Towards a Web Generator of Programming Games for Primary School Children

https://doi.org/10.3991/ijep.v12i4.17335

Javier Erazo-Palacios, Carlos R. Jaimez-González(✉), Betzabet García-Mendoza
Universidad Autónoma Metropolitana, Ciudad de México, México
cjaimez@cua.uam.mx

Abstract—This paper presents a web system to generate programming games for primary school children, with the objective to teach basic programming concepts. It also reviews five existing games with the same purpose, which use either block-based or text-based programming languages. A comparative analysis of the games is provided, based on some important features, such as the presence of a levels map, the use of a block or text language, the presence of a result screen, the presentation of the shortest and previous solutions, among others. The web system is divided into two main parts: the game generator that is used by teachers to create new programming games, and the programming games themselves that are played by students.

Keywords—programming game, primary school children, basic programming concepts, block-based language, educational technology

1 Introduction

In recent years, programming has become part of the curricular content of many schools at various levels. It is common to question the usefulness of learning programming, but one of the main reasons is that, although the knowledge acquired is probably not of practical use, as in many other fields, its utility lies in increasing and enriching the student's perspective [1]. On the other hand, the way programming is taught is a factor to take into consideration when observing the performance results of students. The difference between teaching programming using a visual programming language and a text-based programming language is important, since it affects the way in which a student understands the subject [2].

The work proposed in this paper is a system that generates personalized programming games, the objective of which is to teach basic programming concepts to primary school students. The rest of the paper is organized as follows. Section 2 provides the justification to develop the web generator of programming games. The theoretical framework is presented in Section 3, along with various approaches that have used similar systems, and the results that were obtained in studies conducted around the world. Section 4 presents an analysis and comparison of the features of some existing similar systems, including the proposed system. Section 5 describes the proposed

system, along with some preliminary interfaces. Finally, conclusions and future work are presented in section 6.

2 Justification

The constant role that technology plays today in our lives and in education is undeniable. According to the National Survey on Availability and Use of Information Technologies in Households (ENDUTIH) [3], in 2018 there were more than fifty million Mexicans with Internet access, who represented more than half of the population with six or more years old. Furthermore, one of their main activities was to search and obtain information from the Internet; it was also found that more than fifty million Mexicans used a computer. For these reasons it is important to know how to take advantage of these technological tools and the scope they have from an early age.

Learning programming can be intimidating and, especially, if the way of teaching the subject is not suitable. It is common to observe in university students taking programming courses failing and withdrawing, because they consider the subject difficult [4]. Despite this, learning programming has many benefits. According to an analysis [5] carried out with quantitative data obtained from sixty-five studies about the effects of programming and its cognitive results, it was shown that those students with programming experience obtained better scores in tests of cognitive skills compared to those who had no such experience.

Using computer tools in a classroom to support teaching have different effects on students. Compared to other traditional methods, computer tools make them motivate and participate, making the process attractive and dynamic. Furthermore, it offers an immediate feedback factor, since the student observes his progress instantly [6]. Previously, the use of visual programming languages in teaching and its benefits had been mentioned, compared to the use of text-based ones. An example is the block-based programming, which is quite popular for introductory programming teaching. Its advantages are that the blocks simplify concepts and depend purely on visual recognition, compared to other text-based languages that make learning difficult due to other cognitive aspects [7].

Given the above, the proposed system will be capable of offering an environment to support the teaching of basic programming concepts, through programming games. The target of the system is focused on elementary school students, but a teenager or even an adult can use it and learn.

3 Theoretical framework

There have been some reasons in previous sections that explain the need of creating a system that uses a block-based programming language for learning programming. The use of computers as a means to carry out learning activities is something that occurs in schools at all levels. Its incorporation into the educational system allows teachers to teach and students to learn in ways that were not possible years ago.

However, the use of a computer does not always mean real and quality learning. Many times, students of all levels are subject to moments of frustration and discouragement because the way they are supposed to learn is complicated and boring, resulting in a poor learning experience and sometimes leading to abandon their studies completely. This is the case with programming learning. According to the results of a preliminary study of university students in Scotland [8] where they were asked to mention a subject that was difficult for them and explain what strategies they had used to tackle it, it was found that half of them only used Google as strategy to solve their problems and nothing else. In addition, it was found that many of them did not understand the relationship of that difficult topic with the rest of the subject, others believed that understanding it did not help them much to understand other topics. This means that the students' problem solving skills were actually very poor, as they only concentrated on finding the solution in the quickest and easiest possible way. Furthermore, they lacked of perspective due to their inability to relate concepts from different programming topics. The solution to avoid that the students have problems when learning programming is perhaps to start teaching it when they are little and with a variety of approaches and strategies.

