



## An exploration of interrelationships among presence indicators of a community of inquiry in a 3D game-like environment for high school programming courses

Nikolaos Pellas

**To cite this article:** Nikolaos Pellas (2016): An exploration of interrelationships among presence indicators of a community of inquiry in a 3D game-like environment for high school programming courses, *Interactive Learning Environments*, DOI: [10.1080/10494820.2015.1127819](https://doi.org/10.1080/10494820.2015.1127819)

**To link to this article:** <http://dx.doi.org/10.1080/10494820.2015.1127819>



Published online: 25 Jan 2016.



Submit your article to this journal [↗](#)



Article views: 17



View related articles [↗](#)



View Crossmark data [↗](#)

# An exploration of interrelationships among presence indicators of a community of inquiry in a 3D game-like environment for high school programming courses

Nikolaos Pellas

Department of Product & Systems Design Engineering, University of the Aegean, Syros, Greece

## ABSTRACT

The combination of Open Sim and Scratch4OS can be a worthwhile innovation for introductory programming courses, using a Community of Inquiry (Col) model as a theoretical instructional design framework. This empirical study had a threefold purpose to present: (a) an instructional design framework for the beneficial formalization of a virtual community, by utilizing a Col model which consisted of 81 high school students and (b) the results of linear correlations to amplify the interrelationships among presence indicators (cognitive, social, and teaching) of a Col model to learn basic programming concepts via a 3D multi-user game-like environment underpinned by Papert's theory of constructionism. The findings indicated that social presence (communication and cohesiveness of a group) had not only a direct correlation with the cognitive presence (learning process for the construction of knowledge), but also had a positive association with teaching presence (organization, planning, and guidance of learning activities), reinforcing them as well.

## ARTICLE HISTORY

Received 31 March 2015  
Revised 28 November 2015  
Accepted 1 December 2015

## KEYWORDS

Constructionism; open Sim;  
secondary education;  
Scratch4OS; virtual  
community of inquiry

## Introduction

The increased interest of high school students' participation in online or blended (hybrid) instructional formats using three-dimensional (3D) multi-user virtual worlds (VWs) has been recognized by a growing body of literature (Dickey, 2005; Pellas, 2014b; Zhang, Marksbury, & Heim, 2010). A 3D multi-user VW integrates a digital-oriented environment that is online, persistent (i.e. a virtual environment that still exists, even when some users log out of it and the changes that have been made are permanent), interactive, and accessible for many (distributed or not) users concurrently. The interactivity and social contexts of modeling visually-rich artifacts in 3D multi-user VWs are the most crucial parameters for using this technology.

3D multi-user VWs can provide unique technological capabilities, which are suitable for the development of innovative, but yet nascent, virtual platforms with the respect to cover seriously the perquisite needs of various teaching and learning processes. The enhancement of learning acquires users' interactions in problem-based tasks. A well-established instructional design framework that can assist collaborative-interactive learning activities among users' interactions, following a constructivist orientation to learning is always required (Pellas, 2014b). This technology can influence students' participation in highly engaging instructional contexts through game-like activities better, rather than by using commercial digital games (Chen, Zap, & Dede, 2013).

Undoubtedly, the utilization of 3D multi-user VWs cannot solve all problems that accentuate students' engagement and participation in different educational levels. In this aspect, there is a majority

of significant technological and instructional perspectives that should be referred as important. Firstly, from a technological-functional perspective. 3D multi-user VWs can truly equip a wide field-class distribution through virtual “islands” (grids), in which instructors exploit for the needs of their courses by paying a specific fee. Nonetheless, like other candidate learning platforms, the following issues should be considered: (i) the maintenance financial cost, (ii) the (non-) autonomous (stand-alone) support of authorized users’ protection from others who should not attend with them in a course, and lastly, (iii) the allocation of operating digital resources to all users, so that start thinking about how to manage learning materials in collaborative activities. For this reason, academic researchers (Pellas, 2014b; Rico, Martinez-Mupoz, Alaman, Camacho, & Pulido, 2011) have focused their interest on the utilization of “open source” 3D multi-user VW, like Open Simulator (Open Sim). Although Dickey (2005) has already noted that the built-in tools of 3D multi-user VWs create a high-floor hurdle (i.e. “steep learning curve”) for high school students, the use of two-dimensional (2D) tools underpinned by constructivist-oriented learning approaches needed to overcome these barriers. Therefore, 2D tools should be used during students’ first time entry to low the complexity of a 3D graphical user interface (GUI) with the purpose of allowing them to be engaged easier and not discouraging their participation.

Despite the appropriateness of a constructivist-oriented instructional design framework for addressing the need of a low-floor (an easy way for someone to begin) construction tool to overcome these obstacles in 3D multi-user VWs, Scratch4OS (programmable in Open Sim) have been previously used as a programming tool for the implementation of high-ceiling/wide-walls (abilities for more complicated tasks offered as the time pass) learning tasks (Pellas, Peroutseas, & Kazanidis, 2013). Unlike Scratch (2D visual programming environment), Scratch for Open Sim (Scratch4OS) does not generate Java outputs, but it translates what someone drags and drops from the Scratch’s screen into Open Simulator Scripting Language that is the core of programming primitives or artifacts.

Secondly, from a pedagogical-instructional perspective. 3D multi-user VWs can assist the implementation of interactive-collaborative activities that involve and motivate all members of a community during the learning process, by enhancing as well their co-presence in a common environment. This can positively impact users’ relationships/interactions in a learning process (Pellas, 2014b). Zhang et al. (2010) have already pointed out that communication and social interactions among users in communities using 3D multi-user VWs, can extent students’ achievements. In fact, social interactions using different a/-synchronous communication tools would be the key to better learning outcomes. However, the configuration, coordination, deployment and management of a virtual community seems to be a difficult task without a well-established instructional approach using Second Life (Burgess et al., 2010). The successful development of an instructional process using 3D digital-oriented environments is directly related with students’ persistent engagement, in the direction of understanding the learning material that is given on the one side and on receiving the instructor’s support/feedback to be succeeded common objectives in well-designed instructional settings on the other. Educational activities should not have an impersonal nature, but an insightful one (Pellas & Peroutseas, 2015). This process should be designed in an efficient instructional format, with the view to expand students’ exploration in collaborative learning approaches that are implemented inside the community.

A noteworthy educational discipline that has been influenced positively through the use of 3D multi-user VWs, is Computer Science. A growing body of literature has already shown that 3D open source/multi-user VWs, like Open Sim (Pellas et al., 2013; Rico et al., 2011) or Second Life (Girvan, Tagney, & Savage, 2013; Pellas & Peroutseas, 2015) can provide the necessary multimedia tools for the creation of engaging 3D game-like environments in introductory programing courses focused on the development of students’ computational thinking skills. One of the most well-established learning theories that can be used for game-like activities in 3D multi-user VWs is Papert’s theory of Constructionism (Girvan et al., 2013; Pellas et al., 2013; Pellas, 2014b).

Previous studies (Apostolellis, Stewart, Frisina, & Kafura, 2014; Basawapatna, Repenning, Koh, & Nickerson, 2013; Ioannidou, Repenning, & Webb, 2009) have also emphasized to a great opportunity

in exploring the challenges of utilizing instructional design frameworks in digital game-based environments for assisting students to develop/enhance their personal computational thinking skills. Computational thinking is defined as a problem-solving thinking method that is necessary for anyone who uses computer science techniques to solve problems, design a system and comprehend human learning behaviors (Wing, 2006). It includes knowing how and when a human can use computing tools, knowing what steps someone needs to take to solve a problem, by logically organizing and analyzing data. However, a significant problem is that often students are not sufficiently able to develop a formal step-wise algorithm and propose a solution to a problem using their socio-cognitive skills (Chang, 2014). Indeed, high school students as novice programmers are engaged in digital-oriented environments without learning how to (Chang, 2014; Pellas, 2014b): (a) identify and understand correctly the problem statement; (b) separate the problem down to well-defined smaller pieces; and (c) design a step-by-step solution to solve each of the sub-tasks.

