IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2021, accepted April 18, 2021, date of publication April 22, 2021, date of current version May 4, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3075027*

# First Steps Towards Automatically Defining the Difficulty of Maze-Based Programming Challenges

**IOANNA KANELLOPOULOU, PABLO GARAIZAR, AND MARILUZ GUENAGA**
Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain

Corresponding author: Ioanna Kanellopoulou (ioanna.kanellop@opendeusto.es)

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

**ABSTRACT** In a world where algorithms are ubiquitous, the development of computational thinking competencies is becoming progressively important among students, technology professionals, and 21st-century citizens in general. Educational games as a means of promoting computational thinking skills have gained popularity in recent years. Offering efficient educational games that promote computational thinking competencies requires personalized learning paths through adaptive difficulty. The research presented herein is a first attempt to define a difficulty function for maze-based programming challenges using log data obtained from Kodetu, which is a block-based maze game. Specifically, we conducted three studies with 9- to 16-year-old students who were asked to solve sequences of maze-based programming challenges. Using log data from these studies, we investigated the maze characteristics and the coding limitations that affect performance in the challenges and calculated the performance obtained by the participants using a fuzzy rule-based system. The results showed that the turns in a maze, the number of total steps of a maze, and the blocks provided affect student performance. Using regression analysis, we defined a difficulty function for maze-based programming challenges that considers the weights of these factors and provides a first step towards the design of adaptive learning paths for computational thinking-related educational games.

**INDEX TERMS** Computational thinking, difficulty, educational games, block-based maze game.

## I. INTRODUCTION

The great increase in the use of technology in everyday life during recent years could not leave the field of education unaffected. The COVID-19 pandemic has highlighted the relevance of technological literacy for stakeholders in education more than ever. Homeschooling, online tutoring, and the use of digital tools are some of the challenges that students, teachers, and families have had to overcome. In many sectors, the growing need for technology has increased the demand for computer science professionals [1]–[4]. In particular, computational thinking (CT) is one of the key skills of computer scientists; it is also valuable for professionals in other fields. When she first introduced the concept to the scientific community, Wing referred to CT as a fundamental skill that every person should develop in order to perform in modern society [5]. Therefore, it is necessary to promote and support CT through appropriate educational tools, methodologies, and strategies [5].

Many initiatives (Scratch Day, Hour of Code, and AI Leagues) and tools (Scratch, Alice, Blockly, and AppInventor) have been created to develop CT. The vast majority of these tools use block-based programming features that make programming more approachable for novice programmers [6]. Regarding the type of activities, many of them present maze-based programming challenges [7]–[9] with fixed challenges that are independent of the user's performance during the activity. This means that all participants face the same challenges and in the same order, regardless of their performance during the learning process. However, it would be desirable to have the ability to adapt the learning process to personalize the learner's needs and adapt to their progress so that they accomplish the most effective learning outcomes [10]. To this end, we must be able to estimate

The associate editor coordinating the review of this manuscript and approving it for publication was Saqib Saeed.

the difficulty of these activities so that teachers—or, ideally, a tool—can generate increasingly challenging learning activities with adaptive difficulty based on the learner's performance. According to flow theory, adequate scaffolding through challenges is key to offering challenges that are neither too difficult nor too easy, consequently causing anxiety or boredom [11].

With this aim in mind, we tried to design a maze-based online system to develop CT skills using block-based programming that provides learners with adaptive learning paths. However, the lack of a clear definition of the difficulty in this type of educational maze-based game led us to first define how we can measure the difficulty of such activities. This study presents the first steps towards automatically defining the difficulty of maze-based programming challenges and was validated with 326 K-12 students.

The first goal of this study is to determine the variables of a block-based programming maze game that affect performance. Then, we calculate the performance; based on that performance assessment, our final aim is to define a difficulty function for every maze-based programming challenge in our online platform. To achieve this, we set the following research questions:

1) 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) How do coding limitations (blocks provided and block limit) affect the performance in a maze-based programming challenge?