Over the years, studies have been conducted on the impact and effects of learning programming in children and youth. In 2008, a study [9] was carried out on young people between eight and eighteen years old belonging to a Computer Clubhouse using the Scratch block-based visual programming language. The study claims that Scratch became quite popular with young people, who learned basics of programming in a fun way by creating videos and games, with very spontaneous help from their mentors. An analysis of 425 projects was also carried out to understand what concepts they were learning, among which cycles and interactions with the user stood out.

The previous study shows that using a new and different approach can be beneficial and even entertaining. However, it should also be noted that not everyone has the same problem-solving skills. Some people may find it easier to learn programming with traditional methods, and others may require other special strategies. How to know what tool to use to teach what type of student? How is the strategy related to the student's emotions? This is why it is important to understand the differences and effects between learning programming using a block-based programming language and a text-based programming language as an introductory language to programming, as they are quite noticeable.

According to research carried out in Croatia [10] on fifty students of fifth grade from two primary schools without previous programming experience and with different levels of problem solving skills, interesting results were found regarding the understanding of programming concepts using the Python programming language and Scratch. The research mentions that those students with better problem solving skills performed better on tests with Python than those with an intermediate ability. However, in the Scratch results, these two groups of students performed quite similarly. Intermediate-level students were also compared with low-level students, and the result was that intermediate-level students performed better in both Python and Scratch, indicating that low-level students had programming problems with both approaches.

The research found that more than half of the students liked using Scratch much more and even their opinion improved regarding programming.

In addition to improving learning, the approach used can also tackle common problems in learning and problem solving, while the student learns. It is even possible to use hybrid approaches that make use of both digital technologies and traditional materials such as paper and colors. According to preliminary research [11] carried out in Croatia, it was proposed an approach designed to teach basic programming concepts to primary school children with problems such as difficulty concentrating, as well as the lack of problem solving strategies. The idea proposed by the research was to use a game and a simple problem of navigating a robot through a maze to find a prize, hiding the learning process. The approach consisted of using colored cards, which simulated the directions in which the robot could move through the maze. Then they used a tool called NetLogo, which is an integrated development environment for modeling, to simulate an environment similar to the one in the previous game. The objective of this was to link the two approaches and that the knowledge obtained with the color cards approach would help them solve the same problem, but with the NetLogo approach. The study concluded that children liked the activity and that only one solution created in NetLogo was incorrect among all the solutions, indicating the success of this approach using NetLogo.

This is where you find a learning approach that uses a very important factor: a game. Learning based on games and specifically on computer games is an approach that according to a comparative evaluation with traditional approaches [12], it does achieve that students have better retention and increased motivation, although in this particular case does not achieve better learning results. Motivation is key in learning; if students are motivated, they will have enough reasons to improve and expand their knowledge about what motivates them. In a study [13] focused on the IBM Robocode game, which is a programming game that simulates battles between robots, and conducted on the Robocode online community, it was found that more than half of the participants reported improvements in their programming skills when using such a game. Participants whose age ranges from youth to adults were also found to enjoy playing Robocode; one of the reasons they played it was because they found it fun. In another study [14] conducted on forty-two preschool children whose age ranged between five and a half to seven years old, and focused on an educational environment called PiktoMir, which, unlike other similar games, focuses on introducing a few basic programming concepts, it was found that although five of the seven children under the age of six did not pass the test, the majority did it. The study concludes that although some children disliked the similarity of the tasks to be performed at PiktoMir, half of the children wanted to continue playing and most of them found it fun.

There have been some other initiatives and studies in similar directions. For example, there was an initiative carried out in a workshop [15] for middle to high school students, which was aimed to increase the proportion of women in computer science by using robots. The workshop introduced programming concepts for robotic applications, through a presentation with some programming code in Python. The preliminary evaluation of the workshop showed that it greatly motivated students for learning programming concepts. There was also an experiment [16] carried out with sCool,

which is a multiplatform game that encourages children learning computational thinking and coding in Python. In the experiment, the students learn a given concept and they have to apply it in a practical task; two student groups participated in a formal learning activity; the study showed that the students are interested in learning to code but they have problems to transfer the learned content to similar fields. Another study [17] focused on exploring the potential merits of introducing concurrent programming concepts early in the learning process. The results of the study indicated that uninitiated to programming students at the age of ten years old were able to understand basic concurrency topics, while students at the age of eleven years old with some programming familiarity were able to understand more advanced concepts.

There are many approaches and factors that can be used; however, the constant of the visual factor can be found, which is of great importance when learning programming. It is common to be bombarded with presentations and books with lots of text and little visual material. However, it is important to know that there are other ways to learn and much more important to know that they work. It is also important to understand that the problem does not only lie in the type of material with which it is taught, but also in the students' abilities to solve problems.

4 Existing tools

In this section five different tools are analyzed, which share the main objective of teaching programming concepts, both to children and young people. They are available for mobile devices, desktop computers and laptops. Some of them use a block-based programming language and others use a text-based programming language. A comparative analysis of the tools is also provided at the end of the section.

