What makes a maze-based programming challenge difficult?

Ioanna Kanellopoulou[1[0000-0002-0618-9970]] , Pablo Garaizar[1[0000-0001-8160-9130]] and Mariluz Guenaga[1[0000-0002-0311-2150]]

[1] Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain
ioanna.kanellop@opendeusto.es, [garaizar, mguenaga]@deusto.es

Abstract. Computational Thinking (CT) is one of the key skills within adequate digital literacy for the 21st century. Over the last few decades, several initiatives fostered the development of CT among primary and secondary school students (e.g., The Hour of Code, CodeWeek EU, Scratch Day) using online learning platforms (e.g., Code.org). Many of the activities on these platforms are based on 2D maze-based challenges that students solve using visual code blocks. Our findings are based on the analysis of the platform log data gathered from 326 learners during three studies in which students were asked to solve maze-based programming challenges in our online platform, Kodetu. According to our results, a Kodetu challenge is difficult when: a) the maze has turns and the total number of steps needed to go from the initial position to the endpoint is high, and b) when not only movement blocks, but also loop and conditionals blocks are needed to solve the challenge. The results of this research should be considered when designing learning activities to develop and enhance CT skills through maze-based programming challenges.

Keywords: Computational Thinking, Difficulty, Educational Games, Block-Based Maze Game.

1 Introduction

The increasing digitization of all aspects of our daily lives demands the development of adequate digital literacy [1-4]. Schools and other learning initiatives have put a lot of effort into developing STEAM (Science, Technology, Engineering, Arts and Mathematics) skills in recent years [5-15]. Among them, CT [16] encompasses the skills needed to be able to solve problems with the help of computers: abstraction, pattern recognition, generalization, error correction, algorithmics, and many others.

The most specific of all the skills that make up CT is probably the algorithmic skill, i.e. the ability to define a set of steps in the form of sequences, conditionals, and loops to solve a problem. For this reason, many of the online platforms, mobile applications, and even "unplugged" learning materials (e.g., board games, activity books, toys) focus on the design of algorithms by learners. A common approach in these

This work was supported by the European Union's Horizon 2020 Research and Innovation Program through the Marie Sklodowska-Curie Grant under Agreement 665959.

cases is to use games where players have to guide a character through a 2D grid that may contain obstacles, hazards, or even moving enemies [14, 15, 16]. These games propose Computational Thinking challenges leveraging very few actions (moving forward, backward, turning). Following this approach, we developed the online platform Kodetu at the Deusto Learning Lab of the University of Deusto.

Our goal with this study is to know which features of the Kodetu programming challenges influence learners' performance. Therefore, we designed a wide set of maze-based programming challenges and tested them with a large sample of primary and secondary school students. Based on the performance of these students, we have estimated which factors have more or less influence on their performance and thus on the difficulty associated with each challenge.

Considering this, our research questions are focused on the two aspects that define a challenge in Kodetu:

1. The features of the maze: How do maze characteristics (width, height, total number of steps in the maze, optimal path, maze loops, turns, and numbers of x-crosses and t-crosses) affect the performance in a maze-based programming challenge?
2. Programming constraints: How do coding limitations (blocks provided and block limit) affect the performance in a maze-based programming challenge?

Knowing the answers to these research questions will allow us to design challenge sequences adapted to the proficiency level of each learner and maximize their learning.

2 Difficulty In Maze-Based Programming Challenges

Difficulty is generally defined as the commitment taken to effectively perform an operation [8]. The difficulty of a challenge can also be defined as the probability that a player will fail to solve it [17]. From an educational perspective, difficulty is the ability and the effort necessary to complete an educational task [18].

Difficulty is considered a key factor in promoting the motivation of learners in educational games and resulting in better learning outcomes [17]. For that reason, challenges of increasing complexity and adaptive challenges are considered more effective for learning than non-adaptive ones, since they assess the learner's performance and adjust the difficulty of the next challenge accordingly [19].

Some authors have studied the difficulty associated with educational games that employ mazes [7, 21]. For example, Gallego-Durán et al. [7] created a game in which students had to solve some Pac-Man-like mazes using the Prolog computer language. These authors measured difficulty by defining an easiness function based on the progress/score of an activity, without taking into account the characteristics of the maze. Pelánek and Effenberger [14] analyzed the difficulty and complexity of puzzles and microworld elements. They used factors such as the failure rate and the median time to solve the puzzle to evaluate performance, as well as complexity measures based on the solution of the puzzle and the microworld features.

