Programming Teaching Experience for Boys and Girls following a Gamified Approach

Iago Cruz-García
Computer Science and Statistics Dept.
Universidad Rey Juan Carlos
Móstoles, Madrid, Spain
i.cruz.2016@alumnos.urjc.es

Juan Antonio Martín-García
Computer Science and Statistics Dept.
Universidad Rey Juan Carlos
Móstoles, Madrid, Spain
ja.martin.2016@alumnos.urjc.es

Diana Pérez-Marín
Computer Science and Statistics Dept.
Universidad Rey Juan Carlos
Móstoles, Madrid, Spain
diana.perez@urjc.es

Celeste Pizarro
Applied Mathematics Dept.
Universidad Rey Juan Carlos
Móstoles, Madrid, Spain
celeste.pizarro@urjc.es

***Abstract*—Teaching programming in Primary Education is a worldwide research interest topic. Currently, there are several approaches that are being analyzed with more or less success in terms of learning gains, levels of motivation, and satisfaction. In this paper, the proposal is to use a gamified approach. Hypothesis 1 (H1) is that the gamified approach will increase students' learning gains while also increasing their level of satisfaction and motivation. Hypothesis 2 (H2) is that there will be no significant differences between boys and girls from 10 to 12 years old. During the 2019/2020 academic year, an experiment was carried out with 100 students from 10 to 12 years old. All students have started their programming learning through a gamified approach, varying the resource used (teacher, Blockly, Blockly + video game). The results gathered support H1 and H2 and yield interesting results to continue analyzing the differences between the resources used.**

Keywords — Teaching programming, videogame, gamification, Primary Education

I. INTRODUCTION

Teaching programming in early ages has received a lot of attention worldwide in the latest decades [1]. Some authors assert that children could improve their understanding of the digital world where they live and even develop their computational thinking [2] to be able to solve life problems with computer resources.

This paper focuses on teaching programming to improve primary education children's programming abilities, while they enjoy the learning process and keep their motivation. To achieve this, the proposal focuses on teaching programming in Primary Education by using a gamification approach.

Gamification can be defined as using game design elements in non-gaming environments to increase motivation and satisfaction. Previous research show that gamified environments usually get better results in education than non-gamified ones, as long as the design is done correctly. This consists of introducing rewards, game mechanisms and global design, not only isolated elements [3].

The proposal focuses on using resources as metaphors [4] as if programming was cooking, Blockly [5] and videogames [6]. Using videogames for educational purposes has proven to successfully develop high level abilities in students such as solving problems, teamwork, spatial vision development as well as to reduce stress and develops motor skills [3, 6].

However, to maintain gamification it is important to highlight the need of stablish concrete and achievable goals for the player [7] and challenges with incremental levels of difficulty.

Hypothesis 1 (H1) is that the gamification approach for teaching programming in Primary Education will improve students' learning scores as well as their satisfaction and motivation levels. Hypothesis 2 (H2) is that there will not be significant differences between boys and girls from 10 to 12 years old when learning to program with gamification.

To verify these hypotheses during 2019/2020 school year, an experiment has been done with 100 students from 10 to 12 years old taking 4th, 5th and 6th school years of Primary Education, from October 2019 to February 2020 before the lockdown stopped from continuing in-person data gathering.

During the experiment, all students have followed a gamification approach to understand basic programming concepts: input/output, conditionals and loops, varying the gamified resources: a teacher suggesting goals and challenges in class playing with metaphors to teaching programming, Blockly for introducing basic programming concepts, and a videogame developed as final degree work from first author during 2019/2020 school year.

Research questions are the following ones:

- Q1. Which differences exist between results of different groups?
- Q2. Which differences exist between results from boys and girls?

Results gathered support H1 and H2. This paper is structured in four sections: Section 2 presents the proposal, Section 3 describes the experiment, and Section 4 ends the paper with main conclusions and future lines of work.

