education sciences

MDPI

Article

Systematic Design and Rapid Development of Motion-Based Touchless Games for Enhancing Students' Thinking Skills

Ioannis Altanis *, Symeon Retalis and Ourania Petropoulou

Department of Digital Systems, University of Piraeus, 80, M. Karaoli & A. Dimitriou St., 18534 Piraeus, Greece; retal@unipi.gr (S.R.); rpetro@biomed.ntua.gr (O.P.)
* Correspondence: galtanis@unipi.gr; Tel.: +30-210-344-3329

Received: 1 December 2017; Accepted: 18 January 2018; Published: 22 January 2018

Abstract: During the last few years, there has been a growing interest in students getting engaged in digital game-making activities so as to enhance their thinking skills. The findings of studies that have examined the impact of such initiatives are quite positive, especially concerning the promotion of 21st century skills; however, many students seem to face difficulties in getting a deeper understanding of the game development life cycle. Additionally, students often have difficulties in meaningfully reusing and applying the concepts from various subjects, mainly mathematics and physics, into their game-making tasks or in understanding advanced programming commands while creating their games. The present study presents an innovative game-making teaching approach that suggests a series of steps for the systematic design and rapid development of motion-based touchless games, i.e., games that are based on natural user interaction technologies, like the Microsoft Kinect camera. Findings from evaluation studies in two (2) secondary schools indicate that this approach can increase student motivation, strengthen their computational thinking, enhance their understanding of geometric principles and improve their social skills.

Keywords: technology enhanced learning; Game-based learning; Kinect; Formative evaluation; secondary education; thinking skills

1. Introduction

During the last decade, there has been a considerable interest in the examination of the educational potential of digital game-making activities by students in an attempt to find an engaging way to help them enhance multiple skills [1,2]. Although the findings of relevant studies are quite positive, many students seem to face difficulties in deeply understanding the systematic game development life cycle. This mainly occurs due to the fact that students tend to focus only on certain phases of the game development life cycle. They either focus on the game design phase only or spend a lot of time and energy in becoming competent in building games using specialised authoring tools. As a result, students do not have the opportunity to participate in a systematic design and development process that could lead to the enhancement of higher-order thinking skills such as problem-solving, computational thinking, communication and cooperation. Furthermore, recent studies have found that students often fail to meaningfully embed knowledge and principles from math and physics subjects into their games [3], or understand the use of the many aspects of introductory programming such as variables, loops and Boolean operators [4,5].

In this paper, an innovative approach that promotes the acquisition of multiple thinking skills via game-making activities is being presented. More specifically, students are called to systematic design and create medium fidelity of interactive Kinect motion-based games with the ultimate goals of increasing their motivation, enhancing their learning and understanding of geometric principles and improving their programming skills. Since motion-based touchless technology is a new trend in the field of human–computer interaction (also called Natural User Interaction-NUI), this specific type of interactive games seems to motivate students to get involved. Researchers have already mentioned that students show high interest and motivation in interacting with such type of games that can be played with hand and body gestures and do not require the use of a keyboard, mouse or joystick [6]. Furthermore, students' participation in Kinect-based learning activities strengthens their concentration, encourages active classroom participation and fosters discussion and brainstorming [6]. In addition, boosting students to become creators of their own motion-based touchless games and not just players can be very constructive, especially for kinesthetic learners. These are students who require whole-body movements to process new and difficult information [7]. Also, when students are designing this controller-free type of game, they have to understand and reason about the spatial relations among objects in the 3D space [8]. Designing and developing Kinect games can help children build strong connections among the body, the space and the abstract representation of angles and geometry concepts [9].

Although tools such as Scratch [10] and Kinect2Scratch [11] help students and teachers to overcome technical limitations in developing motion-based touchless games, up to our knowledge there are very few research studies that have examined the advantages of the motion-based touchless game making activities. Till recently, the emphasis has been mainly given on studies that measure the learning effectiveness of the use of such games as learning tools.

This paper tackles this open research topic. It presents the step-by-step systemic design and development of medium fidelity motion-based touchless games by the students themselves, which can lead to an enhancement of their thinking skills and increase the students' motivation. The structure of the paper is as follows: a brief overview of related game-making approaches that appear in the literature and which have given input for the creation of the proposed innovative teaching approach is mentioned. Next, the elements of the proposed approach, i.e., the stages/concrete steps, the worksheets, examples of gesture cards that can help children in conceiving the mechanisms of natural user interaction that will be embedded into their games are presented. Finally, the details of the evaluation study that was performed in the authentic environments of the two schools are given (e.g., participants, evaluation framework and data collection tools). The paper closes with the discussion of the study findings and the future research directions.

2. Students are Game Designers and Developers

The creation of a digital game is an ill-structured problem. In order for students to be able to solve an ill-structured problem and deeply understand the game development life cycle, they should break it into smaller and simpler problems to help them enhance abstraction. Although game development life cycle guidelines (GDLC) have different characteristics and several pros and cons, there are three generic phases accepted in the game development process. These are: "(1) Design and prototype: the process of creating initial game design, game concept, and put it into a form of playable prototype, (2) Production: the process of making the source code, creating the assets, and integrating them as one, (3) Testing: the process of playtesting, whether it is conducted by internal team members or third-party testers" [12]. Afterwards, students should recognise the necessary content of these 3 core phases.

