An Analysis of Interaction Design in Children's Games Based on Computational Thinking

Original Title: Uma Análise do Design de Interação em Jogos Infantis Baseados em Pensamento Computacional

Taciana Pontual Falcão[1], Rafael Santos Barbosa[1], Tancicleide Simões Gomes[2]

[1] Departamento de Estatística e Informática (DEINFO), Universidade Federal Rural de Pernambuco (UFRPE). Rua Dom Manoel de Medeiros, s/n, Dois Irmãos, CEP 52171-900, Recife – PE – Brasil.
[2] Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE), Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária, CEP 50740-560 - Recife – PE – Brasil.

ARTICLE INFO

Article history:
Received 21 July 2016
Accepted 29 June 2017
Available online 1 August 2017

Keywords:
Programming games
Computational thinking
Interaction design
Children

ISSN: 2594-5602

DOI:
10.14210/ijcthink.v1.n1.p16

ABSTRACT

INTRODUCTION: *Several digital games that aim at developing computational thinking in children have emerged in the last few years, as the importance of such ability is increasingly being recognized. Nevertheless, there are no appropriate methods to determine the quality of these games, which hinders their adoption in formal education settings.* ***OBJECTIVES:*** *The general goal of this research is to contribute to the development of a specific evaluation method for children's games that involve computational thinking. In this sense, the specific objective of this paper is to identify which pedagogical and technical aspects of children's interaction with this type of game are proved relevant for developing computational thinking. Methods: An analysis of two games based on the logic of programming was performed through empirical observation of children's exploratory interaction, in the light of heuristics for interface usability, consisting thus of a combination of formative and objective evaluation.* ***RESULTS:*** *Analysis showed that the game's environment for experimenting with programming commands and visualizing the consequent effects was not sufficient for effective guidance, revealing the need for a human mediator. The main aspects identified as needing improvement were the forms of instruction provided to the child; the design of visual representations of commands, which embed key logical concepts; the correspondence between children's context and expectations and the games' simulated world.* ***CONCLUSION:*** *Results point to directions for the creation of specific parameters and adapted heuristics for evaluation of games based on programming logic for children. In particular the main open question is how to design a game so that computational thinking concepts are implicit conditions to solve challenges, making the process engaging but above all leading to the construction of complex concepts such as parameterized commands.*

1. Introduction

The importance of developing computational thinking in school children is increasing as its value in the contemporary society becomes more evident (Barr and Stephenson, 2011) (Wing, 2006). Computational thinking relates to strategies for problem solving through logical formal reasoning, at varied levels of abstraction (Wing 2006), which was pointed by Seymour Papert in the 80's when he created Logo, a programming language for children (Papert, 1980). Nevertheless, decades have gone by and challenges remain as to the best manner to develop children's computational thinking, within the classroom context, and as to which related abilities should be assessed (Barr and Stephenson, 2011). Precursory initiatives include educational robotics, unplugged computing, visual programming tools and digital games that involve programming logic (França et al., 2014; Gomes and Melo, 2013). This article is situated within the context of the latter, which will be called here digital programming games, for the sake of simplicity in terminology.

Generally speaking, research indicates that digital games are powerful tools in the teaching-learning process (Felicia, 2012). More specifically, games that involve computational thinking through problem solving with the support of step-by-step simulations are usually based on programming logic, and do not necessarily use a formal language (Medeiros et al., 2013). The player/learner becomes the protagonist of knowledge construction through investigation and exploration (Felicia, 2012). Besides, games may become motivational elements by offering playful and interactive learning opportunities (Medeiros et al., 2013). However, the use of digital games in educational contexts is still timid, mainly due to two factors: teachers' difficulties in finding games that fit the syllabus; and lack of self-attributed competence to evaluate the pedagogical quality of the games (Medeiros and Schimiguel, 2012). As a matter of fact, despite the good amount of proposals for educational software evaluation, no method for this aim has been consensual and consolidated so far in the literature, and in the case of educational games, methods are scarcer (Brito Junior and Aguiar, 2014).

The main goal of this article is to identify relevant aspects to be evaluated in digital games that focus on developing computational thinking, by promoting discovery learning through exploratory interaction. In this sense, a theoretical analysis based on interface usability heuristics was performed, combined with empirical user research with two games of the kind.