To answer the research questions, we conducted three studies using Kodetu, a block-based maze game developed by the group LearningLab of the Engineering Faculty of the University of Deusto.

The remainder of this paper is structured as follows. Section 2 presents a brief literature review of the main concepts introduced in the current research. Section 3 describes Kodetu, the maze-based game used in the research. In Section 4, we explain the methodology followed to estimate the difficulty of a programming challenge in Kodetu and the results obtained in the three experiments conducted in this study, which are used to define the difficulty function in Section 5. Section 6 discusses these results, the conclusions, the limitations, and future research directions.

## II. LITERATURE REVIEW

### A. DESIGNING EDUCATIONAL GAMES

Game-based learning (GBL) is a significant area of learning that has become more relevant in recent decades. Wu *et al.* [46] conducted a meta-analysis in which they observed that GBL is related to numerous learning foundations. A significant variant of GBL is learning based on digital educational games [47]. Papastergiou [48] showed in her research that learning through educational games is more effective and motivational than using a nongaming approach.

To develop effective educational games, there is a need to set guidelines and specific design frameworks [54]. Ibrahim and Jaafar [55] proposed a model for designing educational games based on three main factors, namely, game design, where the focus is usability, multimodal and fun; pedagogy, which focuses on learning outcomes, motivation theory, self-learning and problem solving; and finally, learning content modeling.

Aleven *et al.* [49] introduced another general framework for the effective design and analysis of educational games. The framework consists of three components: the learning objectives; the mechanics, dynamics and aesthetics of the game; and the instructional principles. Furthermore, the authors presented how the framework can be applied to an educational game called Zombie Division, created to enhance basic mathematical skills. Regarding CT, Malliarakis *et al.* [50] developed CMX, a massive multiplayer online role playing game with the purpose of teaching and enhancing the learning of computer programming. The design framework of CMX includes important concepts in the development of educational games such as the distinct characteristics of the users, the educational material organization and presentation and the scenarios and activities supported by the game. The results of their research conducted with first-year undergraduate students showed that the game increased students' motivation and enhanced their knowledge of computer programming.

### B. BLOCK-BASED PROGRAMMING ENVIRONMENTS

A variety of tools are available for early programmers to develop CT competencies. Scratch, Code.org, Blocky, Alice and MIT AppInventor are some of the most popular tools. They focus on introducing primary and secondary school students to the basic concepts of CT [12]–[15] and improving their skills through the playful characteristics these tools possess [16]. Additionally, research has shown that these block-based programming environments play a significant role in introducing learners to programming and the computer science world [17]–[19], [51] and to general science concepts [20], [21]. These educational environments overcome the problem of the complex syntax of text-based programming languages based on their interaction on drag-and-drop and the natural language descriptions of the blocks [22].

Significant steps have been taken towards making block-based programming environments adaptive [23], [24]; however, it is still difficult to find block-based programming environments that offer adaptive gameplay to fit individual learners' needs [25].

### C. DIFFICULTY IN EDUCATIONAL GAMES

Difficulty is generally defined as the commitment taken to effectively perform an operation [11]. Aponte *et al.* [26] claim that the difficulty of a challenge is the probability that the player will fail at it. As described above, difficulty is considered a key factor in promoting the motivation of learners in educational games and resulting in better learning

outcomes [27]. However, current definitions remain mainly intuitive; that is, difficulty is defined as the ability and the effort necessary to complete an educational task [28]. Therefore, there is a need to define a clear and accurate measurement of difficulty to create efficient adaptive systems.

Adaptive games are considered to be more effective than nonadaptive games as they continuously evaluate the success of the learner and adapt the difficulty of the activities to the individual level [29]. Flow theory, introduced by Csikszentmihalyi and Csikszentmihalyi [27], has been the basis of contemporary game design principles for flow experience. It explains that the goal of a game is to provide learners with challenges that balance the difficulty with their skills so as to prevent boredom (easy challenge-high skills) and anxiety (difficult challenge-low skills) [30].