II. PROPOSAL

A. *Gamified use of metaphors with teachers*

To teach programming teachers need a guide. In our previous work, it was proposed the use of scripts step by step based on metaphors [4]. Underlying concept is that children between 10 and 12 years old usually enjoy cooking, and programming could follow the cooking metaphor.

In both cases there is a goal, make food or achieving program's goal, and to get to that point instructions must be followed step by step. In case of food, instructions would be the recipe steps. Order is important because if, for example, we wanted to cook a potato omelette and oil is not hot, potatoes will not be fried correctly and omelette will not taste good. Likewise, if we want to make a program achieve its goal and we do not follow order of output instructions, the output will not be the expected one.

Table I shows a summary of main metaphors used [4].

TABLE I. SUMMARY OF THE METAPHORS USED

#	Subject	Concept	Metaphor
1	Program, programming, sequence, memory	Program-sequence	recipe
		Memory, variables	Pantry, box
2	Intput / output instructions	Output	Computer Screen
		Input	Keyboard, computer screen result
3	Conditionals	Child taking choices	Computer taking choices
4	Loops	Preparing table for X persons	Computer repeating instructions X times

The playful component as was stated in the introduction has to be planned with concrete goals [7] and challenges with increasing difficulty levels [8]. For this reason, it is proposed that the teacher not only uses metaphors, but that they propose challenge to children where they have to use those metaphors to solve a challenge, such as making a banquet and using pseudocode programming on paper for that purpose, as explained in [9].

Once children get certainty and rewards like being congratulated by the teacher, acknowledgement from their equals in class and some stickers in a physical medal board, then challenge difficulty can be increased. This way, if they could achieve to make the program that allowed them to make the banquet, the following step could be making a program that allows them to organize weekly menus of a fantasy kingdom that could be created by teacher and students.

B. *Use of Blockly*

Blockly [5] is an open source visual programming language in which programs are built using instructions as if they were puzzle pieces. It is similar to Scratch because it also focuses on using blocks, as it shown in Figure 1.

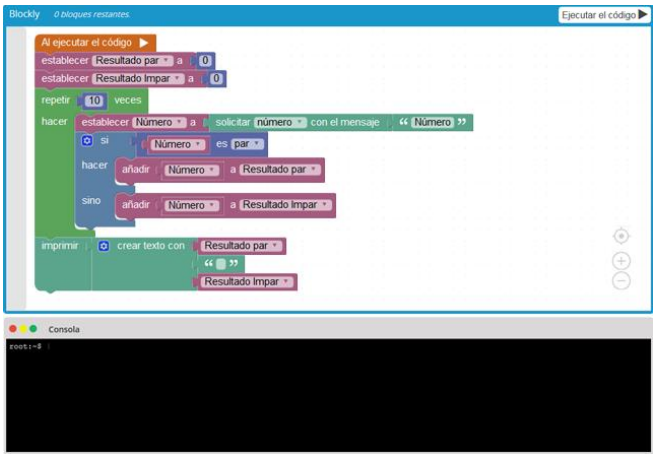

Fig. 1. Blockly screenshot (source: [5])

Blockly has been chosen over Scratch because, in the opinion of the authors and our proposal, because students have a console they could use as if it were a professional programming environment. It is also likely that students from 10 to 12 years old they have already used Scratch in previous years (in example, between 8 and 10 years old), therefore continuing with Blockly allows them to go deeper into knowledge about input/output instructions, conditionals and loops.

Gamification integrates itself in Blockly by elements previously described: concrete goals [7] and increasing difficulty challenges [8]. Goals are given by the teacher because Blockly is a blank canvas for the student to program, but it does not have a guide to show them what to do. Same goals as case II.A are proposed, related to input/output, conditionals and loops concepts.