In order students to design a game idea and create a playable prototype, they should make decisions about the ***elements*** that make up the game as a ***system***, how these elements should be interrelated and ***balance***, in order to create the desired ***flow*** that could lead to positive ***user*** experience (playability). According to the Gamestar Mechanic teacher pack [13], the five (5) core elements of game design are space, goals, mechanics, rules and components (e.g., avatar, enemies and blocks). Through brainstorming, designers cooperate and collaborate with other team members and make important decisions about designing and analysing a game-system, which is crucial for enhancing problem-solving and thinking skills [14]. In addition, storyboards help students to rapidly visualise game flow, thereby creating a low-fidelity game by using only pen and paper. In some cases, tangible cards could support the game design process, either as a learning tool to distinguish game elements [15–17] or as an aid to students for generating creative game ideas [6,17,18].

During the last decade, researchers and educators have used specialised tools such as Scratch, Kodu, GameMaker and AppInventor, for the needs of the production phase in order to help students rapidly build their games. The MIT Media Laboratory Scratch is one of the most popular and easy to use programming languages for creating stories, games and animations, mainly in primary and secondary schools [10]. The fact that many secondary students already have previous experience on Scratch promotes the rapid implementation of other future more advanced studies, overcoming the students' and teachers' initial development issues. Moreover, in order to rapidly develop Kinect games with Scratch, the tool Kinect2Scratch [11], developed by Stephen Howell, bridges the two technologies.

After developing a game, a testing phase is crucial not only for troubleshooting but also for providing an opportunity for the participants to match their initial goals and requirements with their deliverables (design documents, storyboard and demos). For these reasons, qualitative and quantitative analysis can be used especially from multiple sources to enhance the trustworthiness of a study. Common assessment tools are observation field notes, audio/video recordings data, deliverables collection, interviews, evaluation forms and rubrics and pre/post surveys so as to evaluate student attitudes, motivation and programming concepts [19–24]. Furthermore, there are free available web tools that automatically explore the presence of programming concepts in the students' Scratch files, such as Dr. Scratch [25,26] and Scrape [27].

3. The proposed Approach

3.1. Proposed Game Design and Development Stages

The proposed teaching approach consisted of a 6-stage process. (Figure 1)

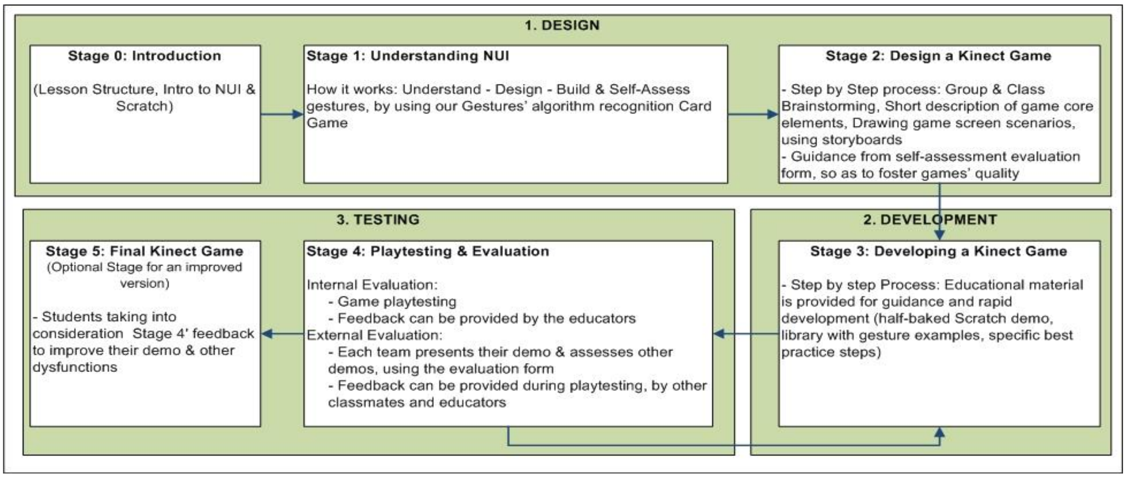


Figure 1. Proposed game design and development stages.

Stage 0 (Introduction) introduced students to the structure of the lesson, the goals, the time schedule and the natural user interaction (NUI), through live Kinect examples and playtesting demos. In addition, a working sheet was provided to students so as to help them refresh their prior knowledge in the use of Scratch (e.g., create sprites and costumes) and to introduce them to a certain number of programming commands (if, if-else, forever, repeat, events and variables), which are considered necessary for the implementation of the teaching approach. Students used the working sheet to complete eight Scratch educational activities during the 1-h session. Finally, a structured online questionnaire was used to capture the students' profile.

Stage 1 (Understanding NUI) provided students with the opportunity to design, create and test body postures and gestures using Scratch, Kinect2Scratch and Kinect camera. Multimedia educational material was provided to introduce participants to natural user interaction technologies (NUI). To boost the students' interest and help them progressively understand NUI, forty (40) examples of gestures were created and organised in four difficulty levels (Figure 2). These gestures were transformed into gesture tangible cards (Figure 3) and were incorporated in an educational game activity for a deeper understanding. Each group of students randomly chose 6 cards (2 green, 2 blue, 1 orange, 1 red) and tried to find the appropriate algorithm. MS Kinect camera was available with a road-light demo in Scratch for live playtesting.

	Difficulty criteria			4. Sublevels - Difficulty criteria		
DIFFICULTY LEVELS	1. Number of required body joints	2. Required Postures for gesture execution (1 or 2)	3. Visibility of body joints in gesture cards (on/off)	4.1 Just transfer a virtual object (eg. just move an object - drag&drop)	4.2 Compare 2 body joints value in x,y or z axis (e.g A>B)	4.3 Compare the distance between 2 body joints in x,y or z axis (e.g. A-B> 100)
Green	2	1 (e.g. raise your left_hand over your head)	on	Included	Included	Included
Blue	3 to 6	1 (e.g. create a bridge with your hands)	on	-	Included	Included
Orange	>=7	2 (e.g. Bowling gesture. 1st Posture: Left_hand behind your body --> 2nd Posture: Left_hand in front of your body- compare body joints in z axis)	on	-	Included	Included
Red	>=3	1 or 2	off	-	Included	Included

Figure 2. Gesture Difficulty Levels.

