Conceptual Foundations for Interactive Programming Activities with the Conjunction of Scratch4OS and Open Sim

Pellas Nikolaos
University of the Aegean, Greece

INTRODUCTION

The widespread utilization of innovative learning environments for Computer Science in general and Programming courses in particular are largely well known. The promotion and best acquisition of basic algorithmic concepts for developing and strengthening the students' analytical-synthetic thinking skills in problem-solving situations is a demanding issue. By formulating conditions and causal relations for teaching basic concepts, scholars should try to create insightful learning scenarios for Computer science education. For a typical teaching intervention of students in algorithmic commands lessons, they always need to learn and visualize their actions in a suitable and stable environment.

During the last thirty years, several doubts about the added value of programming courses were highlighted, mainly in the cognitive research domain. However, teaching basic programming structures considered as an effective issue, which can be efficiently applied to other scientific fields, like Mathematics, Physics and Logic (Papert, 1980) with the acquisition of problem-solving skills to be the most important aspect of this process. Papert (1980) was the first theorist who argued that programming languages should provide to users an environment with an easily manageable interface surface (low-floor), where they can easily understand its' functions and they must define a broader framework for their active participation. This may allow them to exploit the highly interactive (high-ceiling) capabilities of multimedia content offered through more complex creations in a programming environment. Moreover, the technological infrastructure should support different and more complex types of projects (wide-walls) based on users' different interests and needs in order to share their works with others. Of course, it is very difficult to achieve the triptych low-floor/high-ceiling/wide-walls in a single environment.

Although, this intervention presents a variety of problems for the formulation and use of a programming language it should be included drafting the details of how programming commands can be visualized in a programming environment. Indicative results of previous studies have emphasized in some interesting features such as (Harms et al., 2013; Feng & Chen, 2014):

1. The students' misconceptions in understanding and implementing programming structures without the simultaneous execution of actions and commands by using the Logo language in Primary and Secondary education.
2. The problems of how to better handle students the graphical user interface (GUI) in order to implement programming concepts.
3. The maintenance of a learning environment for the coordination and organization of information that is being emerged during the teaching process designed in object-oriented settings.
4. The lack of an efficient instructional format with the use of more "traditional" (face-to-face/in-classroom) teaching methods which have caused some difficulties, such as the design to find students a solution for different problem required collaboration among participants in order to understand different commands and how these can be implemented in a programming environment.

The pedagogical utilization of innovative two-dimensional (2D) and open source three-dimensional (3D) technologically-advanced environments (see virtual worlds-VWs) whose main feature is the open code integration and extensive use of programming elements

DOI: 10.4018/978-1-4666-5888-2.ch469

is until today an interesting research field. In this contemporary era, the technological progress of learning environments has performed them not only as candidate platforms to observe students' exponentially increased maintenance, but also as an effort to interconnect 2D and 3D learning environments which have tailored to the needs and requirements of "Net Generation" (students as "digital immigrants"). By learning basic algorithmic commands, students can strengthen their technological literacy in well-identified constructive activities by involving them with cognitive elements via a multi-dimensional conceptual framework.

The technological infrastructure of educational environments like Scratch in recent years has overly helped students to overcome these problems. In this way younger users (10-15 years) are trying to develop a new culture to upgrade their technological literacy, which may give them incentives to acquire from them a more profound familiarity on how to program correctly a computer system and how they gained knowledge that can be applied in their daily life, something that is the basic requirement of computational thinking. Scratch also offers a 2D visual programming language that allows students to construct in an easy way interactive programs in block-based colorful puzzles.

However, the rapid dissemination of personal computers and the widespread dissemination of Internet connections have allowed the assimilation of 3D multi-user virtual worlds (VWs) as alternative e-learning platforms. VWs can evolve users in socially networked 3D environments can provide interactive 3D multi-user learning spaces or places in different educational disciplines. Meanwhile, a 3D multi-user VW facilitates the social interaction between users via synchronous or asynchronous forms of communication, the integration of multimedia applications or sources from the Web and exchange objects with material produced exclusively by cyber-entities (avatars) in a collaborative learning environment (Pellas & Kazanidis, 2013). A VW with open code infrastructure can be connected with other free plug-in modules to serve the requirements of online or blended (hybrid) courses in a practical-educational field, like this of Scratch4OS, in order to be structured a virtual learning environment based on students' needs and demands (Pellas, Peroutseas & Kazanidis, 2013).

Notwithstanding that a large academic literature body has disclosed the instructional affordances and benefits from the utilization of (social) multi-user VWs like Second Life, it is needed further studies for a new dimension. Open source VWs, as descendants of social VWs, correspond initially to lecture events, student engagement in constructive and innovative social-cognitive pedagogical frameworks, as many researchers and scholars has already noticed (Pellas, Peroutseas & Kazanidis, 2013; Weito, Hui, & Mingyuan, 2011).

Juxtaposing to the above, there is still lacking an investigation not only of the service level or interoperability issues on these 3D environments with an open-ended architecture combined with a free plug-in module of Scratch4OS, but also for the functionality and efficiency results that can replicate this unique virtual learning platform. Despite the widespread interest that is a limited in several case studies (Berns, et al. 2013; Pellas, 2014) with a small sample to explore innovative and alternative teaching methods or models, it is also crucial to be amplified the technological services, capabilities or benefits that open source VWs can endorse in conjunction with 2D free plug-in modules that may supply to the e-Education.

Based on the above problem, this study focuses on the theoretical foundations among the educational opportunities (examples) which can be adopted according to students' needs and demands (affordances) by connecting two innovative environments.

Open Simulator (Open Sim) and Scratch4OS were combined in order to elucidate (in theoretical terms) the added value in Programming courses and constructionism learning framework was proposed. It is belied that the successful conceptual understanding of programming commands can be acquired from beginners (novice) learners through the creation of 3D prototype modeling and artifacts featuring in two interactive activities with the utilization of Scratch4OS in Open Sim: a) a "stylus" for designing 3D objects in SL as a "canvas" to co-design students different scenarios and b) as a "programming" environment for the creation of artifacts ("objects-to-think-with") by integrating behaviors and interactive abilities in visual primitives (objects of the VW). The pedagogical value of Scratch4OS through these applications may be potentially proved in a VW, as it can enhance the visual-spatial and perceptual abilities of students in a multi-user VW, which is a prerequisite for an adequate response to the demands of basic laboratory operations by using modern learning environments for Computer Science courses.