

Evaluating the effectiveness of educational games: a digital game-based approach to teach programming concepts for kindergarteners

Tancicleide C. S. Gomes¹, Fernanda C. de Castro¹, Andreza Alencar²

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)

Recife – PE – Brasil

²Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco (UFRPE)

Recife – PE – Brasil

{tancicleide.gomes, mfernandacastro, andreza.lba}@gmail.com

Abstract. *The development of computational thinking since primary education has been defended over the last few years. In this context, the present work sought to evaluate the effectiveness of educational games through a teaching experience based on digital programming games for children. Finally, these games are evaluated by usability heuristics adapted to educational software. Obtained results indicate the learning of some concepts in a satisfactory way, however few relevant artifacts are pointed out to support the programming teaching in this age group, besides difficulties in the use experience.*

1. Introduction

The fundamentals of Computer Science have been incorporated in the educational curriculum of several countries, having as main exponent the programming topics. In the United Kingdom, for example, programming languages are part of the core curriculum and children since five years old are already contemplated in this discipline [James 2013]. In Australia, primary schools began to replace History and Geography classes by programming classes [Mendonza 2015]. Reference curricula establish the contents, learning objectives, and the skills and competences to be developed (*Computing at School*¹, *CSTA K-12 Computer Science Standards*²).

A set of initiatives and projects around the world involve the presentation of Computer Science through programming. The *Hour of Code* movement³, for example, aims to demystify the learning of programming through digital games inspired by elements of juvenile scenario. Moreover, there are a large number of games available in mobile and web application stores that address programming topics, reflecting the theme relevance.

¹ <http://www.computingatschool.org.uk/cacfs>

² https://www.csteachers.org/page/CSTA_Standards

³ <https://studio.code.org>

Nevertheless, it is necessary to evaluate these games in terms of their educational effectiveness. This study, in turn, evaluates the efficacy of digital games through a teaching experience of programming concepts for children aged four to seven in the second semester of 2015. Obtained results indicated that some digital games were effective to support the understanding of proposed concepts. However, they also suggested that gaps related to the comprehension of some game interaction elements have impacted the understanding of programming concepts. This paper is organized as follows: Section 02 presents a brief overview of teaching and learning experiences of programming concepts for children. In Section 3, related works are introduced. Section 4 presents the methods, the context in which this research was carried out, the participants and the games adopted. In Section 5, the main results will be presented and discussed. Final considerations and possible future work are listed in Section 6.

2. Conceptual Background

Games are considered engaging for a number of reasons: they provide something to be done, satisfaction, intense and passionate involvement, and learning [McGonigal 2012 *apud* Gomes, Melo e Tedesco 2016]. Players feel challenged as they develop skills and experience possibilities without being embarrassed by making mistakes before their peers. In addition to that, games promote intrinsic motivation, engagement and are also able to capture people's attention, providing meaningful and playful interaction. When games are inserted into learning spaces, students actively participate in their learning while having fun.

For example, in a study carried out with 300 children using *Gameboy*⁴ inspired games at Chicago University (USA), both kids familiar with technology and those who did not have access to it outside school were highly motivated [Chiesa 2014]. Another study with professors aimed to understand their opinions about the limits and potential of electronic games: results show that games are considered important elements in the creation of teaching strategies [ibidem].

The use of games in teaching experiences (called as *digital game-based learning*) aims to support learning, performance evaluation and to provide improvements in education. In this way, game elements and features guide the design of pedagogical proposals to make them more immersive and meaningful for students. This study is anchored in these premises and evaluates a programming teaching experience based on digital games for children in primary school.

3. Related Work

The teaching of Computer Science fundamentals in Basic Education has not yet been consolidated nationally. In Brazil, the researches does not represent a coordinated and integrated effort, especially when it comes to teach for children. We have searched for related work published in the annals of two major national conferences in the last ten years. The aforementioned conferences are the “*Workshop sobre Educação em Computação*” (WEI) and the

⁴ Nintendo®'s game console

“Workshop de Informática na Educação” (WIE). Most of these studies were done by initiatives from the *Institutional Program for Teaching Incentive Scholarships (PIBID)*.