Figure 3. Gesture Tangible Cards (Examples).

During stage 2 (Design a Kinect game), students designed their games in a decision-making process, by describing in a structured game design document their ideas, the game's target group and the five (5) core game elements (space, game goal, avatars/enemies' sprites, core-mechanic and rules). In addition, students drew their game scenarios using storyboards for providing a logical organisation, an effective structure and guidance for the reader. The following tools were also provided to foster the student's thinking and self-assessment of the quality of their games:

1. An evaluation rubric [25] for the design artefacts (storyboard and game design document).
2. A 5-Likert scale game evaluation form, which contained 13 criteria based on the common design heuristics and two (2) open questions about the pros and cons of the game (Appendix A).

Stages 3 (Development) and 4 (Playtesting and Evaluation) were implemented as a rapid loop process for development, playtesting, internal evaluation and troubleshooting. Learning resources in the development stage contained: a half-baked demo, gestures' examples library and a working development sheet. During the external evaluation (stage 4): (a) each group of students played their game in the class and (b) playtesting by other groups was also available in order to assess the produced Kinect games. Finally, stage 5 is optional and can work either as an additional stage for the refinement of the games based on the feedback received during stage 4. In practice, due to time limitations, the game making process ends at stage 4.

3.2. Timeline and Deliverables

This approach seems appropriate for secondary school students (12–15-year-olds) who have limited previous experience with specialised programming tools (mainly Scratch like tools). The required timeline is 8 to 9 weeks, depending on the students' prior experience on Scratch and programming concepts. Taken for granted the aforementioned four stages of the game-making design and development process, students have to work on and submit various deliverables that are presented in Table 1.

Table 1. Students Timeline and Deliverables.

Stage	Timeline(Weeks)	Deliverables
Stage 0: Introduction	1	• Initial online questionnaire • Eight (8) Scratch files
Stage 1: Understanding NUI	1	• Working sheet with gesture algorithms
Stage 2: Design a Kinect Game	2	• Game design document and storyboard
Stage 3: Developing a Kinect Game	2–3	• Final product: Kinect game • Kinect Games' evaluation forms • Final Online Questionnaire
Stage 4: Playtesting and Evaluation	2	
Total	**8–9**	

4. Evaluation Study

4.1. Context

The study was conducted in two (2) secondary urban schools in the broader area of Athens, Greece for 9 weeks (March–April 2017). Clubs with students who showed interest in programming and games were formed in both schools to apply the proposed game-making approach. Students of the first school were involved in these tasks during normal school periods. For the other school, this was an after-school initiative. Students were spending one class session of approximately 1.5 h (90 min) per week on this initiative. Due to the time limitation, the optional stage 5 was not performed.

One month prior to the initiation of the program, computer science (CS) teachers were given the content package including the learning resources of the approach, the hardware (Kinect camera) as well as the explanations of the key research questions:

- Were the students' thinking skills enhanced as a result of the approach?

- Was the proposed educational approach implemented successfully and as planned in the school environments?
- Was the proposed educational approach appreciated by the participants?

4.2. Participants

The study population consisted of 22 school students (with an average age of 15 years). Twelve (12) students (10 males, 2 females) participated from the first school and ten (10) students (6 males, 4 females) participated from the second. All students volunteered to participate in the study. They admitted that the development of motion-based touchless games seemed a highly motivating and intriguing task for them. Furthermore, each teacher organised his/her students in groups of two, according to the participants' profile (grouped to mix the skill levels) and eleven (11) groups were created.

Regarding the students' profile, results from an initial online questionnaire indicated that:

- 45.45% to 68.18% of them had excellent grades (18–20/20) in the four STEM lessons (Physics, Mathematics, Chemistry and Computer Science).
- Most of them (55.71%) had zero to little previous experience in playing Kinect games and zero to little participation in other game design/development activities.
- All participants had previous experience in Scratch; however, only 17.24% of them declared they were confident (level 4 and 5 on the 5-Likert scale).
- The students' fundamental initial goal was programming. Moreover, the students' answers regarding the question "*What was/were your reason(s) for enrolling in this program?*" were grouped and are presented in the following Table 2.

Finally, computer science (CS) teachers administered these sessions, coordinated the students' tasks and offered scaffolds about the process, when needed. Moreover, the principal researcher attended some sessions and made observations. He also supported, mainly, the teachers and the students, when necessary.

Table 2. Students' Initial Goals Classification.

Initial Students' Goals Classification	Number of Students
Learn Programming	15/22
Create Games	13/22
Understand the Game Design Process	6/22
Learn Scratch	7/22
Get Familiarised/Interact with Kinect	5/22
Have fun	5/22
Enhance cooperation skills	2/22

4.3. Data Collection

Both qualitative and quantitative data were collected during these sessions. Assessment tools per Research Question (RQ) are presented below in Table 3. They appear in the literature [24–29] as they had been used in related studies. For the needs of this study, they had been slightly adapted (e.g., more specific questions have been added about the motion-based games and natural user interaction).

Table 3. Research Assessment tools.