In particular, exercises' statements requested to children as programming challenges in Blockly are:

- Write a program to show your name in screen (program concept, sequence, I/O).
- Write a program to ask how old a classmate is, and show it in screen (program concept, sequence, I/O, variable).
- Write a program to check if a number is odd or even (program concept, sequence, conditional, I/O, variable).
- Write a program to add, subtract, multiply and divide two numbers (program concept, sequence, conditional, I/O, variable).

As in II.A case when boys and girls successfully completed exercises, they got a congratulation and acknowledgment from the teacher. As students were completing exercises, they were ordered in a way that challenges had an increasing difficulty.

C. *Educational videogame use*

Develop Learning is the proposed videogame designed and developed to satisfy the needs of students who want to learn programming. Figure 2 shows an example of a screenshot of the videogame.

Fig. 2. Develop Learning screenshot (source: own elaboration)

When students were asked if they wanted to have help in the system, most answers told that they wanted to have an animal pet. This is why in the videogame the user represented by the character (4) comes along with a dog as a pet named Binary (2) which guides them through challenges that has to overcome through pseudocode programming.

The pet (2) shows texts (1) to students so they do not have a blank canvas as it happens in Scratch or Blockly, which both of them, despite having tutorials, they do not have accurate instructions about what to do. In easier levels of the videogame, the own videogame shows the code (3), while in harder levels the difficulty increases and students have to write the code to overcome the challenges that are introduced through the different worlds where they are moving.

When application boots, the student will find a main menu consisting of:

- Play menu, where they can access the tutorial and different worlds and levels.
- Settings menu, where they can change interface colours in minigames.
- Help menu, where they can check concepts in a comprehensive way with examples. This menu can also be accessed during minigames.
- Exit button to shut down the application.

The videogame interface was made in a way that it could be used without mouse and keyboard at the same time. In other words, when the user interacts with menus, they do not need to use the keyboard and when they are in a minigame, they do not need to use the mouse. They game also does not require internet connection. These choices were made because not all schools have a good internet connection, and also, laptops are usually used, so using simultaneously mouse and keyboard could get tedious.

For each main concept to teach, there is a world: I/O, conditionals and loops. All worlds shared the same structure: four minigames (descriptions, choosing, reorder and fill in the gaps), a review level and a boss as a final challenge as it shown in Figure 3 and as it was requested by school students as the users following a User Centered Design [10].

Fig. 3. Input/output level (source: own elaboration)

For each minigame, the interface is the same: the console with the code is in the upper right corner and, if there is an explanation, it would be located as a text box at the bottom. In boss levels, the text box with explanations and dialogs is in the lower left corner to leave room to the text area for writing code. Every time a minigame is completed, stars get coloured as part of the gamified approach to rewards.

In the first level minigames are focused on input/output instructions. In description minigames, the player will have to read closely descriptions in the console in the upper right corner and introduce in the well the corresponding word.

If they succeed, the block will stay in the well. The player will have to do this task five times in total. If they chose a wrong block, this and every block inside the well will be thrown (as in a parabola to avoid coming back to the hole) and they will have to start again.

This is based on that, when starting a learning process, attention should be paid to concepts that are being studied. One way to achieve this is to describe those words that have different meaning in the current context to understand further explanations and teachings.

In the choosing minigame, the player must discern the program output, falling through a hole between three possible ones. If they fail, they will come back to the beginning of that floor. If they succeed, they will go to next floor, until they reach the end of the level. The intention is to improve abstract thinking, to understand possible environments and step ahead of solutions to take apart knowledge from context, which means, not to memorize exercises.

In the reordering minigame, the player will have to place blocks on stands which match with the position shown in the console. The order of the console and stands match with occidental reading order: the console goes from top to bottom and stands go from left to right. Likewise, each block has the same colour as the corresponding space to arrange in the console.

The user should look at the console on the upper right corner and at the colour of the text that they should order, because they match with the colour blocks and stands on the scene. From left to right, they should place the block in their equivalent position on the console, from top to bottom. If they are sure about the result, they should press ‘F’ key near the button at the back of the level to check it. If they fail, blocks will disarrange.