Samprayo-Vargas *et al.* assessed the effectiveness of adaptive difficulty adjustment with 234 secondary school students separated into three groups [31]. Each group was given a different activity/game. Two of the groups were identical except for the difficulty adjustment mechanism. The results showed that significantly higher learning outcomes were achieved by the group that played the game with difficulty adjustment. Similarly, Lomas *et al.* sought to clarify whether difficulty indeed affects learners' motivation [32]. They found that difficulty decreased motivation when it was not balanced with the learner's skills.

Several approaches to defining difficulty in games already exist, ranging from measuring the difficulty of video games [33], [34] to assessing the difficulty of educational games [11], [35].

Gallego-Durán *et al.* [11] measured difficulty by first defining an ''easiness function'' of an activity and then defining the difficulty as 1 minus the easiness function. The easiness function depended on the progress/score that the player obtained in a specific timeframe. The effects of specific maze characteristics, such as the maze length and the maze loops in the graph of the maze, on the participants' progress were not measured. They implemented this function in a maze game in which students had to solve some Pac-Man-like mazes by programming in the Prolog language.

Pelánek and Effenberger [18] analyzed the difficulty and complexity of puzzles and microworld elements. They set basic difficulty measures, such as the failure rate and the median time to solve the puzzle, and complexity measures based on the solution of the puzzle and the microworld features. Then, they analyzed the correlation between them.

Other approaches, such as the one developed by McClendon [36], focus on the mathematical measurement of the difficulty of a maze. They used various complexity measures of the hallways in a maze and calculated the overall complexity and difficulty of the graph of the maze. However, maze-based educational games do not use complex mazes; thus, this function is not applicable to the mazes of this type of game. Consequently, using only the hallway measures of a maze is insufficient to define the difficulty of an educational maze-based game. (For example, maze loops, type of blocks used, and other similar aspects should also be considered.)

However, as far as block-based maze games are concerned, there is still no specific approach to measuring their difficulty. In our research, we aim to overcome these limitations and provide a measurement of the difficulty of block-based maze games considering not only the learner's performance in the game but also the characteristics of the activity. To achieve this, we performed several experiments and analyzed participants' interaction logs recorded automatically by the tool using learning analytics techniques.

## III. KODETU

Kodetu (http://kodetu.org/) is an online platform based on the Blockly game ''Maze'' where participants must solve challenges using a block-based programming interface. We used Kodetu to analyze learners' performance and measure the difficulty of the challenges they faced on the platform. The aim of Kodetu is to develop basic programming skills by creating visual programs for solving mazes. It is an educational game that allows one to easily create new individual and sequential challenges.

A Kodetu challenge is a maze level where an astronaut is located at an initial position and the exit of the maze is marked at a different point of the same maze. The challenge is solved successfully when the participant leads the astronaut to the exit of the maze using the visual blocks provided. We use the term sequence to define a group of consecutive Kodetu challenges.

The interface of Kodetu consists of three parts: the maze, the blocks provided to solve the challenge, and the workspace (Fig. 1). The first panel displays the maze that the participants must solve. The participants must lead the astronaut from the beginning of the maze to the endpoint. To achieve this, they use the blocks (programming instructions) provided in the middle part of the interface. There are movement blocks (go forward, turn left, and turn right), loop blocks, and blocks to define one- or two-branch conditionals that check for whether there is a path on the left, on the right, or forward. The user drags and drops these blocks to the third part of the interface, the workspace, where they build the visual program that leads

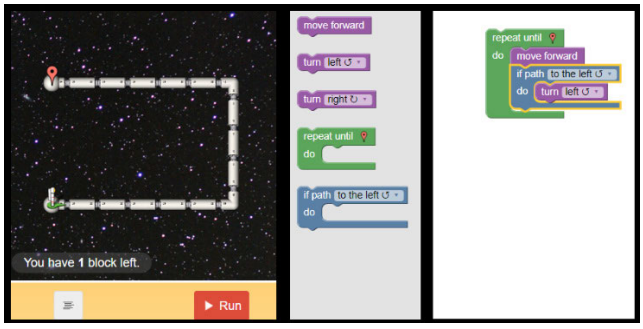


**FIGURE 1.** The Kodetu interface. Mazed-based challenge (left), available blocks to create the solution (middle) and the workspace where the user builds the program (right).

the astronaut to the endpoint. When the user clicks the ''play'' button, the program defined in the workspace is executed, and the astronaut moves according to the programmed instructions. When a new challenge is created, additional features can be added. (For example, we can limit the number of blocks used to build the solution.)