Related works (Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010; Pellas & Peroutseas, 2015) have recommended that higher order and problem-solving are very important skills to ensure the student's success in introductory programming courses. The same researchers have also claimed that programming courses need a well-organized instructional framework, especially when students are members of a digital community. Therefore, it is important to establish an instructional framework that can provide: (a) clear contexts for a learning process to describe the users' actions and responsibilities, based on constructivist-oriented pedagogical underpinnings, with regards to the establishing of their own actions in a 3D environment, in an attempt to recognize how to approach the learning material within a community, and (b) the development and implementation of well-established instructional contexts taking under consideration the social, cultural and cognitive dimensions of knowledge that can be gained by students in favor of acknowledging the benefits of participating and collaborating with their peers in a common learning environment.

The successful development of a learning process using digital-oriented environments is directly related to students' persistent engagement, in place of understanding the learning material that is given. Additionally, it is also essential to receive the instructor's support/feedback in order to accomplish common objectives in collaborative settings (Rovai, 2002). In other words, students' motivation is not only a very important factor for a successful instruction in computer-supported collaborative learning using programming concepts, but also for their engagement (Serrano-Cámara, Paredes-Velasco, Alcover, & Velazquez-Iturbide, 2014). Thence, a further investigation of the factors affecting the substantial presence and users' engagement in a community during their participation in collaborative activities to acquire knowledge is needed.

The community of inquiry (Col) model as an instructional means to organize a digital-oriented classroom allows knowledge construction as a result of teamwork among active participants in learning communities measured with the following presence indicators and indicated by (Garrison, Cleveland-Innes, & Fung, 2010): (a) the interaction of all users and/-with their instructor or with their peers (Cognitive presence – CP), (b) the implementation of constructivist-oriented scenarios that reflect an educational instrumentation to an online environment (Teaching presence – TP), and (c) the enhancement of users' socio-cognitive skills in a collaborative climate that implemented through online or blended instructional formats (Social presence – SP). This was the main reason that little research has been conducted to integrate teaching, social and cognitive presences as three components of an instructional design approach to contextualizing blended format based on the attempt to explore the effects of the three presences on users' interactions.

The utilization of the Col model was focused on different instructional formats via digitally-oriented environments, according to the demands of Computers Science courses in Higher education (Shea & Bidjerano, 2010) or in Secondary education (Angelaina & Jimoyiannis, 2011; Pellas et al., 2013). At the same time, other studies (Chang, 2014; Pellas, 2014b; Pellas & Peroutseas, 2015) have pointed out the need in exploring and creating an organizational-instructional framework based on pedagogical underpinnings that can enhance the degree of collaboration among users (students and instructor) so as to gain complex cognitive skills (syntactic knowledge,

problem-solving, reasoning and planning) more easily. These “skills” are fundamental for human computational thinking. However, less attention has been given to programming tasks based on an instructional framework for students to better organize or handle their management responsibilities, in order to bridge the gap between theory and practice. The theoretical foundation of the Col model can potentially be combined with the pedagogical underpinning of Constructionism for prototyping virtual objects and artifacts in 3D multi-user VWs (Pellas et al., 2013; Pellas, 2014a).

The research question (RQ) that emerged from this notion is as follows: *Is there any significant linear correlation (positive or negative) among social presence, cognitive presence and teaching presence indicators in a virtual community of inquiry that is held in a 3D multi-user game-like environment for high school introductory programming courses?*

This study presents correlation analyses focused on the relationships among the presence indicators of the Col model as an instructional means to regulate students’ activities which can reduce the complexity of their interactions in the direction of completing all learning tasks using a 3D multi-user game-like VW. The purpose of this study is to investigate the correlations among presence indicators (cognitive, social and teaching) in a Col, in which high school students construct an innovative “knowledge domain” for learning sequential programming concepts by co-manipulating, co-constructing and programming collaboratively a 3D game-like mind-trap puzzle. This game-like environment was created in Open Sim and high school students tried to learn how to think “computationally” and executed basic programming concepts collaboratively in constructivist-oriented instructional settings via Scratch4OS.

## Background

### *The exploitation of Col model*

Buraphadeja and Dawson (2008) have identified three models for the analysis of users’ interactions with the purpose of describing an educational process in online digital environments (i. e. i-the model of content analysis of Newman, Webb and Cochrane, ii-the model of Interaction Analysis of Gunawardena *et al.* and iii-the Col model of Garrison, Anderson and Archer). The latter one (the Col model) was regarded by the same authors as the most appropriate for analyzing students’ interactions in educational activities via online digital environments. The reasons are presented below:

- (a) The Col model as a theoretical background offers a valid organizational-instructional framework to analyze students’ interactions in communities (Arbaugh et al., 2008),
- (b) The previous models have not offered the analysis of users’ interactions in well-organized instructional contexts, in contrast to Garrison, Anderson, and Archer (2000) model,
- (c) the roles and purposes of a theoretical framework were not indicated in previous models, so that describe successfully teaching and learning activities, according to constructivist-oriented learning theories via (online) digital environments (Pellas, 2014a), and
- (d) little research has been conducted to integrate the Col presence indicators as components of an instructional approach delivered in blended synchronous learning experiences via digital learning environments (Szeto, 2015). The combination of Col model with constructivist-oriented learning theories for high school courses is still another research challenge.

The above literature suggests that the Col model can provide a more comprehensive view for educational researchers capable of amplifying users’ interactions among social, cognitive and teaching presences in constructivist-oriented instructional settings using digital learning environments.

### ***Educational potential of 3D multi-user VWs***

In spite of utilizing visual programming environments in three-dimensions (3D), such as Alice, Kodu and Agent Sheets, users did not need to engage with computer programming concepts from the outset in order to construct something meaningful. This situation initially creates a “steep learning curve” that cannot facilitate students’ engagement in problem-based tasks (Pellas, 2014b). This “curve” does not provide students with confidence to persevere due to a variety of programming activities, without a specific storytelling or context. As a result, they need to spend much more time to customize, and then learn how to create and program learning objects. Also, the above visual environments have not been designed to encourage the development of a more general understanding for using computational thinking skills (Howland & Good, 2015). Kalelioglu (2015) has pointed out the lack of operational sequence by invalid programming language syntax and the poor use of problem-solving strategies in digital-oriented environments, which can turn attempts at programming into a chaotic endeavor. This situation can affect negatively students’ engagement and participation in introductory programming courses.

The educational benefits and affordances of 3D multi-user VWs can lead to the improvement of programming courses. Some of the most indicative are the following (Dalgarno & Lee, 2010; Pellas, 2014b):

- (a) 3D multi-user VWs present a high level of accuracy and representational fidelity to develop realistic simulations. 2D environments have spatio-temporal drawbacks because of the digital window-based or text-based settings that cannot allow users to exchange opinions with their peers. Of course, users can communicate asynchronously (in different time) in a faceless digital environment. On the other side, 3D multi-user VWs provide the opportunity for (a/-) synchronous, real-time interaction to all users (students and instructor) in a common virtual environment. Thus, students can take feedback from their instructor or other teammates in real-time to solve a problem at the time that it is observed, even in online settings, and not only in class supported (face-to-face) settings,
- (b) a/-synchronous communication tools (VoIP, IM for verbal, or chat text and gestures for non-verbal) can assist the development of learning tasks at a distance. This is already well-known to all young students who attend in 3D multi-user role-playing games or environments (MMORPGs),
- (c) the (co-) manipulation of time-and-space sense, since users need to practice in a common virtual space to co-create 3D visual prototyping, easily design tangible/interactive artifacts and simulate artifacts’ behaviors or characteristics in real-time. The co-manipulation of different objects can offer the creation of artifacts using an instructional scaffolding process (i.e. conceptual cognitive models or constructs created by the same users helping them to understand something more intricate in better circumstances based on their teammates’ or teacher’s assistance).