4.1 Code combat

It is a game [18] that can be played through a web browser; it is for students older than nine years old, where they can learn to program in the Python and JavaScript programming languages. It has a classroom mode, where teacher and student can register in order to monitor progress. It is also possible to register with an individual account or play without registration. In order to access all the available content, the student needs to subscribe. It is possible to change the language between more than thirty different languages. When starting the game, the student is asked to create a character, as well as to choose the programming language to use; it is possible to change these settings later, according to the student's needs. Once the character is created, the student can start to access different levels, categorized by the type of content to learn. Selecting a level displays all the corresponding available missions. Clicking on one of the missions, the student can read the name of the mission, a description of the objective, the content to learn, a button to review the scores and a button to play. When playing the mission, the student can give objects to the character in order to grant her abilities (programming commands). Figure 1 shows the game in action; on the right side of the interface, there is a small editor where it is possible to

add, delete, edit and execute programming commands. The game is shown on the left side of the interface, where the code is executed. Once the objective of the mission is completed, the results are displayed on the screen.

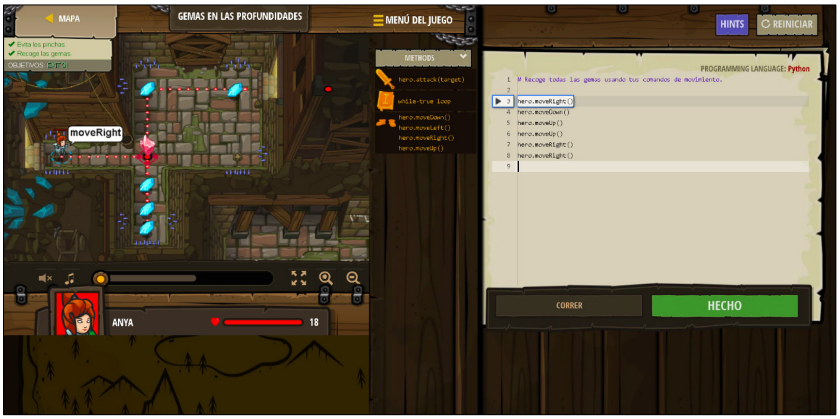

Fig. 1. Interface with the Code Combat game in action

The impact of Code Combat in schools is notorious. An example is that of technology teacher Scott Baily [19], who created the only coding class at Bobby Duke Middle School. The teacher began working with one hundred and eighty students of seventh and eighth grades. With Code Combat, Scott covers four Computer Science courses where students learn concepts like decision statements, functions, boolean logic, and more. Since they started using this tool, the students of this school had written 54,777 computer programs, 839,326 lines of code, and had carried out 613 network and game projects.

4.2 Coding adventure

It is a programming learning system [20] based on the CoffeScript programming language; it is focused on children eight years of age and older, which is part of the Code Monkey teaching platform. In order to access the platform, the user needs to subscribe; however, there is a thirty-day trial period for teachers and students, and fourteen days for other people. This system is available in more than twenty languages. The start screen of Coding Adventure shows a set of activities; when the student selects an activity, the game begins, as it is shown in Figure 2. There is an editor on the right side of the interface, where the student can write commands that the character will perform. On the left side of the interface there is the graphic part, which will reflect the executed code. Once the objective has been achieved, the results are displayed on screen. It is possible to select the level to play through a map, which indicates the number of levels, as well as the objectives to be achieved.

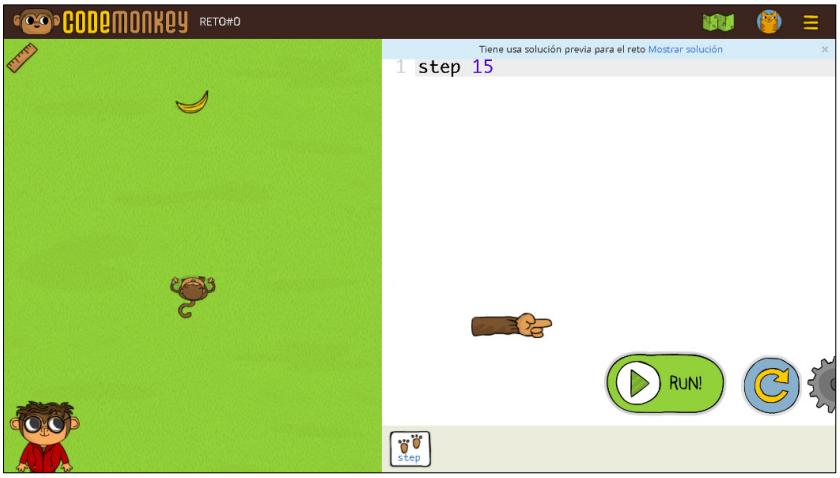

Fig. 2. Interface with the Code Adventure game in action

Regarding the impact that Coding Adventure has had as part of the Code Monkey platform, there is a case study [21] carried out by the academic integration specialist Kristen Fudale, who was looking for a way to teach elementary school children to code, and she found the solution on this platform. The specialist mentions that her students had fun solving the codes, working in pairs and using problem solving skills, critical thinking, among others; all these activities were carried out by her students while they saw them as a simple game.