Much of these initiatives were carried out in public schools as an extracurricular activity. In these studies predominates the use of digital games, visual programming languages and Computer Science Unplugged activities [Bezerra e Dias 2014, Machado, Vasconcelos e Malta 2010, Oliveira *et al.* 2014, Silva *et al.* 2015, Souza *et al.* 2014].

The studies found are mainly concentrated on actions developed in primary school with children aged between seven and fourteen years. Only two studies focused on preschool children [Gomes, Melo e Tedesco 2016; Torezani, Chagas e Tavares 2013]. No studies whose contents were guided by a curriculum for teaching Computer Science in basic education were found in this age group. Another important aspect: among the works retrieved, no digital game-based approaches were found (DGBL, *digital game-based learning*). The digital games emerge as complementary elements: sometimes as a support for other approaches, sometimes as a final product with methodologies whose goal is the development of small games.

Thus, this work is distinguished by presenting a teaching experience of programming concepts for children in early childhood education through a digital game-based approach. This research was developed as part of the school's computer classes. Contents, skills and competences addressed in this research were outlined from a reference curriculum.

4. Study Design

This exploratory study was developed during a university extension project, as part of an action-research carried out in 2015. The results reported in this study refer only to the second semester. Due to the nature of this work and the investigated phenomenon, the present work was developed as a qualitative approach to answer the proposed questions.

4.1. Institution and participants

This research was developed in an educational organization that offers nursery, early childhood education and elementary education in Recife – Pernambuco, Brazil. Extracurricular activities are offered for students since four years old: martial arts, ballet, futsal, informatics, music, swimming and psychomotricity. One of these paper authors acts as a computer teacher in the selected institution. The learning space was a computer lab with ten desktop computers. There were weekly meetings, lasting approximately 40 minutes each one and participation was mandatory. Sixty-nine students aged between four and seven years participated in this study, comprising the following school grades: pre-k, kindergarten and first grade. Only pre-k students had not previously taken computer classes.

4.2. Data collection

Data were collected through participant observation and interviews (individual and collective). Individual semi-structured interviews were carried out with students who demonstrated significantly lower or higher performance than the average of their respective classes, being considered outliers. Collective semi-structured interviews were carried out after the experimentation of each game in order to analyze the general understanding of the class about the presented contents. The interview scripts were composed both of open questions and investigative questions. Audios of individual interviews were recorded and transcribed. No audios were recorded in collective interviews, the teacher-researcher transcribed the data. The participant observation method was chosen because it was considered the most adequate by means of the investigated phenomena. These observations occurred during game experimentation and focused on understanding the students' interaction with the games used.

4.3. Evaluated digital games

This study adopted a teaching approach based on digital games. Selection criteria were: whether they could be used on desktop computers, whether they were targeted to public served and whether they were free, since there was no financial income available to invest in paid options. Four digital games were selected in specialized sites and evaluated: *Code Studio – Course 01*⁵, *Run Marco*⁶, *The Foos*⁷ and *Tynker*. The students were invited to try out the games without any previous explanation from the teacher about the contents covered and/or about how to play. Students could ask for help to teacher and/or colleagues whenever they are in doubt.

5. Results and Discussions

The digital games were evaluated in two steps to answer two research questions: (a) Did digital games provide effective development of skills related to programming concepts? (b) How did the interaction elements of game interface contribute to the learning of these concepts? To answer question (a), the games were used in a programming teaching experience for children four to seven years old. In the subsequent step, results of the first step were analyzed in views of usability heuristics adapted by Falcão and Barbosa (2015) for educational software.

5.1. [QP1] Did digital games provide the effective development of skills related to programming concepts?

The set of skills to be developed were extracted from the CSTA K-12 Computer Science Standards⁸

⁵ This game will only be referred to as *Code Studio*.

