A New Game-based Strategy for Enhancing Youth Programming Skills

Ataollah Zarei[1], Khairiyah Mohd-Yusof[2], Hoshang Kolivand[3], Arash Ahmadi[1], Yousef Al-Hamar[3]

[1] Department of Electrical and Computer Engineering, University of Windsor, Canada
[2] Centre for Engineering Education, Universiti Teknologi Malaysia, Malaysia
[3] Department of Computer Science, Liverpool John Moores University, UK
Zarei.Ataollah@uwindsor.ca, khairiyah@cheme.utm.my, h.kolivand@ljmu.ac.uk,
Arash.Ahmadi@uwindsor.ca, y.k.alhamar@2016.ljmu.ac.uk

Abstract

Teaching computer programming to secondary school students has continued to further attract the researchers' and practitioners' attention in recent years. These young learners do not have enough computer programming knowledge and skills while they continue to waste huge amount of time for playing video games. One way to cope both problems is redirecting students' energy from playing video games to learning video game programming. This qualitative study proposes a solution based on game-based instruction and learning by design approaches to address the aforementioned problems. The research method is Design-Based Research. Qualitative data was collected from the students in an Iranian secondary school and their parents. The main output of this study is in the form of instructional design principles for teaching programming to secondary school students. The problems caused by the students' excessive video game playing, the impact of applying the mentioned principles on spending time on video games, the students' learning quality, and their satisfaction are also taken into account.

Keywords: Computer science education, game-based instruction, learning by design, STEM education, video gaming

1 Introduction

Teaching computer programming, or coding, in secondary schools has attracted researchers' and practitioners' attention more recently due to the progressive dependency of human life on information and communication technology and lack of human resources to cover industries' needs. For example, computer science and programming education are continuously expanding in the U.S. [1]. This trend can also be seen in other countries such as China and South Korea [2-3]. However, in some countries such as Iran, computer programming has not been considered in the secondary school curriculum. In the official curriculum followed by Iranian secondary schools, there is a subject related to information and communication technology. This subject aims to give the students the ability to operate computers at a fundamental level such as word processing, required by many academic courses and employment opportunities; however, there is no specific section for programming.

Teaching computer programming to students, even in universities, is quite challenging [4]. According to [5], limited students' motivation, discipline difficulty, and course arrangement complexity caused a dramatic failure for students in their first computer programming course. The issue of students' motivation for learning programming is also considered in several research studies [6-8] however, in these studies, computer programming was a part of the school curriculum which is totally different from the context of this study. This means that teaching this subject and convincing Iranian secondary school students to learn programming are more challenging.

Many students spend a lot of time playing video games on their PC, mobile phone or games console [9]. This habit can cause problems such as video game addiction and reduction of their learning quality for students [10-11]. Some researchers believe that parents and teachers should use this potential to improve this generation of gamers' learning achievements [12-14]. The idea of taking students' interest in video games as an opportunity for teaching and learning video game programming is supported by some motivational theories and models for education such as constructivism learning paradigm, Maslow's hierarchy of needs theory and Keller's ARCS model.

According to constructivism learning paradigm, learners construct their knowledge actively through linking new information with their previous knowledge. Engaging students in the learning process is usually challenging. Therefore, many educators offer different strategies and methods such as game-based learning [15-16].

*Corresponding Author: Ataollah Zarei; E-mail: Zarei.Ataollah@uwindsor.ca
DOI: 10.3966/160792642020092105006

Based on Maslow's theory, individual needs are the main source of motivation. They can be categorized in different types including biological and physiological needs, safety needs, love and belongingness needs, esteem needs, cognitive needs, aesthetic needs, and self-actualization needs. Esteem needs, as one of these categories, refers to the individual's desire for reputation or respect from others [17].

According to the ARCS model, there are four key factors to encourage and sustain learners' motivation: attention, relevance, confidence, and satisfaction [18]. In fact, learning video game programming can attract students' attention due to their passion for video gaming. It is relevant for the students because there is a strong link between video gaming and the real life of the students due to the considerable time spent on video gaming. To increase students' confidence, the course teacher should present the course objectives and clearly tell students that at the end of the course, they will be able to develop at least one video game. Satisfaction is the students' feeling at the end of the course. It can be increased through several ways such as quality teaching, praise from a higher-up and appropriate feedbacks.

Due to the limited number of research studies that explicitly focus on game-based programming instructional design and its impact on students' time spent playing video games, the aim of this study is to investigate a solution to redirect students' energy from playing video games to learn programming and designing such games. The definition of instructional design suggested by [19] in which instructional design is the practice of creating "instructional experiences which make the acquisition of knowledge and skill more efficient, effective, and appealing". From the aforementioned definition, the research aim can be redefined as providing instructional design principles for teaching programming through games to achieve the objectives. Therefore, the research question is what instructional design principles should be followed to direct students' energy away from playing video games to learn programming?

2 Method

2.1 Research Design

To design and develop teaching and learning activities enhanced by educational technologies, instructional designers and instructors should follow proper instructional design principles. Design-Based Research (DBR) method, which makes the research objectives achievable, was followed in this study because it is a flexible and iterative method to improve and fit the existing design principles with the context [18]. The output of this method is several instructional principles that have solved the practical problems in the context. Although these principles should not be generalized, the study results can be replicated [21].

In the DBR process for educational technology, the practical problem should be analyzed first by the designers, who are also the researchers at the same time, in collaboration with the practitioners. Then a solution based on existing principles and technological innovations should be developed. Next, the solution should be tested and refined through an iterative cycle in practice to achieve a proper solution. Finally, the new design principles will be developed based on the analyzed data and the researchers' reflection during and after the research process [22].

As seen in Figure 1, firstly, students' problems caused by excessive video game playing and their intention to learn were studied through collecting data from the secondary school students, and their parents. Then, an instructional principles set was prepared based on existing principles and game-based instruction solutions. The third phase was implemented in two rounds. In the first round, video game programming was taught based on the prepared principles for 3 months in 12 sessions. The impact of the teaching was studied by collecting data from the students and their parents. Based on the study results, the instructional principles were revised. In the second round, the video game programming course was taught to another group of the students for 3 months in 12 sessions.

