Interactive Learning Environments

ISSN: 1049-4820 (Print) 1744-5191 (Online) Journal homepage: http://www.tandfonline.com/loi/nile20

Programming video games and simulations in science education: exploring computational thinking through code analysis

Varvara Garneli & Konstantinos Chorianopoulos

To cite this article: Varvara Garneli & Konstantinos Chorianopoulos (2017): Programming video games and simulations in science education: exploring computational thinking through code analysis, Interactive Learning Environments, DOI: 10.1080/10494820.2017.1337036

To link to this article: http://dx.doi.org/10.1080/10494820.2017.1337036

Published online: 16 Jun 2017.

INTERACTIVE LEARNING ENVIRONMENTS, 2017
https://doi.org/10.1080/10494820.2017.1337036

Programming video games and simulations in science education: exploring computational thinking through code analysis

Varvara Garneli and Konstantinos Chorianopoulos

Department of Informatics, Ionian University, Corfu, Greece

ABSTRACT

Various aspects of computational thinking (CT) could be supported by educational contexts such as simulations and video-games construction. In this field study, potential differences in student motivation and learning were empirically examined through students' code. For this purpose, we performed a teaching intervention that took place over five weeks, with two-hour sessions per week, plus two more weeks for the pretest and post-test projects. Students were taught programming concepts through a science project; one group represented the function of a basic electric circuit by creating a simulation, while the other group represented the same function by creating a video game in which a player should achieve a score in order to win. Video game construction resulted in projects with higher CT skills and more primitives, as measured through projects' code analysis. Moreover, the video-game context seems to better motivate students for future engagement with computing activities.

ARTICLE HISTORY
Received 28 September 2016
Accepted 29 May 2017

KEYWORDS
Serious video games; simulation; computational thinking; computing education; science education

Introduction

Video games and simulations have the potential to motivate and enhance science learning by supporting conceptual understandings (Clark et al., 2011; Clark, Nelson, Sengupta, & D'Angelo, 2009; Lunce, 2006; Squire, Barnett, Grant, & Higginbotham, 2004). Moreover, contexts such as modelling and simulations, game design and development, and robotics could engage students in key aspects of computational thinking (CT) (Lee et al., 2011; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013). CT is a fundamental skill of the twenty-first century and has been first described by Wing (2006) as "the ability to solve problems, design systems, or understand the human behavior". Several programming-based activities have been applied, supporting learners' ability to read and write using a programming language and to think computationally (Román-González, 2015). Computing courses should provide students with chances for self-expression and exploratory learning using coding skills (Guzdial, 2015). However, there is limited understanding on the differences between contexts such as video-games and simulations construction concerning CT skills development and student motivation. Further research could empirically explore and clearly identify the benefits and limitations of diverse contexts on student learning and motivation.

The purpose of this empirical investigation is to explore CT skills development and student motivation under two diverse approaches. Two middle school student groups were taught computer programming in two different ways; one group represented certain physics concepts by creating a simulation, while the other group copied the same physics concepts on a video game. The above interventions lasted two hours per week for five weeks. The experimental procedure included

CONTACT Varvara Garneli c13garn@ionio.gr Department of Informatics, Ionian University, Corfu, Greece

code analysis, informal interviews, and observations. This research results might clearly identify useful guidelines for educators and course designers and also stimulate more research on the topic. So, the main research question of this study is:

RQ: How different contexts like simulations and video-games construction could influence CT skills development and student motivation?

The paper is structured as follows: in the next section, the literature review and the hypotheses of the research are outlined; subsequently, the evaluation methodology employed is presented; following this, the results are discussed; and finally, the findings are summarized.

Background work and research hypothesis

Educational technology could enrich the ways students learn and interpret learning, in formal and informal settings (Säljö, 2010). Developments in video-gaming such as interactive stories, digital authoring tools, and collaborative worlds could promote several suggestions for powerful types of educational media (Squire, 2003). For example, educational games and simulations transport learners to alternative learning settings which require the maximum of their knowledge, skills, and strategies to be implemented.

Simulations

Simulations are social or physical case studies in which several interactive variables have been applied (Gredler, 2004). In such settings, users can actively participate, even when the reality is too dangerous, expensive, complex, fast, or slow (Forinash & Wisman, 2001), and develop intuitive understandings of abstract phenomena (Clark et al., 2011).

Video games

Video games could be applied in educational contexts offering a competitive setting, in which players must apply their knowledge, in order to win (Gredler, 2004). Although video games and simulations share several characteristics in common, an important difference between them could be the video game's incorporated rules and goals accompanied in many cases by scoring or reward systems aimed at tracking the player's progress (Clark et al., 2009). Video games could be considered as "an immensely entertaining and attractive interactive technology built around identities" which supports good learning principles (Gee, 2014). Moreover, video-game pedagogy could benefit different groups of students using methods that move beyond the conventional tool-based approach (Garneli, Giannakos, & Chorianopoulos, 2016). A plethora of studies have confirmed the impact of video games on student attitudes toward the subject being taught and their motivation to attend and engage (McClarty et al., 2012).

Research hypothesis

Traditional teaching could be made more engaging by integrating video-game development features (Seaborn, Seif El-Nasr, Milam, & Yung, 2012). Moreover, a video-game construction could be an enjoyable introduction to Computer Science providing higher order (Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010; Denner, Werner, & Ortiz, 2012) and creative thinking skills (Bennett, Koh, & Repenning, 2011; Navarrete, 2013). Game making activities are based on the educational potential of games, encouraging learners to become digital producers (Kafai, 2006). So we could hypothesize that a video-game construction context could inspire students and support their performance.

H1: A video-games construction could result in projects with more primitives than a simulation construction.

