 2014; Pellas et al., 2014; Repenning et al., 2010) have reported the following restrictions: a) students’ misconceptions in understanding the programming signification based on the rigorous syntax of text-based computer languages (like Java or C). This process can become frustrating for novice programmers of Secondary education, specifically when a digital (window-based) environment is utilized without the simultaneous execution of commands not being observed so easily; b) management responsibilities of students using a graphical user interface (GUI) that produced only in a digital “faceless” (text-based) environment, without their visual representation during the execution of programming commands to be available; c) the
maintenance cost of a learning environment for the coordination and organization of all activities, providing a large amount of information that cannot be easily assimilated from all students; and lastly d) conventional (lecture-based) instructional formats cannot facilitate students’ efforts to find several solutions to a problem adequately, due to the lack of comprehension of the programming commands’ functions or modes. As a result, students do not make an effort to improve their cognitive development.

Students (aged 12-16 years old) always try to develop a new culture in their personal technological literacy, when they try to gain a more deep familiarity utilizing an innovative computer system. The process of understanding these new technologies has been emerged so quickly and have radically that also changed users’ daily lives applying to their higher order thinking skills (Carbonaro et al., 2010). To date, a variety of visual programming environments with a two-dimensional (2D) or three-dimensional (3D) GUI (see for example Scratch, Alice project, Wu’s Castle, Kodu or Karel the Robot etc.) have been widely used. These programming environments allow novice (or not) users to construct programs and investigate different solutions in problem-based situations for acquiring or enhancing their personal skills. Unlike visual programming environments, such as Scratch, Alice and AgentSheets, users do not need to engage with computer programming from the outset, concerning for the creation of something meaningful. This situation initially creates a “steep learning curve” that cannot facilitate students’ motivation or provide to each one the confidence to persevere in various programming activities, without a specific storytelling or context. As a result, they need to spend much more time to customize or to learn how to create, rather than how to program something meaningfully (Howland & Good, 2015).

Compared to 2D environments, the international literature in Secondary education (Dickey, 2005; Hew & Cheung, 2010; Zuiker & Ang, 2011) has already preconceived 3D multi-user virtual worlds as candidate platforms for the implementation of learning scenarios.
in different educational levels. Constructive, problem-solving, collaborative and interactive learning activities are only some of the most significant pedagogical approaches that can be conducted sufficiently in these “worlds”. Instructional technologists (Cooper et al., 2009; Tüzün et al., 2009; Whitton & Hollins, 2008) have also disclosed students’ needs for participating in better instructional conditions, concerning on how to learn meaningfully by interacting with their peers (verbal or non-verbal) simultaneously in real-time.

Although Dickey (2005) has already noticed that the in-built tools of 3D multi-user virtual worlds present for high school students a high-floor (i.e. steep learning curve) hurdle to overcome, the utilization of 2D tools combined with constructionist learning approaches come to low these barriers for students’ engagement. To address the need in using a low-floor construction tool and to overcome these obstacles, Scratch4SL seems to be the most relevant programming environment for the implementation of high-ceiling/wide-walls learning scenarios (Girvan et al., 2013). The combination of Scratch for Second Life (Scratch4SL) is provided today as the most prominent and useful solution for students to learn basic programming concepts in a 3D multi-user virtual world. The GUI of Scratch4SL has low-threshold computational hardware requirements. However, its conjunction with Second Life, gives users the potential to produce high-impact (wide walls) capabilities and a wide range of applications that can empower their technological literacy (Pellas & Peroutseas, 2015). This environment was designed to be more easily accessible (low-floor) for novice-oriented programming activities. At the same time, by combining Scratch4SL with Second Life, powerfully complex-expressive (high-ceiling) tasks can be achieved. Both environments provide various opportunities for users to construct, explore and investigate complex programming structures in 3D interactive “objects-to-think-with” (objects with behaviour), known alternatively as “artefacts” without spatio-temporal constraints that 2D settings usually provide (Pellas, 2014). Scratch4SL does not generate Java outputs, like Scratch, but it
translates what someone “drags and drops” from the Scratch pallet to Linden Scripting Language (LSL) that is the core of programming primitives or artifacts in Second Life. The aim of using Scratch4SL is to yield the learning process and assist users (teacher and students) to separate or combine puzzle-based programming structures in the pallet. More specifically by using Scratch4SL, students add behaviors to interact with others in a common environment via 3D visual objects or artifacts, on the one hand, and freely sketch visually-rich artifacts via Scratch4SL to Second Life, on the other. These distinctions indicate that Scratch4SL can assist students to overcome the “steep learning curve” during their first time entrance in Second Life (Pellas & Peroutseas, 2015).

The present study argues that basic programming concepts, such as sequence, concurrent or conditional statements in programming courses can become a purposeful technological function for the introduction of novice programmers. However, the utilization of a well-established virtual environment and an instructional design framework which can offer well-organised tasks should always be recognised. This difficulty is expressed sufficiently when students spend much time on learning how to program, without being engaged in practice-oriented experiences, since they always need to think “computationally,” then construct programs, and to not only observe how others implement their programs independently. The need in utilizing a digital-oriented platform to facilitate students on how to program something correctly is totally defined as important for their motivation and participation. Toward to envisage this phenomenon, Second Life for introductory programming courses and Scratch4SL are combined in order to assist students to design and to prototype in visually-rich playable settings of the twenty-four (24) 3D artefacts (capital letters) of the Greek alphabet. Visualizing, coding and predicting the behaviour of these artefacts via Scratch4SL were the main tasks that students should construct, with the respect
to understand better how to use computational thinking skills in collaborative problem-based tasks via Second Life.

The research questions that should be answered according to all the above are the following:

1. *How can Second Life and Scratch4SL contribute positively on students’ motivation through collaborative problem-based activities underpinned by Constructionism?*

2. *Can a 3D multi-user/game-like created with Second Life and Scratch4SL influence positively students’ participation in introductory programming courses?*

The purpose of this study was twofold: (a) to determine the degree of students’ motivation in collaborative problem-based activities, and (b) to investigate the effectiveness in using the Scratch4SL 2D graphical (coloured block-based) palette combined with Second Life to learn basic programming commands and gain/enhance easier computational thinking skills of 28 high school students who participated voluntarily.

**Theoretical Underpinnings**

**Virtual worlds for e-Education: Inherent technological characteristics and affordances**

3D multi-user virtual worlds are defined as computer-generated environments responding in real-time to users’ (avatars) interactions. They can engage in game-like activities with 3D artefacts and communicate with their peers via synchronous or asynchronous communication tools (Bloomfield, 2007). Interaction among users can be achieved in visual and auditory stimuli. Users’ participation in 3D multi-user virtual worlds seemed to discern them as a powerful magnet, especially when they are geographically distributed, without preventing them to take incentives for socialization or collaboration with their peers. Furthermore, these environments have been widely used for the achievement of users’ common learning goals in different educational fields (Dickey, 2005; Pellas, 2014a).
The main feature of 3D multi-user virtual worlds is the common presence of all users for the implementation of collaborative activities that can be delivered by the reward of socialization in a community of avatars. Notable studies (Pellas, 2014; Wang & Burton, 2013) have provided several educational opportunities, by leveraging the virtual world of Second Life as a nascent 3D interactive platform for the implementation of constructive and collaborative activities which can be delivered successfully in online or blended instructional formats. A growing body of literature (Girvan et al., 2013; Pellas, 2014; Rico et al., 2011) has started to utilize 3D multi-user virtual worlds for the implementation of introductory programming courses in collaborative problem-based settings with positive learning outcomes. However, less attention was given on students’ motivation through collaborative courses following blended course delivery methods via 3D virtual worlds. Even more, further attention was not given to the implementation of introductory programming courses at a high school level, based on a specific organizational-instructional framework (Pellas, 2014).