4.3 Lighbot

It is a puzzle game [22] for children aged four and older that uses a block-based programming language; it is available on the App Store, Google Play, and Amazon. Lightbot has a menu, which focuses on learning a certain programming concept. Choosing a level, the student can select the mission to complete. Figure 3 shows the Lightbot game screen, which is divided into two parts: on the right side there is a box where blocks must be placed, which act as instructions (advance, rotate, turn on light); on the left side there is the graphic part, where the executed set of blocks will be reflected.

Once the set of blocks is executed and the goal is reached, the student can advance to the next mission. According to reviews of teachers and students of the Common Sense web site, Lightbot is widely used in schools. A teacher of the Eastside Elementary School shows an example of the use of Lightbot, where she introduced this game to her fifth grade students, who understood basic programming concepts such as algorithms and loops. Those same students visited their fourth grade peers during the Hour of Code, which is an international event focused on teaching and introducing Computer Science to students from around the world, in order to guide them as they had more experience [23].

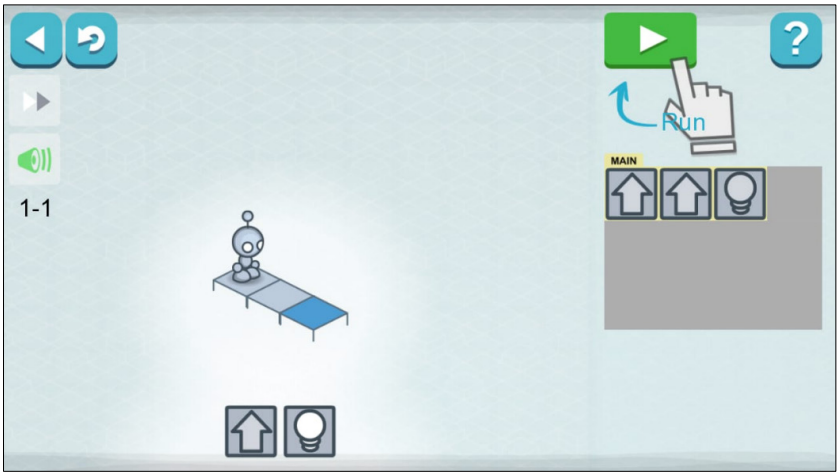

Fig. 3. Interface with the Lightbot game in action

4.4 Code karts

It is a puzzle game [24] for children aged four and older, where the goal is to get a small car to cross the finish line using block instructions. Figure 4 shows the interface of the game; the left side has a box with the set of instruction blocks that can be used to reach the objective, while on the upper side there is a box where the blocks must be placed in order to be executed.

Fig. 4. Interface with the Code Karts game in action

In the rest of the interface, the car can be seen at the start line, the path it can take and the finish line. If the car crashes due to an incorrect instruction block, it loses a

life which reduces the final performance rating. Once the level objective is achieved, the screen with the performance result is displayed.

4.5 Spritebox

It is an adventure game [25] for all ages, in which obstacles are overcome by coding either with blocks or by writing commands with the Swift or Java programming languages, in order to learn programming concepts such as sequencing, parameters, cycles, among others. It is possible to choose English or Portuguese as the language, the features of the character and the type of programming language with which the obstacles will be solved.

Figure 5 shows the interface of the game in action, with a student solving the obstacles with blocks; the right side of the interface shows the blocks that can be used (up, left, right, down); and the left side of the interface shows the game with some empty boxes, in which the student has to place the blocks.