However, existing research does not investigate the specific characteristics of block-based maze games that affect the performance of the learners. Given the wide usage of this type of games to promote CT, it is important to define how these characteristics affect the performance and use them to design effective learning paths of adaptive difficulty. To achieve this, it is necessary to record learners' interactions in as much detail as possible. Numerous authors have followed this approach [23,24,25]. With the aim of conducting data-driven research, we have developed an automatic interaction logging system for Kodetu, as we explain in the next section.

3 Kodetu

Kodetu is an online platform where participants must solve challenges using a block-based programming interface. It is an educational game that allows researchers and teachers to easily create new challenges and challenge sequences.

The interface of Kodetu is divided into three parts (see Fig. 1). The left panel shows the maze-like spaceship, the initial position of the astronaut, and the endpoint marker. Players must lead the astronaut from the beginning of the maze to the endpoint using the programming blocks provided in the center panel of the interface. There are movement blocks (go forward, turn left, and turn right), loop blocks, and one-branch or two-branch conditional blocks. Players drag and drop these blocks to the right panel, i.e. the workspace, where they define the visual program that leads the astronaut to the endpoint. When players click the "play" button, the program defined in the workspace is executed, and the astronaut moves according to the programmed instructions. A Kodetu challenge is solved successfully when the player's program leads successfully the astronaut to the exit of the maze. We use the term sequence to define a group of consecutive Kodetu challenges.

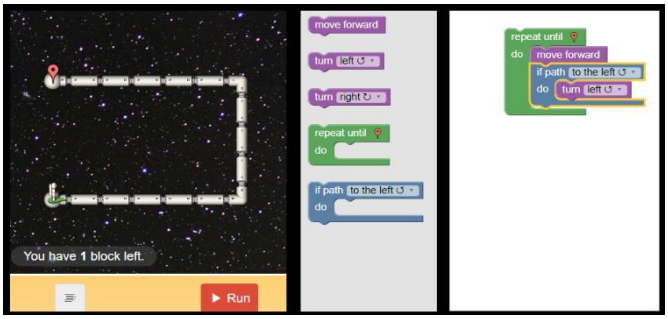

Fig. 1. The Kodetu interface. Mazed-based challenge (left), available blocks to create the solution (center), and the workspace where the user builds the program (right).

Throughout the game, Kodetu logs all player interactions with the interface. All players' data are gathered anonymously and stored in the database under a unique identifier that is automatically generated when the player accesses the platform. By analyzing this usage data, we can estimate the performance of each player.

4 Maze Features And Programming Constraints Analysis

The aim of this study is to determine which maze features and which programming constraints affect the difficulty of each challenge in Kodetu. Therefore, we conducted 3 studies with a very similar design. After a preliminary analysis of data from over 19.000 participants in Kodetu sessions gathered during the last 5 years, we concluded that these factors could affect the difficulty of each challenge: the width and height of the maze, the total number of steps in the maze, the length of the optimal path (from the starting point to the exit), no turns on the optimal path/one-direction turns on the optimal path (that is, only right- or only left-direction turns)/two-direction turns on the optimal path, X-crosses (the possibility to move north, south, west and east from a certain point of the maze), T-crosses (the possibility to move south, west and east from a certain point of the maze), maze loops (a path that allows users to go from one position in the maze to the same position used in the maze without passing through any previous position), blocks available (movement only blocks, loops + movement blocks, and conditionals + loops + movement blocks), and block limits (the number of blocks allowed to be used in a maze challenge). To assess the influence of each factor, we designed sequences of Kodetu challenges and conducted several workshops with primary and secondary school students.

In Study 1, we wanted to know how much maze loops affect the performance in a maze-based programming challenge. We designed 34 Kodetu challenges, separated into pairs that differed only in the number of maze loops(e.g., one challenge was defined as {width: 7, height: 7, optimal path: 24, total steps: 24, maze loops: 0, x-crosses: 0, t-crosses: 0, turns: 2, no block limit, blocks: all available} and another challenge was the same but instead of 0 maze loops, it has 2 maze loops). With these 34 challenges, we prepared 7 sequences of 5 challenges each. A total of 70 participants aged between 11 and 15 years old (44% female, 56% male) had 30 min to solve ten challenges in Kodetu. The first 5 challenges were training challenges and were not considered in the analysis. The last 5 challenges corresponded to one of the 7 challenge sequences mentioned before, randomly assigned to each participant.