The multiple options of collegiality and social aspects in designing and modeling virtual artifacts using a 3D multi-user environment have already focused the attention of many stakeholders on understanding their meaningful value. Incidentally, students do not seem totally isolated in front of a personal computer screen, but their actions acquire some beneficial features of “persistent co-presence.” This concept is intermittently envisioned in the contexts of a collaborative effort, according to users’ coexistence in virtual communities. The co-manipulation and modification of the 3D pre-constructed virtual place can still be continued by other peers, even if someone log out (Pellas, 2014b). In virtual prototyping processes, users are able to interact in team-based activities or combine multimedia 3D visual artifacts developed by basic geometric solids.

**Creating game-like conditions based on constructionist instructional design**

Piaget (1975) described Constructivism as a learning theory, in which each student can construct his/her own knowledge domain. Papert’s (1991) theory of Constructionism expanded Piaget’s theory in terms of helping students to produce their own constructions with their peers. Constructivism in this context gives more emphasis to the cognitive dimension of knowledge and Constructionism gives to the meaning of interaction between humans’ experiences and ideas. As it is already known, Constructionism as a theory suggests student’s exploration for knowledge acquisition in autonomous or collaborative settings (Ang, Zaphiris, & Wilson, 2011). Thus, Constructionism is based on constructivist learning theory with its main instructional context to be the construction of knowledge by the same student in practice-based tasks that can influence users’ interaction with the environment. In this way, students strive to control objects or re-order the cognitive representations to produce meaningful structures.

Latest studies’ results (Anderson & McLoughlin, 2007; Pellas, 2014b) have also shown that the utilization of environments rely on game-like learning situations can extend more easily the development of students’ socio-cognitive skills in collaborative settings. A 3D “game-like” environment can be defined as a simulation-based learning environment that includes a prototype game with playable characteristics inspired by serious games or game-based environments (Li, Cheng, & Liu, 2013).

Constructionism as a theoretical foundation for the development of an instructional design framework has received meaningful correspondence among the view that individuals have on thinking to produce something as a proof of learning in well-designed settings. Table 1 briefly summarizes how learning conceptions based on Constructionism can influence instruction via a 3D game-like environment that created in Open Sim.

The aim of creating Table 1 was to provide an instructional context for students’ coordination to configure a learning activity with their peers in collaborative problem-based activities. The alignment of the above features can extent students’ technological literacy, and their expertise to enhance e-skills by utilizing contemporary computer applications in a 3D multi-user/game-like environment.

**Utilizing Scratch for Open Sim (Scratch4OS) as a programming tool for low-threshold and high ceiling activities**

A series of studies (Chang, 2014; Kalelioglu, 2015; Pellas, 2014b; Voogt, Fisser, Good, Mishra, & Yadav, 2015) has already mentioned the following barriers in introductory programing courses: (a) the

**Table 1.** How the view of knowledge influence the view of instruction and 3D game-like characteristics.

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



students' misconceptions in understanding the importance of gaining knowledge from programming courses to their daily lives, according to the rigorous syntax of text-based computer languages; (b) students' management responsibilities in a GUI without their visual embodiment representations to be appeared, during the execution of programming structures can eliminate the sense of co-presence (psychological sense of each user to be there with others in a common place) and this can result in negative emotions; and (c) lack of students' actions or practice-based consequences in a "faceless" GUI creates a sense of disengagement and cannot foster their participation in collaborative tasks.

According to Kalelioglu and Gülbahar (2014), learn programming in the 2D digital environment of Scratch, did not cause any significant differences in the problem-solving skills of the primary school students, while a slight improvement to the students' self-confidence using their problem-solving abilities observed. A research conducted by Zhang, Liu, Ordóñez de Pablos, and She (2014) has proven that by using Storytelling Alice, students learned fundamental programming concepts in the context of creating animated movies and video games easier. Denner, Werner, and Ortiz (2012) have provided the opinion that game construction can support the learning process insightfully, including both design and programming activities. Other research studies (Carbonaro et al., 2010; Howland & Good, 2015) have identified the effect of teaching programming structures to students, such as higher order thinking, critical thinking, creative thinking, and problem solving, while others (Pellas, 2014b; Voogt et al., 2015) have stressed as an interesting part the process of learning collaboratively how to think "computationally" in well-defined instructional settings.

Scratch4OS has low computational hardware requirements; however the combination with Open Sim encapsulates it with high-impact capabilities and a wide range of applications that can empower students' "technology literacy." It is designed as a supplementary programming tool accessible (low-floor) to novice programmers without previous experience. At the same time, Scratch4OS produces powerfully complex-expressive (high-ceiling) programming language providing multiple opportunities for users to construct, explore and investigate complex programming commands in 3D interactive "thinking" (wide wall) objects or artifacts (Pellas, 2014b). The core of Scratch4OS is based on the 2D GUI of Scratch, and its basic characteristics are focused on a graphical programming language that yields to the initial writing of concurrent or sequence programming commands with colored puzzle-based blocks to be visible and playable by "copy-paste" the code from Scratch4OS to Open Sim objects.

A rationale behind the utilization of Scratch4OS combined with Open Sim for programming courses, like learning how to use the sequential, selection, debugging or repetition programming commands can provide the following benefits (Pellas et al., 2013): (a) a persistent environment for programming, in which users can be engaged in authentic or at least realistic conditions, by using one client server and a fast internet connection; (b) the easy connection of an external media server with Open Sim client viewer can positively affect students' attitudes to enhance experiential learning processes in programming artifacts via Scratch4OS; (c) the 3D interactive and adaptable environment to course needs, in which students and instructors can collaborate as avatars (digital alter-egos) in favor of achieving learning goals by sketching or by inserting behaviors via Scratch4OS to Open Sim objects; (d) students can get visual-feedback of their actions in real-time. This may encourage them to explore and experiment with 3D built-in tools or objects in a common environment; (e) while the use of Scratch (version 1.4.) reduces the "steep learning curve" in programming courses by empowering novice programmers' computational thinking skills to master programmatic constructs, students can focus on how logically can solve a problems before syntax the appropriate algorithm. Scratch4OS allow students to predict if their actions are "syntactically" and observe the consequences of their actions in real time via Open Sim and finally (f) the lack of "invasion" risk by unauthorized users during the teaching process without additional charges can be achieved.



## Method

### *Setting the rationale for conducting this study*

An experimental correlational research methodology at the end of all learning activities was used. This empirical study followed two significant statements. The first was based on previous studies (Dalgarno & Lee, 2010; Pellas et al., 2013) declaration which was common, about a need to conduct further empirical studies concerning how the potential characteristics and features of 3D multi-user VWs should be exploited pedagogically. The second was Garrison et al. (2010) statement. The authors have stated to the point that “the three presences are interconnected and influenced each other in the hypothesized manner” (p. 35). Hence, by understanding the interrelationships of a Col model among SP, TP and CP indicators, instructional technologists and educators can enrich theoretical insights in blended education practices, regardless the digital environment that is utilized (i.e. 2D or 3D). Garrison et al. (2010) have indicated that the Col model presence indicators did not have equal roles. In a Weinert, Bannert, Zumbach, Hoppe, and Malzahn (2011) study, TP and CP have equal roles in shaping the educational experience, a positive sign of SP has influenced students’ perceptions during the execution of learning tasks in collaborative settings.