Research Questions (RQ)	Research Assessment Tools
RQ1: Were the students' thinking skills enhanced as a result of the approach?	For Computational Thinking Skills: • Evaluation rubric, based on Dr. Scratch (Criteria and Best Practices) [25,26] • Scrape tool for automatic quantitative analysis of Scratch programming concepts [27] • Observation field notes were kept on a journal For Social Skills (cooperation, collaboration, team spirit): • Final student's questionnaire [28,29] • Final teacher's questionnaire [28,29]
RQ2: Was the proposed educational approach implemented successfully and as planned in the school environments?	• Evaluation rubric for the assessment of the design artefacts (storyboards and design documents) [24] • 5-Likert scale games' evaluation form (Appendix A—Figure A1)
RQ3: Was the proposed educational approach appreciated by the participants?	• Final student's questionnaire [28,29] • Final teacher's questionnaire [28,29]

Regarding the first research question (RQ1), the enhancement of higher-order thinking skills was examined, in terms of **computational thinking skills (CT skills)** and **social skills**.

To understand whether the students enhanced their **CT skills**, an in-depth analysis of the students' games (Scratch files) was performed, in order to:

(a) Measure the computational thinking (CT) skills that are related to the students' CT competence level on the following six (6) computational thinking (CT) concepts: flow control, abstraction, user interactivity, synchronisation, parallelism and logic. With the aid of the Scrape tool and the Dr. Scratch evaluation rubric the overall CT score per project was calculated by adding up the partial scores of each CT concept. Projects with up to 6 points are considered to prove a Basic CT, while projects between 7 and 12 points are valued as Developing, and projects with more than 12 points are evaluated as Proficient [26].

(b) Confirm if the students followed the common best practices in programming. These practices concerned the avoidance of duplicated scripts (two programs formed by the same blocks and where only the parameters or values of the blocks vary), incorrect names (when the default names of the new characters were left e.g., Sprite1, Sprite2), dead code (parts of programs that are not executed) and not initialising attributes (when the objects' attributes are not correctly initialised).

(c) Confirm if the students deeply understood the CT programming concepts by using a wide variety of programming commands and avoiding leaving any "dead code".

(d) Confirm if the students have understood the connections among the body interaction, the space and the abstract representation of angles and geometry concepts, by adding complex gestures into the game-play.

Regarding the second research question (RQ2), the successful implementation of the proposed approach was examined by **evaluating the quality of the students' deliverables** (design artefacts and Kinect games). Design artefacts were evaluated only by the participated researcher (author of the study) using an evaluation rubric [24]. On the other hand, emphasis was given in the students' final deliverable (Kinect game). The final product of each group (Kinect game) was evaluated both by the researcher and groups of students, in order to a) answer the current question and b) to evaluate the completeness and clarity of the proposed 5-Likert evaluation form by comparing the participants' score (students' and researcher' final score).

Regarding the third research question (RQ3), the acceptance of the proposed approach was examined by evaluating via field observation and questionnaire:

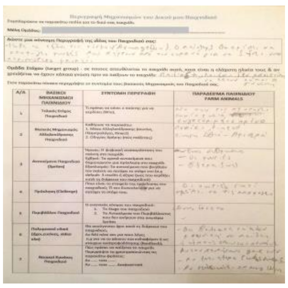
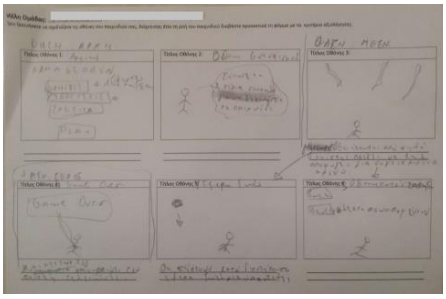
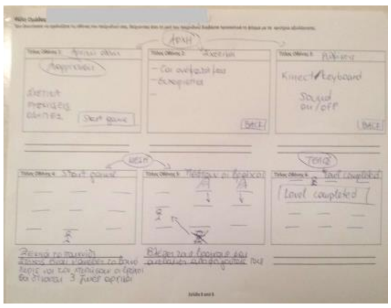
1. The positive feelings about the proposed approach.
2. The level of satisfaction for specific factors/components of the approach.
3. The thoughts about time management and difficulty level (open-ended questions).

5. Results

All eleven (11) groups managed to submit the required deliverables (design prototype and Scratch game) on time with zero dropouts. As an effect, eleven (11) Kinect games were created (1st school: 6 games and 2nd school: 5 games). Most of them (8/11, 73%) represented adaptations of real-world scenarios (e.g., climbing, boxing, catching, simulating a tennis game, pointing a target, imitating a variety of strange postures to pass wall obstacles), having also some unrealistic features. The other three games (3/11, 27%) were based on unrealistic situations or had heroes from comic books (Armageddon, Ice Age, SpongeBob). Each group of students chose a name for their game. Examples from the students' deliverables are presented in Table 4.

Regarding the initial Scratch activities in stage 0 (Introduction), 12/22 students completed these eight step-by-step activities and 9/22 students completed 6/8 activities mostly due to time limitation. In the last 2 activities, students had to create a functional game menu with three buttons and add a 4th button that would be used to select the difficulty level using multiple if statements.

Table 4. Examples from the students' game design and development deliverables.

Game Design Documents	Storyboards	Scratch Games
		
		

5.1. Were the Students' Thinking Skills Enhanced as a Result of the Approach? (RQ1)

The enhancement of thinking skills was examined by giving emphasis on the computational thinking skills (CT skills) and social skills.

(a) **Regarding the existence of the six (6) computational thinking (CT) concepts in students' games.**

As can be seen in Table 5, the concepts in which all projects obtained higher results are, abstraction, user interactivity and parallelism, while two projects got slightly lower values at synchronisation and logic. Most teams did not manage to get high value at flow control since they did not use commands like the "repeat-until". Regarding the final score, all games are evaluated as Proficient, as they gained more than 15 points in a rating scale from 1 to 18.