⁶ <https://next.allcancode.com/home>

⁷ <http://thefoos.com/hour-of-code/>

⁸ This curriculum, until submission of this work, is only available in its draft version on the official website:

reference curriculum. Due to the discipline workload, only the skills of the category "Algorithms and Programming" were considered. The approaches/strategies used to develop the children's skills are described. Identifiers have been preserved as per the original document.

[1A-A-5-3] *Plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) way (e.g. story map, storyboard, sequential graphic organizer).*

[1A-A-3-7] *Construct and execute algorithms (sets of step-by-step instructions) that include sequencing and simple loops to accomplish a task, both independently and collaboratively, with or without a computing device.* These skills do not assume the use of devices and were worked through children's play. They were presented only to broaden the understanding about the study developed. Students have collaboratively created algorithms for everyday tasks. The concept of simple loops was not addressed.

[1A-A-6-8] *Analyze and debug (fix) an algorithm that includes sequencing and simple loops, with or without a computing device.* Non digital and digital games were used to work this activity at different moments. Only the sequencing concept was addressed.

[1A-A-5-2] *Construct programs to accomplish a task or as means of creative expression, which include sequencing, events, and simple loops, using a block-based visual programming language, both independently and collaboratively (e.g. pair programming).* Games use block-based visual programming notations. Students moved characters and performed tasks through a set of connected blocks. Students built solutions in individual and collaborative way. The following concepts were addressed: sequencing, events and simple loops. The use of repetition block was considered difficult (Figure 01). The repeating block was used in two situations: when suggested by the game help option or by the teacher. When the use of the repetition block was not mandatory for the correct response, children did not use it. When questioned about it, children claimed to be unaware of its operation or to find it difficult to use. This behavior was observed in the use of all games, suggesting that this is not only a problem in concept understanding, but it concerns the interaction elements and the concept approach in the game.

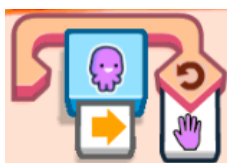


Figure 01. Screenshot of The Foos: repeating block with parameter five (hand with five raised fingers)

[1A-A-3-5] *Decompose (break down) a larger problem into smaller sub-problems with teach guidance or independently.* Students should solve more complex tasks that require decomposition into smaller problems. This ability was not developed satisfactorily in any class, even with

teacher's intervention. The results suggest that children used decomposition unintentionally, without understanding what they were doing. In more advanced stages, the chain of commands was longer, so the children inserted commands and, observing that expected result had not been reached, they keep inserting more commands until the final goal was reached.

5.1.1. Student performance evaluation

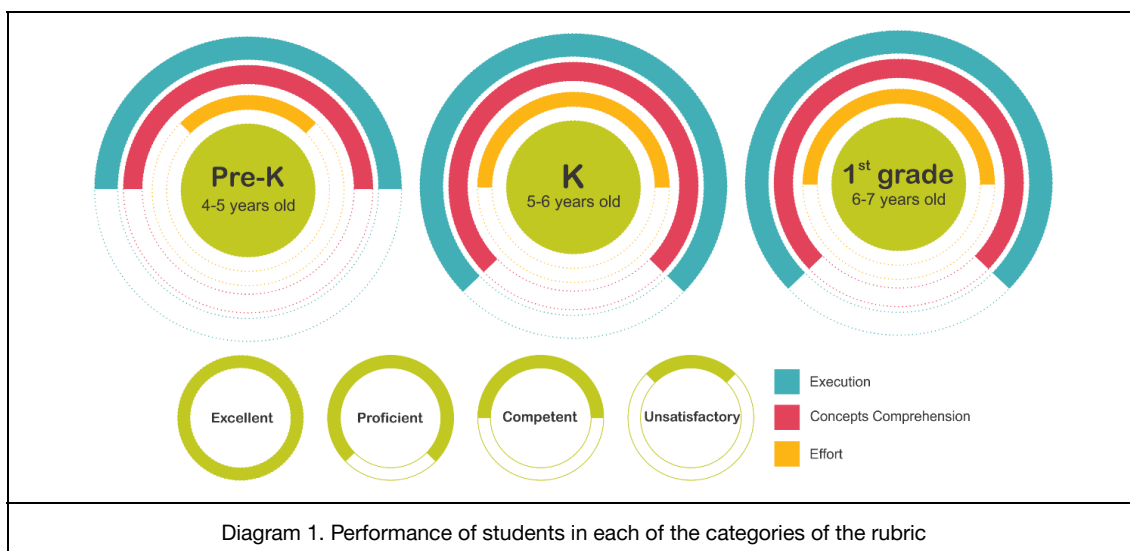
Student performance was evaluated in the development of these skills and competences. The evaluation model is described⁹ in Table 1 which has three axes: Execution (Ex), Concept Comprehension (C) and Effort (Ef). The Execution axis contemplates the algorithmic correctness of proposed solution. The Concept Comprehension axis evaluates if students understand the concepts involved in the proposed solution and, finally, the Effort axis evaluates the effort expended by students to carry out the activities.