Kodetu is able to log every action of the user in the platform and save it in a database. All participants' data are gathered anonymously and stored in the database under a unique identifier that is automatically generated when the user accesses the platform.

Thanks to its interaction logging recording and the challenge creation features, Kodetu is a valuable tool to research in terms of the development of CT in learners. However, we aim to extend its features by adding the ability to adapt to the capabilities and performance as learners progress in the learning path and, ultimately, automatically generate programming challenges personalized for every learner. This computerized adaptive testing (CAT) will allow the complexity of each challenge to be dynamically adapted to learners. However, to achieve this, we must estimate the difficulty of a programming challenge in Kodetu.

## IV. METHODOLOGY

To estimate the difficulty of a programming challenge in Kodetu, we defined a 4-step procedure. First, we identified the characteristics that may affect the difficulty of programming challenges based on our experience with more than 19,000 participants throughout five years of using Kodetu [37]–[39]: 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).

Second, we designed a set of challenge pairs that differ in only one of the aforementioned variables (e.g., a challenge pair with an identical maze size and the same number of available blocks but one of the challenges has one more maze loop) and conducted several workshops to make participants solve the challenges.

Third, we studied how the learners' performance varied according to the characteristics of each challenge (e.g., percentage of successes in the challenge, time required to solve the challenge, and number of attempts).

Fourth, we estimated the difficulty of the challenges by analyzing the relationships between the characteristics of each challenge and the performance obtained by the participants.

The following sections describe a set of three studies that we conducted following this procedure. The data obtained from the three studies allowed us to measure the performance in the challenges, answer the research questions and obtain the difficulty function.

### A. STUDY 1

This study is our first approach to analyzing the difficulty of Kodetu's challenges. We investigated whether and how much maze loops affect the performance in a maze-based programming challenge.

We designed a total of 34 challenges, separated into pairs that differed only in the number of maze loops, in Kodetu. (For example, one challenge is 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 is exactly 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. (Because $7 \times 5 = 35$, one of the challenges was part of two sequences.)

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. Consequently, the first challenges of each sequence do not have a block limitation, and the values of the variables related to the maze increase over the progression of the sequences. Special emphasis was placed on achieving homogeneity between the seven sequences developed in order to avoid having some sequences that are more complicated than others.

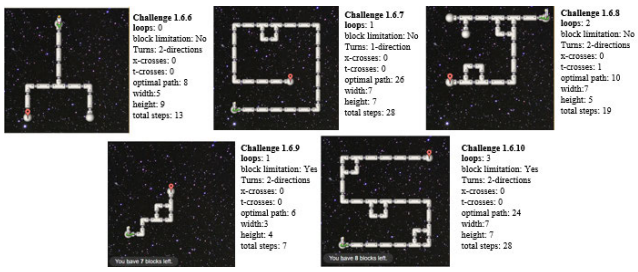


**FIGURE 2.** Example of a sequence-Study 1.

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 five challenges were training challenges and were not considered in the analysis. (The first challenge was explained step by step so that the use of Kodetu was understood properly.) The last five challenges corresponded to one of the seven challenge sequences mentioned before, randomly assigned to each participant. Table 1 presents the percentage of loss in each challenge (the percentage of users who failed to pass a challenge, considering the participants who entered the challenge).