A video-game design approach is aligned with the definition of Wing (2006) regarding CT; designers come to create a system after careful thinking of the users' interface and to apply problem-solving skills in order to implement various video-game features (Kafai & Burke, 2013). The potential of video-games and simulations construction on CT skills has been examined from different viewpoints (Lee et al., 2011; Werner, Denner, Campe, & Kawamoto, 2012). Moreover, a series of studies have already explored video-game design in interdisciplinary curriculums (Wu & Wang, 2012), for example, mathematics or Science (Kafai, 1995; Schanzer, Fisler, & Krishnamurthi, 2013; Yang & Chang, 2013). Such approaches could improve learning in various academic contexts. However, more research could clearly identify potential differences between simulations and video-games construction concerning computing and in particular CT skills development. In this case, we assume that:

H2: A video-game construction could result in projects with better CT skills than a simulation construction.

Educators need to consider more parameters in order to prepare twenty-first century, self-directed, and lifelong learners in the current school settings. There is no doubt that students could become disengaged when dealing with subjects not important or useful to them (Harackiewicz, Tibbetts, Canning, & Hyde, 2014). There are more to be done in order to motivate students instead of using methods such as grading or rewarding. Liu, Wang, and Ryan (2016) defined motivation as "the force which activates, directs, and sustains goal-directed behavior". Lack of motivation could be dealt with by applying interventions which trigger student interest by emphasizing the value of the task, and furthermore sustain and develop this interest in a way that makes possible the reengagement with similar activities and subjects in the future (Hidi & Renninger, 2006). As a result, the following hypothesis regarding the learners' intention to be reengaged in programming-based activities is proposed:

H3: Students' practice in a video-game construction context could positively influence their intention to produce projects using more primitives in comparison to a simulation construction one in a potential reengagement with computing activities.

Course designers and educators come to design interventions in order to inspire students and additionally increase their interest in various academic disciplines. Several parameters might significantly influence such efforts such as the feelings of usefulness, relevance, and autonomy in learning, especially when students consider the discipline from different viewpoints (Harackiewicz et al., 2014; Ryan & Powelson, 1991). Guzdial (2015), for example, suggests that abstract computing concepts should become concrete and related to different interests and values using contexts depending on the specific each time needs. So, we hypothesize that a video-game construction context could influence the students' performance and their intention to create projects in which knowledge and skills from previous activities would be transferred:

H4: Students' practice in a video-game construction context could influence their intention to produce projects with better CT skills in comparison to a simulation construction one in a potential reengagement with computing activities.

Methods

Research design

In this study, we investigated the potential effects of constructing video games and simulations on student learning. For this purpose, we conducted a study which measures CT skills and student motivation and tests the potential effects. The sample of the study consisted of middle school students who practiced in the same computational and physics concepts in two different ways, by constructing a video game or a simulation. The students were taught the same curriculum and constructed the same projects in all sessions except the 5th. In this session, one group completed the simulation by integrating more physics content, while the other group integrated video-games features, transferring the entertaining and competitive potential of video games. In this way, potential differences due to the game features integration could be revealed.

Participants

We performed a between-groups teaching intervention with 44 students. All the students attended the third grade of middle school and were 15 years old. Due to the longitudinal character of the study, the analysis was based on 34 students, 13 boys and 21 girls, the ones who participated in all sessions. They formed two groups and they participated in the study, in two ways. One group represented certain physics concepts by creating a simulation, while the other group copied the same physics concepts on a video game. Both groups were presented with the programming concepts according to the needs of their projects. Both groups were instructed by the same 2 teachers and participants were encouraged to work in pairs during the intervention. However, 4 of them chose to work independently. All of the participants completed the pre- and post-test projects individually. The students were divided into the groups based on the alphabetical order of their names, in the same way classes are normally distributed. From this perspective, our sample was randomly distributed (see Table 1).

Material

The educational context of the study was based on physics and computational concepts. Teachers of IT and science classes and the researchers prepared the educational activities based on the school curricula for students of that age. The students were asked to design an electric circuit using Scratch, a visual programming tool. The electric circuit should consist of one battery and one switch to turn it on or off. When the switch is on, electrons and positive ions move inside the circuit, and an electric lamp also turns on. Both groups represented the function of the electric circuit but through different perspectives. The simulation group was encouraged to represent the circuit functions in order to help someone to study. From this viewpoint, the appropriate interactive variables were implemented, giving users the chance to experiment with the function of a circuit. On the other hand, the video-game group was encouraged to copy the function of a circuit into a video game. Although the purpose of the game was still educational, game features integration was meant to give an enjoyable perspective. Video-game students implemented the appropriate interactive variables like the simulation group. However, video-game users should not just study the circuit, but they should play in order to win. To accomplish this, the students were asked to include at least an avatar, aiming at increasing the game score. However, the implemented game-play was decided by the students themselves and there were no restrictions related to it. Finally, a small storytelling in the beginning of the project was used to provide the description and motivation of it, for both groups.

Measuring instrument

A pretest to examine students' CT skills before the intervention, and a post-test to identify potential differences were given to the students. Mastery or performance could affect student outcomes, but it might be more critical to assess those outcomes which were achieved under student relative autonomy (Deci & Ryan, 2016). Under this perspective, pre- and post-tests were those projects that students

Table 1. Participants of the study.

	Simulation group	Video-game group
Boys	7	6
Girls	10	11
N	17	17

freely decided to create based on their programming skills, in an hour. In addition, the projects which had been created during the intervention were examined.

CT skills include computational concepts such as sequences, loops, events, parallelism, conditionals, operators, data, and additionally computational practices and perspectives. In particular, the construction processes of children while they are engaged in programming activities could be defined as CT practices, and the way students understand themselves and their relationships with others in a technological world could be defined as CT perspectives (Brennan & Resnick, 2012). In this study, students worked on a project framework constructing simulations or video games. In order to accomplish their work, a series of problems needed solutions by implementing various computational concepts and practices. For the needs of this study, CT concepts were examined by Dr Scratch,[1] an online project analysis tool which can assess the CT skills of a scratch project by adding up the partial counts of the various CT skills concepts (see Table 2), and thus calculating the CT skills score (Moreno-León & Robles, 2015). Researchers repeated the work manually in order to cross-check the results. In both ways, CT skills' grading was based on Dr Scratch methodology (see Table 2).