2. Exploratory Interaction and Computational Thinking

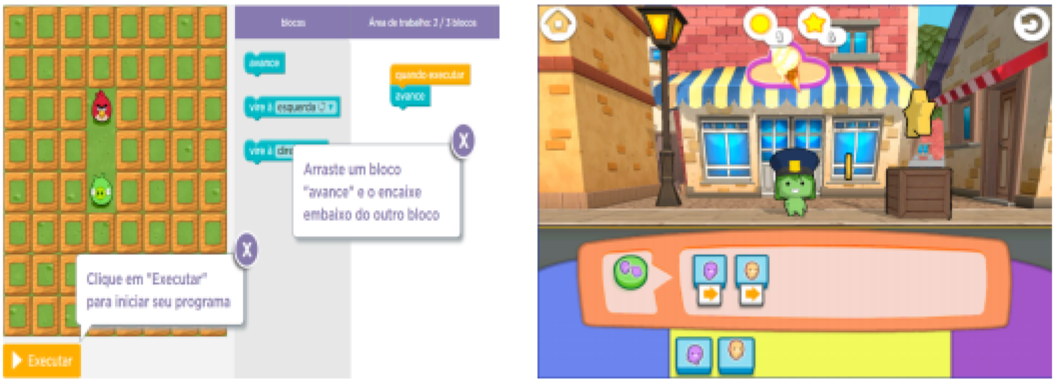
2.1. Discovery Learning

In discovery learning approaches, learners are expected to build knowledge through exploration of external representations (Bruner, 1961). Concepts are not transmitted to them, but constructed autonomously, from interaction with adequate artefacts. This approach is aligned with the constructivist learning theory, according to which the learner must be an active participant of the learning process in order to build meaningful knowledge (Piaget and Inhelder, 1969). Nevertheless, benefits of discovery learning as opposed to direct instruction have not been clearly demonstrated (Mayer, 2004). The approach is mainly criticised for its lack of assistance to the learners, risking leading them to wrong conclusions or knowledge gaps (Alfieri et al., 2011). Guided discovery approaches emerged as potential solutions to this problem, by introducing feedback and scaffolding to structure learners' activity and balance the freedom for the learner to be cognitively active, with guidance to direct them towards meaningful knowledge (Mayer, 2004).

Feedback and scaffolding, as means of guiding learners, can be provided both through educator and artefacts mediation. Digital technologies are powerful mediational artefact, given their potential for providing dynamic interactive feedback and abstract-concrete representational links (Borba and Villareal, 2005). On the other hand, digital technologies introduce in the learning process new aspects to be taken into account, particularly regarding the design of appropriate pedagogical feedback and the pedagogical value of metaphors conveyed through interfaces and simulations. Such aspects are analysed in this article, based on interface heuristics and children's interaction with programming games that aim to develop computational thinking through exploratory interaction.

2.2. Computational Thinking and Digital Games

Besides the motivation associated with playfulness (Medeiros et al., 2013), games challenge players to make decisions based on logical reasoning, strategies, systematic processes and critical evaluation of information, developing cognitive competences for problem solving (Felicia, 2012). This process is mediated by games' interactivity and quick feedback, promoting cycles of reflection and experimentation, which constitutes a form of stimulating computational thinking based on exploratory interaction. Games like Hour of Code[1] (Figure 1, left), The Foos[2] (Figure 1, right) and Lightbot[3] (Figure 2) use block-based visual programming languages, where instructions are composed by dragging and connecting command blocks, focusing more on problem solving and algorithmic logic and less on language syntax (Medeiros et al., 2013), and thus facilitating the development of computational thinking based on computer programming. They provide an environment for exploratory interaction where learners can verify visually and dynamically the effects of sequences of commands of their choice, and establish links between abstract programming logic and concrete actions performed by the games' character.



Arraste um bloco "avance" e o encaixe embaixo do outro bloco

Clique em "Executar" para iniciar seu programa

Figure 1. Hour of Code (left) and The Foos (right).

Lightbot and The Foos are popular games in this field, freely available, and thus were chosen for the evaluation here presented. They both follow the same paradigm of guiding a character through specific tasks of increasing level of difficulty by assembling a sequence of commands (like walk forward, turn left and right, jump, repeat, call a procedure and others). In Lightbot, a robot must walk through a path to reach the blue square and light it up. As the paths become more complex, the task can only be accomplished with the use of logical structures like flow control, loops and procedures.