Widely and well-documented are the examples of using 3D multi-user virtual worlds, including Second Life (Pellas & Kazanidis, 2014), Active Worlds (Dickey, 2010), and Croquet (Reis et al., 2011). All these “worlds” can effectively support student-centered models of learning underpinned by Constructionism fostering the development of students’ higher order thinking and cognitive skills (Leong, 2011; Yu, 2009). Second Life can become an alternative platform really attractive for students’ attention, because of utilizing technological characteristics, users can configure them to become ideal for the implementation of experimental or practice-based learning process. The reasons of using Second Life are described below (Pellas & Kazanidis, 2014):

- **The sense of (co-) presence**, i.e. the psychological immersion of “being there with others” that each user “feel” when he/she enters in the 3D virtual world. This distinction comes in contrast to verbal interaction of students’ face-to-face contacts
in conventional class supported instructional settings or in a “faceless” (window-based or text-based) 2D environment, in which all users can communicate via e-mails or messages asynchronously.

- The *persistent environment* (i.e. an environment that still exists, even when users log out of it and the changes that have been made are permanent, without students’ artefacts or developments to be lost) can give to all users the opportunity to reform and co-manipulate a virtual space or construct their meaningful and effortful constructs collaboratively.

- The *low financial cost* in creating a virtual environment for students’ participation for collaborative learning activities.

- The *social situatedness* using metaphorical representations refers to the execution and implementation of training courses with unique meaning or significance that can be delivered in constructivist-oriented instructional design frameworks.

- The *expressiveness* of interactive 3D graphics can be used properly from students or instructors to present abstract or complex object-oriented concepts using metaphors that are difficult to comprehend in textual forms. This process can help especially learners to interpret the environments or even construct their own interpretations and communicate easier with their teammates.

- The *high fidelity of* metaphors can assist the implementation of collaborative and interactive tasks, in which users are engaged by using visual artefacts (users’ configurable primitives) in real-time.

- The *combination of a 3D multi-user virtual grid with 2D environments*, like Scratch4SL. In this environment, students can utilize a block-based palette for creating 3D visual artefacts in order to govern “dynamic behavior” or understand the spatio-temporal
awareness in a 3D multi-user virtual world with specific programming structures to be introduced via “objects-to-think-with” (artefacts).

• The real-time feedback from their teacher or their peers optically and acoustically, with the respect to observe and encourage students’ exploration by utilizing visual tools or artifacts.

Second Life has offered major changes to various educational disciplines. Some of the most crucial aspects for practice-based learning tasks are as follows:

• The implementation of problem-based learning processes in synchronous communication modes (Good et al., 2008).

• The enhancement of social interactions among participants in order to be produced the anticipated learning outcomes following blended or online course delivery methods (Dickey, 2005).

• Collaborative learning using synchronous collaborative modes can increase students’ engagement (Pellas, 2014).

• The utilization of different instructional formats (distance or blended) in practice-based learning tasks for a variety of different educational levels. These tasks can also be combined with contemporary constructivist-oriented instructional design learning theories (Dimitropoulos et al., 2008).

• The creation and exploitation of innovative learning processes through interactive game-based environments in Second Life that can be easily co-constructed and co-manipulated not only per se, but also during the learning process by the same users, according to their needs and demands (Fiedler & Haruvy, 2009).

• The co-existence of participants in a common and persistent virtual environment (Salmon, 2009).
All along these lines, previous studies (Childs et al., 2012; Koole & Parchoma, 2012) have considered the following ethical issues that can be emanated using 3D multi-user virtual worlds: a) the risk of deception, is about users’ identities and their execution to change their real life names in order to not be recognized from others and make whatever they want, b) culture clashes for the ethics of interaction within the communities which may differ from those of the physical world. Users should respect different culture of each other peer, c) unintentional and deliberate griefing, is about the unwanted users’ participation that can be observed sometimes, and d) emotional attachment to a virtual world is based on the educator’s needs to be aware of the duty of care for students who may find the virtual world a place to which they engaged, and their avatar to become an important aspect of their identity, and e) rule-based solutions for questions should be provided under the ethics of an educational research community with guidelines as a point of view from which users must continue the in-world discussion.

**Constructionism as a proposed learning theory for programming courses**

Constructionism is a learning theory that supports learning as a means of constructing a new knowledge from each person. Learning is the meaning of construction, when each one seek to understand a new knowledge and gain it through interacting activities using his/her prior experience. The added value of this learning theory is consisted with human’s constructs. As a theory, it refers to anything understandable, when it is built in conditions that implied one on a person’s self-reported performance meaning (Harel & Papert, 1991). A constructivist paradigm can induce to students’ meaningful learning, critical thinking, and problem-solving skills. This paradigm regards learning as an individual’s activity, in which each one need firstly to think, then construct and after that share his/her idea/opinion with others to build upon their ideas a new knowledge domain that is formed and further shaped
through reactions and feedback of all, i.e. peers or teacher (Pellas, 2014). As Stager (2001) has also noted, Constructionism theory can give importance to the sense that each student worked on to acquire something valuable for his/her knowledge. Thus, an environment can become a truly valuable medium, not only when it allows the intellectual expression of emotional feelings or ideas among users, but also if it allows the free exploration in order to propose solutions through realistic conditions. A learning approach based on Constructionism is meaningful, when each student develops his/her own “knowledge domain” based on prior knowledge that is already existed within a social context (Papert, 1980).

Hung (2008) has referred that problem-solving instructional contexts can play an important role in helping younger schoolchildren learn and understand how a programming task is performed. Thence, an instructional method is appropriate, when it is used to help students establish their learning objectives and assess their progress toward these objectives. A substantial number of studies (Cooper et al., 2009; Paliokas et al., 2011; Pellas et al., 2013) have tried to redefine students’ interest in the early stages for learning basic programming constructs using 3D programming environments. Kelleher and Pausch (2005) have pointed out that by understanding programming structures via an easy-to-use 3D GUI, students can acquire skills for a logical thinking to solve a problem. Therefore, two are the important challenges that defined as crucial: (a) how to engage students in meaningful activities within a well-organized instructional framework and (b) how this framework can facilitate a learning process, in which students are able to acquire an option, in order to construct and finally propose alternative solutions via a digital-oriented environment. Kommers (2003) has mentioned that the advanced psychomotor skills can be gained successfully in constructivist-oriented learning theories via a 3D game-like environment, due to the high representational fidelity. Li et al. (2013) have stressed that 3D game-like learning environments are able to facilitate students’ activities and foster in-depth learning strategies to a long-term. The same
authors have also highlighted that constructivist-oriented learning theories can help students who have low cognitive background to participate easier in programming courses using a game-like environment.

