Je-LKS PEER REVIEWED PAPERS
Vol. 10, n.1, 2014
ISSN: 1826-6223 | eISSN: 1971-8829
Journal of e-Learning and Knowledge Society
The Italian e-Learning Association Journal

# THE DEVELOPMENT OF A VIRTUAL LEARNING PLATFORM FOR TEACHING CONCURRENT PROGRAMMING LANGUAGES IN SECONDARY EDUCATION: THE USE OF OPEN SIM AND SCRATCH4OS

**Nikolaos Pellas**

University of the Aegean, Department of Product & Systems Design Engineering - npellas@aegean.gr

**Keywords:** concurrent programming; Scratch4OS; Open Sim; Secondary education; 3D mind-trap puzzle

The present study introduces two well-known technologically-advanced environments in a computer laboratory of a Greek High School and highlights the growing challenges and affordances that students face after its utilization. For this purpose it was created a mind-trap puzzle game in a virtual learning platform emanated from the conjunction of the three-dimensional (3D) virtual world Open Sim and the two-dimensional (2D) Scratch4OS programming environment. Nowadays virtual worlds have been acknowledged as the most well-known candidate platforms for the implementation of collaborative learning scenarios in different disciplines of Higher education. However a research about their effect on students' engagement in Secondary education is still infrequent. The aim of this study is twofold: (a) to present the rationale of this conjunction and (b) to describe the students' experiences in order to learn concurrent and visual programming commands in this unique platform. The paper tabulates not

for citations:
**Pellas N.** (2014), *The development of a virtual learning platform for teaching concurrent programming languages in Secondary education: The use of Open Sim and Scratch4OS,* Journal of e-Learning and Knowledge Society, v.10, n.1, 129-143. ISSN: 1826-6223, e-ISSN:1971-8829

only a summary of some problems and challenges existed during the learning procedure, but also with some instructional implications that the increasing role of this unique platform can establish. Conclusive remarks based on students' opinions and experiences briefly encoded some interesting issues which also ascertained.

## 1 Introduction

Since the early '80s the cognitive value of learning programming commands has been widely accepted by many students of Secondary education. More specifically problem-solving activities through programming environments were used for fundamental concepts (such as serial sequencing or repeating procedures of commands etc.) which were otherwise difficult to be constructed with the conventional lecture-based teaching approach. Computer Science can assist students to produce or increase their cognitive thinking by utilizing algorithmic commands. Two main objectives that appeared as significant in these circumstances are related with (Ennis, 1994; Wu *et al.*, 2014): (a) the students' knowledge background for understanding the algorithmic thinking with central concern to recognize the added value of computational thinking, (b) the training session which can help students to conquer problem-solving skills based on algorithmic thinking activities with the use of various programming environments.

Indicative studies have already highlighted some problems in teaching programing courses such as: a) the students' misconceptions on how to understand and grasp the implementation of programs without the simultaneous execution of actions and commands using the Logo-like languages in Primary and Secondary education (Pea, 1986). This may result on students' performances probably when they try to enter in Higher education sectors and finally observed that they still fail to understand basic algorithmic commands (Guzdial & Soloway, 2002), b) the handling problems of the user's interface and the initial programming concepts which are produced (Blackwell, 2002), c) the maintenance cost of a learning environment for the coordination and organization of information that is emerging during the teaching process (Pennington *et al.*, 1995), and finally d) the lack of an instructional plan and the use of more traditional teaching methods have caused some difficulties, such as designing solutions mainly when students try to find solutions to a problem, and try to understand functions or program complex commands (Jenkins, 2002).

In order to stimulate the students' interests and avoid the learning barriers, educators should promote not only innovative methods which can enhance the teaching of programming commands, but also visually-rich environments that can exploit these methods. Over the last five years, Scratch has become a

potential platform for constructive learning activities. It is relative to Logo-like programming environments, and in this notion it is ideal for students in Primary and Secondary education (Pellas, Peroutseas & Kazanidis, 2013). Instructors with the Scratch programming environment can use problem-solving teaching approaches, and as Su *et al.* (2013) have mentioned this platform can become an important part of the entire instructional format. Feng and Chen (2013) have continually provided that learners problematizing scaffolds procedures via Scratch, and actively participate in learning activities via self-questions that improved their self-regulation and enhanced their learning achievements.