This study’s hypothesis is that either correlation studies that utilized 3D multi-user VWs or studies focused on well-structured instructional frameworks. The causal relations can make significant contributions to the field of instructional design technology.

### *Participants*

Eighty-one (81) students (43 male and 38 female) from a public high school in Greece participated voluntarily and shared their experiences. The sample had an average age of 14.53 (with an S.D. = 1.75) and after the signed permission of their parents they attended to this study. None of the students had previous experience with Open Sim, but in the past, all of them had adequately utilized Scratch (version 1.4) as a programming authoring tool. The effort of setting up this project with a probability sampling was made in mid-September to early January of 2014–2015.

### *Instrument and data collection tool*

The questions for this study were adopted by the Col instrument that encompassed 34 close-ended and Likert-scale (from 1 = strongly disagree to 5 = strongly agree) items: 13 items for teaching presence (TP), 9 items for social presence (SP), and 12 items for cognitive presence (SP) (Swan et al., 2008). This instrument has been tested to establish a reliable measurement for three presences (Arbaugh et al., 2008). Also, it was validated using a multi-institutional data set (Swan et al., 2008). This questionnaire was the most important tool to measure how the Col model presence indicators can influence high school students’ interactions in a virtual Col.

Swan et al. (2008) instrument was used after finishing the learning process. This questionnaire was chosen to be the main instrument, because (Akyol, Vaughan, & Garrison, 2011; Garrison et al., 2000): (a) it provides a construct validity to its questions and (b) it is intended as a meaningful instrument to record the answers of those users who participated in blended instructional formats. However, its sub-questions should have the appropriate construct. For example, in the same questionnaire the following issues were observed: (a) CP has the sub-question “I have developed solutions to course problems that can be applied in practice.” In order to establish a question more reliable to this study’s demands, this was adopted as follows: “I had the opportunity to think critically and develop solutions in problem-based programming situations that can be applied in practice” (b) SP has the sub-question “Online discussions help me to develop a sense of collaboration” and it was adopted as follows: “Online discussions via Open Sim combined with Scratch4OS helped me to develop a sense of collaboration with my team in different instructional format (in-class and online)” and lastly (c) TP has the

sub-question “The instructor helped to keep course participants engaged and participated in productive dialogue” and it was necessary to be adopted as follows: “The instructor helped to keep all team members engaged in productive dialogue in Open Sim and provided adequate feedback in real time to assist their participation.”

Data were collected by sending email to all participants. The author of this study contacted the main Computer science instructor asking the permission to conduct directly with students. Based to their approval, the author posted the recruiting letter and a link with the online survey on a message board. The instructor also encouraged students to participate in this study. After students filled out the online consent form, they were directed tried to complete the online survey.

All statistical analyses of the data were performed using SPSS (ver. 21). The respective presence indicators were in the interest of this present study, as bivariate and partial correlation analyses were chosen, for exploring (a) violation of normality assumption, and (b) possible multi-collinearity problems which refuted high correlations among the three presences. Also, to check the reliability of each factorial structure for the three thematic areas, Cronbach’s alpha ( $\alpha$ ) index was used and calculated rates of presence indicators. All rates considered in satisfactory levels with  $\alpha > 0.07$   $\alpha_{SP} = 0.85$ ,  $\alpha_{CP} = 0.82$  and  $\alpha_{TP} = 0.84$ . A series of statistical analyses, such as Sample Kolmogorov–Smirnov test and linear correlation coefficients of Pearson were also followed in order to answer the RQ.

### **Treatment**

The study took 6 weeks and 18 hours of exercise in blended settings (8 hours in the schools’ computer laboratories and another 10 in supplementary online courses when the instructor decided it). The teaching of concurrent programming structures took place in the computer lab (2 teaching hours), but students also participated in online supplementary sessions when it was necessary (2 teaching hours in each meeting). Before the beginning of the research process, it was deemed as appropriate to study all participants voluntarily in blended instructional format. The teaching of basic programming commands, like those of sequential programming commands took place in a computer lab and online supplementary courses, if it was necessary. Initially, even before the beginning of this study, the following issues were investigated: (a) users’ access and navigation in Open Sim using real names and surnames and (b) the connection to the 3D multi-user VW from their home for online courses should probably be established from all students.

During the learning process the researcher (author) and the main Computer Science instructor were facilitators of the entire process. The instructor had a role to assist not only their first attempt to execute the different programming structures, but also to evaluate optically and predict artifacts’ behavior, in the interest of correcting the execution of students’ actions in specific time schedule that provided vice versa. The Computer science instructor and the researcher had the responsibility to attend all courses (face-to-face and supplementary online) and to assist students’ efforts. The researcher was setting the main problem statement and purpose of students’ participation in a 3D game-like environment with the respect to learn how to think “computationally” in collaborative instructional settings. No research approach was found to provide instructional affordances for high school students in programming courses. Thence, the problematic assumption of this study was based on: (a) how potentially can students use their personal computational thinking skills, which are used in concurrent programming languages in order to engage and participate positively/negatively in virtual communities of inquiry, (b) how Scratch4OS can be used for the implementation of collaborative problem-based tasks to overlap the “steep learning curve” using a 3D multi-users/game-like environment and (c) how a blended instructional format based on Constructionism enhances the generated 3D visual modeling (virtual prototyping process). This process can lead to a better understanding of learning activities in Open Sim.

This treatment had a twofold purpose: (a) firstly to familiarize students in basic programming structures, such as sequence, conditional statements (if/else) and loops (while, for) through the creation of artifacts in a 3D multi-user game-like environment; and (b) secondly to recognize an easier

way of enhancing students' engagement in programming courses collaboratively, by programming structures via Scratch4OS and thereafter by combining the code to predict or move 3D artifacts (smaller primitives) to assembling a 3D mind-trap puzzle cube. This process may enable students to develop and produce alternative solutions in problem-based learning conditions through a 3D multi-user game-like environment, supposing to strengthen further their technological literacy. Students in face-to-face settings on their side were sitting in pairs to side-by-side desktop computers. All of them needed to download Scratch4OS and Open Sim client viewers for their in-world entrance. It was decided to use each one of these two environments respectively, i.e. one day to get used Scratch4OS from one student and on the other day to get used Open Sim client viewer by changing his/her role with another student respectively. After that, each team with four students was scheduled to be inside Open Sim All students also allocated to Open Sim randomly in pairs (real-time experiment) without control group to be used. This procedure was more helpful when it is delivered in online instructional format, as seemed to be appropriate for each student to use both environments and participate at a distance with their peers.

Weekly notes were sent to each team via email to recapitulate each time and provide important upcoming information regarding contribution of each student using a 3D multi-user game-like environment. Despite the prominent role that teaching presence might play, this study expected that instructional effects of the three Col presence indicators could contribute to attainment of the learning outcomes in a balanced, interrelated with blended synchronous learning and teaching process. According to Li and Kraemer (2014), the social effects of pair programming serve to mitigate the phenomenon of "bounded rationality" with the result of (novice) programmers' behaviors seemed that can be aligned with good practice-based learning tasks. For this reason, each presence indicator was proposed to be responsible for a specific instructional purpose involving various types of learning activities with intended instructional effects (see Table 2).