However, Dr Scratch cannot detect fluency on a certain CT concept or assess scores such as debugging and remixing skills. Moreover, a project with the appropriate blocks could get a high CT score, although its functionality might be useless (Moreno-León & Robles, 2015). Due to these limitations, this study was based on more quantitative and qualitative factors.

Performance was considered as a behavioural measure of motivation in terms of higher accuracy and higher amount of work done. In particular, the number of primitives (amount of work done) was examined quantitatively and higher accuracy (code which cannot be executed) was examined qualitatively. These measures were used to capture the students' motivation (Touré-Tillery & Fishbach, 2014).

Table 2. CT skills grading (Moreno-León & Robles, 2015).

CT skills grading	1	2	3
Abstraction and problem decomposition	More than one script and more than one sprite	Definition of blocks	Use of clones
Parallelism	Two scripts on green flag	Two scripts on key pressed Two scripts on sprite clicked on the same sprite	Two scripts on when I receive message Create clone Two scripts when %s is >%s Two scripts on when backdrop change to
Logical thinking	If	If else	Logic operations
Flow control	Sequence of blocks	Repeat, forever	Repeat until
User Interactivity	Green flag	Key pressed Sprite Clicked Ask and wait Mouse blocks	When %s is > %s Video Audio
Data representation	Modifiers of sprites properties	Operations on variables	Operations on lists
Synchronization	Wait	Broadcast When I receive message Stop all Stop program Stop programs sprite	Wait until When backdrop change to Broadcast and wait

Table 3. Study's quantitative factors.

Factors	Description	Source
CT	Computational thinking skills	Moreno-León and Robles (2015)
Nr_of_P	Total number of primitives used in the students' projects	Touré-Tillery and Fishbach (2014)

Table 3 lists the study's factors, their description, and the source from which they were adapted.

We also examined the students' pre- and post-test projects qualitatively, in terms of decisions and choices while programming and execution code errors. Finally, educators kept notes concerning the students' opinion about the intervention, and their projects. The semi-structured interview guide can be found in the Appendix. Project qualitative data, semi-structured interviews with the students, and observations provided the vehicle for interpreting, validating, and discussing the results.

Procedure

In this study, we examined the differences between a video-game and a simulation construction context on CT skills development and the students' motivation.

In the beginning of the intervention, the participants were informed that they will attend programming lessons using the Scratch Environment. Moreover, the students were additionally informed that the lessons will be conducted in a project framework and a small description was given to them. Then, the students created a project according to their prior programming skills (pretest) (Figure 1).

The interventions lasted 5 sessions of two hours each. Computer programming concepts were introduced to both groups as needed (Meerbaum-Salant, Armoni, & Ben-Ari, 2013). Table 4 describes the programming curriculum that was taught and the programming activities in which students were involved, for each group.

In session 5, both groups were taught random values. However, their projects were completed in different ways. Simulation group added a lifelong variable, aiming to randomly influence battery duration. At the same time, the video-game group added an avatar, aiming at achieving a score (variable) and winning (see Figure 2). For example (see Figure 2), an avatar – octopus – appeared when the battery and the switch were on and aimed at destroying the circuit by collecting electrons and avoiding ions. Finally, both groups spent some time in order to refine their work. Both projects needed the same type of CT skills to be implemented.

All students were encouraged to decide the design and coding of their projects. Extra help and information were provided upon request.

In the end, the students were asked to create another project based on the acquired programming skills (post-test) (Figure 3).

The empirical study was conducted in the context of secondary education between January 2014 and February 2014 at a Greek state middle school. The school is located in an urban area and may be considered typical in terms of the number of students, the attendance objectives, and the school infrastructure. This setting might be interesting for educators and researchers because the intervention was conducted in real classroom conditions.

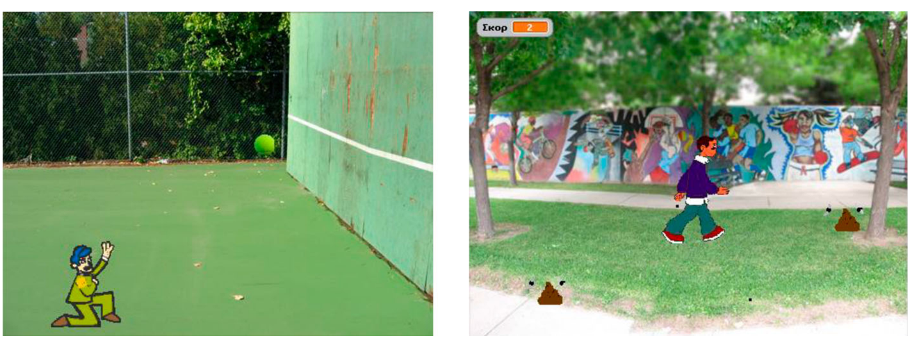


Figure 1. Pretest examples.

Table 4. Students' project description/group.

Programming curriculum	Video-game project	Simulation's project
1. Coordination and synchronization (e.g. Broadcast, When I receive, wait)	Students created two screens, the introductory one and the main screen of the project	
2. Loops and Pen commands for designing	Circuit design using pen commands	
3. Conditionals, variables, and event handlers	Battery and switch which could be turned on/off	
4. Operators for numerical (and boolean) values	Lamp, electrons, and ions move was controlled by the battery's and switch logic value	
5. Random values	Game features: score, avatar, and game-play - Project refinement	Random value to battery sprite in order to influence circuit's duration. Project refinement

Data analysis

Thirty-four middle-school students were involved in this intervention and were divided into two groups: a video-game and a simulation one. First, the projects, which had been completed during the treatments, were assessed. The non-parametric Mann–Whitney *U* test was run to determine potential differences between the groups. In addition, the effect size, which could describe the strength of a phenomenon, was calculated by Eta-squared measure (η^2) (Tomczak & Tomczak, 2014). Then, the pre- and post-test projects were examined. In particular, a non-parametric Wilcoxon signed-rank test (z) was used in order to determine whether there was a median difference between pre- and post-test projects (Conover, 1999). Size effect was examined by the correlation coefficient (r) (Tomczak & Tomczak, 2014). Finally, potential differences between the groups in the pre- and post-phase of the intervention were explored using the non-parametric Mann–Whitney *U* test. Additionally, size effect was calculated by Eta-squared measure (η^2) (Tomczak & Tomczak, 2014).