The internationally-changing socio-economic conditions today demand from every student to acquire all those skills that can make her/him actively to participate and able to respond on social demands and needs, both in cultural and cognitive level. In this post-modern period the technological, economic and social changes have provoked many areas of the compulsory education to implement different projects in flexible course delivery modes (online or blended) such as those of Computer Science which can play a pivotal role even in students' lifelong learning. Mainly scholars and educators (Hadjerrouit, 2009; Kordaki, 2013) have focused on the construction and transportation of students on platforms with 2D graphical user interface (GUI), but until now most of them have started to implement collaborative scenarios in 3D technologically advanced environments, like those of virtual worlds (VWs). Open Sim standalone server as an innovative open source VW has a potential game-based background due to its inherent 3D technological infrastructure which can be combined with the free plug-in Scratch4OS in order to be implemented various programming learning scenarios. Even more its inclusion can empower the learning dimension in a socio-constructive framework that meets the students' needs following a series of methodological and organizational principles.

In these circumstances, a rationale behind the utilization of Scratch in Computer Science courses is because of the students' response for the easy acquisition of basic algorithmic structures knowledge generally, and from lessons like the serial sequence, selection, debugging or repetition process of commands specifically. The logical conjunction between Open Sim and Scratch4OS emanated from: (a) the easily manageable and scalable infrastructure which can be built on authentic or at least realistic conditions by using one client server and internet connection, (b) the interconnection with external media server with the Open Sim client viewer can positively affect students' attitudes and enhance their learning expediencies, (c) the 3D interactive and adaptable persistent workflow where students and teachers can communicate and collaborate as avatars (digital alter-egos) to achieve common goals in a multi-user virtual

environment, (d) the students' groups can get in real-time and in a common place visually-rich feedback from their actions, something that can encourage them to explore and experiment with 3D tools or objects, and (e) finally, the lack of "invasion" risk to unwanted users during the teaching process at no additional charge.

Notwithstanding the radical restructure of 3D technologically-advanced environments affected and motivated the learning outcomes in different disciplines, there is still paucity of a research showing the anticipated students' experiences from the conjunction between SL and S4SL for programming courses. The purpose of this study was to familiarize students with various procedures in order to learn concurrent and visual programming languages with the most contemporary and technologically-advanced environments of Scratch (4OS) and Open Simulator (Open Sim) standalone server. Finally, this study provides via a close-ended questionnaire students' experiences and opinions. This process enables students to develop or enhance their algorithmic thinking in problems-solving activities held in a game-based virtual environment with a "mind trap puzzle" in order to strengthen their ICT-based literacy skills.

## 2 Review of the recent literature

### 2.1 The use and value of algorithmic thinking

Since early '90s when it was firstly appeared personal computers (PCs) and started their global technological growth based on the ever-evolving digital technologies (Internet, mobile devices, etc.), algorithmic thinking became an extremely important area of knowledge in education. Even though it highlights both the strength and effectiveness of the analytical manner of human thinking to solve problems, whilst on the case of the latter one provides the foundation for successful design and development of software products (Kolikant & Pollack, 2004). More important valuable purposes of teaching programming commands to students mainly are: (a) to reinforce the students' analytic and synthetic thinking; (b) to acquire methodological skills in order to solve simple or complex problem-based in simulation-based activities; (c) to develop algorithmic skills (problem analysis, algorithm design, structured thinking etc.); (d) to empower students' creativity and imagination.

By recognizing the timeless benefits of acquiring algorithmic thinking concepts, it is also imperative to utilize instructors and educators innovative platforms or environments that can introduce students to these basic concepts. Scratch is a 2D programming environment based on Squeak. Being a dynamic language, it allows users to code change colored programming commands

puzzles even during execution of the programs via drag and drop technique. It aims to teach programming concepts from children (K-12) to adolescents (13-17 years old) enabling them to create interactive games, videos or music.

The core of Scratch4OS is based on the 2D GUI of Scratch and its basic features focused on a graphical programming language combines the initial writing of concurrent or concurrent programming languages with colored puzzle-based blocks which are visible and playable through VWs (e.g. Open Sim). Thence, by snapping together graphical blocks for constructing, modifying and utilizing visual artifacts (virtual prototyping process), users can see the effects of their work in a common virtual place. The puzzle-based configuration promotes a direct 3D visual feedback to students at the same time where they interact with the VW and consequences of these actions can become clearly qualified. In this notion, Scratch4OS can motivate users' imagination, curiosity, control statements cooperation, competition and recognition of false both via the system's or the instructor's feedback.