Although a class-supported instructional course delivery method is the most common in use by high school students (regardless their cognitive background) to learn a programming language by memorizing the commands (reserved words) or to syntax a simple program, there are some other programming languages (i.e. Logo or Python), which require from users to operate correctly or grammatically exactly the rules in visually-rich settings to affect knowledge acquisition. One discriminative characteristic that educators often recognize using 2D or 3D technologically-advanced environments is that many high school students are not prepared to think "computationally" (Zhong, Wang, Chen, & Li, 2015).

However, they are usually exposed to numeric computation without being also involved in well-designed instructional settings to acknowledge the management responsibilities in their actions. In other words, students, by cultivating their metacognitive knowledge, could develop or enhance their computational thinking skills to create programs based on a logical way of thinking and to propose a solution. The Col model presence indicators as instructional constructs comprised of three instructional components which were postulated to guide the content presented or demonstrated inside the peer communication atmosphere and inform the discourse that would facilitate students' participation in the blended synchronous mode. The aim of utilizing this model was to determine if the creation of 3D artifacts as game design prototypes by using Scratch4OS programming tool in Open Sim following Papert's theory of Constructionism, can assist students to use their personal computational thinking skills.

The appropriateness of instructional conditions through a Col, in which students were engaged need to be investigated. For this attempt, it was imperative not to measure what students have learned more easily or more effectively through a comparative study, during their attendance in introductory programming courses. The purpose was to measure how they can be motivated and engaged in communities of inquiry and programming tasks via a 3D game-like environment. This process can help instructional technologists and Computer Science instructors to recognize how well students can participate in blended instructional format in a/synchronous modes, by utilizing Scratch4Os and Open Sim and use them meaningfully to develop/enhance their computational thinking skills.

**Table 2.** The instructional effects of the three presence constructs in blended course delivery method.

Coding scheme of a Col model	Presence indicators	Instructional phases	Collaborative learning process in constructivist-oriented settings	In-class (face-to-face) and online learning modes
<i>Teaching presence</i>	Instructional management Building understanding Direct instruction	<ul style="list-style-type: none"> <li>– Assign students' roles with the learning material in Open Sim.</li> <li>– Introduction to the learning activity and his/her facilitation role was announced.</li> <li>– Instructors' final decision for accomplishing in-world problem-based learning tasks each team collaboratively in real time</li> </ul>	<ul style="list-style-type: none"> <li>– Team discussions of exercises and practices by constructing and predicting artifacts' behaviors to complete their tasks required.</li> <li>– Team-based problem-solving exercise in each team are provided to complete "fading scaffolding" tasks in which the main instructor tried to be abandoned and let students free to execute and correct as possible as they can their programming structures.</li> <li>– Final team-based reports/presentations are announced</li> </ul>	<ul style="list-style-type: none"> <li>– Open discussions to share or propose solutions to a problematic learning situation.</li> <li>– Project formative assessment by firstly main instructor and secondly by re-construction of the codes from the team (real-time feedback was provided).</li> <li>– The main instructor should conducted with team members regardless the different modes in which he/she attended (online or face-to-face).</li> </ul>
<i>Social presence</i>	Emotional expression Open communication Group cohesion	<ul style="list-style-type: none"> <li>– Team-based conversations and decisions that should be provided in the learning task.</li> <li>– Introduction to the main collaborative course plan.</li> <li>– Formalization of each team instructions (students had different socio-cognitive background).</li> </ul>		
<i>Cognitive presence</i>	Triggering events Exploration Integration Resolution	<ul style="list-style-type: none"> <li>– Decision processes using students' computational thinking and their higher order skills as the time (30 minutes to finish each programming structure, that is, sequence or conditional statements) to be finished exercises that pressed them. They should use different programming structures to complete the task. Firstly, they should use sequence of structures and then try to combine artifacts using Scratch4OS with programming conditional statements or loops.</li> <li>– Back to other team members and initial results presentation should be presented to receive instructor's feedback</li> <li>– Teams' report presentation to the entire virtual class</li> </ul>		

### Experimental setup

In the experimental setup, students (participants) was needed to collaborate and study efficiently in a 3D multi-user game-like environment, in which they must first of all develop eighteen (18) 3D "interlocked" puzzles in a shared virtual environment. All participants should firstly construct and

place the 3D “assembled” mind-trap pieces (i.e. modified visual primitives) in a co-manipulation task. They were able to communicate (verbally or via text) and exchange ideas or information with their peers, such as positioning a visual object, and programming every “interlocked” piece sequentially, in order to construct each side of the 3D mind-trap cube.

Each student’s starting viewpoint was different from his/her partner for each new 3D prototype. They freely modified different viewpoints during the task, something that helped them to avoid complex mental rotations and to co-manipulate easier the virtual objects and artefacts. However, additional feedback was not given about the other’s viewpoint. Furthermore, 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 need to have the same color as their team and (ii) all quadrats were not lateralized, regarding to their right, left, front, or back sides being always dependent on the position that each student wants to start programming and place each piece in the nets of the 3D mind-trap puzzle.

The development, configuration and exploitation of a 3D puzzle-based cube surely helped students to handle the functionality and “objects-to-think-with” assembled primitives in puzzle-solving situations as “design exploration metaphors.” Similarly noteworthy, students were also able to compromise the strength of designing puzzle-based games in Open Sim. In addition, puzzles denote a playable exploration by leveraging basic characteristics of “solving a 3D playful mind trap puzzle” as an educational opportunity to learn sequential programming constructs. Puzzle-assembled primitives comprised by the several tool-smiths, permitting users to interactively create the appropriate code via Scratch4OS, control programming structures by utilising problem-solving strategies and visually report the content of design puzzles in Open Sim. Puzzle-based processes in this stage were established as 3D metaphorical experiences rather than simple mechanisms of 2D design puzzle-based representations. [Figure 1](#) depicts both a virtual prototype 3D “assembled collage” in the front nets of the 3D cube and the instructional principles that were based on Col model presence indicators.

This 3D game-like environment should facilitate users’ collaboration. Collaboration in problem-solving tasks promoted according to students’ reflections in interactive activities using various communication tools. Open Sim enables a-/synchronous collaboration among avatars in a common virtual grid (with voice calls or gestures) to edit the same primitive and share the same code while programming each primitive via Scratch4OS (Pellas, 2014b).

To become successful a learning process, all teams needed to accomplish each side of the cube so that predict the effect of their programming structures in Open Sim based on conditional state-ments (if/else) as fast as they could do, due to the time pressure. A collection of awards was also included in this 3D multi-user game-like to motivate and engage students in specific time schedule. Beyond the basic winner prize, there were also a set of other awards determined the final winner team in case of a tie. The awards were the following: (i) “write the code correctly”: it was given to the group that has correctly written all programming codes, (ii) “degradation of the problem in right settings”: it was given to the group that has discovered the largest number of questions to deconstruction and comprehension of the main problem in small pieces, even if finally they have not answered them correctly and (iii) “best short time spent to assemble and program interactive arte-facts” *award*: it was given to the group that assembled and programmed correctly almost all the artefacts without destroying the side of other teams.