In addition, this study gathered information from the various informal conversations with students and observations during the intervention. Conversations were conducted randomly with the students who wished to participate. Researchers guided the conversations to probe different aspects of the students' opinion and learning throughout the treatments. Educators encouraged students to talk about their experiences. Informal handwritten notes of the students' answers were made by the researchers during these conversations. Moreover, an extended qualitative study of the projects was performed. Finally, an inductive content analysis of the qualitative data was conducted. First, all the interesting phrases within the informal notes were underlined. Then, an extended discussion among the researchers to code the results of the study was conducted. The steps of the qualitative content analysis (Zhang & Wildemuth, 2016) are described below:

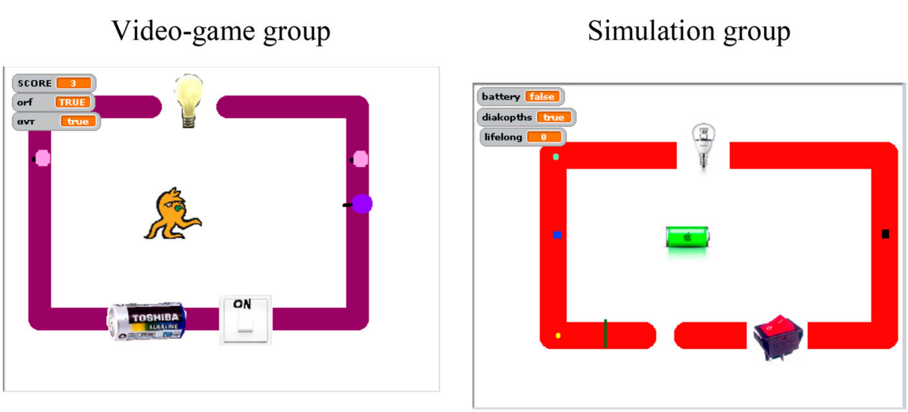


Figure 2. Differences on the 5th session's projects.

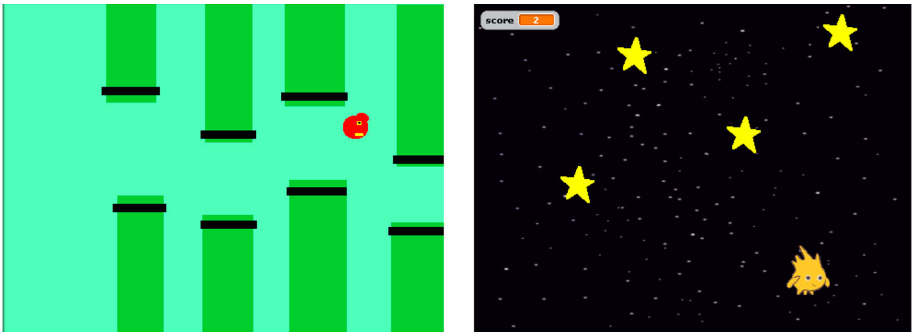
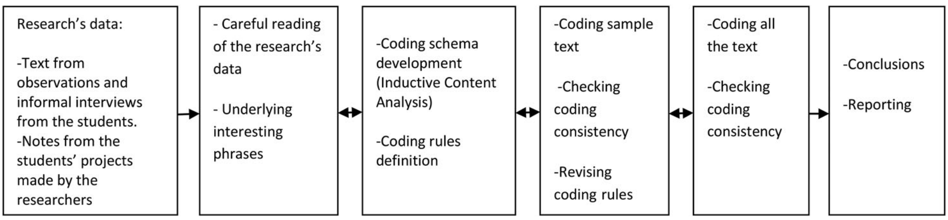


Figure 3. Post-test examples.



The developed coding schema consisted of the following categories: the teaching approach, easiness, and students' feelings about their projects. Moreover, the projects were qualitatively examined concerning the students' decisions to integrate game features in their post-projects and code's accuracy (see Table 5).

Results

Differences between projects created during the treatments

A Mann–Whitney *U* test was run to determine if there were differences between the simulation ($N = 9$) and the video-game group ($N = 10$) in CT and Nr_of_P scores. Distributions of the above scores were similar for both groups, as assessed by visual inspection. CT median was statistically significantly higher in the video-game group (Mdn = 18.500) than in the simulation group (Mdn = 17.000), $U = 75.500$, $z = 2.615$, $p = .010 < .05$, using an exact sampling distribution for *U* (Dineen & Blakesley, 1973). The size effect (n^2) of the CT score was medium (.36). Nr_of_P median was statistically significantly higher in the video-game group (Mdn = 223.500) than in the simulation group (Mdn = 153.000), $U = 81.000$, $z = 2.941$, $p = .002 < .05$, using an exact sampling distribution for *U* (Dineen & Blakesley, 1973). The size effect (n^2) of Nr_of_P score was large (.46) (see Table 6).

Table 5. Coding schema.

Coding categories		Coding rules/examples
Teaching approach	Whether the students felt that the teaching approach met their needs	"I have learned many things about programming and I can create my own projects", "I could complete the required tasks"
Easiness	Students' opinion about the easiness/ difficulty of the approach	"easy, difficult or, tired"
Students' projects	Students' feelings about their projects	"I'm satisfied/pleased with my project"
Execution errors	Code which could not be executed	
Video-game features integration	Integration of video-game features in the students' post-projects	score, game-play, etc.

Table 6. Differences between projects created during the treatments.

	Group medians				
	Simulation	Video game	*U*	*P*	η^2
CT	17.000	18.500	75.500	.010*	0.36
Nr_of_P	153.000	223.500	81.000	.002*	0.46

*The mean difference is significant at the .05 level.

Differences between pretest and post-test projects

The Wilcoxon signed-rank test (*z*) was used in order to determine whether there is a median difference between the pre- and post-test projects (Conover, 1999). The dependent variables' distribution of differences was checked and it was symmetrically shaped for all cases.