The design of the sequences (Fig. 2) aimed to achieve increasing difficulty and a smooth transition from one challenge to the next based on an initial estimation of the difficulty of the challenges.

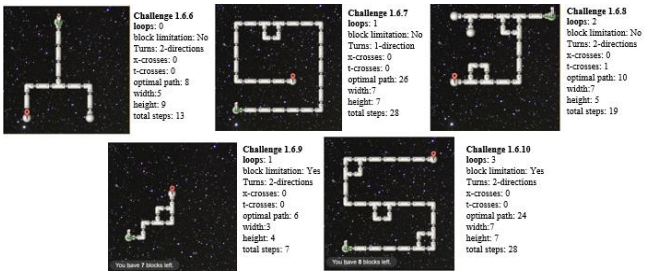

Fig. 2. Example of a sequence-Study 1

To determine whether the number of maze loops in a challenge affects success, we performed one-way ANOVA in which the dependent variable was the percentage of success in a challenge and the independent variable was the number of maze loops (four groups: 0, 1, 2, or 3 maze loops in each challenge). The results showed that no statistically significant difference existed between groups [$F(3,31) = 0.705$, $p = 0.556$]. We ran another one-way ANOVA in which the groups of the independent variables were challenges with no maze loops and challenges with maze loops. The results showed that no statistically significant difference existed between the groups [$F(1,33) = 1.604$, $p = 0.214$]. Therefore, these results suggest that maze loops in Kodetu challenges do not cause additional difficulty for the players.

In Study 2, we wanted to know how maze characteristics (turns, and numbers of x-crosses and t-crosses) and coding limitations (blocks provided and block limit) affect the performance of Kodetu players. We designed forty challenges following the same principles of Study 1 (i.e., 20 pairs of challenges where all the variables were the same except for one), and we created 6 sequences of 7 challenges each. A total of 197 participants aged 9 to 11 years old (49% female, 49% male, and 2% other) had 60 min to solve 12 challenges in Kodetu. The first 5 challenges were training challenges with no block limit, and they were not considered in the analysis. The next 7 challenges corresponded to one of the six challenge sequences randomly assigned to each participant.

We conducted a one-way ANOVA test to compare the effect of the maze turns on the percentage of success on challenges with no turns, one-direction turns, and two-direction turns. There was a significant effect of the type of turn on the success rate at the $p<0.05$ level for the three conditions [$F(2,39) = 3.722$, $p = 0.033$]. However, we did not find a significant effect using the Tukey HSD and Duncan post hoc tests in terms of pairwise comparisons.

Regarding programming constraints, an analysis of variance showed that the effect of the blocks available was significant [$F(2,39) = 20.032$, $p = 0.000$]. Post hoc analyses using the Tukey HSD post hoc criterion for significance indicated that the success percentage was significantly higher in the conditions in which movement only blocks (go forward-turn left-turn right) ($M = 0.934$, $SD = 0.0755$) and loops + movement blocks ($M = 0.945$, $SD = 0.0544$) were provided than in the other condition (conditionals + loops + movement blocks) in which $M = 0.55$ and $SD = 0.2899$. Similarly, statistically significant differences in the means of the challenges with and without a block limit were observed with $F(1,40) = 17.902$ with $p = 0.000$.

One-way ANOVA showed that the analysis was not significant for the effect of the numbers of x-crosses [$F(3,38) = 0.978$, $p = 0.413$] and t-crosses [$F(3,38) = 2.034$, $p = 0.125$] on success.

Study 3 is a replica of Study 2 with older participants. A total of 59 participants aged 15-16 (37% female, 59% male, and 3% other) had 60 min to solve 12 challenges in Kodetu. The first 5 challenges were training challenges with no block limit and were not considered for analysis. The last 7 challenges corresponded to one of the six challenge sequences prepared before and randomly assigned to each participant.

With the data of Study 3, we wanted to compare the effects of the maze characteristics and the coding constraints on the success percentage of the participants. A one-

way ANOVA test was conducted to compare the effect of the maze turns on the percentage of success on challenges with no turns, one-direction turns, and two-direction turns. There was a significant effect of the type of turn on the success rate at the $p<0.05$ level for the three conditions [$F(2,39) = 3.997$, $p = 0.026$]. An analysis of variance showed that the effect of the blocks available was significant [$F(2,39) = 7.768$, $p < 0.001$] for the three conditions (movement only blocks, loops + movement blocks and conditionals + loops + movement blocks). However, we did not find a significant effect using the Tukey HSD and Duncan post hoc tests in terms of pairwise comparisons. Furthermore, one-way ANOVA showed that the analysis was not significant for the effect of the numbers of x-crosses [$F(3,38) = 0.113$, $p = 0.952$] and t-crosses [$F(3,38) = 2.054$, $p = 0.123$] on success.

