



# MiniColon; Teaching Kids Computational Thinking Using an Interactive Serious Game

Reham Ayman<sup>(✉)</sup>, Nada Sharaf, Ghada Ahmed, and Slim Abdennadher

Department of Computer Science, Faculty of Media Engineering and Technology,  
German University in Cairo, Cairo, Egypt

reham.saad@student.guc.edu.eg,  
{nada.hamed,ghada.bahaeldin,slim.abdennadher}@guc.edu.eg

**Abstract.** Computer science is about learning how to think. It is applicable to everything. Most industries in the 21<sup>st</sup> century are directed towards digitized processes using up-to-date technologies. This increases the need for students to know how these technologies work, especially with the availability of smart and handy devices. As a preliminary step to learning programming, students should learn basics of logical and computational thinking. Computational thinking comprises the skills, concepts, and behaviors used to solve problems. Some visual platforms, such as Scratch and Alice [6, 17], were introduced to teach children programming. The platform introduced in this paper aims at teaching young children (in the age range of 8–9 years old) the preliminary concepts of programming including *sequential blocks*, *conditional blocks*, and *iterative blocks*. This is achieved through an interactive, gesture-based game, where the Kinect Sensor device is used to control the gameplay instead of playing it using a normal PC. The game was tested using a between-group experimental model with two subgroups; an experimental group and a control group. The experimental group used the game whereas the control group was taught using a traditional educational method. The results showed a significant difference between the two groups with a P-value of  $< 0.05$  for both the learning gain and the engagement level.

**Keywords:** Serious games · Programming · Computational thinking  
Interactive · Children · Kinect

## 1 Introduction

Computational thinking is increasingly becoming a needed skill [5]. It helps individuals identify a problem and the steps needed to solve it. With the widespread of computers and web applications, a lot of opportunities have been added to the worldwide job market. This encouraged a lot of individuals to try to learn to program.

In addition, there has been an increasing trend recently to introduce programming to children in their early stages [13]. Many schools started to offer

introductory courses to teach programming and computational thinking. These courses help the students to have a better way of thinking and enhance their problem-solving skills.

Some platforms were designed for introducing computational thinking and programming basics to children such as Scratch and Alice [6,14]. These platforms provide a visual programming language to eliminate the need for learning any syntax or understanding complex code structures. In addition, serious games were also employed to introduce programming concepts [2,19]. However, most of these serious games that were played on computers or mobile devices, were missing the interactivity factor. Studies showed that adding the factor of interactivity motivates the students to learn more about programming [1]. The aim of the introduced platform is to add to the existing games the factor of interactivity. Therefore, Kinect [16] is used to capture the movement of the child and incorporate it into the game.

The paper is organized as follows: Sect. 2 provides previous work related to the education of computational thinking through serious games. Section 3 discusses the methodology and explains how the levels of MiniColon are designed. Section 4 explains the experimental model used. Section 5 shows the tests conducted and the results. Finally, Sect. 6 concludes the work and describes directions for future work.

## 2 Related Work

Various methods of teaching were applied to teach children programming. Visual programming environments were used to enable children to build programs through simple interfaces [9]. Such platforms include Scratch [17] and Alice [6]. According to Armoine et al. [3], teachers reported that using Scratch increased the learning efficiency. Students also reported that they were encouraged to learn more about computer science.

“Program your robot” [11] is a serious game developed for teaching programming and computational thinking. The game aimed at integrating the game-play with the programming concepts and computational thinking skills. The game was evaluated using 25 students. The students enjoyed the game and reported that this type of games can improve their problem-solving skills.

Another example of an educational game for programming used tangible electronic building blocks instead of visual programming is presented in [27]. The Tangible electronic blocks are divided into sensors and sources for these sensors. The child can build a structured block that does a function by connecting the blocks together. The older children were able to debug their structure in case of a problem or undesired function. The results showed that young aged children did not realize the concepts of programming using the Electronic Blocks. Therefore, this way was not effective enough for the young children.