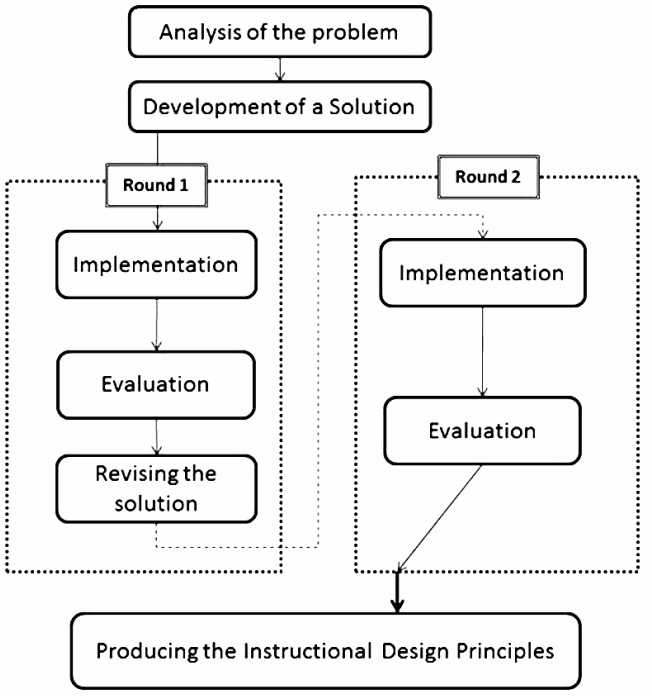


Figure 1. The research design framework

2.2 Context and Participants

The research data was collected from students at an Iranian boys' secondary school aged 12 to 16 years and their parents in Malaysia. To analyze the practical problem, the required data was collected from 37

participants through an open-ended questionnaire completed by 25 students, and an informal interview with 12 parents. The programming classes were held on weekends with 8 students in the first round and 16 students in the second round of the programming course. The participants for the problem analysis phase were selected through the snowball sampling method [22]. The participants for the two rounds of implementation and delivery including 8 students and their parents for the first phase, and 16 students and their parents for the second phase were selected through homogenous sampling method [22]. The data was collected from all students who participated in the programming courses along with their parents.

2.3 Data Collection Instruments

The qualitative data collection instruments which were used for the problem analysis and evaluation of the results of the treatment are described in the following subsections. The validity of the mentioned questionnaires in 2.3.1 and 2.3.5, interview protocols in 2.3.2 and 2.3.6, observation protocol in 2.3.4 and performance test in 2.3.3 sections was checked and confirmed by two experts in the field of engineering education and computer science.

2.3.1 Problem Analysis Open-ended Questionnaire

This questionnaire was designed to collect initial data including time spent for playing video games, perception toward video game playing, intention to learn game programming and preferred programming type from the students to analyze the problem raised by excessive video game playing and their intention to learn game programming.

2.3.2 Problem Analysis Interview

An informal interview with the parent of the students was done to understand the problems caused by the students' compulsive video gaming. They were asked about time spent by their children for playing video games, the impact of video gaming on their children, and their opinion about redirecting the students' energy and passion from playing video games to developing video games.

2.3.3 Learning Performance Test

The students' learning was evaluated through the pre and post-test method. Due to using different instructional designs in round 1 and round 2, the instructor was able to cover more topics in round 2. Therefore, different performance tests were designed and used for each round. These performance tests are explained in details in sections 3.3.2.1 and 3.3.5.1. The validity of these tests was confirmed by two experienced computer programming lecturers.

2.3.4 Improvement Observation

The required data and researchers' reflections were recorded in the field notes [22] during the teaching sessions. In fact, descriptive and reflective field notes were recorded for each student on a separate sheet by the instructor. To improve the quality of collected data, an observation protocol was prepared and used. The students' learning activities such as problem posing, and answering to the questions raised by the instructor or their classmates were recorded on the sheets. Then, the instructor's evaluations and reflections on the students' learning activities were added separately in the next column beside the related activity.

2.3.5 Self-evaluation Open-ended Questionnaire

The students' time spent playing video games, their perception toward the complexity of the video games which they were playing, time spent for programming, their view of game programming, their difficulties in learning programming, and their recommendations for improving the quality of the teaching methods and the learning materials were all collected through this instrument.

2.3.6 Parents' Feedbacks Informal Interview

To collect the parents' feedback about their children's video gaming and their improvement in the game programming, the researchers carried out informal interviews. The time spent playing video games and doing programming by their children, their perception about game programming, and their own recommendations for improvement were collected through this instrument. To improve the quality of data analysis, the interviews were recorded and transcribed [22].

2.4 Data Analysis Method

In this study, the general method for qualitative content analysis was followed for analyzing the research data and NVIVO 8 tool was used to facilitate the data analysis process. According to [22], the data analysis process for qualitative data includes: initially reading through text data, dividing the text into segments of information, labeling the segments of information with codes, reducing overlap and redundancy of codes, and finally collapsing codes into themes.

To increase the validity of the results of this qualitative study, different types of triangulation including data triangulation which means collecting data from different sources such as students and parents, methodological triangulation which means using different methods of data collection such as open-ended questionnaire, observation and interview, and theory triangulation which means using and

comparing the results with existing theories such as Maslow's hierarchy of needs theory and Keller's ARCS model were used [20, 23].

3 Research Process and Findings

Regarding the research design, the main stages of the study include analysis of the problem, designing a solution, two rounds of implementation, evaluation, and refinement which enables the researchers to produce the instructional design principles. These stages are explained in the following subsections.