Simulation group

Seventeen participants were taught computer programming by constructing a simulation in order to represent certain physics concepts. A pretest to examine CT and Nr_of_P scores before the intervention and a post-test to identify potential differences were given to the students. Of the 17 participants, the CT score increased in 8 projects and decreased in 2 projects, whereas 7 projects had no improvement. The difference in the CT score was symmetrically distributed, as assessed by a histogram. A Wilcoxon signed-rank test determined that there was a statistically significant increase in the CT score between the pre- and the post-phase, $z = -2.310$, $p = .021 < .0\ 5$. The size effect (*r*) of the CT score was medium (−.31) (see Table 7).

Of the 17 participants, the Nr_of_P score increased in 8 projects and decreased in 6 projects, whereas 3 projects had no improvement. The difference in the Nr_of_P score was symmetrically distributed, as assessed by a histogram. A Wilcoxon signed-rank test determined that there was no statistically significant difference in the Nr_of_P score between the pre- and the post-phase, $z = -.659$, $p = .510 > .0\ 5$ (see Table 7).

Video-game group

Seventeen participants were taught computer programming by copying on a video-game the same physics concepts. A pretest to examine CT and Nr_of_P scores before the intervention and a post-test to identify potential differences were given to the students. Of the 17 participants, the CT score increased in 8 projects and decreased in 2 projects, whereas 7 projects had no improvement. The difference in the CT score was symmetrically distributed, as assessed by a histogram. A Wilcoxon signed-rank test determined that there was a statistically significant increase in the CT score between the pre- and the post-phase, $z = -2.298$, $p = .022 < .0\ 5$. The size effect (*r*) of the CT score was medium (−.31) (see Table 8).

Of the 17 participants, the Nr_of_P score increased in 14 projects and decreased in 3 projects. The difference in the Nr_of_P score was symmetrically distributed, as assessed by a histogram. A Wilcoxon signed-rank test determined that there was statistically significant difference in the Nr_of_P score between the pre- (Mdn = 21.000) and the post-phase (Mdn = 31.000), $z = -3.198$, $p = .001 < .0\ 5$. The size effect (r) of the Nr_of_P score was large (−.43) (see Table 8).

Table 7. Simulation group pretest and post-test projects.

	Simulation group (*N* = 17)								
	Medians								
	Pretest	Post-test	Post–pretest	*Z*	*P*	Post > Pretest	Pre > Post-test	Pre = Post-test	*r*
CT	6.000	6.000	.000	−2.310	.021*	8	2	7	−0.31
Nr_of_P	20.000	18.000	.000	−.659	.510	8	6	3	–

*The mean difference is significant at the .05 level.

Table 8. Video-game group pretest and post-test projects.

	Median			Video-game group ($N = 17$)					
	Pretest	Post-test	Post–Pretest	Z	P	Post > Pretest	Pre > Post-test	Pre = Post-test	r
CT	6.000	6.000	0.000	−2.298	.022*	8	2	7	−0.31
Nr_of_P	21.000	31.000	13.000	−3.198	.001*	14	3	–	−0.43

*The mean difference is significant at the .05 level.

Differences between the groups concerning their improvement after the implemented treatments

A Mann–Whitney U test was run to determine if there were differences between the simulation ($N =$ 17) and the video-game group ($N = 17$) in CT and Nr_of_P scores. Distributions of the above scores were similar for both groups, as assessed by visual inspection. The median CT score was not statistically significant in the video-game (.000) and the simulation (.000) group, $U = 149.000$, $z = .161$, $p = .892$, using an exact sampling distribution for U (Dineen & Blakesley, 1973) (see Table 9). The Nr_of_P median was statistically significantly higher in the video-game group (Mdn = 13.000) than in the simulation group (Mdn = .000), $U = 215.500$, $z = 2.448$, $p = .013 < .05$, using an exact sampling distribution for U (Dineen & Blakesley, 1973). The size effect (n^2) of the Nr_of_P score was small (.18) (see Table 9).

Qualitative analysis of the study

Data from projects, interviews, and researchers' observations were used to triangulate research quantitative findings. A first step was to explore which type of projects students chose to create in the pre- and post-phase of the intervention. For example, some students chose to integrate video-game features in their projects, for example, score and game-play, while some others chose to represent dialogues between friends or car and plane accidents in their projects. Moreover, the functionality of the post-test projects was examined by the researchers. Finally, a semi-structured interview guide was used for the in-depth interviews with the students. The students were asked about the implemented approach and the difficulties they faced. Interesting findings were obtained by exploring the students' feelings about their projects. The interviews were conducted randomly during and after the end of the intervention, while notes from interviews and observations were kept by the researcher.

After the projects were examined and the interviews were conducted, all personal information were removed from the collected data before digitalizing them. More analysis was not expected to provide radically different or more in-depth material. Finally, an inductive content analysis was conducted in order to systematically identify properties, attributes, and embedded patterns. This technique could be applied for identifying and analysing issues in the gathered data (Maguire & Bevan, 2002). First, patterns were identified by reviewing the notes, and then we tried to match the patterns with the appropriate notes in order to better explain the quantitative results of the study.

The post-test projects were examined in order to identify pieces of code which could not be executed. Researchers did not find any execution errors except one or two cases per group. Then, the pre- and post-test projects were categorized based on the integration of video-game features. Due to the students' free decision concerning the type of their pre- and post-test projects, this research

Table 9. Differences between the groups' projects in the pre- and post-phases of the intervention.

	Group medians				
	Simulation	Video game	U	P	η^2
CT	.000	.000	149.000	.892	–
Number of primitives	.000	13.000	215.500	.013*	.18

*The mean difference is significant at the .05 level.

explored their intention to integrate video-game features in their projects. In the pre-phase, simulation group students created mostly projects with no video-game features and five video games, while video-game group students created only projects with no video-game features. It seems that the simulation group was more familiar with video-game programming, before the intervention. In the post-phase, the number of projects which included video-game features was increased for both groups, but the difference was bigger for the video-game group (see Figure 4). Despite the bigger difference, most video-game group projects did not integrate game features in the post-phase of the intervention. Nevertheless, some post-projects were still quite complicated and based on interesting contexts.