## Results and discussion

The Pearson’s correlation coefficient considered as the most well-established toward to test the RQ about the data regularity. More specifically, the sample of 81 students was regarded as essential to provide all information that this study needed. Consequently, the descriptive data gathered through correlations of the test normality Kolmogorov–Smirnov ([Table 3](#)) and it was observed that the average (Mean- $M$ ) of the three presence indicators were generally at high levels ( $M_{SP} = 4.22$ , S.D. =

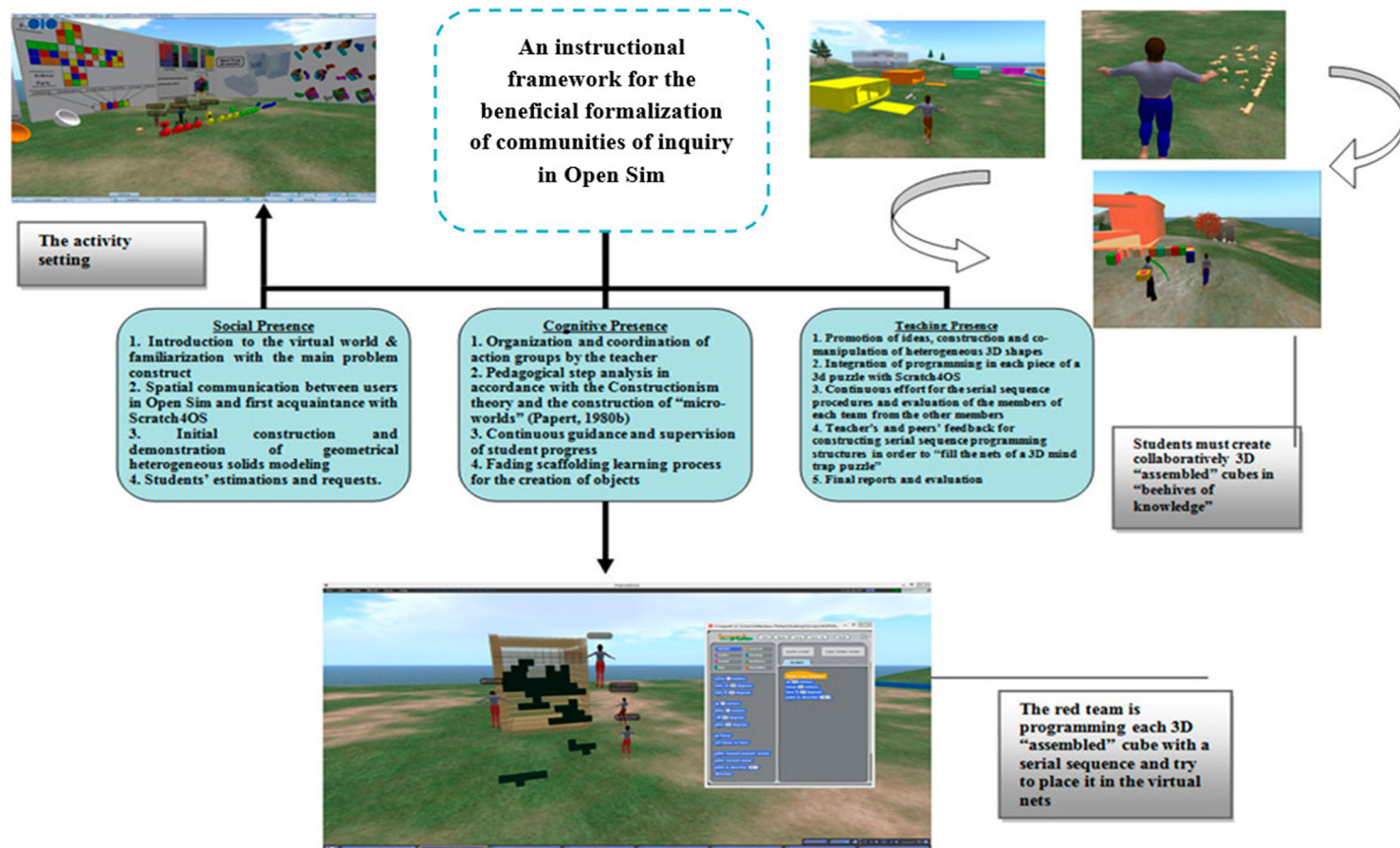


Figure 1. Activities through the conjunction between Open Sim and Scratch4OS.



0.84;  $M_{CP} = 3.94$ , S.D. = 0.56;  $M_{TP} = 3.88$ , S.D. = 0.87). However, SP had a mean higher than the other two.

The statistical significance ( $p$ -value) for all variables was higher than .05 ( $p_{SP} = .412$ ,  $p_{CP} = .284$  and  $p_{TP} = .147$ ). Consequently, there are indications that the measurements of this study's variables followed the normal distribution. Pearson's analysis was used to test the linear regression (Table 3).

Based on Table 4, a statistically significant positive linear correlation between the indicators of the SP and CP was noticed. A coefficient of linear correlation ( $r = .612$ ) suggested that by increasing social and emotional competence of student engagement in collaborative problem-based activities, the index of CP simultaneously increased. Furthermore, a linear correlation between SP and TP ( $r = .548$ ) was also appeared. This association brought to the front the feeling of the user's socialization and their initiatives that can be provided to each user through an organizational-instructional design framework to the extent of facilitating programming knowledge acquisition. After the completion of activities in the 3D multi-user VW, it is ascertained that in this study, SP had a positive linear correlation with indices of the CP and the TP. In conclusion, there was a positive linear correlation between the two indices of Col by utilizing a 3D game-like environment.

An additional positive aspect also resulted and consisted with Swan et al. (2008) study. Findings of this study confirmed the following: (a) students had the opportunity to co-create, experiment with others and program complex 3D objects using Open Sim and Scratch4OS. Both environments are intended to be appropriate for practice in the terms of visual effects for the implementation of sequential programming constructs to enhance students' technological literacy and met the immediate demands of instruction for introductory programming courses (CP); (b) several tasks assigned for each team, in which all students were involved in collaborative problem-based situations (SP); and finally (c) the facilitating role of the Computer Science instructor during the learning process assisted each student to have a specific role in a collaborative climate with mutual assistance, while supporting them all (TP). Students were engaged and seemed to understand better how to construct their own knowledge domain in well-designed instructional contexts based on a Col model. Moreover, it was revealed that students' participation and engagement delivered in-class and outside of it in

**Table 3.** Descriptive results and normality test.

		SP	CP	TP
Normal parameters	Mean	4.22	3.94	3.88
	Standard deviation	.084	.056	.087
Most	Absolute	.111	.124	.156
	Positive	.046	.111	.331
Extreme	Negative	-.251	-.067	-.264
	Differences			
Kolmogorov-Smirnov Z		0.788	.644	.814
Asymp. Sig. (2-tailed) ( $p$ -value)		.412*	.284*	.147*

\*Significant at  $p < .05$ .

**Table 4.** Pearson's linear correlations.

		SP	CP	TP
SP	$r$	1	.612**	.548*
	$p$		.002	.053
	$r^2$		.623	.089
CP	$r$	.612**	1	.178
	$p$	.002		.083
	$r^2$	.623		.078
TP	$r$	.548*	.178	1
	$p$	.053	.083	
	$r^2$	.089	.078	

\*Significant at  $p < .05$ .

\*\*Significant at  $p < .01$ .

online settings were increased (SP). Each one was able to better understand learning materials offered through specific activities. As a result, they started to become “seekers of knowledge,” rather than passive recipient of the instructor’s commands (CP).