3.1 Analyzing the Problem

Figure 2 shows the students' spent time for video gaming during weekdays and weekends. It reveals that the students usually spent a lot of time on playing video games. In fact, most of them played more than 5 hours per day on weekends and 2 hours per day on weekdays. Student 9 said: "I play video games by my tablet as much as I can on holidays. I give up only for lunch and dinner" and student 3 stated: "I like video games. You know, it's cool. I play Clash of Clans game by my smartphone

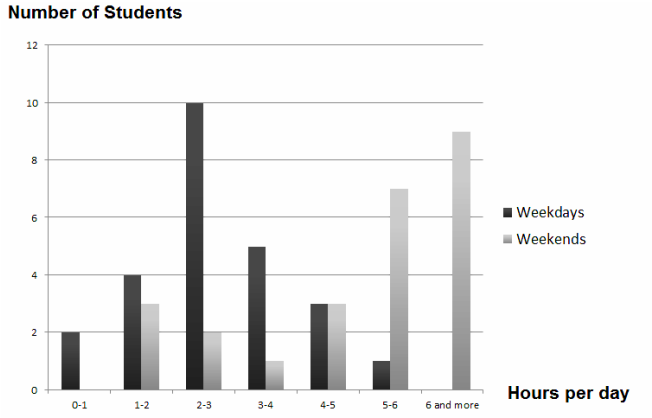


Figure 2. Amount of time spent playing video games during weekdays and weekends

everyday after finishing my assignments". All students had their own smartphones or/and tablets. They were used to using their devices mainly for gaming.

The result of analyzing collected data from the students through the problem analysis questionnaire and informal interview with the parents is shown in Table 1. It shows that students' view of excessive video game playing is different from their parents'. However, both groups embraced teaching game programming to the students.

Table 1. Result of the problem analysis

Core theme	Sub-theme	Percentage of Proponents	
		Student	Parent
Problems of excessive video gaming	Time wasting	12	91.67
	Source of challenge	80	83.33
	Academic failure	60	100.00
Opportunity of students' interest to video gaming	Learning Computer Programming	88	91.67

The Potential of Teaching Game Programming for Redirecting Students' Energy

Students' background knowledge and skills for developing easy computer or mobile games are not satisfactory, because it is not considered in their curriculum and nobody taught it to them previously. According to Table 1, teaching game programming to the students is an appropriate idea and sufficiently attractive to convince students to learn programming. The students were usually interested in this idea and asked the researchers if they could start soon. Student 8 said: "I have many ideas for providing new games because I play different games a lot" and Student 17 stated: "I would definitely register to the course because I like to develop my own game". The parents supported this idea, too. For example, parent 4 stated: "learning programming will help children to find better jobs in the future".

In addition, 82 percent of the students preferred to learn mobile game programming on computer game programming. They claimed different reasons to support their idea. For example, students 21 said: "I prefer to learn mobile game programming because smartphones have more features for gaming than desktop computers" and student 13 said: "I would like to learn mobile game development because all people play on their smartphones. If I create my own mobile game, I will have more potential customers to buy it."

3.2 Development of a Solution

To solve the described problems, two approaches are followed based on the results of the analysis stage. These approaches are:

- Teaching programming to the secondary school students should start by teaching mobile game programming.
- Learning by design should be followed to design teaching activities for each session.

The first approach is supported by the result of the analysis stage in which the students preferred to learn mobile game programming than computer game programming. The learning by design instructional

principle, as the second approach, was chosen as the basis for instruction because it is a proper instructional principle for teaching programming through designing mobile games. Kolodner et al. [24] proposed the following steps as the instructional principles based on learning by design approach to use in class:

(a) Start with a design challenge.

(b) Provide an opportunity for students to think and develop their ideas.

(c) Encourage and help students to improve their ideas in groups and identify what they need to learn to progress and address the design challenge.

(d) Provide an opportunity for the students to implement their ideas and solutions.

(e) Encourage students to reflect and evaluate what they have learned, the future applications of this knowledge, and any difficulties they had with the learning process.

(f) Return to step "c" if required.

These principles were considered as the first version of the design principles for teaching game programming to secondary school students which were improved in the research process. Choosing a programming language is very important for teaching programming. Regarding the abundance of Android devices compared to iOS devices, Java programming language on Android Studio IDE was chosen to be taught.

3.3 Iterative Cycle of Test and Refinement

The process of implementation and evaluation was run in two rounds. The process of implementation of the first version of the instructional design principles in practice and analysis of the evaluation results are explained in detail in the following subsections.

3.3.1 Description of the First-round Implementation

In this step, 8 students between ages 13 and 15 years, who play video games for more than 3 hours per day on average participated in the programming course. The course aim was learning the basic concepts of Java programming in Android Studio through teaching the process of designing and programming a paddle and ball game by one of the researchers during two months in 8 sessions. Each session took 4 hours. In the first two sessions, the concept of programming was explained. The preparation of the IDE including JDK and Android Studio was taught. Setting up an Android device in different ways was practiced. During the next six sessions, variables and data types, operators, conditional statements, arrays and loops, algorithms and functions were taught.

For each session, teaching activities were designed based on the design-based instructional principles [24]. For example, to teach the concept of variables by following the first version of the instructional design principle, the following activities were done during part of session 3 of the class:

(a) Starting with the question: "How can you check if the ball hits the paddle?"

(b) Students had time to think then discuss with peers.

(c) Students were then given the answer to the question if they had not come to the correct conclusion themselves.

(d) Ask students to change the location of the ball by changing its x and y in the code which was delivered at the beginning of the class.

(e) Ask students to reflect on the concept of variables and write their perception toward variables. The instructor can provide a new challenge for preparing the students to learn the next concept (data types).

3.3.2 Evaluation Results of the First Round

To evaluate the quality of the course, the students' learning quality, the impact of this course on spending time video gaming and their satisfaction of the course were studied.

Learning quality. Students' learning was evaluated through the pre and post-test method. A performance test including 20 objective questions was designed to investigate students' learning levels for each topic. It is important to note that higher-order cognitive levels including evaluation and creating was evaluate through the teacher's observation of students' activities in the class. Table 2 shows the number of questions that were designed for each topic, the corresponding learning level, and percentage of correct answers for each question for pre and post-test. Although most of the students understand all topics, their results for applying and analyzing levels, especially for some topics such as function, operators, and variables were unsatisfactory.