An important skill to solve problems is the ability to organize elements that compose them. As in mathematics the first thing that is tasked in an exercise is to group known data or in morphosyntactic analysis to locate subject and predicate first, in programming it should be logically structured with variables at the start to use them in later instructions.

In fill in the gaps minigame, the player should enter the blocks in the stablished order. If they are right, they will light up in green, otherwise in red. This tries to emulate usual exercises, made to learn a language, filling empty spaces with a word from an array so the sentence makes sense. This minigame seeks to complement the previous ones, because at this point, the user should understand the structure and instructions purposes.

In the review minigame, the player will go through the four previous levels in order and in a reduced way. In other words, they should complete description, choosing, reordering and fill in the gaps minigames to succeed.

In this level, every previous minigame will appear in reduced versions, to build on what has been learnt and to get ready for the final level against the boss. It is done to be similar as the previous preparation for an exam, to remember what they have learned and to face the test with fresh knowledge.

In the boss minigame, the player should pay attention to the helper dog Binary indications, and the boss, to manually write code to advance. If the player makes a mistake, a basic hint will be offered, a guide of what is required. If they make another mistake, an explicit hint will be offered. Another mistake, the player should start again the level.

In these levels, the player should write, in the designated area, instructions to complete the needed program, when they are instructed to until the end of the level. Throughout this level, if the player fails a hint will be offered to know what they should write. If they fail again, another explicit hint will be offered. At third fail, they will restart the level.

This minigame was inspired by exams and final knowledge tests to put the student in a light pressure to address the situation in a playful way. This tries to fulfil the need of concrete objectives and challenges to get good results with videogames [7,8].

III. EXPERIMENT

A. Sample

100 Primary Education students from 10 to 12 years old has participated in the experiment. 46% of them are boys and 54% are girls, from two public schools from Madrid.

94% of students answered the user profile survey, used as reference for designing and developing the gamified system. In this user profile survey, there were questions about how familiar they were with technology and videogames, their personal preferences, how they handled challenges and their willingness to programming.

The goal of these questions was to establish how to approach the design of help systems, menu navigation, how long where game sessions and what goals had to be accomplished. Another goal for these questions was to establish the structure and dynamic for the programming classes since these concepts are new for these students.

B. Procedure

During the research, students were allocated in three groups: students using Blockly + videogame, students using Blockly and students taught by the teacher.

All groups followed a gamified approach. Since the videogame was in development during these classes were being taught, first sessions were covered always by Blockly (except for the group taught by the teacher, which only used blackboard, chalk, paper and pen) and it was since January when videogame could also be used for the Blockly + videogame group.

Every group made a programming pre-test (October 2019) to check initial concepts that could have been obtained externally, and also to check how old students were and what devices they usually use. For resolving this test, it was required to have knowledge about the following concepts:

- Implementation of an output instruction.
- Implementation of an input instruction and an output instruction.
- Implementation of output instructions specified by conditional blocks.
- Implementation of an infinite loop, with output instructions.

This test was presented again to students after four sessions (December 2019) to check their progress until that date, reviewing differences between groups.

These sessions, in the three groups, consisted in lessons about the following basic programming concepts: variable, memory, instruction, sequence and program. Groups also worked in exercises, with previously mentioned resources, requesting to introduce the input and output flow of a program, with exampled based on natural language that allowed students to associate familiar concepts with the ones that were being presented to them.

Class dynamic was structured always in the same way: it was started with a recap of previously explained teachings, after that, a new definition was introduced, and then students had to do a new practical exercise to solidify acquired concepts, solving any doubts that could be asked.

The same test was presented again to students in February 2020, just before the lockdown due to COVID-19, which disallowed to continue with the investigation, since every on-site class in Spain was cancelled. The goal was to check the evolution in grades of students from each group to evaluate the gamified approach and if there were any differences between boys and girls.