There are other technologies that are used, such as motion-based touchless and virtual reality. Motion-based touchless technology is a new trend in the

field of human-computer interaction<sup>1</sup>. These types of interactive games motivate the students to learn more using game-based technologies [1]. Participating in Kinect-based learning activities encourage students to brainstorm and active discussions participation in the classroom. This was proved by [1] that included Scratch, Microsoft Kinect, and Kinect2Scratch. Microsoft Kinect is a sensor that tracks body motion and gestures. Kinect2Scratch allows data from a Microsoft Kinect controller to be sent to Scratch. According to the result of this study, students showed more interest and learning skills regarding their computational thinking skills and social skills, during and after the experiment. According to [20], students who participated in the experiment of VR-based game liked it a lot. Virtual Reality technology is about immersing the player in a virtual world that they can interact with. The game was controlled using VR and Kinect technologies. The Kinect sensor was used to make the player move freely without the need to hold the physical controller of the VR set. Students had to design a dance choreography using coding blocks. Then the character in the VR environment performed the dance in front of the player.

Kinect was also used for game-based learning in different fields. It was used to teach hearing impairments students the American sign language [10]. In this application, motions were captured using Kinect one, then they were encoded. Afterwards, a game was developed using Unity and Kinect one. The game was tested against a face-to-face learning method. The result showed a significant effect of the game compared to the traditional method with a (P-value < 0.05). Kinect was also used for teaching the spatial visualization skills [25]. The participants in this work agreed that the interactive learning system made them aware of various things and increased their interest.

### 3 Methodology

Serious games have proven to be effective in teaching children, as they enjoy the game while learning the concept subconsciously. Our aim is to teach children an introduction to logical thinking and programming. This is achieved through an interactive serious game that has an enjoyable theme and storyboard. Having a story makes the game more enjoyable and playable [4]. The storyboard is designed to be simple and yet attractive to the target age group. The main idea of “MiniColon” is that the main character is trapped on an island. The character has to collect some fruits and supplies to escape from the island. The main character is a carrot and it is moving forward all the time during the gameplay. The player controls the motion of the carrot in order to achieve the goal of each level. The interaction with the game was intended to be engaging and yet seamless so that the player remains focused on the main concept. Kinect was used to detect the interactions of the player with the game. Kinect is a hands-free motion control device which implied that the players will not hold any controller in their hands.

<sup>1</sup> A field of study focusing on the design of computer technology and, in particular, the interaction between humans (the users) and computers.

Motivation is the key to learning as it helps the students to fully engage in learning [26]. Thus, after finishing the level a sound referring to winning is played as well as a congratulation message is displayed. In level 3 and 4, a sound is played on opening the chests to indicate the chest is opened. Also, the objects to be collected in the scene bounce regularly to be more visible and attract the student attention. Moreover, throughout all the game levels, a motivational background music is played.

### 3.1 Game Structure

MiniColon aims at teaching children the preliminary steps of programming and logical thinking which are *sequential blocks*, *conditional blocks*, and *iterative blocks*. Each concept has a group of levels that the player has to pass. The last level of every asks the player to trace a code snippet. Teaching the student the skill of code tracing is important as it helps in improving code writing skills later on [12]. The code snippets are written in Python language as it is close to pseudocode [24].

**Sequential Levels.** An algorithm is a set of instructions used to achieve a certain task where the order of the steps is crucial [8]. Delivering this concept to children can be done by describing the instructions for a certain activity in the correct order to reach their target. The consequences of missing a step or changing the order of the instructions have to be demonstrated to the player. The sequence concept is introduced in the game by showing the player a set of instructions that have to be done in the same order. In the first level of the game, the player is shown some objects to be collected in order. These objects are displayed on the left side of the screen as shown in Fig. 1. If the player tries to collect an object out of order a hint will be displayed stating the object that should be collected now as shown in Fig. 2a.

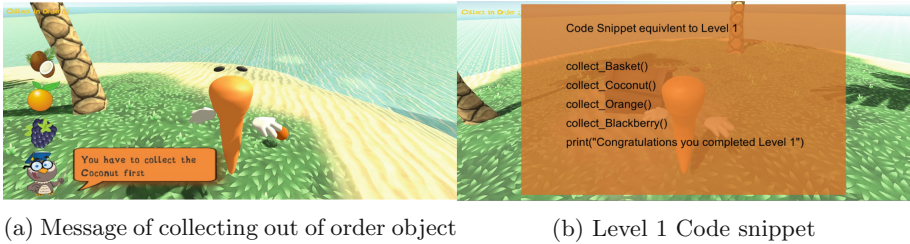
Once the player is done with all the instructions for this level, a code snippet in python is displayed as shown in Fig. 2b. The code is equivalent to the steps the player followed to collect some objects from the island. A final congratulation message is then displayed to motivate the player and inform him/her that the current level is completed successfully.