According to the researcher's observations and field notes taken from the students' class activities during teaching hours, the students could not design proper algorithms to create new games or add new behaviors to the game objects to change their behavior. For example, the researcher asked them to provide a function to sense ball lost and do the required changes in the corresponding variables. But they did not have any idea how to accomplish this. After explaining the idea and algorithm, it was still difficult for most of them to convert the algorithm to code.

Table 2. Average of students' score for each question in pre and post-test assessment

No	Topic	Learning Level	Percentage of correct answers	
			Pre-test	Post-test
1	Basic concepts of programming	Understanding	37.5	100
2	Basic concepts of programming	Understanding	25	87.5
3	Data types and variables	Understanding	12.5	75
4	Data types and variables	Applying	0	62.5
5	Data types and variables	Understanding	12.5	87.5
6	Data types and variables	Applying	12.5	50
7	operators	Understanding	25	87.5
8	operators	Applying	12.5	75
9	operators	Applying	0	62.5
10	operators	Analyzing	0	50
11	conditional statements	Understanding	25	87.5
12	conditional statements	Applying	25	75
13	conditional statements	Analyzing	0	62.5
14	arrays and loops	Understanding	12.5	75
15	arrays and loops	Applying	0	37.5
16	arrays and loops	Applying	0	50
17	arrays and loops	Analyzing	0	50
18	function	Understanding	0	62.5
19	function	Applying	0	37.5
20	function	Applying	0	50

Impact of the Solution on Students' Spending Time on Playing Video Games. To study the impact of the programming on students' time spent on video gaming, the required data was collected from different sources. According to the data collected through the students' improvement self-evaluation open-ended questionnaire, the average time spent on video games was reduced by 5-6 hours per week. According to the parents' feedback, all of them claimed that there was no considerable difference in the time spent gaming by their children before and during the course.

Satisfaction. The results of the qualitative data analysis from the students' self-evaluation open-ended questionnaire is shown in Table 3. The results revealed that choosing Java on Android studio IDE as the first programming language for developing video games was not a proper strategy. Because the students believed Java programming language is difficult to learn since it is full of details, very sensitive, existing online resources are usually boring or too complex. Finally, all students confirmed that mobile game programming is more difficult and time-consuming than their expectations. The parents' feedback was generally positive, however, some of them reported a concern related to students' motivation. For example, parent 1 stated: "my son's interest in programming was reduced because making a mobile game is a little bit difficult and very time consuming".

Table 3. Students' satisfaction evaluation result

Satisfaction Evaluation item	Sub-items	Students' perception	Proponent percentage
Course content	Programming language (Java)	Difficult to learn	100
		Sensitive IDE	100
		Free to use	100
	Learning resources (e.g. The Java Tutorials and YouTube videos)	Complex	87.5
		Boring	87.5
		Easy to access	75
Teaching method	Lecture	Clear explanation	75
	Learning activities (in and off class)	Complex	75
		Boring	62.5
General evaluation		More complex than expectation	100

3.3.3 Revising the Solution of the First Round

According to DBR research process, the first version of the design principles should be revised based on the evaluation results of the first round. Table 4 shows the main problems and weak points of the first round and the recommended solutions. Choosing the first programming language for teaching and learning coding can be quite challenging because there is no specific rule for choosing the most suitable one [25].

Table 4. The researchers' solutions to improve the first version of instructional design principles

Problem	Solution
Java coding was difficult to learn	Visual Programming Language should be used
Sensitive IDE	Block-based programming language should be used
Learning resources was boring (text-based)	Game-based examples Supporting community
Boring learning activities	Start from an easy game Follow learning by design method and try to teach developing a new game in each session
Difficulties in providing new algorithms	Game-based content and method Learning by design method should be followed
More difficult and time-consuming than students' expectation	Teach a programming language which enables the students to develop a game faster and easier

Based on the proposed solutions in Table 4 for selecting an appropriate programming language for teaching coding, Scratch, which was provided and supported by the Massachusetts Institute of Technology (MIT) Media Lab Lifelong Kindergarten Group, was chosen [26]. The Scratch environment is visual and block-based so the students can understand it easily. Each type of programming block in Scratch has a specific shape that can integrate only with other convenient blocks. This ability prevents students from making syntax or semantic mistakes when they are coding. Furthermore, there are a lot of samples and supporting documents in the scratch community which can support students' online learning.

3.3.4 Description of the Second-round Implementation

In the second round, another set of 16 students between ages 13 and 15 years who play video games more than 3 hours per day on average were taught to design and program some video games including the paddle and ball game through Scratch by the same instructor during a two-month period (32 hours). To make the results of the second round comparable to the first and regarding the similarities of math with programming logic, these participants were chosen from students with an average math score close to the previous group's average, (3.2 vs 3.4). It is noteworthy that there is a lot of differences between Scratch and Java in structure and logic. For example, many complexities such as data types are hidden from users in Scratch. Therefore, it was not efficient to follow the course syllabus from Round 1. The Scratch course syllabus included two introductory sessions: computer programming basic concepts, algorithms and introducing the scratch environment.

Regarding the students' difficulties in designing proper algorithms, the researcher taught the concept of algorithm and problem-solving in the first session through an unplugged activity. The "My Robotic Friends" game, which was designed by ThinkerSmith for teaching algorithms, was chosen to achieve this objective because the concepts of algorithm and coding were embedded in this activity, it is interesting and easy to implement this in the class, and the students can do this activity individually or in a group. In this game, the students write required instructions to arrange the cup as shown in Figure 3.