C. Results

In the first place, a descriptive study of data is made, which shows grades obtained in the test made in different dates: October 2019, December 2019 and February 2020.

Figure 4 shows box plots regarding the punctuation variable, separate by the different proposals. Table II sums up the numerical results. This table does not show the median, since its value its always 0.00. As can be seen, in October 2019 students did not know how to correctly answer any question.

Subsequently, in December 2019, student groups that used Blockly and Blockly with Develop Learning showed a similar behaviour. Worst average results were obtained from control group.

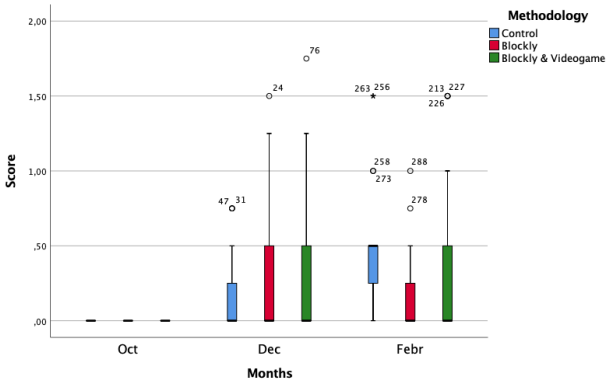

Fig. 4. Boxplots of the scores per methodology

TABLE II. AVERAGE AND STANDARD DEVIATION OBTAINED FROM STUDENTS' SCORES IN THE TESTS

n=100	December		February	
	$\bar{x}$	sd	$\bar{x}$	sd
Control	0.18	0.26	0.51	0.46
Blockly	0.35	0.46	0.19	0.27
Blockly + Video-game	0.32	0.43	0.30	0.45

It should be mentioned that the three distributions are strongly asymmetrical to the right, which means that there is a greater clustering of students with low grades in every case. The median in the three cases is 0, but the average is higher due to this asymmetry.

However, in February 2020, the control group's data distribution becomes symmetrical to the left, which means that there is a high clustering of high grades over low grades. Therefore, even if there are a lot of atypical data, median shows that this control group is the one that got better results. There is a greater scatter in data from Blockly and videogame groups.

Related to meaningful differences that could be between different groups, and on the basis of a scenario where there are not any previous learning from students as can be seen in their grades, a process to check if there are differences between groups is started.

For that purpose, a two factors with interaction Anova has been performed, where one factor is the month when the test was performed and the other one is the group (teacher, Blockly, Blockly + videogame), and the obtained grade as the answer variable. Table III shows the obtained results.

It can be seen that there are no isolated meaningful evidence between groups, but there are in months. A more detailed analysis of this result is found in a post-hoc analysis. This analysis shows that the meaningful difference between grades is found related to October, in other words, there are no meaningful differences between December and February grades, but between October and December and between October and February, something that was already guessed in the previous descriptive study.

TABLA III. ANOVA OF TWO FACTORS WITH INTERACTION

	Sum of squares	gl	F	Sig.
Month	5.94	2	26.22	0.00
Group	0.08	2	0.38	0.68
Interaction	1.66	4	3.67	0.06

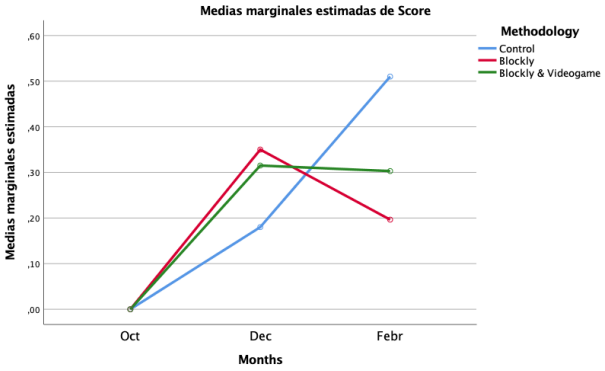

Fig. 5. Marginal averages of the scores obtained

Furthermore, this significative interaction shows that depending on the month, the efficiency of each group has behaved differently. Thus, considering only December data, there is not a meaningful difference between groups, but a difference in February can be noticed, with control group being the best positioned, followed by the one with Blockly and Develop Learning and in the last place the one with Blockly. Figure 5 shows this fact.