The basic concern of the present study is the successful conceptual and practical-didactic understanding of programming commands that acquires from the same students the modeling of 3D visual prototypes via artifacts in order to present two interactive applications promoted by: a) a "stylus" for drawing programming objects, as the 3D space becoming a free-stylish "canvas" or b) a "programming" environment where students creating "objects-to-think with" by integrating behaviors and interactions with artifacts. The pedagogical value of Scratch4OS originated from its conjunction with Open Simi as a unique platform which can reinforce the student's visual-spatial and perceptual abilities in a common virtual space appeared as a prerequisite factor for an ideal response to the requirements of basic laboratory activities in future-driven teaching plans.

## 2.2 Open source virtual worlds

The 3D open source virtual worlds (OSVWs), and in particular Open Simulator (Open Sim) can offer multiple advantages compared to actual 3D game-oriented technologies (e.g. Online video games). Most importantly is the student engagement in case studies, the active or experiential learning procedures by harnessing with 3D visual simulations for teaching specific learning concepts or even the co-manipulation of different objects or primitives (e.g. construction of visual artifacts by avatars). A vast majority of notable studies (Cheng & Su, 2011; Kemp, Livingstone & Bloomfield, 2009) has disclosed the quintessence of functional capabilities and affordances of Open Sim in various disciplines of e-Education. Other preliminary findings (Berns, Gonzalez-Pablo & Camacho, 2013; Pellas, Peroutseas, & Kazanidis, 2013) have shown that

open source VWs can support various types of educational activities, escaping from the traditional and shifting to innovative frameworks in order to acquire students the knowledge with more constructive approaches.

## 2.3 Game-based learning in virtual environments

As it was previous researchers in the field of Information and communication Technologies (ICT) generally and Computer Science has already recognized the crucial role of algorithmic thinking in electronic environments mainly based on animations or game-based virtual environments. Alice is the most well-known programming environment and an object-based computer-oriented environment that offers a programming integrated development environment (IDE). In this environment users with the drag and drop technique tried to create computer animations by using 3D models and storytelling learning situations (Dann, Cooper & Pausch, 2006).

As regards the use of serious games Chesney, Chuah and Hoffman (2009) have created five virtual games where they studied some realistic economic phenomena and found the positive response of students' behaviors which were similar to those encountered in real situations. Another interesting aspect is that the game creation through the use of visual tools that VWs offer. Bilyeu et al (2007) have used two games (considering the solubility of sugar in the first and in second to investigate factors that cause erosion and deposition of sediment) with their findings to confirm that students not only have fun but also learned. However, as it was disclosed from the same researchers it is needed further research in order to be determined the overall educational benefits.

Konstantinou, Varlamis and Giannakopoulos (2009) have described the deployment of a course in Informatics through a 3D virtual environment focusing on the technical aspects with all the required preparations in a Greek High school to be included and the results were very positive from the utilization of Open Sim. Turkay and Tirthali (2010) have examined students' leadership skills developed by students from thirteen to eighteen years old by using a virtual environment. However, it should be highlighted a fundamental parameter of the complexity that every 3D environment governed which should be taken seriously into account from the instructor and therefore it is urgent to be organized a teaching plan before the beginning.

The problematic assumption that this study comes to answer that the aforementioned literature didn't really pretend to expose or solve students' problems in learning programming commands, basically in the Secondary education, but all of them described only learning circumstances. Moreover as it was previously shown there was no connection between the two environments or even any collaborative process to be implemented, as all of these were in blended

settings. From all the above it seemed that the utilization of open source VWs may have a positive effect as valuable candidate learning platforms.

Breakthrough in these aspects, VWs in the last seven years have emerged as candidate platforms for contemporary educational practices by providing edutainment (education and entertainment) approaches, socialization and collaborative learning scenarios for spatially (or not) distributed users. Juxtaposing to the above, remarkably little research was done not only for the investigation of the students' experiences in Computer Science but also in learning programming commands via Scratch4OS and Open Sim.

## 3 Method

### 3.1 The research intention and purpose of this study

The research efforts were focused on conducting a case study. Forty six (46) students of a Greek high school involved after the signed permission of their parents. The present research follows this of Abbas (2010) study who argued that small scale research projects on the integration of virtual reality (VR) training can offer an added value. In contrast, a large-scale project can quickly be devalued due to the rapid evolution of this technology. It was used a particular virtual place, where each student represented as avatars allowed to collaborate and communicate with other peers who enrolled in the same Computer Science course.