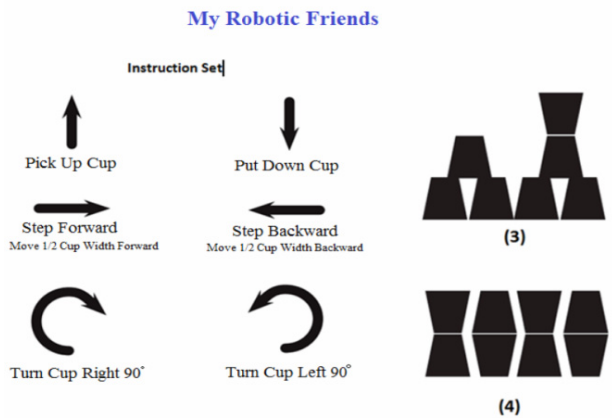


Figure 3. My Robotic Friends game, taken from ThinkerSmith, licensed under CC BY 2.0

To check the correctness of the instructions, one of the students plays the role of a robot and arranges the cups by running the instructions of another student.

The next six sessions were devoted to teaching some basic programming concepts such as motion, loops, conditional statements and variables, events and sensing, graphics, sound and animation, messaging and block, clone, and strings, by designing and developing three different videos games. For example, to teach the messaging concept through designing and developing a paddle and ball game in session 4, this process was followed:

(1) Starting with the question: "How can you go back to the initial situation and restart the game when the player lost the ball?" How can you tell the gamer "Game Over" when he or she has lost the ball? (Figure 4.)

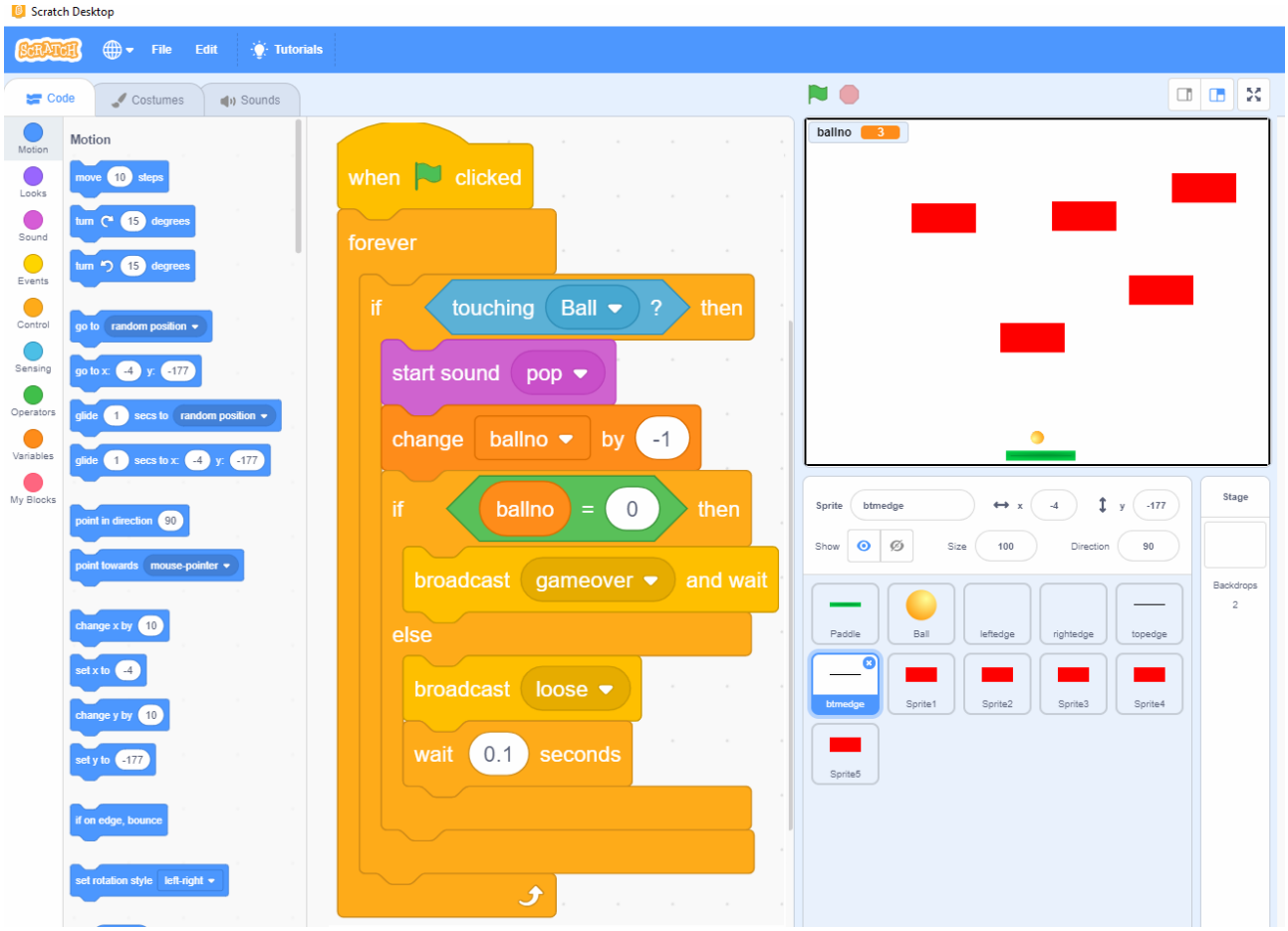


Figure 4. A programming script for the btmedge sprite designed by a student to answer the question

(2) Students first had time to think and then discussed with peers in a small group.

(3) Students shared their solutions to deal with the conditional situation.

(4) Ask students to implement their solutions. Support students during the implementation process. Make sure all students do it successfully.

(5) Ask students to reflect on the concept of messaging and its applications in programming and write notes in their notebooks. Asking a follow-up question: "How can you sense and show the destruction of a brick when the ball hits it?" and going back to step c to evaluate and ensure students' learning.

3.3.5 Evaluation Results of the Second Round

The students' learning quality, the impact of this course on their video gaming and their satisfaction from the course were studied. Furthermore, the results of data analysis showed more benefits of teaching programming based on the second-round design and implementation which are explained below.

Learning quality. Students' learning was evaluated through the pre and post-test method. A new performance test including 20 multiple-choice questions was designed to investigate students' learning levels for each topic. Also, the students' higher-order cognitive levels including evaluation and creation was evaluated through the teacher's observation from students' activities in the class. Table 5 shows the topic, learning level, and percentage of correct answers for pre- and post-test for each question. The table shows that students' results were generally satisfactory and most of the students achieved the learning objectives.