When the study is made splitting between boys and girls, it doesn't show meaningful differences between groups. In October 2019 test case, both boys and girls got an average score of 0 because they didn't have any knowledge about programming. In December 2019, both girls and boy average score was 0,28 and in February 2020, average score for boys was 0,31 and for girls 0,27.

IV. CONCLUSIONS

Gamified approach for programming teaching in Primary Education has significantly improved the learning profit both from students taught by the teacher (unplugged approach) and from the ones with Blockly and the videogame. This validates the importance of teaching using playful resources with increasing difficulty challenges.

Answering the research question about what differences exist between results from different groups, against expected, better results are obtained by students from the group with teacher, without Blockly and the videogame.

Answering the research question about what differences exist between results from boys and girls, there are not meaningful differences between groups. This means that both girls and boys can learn programming using gamified approach. This result is very important, especially for girls from 10 to 12 years old who could think, based on social

stereotypes, that they couldn't enjoy videogames or that they couldn't be able to reach the same performance as their male classmates.

As future work, it is planned to continue the study during more time when in-site classes can be resumed if COVID-19 allows to it. In addition, it is planned to add the age component to the study, comparing obtained results in a larger age range, from 10 to 15 years old, to identify if there are meaningful differences and interactions between gamified resources, age and gender components.

ACKNOWLEDGMENTS

Research funded by the TIN 2015-66731-C2-1-R project and the Madrid Regional Government, through the e-Madrid-CM project (P2018/TCS-4307). The e-Madrid-CM project is also co-financed by the Structural Funds (FSE and FEDER). We would also like to thank the school, teachers and students for their collaboration.

REFERENCES

[1] Computer Science Teachers Association. Computer Science K–8: Building a Strong Foundation. 2012. Available from: http://csta.acm.org/Curriculum/sub/CurrFiles/CS_K8_Building_a_Foundation.pdf.

[2] J.M. Wing. "Computational thinking". Communications of the ACM, 49(3), pp. 33–35, 2006.

[3] C. Groening, C. Binnewies. "Achievement unlocked! - The Impact of digital achievements as a gamification element on motivation and performance". Computers in Human Behavior, Volume 97, pp. 151-166, 2019.

[4] D. Pérez-Marín, R. Hijón-Neira, M. Martín-Lope. "A Methodology Proposal based on Metaphors to teach Programming to children". Revista: IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 13(1), pp. 46-53, 2018, https://doi.org/10.1109/RITA.2018.2809944

[5] Blockly, https://blockly.games/?lang=es (fecha última visita: 06/2020)

[6] C. González-González, F. Blanco-Izquierdo. "Designing social videogames for educational uses". Computers & Education, Volume 58, Issue 1, pp 250-262, 2012.

[7] K. Øygardslia, P. Aarsand. "Move over, I will find Jerusalem": Artifacts in game design in classrooms". Learning, Culture and Social Interaction,Volume 19, pp. 61-73, 2018.

[8] M.J. Dondlinger. "Educational video game design: A review of the literature". Journal of applied educational technology, 4(1), pp. 21-31, 2007.

[9] A. Bacelo-Polo, R. Hijón-Neira, D. Pérez-Marín. Piensa y programa con Scratch…en casa y en clase. Anaya. 2018.

[10] J. Lorés (Ed.). La Interacción Persona-Ordenador. Disponible on-line en https://aipo.es/libro/pdf/00Portad.pdf (fecha última visita: 06/2020)