Table 1. Student Performance Evaluation Rubric

	Unsatisfying	Competent	Proficient	Excellent
Ex	Code / algorithm does not work or has major faults.	Code / algorithm works mostly, or has minor faults.	Code / algorithm works as expected by the student.	Code / algorithm is functional, refined and executed as efficiently as possible.
C	Student is not able to understand how his code / algorithm should work and is unaware of the process.	Student can describe how a big part of his code / algorithm should work and demonstrate some understanding of the content.	Student can describe how his code / algorithm should work and solve problems that prevent desired results.	Student can describe how his code / algorithm works, how he wrote it and helps others with their codes.
Ef	Student demonstrates minimal effort does not make use of classroom time effectively and work is incomplete. Student refuses to explain more than one idea.	Student performs only the minimum necessary. Student has more than one idea, but does not persist.	Work is completed in an above-average way, although more could have been done. Student explores various solutions.	Work has been successfully completed and exceeds teacher expectations. Student demonstrates and explores multiple ideas and solutions, besides making inquiries.

The following diagram summarize student performances grouped by education level and categorized by the axes of the rubric and their respective attributes.

9



5.2. [QP2] How did the interaction elements of game interface contribute to the learning of programming concepts?

In this second stage, the present study evaluates the results obtained in the first stage from a set of heuristics adapted for educational software [Falcão and Barbosa 2015]. Falcão and Barbosa propose the adaptation of some heuristics to the context of educational software distributed in three categories: (1) *Exploratory interaction*, (2) *Visual metaphors of the interface*, (3) *Interaction design*.

Exploratory interaction

[H1] *Help feature and documentation: if necessary, help feature should be easily found and consulted, being objective and simple.*

Educational games must have suggestive interaction elements that allow children to play without the need to resort to manuals and/or tutorials. If the player needs it, he should have simple and quick ways to find help. None of the games evaluated has features that allow the player, for example, to observe the operation of a specific command at any time in the game if he has doubts about how it works. This is an especially important aspect since children are the target audience. Tips on how to solve the proposed challenge involving available commands are not offered. Instead, all evaluated games provide a level in which the player just follows a tutorial and then is presented to a new level in which he must use the newly presented commands/concepts by himself.

[H2] *Visibility of system status: The system must keep the user informed through appropriate feedback.*

Feedbacks are important components for building knowledge and reflection, as they allow the learner-player to recognize their progress throughout the learning process. Not understood

feedback stops the opportunity for reflection and interferes with learning, especially in debugging experiments. Evaluated games had feedback elements that did not help in the understanding of progress, they only indicated if the result obtained by the inserted commands was in agreement with the expected result or not. This is an important aspect because it conveys the impression that "It is all wrong" instead of "Something is wrong".