Therefore, it is observed an urgent need for the development and implementation of an instructional design framework based on constructivist approaches. Such an approach can be introduced on the one hand for novice programmers (high school) students’ assistance for interactive tasks to motivate and participate in introductory programming courses on the other (Li et al., 2013; Pellas et al., 2013). These contexts can allow students to participate or develop their knowledge domain collaboratively via game-based environments, while at the same time they can gain problem-solving skills, can communicate and can exchange views/ideas in order to amplify better basic programming concepts with their peers.

Instructional technologists (Paliokas et al., 2013; Pellas et al. 2014; Rico et al., 2013) have suggested that by using 3D game-based environments to teach introductory programming commands based on Constructionism can be very advantageous for students’ motivation and participation. However, before students start to be introduced in a 3D environment, they should firstly be involved in low-floor programming concepts by using a low-threshold GUI in order to avoid abrupt information load (i.e. steep learning curve) that is rising. After that, students should have the opportunity to implement more complex programming structures and understandings on how to use tools or objects to more complicated (high-ceiling) activities. This process can assist them to gradually be engaged in fading scaffolding learning processes leading to more meaningful activities (Su et al., 13). This dimension is also important for educational technologists who want to design a socially-rich constructive learning activity. Along these lines, Girvan and Savage (2010) have mentioned that learning approaches based on Constructionism can provide an additional
opportunity for students to participate in more meaningful tasks, by leveraging the technological potential and instructional affordances of 3D multi-user virtual worlds.

The above literature suggests that Constructionism can be really useful theoretical foundation to empower students’ higher order or programming skills in game-like learning environments in favour of: (a) engaging students in a 3D persistent environment and managing control learning activities more easily and (b) facilitating an exploratory learning activity. Like active participants, students test and self-reflect upon their ideas to complete a task. According to the potential benefits, this study hypothesizes that an instructional design framework underpinned by Constructionism can enhance students’ learning experiences in game-like settings. Thence, a study using Scratch4SL and Second Life was conducted to prove this assumption.

Using 3D visual environments as instructional means for programming courses

From 2005 and now on, several research efforts seek to investigate the enhancement of students’ motivation in collaborative activities. The vast majority of studies (Brennan & Resnick, 2013; Kafai & Burke, 2013) have pointed out that it would be useful for students to participate in a digital environment not only to learn actively, but also to exchange ideas and communicate with other peers with a purpose of solving complex learning tasks. Harlow and Leak (2014) have considered as particularly important the role of collaboration among students in programming courses. The same authors considered collaborative approaches as more effective than class supported which are based on the instructor’s lectures. Teaching basic programming concepts and specifically sequential or concurrent commands is imperative, because students have trouble in understanding the synchronization of a schedule for the computational thinking skills usage in problem-solving learning tasks (Feng & Cheng, 2014; Lee, 2011). Fields et al. (2014) have found that asynchronous forms of communication
using Scratch 2.0 affected positively 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 collaboration, inasmuch as collaboration among students using only asynchronous communication tools have not satisfied and engaged them to participate extensively. To these demands, a need for utilizing a 3D multi-user virtual world in different instructional format has been also raised for two reasons: firstly, due to the use of a visual (game-like) and persistent environment owing to understand basic programming structures better, and secondly due to the use of various a-/synchronous communication and built-in tools which are freely available.

Werner et al. (2009) have argued that a 3D visual (game-based) environment can help students to understand basic programming structures better. Recent studies (Fields, et al., 2014; Repenning et al., 2011) have shown that better results regarding the socialization of students to participate in learning activities can also lead to positive learning outcomes. Li et al (2013) have opined positively to the use of an instructional model based on Constructionism, with students to know from the outset their actions. This may counterbalance to some extent the complexity that encountered through users’ first time entry. 3D visual programming environments have an added value for programming courses causing students’ persistent motivation and engagement, due to the easy creation from specific visual representations of a program. Storytelling Alice is another 3D visual programming environment that can be used for creating animations to tell a story, playing an interactive game, or creating a video to share on the Web. However, in this environment, users only modify objects and camera’s movements. Of course the movements of a virtual character with a tool gallery have its own limits (Kelleher & Pausch, 2005). For example, students who use Storytelling Alice were immediately able to view programs running in real time. All of their actions were related with programming "blocks" leading to an understanding of the
actual functioning for a variety of programming language structures through digital animated scenes (Harlow & Leak, 2014; Kelleher & Pausch, 2005). In fact the benefits in using 3D visual environments like Storytelling Alice or Kodu, are compromised by the development of an intuitive understanding of the basic concepts, in which visual or acoustic feedback is provided.

Both “debugging” and “feedback” processes in a 3D multi-user virtual world are visually and acoustically accessible to all students who want to correct mistakes that have been identified by their fellows or the instructor. This feature can be very helpful and insightful for understanding a learning process in collaborative settings better. Second Life provides a common 3D persistent multimedia environment that offers the simultaneous existence of (distributed or not) users with the abundance of realistic metaphors and visual objects or artifacts mimicking real life situations.

Based on this view, students’ interactions and activities should be adopted in a 3D multi-user environment and should firstly attract their attention in order to improve their thinking and creativity skills. Students can also create meaningful activities, in which they interact with others to (co-) construct their own meaningful knowledge domain using a 3D computer-mediated world.

A rationale to conduct this study

The statement of this case study has followed Dalgarno’s and Lee’s (2010) declaration. The authors claimed that further studies need to be conducted so that understand Computer Science teachers how the technological and instructional affordances of 3D multi-user virtual worlds must pedagogically be exploited further. The inspiration of this study comes from the contribution rising from the combination of Second Life with Scratch4SL for various engaging programming activities, due to: a) a 3D low-cost persistent virtual grid
which can assist the implementation of different instructional formats (blended/online); b) the technological features which can produce real-time feedback on users’ interactions to create and syntax multiple coding schemes in visually-rich or in realistic problem-based learning settings; c) the a-/synchronous communication tools and realistic aesthetics of a 3D virtual grid that permit users to engage in situations that mimic those of real life; and last but not least d) the flexibility and adaptability of co-manipulating a 3D virtual grid can assist users to create at the beginning a learning environment, according to their needs or demands. This process helps them to organize/coordinate their teams and enhance the sense of co-presence to achieve common objectives.

A rationale behind the intervention of Scratch4SL and Second Life for introductory programing courses was inspired by the related literature (Girvan et al., 2013; Pellas et al., 2013) that have stated to the following points of view:

- Scratch4SL can become a very important tool for students in order to understand the contemporary functions of a learning environment. The GUI of Scratch is well-known today in Secondary education. However, programming courses can give students many benefits, but undoubtedly convey more complex activities that create positive or negative challenges to novice users during their engagement. Many times they express dissatisfaction regarding to the value and the appropriate skills that can be gained.

- The construction of meaningful (ill-defined) 3D collaborative problem-based activities cannot be implemented throughout 2D programming environments in real-time.

- The increasing number of high school students who play online games (see World of Warcraft or EverQuest II) so that recognize Second Life as another candidate platform. Second Life can provide unique features that can attract students’ attention due to the personalization of their avatars, the multimedia 3D interface and of course
the potential of creating interactive artifacts in visually-rich settings through game-based environment that entails all the above features.