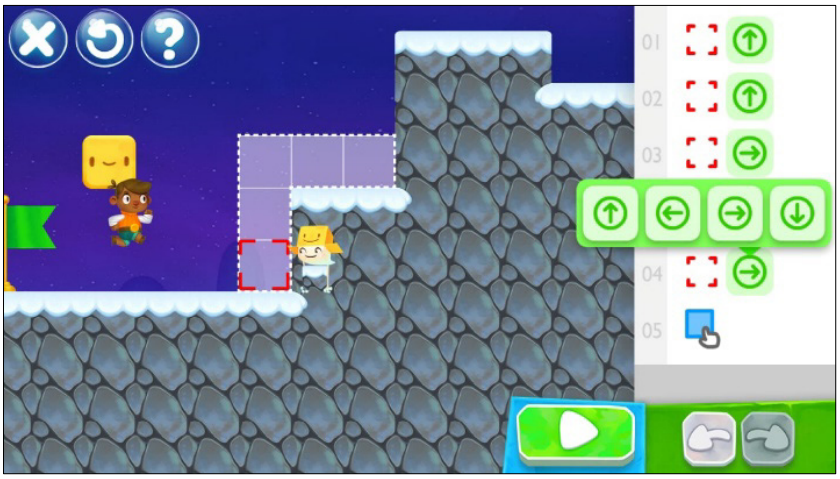

Fig. 5. Interface with the Spritebox game in action

4.6 Comparison table

This section provides an explanation of important features found after analyzing the previously presented tools, as well as the features that the proposed system is expected to have. Table 1 shows a comparison of these features. The tools shown in Table 1 are the following: T1) Code Combat; T2) Coding Adventure; T3) Lightbot; T4) Code Karts; T5) Spritebox; and T6) The proposed system. The tick indicates that the tool has the feature, while the cross indicates that the tool does not have it.

Table 1. Features of the analyzed tools and the one proposed

Feature	T1	T2	T3	T4	T5	T6
Administrator mode	×	×	×	×	×	✓
Map of levels	✓	✓	✓	✓	✓	✓
Target window	✓	✓	✓	×	✓	✓
Block and text	×	✓	×	×	✓	×
Help button	✓	✓	✓	✓	✓	✓
Reset button	✓	✓	✓	✓	✓	✓
RPG elements	✓	×	×	✓	✓	✓
Result screen	✓	✓	×	✓	×	✓
Shortest solution	×	✓	✓	✓	×	✓
Previous solution	×	✓	×	×	×	×
Language selection	✓	✓	✓	✓	✓	×
Sounds	✓	✓	✓	✓	✓	✓

Administrator mode. This feature is essential given the nature and objective of the proposed system. It allows the creation of levels at a visual and logical form, through an editor whose interface must be suitable to be used by a teacher.

Map of levels. This feature is considered important for the proposed system because it allows a good visualization and navigation of levels.

Target window. It is considered essential for the system, since the instructions are a fundamental part of solving a problem.

Block and text. This hybrid feature is interesting, as it aims to establish a relationship between the visual aspect and the text aspect. However, this feature is not considered for the proposed system because it is not taking into account the learning of a text-based programming language.

Help button. This feature is essential because it is not only found in all the tools reviewed, but also provides the necessary support so that the student can complete the objective and have a good user experience.

Reset button. It is essential since the student must be given the option to start again from the beginning at the same level if desired, without having to navigate between screens to find it again.

RPG elements. This term is quite broad, since even those tools that have this feature implement it differently. The main RPG element to be implemented in the proposed system is the character selection, in order to add some visual variety.

Result screen. This feature is considered important for the system because it helps the student to visualize his performance in the corresponding level.

Shortest solution. It is considered for the proposed system because it offers a learning factor. It shows the student the shortest solution for the given problem.

Previous solution. This feature is not considered for the system, since it was only found in one tool and did not contribute much to the user experience.

Language selection. It is not considered for the proposed system because it will only be available (for the moment and for the project purposes) in Spanish, since the demographic objective of the assessment tests are Mexican primary school students.

Sound. This feature is considered for the system, since it improves the user experience and allows creating a relationship between the sound and the action that takes place on the screen.

A more general review and comparison of educational games designed for teaching computer programming and computational logic can be found in [26]. Another review of emerging technologies in education is carried out in [27].

5 Proposed web system

This section presents the proposed web system, which is divided into two parts: the game generator that corresponds to the teacher part and the game itself that corresponds to the student part.

5.1 Game generator

The proposed generator has an interface through which the teacher can create and save custom boards and generate the necessary elements for its execution. The generator has the following modules for its correct operation.

Visualization of menu. This module corresponds to the initial interface of the generator. It has links to access all the features of the generator.

Creation of new game. This module corresponds to the creation of a new game. It allows the teacher to choose the number of boards that will be part of the game.

Creation of generic board. This module corresponds to the initial creation of the board. A logical representation of the board has to be created using a matrix that allows to allocate elements in their corresponding boxes; it is displayed at the time of editing. It allows the teacher to choose the basic features of the board, such as size and starting point.

Visualization of editor. This module corresponds to the visualization of the editor. It contains all the visual aspects required to create a playable board.

Logical board. This module corresponds to the logical result of the editor. The changes carried out in the editor are updated and saved in the previously created generic board.

Drag and drop board game elements. This module corresponds to the board edition and arrangement of elements. It allows to drag and drop elements on the board, such as obstacles and objectives.

Creation of the shortest solution. This module corresponds to the design of the shortest solution proposed manually by the teacher.

Board storage. This module corresponds to a small system that saves logical states that represent the board, in order to be able to regenerate it in the future if desired.

Generation of files. This module corresponds to the generation of files corresponding to the created game.

5.2 Game

The game has an interface that allows students to have access to the boards created by the teacher in the generator. The game has the following modules.

Visualization of start screen. This module corresponds to the main screen of the game. It has a link to the level selection screen.