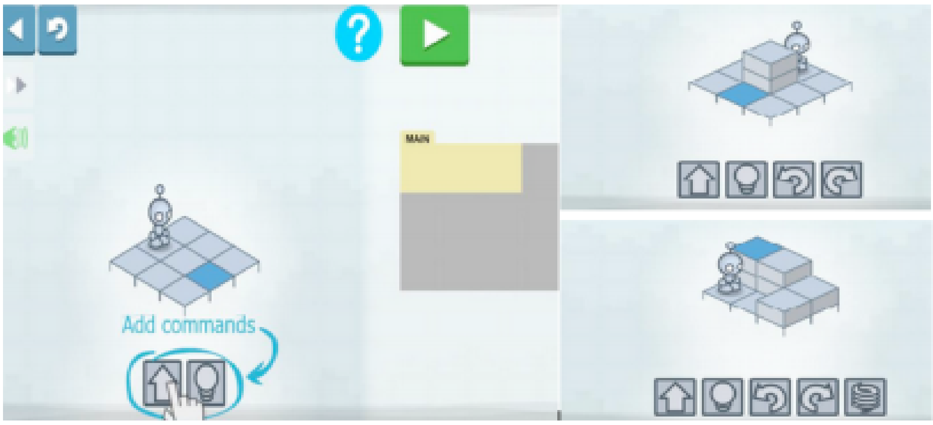


Figure 2. Lightbot's stages 1, 2 and 3.

The Foos offers a larger variety of tasks and characters, who move forward or backwards in a 2D interface, having to escape enemies and obstacles and may also acquire objects of interest along the way.

[1] hourofcode.com
[2] www.thefoos.com
[3] lightbot.com

3. Evaluation of Lightbot and The Foos

Several methods have been proposed in the literature for evaluating educational software, but none became consensual so far. Mostly, methods suggest considering technical (software quality, ergonomics and usability) and pedagogical aspects (Brito Junior and Aguiar, 2014). Oliveira (2001) distinguishes between (complementary) objective and formative evaluations. In the objective approach, a specialist verifies a set of pre-defined criteria, while the formative evaluation analyses user interaction. We adopt Oliveira's classification to present the evaluation of Lightbot and The Foos performed with children and complemented by usability heuristic analysis, with the goal of investigating discovery learning opportunities through exploratory interaction, with a special attention to the role of interface components.

Formative evaluation took place in distinct moments. Lightbot was evaluated with 11 children (5 boys and 6 girls), aged 6 - 9 years (except one 5-year-old and one 13 year-old). They were recruited randomly at a public square near the university, after obtaining authorization from their guardians. The activity was performed at the square, using the researchers' smartphones, playing levels 1, 2 and 3 (Figure 2). Instructions were given on demand exclusively. Sessions lasted for 15 - 30 minutes, depending on the child's interest. The Foos was evaluated with 39 children, aged 5 - 7 years, from two private schools, using desktop computers at the schools' lab. Children's parents gave formal authorization for their participation in the research activity. At both schools, the Informatics teacher, author of this article, presented The Foos as the day's activity. According to the number of available computers, children played individually or in pairs, for 20 - 30 minutes. As in usual classes, children were allowed to ask for the teacher's help as they wished, and instructions were given accordingly. Data was analysed qualitatively, through field notes and photographs.

Objective evaluation was performed according to Nielsen's software usability heuristics (Nielsen and Mohlic, 1990), the most consolidated set of usability aspects for interface evaluation (Cybis et al., 2010). It is important to acknowledge that the context of educational software has its own specific characteristics, representing a category to which the heuristics may not necessarily apply. However, despite a good amount of proposals of specific evaluation methods for educational software (Brito Junior and Aguiar, 2014), none of them has proved ideal, and there are even less options to evaluate educational digital games. So, assuming that Nielsen's heuristic evaluation is formally considered an effective method of interface evaluation, we transposed the heuristics set to an educational context in order to ground the qualitative analysis of the formative evaluation, in a complementary manner.