Before the learning process, two difficulties should be overcome. Firstly, Second Life has been addressed for adults and many high school students were skeptical about its value, and secondly the development of a 3D virtual grid requires users’ subscription, with their real first (sur-) names were the perquisite factors for the construction of their avatars. The two obstacles circumvented by providing a collaborative space in a secured virtual island (grid) for all users (teacher and students). The virtual grid was secured solely for this study and it was prevented the entrance to unauthorized users. The access was possible only to those who had personal accounts with their real names. Incidentally, all students seemed to use both versions of the programming environments: (a) the standard version of Scratch\(^1\) and (b) the qualified version of the Scratch4SL\(^2\), and other programs as Flashlight-VNC, TightVNC and Xampp for their access. Students’ personal computers in supplementary online courses were also used.

**Challenges in using Scratch4SL and Second Life**

The worldwide interest in utilizing 3D multi-user environments has astonishingly been increased in the last eight years. Educators and scholars seek to investigate the possibilities for students to collect the learning material for knowledge acquisition easier. In-world collaboration or communication among students can produce positive learning outcomes rising from their interaction in tasks, which are otherwise difficult to thrive in the conventional class supported settings (Zheng et al., 2009). Nevertheless, a perception that should be taken seriously into account before the combination of both environments is the sense of “presence” as an integral part of the entire experience that users need to feel.

\(^1\) [http://scratch.mit.edu/](http://scratch.mit.edu/)

\(^2\) [http://web.mit.edu/~eric_r/Public/S4SL/](http://web.mit.edu/~eric_r/Public/S4SL/)
Whereas this term is often seen as a “state of the mind” on the one side, while the term of psychological “immersion” is an in-world experience, in which users are involved in learning scenarios that are implemented is on the other (Ruggeroni, 2001). This occurs through the GUI that is presented on the user’s computer screen as twofold active contexts, including one 2D environment (Scratch with moderately-present GUI) and another one 3D environment (Second Life with a highly-presence GUI). In these circumstances, learning environments with simple 2D graphics, like Scratch did not really provide the sense of students’ presence (i.e. it is unlikely to feel like “being there” in a-world of falling puzzle blocks). This process becomes highly immersive for constructivist-oriented and collaborative activities, in which (distributed) users interact as cyber entities (avatars) in a common virtual grid. Besides students’ entrance, the fundamental characteristics of 3D multi-user virtual world technology that should be mentioned, such as “presence” and “immersion” cannot exist without creating and managing the learning materials. Thus, the acquired not only from the students’ side, but also from teacher’s feedback in real-time (Pellas, 2013).

The programming language of Scratch allow students to create a wide range of animations, interactive games, and music applications. Novice developers (students) construct new scripts and become familiar with important ideas from Mathematics and Physics to Computer science, as they try to consider creatively how to solve problem-based situations. Scratch offers a programming language that displays in front of the user’s computer screen many intimate features that help students to create various animated programs (Maloney, 2010). Scratch for Second Life (Scratch4SL) is a programming tool, in which users can simultaneously work in two different environments (i.e. a learning 2D block-based environment and another one 3D environment that must be modified for educational purposes). The puzzle-based programming structures of Scratch4SL pallet allow users to add behaviors into visual objects or primitives of Second Life. The core of Scratch4SL is based on
the 2D environment of Scratch, with its basic choices to be focused on a graphical programming language that allow the writing of programming languages via colored sprites. Supplementary, students can construct programming objects (primitives) or artifacts in the Second Life by interlocking together graphical blocks and utilizing visual artifacts ("virtual prototyping processes") with a “stylus” (pen), urging students to construct, modify and customize collaboratively without physical distractions. The puzzle-based configuration of programming commands in combination with LSL can promote the direct visual feedback to students at the time that they want it, as they can interact with others via 3D multi-user virtual world and consequences of their actions can become clearly qualified on their computer screen.

While Scratch4SL provides “low-floor” (ease accessible) programming structures, it is important to be aware that its conjunction with Second Life can potentially assist users to create “high ceiling” (powerfully expressive) artifacts in 3D visual prototyping. Graphical blocks are selected and dragged to the Scripting pallet of Scratch4SL, since students seek to construct a concurrent code that can be depicted in Second Life. By clicking on the “Copy Linden Script” button in the user interface, Scratch4SL code is compiled equivalent into the LSL code. Each user in Second Life can create a default script, copy from Scratch4SL’s programming language, and then he/she must paste the code and replace it in 3D visual object scripts in order to complete the process.

**Research Methodology**

**Design of the study**

In the current case study, a learning activity with a small sample of students (n=28) was conducted in a 3D game-like environment. A 3D “game-like” environment is a simulation-based learning environment that includes a prototype game with playable
characteristics which is inspired by serious games or game-based environments (Li et al., 2013).

In order to better understand students’ aspects and conclusive insights, it was decided to present students’ opinions, which analyzed from interviews (before and after their intervention) and from their answers in a closed-ended questionnaire to be measured their motivation respectively. The data were collected, firstly by using an interview with an open-ended questionnaire to encode students’ experience in focus groups, based on their interactions as avatars in collaborative problem solving activities that held in Second Life and secondly by a closed-ended questionnaire to be measured their in-world motivation (Glynn et al., 2009). The students’ answers from the interview process were conducted in order to be checked the collected data and thence all questions were started with respect to: (a) the students’ overall experience and evaluation of all sub-units of these lessons, (b) the assessment of students’ achievements in learning goals, and (c) the estimation of students’ abilities and difficulties, when studying collaboratively with their peers.

Sample

The sample consisted of twenty-eight ($n=28$) students aged between 14-15 years old with thirteen (13) female and ten (15) male who participated after the signed permission of their parents. Students who finally had difficulties in learning how to program and then comprehend a proposed solution to a problem. Moreover, they have not got an extensive experience to the execution of programming structures through other innovative learning platforms, but only with Scratch, Kodu and Alice. Indeed, they had only two weeks before getting involved in both environments regarding the initial conditions of their exposure via Scratch4SL and Second Life. Some important demographic information that should be referred are as follows: a) 20 students had from mediate grades (13-16/20) and only 8 of them
had from mediate to high grades (17-20/20), b) 12 students have daily visit in Web-based environments or transactions, 12 of them only monthly and 2 have not visited any Web source. Kodu and Storytelling Alice were two 3D visual programming environments that students had the opportunity to learn how to code. However, they had many difficulties on their usage (16/28).

Materials and instruments

In both questionnaires, the anonymity of students was kept and all questions were translated in the Greek language. Firstly, for the qualitative analysis, one quantitative was utilized, according to students’ opinions and aspects. For the quantitative analysis SPSS (ver. 22) statistical package was used for data processing and reliability. The quantitative research separated in two subsections. The first had questions for measuring students’ experience in games. The game experience questionnaire asks two closed questions about participants’ previous experiences with video games (e.g., 1. For the last week, how many hours for the average week did you spend on playing video games or massively online role-playing games-MMORPGs? 2. For the last week, how many hours for the average week did you spend on social networks?). Answers were given in predetermined categories for ranges of hours or experiences.