Qualitative analysis of the data collected from the students' self-evaluation questionnaire showed that they understood the basic concept of algorithm and coding deeply. For example, student 4 mentioned: "I understood that I should consider all details in programming and coding." and student 12 stated: "this game teaches us coding." Student 6 reported: "This game convinced me that coding is very sensitive, and a single mistake can destroy the whole project."

In addition, most of the students achieved higher level of learning programming. According to the researcher's observations and field notes taken from the students' activities in the class during teaching hours, they were able to implement small scripts and integrate different concepts such as loops and conditional statements, and solve the given programming problems. For instance, the designed script by student 1 to answer the challenging question described in section 3.3.4 is shown in Figure 4.

Table 5. Average of students' score for each question in pre and post-test assessment (second round)

Question No	Topic	Learning Level	Percentage of correct answers	
			Pre-test	Post-test
1	Basic concepts of programming	Understanding	31.25	100
2	Position and movement of objects	Understanding	12.5	100
3	Position and movement of objects	Applying	0	100
4	Data types and variables	Understanding	18.75	93.75
5	Data types and variables	Applying	0	93.75
6	operators	Analysis	6.25	68.75
7	operators	Applying	12.5	87.5
8	conditional statements	Understanding	12.5	93.75
9	conditional statements	Applying	0	93.75
10	events and sensing	Understanding	0	87.5
11	events and sensing	Applying	0	75
12	loops	Analysis	0	81.25
13	loops	Applying	6.25	93.75
14	graphics	Applying	0	87.5
15	sound	Applying	0	93.75
16	animations	Applying	0	87.5
17	messaging and block	Understanding	0	87.5
18	messaging and block	Applying	0	75
19	String	Applying	0	68.75
20	clone	Understanding	0	62.5

The data analysis of the students' self-evaluation questionnaire also revealed that students like the existing teaching methods and they prefer to work in small groups, learning in the class, rather than working individually. For instance, student 16 said: "I prefer to work in a group because I learn faster" and student 6 stated: "I like to work in a group because we can share our knowledge and experience." In addition, use of weekly assignments and online support were suggested by the students and their parents after the second week of implementation. In practice, the instructor taught the process of developing a new game each week and asked the students to add more features to the game as an assignment. Furthermore, the researchers provided an online support community using Telegram messenger based on the students' suggestions. The students were uploading their assignments to the community and asking their teachers and peers questions.

Impact of the solution on students' spending time on playing video games. According to the data collected through students' improvement self-evaluation open-ended questionnaire, round 2 treatment was more successful than the first round for reducing time spent on gaming. However, the students still spent a considerable amount of their free time on video gaming. The time spent on playing video games was found to reduce by an average of 7-8 hours per week, that is 2 hours less than in the first round. The average spending time for video gaming was reduced by 2-3 hours during weekdays and 5-6 hours during weekends.

Satisfaction. The result of descriptive data analysis from the student self-evaluation open-ended questionnaire is shown in Table 6. The result shows that all the students found the course interesting and useful. They confirmed that teaching game programming is a proper strategy to attract students' interests to learning programming. They also believed that "my robotic friend" game, used for teaching algorithm concepts", and Scratch programming were interesting and helpful.

Table 6. Students' satisfaction evaluation result (Second round)

Satisfaction item	Sub-items	Students' perception	Proponent percentage
Course content	My Robotic Friends game (for teaching Algorithm and basic concepts of programming)	Interesting	100
		Helpful	100
	Programming language (Scratch)	Easy to learn	87.5
		Interesting	93.75
		Free to use	100
	Learning resources (Scratch community, YouTube videos, etc.)	Easy to learn	81.25
		Useful	93.75
		Easy to access	100
Teaching method	Lecture	Clear explanation	81.25
	Learning activities (in and off class)	Useful	87.5
		Funny	81.25
General evaluation		Useful and interesting	100

Students' feedback through the open-ended questionnaire revealed that they like to learn mobile game programming as the next course and they asked the researchers to start a course for teaching mobile game programming as soon as possible. When asked them to write their suggestions for improving the quality of the course, only one of them suggested more practice while others made no suggestions and said the current method is good enough for them. The parents' feedback was equally exciting. For example, parent 2 said: "my son does coding for several hours continuously and when we call him he does not hear us at all".

It is noteworthy that sharing the developed programs with others through different ways is one of the helpful capabilities of the Scratch which impacts student learning and satisfaction. In fact, the ability to share products can increase students' motivation and learning. For example, when one of the students shared his game on the Scratch community, he was very excited to receive likes and some feedback from the other community members. Also, one of the parents said: "My son shared his game on his Facebook page and received a lot of likes from his friends. He enjoyed getting likes and comments."

Increasing students' passion to broaden their knowledge for game development. This course increased students' interest to learn programming beyond of the course syllabus. For example, parent 2 stated: "My son really likes programming with Scratch. He made nice games by following some video tutorials on YouTube. When he explained the game design, I understood he used some commands and concepts such as messaging before being taught in the class by his teacher." Furthermore, these activities can improve the students' self-confidence. For example, parent 8 stated: "My son would like to learn mobile game programming as soon as possible. He told me he is going to produce new games and share them on Google Play Store."

4 Discussion and Conclusion

4.1 Challenges of Managing Students' Energy

The results of the study about the impacts of video gaming on the secondary students raise many concerns and need more consideration. Whether we agree to consider excessive video game playing as a type of addiction or not, it consumes an important period in a student's life which could be spent on more useful activities. Directing students' energy away from playing video games to designing and producing them can be considered a solution. In addition to the fundamental motivation theories mentioned in section 1, this method can be supported by the attention switching strategy for facing addiction [26]. It seems that it is not easy to reduce students' time spent on video gaming considerably. More research is needed to improve this method through changing parameters such as class schedule, new ways for motivating and supporting students and so on. Providing diverse opportunities to improve student's self-confidence, self-directed learning, analytical view, and entrepreneurship skills can be considered as positive points of the course, however, it needs further study to extrapolate the results to a wider demographic and to create a stronger impact.