Table 5. CT score of the 11 students' Kinect games.

CT Programming Concepts	Basic	Developing	Proficient
Flow Control		7/11	4/11 games with "Repeat-Until" commands
Abstraction			11/11
User interactivity			11/11
Synchronisation		2/11	9/11 games with "Wait-Until" commands
Parallelism			11/11
Logic		2/11	9/11 games had more than one condition inside the If block.

(b) **Students followed the common best practices in programming:** As shown in Table 6, students managed to follow the proposed best practices and avoided common programming mistakes such as duplication of scripts, incorrect sprite names, dead code and not initialisation of sprites' attributes.

Table 6. Analysis of scripts for identifying best practices (results from 11 games).

Students should Avoid	Detected In
Duplicated scripts	9.06% (156/1721 scripts)
Incorrect sprite names	9.09% (33/363 sprites)
Dead code	0.35% (6/1721 scripts)
Not initialising sprites' attributes	1.10% (4/363 sprites)

(c) **Complexity of the produced games:** Quantitative analysis supported by the Scrape tool showed that students deeply understood the CT concepts, as they used a wide variety of programming commands (average number of programming commands per game: 966), scripts (average number of scripts per game: 156), sprites (average number of sprites per game: 33) and sprites' costumes (average number of costumes per game: 43). In addition, an average number of 13.09 previous versions per game were created by each group of students. Finally, according to the researcher's field notes, students used logic, flow control and abstraction techniques to express their thoughts, to debug/update their games and to make decisions about the proper body joints and gesture algorithms.

(d) **Understanding the connections among the body interaction, the space and the abstract representation of angles and geometry concepts, by creating complex gestures for the needs of user interaction:** The analysis of the NUI gestures that had been embedded into the games showed that 7/11 games (63.64%) included gestures that required the execution of 2 or 3 postures and 10/11 games (90.91%) were coded for tracking 4 to 9 different body joints. For example, students simulated the climbing gesture by tracking 7 body joints for better accuracy and having to perform 2 different body postures. Another group of students used 9 different body joints in their game. In addition, 7/11 games (63.64%) included more than one gesture (2 to 4 gestures).

Results about the students' social skills enhancement are provided in Figure 4. Students' answers indicated that working in groups helped them not only to produce a better game but also to learn how to cooperate, how to collaborate with their partner and how to participate in a role-playing presentation. Teachers also confirmed that students worked really well in groups, as they were collaborating to find the best solution for their game product. It was also mentioned by teachers that when the group members collaborated with other groups they provided useful feedback or there were changing ideas.

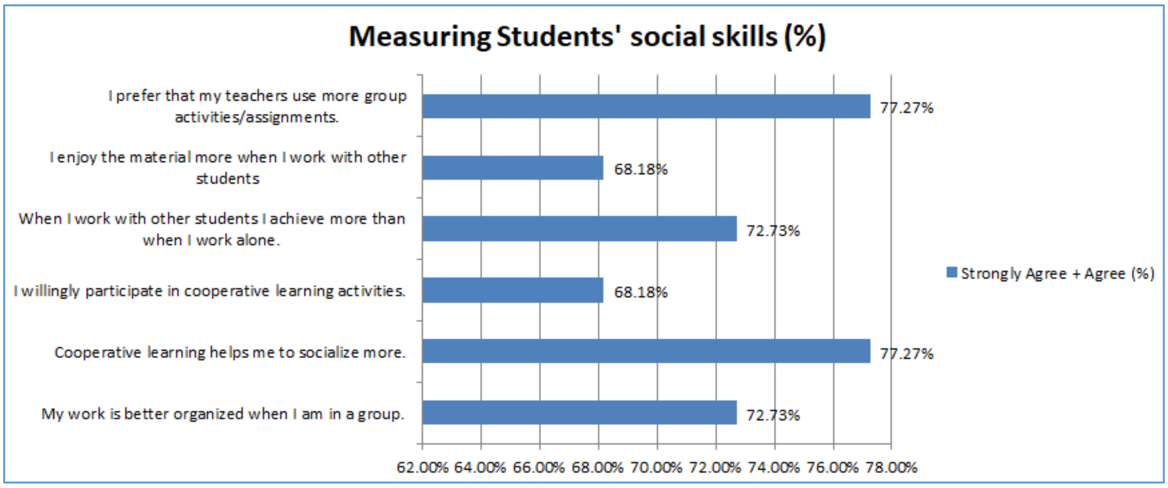


Figure 4. Measuring Students' Social Skills (Students' answers).

5.2. Was the Proposed Educational Approach Implemented Successfully and as Planned in the School Environments? (RQ2)

The successful implementation of the proposed approach was examined by evaluating the quality of the students' deliverables (design artefacts and Kinect games). Final versions of the students' design artefacts (storyboards and design documents) were evaluated using a design evaluation rubric from literature [26]. The results are provided in Table 7.

Table 7. Design Artefacts' Assessment Results.