The second questionnaire for the measurement of motivation was based on a 30-item validated instrument that proposed by Glynn et al. (2009). This questionnaire provides the relevant evidence of construct validity, revealing how students’ motivation affect learning. This process consisted of five dimensions: intrinsic motivation/personal relevance, self-efficacy and assessment anxiety, self-determination, career motivation, and grade motivation. All questions were adopted according to the needs of this study. For example, while the first question was “I enjoy learning the science”, it has changed like “I enjoy learning the programming courses”. Answers were based on a 5-point Likert scale (1=strongly disagree to
Cronbach’s alpha in satisfactory levels for the questionnaire’s constructs ($a=0.75$) was identified. Results are obtainable in Table 1. The questionnaire of this Table was utilized for the measurement of students’ motivation and specifically the engagement their happiness to use both environments, the improvements of their understanding in computational thinking skills, how helpful courses were for them and the challenges of coding with their peers collaboratively.

Secondly, focus groups supported this study for the following reasons: (a) the purpose of this research method was to generate participants’ ideas in more conducive settings. The researcher’s role was to moderate the conversation of each group, (b) each student group was separated in heterogeneous of 4-5 students and each one of them called by interpersonal note cards, (c) all students who attended in the interview were sitting near to the researcher’s avatar and shared their ideas, opinions and perspectives, (d) the focus group discussions encouraged students to collectively afford even a common opinion about their actions. This process may have a risk, because some students may not have the time to express opinions freely.

Dialogues were registered via a digital voice recorder, because all interviews were conducted in Second Life, written out in digital documents and scored with an observation scheme based on research literature for a digital-oriented environment that proposed by Hämäläinen et al. (2008). The scheme proposed a classification of dialogues enacting into different functional levels of gameplay (see Table 2). The analysis of the qualitative data was made via Nvivo (ver. 10) collection tool, as it should be aggregated all digital documents, putting all of them in a common basis and finally the main results from the interview extracted. Results are described in Table 2.

The interview was made during the final course by the main researcher and the Computer Science instructor who helped on students’ coordination in the computer laboratory
with the respect to identify better their attitudes or reactions from: (a) the utilization only of Second Life and Scratch4SL, (b) the correct creation of programming structures using sequence, conditional statements and iteration commands in collaborative problem-based activities, and (c) the empowerment of a collaborative climate among students. Both the instructor and researcher have suggested (independently each one) that all activities considered as noteworthy (based on what was happening inside the computer laboratory or not) either in online or face-to-face courses.

**Treatment**

Computer Science educators need to face several problems in the teaching and learning process. Many high school students enter in introductory programming courses, without being prepared to think “computationally”, while they are generally exposed to numeric computation (Cooper et al., 2010). Moreover, a significant problem is that students are not often sufficiently able to develop a formal step-wise algorithm and propose a solution to a problem. Other skills to this notion cannot be gained in areas, such as word problem solving in mathematics adequately and the formal specification of a problem. Thus, students need to learn how to (Cooper et al. 2000; Pellas, 2014):

a) identify and understand correctly the problem statement,

b) separate the problem in number of well-defined smaller pieces, and

c) design a step-by-step solution to solve each of the sub-tasks.

The above skills are defined as very crucial to ensure the student’s success in introductory programming courses. In this demand, an innovative way for students to use Second Life and Scratch4SL for developing programming thinking is presented. The statement of the present case study is that the combination of Second Life and Scratch4SL can
become valuable platforms for supporting students’ motivation to participate in collaborative problem solving using their computational thinking skills.

The aim of this process covered the same learning material with 28 students who participated in blended course delivery methods using Scratch4SL and Second Life. This treatment had a twofold objective. Firstly, to familiarize students in basic programming concepts, such as sequence, iteration or loops through the creation of artifacts in a 3D multi-user game-like environment and finally to present their creations to their peers and teacher. Secondly, to assist students who enrolled in programming courses to study with their peers collaboratively, by coding structures in 2D Scratch4SL environment. Thereafter, the task was completed by combining the code to predict or move 3D artifacts (via 3D stylus) and complete each letter in Second Life. This process enabled students to develop and produce alternative solutions in problem-based learning situations through a 3D multi-user game-like environment, in place of strengthening further their technological literacy skills. They planned to utilize Scratch4SL in an identical educational 3D virtual grid, in which other peers need to see the results in common. This cannot be observed in the 2D "scene" of Scratch. The difference between 2D GUI of both environments is depicted in Figure 1. Four (4) basic courses were implemented in the school’s computer laboratory and another two (2) at a distance (each one in every week). Each team had five “experts-leaders” and their role included the leadership by moderating each team’s actions to accommodate the perceived students’ processes in virtual prototyping that provided each time. The instructor’s role was more facilitating, to ensure the successful combination of all project’s parts. Students with this role were chosen from the Computer Science teacher, and the criterion was their high grades in this course. The teacher’s role was more facilitating, in order to ensure the successful combination of all parts to be developed one Greek letter each time.
The process lasted six (6) weeks in the second triminister of 2014. Before starting the research process, it was appropriate for the Computer Science instructor to study the following issues: (a) the access and navigation in Second Life from the school’s computer laboratory, where all users attended to the main laboratory (the physical environment) and in online sessions through Second Life (the artificial environment) respectively and (b) the main technical issues should be resolved when students were attended in Second Life for supplementary online activities. These courses were at a distance, as the researcher wanted to see students’ behaviors and reactions in relation to their class-supported (face-to-face) experiences. The students’ answers from the interview recorded to provide positively or negatively their reactions in the computer laboratory. In the interview, all participants have presented their personal opinions and aspects about the use of Second Life and Scratch4SL. The data obtained to address the adverse conditions before the research process was started. Students needed to participate in laboratory courses as extracurricular (supplementary) online activities in collaborative problem-solving exercises that held in Second Life.

Despite the similarities, other significant differences appeared as significant too. Figure 1 shows the structures in Scratch4SL “when I am creating” (left column, fourth structures), without corresponding in Scratch with the structures “when the space key pressed” (left column, the second structures) were used. When trying to use the same programming structures, there was not identified any difference in writing to the palette of Scratch4SL, and thus the learning material as artifacts to be depicted more easily in Second Life.
Design guidelines of a game-like environment and instructional support based on Constructionism

"Programming in 3D technology-advanced environments" and the activity of "Designing and prototyping twenty-four (24) capital letters of the Greek alphabet using Scratch4SL in Second Life" was titled in this school task. A preliminary two-hour lesson for all students was dedicated to the description of the whole process that should be followed in acquaintance with Second Life, the disclosure of users’ accounts and their guidance to install the necessary programs on their personal computers. Only four (4) basic courses were implemented in the school’s computer laboratory and another two (2) at a distance.

The first-part for the construction of the overall twelve (12) letters in two-hour courses was decided. The first three courses were devoted to familiarize all participants with Scratch4SL in pursuance of creating different in-world visual artifacts by drawing basic geometric shapes. All artefacts should shift to the right shape or height in favor of designing easier each letter in the 3D virtual grid. The learning of the programming structure’s recurrence to all Greek letters with curves (e.g. Y and Ω) that used was also necessary. The
uppercase letter “A” was created by an equilateral triangle. The next two courses were devoted to the construction of the remaining twelve (12) letters.

The levels of difficulty (a fading scaffolding process) for the construction and programming of letters vary (each one letter has its own particularity, inclined lines, circles, and parallel lines), as it was felt more appropriate to separate them into two categories: (a) simple (Γ, Δ, Ε, Η, Ι, Λ, Ξ, Ο, Π, Τ, Υ, Χ) and (b) complex (Α, Β, Ζ, Θ, Κ, Μ, Ν, Ρ, Σ, Φ, Ψ, Ω). Students assigned a letter from each category, so that the activity could be graded easier.