Visualization and selection of levels. This module corresponds to the levels screen. Each level corresponds to a board and has a link to allow access to that board. In addition, it indicates whether the shortest solution has been achieved at each level.

Character selection. This module corresponds to a character selection interface. It allows the student to select a character from those that are available and that is reflected when playing a level.

Logical representation of the board. This module allows creating a sample board to carry out tests. A matrix is used to allow elements to be placed in their boxes.

Visual representation of the board. This module corresponds to the process of visually representing the previously generated board. The appearance of the board reflects the logical design of the board.

Drag and drop blocks. This module corresponds to the interaction with the blocks in the resource box. It allows to drag blocks and drop them in the boxes.

Logical execution. This module corresponds to the process of going through all the blocks of the execution box to determine the behavior of the character. It allows communication with the visual execution module to reflect the visual change on the board in real time.

Visual execution. This module corresponds to the visual representation of the execution of blocks. It has contact with the logical execution module to know how it should behave visually.

Restart the board. This module corresponds to the complete reset of the board. It loads the initial logic state and continue the game normally.

Verification of game status. This module allows verifying if the objective of the level has been accomplished. It goes through all the boxes on the board to determine if there are still targets to meet.

Verification of shortest solution. This module allows verifying if the solution used is the shortest solution proposed by the teacher. It compares both logical sequences in order to determine this.

Visualization of the result screen. This module corresponds to the visualization of the result after successfully completing a level. It provides links to repeat the level or advance to the next, as well as show whether or not the shortest solution was reached.

Manipulation of sound. This module allows to enable or disable the sound effects of the game according to the user's preference.

5.3 Interface prototypes

The design of the web system interface was represented thorough prototypes, which were created with the web tool NinjaMock [28]. The prototypes allowed greater clarity to develop the functionality of the system. The left side of Figure 6 shows

the interface with the levels available, as well as the performance obtained in each of them. The right side of Figure 6 shows the interface to select the character that represents the player in the game.

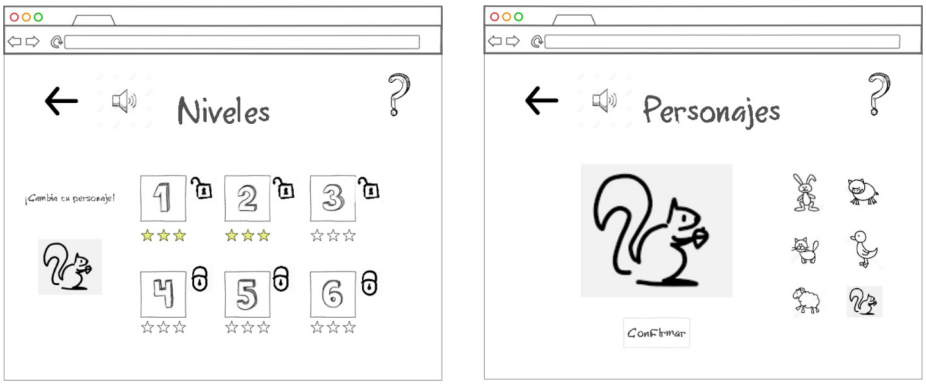

Fig. 6. Interface prototype to select the level and character to play a game

The left side of Figure 7 shows the game in progress. The objective is to make the character collect all the objectives, in this case acorns, to complete the level. This is done by entering block commands and executing them to control the behavior of the character. The right side of Figure 7 shows a small window that notifies that the level has been completed, as well as the performance achieved, and the option to continue to the next level.

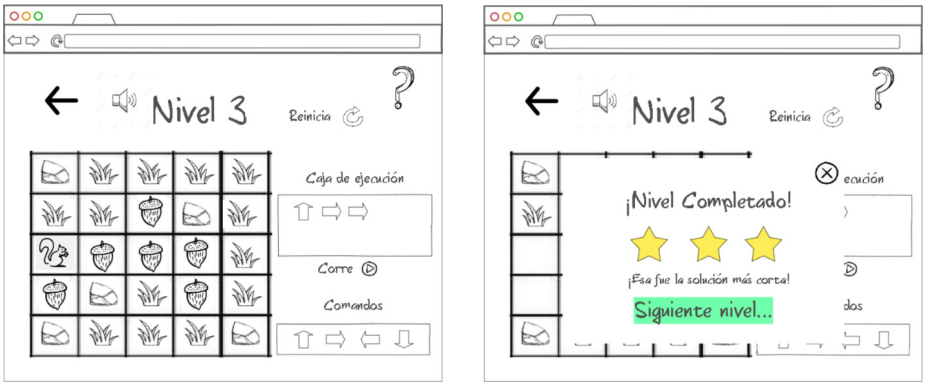

Fig. 7. Interface prototype with the game in progress

5.4 Preliminary game interface

Figure 8 shows the preliminary interface of the game. On the left side of the interface the board is shown, where there are different elements on it: the rabbit, which is the character that will be moving along the board; the objectives, which are the carrots; the obstacles, which are the holes; and the rest of the boxes have grass. The right

side of the interface contains the commands panel with the following actions: walk, jump, turn right, turn left, cycle and function. These commands can be dragged and dropped on the execution panel below the board, in order to determine the sequence of actions that the rabbit should follow in order to eat all the carrots on the board.