[H3] *Errors: System must prompt for user action to prevent unwanted actions. In case of errors, the messages must inform the cause, the consequences and the solution.*

In educational games, players use to make conceptual mistakes, which make up the journey of building knowledge. In this way, games need to provide means to point out these errors and suggest possible solutions rather than treating all the mistakes indiscriminately. None of the games used provide this kind of support. It is considered that heuristics [H3] and [H2] are some of the most important aspects that help students in learning concepts, making it more fluid and facilitating exploratory interaction.

[H4] *User control and freedom: system must provide means for the user to perform the actions he wants with the possibility of undoing and redoing them.*

Games evaluated had no functionality to last action undo or redo. This could be a very important feature, especially in code debugging experiences.

Visual metaphors of the interface

[H5] *Recognition instead of recall: Relevant information and objects must be visible to minimize the user's memory workload.*

We consider that [H5] is related to [H6]. Therefore, before recognizing the elements of interaction rather than remembering, the user must be able to understand the symbols and elements presented. However, although the interface had all relevant information and objects, players did not understand some interaction elements.

Interaction design

[H6] *Correspondence between the system and the real world: system must speak user's language and use familiar concepts and conventions of real world.*

None of the evaluated games addressed this heuristic in an adequately way. Notably, language was the most striking factor. In three of the evaluated games, the predominantly textual language limits comprehension, since children are still in the literacy phase. The symbols did not refer to familiar concepts: for example, the symbol of infinity as a parameter for the repetition block to represent the repetition of one or more commands many times (Figure 02, right). Or the terms North, South, East and West represented by their initial letters (Figure 02, left). These nomenclatures make no sense to children who are still developing perception and laterality, since they do not distinguish left or right. The use of the letter 'O' ("oeste", meaning west) on the directional arrow to the left resulted in questions such as: "But, teacher, why 'O' and not 'E'? It's wrong, right (sic)?".



Figure 02. Screenshots: Directional arrows of Code Studio 01 (left), infinite parameter in the replay block of the game The Foos (right)

Table 2 summarizes the results obtained in the heuristic evaluation. The results are classified in (M) meets, (PM) partially meets and (NM) does not meet.

Table 2. Summarization of the heuristic evaluation results

HEURISTIC/ GAME	Code Studio	Run Marco	The Foos	Tynker
[H1]	PM	PM	PM	PM
[H2]	NM	NM	NM	NM
[H3]	NM	NM	NM	NM
[H4]	NM	NM	NM	NM
[H5]	PM	PM	M	PM
[H6]	PM	PM	PM	PM

5.3. Collaboration, engagement and other aspects

Other aspects were observed, although it was not the object of this study: as collaboration, autonomy, performance and interest. Students worked in pairs because of the number of computers. Initially, the pairs were formed only considering the affinity between the children. We observed that pairs with very different performances and profiles tended to favor the more autonomous and better performing students. Students with greater learning difficulties and withdrawn profile began to participate less or simply not participate, demonstrating dissatisfaction and disinterest.

As a result, pairs began to be directed by the teacher, considering profiles and learning difficulties. With this rearrangement, students with more difficulties remained participative and did not demonstrate to be ashamed since their peers had similar learning difficulties. In the case of pairs with less proactive profile children, it was common for both to wait for teacher intervention when faced with the most challenging levels. After this, we started to consider the level of autonomy in pair's formation.

Another detail observed was the relationship between persistence and performance.

Children with a *more persevering* profile, and not necessarily with the best performance, faced the mistakes found as opportunities for learning and were happier when they *went wrong by little* and continued trying. During this research, we did not notice possible relations between obtained performance and student's gender. Notably, other studies are needed to deeply understand these issues.

6. Final Considerations

This exploratory study presented the evaluation of educational digital games effectiveness in the development of programming-related skills from a teaching experience with children aged four to seven years.