In order to make sure that the player grasped the concept correctly, the second level then asks the player to trace a snippet of code. The code is displayed as shown in Fig. 3a, where the player has to follow it by doing the actions corresponding to the code.

**Conditional Levels.** A program consists of some events that are controlled by conditionals. In general, conditions control the flow of the program as they have to be achieved in order for some event to be triggered. The conditional statements are expressed in most programming languages using the if-then construct; where the condition evaluates to a boolean state that has to be true for the then part to be done.



**Fig. 1.** Collecting a coconut

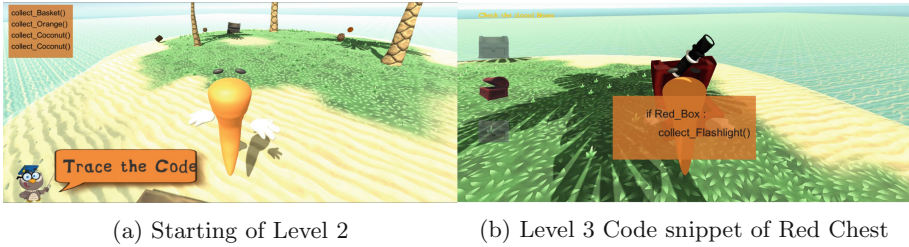


**Fig. 2.** Messages and code snippets

MiniColon teaches the conditional concept through level 3 and level 4. Level 3 applies this concept by having three chests which the player can open in any order. Upon opening any chest, an object will appear with the equivalent code snippet being displayed. For example, the red chest has a flashlight inside. After opening it, a code snippet saying (`if red_box: collect_flashlight()`) appears, as shown in Fig. 3b. This helps the player to grasp the idea that in order to achieve the then part (in this case collecting a certain object), the condition has to be true (in this case the corresponding box is open).

Level 4 is the code tracing level. This level tests the ability of the players of understanding the sequential algorithm and condition concepts together. This is achieved by having the first line of code to collect a backpack. After that, there are conditions corresponding to each chest. The player has to trace them in order. If he/she tries to open a chest before the required one, or opening the red chest before collecting the backpack, he will be notified with a message to collect or open the required object.

**Iterative Levels.** In computer programming, an iterative block is a sequence of instructions that is continually repeated. Its main aim is to do a set of instructions repeatedly for a certain number of times, or until the condition of the iterative block is false. The game has two levels for teaching the iterative concept. In level 5 scene the island has many scattered coconuts (more than 4), and on the left side of the screen, there is an image of a coconut with a number attached to it. This number is a counter which is initially four; meaning that



**Fig. 3.** Level 2 and 3 features (Color figure online)

the player has to collect four coconuts to proceed to the next level. Every time the player collects a coconut, the number is decremented by one until it reaches zero. On reaching zero, an equivalent code snippet is displayed. The condition of the while loop says that (`while number_collected < 4:`), which means that if the `number_collected` is 4, the condition will be false and the iterative block will not be entered.

In Level 6, the students are shown a code snippet and they have to trace it (as in Level 2 and 4). This level combines two concepts, sequential algorithm, and iterative concepts. The players are required to first collect four coconuts then three blackberries. This is achieved by having two iterative blocks in sequence, the first is to collect the coconuts and the second is to collect the blackberries. In addition to the code snippet, there are images on the left of the screen, a coconut, and a blackberry, each has a counter set to zero at the start of the level. The number attached to coconut image is incremented by collecting the coconuts. Once all the four coconuts are collected, the image disappears to indicate that the first iterative block condition is no more satisfied and that the player has to continue to the next instruction i.e. next iterative block.