Fig. 8. Preliminary game interface with a board and commands to be executed

The function panel located below the commands panel, allows to create a function, which is a sequence of commands that can be used as a whole in the execution panel. It should be noticed that the execution panel shown in Figure 8 has five actions to be executed: jump, turn right, walk, jump and turn right.

The green button next to the board makes the rabbit to execute the sequence of actions specified in the execution panel. A correct sequence of actions will make the rabbit to eat all the carrots on the board, and avoid the obstacles. The system will show a result screen showing the shortest solution to that specific board.

6 Conclusions and future work

This paper presented a web system to generate programming games for primary school children, with the objective to teach basic programming concepts. The web system is divided into two main parts: the game generator that is used by teachers to create new programming games, and the programming games themselves that are played by students. The modules of each part were described, some interface prototypes were shown, as well as the preliminary game interface.

It was carried out a comparative analysis of five existing games with the same purpose, which use either block-based or text-based programming languages. The most relevant features were highlighted, such as the presence of a levels map, the use of a block or text language, the presence of a result screen, the presentation of the shortest and previous solutions, among others.

Further work is needed to complete the development of the web system in both parts: the game generator and the visualization of games. The web system will be

evaluated by teachers and students in three aspects: functionality, usability, design and didactic features. It is also planned to put the system on a web server so that teachers and students can use it.

Finally, it should be noticed that there have been other initiatives to support the teaching-learning process of computer programming at different levels, such as a web application that shows the execution of a program through a graphic visualization [29], a web application to create flowcharts [30], and a set of learning objects to support structured programming undergraduate courses [31].

7 References

[1] Reichert, R., Nievergelt, J., Hartmann, W. (2001). Programming in schools - why, and how? In C. Pellegrini, A. Jacquesson (Eds.): Enseigner l'informatique, pp. 143-152.

[2] Weintrop, D., Wilensky, U. (2015). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. Proceedings of the 11th International Conference on International Computing Education Research (ICER'15), pp. 101-110. https://doi.org/10.1145/2787622.2787721

[3] INEGI. (2019). En México hay 74.3 millones de usuarios de Internet y 18.3 millones de hogares con conexión a este servicio: ENDUTIH 2018. Press release 179/19.

[4] Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. ACM Inroads, vol. 2 (4), pp. 71-76. https://doi.org/10.1145/2038876.2038894

[5] Liao, Y.K C., Bright, G. W. (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. Journal of Educational Computing Research, 7 (3), pp. 251-268. https://doi.org/10.2190/E53G-HH8K-AJRR-K69M

[6] Caccuri, V. (2013). Educación con TICs. 1st ed. Buenos Aires: REDusers. https://issuu.com/redusers/docs/educacion-con-tics

[7] Bau, D., Gray, J., Kelleher, C., Sheldon, J., Turbak, F. (2017). Learnable programming: blocks and beyond. Communications of the ACM, 6 (6). https://doi.org/10.1145/3015455

[8] Bain, G., Barnes, I. (2014). Why Is Programming So Hard to Learn? Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference (ITICSE '14), Uppsala, Sweden. http://dx.doi.org/10.1145/2591708.2602675

[9] Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., Rusk, N. (2008). Programming by choice: urban youth learning programming with Scratch. Proceedings of the 39th ACM Technical Symposium on Computer Science Education (SIGCSE '08), 40 (1), pp. 367-371. https://doi.org/10.1145/1352322.1352260

[10] Mladenović, M., Krpan, D., Mladenović, S. (2017). Learning programming from Scratch. Proceedings of the International Conference on New Horizons in Education 2017, Berlin, Germany.

[11] Zaharija, G., Mladenović, S., Boljat, I. (2013). Introducing basic Programming Concepts to Elementary School Children. Procedia – Social and Behavioral Sciences, vol. 106, pp. 1575-1584. https://doi.org/10.1016/j.sbspro.2013.12.178

[12] Egenfeldt-Nielsen, S. (2005). Beyond Edutainment Exploring the Educational Potential of Computer Games. PhD Thesis, IT-University of Copenhagen, Copenhangen, Dinamarca.