	Design Artefacts' Score			Design Artefacts' Score (%)		
Evaluation Criteria for Storyboards and Design Documents	Excellent (A)	OK (B)	Weak (C)	Excellent (A)	OK (B)	Weak (C)
Design describes or shows the objective, purpose and scope	4/11	7/11	0/11	45.45%	36.36%	18.18%
Design describes or shows the reader who the intended audience is	10/11	1/11	0/11	90.91%	9.09%	0.00%
Design describes any media planned for the project	3/11	3/11	5/11	27.27%	72.73%	0.00%
Design has a logical organisation, effective structure and guide for the reader	4/11	5/11	2/11	45.45%	36.36%	18.18%
Design is succinct and clear	4/11	5/11	2/11	45.45%	36.36%	18.18%
Design depicts a clear ending	4/11	5/11	2/11	45.45%	36.36%	18.18%
Design has few mechanical defects	3/11	6/11	2/11	45.45%	36.36%	18.18%

According to the above results, most of the design prototypes (9/11, 82%) fulfilled the design requirements with some blanks (unclear design, general or vague description). On the other hand, 2/11 design artefacts (18%) had many mechanical defects (storyboards' structure was not clear, story arcs or turning points were not described and the design needed revision for clarity).

Also, the quality of the produced Kinect games was assessed by students and the participated researcher (author of the study) using a proposed 5-Likert evaluation form (Appendix A), which composed of thirteen (13) criteria (common design heuristics from literature). Results regarding the accomplishment of the evaluation forms' criteria are provided below in Figure 5.

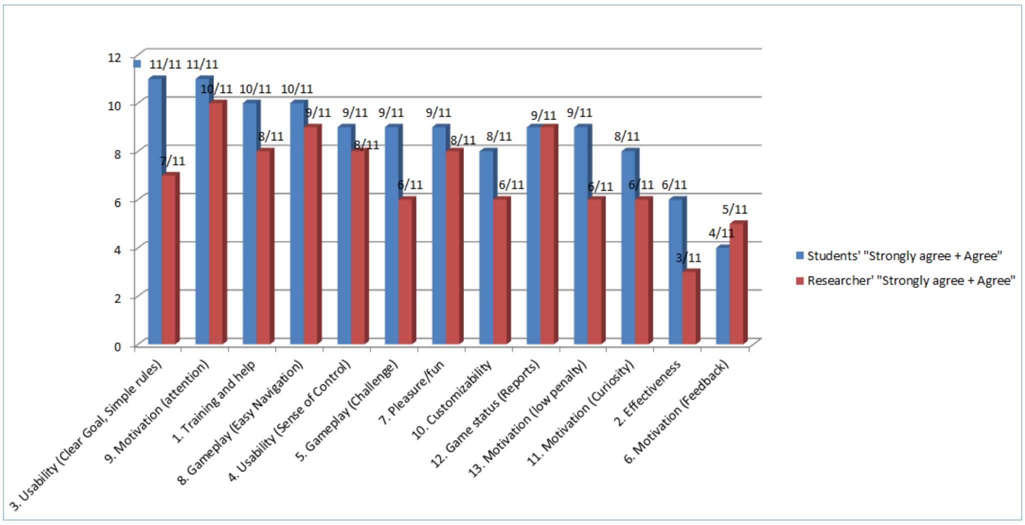


Figure 5. Results from the accomplishment of the evaluation forms' criteria.

According to the data shown in Figure 5, the produced games scored high for their usability. High scores were also given for some crucial criteria such as (i) keeping the users' attention during gameplay, (ii) the provided instructions (training and help) and (iii) the ability to easily navigate in the gaming environment. Moreover, due to some functionality problems on playtesting, the criteria for effectiveness and feedback received a lower score.

5.3. Was the Proposed Educational Approach Appreciated by the Participants? (RQ3)

The acceptance of the proposed approach was examined by evaluating:

1. The positive feelings for the proposed process.
2. The level of satisfaction for specific components of the proposed approach.
3. The thoughts about time management and difficulty level (open-ended questions).

1. **Positive feelings for the proposed process:** Based on the students' answers on the final questionnaire: (i) 96.56% of them liked the project, (ii) 72.41% of them answered that they were motivated by the fact that they had to create a Kinect game and (iii) 93.10% of them would encourage their schoolmates to participate in similar initiatives. The teachers' answers were also consistent with the students' answers (both the teachers selected the "strongly agree" option).
2. **Level of satisfaction for specific factors/components of the approach** (strongly agree and agree): Even though students and teachers had no previous experience on such kind of projects, the majority of the students were satisfied with:

    - The learning resources, the project duration and the project structure (69%).
    - The guidance given by the CS teachers and the classmates (75.86%).
    - The support and encouragement offered by the principal researcher (93%).

    Regarding the game-making process, 75.81% of students strongly agreed and agreed that:

    - They gained a good understanding of concepts/principles in this field.
    - They learned to apply the principles of this program (follow specific steps and practices), in order to create their own Kinect game in a systematic process.

    Both the teachers were also satisfied with the above factors/components of the approach.

3. **Their thoughts about time management and difficulty level**

According to the teachers' answers, they were willing to extend the duration of the project, mainly focusing on the development phase. Moreover, according to the students' answers, the extra average time a student spent at home during the 8-week project was a total of 8.36 h (min: 2.5 and max: 14). Most of the students (15/22, 68.18%) mentioned that they could have dedicated "More time" in order to upgrade their games, to fix software bugs or to continue the fun learning activity by building a better version of the game. The aforementioned positive results were also confirmed by the students' and teachers' comments about their experience. Extracts of these comments are presented in Table 8.

Table 8. Students' and Teachers' general comments.