### 3.2 Implementation

Motion-based touchless technology is a new trend in the field of human-computer interaction. It can be used to increase student motivation, strengthen their computational thinking and enhance their understanding [1]. “MiniColon” is a motion-based touchless game that was implemented in three stages. The first stage was to implement the game based on the target group. The following stage was implementing the leaning motion using the motion-based touchless technology, which is the Kinect. Finally, the game was integrated with the Kinect. The game is implemented using Unity [7] game engine. The movement of the character is simulated by the leaning gesture; such that leaning to the right is interpreted to move the character to the right direction and the same analogy for the left direction. Leaning is the way of moving the upper part of the body to the right and left, as the upper body makes an angle with the right or left leg depending on the leaning side. It was chosen because it is an easy motion for

children to do and does not require a lot of activity to allow them to concentrate on the delivered concepts from each level.

The lean gesture was built using Kinect studio and Kinect visual gesture builder [22]. It was recorded by the camera of the Kinect sensor using Kinect studio software. The act of the motion was recorded several times. The visual gesture builder then uses the recorded motions to train a model using a machine-learning algorithm to create the leaning gesture. The right lean gesture is trained and saved in a database file called Lean. The same is done for the left lean gesture. Thus, the file Lean is the database file containing both right and left lean gesture. To access the database file a C-Sharp class `GestureDetector` is implemented. The class job is to read the database file and compare the tracked gesture with the lean gestures saved in the database.

The integration between the tracked gestures from the Kinect and the game levels is implemented using a script `KinectManager` in C-Sharp language. The functionality of the script is to initialize the Kinect sensor on opening the game. It also passes the tracked gesture from the Kinect to the `GestureDetector` class to be checked if it is leaning right or left.

In order to collect an object, the closed hand state gesture [22] of the right/left hand is checked. The two hands are included to make it easier for the students to interact with the hand which the student has more control of.

Moreover, after finishing the level, the player waves also with his right/left hand to go to the next level. Detecting the wave is implemented by checking the opened hand state [22].

## 4 Experimental Design

This project experiments the effect of interactive serious games to teach young aged children the main concepts of programming, compared to the traditional methods. This is achieved by implementing MiniColon and testing it with young aged children. This is examined by how much did the children interacted with the game, enjoyed, and learned from the game. The targeted group is children of the age between (8–9). The aim of the experiment is to proof or reject the null hypothesis. The null hypothesis states that there is no difference between the effect of using “MiniColon” and using traditional teaching methods in learning programming. The first hypothesis (H1) states that there is no difference between the learning gain between the two methods. The second hypothesis (H2) claims that there is no difference in the engagement level of the participants between the traditional method and MiniColon.

The model used is a between-group design that has two independent sub-groups: a control group and an experiment group. The experiment group used Kinect interactive game “MiniColon”, and control group experienced traditional face-to-face learning method using presentation slides [23]. The slides were designed to be identical to the game flow and structure. The participants for both groups were selected randomly. The only required factor that, they should not have a previous background in programming to ensure homogeneity of the



experiment. The experiment for each participant of the experiment group is held individually in a normal room. Hard copied test for the participants, hard copied questionnaires, and presentation slides are the materials used in the experiment. The experiment process consists of four stages, first the pre-quiz, then the experiment method, then the post-quiz, and finally the engagement test.

## 5 Test and Results

A sample population of 15 children (between the age of 8 and 9) was divided into two groups (an experiment group:  $n = 8$ , and a control group:  $n = 7$ ). Each group was introduced to its assigned learning method. Experiment group participants played the 6 Level game using the Kinect. While the control group participants were taught using face-to-face learning method. This testing approach is inherited from a similar research, that was testing the effectiveness of serious game against traditional methods [23]. The duration for both groups took approximately the same amount of time (between 15 to 20 min). To compare between two learning methods, learning gain and engagement level of both groups were measured after using the corresponding learning approach. Analyzing the data comparison between the two groups, an independent t-test using SPSS (Statistical Package for the Social Sciences) is applied. The independent sample t-test is used to clarify if there is a significant difference in the learning gain and the engagement level between the control group and the experiment group.

### 5.1 Learning Gain

Learning gain is how much the participant learned from the activity. It can be tested using oral questions, observations from the participants, and pre and post quizzes. The participant is tested before the activity to know if he has previous knowledge about the activity. After the activity is finished, the participant is tested again to know how much information did he learned from the activity.