Obtained results show that games can favor the learning process of the concepts presented. In addition to that, results also show that the interaction elements of the interface play a direct role in the understanding of concepts, as well as the very approach of these concepts within the games: the environment, the narrative, the character and other aspects. Results from heuristic evaluation suggest that the elements of interaction, symbols and adopted language influenced negatively the learning of programming concepts. The feedback, the helping feature and the way games deal with students' mistakes need improvement in order to make the learning experience more effective.

We hope this paper can collaborate with the incorporation of computational thinking in primary school, especially with digital game-based experiences. Future work includes a longitudinal analysis of predicted skills development through qualitative and quantitative methods of data collection and analysis.

References

- Barros, L., Ribeiro, S. P. S. e Oeiras, J. (2009). Projeto de Extensão Universitária para apoio e realização da Olimpíada Brasileira de Informática em Escolas. In: XXIX Congresso da SBC - XVII Workshop de Ensino de Computação, Bento Gonçalves.
- Bezerra, F. e Dias, K. (2014) Programação de Computadores no Ensino Fundamental: Experiências com Logo e Scratch em Escola Pública. In: XXXIV Congresso da SBC - XXII Workshop de Ensino de Computação, Brasília.
- Falcão, T. P., e Barbosa, R. "Aperta o Play!" Análise da Interação Exploratória em um Jogo Baseado em Pensamento Computacional. In: Anais do Simpósio Brasileiro de Informática na Educação. 2015. p. 419.
- Gomes, T., Melo, J., e Tedesco, P. Jogos Digitais no Ensino de Conceitos de Programação para Crianças. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de

Informática na Educação-SBIE). 2016. p. 470.

James, M. (2013) "UK To Teach Programming Starting At Age 5", <http://www.i-programmer.info/news/150-training-a-education/6077-uk-to-teach-programming-starting-at-5.html>, Março.

Machado, E. Z. A., Vasconcelos, I. R., e Malta, K. (2010) Uma Experiência em Escolas de Ensino Médio e Fundamental para a Descoberta de Jovens Talentos em Computação. In: XXX Congresso da SBC - XVIII Workshop de Ensino de Computação, Bento Gonçalves.

McGonigal, J. (2012) A realidade em jogo. Rio de Janeiro: Editora Best Seller.

Mendonza, M. (2015) "Australia Replaces History and Geography With Coding in New Primary School Curriculum", <http://www.techtimes.com/articles/86669/20150921/australia-replaces-history-and-geography-with-coding-in-new-primary-school-curriculum.htm>, Março.

Oliveira, M. L. S., Souza, A. A., Ferreira, A. e Barbosa, E. F. S. B. (2014) Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In: XXXIV Congresso da SBC - XXII Workshop de Ensino de Computação, Brasília.

Papert, S. Final report of the Brookline LOGO Project. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1979.

Silva, S. F., Barbosa, A. F., Souza, A. A., Lauyse, E. G. D. S. M. e Rogério, S. D. O. S. (2015) Relato de Experiência de Ensino de Computação no Ensino Fundamental em Estágio Supervisionado da Universidade de Pernambuco no Campus Garanhuns. In: XXXV Congresso da SBC - XXIII Workshop de Ensino de Computação, Recife.

Torezani, C., Chagas, L., and Tavares, O. (2013). *Newprog* - um ambiente online para crianças aprenderem programação de computadores. In ~ XXV Workshop de Informática na Escola (WIE 13), Campinas, SP.

Souza, S. M., Rios, M. S., Rodrigues, C. A., Santos, D. M. B. e Bittencourt, R. A. (2015) Oficinas de Programação com Ambientes Lúdicos para Meninas do Ensino Fundamental. In: XXXV Congresso da SBC - XXIII Workshop de Ensino de Computação, Recife.

Wing, J. (2008) "A US View of a Grand Challenge: Establishing Computational Thinking for All". In: Grand Challenges in Computing - Research, Edited by: John Kavanagh and Wendy Hall. UK Computer Research Committee. United Kingdom.