4. Results

4.1. Formative Evaluation

Children's motivation was perceived when playing both games. Surprisingly enough, the outdoors environment with moderate noise and many distractions (public square) where Lightbot was tested did not interfere in children's concentration. Most children said they would like to play the game again and would try harder to finish the phases. In most cases, the researchers had to interrupt the interaction so that other children could play. Similarly, no requests to stop playing were made by the children who interacted with The Foos. They demonstrated that they liked the interface and bonded emotionally with the characters. This was perceived through children's verbal interaction with the game's characters, including requests, complaints and praising them for good behaviour. For example, a boy asked, in a supplicating tone, when noticing the character behaving unexpectedly: "ooh, little toy man, don't do that!!!"; and a girl exclaimed, about the celebration dance performed by the character at the end of each phase: "nice! We got it, she's gonna do the butt dance!!".

Due to the recruitment circumstances, Lightbot's evaluation was followed by parents or other guardians, who spontaneously interfered, putting pressure and placing their personal expectations on children. Interventions included: explanations about the game's goal; instructions ("read the things, V., read what's in the squares!"); and even orders ("the last one, G., it's here, oh for God's sake, press play!!"). As a result, some children, clearly intimidated, simply declared they liked the game, even though many difficulties in their interaction were perceived.

The facility children presented to pass the first levels of The Foos worked well to get them involved in the game. When reaching harder levels, the interest built for the game made the will to solve the challenge stronger than the frustration of not getting it done easily, preventing giving up. During the weeks that followed the activity, children often asked the teacher to play The Foos again.

A frustrating aspect perceived was the low speed of the game processing, causing delayed feedback for the children, in some situations: "it's like he's sleeping, what's taking him so long to get walking?!".

Boys who played Lightbot demonstrated higher motivation, interest and performance than girls. The game's interface can be considered slightly more appealing for boys. With The Foos, no gender difference was noticed.

These qualitative evidences demonstrate that the games tested overcame the stigma of educational games, traditionally considered boring and dull (Kirriemuir and McFarlane, 2004). Especially in the case of The Foos, the graphical interface, script and characters make the game appealing and involving for the target audience. Beyond that, The Foos manages to develop computational thinking through programming logic in an almost implicit manner, i.e. children do not notice they are learning some formal content, and keep motivated and focused to overcome the challenges presented. Lightbot's interface and scenario are less ludic, and more explicit on programming. Even so, several of the participant children's guardians were interested in downloading the game for their children to play again in the future, considering it educational and non-violent.

Besides these general observations, content analysis performed with field notes generated more specific findings in two categories: exploratory interaction; and interaction and interface design. These categories are discussed next combined with an analysis based on Nielsen's usability heuristics (Nielsen and Mohlic, 1990) (indicated by 'H'). Three of Nielsen's heuristics were not used in the analysis as they did not relate to the aspects discussed: consistency, flexibility and efficiency of use, and minimalist design).

4.2. Objective Evaluation

4.2.1. Exploratory Interaction: Free versus Guided

The six first phases of the tested version of The Foos mainly involved using the commands of 'move' (forward and backward) and 'jump', and were considered easy by the children, who played autonomously. The very first phase (Figure 3), which simply consisted of moving the character forward to catch a star, was completed in seconds by all children, without help, with spontaneous comments like: "this is sooo easy!!". Most children started asking for constant help from phase 9, as the more advanced commands were introduced, and caused a certain level of frustration.

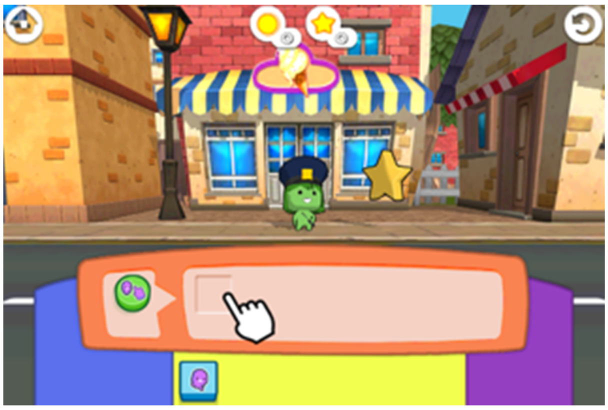


Figure 3. The Foos - Phase 1.