In order to analyze the differences in the results between Studies 2 and 3, we conducted a one-way ANOVA to compare the effect of age on success in the group of 9- to 11-year-olds and in the group of 15- to 16-year-olds. We found that age had a significant effect on success at the $p<0.05$ level for the two groups [$F(1,82) = 8.437$, $p = 0.005$].

5 Discussion

The present research provides a quantitative analysis of data obtained from the Kodetu platform, which advances our understanding of the maze characteristics and coding limitations that affect participants' performance in maze-based programming challenges. Our findings are based on the analysis of the platform log data gathered from 326 learners during three studies.

After analyzing the data obtained from Study 1, we found that the existence of maze loops in the challenges did not affect learners' success rate. Thus, the high failure rate, especially in the last challenges of the sequences, cannot be explained by the maze loops. One of the reasons for this finding may be that as long as learners can cognitively solve the challenge, the maze loops do not affect their performance (i.e., they are not challenged by the maze, but by the code needed to make the astronaut get to the endpoint).

The results from Study 2 indicated that challenges that provide conditionals and loop blocks (in addition to movement blocks), as well as challenges with block limits, need more time/interactions/attempts to be solved. This is in line with the results from previous research [14] as the use of blocks of conditionals and loops to solve a challenge requires advanced CT skills [26]. Furthermore, the data analysis shows that turns in the optimal path affect learners' performance in a challenge; however, it is not significant if there are one- or two-direction turns. This suggests that as long as the optimal path is not a straight line, the challenge is complex despite the direction of the turns.

The results from Study 3 were consistent with the results from Study 2. The statistical analysis showed that the age of the participants turned out to be an important factor, which indicates that age plays a key role in the success rate of this kind of

coding challenges. However the factors that affected the success rates were the same in both studies.

Consequently, a Kodetu challenge is difficult when: a) the maze has turns and the total number of steps needed to go from the initial position to the endpoint is high, and b) when not only movement blocks, but also loop and conditionals blocks are needed to solve the challenge. The results of this research should be considered when designing learning activities to develop and enhance CT skills through maze-based programming challenges.

References

1. "Computer and Information Technology Occupations : Occupational Outlook Handbook: U.S. Bureau of Labor Statistics." https://www.bls.gov/ooh/computer-and-information-technology/home.htm (accessed Sep. 30, 2020).
2. "ICT specialists in employment - Statistics Explained." https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_in_employment (accessed Sep. 30, 2020).
3. "2015 Joint Report of the Council and the Commission on the implementation of the strategic framework for European cooperation in education and training (ET 2020) - New priorities for European cooperation in education and training," p. 11.
4. OECD, "Better Skills, Better Jobs, Better Lives." 2012, [Online]. Available: https://www.oecd-ilibrary.org/content/publication/9789264177338-en.
5. D. Koupritzioti and S. Xinogalos, "PyDiophantus maze game: Play it to learn mathematics or implement it to learn game programming in Python," Educ Inf Technol, vol. 25, no. 4, pp. 2747–2764, Jul. 2020, doi: 10.1007/s10639-019-10087-1.
6. Ž. Ternik, A. Koron, T. Koron, and I. N. Šerbec, "Learning Programming Concepts Through Maze Game in Scratch," in European Conference on Games Based Learning; Reading, Reading, United Kingdom, Reading, Oct. 2017, pp. 661–670, Accessed: Feb. 02, 2018. [Online]. Available: https://search.proquest.com/docview/1967728949/abstract/573F8E28B0344D95PQ/1.
7. F. J. Gallego-Durán, R. Molina-Carmona, and F. Llorens-Largo, "Measuring the difficulty of activities for adaptive learning," Univ Access Inf Soc, vol. 17, no. 2, pp. 335–348, Jun. 2018, doi: 10.1007/s10209-017-0552-x.
8. J. Guggemos, "On the predictors of computational thinking and its growth at the high-school level," Computers & Education, p. 104060, Oct. 2020, doi: 10.1016/j.compedu.2020.104060.
9. L. Zhang, J. Nouri, and L. Rolandsson, "Progression Of Computational Thinking Skills In Swedish Compulsory Schools With Block-based Programming," in Proceedings of the Twenty-Second Australasian Computing Education Conference, New York, NY, USA, Feb. 2020, pp. 66–75, doi: 10.1145/3373165.3373173.
10. S. Trilles and C. Granell, "Advancing preuniversity students' computational thinking skills through an educational project based on tangible elements and virtual block-based programming," Computer Applications in Engineering Education, vol. n/a, no. n/a, doi: 10.1002/cae.22319.
11. I. Fronza, L. Corral, and C. Pahl, "Combining Block-Based Programming and Hardware Prototyping to Foster Computational Thinking," in Proceedings of the 20th Annual SIG