Students' thoughts about the Program	Teachers' thoughts about the Program
"It was nice. We tried something new and different!"	1st teacher comments: *"On the current analytical and clock program (1 h per week for CS Course) the approach seems to enhance the students' programming skills and motivate them not only because they used Kinect camera but also due to the "game" part. It worked really well as an initial step for students who are interested in Computer Science and the future use of more advanced authoring tools. Through those kinds of projects, students understand the development stages, the required distribution of work, the process and the results of playtesting. It was a constructive and appreciable experience."* 2nd teacher comments: *"Pleasant, productive and educational process. It certainly won the students' interest."*
"Outstanding experience and I'll do that again."	
"It was a very useful learning program."	
"I'm very pleased because I've learned a lot about the Kinect camera and new potentials of scratch." *"I gained experience from this Kinect process and I'm feeling more confident in building any kind of game."*	
"It was a beautiful experience due to the fact that we learned how to follow a process for building Kinect games, we cooperated and we interacted with our classmates."	
"I'm very pleased because I've learned a lot about the games' functionality and the required scripts for programming."	
"A beautiful experience. I believe that I improved my CT skills."	
"The program motivates someone to work even more in this area."	
"I've learned how a game is working using variables, blocks, loops etc. *and at the same time it was fun and interesting."*	

6. Discussion-Conclusions

It is well-documented in the literature that there is a need to develop new teaching methods for helping young children, acquiring computation thinking and programming skills [30]. The current paper presents a new teaching approach that advocates the systemic game-making of Kinect motion-based touchless games that can lead to the enhancement of the students' thinking skills. Students understood that designing a game means solving an ill-structured problem via an iterative and step-wise process using a variety of thinking and behavioural skills. The results of a study from two junior high schools indicated that students strengthened their computational thinking skills as they have managed to develop good quality games by applying complex programming commands/concepts in an effective way. These results are comparable with the results of other studies that measured the potential of rapid digital game creation as a way to teach computational thinking as part of programming courses [31,32]. Like other studies related to game-design [33,34], the present study indicated that the proposed group-based game-making approach helped in the enhancement of the students' social skills (cooperation, collaboration). Finally, it was clear that the iterative process of designing and creating a Kinect motion-based game was engaging and highly motivating. Students showed enthusiasm and mentioned that they wanted to repeat it or propose it to their classmates. Also, by asking students to develop medium fidelity of these types of games, students succeeded in improving their spatial thinking, an important predictor of achievement in STEM, and managed to understand the connections among the body interaction, the space and the abstract representation of angles and geometry concepts. More evaluation studies of the application of this innovative approach, which could help in generalising the findings of the present study, are underway.

Acknowledgments: There were no sources of funding. We gratefully acknowledge both Computer Science teachers (Mr Otzakoglou Thodoris & Mrs Kokkori Athina) and students from the participating schools for their active participation during the implementation of this study and their feedback comments. Special thanks to Mr. Kavroulakis Antonis, M.Sc. student, for his support during the preparation and implementation of stage 1 of this study.

Author Contributions: Ioannis Altanis is a Phd candidate. This work was part of his research. Ioannis Altanis designed the proposed method, performed the experiments, analyzed the data and wrote the paper in consultation with Symeon Retalis (professor and supervisor of the proposed research). Professor Symeon Retalis and Dr. Ourania Petropoulou (Postdoctoral researcher) contributed to the design of the experimental procedures and the revisions of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

GAME EVALUATION FORM (Game Title): ……………………………………………………

Team Evaluators: ……………………………………………………………………

A/A	Criteria & Short Decription	Level of Agreement					Comments (optional)
		Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree	
1	**Training and help:** The game provides specific pre/in-game instructions						
2	**Effectiveness:** The game's feature is playable and operating well						
3	**Usability:** I'm aware at any time the actions should proceed in the game, because the game fulfill the following characteristics: • Unique & Clear Goal, Simple and few rules, Consistency in gameworld, Clear audio & graphics						
4	**Usability (Sense of Control):** The game is easy to use and fulfill the following characteristics: • Sense of Control, Minimum number of Controls, No need for precise positioning and aiming, Game elements were moving as slowly as needed to allow the player time to react						
5	**Gameplay (Challenge):** The difficulty level of the game increases as time passes (progressive difficulty), in order to give you the proper satisfaction						
6	**Motivation (Feedback):** The game provides positive/negative feedback (eg rewarding by audio/visual messages or by gaining/loosing lives & time)						
7	**Pleasure/fun:** I was having fun playing this game						
8	**Gameplay (Easy Navigation):** The game world was simple and visible enough, reducing the need for carefully timed actions, in order for me to navigate easily						
9	**Motivation (attention):** The game managed to keep our attention during gameplay						
10	**Customizability:** The game provides a satisfying level of configuration/customization in order to adapt to player's needs (the ability to customize video and audio settings, difficulty or game speed)						
11	**Motivation (Curiosity):** The game enhances our curiosity in order for us to keep playing						
12	**Game status (Reports):** The game provides clear and proper information/reports for player's status						
13	**Motivation (low penalty):** The low penalty for errors motivates us to play this game again, in order to improve our score						
14	**If you would change something in the game, what would it be;**						
15	**What did you like the most in this game;**						

Figure A1. 5-Likert scale games' evaluation form.

References

1. Hava, K.; Cakir, H. A systematic review of literature on students as educational computer game designers. In Proceedings of the EdMedia: World Conference on Educational Media and Technology, Waynesville, NC, USA, 20 June 2017; Johnston, J., Ed.; Association for the Advancement of Computing in Education (AACE): Washington, DC, USA, 2017.
2. Kafai, Y.B.; Burke, Q. Constructionist gaming: Understanding the benefits of making games for learning. *Educ. Psychol.* **2015**, *50*, 313–334. [CrossRef] [PubMed]
3. Ke, F. An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Comput. Educ.* **2014**, *73*, 26–39. [CrossRef]
4. Denner, J.; Werner, L.; Ortiz, E. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Comput. Educ.* **2012**, *58*, 240–249. [CrossRef]
5. Grover, S.; Basu, S. Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. In Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March 2017.
6. Hsu, H.J. The potential of kinect in education. *Int. J. Inf. Educ. Technol.* **2011**, *1*, 365–370. [CrossRef]
7. Kinesthetic Learning (Wikipedia). Available online: https://en.wikipedia.org/wiki/Kinesthetic_learning (accessed on 10 November 2017).