Procedure: Before starting the two different learning methods, participants were given a printed pre-quiz. The quiz consists of six questions, divided into three groups, each has two questions. Each group is related to one of the three targeted programming concepts *sequential blocks*, *conditional blocks*, and *iterative blocks* respectively. The quiz has five MCQ questions and a “What is the output?” question. Some of the children required help in reading or understanding the synthesis of the questions. The questions are designed to be compatible with the content of the learning methods. After the participant completes his assigned learning method, he is given an identical copy of the pre-quiz. The learning gain is calculated by subtracting the number of correct answers that the participant got in the pre-quiz from the number of correct answers he got in the post-quiz.

Results: The results of the tests show that the learning gain achieved from the group using the Kinect game ( $M = 2.75$ ,  $SD = 1.389$ ) is significantly higher than the group that was introduced to the traditional educational method ( $M = 0.43$ ,



SD = 0.787) ( $t(8) = 4.04$ ,  $p < 0.05$ ) with a difference of 2.32 and standard error difference of 0.57. This rejects the hypothesis that stated that there is no difference in the learning gain level of the MiniColon game to the traditional learning method (H1).

## 5.2 Engagement

Engagement is the concept of the presence or the sense of “being there” in the game. Involvement, immersion, arousal, attention, interest, identification, enjoyment, effort, and flow are the factors that defined engagement with the game. There are many approaches that are used to measure engagement, self-reports on questionnaires is the most used to measure the engagement level of the games [15]. Thus, in this experiment self-reported questionnaire is used.

Procedure: This test is held to measure how the two groups differ in the engagement level. It is a questionnaire that consists of 9 items. The answers to the questionnaire are 5-Likert scale. This questionnaire is proposed by [21] to measure the control and enjoyment of the activity, to indicate the overall flow. After completing the assigned learning method, the researcher asks the participant the questions of the questionnaire and then records the answers of the participant in a hand-copied version of the questionnaire. The researcher has to ask the participants the questions of the questionnaire instead of handing it to them to fill it alone. This is due to their young age, which led to some ambiguity in understanding some words. Conclusions about the engagement level of the two learning methods are drawn, by comparing the result of both groups.

Results: The results of the tests show that the engagement revealed from the group using the Kinect game ( $M = 4.33$ ,  $SD = 0.428$ ) is significantly higher than the group that used the traditional educational method ( $M = 2.93$ ,  $SD = 0.96$ ) ( $t(8) = 3.5$ ,  $p < 0.05$ ) with a difference of 1.4 and standard error difference of 0.397. This rejects the hypothesis that stated that there is no difference in the engagement level of the MiniColon game to the traditional learning method (H2).

## 5.3 Flow

Flow according to Csikszentmihalyi, is the channel between challenge and skill that the user experience in any activity [18]. Anxiousness is the result of experiencing the high challenge and low skill, while boredom is the result of the opposite case. So the best case is the balance between the two factors. Flow test is used to measure how did the player feel about the flow of the game from the first level passing through all the other levels.

Procedure: After each level of the game, the test of flow is taken by every participant. According to [21], the questionnaire consists of two five Likert scale questions: “How did the participant find this level challenge?” and “Is the skill of the participant appropriate for this level?”. After finishing the level, the researcher asks the participant these questions and record the participant’s answer on a hand-copied version of the questionnaire, which means 12 answers for each participant.

Results: Fig. 4a illustrates the average of skill and challenge in each level. While the bars in Fig. 4b is calculated according to [21] by the equation.

$$0.25 \times (Skill - Challenge) \tag{1}$$

This graph shows how the flow of each level is far from the optimal flow (zero flow). The figure shows a significant difference at level one, which means that the participants were bored at the start of the game. This is because their skills were higher than the challenge factor of the game. But the more the levels they went through, the more challenging the game. Until the last two levels are reached, the challenge factor gradually decreases. Most of the participants felt that their skills are suitable for the game, with average skill of 3.25 for all levels.