The main hypothesis was if the combination between these two innovative platforms (Open Sim and Scratch4OS) can assist students to better understand basic programming concepts or even strengthening their ICT-based literacy skills. The research questions raised are as follows:

> *a) Did the project contribute to students' motivation in order to better understand and easily learn basic algorithmic commands (concurrent and visual commands)?*
> *b) Can the utilization of SL and Scratch4OS be considered as a platform that can be used for students in order to resolve problem-based activities through collaboration processes and remove in this vein any intractable boundaries that substantially observed from prior "conventional" methods?*

### 3.2 The project setting

In order to use the same commands for all students it was decided the learning materials which should be designed in 3D geometric shapes and can be based on the algorithmic thinking. The learning axis of the chosen activity entitled as "*Designing and prototyping with the 3D visual programming language of Scracth4OS a mind trap puzzle*" in which students sought to learn concurrent programming and problem solving activities. Initially, a preliminary 2-hour

lesson for the student group dedicated to the description of the whole process that needed to be followed in acquaintance with the Open Sim, the disclosure of users' accounts and the students' guidance to install the necessary programs on their computers.

The process lasted two weeks with a total of six (6) teaching hours (~40 minutes) educational activities and took place in the second quarter of the school year 2012-2013. Before the beginning of the research process, it was deemed as appropriate to study students in a blended course delivery learning method with voluntary participation. The teaching of basic programming commands like those of concurrent programming that took place in the computer lab, but there were also some other supplementary online courses when it was necessary. Initially, even before the beginning of the pilot project, the following issues were investigated: (a) the access and navigation in Open Sim, where technical issues resolved in the laboratory, and (b) the connection in the VW from their home that should probably be established for students (mainly for online sessions).

In the experimental setup students needed to efficiently collaborate and work on the VW, where they must first of all develop eighteen (18) 3D "interlocked" puzzles in a shared virtual space. Students in Open Sim firstly constructed and placed the 3D assembled mind-trap pieces (i.e. modified visual primitives) in a co-manipulation task, where they must be able to communicate (verbally or text-based brainstorming) and exchange spatial information, such positioning a visual object, and programming every interlocked piece with concurrent programming commands in order to construct each side of the 3D cube nets. In this case it was conducted this experiment because students should understand the contents in a shared virtual space to understand how can construct a common spatial representation. The participants were asked to perform in two-dyads for a co-manipulation task. Additionally, several constraints were introduced to prevent the quadrats from being used as contextual references instead of the stable lateralized visual landmark: (i) all quadrats should have fairly dissimilar geometric shapes, but all of them have the same colour as their team has, and (ii) all quadrats were not lateralized, analogous to their right, left, front, or back sides are always dependent on each student's position that wants to start programming and placing each piece in the nets of the 3D puzzle.

The development, configuration and exploitation of 3D puzzle-based cube surely helped students to handle the functionality and "objects-to-think-with" assembled primitives in puzzle-solving situations as the "*design exploration of metaphors*". Similarly noteworthy, the trainees were also able to compromise the strength of designing puzzle-based games in a VW. In addition, puzzles denote a playable exploration by leveraging basic characteristics a "solving a 3D playful mind-trap puzzle" as an educational opportunity to learn students

basic algorithmic commands. Puzzle-assembled primitives comprised the several tools-smiths; permitting users to create interactively their workshops, control the problem-solving strategy and report visually the content of design puzzles. Puzzle-based procedures in this stage established as 3D metaphorical experiences rather than simple mechanisms of 2D design puzzle-based representations. Figure 1 depicts one example a virtual prototype 3D "assembled collage" in the front nets of the cube.