8. Spatial Ability (Wikipedia). Available online: https://en.wikipedia.org/wiki/Spatial_ability (accessed on 10 November 2017).
9. Wai, J.; Lubinski, D.; Benbow, P.C. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *J. Educ. Psychol.* **2009**, *101*, 817–835. [CrossRef]
10. Scratch MIT. Available online: http://scratch.mit.edu/ (accessed on 24 November 2016).
11. Howell, S. Kinect2Scratch (Version 2.5) [Computer Software]. Available online: http://scratch.saorog.com (accessed on 24 November 2016).
12. Aleem, S.; Capretz, L.; Ahmed, F. Game development software engineering process life cycle: A systematic review. *J. Softw. Eng. Res. Dev.* **2016**, *4*, 1–30. [CrossRef]
13. GameStar Mechanic (GSM)-Getting-Started-Teacher Pack 2011. Available online: https://sites.google.com/a/elinemedia.com/gsmlearningguide/getting-started-teacher-pack (accessed on 30 November 2016).
14. Bass, K.M.; Dahl, I.H.; Panahandeh, S. Designing the game: How a project-based media production program approaches. steam career readiness for underrepresented young adults. *J. Sci. Educ. Technol.* **2016**, *25*, 1009–1024. [CrossRef]
15. Schell, J. *The Art of Game Design: A Book of Lenses*; Kaufmann, M., Ed.; Elsevier: Amsterdam, The Netherlands, 2008.
16. The 35 Gamification Mechanics Toolkit v2.0. Available online: http://www.epicwinblog.net/2013/10/the-35-gamification-mechanics-toolkit.html (accessed on 5 November 2016).
17. Mueler, F.; Gibbs, M.R.; Vetere, F.; Edge, D. Supporting the creative game design process with exertion cards. In *Proceedings of the CHI '14 Conference on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2014.
18. Hornecker, E. Creative idea exploration within the structure of a guiding framework: The card brainstorming game. In Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction, Cambridge, MA, USA, 24–27 January 2010.
19. Ke, F.; Im, T. A case study on collective cognition and operation in team-based computer game design by middle-school children. *Int. J. Technol. Design Educ.* **2014**, *24*, 187–201. [CrossRef]
20. Baytak, A.; Land, S.M. An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educ. Technol. Res. Dev.* **2011**, *59*, 765–782. [CrossRef]
21. Mouza, C.; Pan, Y.; Pollock, L.; Atlas, J.; Harvey, T. Partner4CS: Bringing computational thinking to middle school through game design. In Proceedings of the Stanford University FabLearn Conference on Creativity and Fabrication, Palo Alto, CA, USA, 25–26 October 2014.
22. Kafai, Y.; Vasudevan, V. Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers. In Proceedings of the Workshop in Primary and Secondary Computing Education, London, UK, 9–11 November 2015.
23. Feng, C.Y.; Chen, M.P. The effects of goal specificity and scaffolding on programming performance and self-regulation in game design. *Br. J. Educ. Technol.* **2014**, *45*, 285–302. [CrossRef]
24. Evaluation Criteria—Storyboards or Design Documents. Available online: http://www.write4web.com/wp-content/uploads/2011/10/My.Rubric.storyboard_design.pdf (accessed on (accessed on 24 November 2016).
25. Dr. Scratch (Analyze your Scratch projects here!). Available online: http://www.drscratch.org/ (accessed on 10 May 2016).
26. Moreno-León, J.; Robles, G. *Analyze Your Scratch Projects with Dr. Scratch and Assess Your Computational Thinking Skills*; AMS: Amsterdam, The Netherlands, 2015.
27. Happy Analysing (Only Works with Scratch 1.x). Available online: http://happyanalyzing.com/ (accessed on 12 September 2016).
28. Gravestock, P.; Gregor-Greenleaf, E. *Student Course Evaluations: Research, Models and Trends*; Higher Education Quality Council of Ontario: Toronto, ON, USA, 2008; Available online: http://www.heqco.ca/en-CA/Research/Research%20Publications/Pages/Home.aspx (accessed on 10 December 2016).
29. Farzaneh, N.; Nejadansari, D. Students' attitude towards using cooperative learning for teaching reading comprehension. *Theory Prac. Lang. Stud.* **2014**, *4*, 287–292. [CrossRef]
30. Leal, A.; Ferreira, D. Learning programming patterns using games. *Int. J. Inf. Commun. Technol. Educ.* **2014**, *12*, 23–34. [CrossRef]
31. Kuruvada, P.; Asamoah, D.A.; Dalal, N.; Kak, S. The Use of Rapid Digital Game Creation to Learn Computational Thinking. 2010. Available online: http://arxiv.org/pdf/1011.4093.pdf (accessed on 12 November 2017).

32. Wu, M.L.; Richards, K. Facilitating computational thinking through game design. In *International Conference on Technologies for E-Learning and Digital Entertainment*; Springer: Heidelberg/Berlin, Germany, 2011.
33. Scaffidi, C.; Chambers, C. Skill progression demonstrated by users in the scratch animation environment. *Int. J. Hum.-Comput. Interact.* **2012**. [CrossRef]
34. Williams, L.; Wiebe, E.; Yang, K.; Ferzli, M.; Miller, C. In support of pair programming in the introductory computer science course. *Comput. Sci. Educ.* **2002**, *12*, 197–212. [CrossRef]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).