Among letters of the same category there were variations in the degree of difficulty, which did not make additional categorizations, because of the additional complexity that could be created during the process and this could provide students’ distraction.

Each letter was constructed by two students in one desktop computer or laptop (all students were separated in pair because computers were less than the number of participants) in a computer laboratory and they separated not only according to their age-homogeneity, but also with different cognitive-heterogeneity, in order to help and learn their objectives with other peers simultaneously. This process was implemented with two letters, one simple (e.g. Π) and a more complex (e.g. Φ) for each student so that create an imaginary chain, in which each student had the same letter to the next and the other same as the previous one. The last lesson was only for collaborative activities among students for the integration of the individual programs in order to construct and conjunct all letters that they finally programmed through Scratch4SL to write a Greek word (e.g. “ΦΩΤΙΑ”).

This study seeks to investigate students’ motivation with the purpose of getting used the process of communication and collaboration with their peers to construct their own learning material in a 3D interactive environment. Collaborative problem-based activities provided in Second Life, because it could promote students’ reflections and enhance at a large extent the learning process. Second Life enables collaboration among users (avatars) via synchronous
communication tools in a common virtual grid to edit the same primitive or share the same code by programming via Scratch4SL. All users (students and teacher) had the opportunity to access and leave in-world objects or messages to the other members (group of 4 avatars as a team or private messages-IM). When a user seeks to provide a solution on how to program or collide a primitive successfully logged in all his/her messages and he/she could see all the objects left in this world by others when permissions were activated. Accepting Reigeluth’s (1983) proposal for using instructional design methods focused on means to attain given goals which offer guidelines for students’ activities to achieve a given goal, this study suggests an instructional design framework based on Constructionism. The two environments are predominately depicted in Figure 2 in place of assisting Computer Science instructors to administrate and coordinate a learning process based on learning principles that Constructionism as a theoretical foundation for the development of an instructional design framework.
Figure 2: The overview of an instructional design framework based on Constructionism

The multiple options of collegiality and social aspects can be assumed as meaningful through the modeling and designing different artifacts in a 3D multi-user environment that has attracted students’ attention to understand better the value of this learning process. Indeed, students seemed not totally isolated in front of a personal computer screen, but their actions acquired some beneficial features of the “persistent co-presence” in a common virtual environment. This concept is intermittently envisioned in the contexts of a collaborative effort with the users’ coexistence and “co-presence” in virtual teams to be reinforced.
Students have started to study collaboratively with some exercises to be assigned for them in order to meet with other peers (scheduling their time presence in the 3D multi-user virtual world) and all together to configure the educational virtual area and start to program artifacts (letters of the Greek alphabet). The data obtained through a process that was found necessary to answer and amplify the research questions through interviews from students, according to their in-world interactions in collaborative problem solving activities.

**Results and Discussion**

This study results revealed some very interesting results, which are the following:

*I. Users’ experiences with video games:* All students have played video games, like Doom or Grand Theft Auto (quest. 1-65.2%) indicating their experience in utilizing 3D desktop environments. This seems to have the subsequent use of other 3D multi-user virtual environments; however with the participation of students in online games (MMORPGs) (quest. 2-74.2%), they have similar experiences attended with others in distance learning environments. At almost all cases, they have also spent more than three (3) hours for playing MMORPGs or other online video games.

*II. The user’s personal needs for communication using social networking systems:* The first question (quest. 1) showed that 70% of students have used 1-3 hours social networking sites like: (a) Facebook (84,3%), (b) YouTube (37,7%) and MySpace (52,4%). The connection of these instruments (quest. 2) reflected by exchanging image files and audio from the Second Life to YouTube (75,7%), acquaintance with the profile of the “real” lives through Facebook (55,7%), while others worked in groups and on-line environments, such as War of Warcraft (84,3%).

At the beginning of the school year, students’ familiarization with basic programming commands was delivered using Scratch, Kodu or Storytelling Alice. Unfortunately, students
were not ready to implement several programming learning scenarios. As a result, whatever they have learned before Scratch was not easily understandable if it can be implemented in Second Life, and thus all students spent some time to learn. Notwithstanding students quick adaption for the construction of geometric shapes was enacted; they were also impatient for the next desired shape, to complete successfully each programming structure that entrusted to them. Consequently, during the learning task on how to co-construct Greek capital letters, students were very creative, as it was proved by the proportion of alternative solutions, using the “stylus” (pen) of Scratch4SL. In the last part of the integration in the learning process, students have showed great satisfaction and enthusiasm for their final creations in Second Life. Students were positively surprised from their collaboration in supplementary (online) outside of a regular school’s schedule. The teacher’s assistance was provided timely and at the same time to all students during the course and in the same virtual grid that each team had at the begging of the each course. Students’ answers revealed the positive acceptance of how the instruction in virtual settings was described in these contemporary situations supposing to be familiar with "innovative" and "interactive" learning situations, after passing hours using both platforms. Since some students stated that "it was a boring learning task”. This was characterized by a small portion of students somewhat, because they had some difficulties during their first introduction, despite that in their vast majority found the learning process "more interesting”.

The study results that unveiled from the analysis of qualitative and quantitative data are remarkably to provide information about new dimensions that programming courses can include. Based on the corresponding data, students engaged positively to the entire study. Generally, students have not found any difficulties to use interactive built-in or communication tools when they interacted with others and when they created visual artifacts. Students without being asked took initiatives on their own to co-construct (different in each
time) their own artifacts by adding relevant programming structures. Nevertheless, they obtained that the combination of Second Life and Scratch4SL was demanding compared to the 2D conventional. This difficulty made some students to be disappointed and provoked at the beginning with some uncomfortable reactions to be observed. However, when they accustomed to study collaboratively with their peers in Second Life, Scratch4SL seemed very functional for each one, especially in the part of writing basic programming structures by using puzzle-based structures into Second Life.

Table 1: Questionnaire for measuring students’ motivation in introductory programming courses (adapted from Glynn et al., 2009)