Kozan and Richardson (2014) have stressed that CP significantly influenced the relationship of SP and CP, whereby CP seemed to play a dominant role among the three presence indicators. In contrast, SP was indicated to have a positive correlation with the other two presence indicators based on this study results. This brings to the front issues of sociality and collaboration as more significant in a 3D multi-user environment, in which students’ engagement in collaborative settings influenced their cognitive development and the enhancement of their socio-cognitive thinking skills. Despite the different views that can be indicated, it should be mentioned that all Col presence indicators are considered as crucial for the educational experience in blended instructional formats (Shea & Bidjerano, 2010). The study results were also consistent with two previous works. Firstly, consistent with Werner and Denning (2009) who has asserted that a 3D game-based environment can help students to understand basic programming structures. Secondly, with Paliokas, Arapidis, and Mpimpitsos (2011) who have stressed that programming language transfer in 3D Logo-like environments with playful characteristics has influenced positively students’ participation in programming courses. The results of this study demonstrated an innovative domain in which a 3D game-like environment assisted students’ engagement in collaborative problem-solving tasks using Constructionism as a theoretical underpinning (CP). Lastly, students’ roles were always formed in a collaborative climate and increased their positive attitudes to collaborate with other peers (SP). The blended instructional format also assisted students who had different socio-cognitive backgrounds to gradually become active through collaborative problem-based settings.

Furthermore, students’ attention and time spent on participating in educational activities (TP) seemed to be directly correlated with the exploration, collaboration, teaching plans for the organization, and construction knowledge in well-established instructional settings (CP). The fundamental tendencies for attending introductory programming courses, using a 3D game-like environment such as interaction, sense of co-presence, 3D embodied representations in a challenging persistent workflow motivated students’ participation.

Previous studies (Pellas et al., 2013; Traphagan et al., 2010) that utilized a Col model in 3D multi-user VWs have already revealed the high level of the students’ satisfaction and acceptance in various educational activities. Moreover, learning activities organized within the Col model were considered as significant to support the effective conduct of training courses, in which more easily they try to eliminate organizational complexities and management responsibilities of problems that usually occurred during the first entrance in Open Sim. This distinction clearly indicates that Open Sim can be transformed into an alternative educational platform, in which students can strengthen their computational thinking skills in a collaborative climate. To this notion, they can create a common ground for positive educational implications in similar future-driven tasks.

## Conclusion

In this study, the effect of SP correlation with the CP and TP using a Col model was measured. The results have shown that the SP was significantly correlated with the CP and the TP. These correlations brought a linking part between students’ performance and the “sense of belonging” in a Col as particularly important, especially when group members utilize artifacts and tools in a 3D multi-user VW. The study results reflected not only to all members’ performances that increased, but also to the increased performance of each student in programming courses. Furthermore, the interaction among members (SP) seems to be directly correlated with the CP by determining the exploration, collaboration, teaching plan organization and construction of knowledge for community members. The contribution of Col model presence indicators as theoretical principles for analyzing students’ interactions in a 3D game-like environment becomes to this notion as crucial parameters for an instructional process.

Also, the study findings revealed some other serious pedagogical issues, such as the sense of sociability, interactivity and team collaboration (namely as the principles of the SP) that developed among students. These new foundations can serve new forms of presence indicators and their impact in a Col using a 3D game-like environment for introductory programming courses. Moreover, the attentiveness and time spent on educational activities (CP) seemed to be directly correlated with the TP by determining the exploitation of artefacts via Scratch4OS exploration and collaboration among students, allowing the implementation of constructivist-oriented instructional plans in a blended instructional format. It can be strongly recommended that fundamental tendencies like interaction, sense of co-presence via 3D embodied realistic representations, and a challenging persistent workflow can motivate students to attend and participate pleasantly. It is ultimately understandable that student engagement based on the exploration of presence indicators' interrelationships can be considered as a valuable parameter for successful learning outcomes. The formation of a well-structured and multi-factorial instructional framework for users' interactions can lead to the reinforcement of collaboration and communication with their peers or the instructor. This division very clearly suggests the unique constructs and issues posed to the students' engagement which may guide instructors to determine the suitability and usability of Open Sim in particular learning tasks. Additionally, a constructionist instructional framework based also on Col model presence indicators can give a first positive impact and emphasis on the role of students' active engagement in collaborative problem-based activities for high school introductory programming courses.

The present study carries several implications for instructional technologists and educators, and those with broader research interest in using Col model. These are the following: (a) a clear implication with regards to students' computational thinking skills usage and their specific responsibilities to be taken under account in blended instructional conditions. Col model presence indicators can become the main constructs of an instructional design model; (b) an innovative view of how knowledge acquisition based on Constructionism can influence the view of instruction and design of a 3D game-like environment, in which students' management responsibilities are associated with Col model presence indicators. Last but not least (c) the study findings recommend that the Col model in 3D multi-user VWs can be a useful building block upon which a theory for online and in-class instructional format based on Constructionism can be developed.

Finally, the successful exploitation of the Col model in Open Sim for high school programming courses can be primarily considered as valuable for students due to the formation of a well-structured and multi-factorial framework analysis among users' interactions. This can lead to the refinement of cooperation, communication and community members' engagement. From an instructional- applied level, the main socio-cognitive underpinning was achieved by studying three interrelated presence indicators in an instructional design framework, as it has been successfully presented by a substantial number of scholars and academic researchers (Garrison et al., 2010; Pellas et al., 2013). Likewise, it can be truly believed that the theoretical framework of presence indicators of a Col model can give a first-impact and special emphasis on the role of students' interactions in 3D multi-user VWs.

Inevitably, some notable limitations of this study that should be considered are as follows: (a) the sample size was voluntary (81, i.e. 86.5% correspondents of the total 94 students who enrolled in programming courses) from a Greek high school with up to middle response; (b) this study was deployed and students were taught with a systematic support in a blended session, because the instructor's feedback was sufficient and daily; and (c) students' characteristics may differ from other schools, and this study's findings cannot be generalized.

## Acknowledgment

The author would like to thank all of high school students for participating in this study. Special thanks go to main Computer Science teacher for his prompt assistance and feedback.

## Disclosure statement

No potential conflict of interest was reported by the author.

## Notes on contributor

**Pellas Nikolaos** is a Ph.D. Candidate at the Department of Product and Systems Design Engineering, University of the Aegean, in Greece. He holds a Master degree in Education. His thesis was about the utilization of 3D VWs in e-Education, in which he proposed an organizational-cybernetic framework. His research interests include the use of 3D open-source VWs as “Metaverse” skins in contemporary instructional formats (blended or online) for collaborative learning tasks. He has published more than 30 articles in peer-reviewed Journals, International and National Conferences and book chapters for the domain of instructional design via 3D multi-user VWs.