The students of both groups worked in a project framework which was considered a challenging and complicated approach, at least in the beginning. Only 41% of both groups described the approach as an easy one. Many students needed help and advices by the teachers, while some others preferred to work independently. Some of them complained about various aspects such as the coding difficulties or the graphics quality. Despite the various challenges, most students managed to successfully complete the required tasks and even suggested their own innovative solutions. Moreover, many of them felt that they increased their programming skills during the intervention, especially in the simulation group (58%). Moreover, students had different needs. For example, two girls preferred working on certain programming tasks as in this way they could better learn and successfully complete their work. From this viewpoint, the idea of designing an electric circuit was quite helpful to them and made computer programming and scratch more interesting. Some others applied a lot of imagination on their projects, especially in the video-game group. For example, a boy included a helicopter in the game-play of his project in order to give a funny perspective.

Fifty-nine per cent of video-game students were satisfied with their projects. Some of them mentioned that "Despite the several difficulties we faced in the beginning of the intervention, we liked our work and want to make more programming based projects in the future" or "we learned many things by designing this project" and some of them "wanted to upload their projects on the internet". At the same time, less simulation students, only 41% of them, were pleased with their work. Although they felt confident about their programming skills, they felt tired by the whole effort.

We should mention that some students felt surprised with the idea of integrating game features in an electric circuit. They felt that this action "will destroy their work". Despite their first reaction, the whole effort turned out to be fun for many of them. Finally, a common feeling in both groups was concerning the physics context, at least in the beginning of the intervention. For example, they wondered "what kind of video-game we will create within a physics context" or "but why physics". A boy was really disappointed and stated that "I do not like physics and if I decide to create another project, it will certainly be in another context".

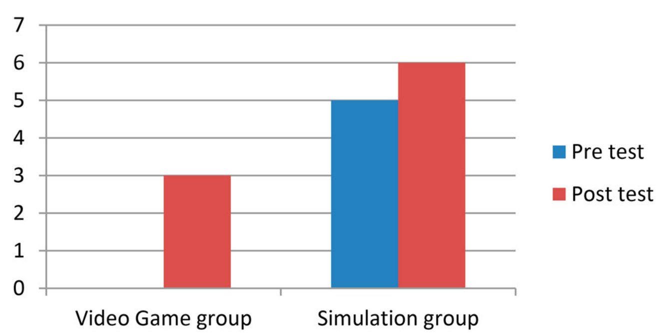


Figure 4. Video-game features integration between the pre- and post-phase.

Discussion

This study explored the potential differences in CT skills development and motivation in two diverse construction contexts: a simulation and a video game. The students constructed their projects in order to learn science and computational concepts. They were taught the same curriculum and constructed the same projects in all sessions except the 5th. In this session, one group completed the simulation by integrating more physics content, while the other group integrated video-game features, transferring the entertaining and competitive potential of video games. In this way, potential differences due to the game features integration could be revealed. Although in game-based simulations the absolute accuracy of the model representation might be sacrificed for the sake of fun, the benefits for the learners' motivation might be significant (Sawyer & Rejeski, 2002). Moreover, learning settings could vary depending on the different learning goals and student needs. From this study viewpoint, CT skills development and student motivation were examined.

Both group projects, the simulation and the video-game one, needed the same type of CT skills. Although the approach was considered demanding according the students' interviews and the teachers' observations, all the students created projects with advanced CT skills during the intervention. Comparing the projects, video-game ones included significantly more primitives (H1) and advanced CT skills (H2). Moreover, the video-game students were more satisfied with their work. It seems that the video-game construction approach could be demanding, but it better motivates student learning of coding and science content and also promotes CT skills development.

In the post-phase of the intervention, the video-game projects were significantly improved in both CT skills and the number of primitives scores, while the simulation projects were improved only in CT skills. Comparing the scores of the groups, a significant difference was observed only in the number of primitives' score, for the video-game group (H3), while there was no significant difference in CT skills (H4). Additionally, the qualitative analysis revealed no difference concerning the post-test projects' accuracy. Therefore, video-game students had higher performance due to the difference in the amount of work done. Quantitative and qualitative results suggested that the role of game designer could influence the students' motivation more when they are reengaged in programming activities. Instructional materials should be motivating for students by using appealing contexts (Parker & Lepper, 1992).

During the intervention, video-game students created projects with advanced CT skills integrating video-game features such as score and game-play. The qualitative analysis revealed that some students chose to create video games in the post-phase, but most of them did not. The students had already experienced the challenge of the video-game construction process, which needs a lot of time and professional work (Van Eck, 2006). Nevertheless, the intervention context did not affect the students' motivation towards computing. The video-games students chose to create interesting projects within multiple contexts using many primitives. Besides, students use several digital applications in their daily life and probably would prefer to create similar applications with "professional" graphics and sound, advanced user interfaces, and emotions (Zyda, 2005). This potential could encourage more of them to become digital producers. Professionals from the video-game industry could support such varied needs of students and their expectations by providing several powerful environments.

There are various CT assessment tests such as, for example, multiple-choice questions (Grover, Cooper, & Pea, 2014; Román-González, 2015). Such methods could be applied to a large number of students and provide valuable information. However, this study was aimed at assessing the students' intention to use and further develop the acquired skills and, thus, such tests might give faulty results and drive to misleading decisions. This study's evaluation methodology was based on the students' projects themselves, which were examined quantitatively and qualitatively and became an indicator of CT skills development. Moreover, we examined not only the completed projects during the intervention, but additionally 2 projects which were created by the students in the beginning and at the end of the intervention, the pre- and post-test projects. These projects aimed at exploring those skills which the students chose to include in their projects. In this way, potential differences among students' motivation due to the treatments might be identified.