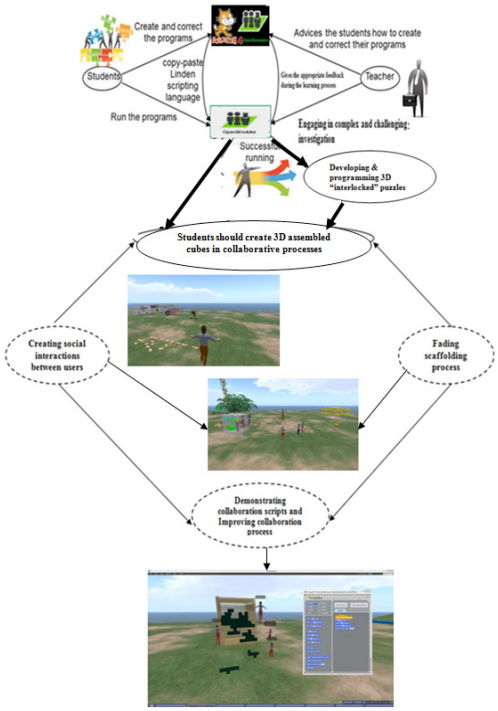


Fig. 1 - A theoretical framework for students' coordination in Open Sim

## 4 Results and Discussion

The aim of this project was to examine students' engagement and experiences. The research was based on an experimental design held in a Greek High School and criteria of a quantitative research method.

The construction of the main user-oriented instrumentation for this study aimed to measure their first-impact experiences. After completing various collaborative activities in programming courses, 46 correspondents answered 11 close-ended questions, according to a 5-point Likert scale (1=disagree to 5=strongly agree) questionnaire. The reliability of each factorial structure for the three thematic areas, the Cronbach's alpha (a) index was used and calculated

prices considered as satisfactory ($a_{Table1}$=0.76 and $a_{Table2}$=0.77).

Table 1
STUDENTS' OPINIONS ABOUT COLLABORATION AND COMMUNICATION IN OPEN SIM

| Questions | N | Min | Max | Mean | SD |
|---|---|---|---|---|---|
| 1.Were the design of geometric solids and the integration of concurrent programming commands in 3D primitives a pleasure collaborative process? | 46 | 3 | 5 | 4.03 | 1.52 |
| 2. Did you think that the unique interface between Scratch4OS and Open Sim responded adequately to the contemporary requirements in order to find out solutions in problem-based learning situations? | 46 | 3 | 5 | 4.23 | 1.72 |
| 3. Was the learning plan and structure of objects for welding larger items a motivating task? | 46 | 2 | 4 | 3.72 | 1.03 |
| 4. Did you find useful the a-/synchronous communication forms to perform in collaborative activities with other peers? | 46 | 2 | 4 | 4.45 | 1.14 |
| 5. Was the appropriate time feedback from your instructor during the execution of your activities useful? | 46 | 2 | 5 | 4.04 | 1.51 |
| 6. Was the sense of co-presence with others considered as more useful for the communication and organization of the learning experience? | 46 | 2 | 5 | 3.89 | 1.42 |

Table2
THE ASSESMENT OF THE COLLABORATIVE LEARNING PROCESS

| Questions | N | Min | Max | Mean | SD |
|---|---|---|---|---|---|
| 1. Were the available tools (scripts, textures, note cards or communication forms) from the system valuable for your engagement and contribution in the collaborative learning task? | 46 | 3 | 5 | 3.73 | 1.28 |
| 2. Were the allocations of activities finally helped you to co-construct and co-manipulate artifacts based on each member skills? | 46 | 2 | 4 | 3.42 | 1.43 |
| 3. Was the collaboration with your peers and instructor in the virtual environment successful to conduct with concurrent programming courses? | 46 | 2 | 4 | 3.25 | 1.44 |
| 4. Was the instructor really helpful to maintain the concentration of your thoughts in order to collaboratively resolve particular issues with others? | 46 | 3 | 4 | 3.17 | 1.74 |
| 5. Was there a certain comfort during the interaction with other participants in a learning path and was your personal motivation as a member of your team insightful? | 46 | 2 | 5 | 4.25 | 1.77 |

The data obtained through this process in order to answer the rsearch questions collected by using close-ended questions and Likert-scaled questionnaire from the implementation of problem-solving activities. Obviously, the descriptive methodological research approach seemed that strengthened the reliability and validity of research results and the validity of these conclusions. The above provisions have laudatory unveiled three significant results:

(a) The students' participation in concurrent programming course and problem-solving activities: Regarding the first research question, it was found that the 3D virtual reality (VR) infrastructure contributed to the teaching of programming, something that it was also previously confirmed from previous studies (Lim & Edirisinghe, 2007; Ritzema & Harris, 2008). These findings were reinforced from the students' positive responses about their experiences in Computer Science courses.