## References

- Akyol, Z., Vaughan, N., & Garrison, D. R. (2011). The impact of course duration on the development of a community of inquiry. *Interactive Learning Environments*, 19(3), 231–246.
- Anderson, E. F., & McLoughlin, L. (2007). Critters in the classroom: A 3d computer-game-like tool for teaching programming to computer animation students. In *SIGGRAPH '07: ACM SIGGRAPH 2007 educators program* (p. 7). Retrieved from <http://dl.acm.org/citation.cfm?id=1282048>
- Ang, C., Zaphiris, P., & Wilson, S. (2011). A case study analysis of a constructionist knowledge building community with activity theory. *Behavior & Information Technology*, 30(5), 537–554.
- Angelaina, S., & Jimoyiannis, A. (2011). Educational blogging: Developing and investigating a students' community of inquiry. In A. Jimoyiannis (Ed.), *Research on e-learning and ICT in education* (pp. 167–180). New York, NY: Springer.
- Apostolellis, P., Stewart, M., Frisina, C., & Kafura, D. (2014). RaBit EscApe: A board game for computational thinking. In *Proceeding IDC '14 Proceedings of the 2014 conference on Interaction design and children* (pp. 349–352). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=2610489&preflayout=tabs>
- Arbaugh, J., Cleveland-Innes, M., Diaz, S., Garrison, D. R., Ice, P., Richardson, J., & Swan, K. (2008). Developing a community of inquiry instrument: Testing a measure of the community of inquiry framework using a multi-institutional sample. *The Internet and Higher Education*, 11(3–4), 133–136.
- Basawapatna, A., Repenning, A., Koh, K. H., & Nickerson, H. (2013). The zones of proximal flow: Guiding students through a space of computational thinking skills and challenges. In *Proceedings of the ninth annual international ACM conference on international computing education research* (pp. 67–74). (San Diego, CA, August 12–14, 2013). New York, NY: ACM.
- Buraphadeja, V., & Dawson, K. (2008). Content analysis in computer-mediated communications: Analyzing models for assessing critical thinking through the lens of social constructivism. *American Journal of Distance Education*, 22(3), 130–145.
- Burgess, M. L., Slate, J. R., Rojas-LeBouef, A., & LaPrairie, K. (2010). Teaching and learning in Second Life: Using the Community of Inquiry (Col) model to support online instruction with graduate students in instructional technology. *Internet and Higher Education*, 13(1–2), 84–88.
- Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers and Education*, 55(3), 1098–1111.
- Chang C.-I. (2014). Effects of using Alice and Scratch in an introductory programming course for corrective instruction. *Journal of Educational Computing Research*, 51(2), 185–204.
- Chen, J. A., Zap, N., & Dede, C. (2013). Using virtual environments to motivate students to pursue STEM careers: An expectancy-value model. In S. D'Agustino (Ed.), *Immersive environments, augmented realities, and virtual worlds: Assessing future trends in education* (pp. 42–56). Hershey, PA: IGI Global.
- Dalgarno, B., & Lee, M. (2010). What are the learning affordances of 3D virtual environments? *British Journal of Educational Technology*, 41(1), 10–32.
- Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(2), 240–249.
- Dickey, M.D. (2005). Brave (new) interactive worlds: A review of the design affordances and constraints of two 3D virtual worlds as interactive learning environments. *Interactive Learning Environments*, 13(1–2), 121–137.
- Garrison, D. R., Cleveland-Innes, M., & Fung, T. S. (2010). Exploring causal relationships among teaching, cognitive and social presence: Student perceptions of the community of inquiry framework. *The Internet & Higher Education*, 13(2–3), 31–36.
- Garrison, R., Anderson, T., & Archer, W. (2000). Critical inquiry in a text-based environment: Computer conferencing in higher education. *The Internet and Higher Education*, 2(2–3), 87–105.
- Girvan, C., Tagney, B., & Savage, T. (2013). SLurtles: Supporting constructionist learning in second life. *Computers & Education*, 61(1), 115–132.

- Howland, K., & Good, J. (2015). Learning to communicate computationally with flip: A bi-modal programming language for game creation. *Computers & Education*, 80(2), 224–240.
- Ioannidou, A., Repenning, A., & Webb, D. C. (2009). AgentCubes: Incremental 3D end-user development. *Journal of Visual Languages & Computing*, 20(4), 236–251.
- Kalelioglu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52(2), 200–210.
- Kalelioglu, F., & Gülbahar, Y. (2014). The effect of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33–50.
- Kozan, K., & Richardson, J. C. (2014). Interrelationships between and among social, teaching, and cognitive presence. *Internet and Higher Education*, 21, 68–73.
- Li, Z., Cheng, Y., & Liu, C. (2013). A constructionism framework for designing game-like learning systems: Its effect on different learners. *British Journal of Educational Technology*, 44(2), 208–224.
- Li, Z., & Kraemer, E. (2014). Social effects of pair programming mitigate impact of bounded rationality. In *Proceeding SIGCSE '14 Proceedings of the 45th ACM technical symposium on computer science education* (pp. 385–390). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=2538968>
- Paliokas, I., Arapidis, C., Mpimpitsos, M. (2011). Game-based early programming education: The more you play, the more you learn. In Z. Pan et al. (Eds.), *Transactions on Edutainment IX, LNCS 7544* (pp. 115–131). Berlin, Heidelberg: Springer-Verlag.
- Papert, S. (1991). *Mindstorms: Children, computers and powerful ideas*. New York, NY: Basic Books.
- Pellas, N. (2014a). *Virtual communities of inquiry*. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (pp. 7564–7574). Hershey, PA: IGI Global.
- Pellas, N. (2014b). Exploring interrelationships among high school students' engagement factors in introductory programming courses via a 3D multi-user serious game created in Open Sim. *Journal of Universal Computer Science (J. UCS). Special Issue: Immersive Education: What does the future hold*, 20(12), 1608–1628.
- Pellas, N., & Peroutseas, E. (2015). Gaming in second life via Scratch4SL: Engaging high school students in programming courses. *Journal of Educational Computing Research*. doi:10.1177/0735633115612785
- Pellas, N., Peroutseas, E., & Kazanidis, I. (2013). Virtual communities of inquiry (VCol) 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: ACM Press.
- Piaget, J. (1975). *The child's conception of the world* (original work published 1929). Totowa, NJ: Little field Adams.
- Rico, M., Martinez-Mupoz, G., Alaman, X., Camacho, D., & Pulido, E. (2011). Improving the programming experience of high school students by means of virtual worlds. *International Journal of Engineering Education*, 27(1), 52–60.
- Rovai, A. P. (2002). *Building sense of community at a distance*. International Review of Research in Open and Distance Learning, 3. Retrieved June 10, 2013 from <http://www.irrodl.org/content/v3.1/rovai.html>.
- Serrano-Cámara, L., Paredes-Velasco, M., Alcover, C., & Velazquez-Iturbide, J. (2014). An evaluation of students' motivation in computer-supported collaborative learning of programming concepts. *Computers in Human Behavior*, 31(2), 499–508.
- Shea, P., & Bidjerano, T. (2010). Learning presence as a moderator in the community of inquiry model. *Computers & Education*, 59(2), 316–326.
- Swan, K., Richardson, J., Ice, P., Garrison, D. R., Cleveland-Innes, M., & Arbaugh, J. B. (2008). Validating a measurement tool of presence in online communities of inquiry. *E-Mentor*, 2(24), 1–12.
- Szeto, E. (2015). Community of Inquiry as an instructional approach: What effects of teaching, social and cognitive presences are there in blended synchronous learning and teaching? *Computers & Education*, 81, 191–201.
- Traphagan, T., Chiang, Y., Chang, M., Wattanawaha, B., Lee, H., Mayrath, M., ... Resta, P. (2010). Cognitive, social and teaching presence in a virtual world and a text chat. *Computers & Education*, 55(2), 923–936.
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technology*. doi:10.1007/s10639-015-9412-6
- Weinel, M., Bannert, M., Zumbach, J., Hoppe, H. U., & Malzahn, N. (2011). A closer look on social presence as a causing factor in computer-mediated collaboration. *Computers in Human Behavior*, 27(1), 513–521.
- Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education*, 42, 29–49.
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49, 33–35.
- Zhang, J. X., Liu, L., Ordóñez de Pablos, P., & She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using Alice. *International Journal of Engineering Education*, 30(3), 560–565.
- Zhang, Q., Marksbury, N., & Heim, S. (2010). A case study of communication and social interactions in learning in second life. In *Proceedings of HICSS'09 – 42nd Hawaii international conference on system sciences* (pp. 1–9). Hawaii. Retrieved from <https://www.computer.org/csdl/proceedings/hicss/2010/3869/00/01-01-01.pdf>
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2015). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*. doi:10.1177/0735633115608444