<table>
<thead>
<tr>
<th>Questions</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>01. I enjoy learning to program using Scratch4SL and Second Life.</td>
<td>3</td>
<td>5</td>
<td>4.02</td>
<td>1.82</td>
</tr>
<tr>
<td>02. The gained programming structures are related to my personal goals.</td>
<td>3</td>
<td>5</td>
<td>3.82</td>
<td>1.22</td>
</tr>
<tr>
<td>03. I like to collaborate with other students in programming exercises.</td>
<td>3</td>
<td>5</td>
<td>3.74</td>
<td>1.79</td>
</tr>
<tr>
<td>04. I am nervous about how I should participate on the programming exercises.</td>
<td>3</td>
<td>4</td>
<td>3.65</td>
<td>1.69</td>
</tr>
<tr>
<td>05. If I am having trouble learning the programming structures, I try to figure out why.</td>
<td>2</td>
<td>5</td>
<td>3.45</td>
<td>1.45</td>
</tr>
<tr>
<td>06. I become anxious when it is time to take an exercise in programming courses.</td>
<td>2</td>
<td>5</td>
<td>3.91</td>
<td>1.71</td>
</tr>
<tr>
<td>07. Earning a good grade in programming courses is important to me.</td>
<td>1</td>
<td>5</td>
<td>3.22</td>
<td>1.22</td>
</tr>
<tr>
<td>08. I put enough effort into learning various programming structures.</td>
<td>1</td>
<td>4</td>
<td>3.14</td>
<td>1.11</td>
</tr>
<tr>
<td>09. I use my personal strategies to ensure that I learn how to abstract and then command programming structures well.</td>
<td>2</td>
<td>4</td>
<td>3.77</td>
<td>1.57</td>
</tr>
<tr>
<td>10. I think about how learning the programming can help my thinking skills.</td>
<td>1</td>
<td>4</td>
<td>2.98</td>
<td>1.09</td>
</tr>
<tr>
<td>11. I think about how programming courses will be helpful to me.</td>
<td>2</td>
<td>5</td>
<td>3.88</td>
<td>1.31</td>
</tr>
<tr>
<td>12. I expect to do as well as or better than other students.</td>
<td>2</td>
<td>5</td>
<td>3.47</td>
<td>3.47</td>
</tr>
<tr>
<td>13. I worry about failing on expanding my personal programming skills.</td>
<td>2</td>
<td>5</td>
<td>3.99</td>
<td>1.55</td>
</tr>
<tr>
<td>14. I am concerned that other students are better in programming courses.</td>
<td>3</td>
<td>5</td>
<td>3.97</td>
<td>1.67</td>
</tr>
<tr>
<td>15. I think about that my grade in programming courses will affect my overall grade point average.</td>
<td>2</td>
<td>5</td>
<td>3.55</td>
<td>1.55</td>
</tr>
<tr>
<td>16. The programming courses that I learned is more important to extend my personal skills than the grade I receive.</td>
<td>2</td>
<td>5</td>
<td>4.02</td>
<td>1.02</td>
</tr>
<tr>
<td>17. I think on how learning in the programming course can help me with other courses.</td>
<td>2</td>
<td>4</td>
<td>4.88</td>
<td>1.88</td>
</tr>
<tr>
<td>18. I hate taking the programming tests.</td>
<td>3</td>
<td>4</td>
<td>3.56</td>
<td>1.62</td>
</tr>
<tr>
<td>19. I firstly think logically and then use the programming structures to solve a problem.</td>
<td>2</td>
<td>4</td>
<td>2.77</td>
<td>1.37</td>
</tr>
<tr>
<td>20. It is my fault, if I do not understand the programming courses.</td>
<td>2</td>
<td>5</td>
<td>3.98</td>
<td>1.87</td>
</tr>
<tr>
<td>21. I am confident I will do well on the programming labs and other supplementary online courses.</td>
<td>3</td>
<td>5</td>
<td>3.58</td>
<td>1.55</td>
</tr>
<tr>
<td>22. I find learning the programming commands interesting.</td>
<td>3</td>
<td>4</td>
<td>3.16</td>
<td>1.62</td>
</tr>
<tr>
<td>23. The programming structures I learn are relevant to my daily life extension.</td>
<td>2</td>
<td>4</td>
<td>3.38</td>
<td>1.28</td>
</tr>
</tbody>
</table>
24. I believe I can master the knowledge and skills in the programming course.
25. The programming I learn has practical value for me.
26. I prepare well for the programming feedback from the instructor.
27. I like to participate in a programming course because it challenges me.
28. I am confident that I will do well on the programming tests.
29. I believe I can earn a grade of “A” in the programming course.
30. Understanding the programming structures gives me a sense of accomplishment.

N, number of items; M, Mean; SD, Standard Deviation

By answering the main research questions according to the above data (see Table 1), several advantages from the utilization of Second Life and Scratch4SL about students’ motivation and participation in programming courses are observed. Students in these lines tried to learn in collaborative problem-based tasks, and finally found helpful the entire treatment. They also felt more confident on answering to a problem statement by taking as well as initiatives to program correctly their structures. Worth noting is the fact that finally students: (a) engaged in collaborative practice-based tasks through typical class supported instructional settings and online supplementary online to the previous one via Second Life, (b) felt more confident by following a blended instructional format in which expanded their personal skills, (c) changed their opinions on how helpful programming course maybe, giving students the sense of accomplishment, (d) self-efficacy in several tasks was enhanced and last but not least (e) carried out greater with their personal management responsibilities in a collaborative climate with their peers in a 3D game-like multi-user virtual world.

According to the quantitative data, students were motivated and the conclusion of this study extracted the following outcomes: (a) better students’ understandings on how to think logically in programming activities that they are engaged, (b) the enhancement of students’ problem-solving skills in collaborative learning settings also helped them to study in blended instructional formats with their peers, and (c) students finally acquired the confidence that was missing in conventional in-class instructional settings and they carried about their advancement in technological literacy skills.
Regarding students’ participation in Second Life, they had shown exciting expressions, according to their comments during the interview (qualitative data). Indicative examples of their excitement were the following words/phrases: "fantastic", "like a game with an avatar, but I learn collaboratively with my peers," while many of them denoted that "the lesson was very interesting as we constructed programming structures in a game-based environment."

For Scratch4SL usage, students claimed that "they have not previously so well being worked together as it was in this project". All students worked with at least one classmate and some notable answers from the instruction were the following: "first time someone stand by me in order to help me to program something"; "I studied much longer than in other courses"; "I love to solve the queries of my classmates"; "In a collaborative climate, I was asking my friends to help me and then, where it was necessary, my instructor"; "it was great that we could connect in the afternoon and solve the exercises all together"; "I was not usually solve exercises at home, but the learning process in Second Life was so valuable that when it was possible we tried to connect on it and studied with others together."

In Table 2, students’ answers before and after getting involved in Scratch4SL and Second Life were encoded. These are as follows:

**Table 2: Students’ perspectives before and after participating in the learning process with Scratch4SL and Second Life**

<table>
<thead>
<tr>
<th>Significant themes</th>
<th>Before getting involved in the learning process with Scratch4SL and Second Life</th>
<th>After getting involved in the learning process with Scratch4SL and Second Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Attending and motivating students’ using Scratch4SL to Second Life</td>
<td>(a) “The field of ICT and generally the Computer Science is quite interesting, but I cannot figure out the lessons associated with programming structures easily, because I cannot follow and watch the instructor’s guidance, when others do fuss.”</td>
<td>(a) “The course was conducted in collaborative contexts and it was provoked my interest and my personal motivation to acquire the programming constructs.”</td>
</tr>
<tr>
<td></td>
<td>(b) “I believe that Informatics is a good lesson, because it presents to us new concepts which are really helpful and insightful.”</td>
<td>(b) “It was finally a pleasant lesson enabled us to produce our knowledge by co-constructing programming structures in collaborative settings via Scratch4SL and Second Life.”</td>
</tr>
</tbody>
</table>
| 2. Empowering students’ collaboration in problem-solving situations | (a) “I wish to have more freedom to move or navigate in an environment to help others that cannot alone configure | (a) “I believe that cooperation with other peers in the virtual environment was the key to our success to perform programming structures. As well as the
3. Conquering easier the syntax of programming structures

(a) “Usually, I get bored in class and get tired because it has interesting activities. The programming course is generally so difficult for me.”
(b) “I do not like sitting each student on his own computer and the teacher simply delivers the lesson. I would like to know from the outset what learning activities and in which frameworks this lesson must be implemented.”
(c) “In our conversation with other peers, we discussed and addressed basic problems to understand programming structures. Although the typical school hour was usually against us”

4. Improving peer relations in Constructionism framework

(a) “Usually I did not use to study with my peers, but also 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 and not the others.”