Conference on Information Technology Education, New York, NY, USA, Sep. 2019, pp. 55–60, doi: 10.1145/3349266.3351410.
12. A. Eguíluz, P. Garaizar, and M. Guenaga, "An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills," in Simulation and Gaming, D. Cvetković, Ed. InTech, 2018.
13. D. Weintrop, "Block-based programming in computer science education," Commun. ACM, vol. 62, no. 8, pp. 22–25, Jul. 2019, doi: 10.1145/3341221.
14. R. Pelánek and T. Effenberger, "Design and analysis of microworlds and puzzles for block-based programming," Computer Science Education, vol. 0, no. 0, pp. 1–39, Oct. 2020, doi: 10.1080/08993408.2020.1832813.
15. W. Min et al., "Promoting Computer Science Learning with Block-Based Programming and Narrative-Centered Gameplay," in 2020 IEEE Conference on Games (CoG), Aug. 2020, pp. 654–657, doi: 10.1109/CoG47356.2020.9231881.
16. J. M. Wing, "Computational Thinking," Commun. ACM, vol. 49, no. 3, pp. 33–35, Mar. 2006, doi: 10.1145/1118178.1118215.
17. M. Csikszentmihalyi and I. S. Csikszentmihalyi, Optimal Experience: Psychological Studies of Flow in Consciousness. Cambridge University Press, 1992.
18. F. J. Gallego-Durán, R. Molina-Carmona, and F. Llorens-Largo, "An Approach to Measuring the Difficulty of Learning Activities," in Learning and Collaboration Technologies, Cham, 2016, pp. 417–428, doi: 10.1007/978-3-319-39483-1_38.
19. S. Vanbecelaere, K. V. den Berghe, F. Cornillie, D. Sasanguie, B. Reynvoet, and F. Depaepe, "The effectiveness of adaptive versus non-adaptive learning with digital educational games," Journal of Computer Assisted Learning, vol. 36, no. 4, pp. 502–513, 2020, doi: 10.1111/jcal.12416.
20. K. Kiili, S. de Freitas, S. Arnab, and T. Lainema, "The Design Principles for Flow Experience in Educational Games," Procedia Computer Science, vol. 15, pp. 78–91, 2012, doi: 10.1016/j.procs.2012.10.060.
21. M. Szabó, K. D. Pomázi, B. Radostyán, L. Szegletes, and B. Forstner, "Estimating task difficulty in educational games," in 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Oct. 2016, pp. 000397–000402, doi: 10.1109/CogInfoCom.2016.7804582.
22. M. S. McClendon, "The Complexity and Difficulty of a Maze," presented at the Bridges: Mathematical Connections in Art, Music, and Science, 2001, Accessed: May 30, 2018. [Online]. Available: http://at.yorku.ca/c/a/h/d/25.htm.
23. Eguiluz A, Guenaga M, Garaizar P, Olivares-Rodriguez C (2017) Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis. IEEE Transactions on Emerging Topics in Computing PP(99):1–1. https://doi.org/10.1109/TETC.2017.2768550
24. Grover S, Basu S (2017) Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM, New York, NY, USA, pp 267–272
25. Piech C, Huang J, Nguyen A, Phulsuksombati M, Sahami M, Guibas L (2015) Learning Program Embeddings to Propagate Feedback on Student Code. In: International Conference on Machine Learning. PMLR, pp 1093–1102
26. J. Moreno-León and G. Robles, "Dr. Scratch: a Web Tool to Automatically Evaluate Scratch Projects," in Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE '15, London, United Kingdom, 2015, pp. 132–133, doi: 10.1145/2818314.2818338.