(b) The acquisition of knowledge and e-skills: The dynamic dimensions of students' engagement with the virtual platform and especially the development of constructive experiential or control e-skills, like collaboration, active participation, and group grounding skills, affected more positively the understanding of algorithmic concepts.

(c) Sociability, cooperativeness emotions and feelings among students: The data revealed another interesting finding about the collaborative processes and the cooperativeness between students and instructors from the analysis of recorded behaviors and the reactions of students during the course. In the study group, communication among other peers and instructor was pretty remarkable, and as well as the collaboration among helping each other.

After collecting the research data, the study results were extracted the following implications:

- The link between the two environments was an innovative proposal for monitoring students' participation as they created small groups easier. This helped them to be engaged in a 3D interactive environment and they learned easier algorithmic commands.
- Students who participated in the laboratory courses have showed that the incarnation of creating cyber entities (avatars) really helped them to be enrolled with others and communicated to create a new knowledge domain.
- Students from the first time were not as excited as the researchers expected because they were confused by 3D GUI. But, after a few minutes they gradually started to be exhorted from other classmates to partici-

pate and interact to achieve their learning goals.
- Some students gained greater confidence regarding their opportunity to collaborate with others to solve a problem-based situation in a multi-user virtual place, taking their own initiatives or even expressing queries and opening new horizons for discussions.

While there were certainly discerned some advantages, there were also significant disadvantages which must be taken seriously into account. These are:
- *The students' difficulties to use and navigate avatars in the entire virtual place before setting up their own groups*: Although, it was not considered as difficult to control avatars' movements. Students should firstly understand the concepts of motion or space perception and coordination with others to formulate in collaborative climate the desired purpose.
- *The coordination of students' activities*: During the learning procedure, especially when they implemented scenarios with the blended or (supplementary) online course delivery method students always needed an additional feedback from the instructor.
- *Students' improper behaviors and distractions in the VW*: An important point of view that observed was the fragmented behavior only in few times. It seemed that students were occupied sometimes with other things and not with the real learning goals. Many times there was also identified the phenomenon of cyber bullying between peers via asynchronous communications tools (IM or chat text).

## Conclusion

Based on the above analysis it can be established that the use of Open Sim and Scratch4OS must positively contributed to students' motivation and engagement to achieve the learning outcomes. In fact this can be taken for granted, because there are positive points included: (a) a significant upgrade of students' motivation, the degree of involvement and cooperation with other peers and (b) an important contribution of learning outcomes on gathering information and achieving learning goals.

The present study was developed to address three core problems in educational programming courses through Open Sim and Scratch4OS:
- Many programming languages were not designed to be useful for the production of no-complex code commands and thus all of them have introduced a mutual complexity. Scratch4OS can be used solely in order to teach educators and instructors programming commands without the complex semantics of production languages such as C# and Java.

Students can place objects from Scratch4OS's gallery into the VW of Open Sim in which they can co-manipulate, and co-configure by essential programming constructs. In this dimension students can use the drag and drop technique tiles that represent logical serial sequence commands and in Scratch4OS can be visually-rich presented via artifacts (objects-to-think-with).

- Scratch4OS is connected with Open Sim can help students to not remember all complex syntax codes. The above environment utilized as a unique learning platform to create "object-to-think-with" via programming commands, where students commonly being engaged in a two environments without confusing issues to be merged.
- Scratch4OS is not designed to appeal on specific sub-populations exposed to computer programming, such as Alice which was centered on female students of middle school age and encouraged users to involve only with storytelling learning techniques. Scratch4OS can be generally used by novice programmers (male or female) from Primary and Secondary education to Universities for learning basic algorithmic commands, without demanding from students' significant cognitive background on algorithmic thinking.

The future work may investigate interoperability issues that facilitate students' participation of several Greek High schools in this project, and the exploitation of different instructional formats according to other interdisciplinary learning disciplines.

# REFERENCES

Abbas, A. (2010), *Learning Dynamics and Control in a Virtual World*. In Proceedings of EDUCON'10 - IEEE Education Engineering (pp. 737-741). Spain.

Berns, A., Gonzalez-Pablo, A. & Camacho, D. (2013), *Game-like language in 3-D virtual environments*. Computers and Education, 60(1), 210-220.

Blackwell, A. (2002), *What is programming?* Paper presented at the 14th workshop of the Psychology of Programming Interest Group (pp.204-218). Brunel, Middlesex, UK.