4.2 Instructional Design Principles for Teaching Programming to Secondary School Students

Based on the evaluation result of the second round (see 3.3.5) and the researchers' reflection during the implementation of two rounds teaching of computer programming to the secondary school students, the following instructional design principles have been developed (Table 7).

Table 7. Instructional design principles for teaching programming in secondary schools

Principle	Recommendation
Start with a proper programming language	Use a block-based programming language Use a supportive programming language
Use game-based learning materials	Unplugged activities should be used to teach basic programming concepts A proper game should be prepared for each teaching session
Follow design-based teaching principles.	(a) Start with a question or designing challenge. (b) Provide enough time for students to find a solution and discuss with peers in a small group. (c) Support students to improve their solutions. (d) Give an opportunity for students to test their solutions. (e) Ask students to reflect on the results of their experience and take notes.
Supporting activities are required	Give a proper assignment to the students Support students through an online network

4.2.1 Start with a Proper Programming Language

Regarding Table 4 and better results of the second round which is already explained in section 3.3.5, a block-based visual programming environment is easier and faster for learning programming concepts than text-based programming environments. Therefore, teaching programming to the students should start from block-based programming languages rather than text-based programming languages, because they provide a visual element for each programming command or structure. This approach is supported by previous research findings for teaching and learning programming [7, 27]. Using visualization for improving the quality of learning is supported by several learning theories such as the cognitive theory of multimedia learning [28]. These types of programming languages make programming easier by reducing both the complexity and the syntax errors. These features increase student satisfaction and motivation for learning programming [18]. Furthermore, the programming language environment should support programmers through different ways. It should provide easy access to diverse learning materials for programming. These materials should be easy to understand for secondary school students. Furthermore, students should be able to share their products easily with their friends or their chosen programming communities. The audiences should be able to run, rate and comment on these products easily. For example, the Scratch programs can be easily shared and run on the web, particularly in its online community. These facilities are aligned with design-based instruction [24] and constructivist instructional design for online learning [29].

4.2.2 Use Game-based Learning Materials

Regarding the study results, especially from the second round and students' satisfaction from the game-based instruction, programming instructors should use game-based learning materials and methods as much as they can. Game-based learning, which is supported by several learning theories such as motivational design for instruction [18], can be considered as the core strategy for teaching programming to secondary school students.

In the first sessions, basic concepts of algorithm, programming languages and commands, and how computers run programming commands should be taught via unplugged activities. The instructors should design game-based activities to attract students' interest in the course in the first sessions. The teachers should provide proper learning materials and instructions to guide students during unplugged activities. Also, they should define learning objectives for each session and choose a proper game for designing and implementing that covers the intended objectives. They should try to find interesting games regarding students' age groups.

4.2.3 Follow Design-based Teaching Principles

Based on the successful results of the second round of the study, the development process of the game should be divided into stages by the instructor. Then, for each stage, the instructor should use the design-based instruction principles [24] as shown in Table 7. Students should be encouraged to follow similar steps when they develop a new game out of class time.

4.2.4 Supporting Activities Are Required

Regarding the importance of programming practice and the instructor's feedback, and the successful results of the second round of this study, supporting activities should be employed by both teachers and students. Teachers should provide proper assignments for students, go through students' solutions for the assignments and leave appropriate comments. On their part, students should do their assignments to improve their programming skills and be prepared to learn new skills in the next session. Students should also be supported through an online community of their teachers and classmates so they can ask questions and share their ideas and products.

4.3 Future Research

Although this study answers the research question, it raises new questions that should be answered through future research. For example, different sides of video game development were attractive for the students (see 3.3.5). This raises more questions such as whether other technical skills like graphics and sound design or writing skills should or should not be considered in video game programming course planning for the secondary school students. In addition, the sample size in this study is relatively small, hence, the results cannot be generalized to a wider demographic. It can however be easily replicated. In fact, this study can be considered as a basis for future researches with a wider range of participants. For instance, further quantitative studies can be done to check the effectiveness of the proposed instructional principles among a large number of the students.

Acknowledgments

We thank all both universities for this collaborative research and special thanks to the LJMU for the financial support of this research.

References

[1] Y. B. Kafai, Q. Burke, Computer Programming Goes Back to School: Learning Programming Introduces Students to Solving Problems, Designing Applications, and Making Connections Online, *Phi Delta Kappan*, Vol. 95, No. 1, pp. 61-65, September, 2013.

[2] Y. Qian, J. D. Lehman, Correlates of Success in Introductory Programming: A Study With Middle School Students, *Journal of Education and Learning*, Vol. 5, No. 2, pp. 73-83, March, 2016.

[3] I. Park, D. Kim, J. Oh, Y. Jang, K. Lim, Learning Effects of Pedagogical Robots with Programming in Elementary School Environments in Korea, *Indian Journal of Science and Technology*, Vol. 8, No. 26, pp. 1-5, October, 2015.

[4] T. Yamanishi, K. Sugihara, K. Ohkuma, K. Uosaki, Programming Instruction Using a Micro Robot as a Teaching Tool, *Computer Applications in Engineering Education*, Vol. 23, No. 1, pp. 109-116, January, 2015.

[5] U. Nikula, O. Gotel, J. Kasurinen, A Motivation Guided Holistic Rehabilitation of the First Programming Course, *ACM Transactions on Computing Education*, Vol. 11, No. 4, pp. 1-38, November, 2011.

[6] S. Reardon, B. Tangney, Smartphones, Studio-based Learning, and Scaffolding: Helping Novices Learn to Program, *ACM Transactions on Computing Education*, Vol. 14, No. 4, pp. 1-15, February, 2015.

[7] A. Theodoropoulos, A. Antoniou, and . Lepouras, How Do Different Cognitive Styles Affect Learning Programming? Insights from a Game-based Approach in Greek Schools, *ACM Transactions on Computing Education*, Vol. 17, No. 1, pp. 1-25, January, 2017.