In general, our results could suggest the enrichment of the learning process with contexts like video-game construction in multidisciplinary settings aiming at motivating students' learning and promoting CT skills development. Although this study has proposed video-game construction in a science context, there are some limitations. First, qualitatively different learning environments offer different kinds of learning experiences and thus serve different learning goals (Rosen & Salomon, 2007). Despite our efforts to base our interventions on the same learning theory, we could not avoid small differences due to the design of the study. Thus, some results might have been influenced by such differences. Moreover, the generalizability of these results must be carefully considered, because the field study was conducted in a specific context (e.g. content and age). Finally, CT skills development and motivation towards computing were mostly based on the projects' code analysis of a small number of participants. However, research on the topic is still limited and, thus, we used more in-depth methods such as interviews and observations in order to provide a complementary picture of the findings.

This study's results might offer interesting suggestions to educators and course designers concerning student learning and motivation in multidisciplinary programming-based contexts. Future research could explore more interesting parameters such as different learning goals, student needs, or social interactions and evaluate new powerful environments in order to support different needs and expectations and enrich formal and informal learning settings.

Conclusions

This work's viewpoint considers computing as a powerful medium using programming to motivate exploratory learning and promote CT skills development. In particular, the effects of contexts like simulation and video-game construction on CT skills development and student motivation were explored. The results of this research were based on project code analysis, in a quantitative and qualitative manner.

Based on the findings of this research, some useful guidelines could be outlined. We found that despite the challenges of a video-game construction approach, learning coding and science concepts through game making could motivate student learning, in formal school settings. Moreover, the role of game designer could influence the students' intention to be reengaged with programming-based activities. Finally, the students' need to create "real" digital applications based on advanced graphics, sounds, and user interfaces was revealed. Powerful environments could better motivate students supporting different needs and expectations.

Note

1. http://drscratch.programamos.es/

Acknowledgements

The authors would like to thank all the students and the school teachers for their participation in the didactic intervention. We would also like to thank Michail N . Giannakos for his help in designing and conducting this research and Letizia Jaccheri for her useful comments and support in an earlier phase of the project. In the reported research, the appropriate procedures were followed according to institutional contexts and ethics. Every student participated in the study voluntarily. Data obtained from the students were fully anonymous, and derived through pre- and post-tests, surveys, and observation. The survey did not contain questions of a personally sensitive nature. The collected data were analysed in an aggregate manner and used exclusively for educational and research purposes.

Disclosure statement

There is no conflict of interest within this study, and the data set sharing is a natural extension of open access, at the request of the authors.

Notes on contributors

Varvara Garneli is a PhD candidate at the Department of Informatics of the Ionian University. Her current research involves the use of game programming in computing and science education.

Konstantinos Chorianopoulos is a maker, scholar, and mentor. He has designed open systems for research and learning. He has been teaching at the Ionian University and at the Hellenic Open University.

References

Bennett, V. E., Koh, K. H., & Repenning, A. (2011, June). CS education re-kindles creativity in public schools. In *Proceedings of the 16th annual joint conference on innovation and technology in computer science education* (pp. 183–187). New York, NY: ACM.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver (pp. 1–25).

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, *55*(3), 1098–1111.

Clark, D. B., Nelson, B. C., Chang, H. Y., Martinez-Garza, M., Slack, K., & D'Angelo, C. M. (2011). Exploring Newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in Taiwan and the United States. *Computers & Education*, *57*(3), 2178–2195.

Clark, D., Nelson, B., Sengupta, P., & D'Angelo, C. (2009, October). Rethinking science learning through digital games and simulations: Genres, examples, and evidence. In *Learning science: Computer games, simulations, and education workshop sponsored by the National Academy of Sciences*, Washington, DC.

Conover, W. J. (1999). *Practical nonparametric statistics* (3rd ed.). Hoboken, NJ: Wiley.

Deci, E. L., & Ryan, R. M. (2016). Optimizing students' motivation in the Era of testing and pressure: A self-determination theory perspective. In W. C. Liu, J. C. K. Wang, & R. M. Ryan (Eds.), *Building autonomous learners* (pp. 9–29). Singapore: Springer.

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*(1), 240–249.

Dineen, L. C., & Blakesley, B. C. (1973). Algorithm AS 62: Generator for the sampling distribution of the Mann–Whitney U statistic. *Applied Statistics*, *22*, 269–273.

Forinash, K., & Wisman, R. (2001). The viability of distance education science laboratories. *The Journal (Technological Horizons in Education)*, *29*(2), 38.

Garneli, V., Giannakos, M., & Chorianopoulos, K. (2016). Serious games as a malleable learning medium: The effects of narrative, gameplay, and making on students' performance and attitudes. *British Journal of Educational Technology*. doi:10.1111/bjet.12455

Gee, J. P. (2014). *What video games have to teach us about learning and literacy*. Macmillan. Retrieved August 10, 2016, from http://s3.amazonaws.com/academia.edu.documents/

Gredler, M. E. (2004). Games and simulations and their relationships to learning. *Handbook of Research on Educational Communications and Technology*, *2*, 571–581.

Grover, S., Cooper, S., & Pea, R. (2014, June). Assessing computational learning in K-12. *Proceedings of the 2014 conference on innovation & technology in computer science education*, Uppsala, Sweden, pp. 57–62.

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, *8*(6), 1–165.

Harackiewicz, J. M., Tibbetts, Y., Canning, E., & Hyde, J. S. (2014). Harnessing values to promote motivation in education. *Advances in Motivation and Achievement*, *18*, 71–105.

Hidi, S., & Renninger, K. A. (2006). The four-phase model of interest development.*Educational Psychologist*, *41*(2), 111–127.

Kafai, Y. B. (2006). Playing and making games for learning instructionist and constructionist perspectives for game studies. *Games and Culture*, *1*(1), 36–40.

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. New York: Routledge.

Kafai, Y. B., & Burke, Q. (2013 March). The social turn in K-12 programming: Moving from computational thinking to computational participation. *Proceeding of the 44th ACM technical symposium on computer science education*, Denver, Colorado, USA, 603–608. doi:10.1145/2445196.2445373

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., … Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, *2*(1), 32–37.