The Foos adopts a particular way of instructing the player, through the image of a hand (pointing with one finger, as shown in Figure 3), which simulates the click and drag actions with every newly introduced command, inducing the player to do the same. Such way of instruction / help was efficient in capturing children's attention and make them imitate the hand's action, thus learning how to interact with The Foos. However, with the introduction of advanced commands, the 'hand instruction' did not work so well. For example, despite the hand animation, children did not understand the choice of the walking direction by clicking on the arrow of the 'walk' command (which opened an option menu with two arrows - left and right - corresponding to backward and forward respectively) (Figure 4). We noticed children's lack of comprehension because they repeated the learned interaction but did not choose the appropriate direction, revealing that they were performing the action automatically, but did not know why.

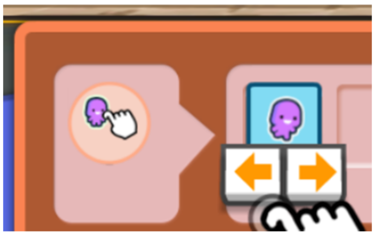


Figure 4. Choice of the direction for walking.

A similar situation occurred with the command of repetition, introduced after a phase where three boxes must be piled up for the character to reach all stars. In the following phase, using the repetition command (which is introduced parameterized to 4 loops, as indicated by a hand showing 4 fingers), the child is expected to pile up four boxes without having to place four 'box' commands in the execution area. The pedagogical sequence is clear: the repetition command solves an issue presented previously. Nevertheless, this was not perceived by the children, who presented two types of behaviour: keep using a certain amount of the same command as a way of repeating the action like before, ignoring the introduction of the repetition command; or imitate the hand simulation and dragging the repetition command to the execution area, however showing no understanding of its effect. For example, a boy who used the repetition command to pile boxes up changed inadvertently the parameter to infinite. As a result, boxes kept being piled up non-stop (Figure 5), pushing the character up, to which the boy reacted saying with surprise: "damn! I put too many!!".

The use of parameters with the repetition command was not immediately noticed by the children, because when it is first introduced, it comes with the needed number of repetitions. The graphical design of the command does not facilitate the perception of the parameter, and when it needed to be changed to complete the phase, children asked for help.

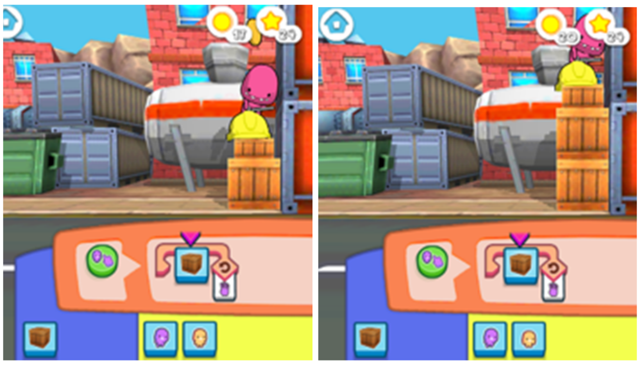


Figure 5. Repetition command used to pile boxes.

Analysing exploratory interaction with Lightbot, we perceived that, when attempting to discover how to play, children were confused and asked for help. Children were shown the game's tutorial, but did not understand the written instructions. In most cases, the researchers had to help them with hints during the interaction, leading to quite a high level of intervention.

Also regarding the comprehension of what is happening, The Foos indicates the execution of commands with a pink, broad arrow (Figure 6). Several children did not understand the function of this mark, and in cases of slow execution, thought that there was something wrong with the game. In Figure 6, for example, the policewoman remains facing the blue monster, performing very subtle movements and making no sounds, which conveyed the idea for the children that nothing was happening.

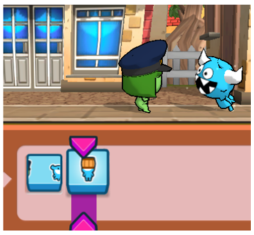


Figure 6. Status of the system in execution.

Complementing the findings above with a heuristic analysis, we discuss next aspects that must be taken into account to promote children's autonomy in the process of exploratory interaction.

[H] Help and documentation: if need be, help must be easy to find and browse, being objective and simple. Help in educational games is a challenge for designers, given that children have little patience to study tutorials and read instructions, or do not understand tutorials, as happened with Lightbot. Ideally, educational software should be intuitive enough so that users interacted without the need of documentation. The alternative implemented by The Foos, based on imitation, proved interesting for teaching children to interact with the game without a tutorial. Pedagogically speaking, however, doing by imitation risks leading children to take actions and not knowing why. Imitation, in this context, is the opposite of the learner's autonomous discovery: if on one hand it facilitates interaction, on the other hand it restricts the learner's exploration space, and can reduce potential reasoning which would lead to a better comprehension of actions chosen.