[8] M. M. McGill, Learning to Program with Personal Robots: Influences on Student Motivation, *ACM Transactions on Computing Education*, Vol. 12, No. 1, pp. 1-32, March, 2012.

[9] R. Festl, M. Scharkow, T. Quandt, Problematic Computer Game Use Among Adolescents, Younger and Older Adults, *Addiction*, Vol. 108, No. 3, pp. 592-599, March, 2013.

[10] D. J. Kuss, M. D. Griffiths, J. F. Binder, Internet Addiction in Students: Prevalence and Risk Factors, *Computers in Human Behavior*, Vol. 29, No. 3, pp. 959-966, May, 2013.

[11] F. Rehbein, T. Mößle, Video Game and Internet Addiction: Is There a Need for Differentiation?, *Sucht*, Vol. 59, No. 3, pp. 129-142, January, 2013.

[12] J. C. Beck, M. Wade, The Kids Are Alright: How the Gamer Generation is Changing the Workplace, *Harvard Business Press*, 2006.

[13] G. Bekebrede, H. J. G. Warmelink, I. S. Mayer, Reviewing the Need for Gaming in Education to Accommodate the Net Generation, *Computers & Education*, Vol. 57, No. 2, pp. 1521-1529, September, 2011.

[14] M. Kirchgessner, D. J. Ketelhut, Video Games and Learning: Teaching and Participatory Culture in the Digital Age, *Science Education*, Vol. 96, No. 5, pp. 963-965, September, 2012.

[15] L. G. Polin, A Constructivist Perspective on Games in Education, in: D. W. Kritt (Ed.), *Constructivist Education in an Age of Accountability*, Palgrave Macmillan, Cham, 2018, pp. 163-188.

[16] C. Obikwelu, J. C. Read, The Serious Game Constructivist Framework for Children'S Learning, *Procedia Computer Science*, Vol. 15, pp. 32-37, 2012.

[17] A. H. Maslow, R. Frager, J. Fadiman, C. McReynolds, R. Cox, *Motivation and Personality*, Pearson, 1987.

[18] J. M. Keller, First Principles of Motivation to Learn and e3-learning, *Distance Education*, Vol. 29, No. 2, pp. 175-185, July, 2008.

[19] M. D. Merrill, L. Drake, M. J. Lacy, J. Pratt, Reclaiming Instructional Design, *Educational Technology*, Vol. 36, No. 5, pp. 5-7, September, 1996.

[20] T. C Reeves, Design Research from a Technology Perspective, in: J. V. D. Akker, K. Gravemeijer, S. McKenney, N. Nieveen (Eds.), *Educational Design Research*, Routledge, 2006, pp. 52-66.

[21] The Design-Based Research Collective, Design-based Research: An Emerging Paradigm for Educational Inquiry, *Educational Researcher*, Vol. 32, No. 1, pp. 5-8, January, 2003.

[22] J. W. Creswell, Educational Research-Planning, Conducting, and Evaluating Quantitative and Qualitative Research, Pearson, 2012.

[23] N. K. Denzin, The Research Act: A Theoretical Introduction to Sociological Methods, *Routledge*, 2017.

[24] J. L. Kolodner, D. Crismond, J. Gray, J. Holbrook, S. Puntambekar, Learning by Design from Theory to Practice, *Proceedings of the International Conference of the Learning Sciences* (ICLS-98), Atlanta, GA, 1998, pp. 16-22.

[25] M. S. Farooq, S. A. Khan, F. Ahmad, S. Islam, A. Abid, An Evaluation Framework and Comparative Analysis of the Widely Used First Programming Languages, *PLoS ONE*, Vol. 9, No. 2, e88941, February, 2014. Doi: https://doi.org/10.1371/journal.pone.0088941.

[26] J. Maloney, M. Resnick, N. Rusk, B. Silverman, E. Eastmond, The Scratch Programming Language and Environment, *ACM Transactions on Computing Education (TOCE)*, Vol. 10, No. 4, pp. 1-15, November, 2010.

[27] D. Weintrop, U. Wilensky, Comparing Block-based and Text-based Programming in High School Computer Science Classrooms, *ACM Transactions on Computing Education*, Vol. 18, No. 1, pp. 1-25, December, 2017.

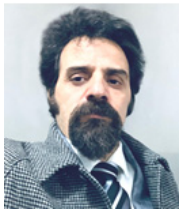
[28] R. E. Mayer, *Multimedia Learning*, Cambridge University Press, 2009.

[29] M. Tam, Constructivism, Instructional Design, and Technology: Implications for Transforming Distance Learning, *Educational Technology & Society*, Vol. 3, No. 2, pp. 50-60, April, 2000.

Biographies

Ataollah Zarei is a visiting scholar at University of Windsor, and a Ph.D. candidate of Engineering Education at Centre for Engineering Education, Universiti Teknologi Malaysia. He obtained bachelor of Computer Engineering in 1999 and Master of Education in 2010. His research area include Engineering Education, Mobile Learning and Game-based learning.

Khairiyah Mohd-Yusof is a Professor and the Director of Universiti Teknologi Malaysia, Centre for Engineering Education. She is the President for the Society of Engineering Education Malaysia (SEEM). Her research area is Innovative Teaching and Learning Practices, Curriculum Design, Faculty Development and Engineering Education for Sustainable Development.



Hoshang Kolivand is a senior Lecturer in department of Computer Science, Liverpool John Moores University, UK. He has received his MSc from AmirKabir University of Tehran and his Ph.D. from Universiti Technologi Malaysia.

Arash Ahmadi received the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2008. He is a Faculty Member with the University of Windsor, ON, Canada. His current research interests include neuromorphic, bio-inspired computing and engineering education.

Yousef Al-Hamar received his MSc from Loughborough university and his PhD from Liverpool John Moores University, UK. He is currently working in Qatar Foundation his current research interest social engineering attacks, cyber security, user behavior impact on security and user training awarness.