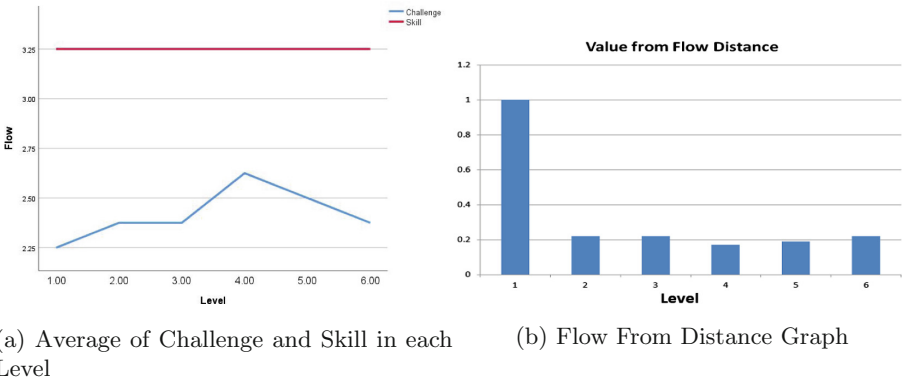


Fig. 4. Flow results

## 6 Conclusion and Future Work

Using interactive serious games for educating children computational thinking and programming is a growing research field. The main objective of this project was to test the effectiveness of interactive game for computational thinking and programming learning using the Kinect sensor on children’s learning gain. “Mini-Colon” is an educational game for children of the age between 8–9 that teaches three main concepts in programming (*sequential blocks, conditional blocks, and iterative blocks*). The game is controlled using the Kinect sensor to add interactivity to the game. The game was designed in a cartoonish and enjoyable way to attract the children. The game was evaluated with two different groups, one group tested the game and the other group was taught in a traditional educational method. The traditional educational method was a teacher who used slides to assist in the education process. The result showed that playing the game has more effect on the learning gain and the engagement level of the children than using a traditional educational method. According to the game flow results, the

skill level of the participants was higher than the challenge level of the game. Most of the participants found level 1 not challenging and their skill is suitable for the game. At each level, the challenge level is increasing, but it decreases again when level 5 is reached. Thus, the challenge level of the game needs to be increased especially for level 1. According to the results, adding gesture recognition factor to educating computational thinking using Kinect helps the students to be more engaged in the education process.

More than twenty schools were contacted, however, only one school was interested in trying the project with its children. This is due to cultural reasons for not engaging in modern teaching technologies. Also, some children were not interested in trying the game or joining the activity of the traditional educational method. These led to the lacking number of participants in both groups.

Some of the students suggested that it is better if the main character can move backward. Also, more levels will be implemented to target more advanced concepts in computational thinking and programming such as functions and data structure. We need to do a more extensive long-term study on children, to test the effectiveness of the game in the long run.

## References

1. Altanis, I., Retalis, S., Petropoulou, O.: Systematic design and rapid development of motion-based touchless games for enhancing students' thinking skills. *Educ. Sci.* **8**(1), 18 (2018)
2. Alvarado, C.: CS Ed week 2013: the hour of code. *SIGCSE Bull.* **46**(1), 2–4 (2014)
3. Armoni, M., Meerbaum-Salant, O., Ben-Ari, M.: From scratch to “real” programming. *ACM Trans. Comput. Educ. (TOCE)* **14**(4), 25:1–25:15 (2015). <https://doi.org/10.1145/2677087>
4. Arnold, S., Fujima, J., Jantke, K.P.: Storyboarding serious games for large-scale training applications. In: Foley, O., Restivo, M.T., Uhomoihi, J.O., Helfert, M. (eds.) *CSEDU 2013 - Proceedings of the 5th International Conference on Computer Supported Education*, Aachen, Germany, 6–8 May 2013, pp. 651–655. *SciTePress* (2013)
5. Barr, D., Harrison, J., Conery, L.: Computational thinking: a digital age skill for everyone. *Learn. Lead. Technol.* **38**(6), 20–23 (2011)
6. Carnegie Mellon University: Alice. <https://www.alice.org/>. Accessed 3 June 2018
7. Craighead, J., Burke, J., Murphy, R.: Using the unity game engine to develop sarge: a case study. In: *Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008)* (2008)
8. Dumont, J., Sicard, P., Stum, J., Zanife, O.: Algorithm definition accuracy and specification, vol. 2: CMA altimeter level 2 processing. CNES, Toulouse (2001). SMM-ST-M2-EA-11005-CN
9. Kalelioğlu, F., Gülbahar, Y.: The effects of teaching programming via scratch on problem solving skills: a discussion from learners' perspective. *Inform. Educ.* **13**(1), 33–50 (2014)
10. Kamnardsiri, T., Hongsit, L.O., Khuwuthyakorn, P., Wongta, N.: The effectiveness of the game-based learning system for the improvement of American sign language using kinect. *Electron. J. e-Learn.* **15**(4), 283–296 (2017)

11. Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Soc. Behav. Sci.* **47**, 1991–1999 (2012)
12. Kumar, A.N.: Solving code-tracing problems and its effect on code-writing skills pertaining to program semantics. In: Dagiene, V., Schulte, C., Jevsikova, T. (eds.) *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCS 2015, Vilnius, Lithuania, 4–8 July 2015*, pp. 314–319. ACM (2015). <http://dl.acm.org/citation.cfm?id=2729094>
13. Little, P., Wimer, C., Weiss, H.B., et al.: After school programs in the 21st century: their potential and what it takes to achieve it. *Issues Oppor. Out-Of-Sch. Time Eval.* **10**(1–12) (2008)
14. Maloney, J.H., Peppler, K.A., Kafai, Y.B., Resnick, M., Rusk, N.: Programming by choice: urban youth learning programming with scratch. In: Dougherty, J.D., Rodger, S.H., Fitzgerald, S., Guzdial, M. (eds.) *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2008, Portland, OR, USA, 12–15 March 2008*, pp. 367–371. ACM (2008)
15. Martey, R.M., et al.: Measuring game engagement: multiple methods and construct complexity. *Simul. Gaming* **45**(4–5), 528–547 (2014)
16. Microsoft: Microsoft Kinect. <https://developer.microsoft.com/en-us/windows/kinect>. Accessed 15 Mar 2018
17. MIT Media Lab: Scratch. <https://scratch.mit.edu/>. Accessed 1 June 2018
18. Nakamura, J., Csikszentmihalyi, M.: The concept of flow. In: Csikszentmihalyi, M. (ed.) *Flow and the Foundations of Positive Psychology*, pp. 239–263. Springer, Dordrecht (2014). [https://doi.org/10.1007/978-94-017-9088-8\\_16](https://doi.org/10.1007/978-94-017-9088-8_16)
19. Orsini, L.: Kodable teaches kids to code before they learn to read (2013)
20. Parmar, D., et al.: Programming moves: Design and evaluation of applying embodied interaction in virtual environments to enhance computational thinking in middle school students. In: Höllerer, T., Interrante, V., Lécuyer, A., Suma, E.A. (eds.) *2016 IEEE Virtual Reality, VR 2016, Greenville, SC, USA, 19–23 March 2016*, pp. 131–140. IEEE Computer Society (2016). <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7499993>
21. Pearce, J.M., Ainley, M., Howard, S.: The ebb and flow of online learning. *Comput. Hum. Behav.* **21**(5), 745–771 (2005)
22. Rahman, M.: *Beginning Microsoft Kinect for Windows SDK 2.0: Motion and Depth Sensing for Natural User Interfaces*. Apress, New York (2017)
23. Salah, J., Abdennadher, S., Sabty, C., Abdelrahman, Y.: Super alpha: arabic alphabet learning serious game for children with learning disabilities. In: Marsh, T., Ma, M., Oliveira, M.F., Baalsrud Hauge, J., Göbel, S. (eds.) *JCSG 2016. LNCS, vol. 9894*, pp. 104–115. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-45841-0\\_9](https://doi.org/10.1007/978-3-319-45841-0_9)
24. Stajano, F.: Python in education: raising a generation of native speakers. In: *Proceedings of 8th International Python Conference*, pp. 2000–2001 (2000)
25. Tsai, C., Kuo, Y., Chu, K., Yen, J.: Development and evaluation of game-based learning system using the microsoft kinect sensor. *IJDSN* **11**, 498560:1–498560:10 (2015)
26. Wang, Y.P., Chiang, C.S., Su, C.Y., Wang, C.C.: Effectiveness of virtual reality using Wii gaming technology in children with down syndrome. *Res. Dev. Disabil.* **32**(1), 312–321 (2011)
27. Wyeth, P., Purchase, H.C.: Tangible programming elements for young children. In: Terveen, L.G., Wixon, D.R. (eds.) *Extended abstracts of the 2002 Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, 20–25 April 2002*, pp. 774–775. ACM (2002)