[H] Visibility of system status: the system must keep the user informed through appropriate feedback. In the context of this work, this heuristic complements the previous one. In educational contexts, the concept of "appropriate feedback" contemplates processes of scaffolding, through which technologies become mediator tools in the learning process (Borba and Villareal, 2005). Effective scaffolding helps children to explore the game autonomously, with less external guidance, supporting discovery learning. Furthermore, the importance of feedback was demonstrated in this research also regarding details like slow interactions, where some children thought nothing was happening.

[H] Errors: the system must ask for users' confirmation of relevant actions, preventing undesired effects. When errors do happen, messages must inform the cause, consequences and solution. Guidelines regarding the management of errors apply to educational software just as to any other software. However, it is fundamental to expand the reflection on errors in the educational context to include conceptual error, as opposed to interaction error. The type of feedback conveyed by educational software to a conceptual error is determined by the pedagogical orientation of the software (essentially, constructivist or behaviourist). But, generally speaking, the system could give hints after a sequence of incorrect attempts of the learner. In the sessions with Lightbot, children several times made the robot do a complete turn when they discovered they were rotating to the opposite side, instead of selecting the correct rotation command. When identifying such an action, the game could suggest using the other command, which is a form of scaffolding. With The Foos, the repetition command was used in mistaken ways, e.g. when a child piled up an "infinite" number of boxes and the game did not give conceptual feedback on that action.

[H] User control and freedom: the system must provide ways for the user to perform desired actions, with the possibility of undoing and redoing. This is crucial for discovery learning, for it allows the learner to explore the interface safely, not fearing the error. In pedagogical terms, undo-redo is one of the great gains brought by digital interactive technologies, allowing the learner to experiment, try various actions and interpret the consequences. With both games here analysed, children felt mostly confident to explore.

4.2.2. Interface and Interaction Design

The basic navigational elements of The Foos, i.e. start the game, move to the next phase and go back to previous phase, did not cause any difficulty in interaction. Composing the sequence of commands by dragging blocks was also easily understood. The exception was the action of executing the sequence of commands by clicking on the character itself, which proved problematic. This specific design choice was not intuitive for the children, who even after doing it a couple of times kept forgetting how to execute, repeatedly hesitating. This indicates that children did not learn this interaction because it was not meaningful for their mental models. Trying to find the way to do it, some of them clicked on the arrow of the 'walk' command, opening inadvertently the choice of backward / forward (Figure 4). The child then clicked on one of the options, which sometimes inverted the direction in which the character was moving. Apparently, children clicked the arrow because they associated it with the common 'play' icon of general use interfaces (which resembles an arrow).

This idea is reinforced by the fact that Lightbot uses the representation of 'play' to start the execution, which was very well understood by the children in this research.

On the other hand, children who interacted with Lightbot needed to be explained about the commands icons (Figure 2), as the tutorial was not effective in teaching their functions.

In particular, the rotating (left / right) commands, represented by rounded arrows, were of very hard use for the children. They could not establish mental mappings between the robot's movement and researcher's instructions like "you should turn to the **opposite** side", or even the notions of **left** and **right**. Children did better when the researcher used their finger to make turns in the air, showing how the robot should turn, simply saying "turn this way", "turn that way". It must be considered that the robot's path in Lightbot is in three dimensions, demanding quite sophisticated spatial reasoning for mentally visualising the robot's moves, and the design of the rotating commands does not facilitate this reasoning.

The Foos, being a bi-dimensional interface, does not have the concept of turning, only moving forward and backward. Still, as discussed before, the choice between the arrows that indicated the direction of the character's walking (Figure 4) was also a source of misunderstanding. Possibly, children got used to the forward command during the first few phases, and the introduction of a choice within the same command block was not easily assimilated. This difficulty with parameterised commands was also identified with the repeat command, as discussed in section 4.2.1.