The utilization of Second Life in this process contributed the entire learning process based on students’ answers, although that this cannot be guaranteed always at the first time when other instructional technologists or educators may want to implement something similar. Among these, some positive signs revealed further fundamental points of view that must be underlined:

(a) The increasing motivation, the degree of engagement and cooperation among students, as it was also proved consistent to Abbas (2010) study and
(b) The increasing level of students’ achievements in collaborative practice-based tasks.
In the last twenty years, students’ programming skills has been significantly enhanced by using problem solving learning methodologies. However, little are known about what can students gain from collaborative learning in a programming course (Lee, 2011; Hwang, 2012). The current study presented findings which were based on the utilization of Second Life and Scratch4SL. The positive climate created meaningful efforts that exemplified the main reasons of utilizing again both platforms.

The data essentially confirmed students’ positive attitudes for this learning process. Students showed a progress by utilizing their computational thinking skills, like problem-solving skills in well-defined constructivist-oriented instructional learning phases. Main activities were focused on the process of thinking and learning giving the importance to the cognitive development of students’ skills that delivered in collaborative problem solving activities. The learning approach based on Constructionism face technology as an instructional means for developing and strengthening students’ programming skills (Papert, 1981). The study results were also consistent with O’Donnell (2006) who has found that students’ motivation during a problem-based situation can be succeeded at higher levels, when all students interact with their peers verbally. As a result, they can contribute easily to cognitive structures toward learning objectives that can be delivered in group-based activities in order to be achieved the appropriate learning outcomes. Students using Scratch4SL performed better collaboratively within specific learning contexts and seemed regardless their cognitive backgrounds in programming with others. The 3D multi-user virtual world of Second Life has provided to students more cognitive challenges in team-based situations, enhancing their attention and motivation. The conventional course in a computer laboratory was limited and it was replaced by a course, in which the teacher guided the entire process via Second Life remotely. Finally, this study suggests that students had a significant progress by coordinating, collaborating with others when they seek to find alternative solutions in
problem-based situations, something that occurred as a significant extent for knowledge acquisition.

The contribution of the current study can be important due to some meaningful findings that are presented below:

(a) Exemplify students’ perceptions, experiences, and opinions, according to their answers during the learning process that is implemented in a secured virtual grid.
(b) Assist Computer Science teachers to enhance an organizational-pedagogical teaching plan in order to transfer students’ knowledge domain or to acquire the learning material based on their initiatives.
(c) Propose and present a constructivist-oriented instructional design framework based on Papert’s theory of Constructionism, in case of facilitating of students’ management responsibilities to gain knowledge.

Conclusion and educational implications

The utilization of Scratch4SL seemed to enhance students’ motivation and participation in programming courses. Programming courses that underpinned by Constructionism can give an added value to the educational development, according to students’ cognitive functions and the use of their computational thinking skills in collaborative problem solving situations. The teaching of programming structures include some important sections that can produce several challenges for students who want execute various applications for understanding better not only to abstract a problem, but also to learn how code correctly, and finally to propose a solution. Designing an introductory programming course delivered in blended instructional formats is an inherently difficult task, and as in this demand, it is generally needed careful thoughts for the use of the instructional design to all activities in order to be effective in achieving the stated learning objectives, efficient in the
use of instructor and student time, appealing to all users who involved (i.e., students and instructors).

The current study incorporated the use of more challenging activities in collaborative practice-based tasks via 3D visually-rich metaphors, which reflected on students’ attention and encouraged them to discuss how to identify or address programming structures, as well as promoting their self-growth. The study findings also revealed some interesting outcomes, which were attributed on students’ participation enhancing the communication and interaction with others, improving trust to others, and creating a greater awareness of their behaviors that lead to better learning outcomes. Beyond the difficulties of a programming activity at its’ first stage following a blended course delivery method, all users considered the schedule of this process as meaningful and valuable.

According to the above, some educational implications that should be noticed are the following:

- Students seemed that learned how to think critically and logically in order to execute programs using their personal computational thinking skills. They tried to solve theoretically complex programming tasks, taking into account a series of sub-tasks for the implementation of their ideas by utilizing a-/synchronous communication tools of Second Life in collaborative problem-solving situations.

- It was very difficult, if not impossible to construct all students together similar syntactic errors using Scratch4SL. They communicated a-/synchronously as teams to exchange ideas or opinions, something that was needed from each one via Scratch4SL and they had the chance to share the code to others for the construction of an algorithm optically in Second Life.
• Programming commands via Scratch4SL based on a set of graphical programming blocks and items should be “snapped together” with the respect to create a program that can become really insightful for students’ socio-cognitive development.

• The minimal cost of the assembled colorful programmable blocks from Scratch4SL to LSL via 3D “objects-to-think-with” has allowed students to participate in object-oriented tasks.

Some of the most significant concluding remarks of this case study are presented below:

(a) The diversity of students’ self-presentations in the 3D multi-user virtual world of Second Life. Each student had a private digital alter-ego (avatar) to feel, mimicking his/her real’s life and freely communicate with others and helped them further to study in problem-based tasks collaboratively.

(b) The free navigation in a secured virtual grid, in which students could participate in class and in online Second Life supported programming courses was very helpful for the implementation of constructivist-oriented scenarios. This treatment seemed to be crucial for the creation of more attractive conditions in a collaborative climate.

(c) The variety of multimedia sources provided several benefits regarding the use of students’ computational thinking skills.

(d) The 3D persistent game-like environment allowed the customization of virtual objects to co-develop and co-manipulate an innovative knowledge (object-oriented) programming domain that allowed students remotely to participate in meetings with their teammates, outside of the specific teaching hours. In these circumstances, students usually try to discover new areas of interest, and become practically "seekers of knowledge" and not mere recipients of teacher’s directions or regulations.

Lastly, a significant progress on students’ coordination to configure a learning activity with their peers and plan to solve problem-based activities was observed in collaborative
tasks. This depends on their technological literacy improvement, and their expertise to enhance e-skills by utilizing contemporary computer applications in 3D multi-user/game-like environments. Furthermore, noteworthy is the unique exploitation of a game-like platform created from Second Life and Scratch4SL. This combination enhanced the students’ socialization and collaboration.

**Limitations**

Throughout the learning process, the following limitations need to be referred:

- Students did not have the necessary expertise for the implementation of virtual meetings, and thus they found difficult to use Second Life at the beginning.
- In (supplementary) online courses, students found difficult to collaborate with other peers in common time, because sometimes their peers could not be attended.
- Second Life has prerequisite hardware requirements that cannot be served by some computers in the school’s laboratory or even from students’ personal computers. Due to the limited number in the sample, it cannot be generalized the study results and thence the positive impact on students’ motivation and participation cannot be guaranteed to future studies.
Future work

A future study should include the statistical analysis of quantitative data from an experimental and control group. A new comparative study with an experiential and a control group of high school students can prove some weaknesses or strong arguments of why Computer Science teachers can provide better the instructional affordances of the combination between Scratch4SL and Second Life for introductory programming courses.

References


at the Proceedings of the 41st ACM technical symposium on Computer science education.