Liu, W. C., Wang, J. C. K., & Ryan, R. M. (2016). Understanding motivation in education: Theoretical and practical considerations. In W. C. Liu , J. C. K. Wang, & R. M. Ryan (Eds.), *Building autonomous learners* (pp. 1–7). Singapore : Springer. doi:10.1007/978-981-287-630-0

Lunce, L. M. (2006). Simulations: Bringing the benefits of situated learning to the traditional classroom. *Journal of Applied Educational Technology*, *3*(1), 37–45.

Maguire, M., & Bevan, N. (2002). User requirements analysis: A review of supporting methods. In J. Hammond, T. Gross, & J. Wesson (Eds.), *IFIP WCC 2002. IFIP, 99* (pp. 133–148). Boston, MA: Springer.

McClarty, K. L., Orr, A., Frey, P. M., Dolan, R. P., Vassileva, V., & McVay, A. (2012). A literature review of gaming in education. *Gaming in Education*. Retrieved January 2017, from http://images.pearsonassessments.com/images/tmrs/tmrs/Lit_Review_of_Gaming_in_Education.pdf

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, *23*(3), 239–264.

Moreno-León, J., & Robles, G. (2015). Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills, Amsterdam, Netherlands, pp. 12–15.

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education*, *69*, 320–331.

Parker, E. L., & Lepper, M. R. (1992). Effects of fantasy contexts on children's learning and motivation: Making learning more fun. *Journal of Personality and Social Psychology*, *62*(4), 625–633.

Román-González, M. (2015, July 6–8). Computational thinking test: Design guidelines and content validation. In *Proceedings of the 7th annual international conference on Education and New Learning Technologies* (EDULEARN 2015), Barcelona (pp. 2436–2444). Retrieved August 2016, from https://library.iated.org/view/ROMANGONZALEZ2015COM

Rosen, Y., & Salomon, G. (2007). The differential learning achievements of constructivist technology-intensive learning environments as compared with traditional ones: A meta-analysis. *Journal of Educational Computing Research*, *36* (1), 1–14.

Ryan, R. M., & Powelson, C. L. (1991). Autonomy and relatedness as fundamental to motivation and education. *The Journal of Experimental Education*, *60*(1), 49–66.

Sawyer, B., & Rejeski, D. (2002). *Serious games: Improving public policy through game-based learning and simulation*. Foresight and Governance Project, Whitepaper for the Woodrow Wilson International Center for Scholars Publication. Retrieved March, 2016, from https://www.wilsoncenter.org/sites/default/files/ACF3F.pdf/

Säljö, R. (2010). Digital tools and challenges to institutional traditions of learning: Technologies, social memory and the performative nature of learning. *Journal of Computer Assisted Learning*, *26*(1), 53–64.

Schanzer, E., Fisler, K., & Krishnamurthi, S. (2013). *Bootstrap: Going beyond programming in after-school computer science*. Presented at SPLASH_E Education track of the OOPSLA/SPLASH conference, Indiapolis, IN.

Seaborn, K., Seif El-Nasr, M., Milam, D., & Yung, D. (2012, February). Programming, PWNed: Using digital game development to enhance learners' competency and self-efficacy in a high school computing science course. *Proceedings of the 43rd ACM technical symposium on computer science education*, Raleigh, CA, USA., pp. 93–98. doi:10.1145/2157136.2157169

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351–380.

Squire, K. (2003). Video games in education. *International Journal of Intelligent Games & Simulation*, *2*(1), 49–62.

Squire, K., Barnett, M., Grant, J. M., & Higginbotham, T. (2004, June). Electromagnetism supercharged! Learning physics with digital simulation games. *Proceedings of the 6th international conference on learning sciences* (pp. 513–520). Santa Monica, CA: International Society of the Learning Sciences.

Tomczak, M., & Tomczak, E. (2014). The need to report effect size estimates revisited. An overview of some recommended measures of effect size. *Trends in Sport Sciences*, *21*(1), 19–25.

Touré-Tillery, M., & Fishbach, A. (2014). How to measure motivation: A guide for the experimental social psychologist. *Social and Personality Psychology Compass*, *8*(7), 328–341. doi:10.1111/spc3.12110

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE Review*, *41* (2), 16.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012, February). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM technical symposium on computer science education*, Raleigh, NC, USA, pp. 215–220. doi:10.1145/2157136.2157200

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wu, B., & Wang, A. I. (2012). A guideline for game development-based learning: A literature review. *International Journal of Computer Games Technology*, *2012*). Article ID: 103719. doi:10.1155/2012/103710

Yang, Y. C., & Chang, C. L. (2013). Empowering students through digital game authorship: Enhancing concentration, critical thinking, and academic achievement. *Computers & Education*, *68*, 334–344.

Zhang, Y., & Wildemuth, B. M. (2016). Qualitative analysis of content. In *Applications of social research methods to questions in information and library science* (p. 318). ABC-CLIO. Retrieved January 2017 from https://www.ischool.utexas.edu/~yanz/Content_analysis.pdf
Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, *38*(9), 25–32.

Appendix: Semi-structured Interview Guide

Capturing answers: Recording of answers will be done through taking notes. This procedure allows the interviewer to highlight key points, guide the discussion depending on different students' reactions, and may make the production of the final notes and their evaluation quicker because there is no need to wade through large files of transcripts.

Develop a rapport with the respondent: Obtaining meaningful information from respondents could be easier if the interview's atmosphere is not formal. This can be done by using questions related to students' hobbies, their spare time, and so on. Another significant parameter could be that the questions should lead to detailed answers and not a simple "Yes" or "No".

Examples of questions:

- Was the instructional approach according to your needs?
- Which difficulties did you face during the lessons?
- Do you like the project you created?
- Do you want to share your work with the others?

It is good to have a set of questions at hand, but the interviewer needs to also be prepared to expand on or probe the predetermined questions as the need arises. This is the essence of qualitative interviews.

End the interview: Deciding when to end an interview may depend on several factors. For example, interviewers may feel that they have exhausted their questions, and that they are no longer getting new information, or if the respondent seems tired or has other commitments to attend to.

Finally, it is important to thank the respondents for their time. It is also a good practice for interviewers to summarize the key points that they feel the respondent has provided, because this gives the respondent a final chance to expand or clarify any points.