Another very important aspect related to the characters' movement was the re-execution of a sequence of commands, within trial-and-error exploratory processes in which children engaged. In Lightbot, every time the button 'play' is pressed, the robot goes back to the starting position and then executes the sequence of commands that are found in the specific box. We found that children expected that the robot would continue moving from its last position, and were confused by the actual behaviour. They accumulated commands in the control box, thinking that they were adding new moves from the robot's current position, and did not tend to remove commands that were found inadequate. In The Foos, the character actually continues from the last position (i.e. where it stopped after executing the commands). However, as in Lightbot, all commands in the box are re-executed. Again, children expected that only the lastly added commands would be performed. In advanced phases, which involved several actions to be performed, children tended to decompose the challenge in small tasks to reach the final goal, and place and execute the corresponding commands as they thought of the solution for each part. However, they did not remove the commands that solved previous tasks, making the character repeat all commands from the first one, at every execution. To sum up, none of the games delivered the interaction expected by children, that the character would continue from where it stopped and only execute the lastly added commands. That is not to say, however, that this should be the most adequate design, if we think about programming logic, where the whole program is run at every execution.

It was also noted that children often forgot the second part of Lightbot's goal, i.e. lighting up a lamp on the blue square. They focused on reaching the blue square and considered they had accomplished the task by getting there. Finally, the design of the 'jump' command in Lightbot was confusing for the children, who thought that they needed to associate a 'move forward' command to every jump (as in most video games). Instead, the jump command already made the robot move to the next square.

As done in the previous section, the analysis above is complemented with a heuristic analysis, now discussing aspects more specifically related to the design of child-game interaction.

[H] Recognition rather than recall: relevant information should be visible to minimise the user's memory workload. In the context targeted by this research, there is one important aspect to be added to this heuristic: besides being visible, key interface elements must be easily comprehensible. Lightbot's rotating commands were not understood and thus their effect was not memorised, making children experiment at each level to discover their functioning again, instead of making use of them to discover more advanced concepts. An example from The Foos is the difficulty to remember how to start the execution.

[H] Match between system and the real world: the system should speak the user's language and use familiar concepts and real-world standards. In simulations like the games analysed, we propose thinking in a broader perspective, considering virtual world, real world and learner's context and profile. In the virtual world where Lightbot's robot lives, for example, a jump takes the robot up and forward, which was not intuitive probably due to the functioning of most video games (where the specific jump command only takes the character up), even though we could assume that a person who jumps in the real world most often jumps forward. Another unexpected behaviour was the robot restarting from the beginning of the path at every execution (as in the real world people continue from where they stopped), or the Foos' character continuing from the last position but repeating the commands that made it get there. Finally, children's mental model considered that the goal was to reach a final place, and did not internalise the need for an action with no significant meaning in the context, as turn a lamp on (in Lightbot).

5. Conclusions and Future Directions

This paper presented the evaluation of two programming games for children that support the development of computational thinking. We performed an analysis which combined empirical observation and theoretical framework based on usability heuristics. We identified interactional aspects to be taken into account in the design and evaluation of digital educational exploratory games, particularly involving computational thinking. Games like Lightbot and The Foos potentially promote discovery learning, which in turn is grounded on constructivist and constructionist theories (Piaget and Inhelder, 1969; Papert, 1980). However, analysis revealed obstacles in the discovery learning process that were not overcome despite the capabilities provided by the mediation of digital technologies. Some of these obstacles are known difficulties of discovery learning, mainly related to the quantity and quality of feedback needed to guide learners. The present research showed that the environment for experimenting with programming commands and visualising the effects (movement of the game character), although very appropriate for exploration, was not sufficient for effective guidance, revealing the need for a human mediator. Specific reasons for this need to be better investigated, but analysis identified some of them, in the light of usability heuristics. The main obstacles identified were:

- A tutorial as a help resource was not productive, as children did not benefit from it. Instruction through imitation was effective to teach interaction, but failed in the context of some pedagogical aspects. Constructive hints during interaction, based on user interaction history, could provide more appropriate scaffolding;
- The meaning of visual representations of some key graphical elements (particularly rotational arrows) was not internalised by children, hindering interaction and conceptual comprehension. Visualising rotation implies spatial reasoning and the design of the icons did not facilitate this process.

- In some cases, a mismatch between children's context / expectations and the games' simulated world was noticed: the goal of turning a lamp on; the effect of jumping; the character resuming execution by going back to the initial position or repeating commands previously executed.