Bilyeu, B., Liu, Ch., Franklin, T., & Chelberg, D. (2008), *Using Games Created in Second Life to Teach Middle School Science Content*. In Proceedings of SLCCEDU'08- Second Life in Education (pp. 1-5). USA.

Chesney, T., Chuah, S., & Hoffmann, R. (2009), *Virtual World Experimentation: An Exploratory Study*. Journal of Economics Behavior & Organization, 72(1), 618-635.

Chen, H-J. & Su, C-C. (2011), *Constructing a 3D virtual world for foreign language learning based on open source freeware*. Edutainment technologies. Educational

games and virtual reality/Augmented reality applications. (pp. 46-63). Lecture Notes in Computer Science, 6872.

Dann, W., Cooper, S. & Pausch, R. (2006), *Learning to Program with Alice*. USA: Prentice Hall.

Ennis D. L. (1994), *Computing, problem-solving instruction and programming instruction to increase the problem-solving ability of high school students*. Journal of Research on Computing in Education, 26(4), 489-496.

Hadjerrouit, S. (2009), *Didactics of ICT in secondary education: conceptual issues and practical perspectives*. Issues in Informing Science and Information Technology, 6, 153–178.

Feng, C-Y. & Chen, M.-P. (2013), *The effects of goal specificity and scaffolding on programming performance and self-regulation in game design*. British Journal of Educational Technology. Doi:10.1111/bjet.12022.

Guzdial, M. & Soloway, E. (2002), *Teaching the Nintendo generation to program*. Communication ACM, 45(4), 17–21.

Kemp, J., Livingstone, D. & Bloomfield, P. (2009), *SLOODLE: Connecting VLE tools with emergent teaching Practice in Second Life*. British Journal of Educational Technology, 40(3), 551-555.

Kolikant, Y. B.-D. & Pollack, S. (2004), *Establishing computer science professional norms among high-school students*. Computer Science Education, 14(1), 21-35.

Konstantinou, N., Varlamis, I. & Giannakoulopoulos, A. (2009), *Using 3D worlds in an educational network*. Workshop on Informatics in Education held in conjunction with the 13th Panhellenic Conference in Informatics, PCI 2009 (pp. 94-97).

Kordaki, M. (2013), *High school computing teachers' beliefs and practices: A case study*. Computers & Education, 68(1), 141-152.

Lim, J. & Edirisinghe, E. (2007), *Teaching Computer Science Using Second Life as a Learning Environment*. In Proceedings Ascilite Singapore (pp. 583-586). In R. W. Lucky, Automatic equalization for digital communication. Bell System Technology Journal, 44 (4), 547–588.

Jenkins, T. (2002), *On the difficulty of learning to program*. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (pp. 53-58). United Kingdom: Loughborough University.

Pea, D. (1986), *Language-independent conceptual "bugs" in the novice programming*, Journal of Educational Computing Research, 2(1), 25-36.

Pennington, N., Lee, A. & Rehder, B. (1995), *Cognitive activities and levels of abstraction in procedural and object-oriented design*. Human-Computer Interaction, 10, 171-226.

Pellas, N., Peroutseas, E. & Kazanidis, I. (2013), *Virtual communities of inquiry (VCoI) for learning basic algorithmic structures with Open Simulator & Scratch4(OS): A case study from the Secondary Education in Greece*. In K. Diamantaras, G. Evangelidis, Y. Manolopoulos, C. Georgiadis, P. Kefalas, D. Stamatis (Eds.), Balkan Conference in Informatics, BCI '13 (pp. 187-194). Thessaloniki, Greece: ACM Press.

Ritzema, T. & Harris, B. (2008), *The use of Second Life for Distance Education*. Journal of Computing Sciences in Colleges, 23(6), 110-116.

Su, A., Yang, S., Hwang, W. Chester S. J. Huang & Ming-Yu Tern, M-Y. (2013), *Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy*. British Journal of Educational Technology. Doi: 10.1111/bjet.12058.

Turkay, S., & Tirthali, D. (2010), *Youth Leadership Development in Virtual Worlds: A Case Study*. Procedia Social and Behavioral Science, 2, 3175-3179.

Wu, H-T., Hsu, P-C., Lee, C-Y., Wang, H-J., Sun, C-K. (2014), *The impact of supplementary hands-on practice on learning in introductory computer science course for freshmen Computers & Education*, 70(1), 1-8.