[13] Long, J. (2007). Just for Fun: Using Programming Games in Software Programming Training and Education — A Field Study of IBM Robocode Community. Journal of Information Technology Education, vol. 6, pp. 279-290. https://doi.org/10.28945/216

[14] Rogozhkina, I., Kushnirenko, A. (2011). PiktoMir: teaching programming concepts to preschoolers with a new tutorial environment. Procedia - Social and Behavioral Sciences, vol.28, pp. 601-605. https://doi.org/10.1016/j.sbspro.2011.11.114
[15] Keller, L., John, I. (2020). Motivating Female Students for Computer Science by Means of Robot Workshops. International Journal of Engineering Pedagogy (iJEP), 10(1), pp. 94–108. https://doi.org/10.3991/ijep.v10i1.11661
[16] Steinmaurer, A., Pirker, J., Gütl, C. (2019). sCool – Game-Based Learning in Computer Science Class: A Case Study in Secondary Education. International Journal of Engineering Pedagogy (iJEP), 9(2), pp. 35–50. https://doi.org/10.3991/ijep.v9i2.9942
[17] Fatourou, E., Zygouris, N. C., Loukopoulos, T., Stamoulis, G. I. (2018). Teaching Concurrent Programming Concepts Using Scratch in Primary School: Methodology and Evaluation. International Journal of Engineering Pedagogy (iJEP), 8(4), pp. 89–105. https://doi.org/10.3991/ijep.v8i4.8216
[18] Code Combat. The most interesting game to learn programming. Available: https://codecombat.com
[19] Bobby Duke Case Study. Bobby Duke Middle School. Available: https://codecombat.com/images/pages/impact/pdf/CodeCombat_CaseStudy_BobbyDukeMS.pdf
[20] Coding Adventure. Teach text-based coding by helping a monkey to catch bananas. Available: https://www.codemonkey.com/coding-adventure/
[21] Fudale, K. (2016). Case Studies for CodeMonkey. Ohio, USA: Edsurge. Available: https://www.edsurge.com/product-reviews/codemonkey/educator-case-studies
[22] Lightbot. Puzzle game based on coding. Available: https://lightbot.com
[23] Jana, C. (2016). Teacher Review for Lightbot: Programming Puzzles. Commonsense. Available: https://www.commonsense.org/app/lightbot-programming-puzzles-teacher-review/4132306
[24] Code Karts. A fun app to develop observation, concentration and logic. Available: https://montessori.edokiacademy.com/en/our-games/discovery/car-game
[25] Spritebox. A full-blown adventure game that gets you coding. Available: https://spritebox.com
[26] Pereira, J., Frango, I. (2020). A Systematic Review on Open Educational Games for Programming Learning and Teaching. International Journal of Emerging Technologies in Learning, 15 (9), pp. 156-172. https://doi.org/10.3991/ijet.v15i09.12437
[27] Sosa, E., Salinas, J., Benito, B. (2017). Emerging Technologies (ETs) in Education: A Systematic Review of the Literature Published between 2006 and 2016. International Journal of Emerging Technologies in Learning, 12 (5), pp. 128-149. https://doi.org/10.3991/ijet.v12i05.6939
[28] NinjaMock. Online wireframe and mockup tool. Available: https://ninjamock.com/
[29] Jaimez-González, C. R., Castillo-Cortes, M. (2020). Web Application to Support the Learning of Programming Through the Graphic Visualization of Programs. International Journal of Emerging Technologies in Learning, 15 (6), pp. 33-49. https://doi.org/10.3991/ijet.v15i06.12157
[30] Vazquez-Peñaloza, F., Jaimez-González, C. R. (2019). Towards a Web Application to Create Flowcharts for Supporting the Teaching-Learning Process of Structured Programming Courses. American Journal of Educational Research, 7 (12), pp. 976-982. http://www.sciepub.com/education/abstract/11332
[31] Luna-Ramírez, W. A., Jaimez-González, C. R. (2014). Supporting Structured Programming Courses Through a Set of Learning Objects. Proceedings of the IEEE International Conference on Information Society (i-Society 2014), pp. 124-128, London, UK. https://doi.org/10.1109/i-Society.2014.7009024

8 Authors

Javier Erazo-Palacios is a mobile and web application developer. He is studying a BSc degree in Information Technologies and Systems from the Universidad Autónoma Metropolitana Campus Cuajimalpa. His research interests include technologies for supporting education, mobile and web application development.

Carlos R. Jaimez-González is a Professor at the Information Technology Department at the Universidad Autónoma Metropolitana Campus Cuajimalpa, in Mexico City. He received his PhD degree in Computer Science from the University of Essex, United Kingdom in 2011. His research interests include technologies for supporting education, interoperability in distributed systems, XML and related technologies, and the development of web and e-commerce applications. He has a distinction as a national researcher from the Mexican Government.

Betzabet García-Mendoza is an Associate Professor at the Information Technology Department at the Universidad Autónoma Metropolitana Campus Cuajimalpa, in Mexico City. She received her MSc degree in Design, Information and Communication from the Universidad Autónoma Metropolitana. Her research interests include technologies for supporting education and web application development.

Article submitted 2020-07-28. Resubmitted 2022-04-08. Final acceptance 2022-04-09. Final version published as submitted by the authors.