This piece of research contributes with input for the construction of a specific method of evaluation for exploratory games involving computational thinking. The analysis based on usability heuristics performed in this paper, although useful to point out key aspects of interaction, revealed, on the other hand, several limitations when it comes to their application to the educational context, and directions for the creation of a specific set of parameters and / or adapted heuristics.

Evaluation of other similar games is needed to consolidate the indications of this research. We intend to adopt a bottom-up process, i.e. generate evaluation criteria from the use and analysis of several games of the kind, instead of trying to use methods for educational software in general.

Finally, there is one particular open question we would like to answer for this kind of game: how to design the game so that computational thinking concepts are implicit conditions to solve the challenges, making the process fun, but also leading to the construction of complex concepts such as parameterised commands? In order to answer this, longitudinal studies are needed that evaluate the development of children's computational thinking in the long term, stimulated through interaction with this kind of game.

Acknowledgements

This work was performed as part of the course Technologies for Learning, classes from the second semester of 2013 and first semester of 2015, with Computer Science students from the Federal Rural University of Pernambuco (UFRPE). We thank student-researchers Alcantara Rodrigues, Eugênio Maia, Fábio Menezes, Gustavo Barbosa, Igor Lucena, Isabella Rocha, Pedro Silva and Verônica Santos, and of course all children who took part in the studies.

References

ALFIERI, L.; BROOKS, P. J.; ALDRICH, N. J. and TENEMBAUM, H. R. Does discovery based instruction enhance learning? Journal of Educational Psychology, vol. 103, n.1, pp. 1-18, 2011.

BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? ACM Inroads, vol. 2, n.1, pp. 48-54, 2011.

BORBA, M. C.; VILLAREAL, M. E. Humans-with-Media and the Reorganization of Mathematical Thinking, Springer, USA, 2005.

BRITO JUNIOR, O. O.; AGUIAR, Y. P. C. Análise de abordagens objetivas para avaliação de softwares educativos. Proceedings of: 13º Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais - IHC'14, Foz do Iguaçu, Brasil, 2014.

CYBIS, W.; BETIOL, A. H.; FAUST, R. Ergonomia e Usabilidade - Conhecimentos, métodos e aplicações, Editora Novatec, São Paulo, 2010.

FELICIA, P. Digital Games in schools: a handbook for teachers. European Schoolnet, 2012.

FRANÇA, R. S.; FERREIRA, V. A. S.; ALMEIDA, L. C. F. de; AMARAL, H. J. C. do. A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. Proceedings of: XXII Workshop sobre Educação em Computação - WEI, Brasília, Brasil, 2014.

GOMES, T. S. G.; MELO, J. C. B. de. App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação. Proceedings of: Congresso Brasileiro de Informática na Educação - CBIE, Campinas, Brasil, 2013.

KIRRIEMUIR, J.; McFARLANE, A. Report 8: Literature Review in Games and Learning. Futurelab Series. Acessed on June 30th, 2015. Available at: http://www2.futurelab.org.uk/resources/documents/lit_reviews/Games_Review.pdf, 2004.

MAYER, R. E. Should There Be a Three-Strikes Rule Against Pure Discovery Learning? The Case for Guided Methods of Instruction. American Psychologist, vol. 59, pp. 14-19, 2004.

MEDEIROS, M. O.; SCHIMIGUEL, J. Uma Abordagem Para Avaliação De Jogos Educativos: Ênfase No Ensino Fundamental. Proceedings of 23º Simpósio Brasileiro de Informática na Educação - SBIE, RJ, Brasil, 2012.

MEDEIROS, T. J.; DA SILVA, T. R.; ARANHA, E. H. S. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. Revista Novas Tecnologias na Educação - RENOTE, vol. 11, n.3, pp. 2-10, 2013.

NIELSEN, J.; MOHLIC, R. Heuristic evaluation of user interfaces. Proceedings of ACM CHI'90 Conference, Seattle, USA, 1990.

OLIVEIRA, C. C. de; COSTA, J. W. da; MOREIRA, M. Ambientes informatizados de aprendizagem: produção e avaliação de software educativo. Papirus, Campinas, 2001.

PAPERT, S. Mindstorms - Children, computers and powerful ideas. Basic Books, NY, 1980.

PIAGET, J.; INHELDER, B. The psychology of the child. Basic Books, NY, 1969.

WING, J. M. Computational thinking. Communications of the ACM, vol. 49, n.3, pp. 33-35, 2006.