**TABLE 1. Percentage of participants who failed to solve a challenge, Study 1.**

| Sequence | Challenge 6 | Challenge 7 | Challenge 8 | Challenge 9 | Challenge 10 |
|---|---|---|---|---|---|
| 1.1 | 11% | 0% | 13% | 71% | 50% |
| 1.2 | 0% | 27% | 13% | 71% | 100% |
| 1.3 | 0% | 0% | 33% | 0% | 33% |
| 1.4 | 0% | 0% | 0% | 0% | 38% |
| 1.5 | 0% | 0% | 45% | 0% | 67% |
| 1.6 | 14% | 33% | 0% | 50% | 100% |
| 1.7 | 11% | 13% | 0% | 29% | 40% |

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 [$F(1,33) = 1.604$, $p = 0.214$]. Despite the noticeable decline in the success rate of learners in the last challenges, these results suggest that the presence of maze loops in Kodetu challenges does not result in an added difficulty for participants.

There are several limitations to this study. First, the lack of sufficient time to perform the ten challenges (five training and five to test) could explain the low success rates in the last challenges of each sequence. Second, the order of the challenges in each sequence may have prevented us from reaching conclusions regarding the difficulty of each challenge. The challenges were ordered based on our initial assumptions regarding their difficulty, but we found sequences in which 67% of the participants succeeded in one challenge while the next was passed by 100% of participants who reached it. Therefore, this 100% success rate informs us only that the second challenge is not more difficult than the previous one; however, we cannot conclude whether it is perceived as being much easier, slightly easier, or of equivalent difficulty because those who passed it were those who also passed the previous challenge.

Considering the above, in the following study, we increased the time available to solve the challenges to 60 min, increased the number of test challenges from five to seven (duplicating the session duration is adequate for adding two more challenges), and made an effort to better define sequences of increasingly difficult challenges.

### B. STUDY 2

In this study, we sought to answer the following research questions: 1) How do maze characteristics (width, height, total number of steps in the maze, optimal path, turns, and numbers of x-crosses and t-crosses) affect the performance of learners in a maze-based programming challenge? 2) How do coding limitations (blocks provided and block limit) affect the performance in a maze-based programming challenge?

Therefore, we designed forty challenges following the same principles of Study 1 (for example, 20 pairs of challenges where all the variables were the same except for one), and we created 6 sequences of 7 challenges each. (Because $6 \times 7 = 42$, two of the challenges were part of two sequences).

A total of 197 participants aged 9 to 11 years old (49% female, 49% male, and 2% other) had 60 min to solve twelve challenges in Kodetu. The first five challenges were training challenges with no block limit, and they were not considered in the analysis. The first challenge was explained step-by-step so that the functioning of Kodetu was understood. The next seven challenges corresponded to one of the six challenge sequences randomly assigned to each participant.

To investigate the effect of the variables, we performed several statistical tests. We conducted a one-way ANOVA test to compare the effect of the 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.

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 $SD = 0.2899$.

Regarding the block limit, 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.

Regarding the remaining variables, considering that they were not divided into groups, we performed correlation tests to evaluate the association between these variables and the success rate. The success rate and maze width were weakly negatively correlated ($r = -0.327$, $p = 0.035$) whereas the success rate and height were not correlated ($r = -0.205$, $p = 0.194$). The success rate and optimal path were also moderately negatively correlated ($r = -0.464$, $p = 0.002$),

and the same occurred for the success rate and number of steps in the maze ($r = -0.506$, $p = 0.001$).

Furthermore, the percentage of loss in each challenge is presented in Table 2.

**TABLE 2. Percentage of participants who failed to solve a challenge, Study 2.**

| Sequence | Challenge 1 | Challenge 2 | Challenge 3 | Challenge 4 | Challenge 5 | Challenge 6 | Challenge 7 |
|---|---|---|---|---|---|---|---|
| 2.1 | 0% | 9% | 3% | 10% | 4% | 59% | 100% |
| 2.2 | 15% | 0% | 0% | 26% | 33% | 25% | 90% |
| 2.3 | 5% | 3% | 0% | 13% | 36% | 17% | 38% |
| 2.4 | 8% | 0% | 0% | 13% | 26% | 0% | 73% |
| 2.5 | 16% | 4% | 20% | 50% | 10% | 11% | 50% |
| 2.6 | 3% | 3% | 0% | 48% | 10% | 5% | 89% |

In Table 3, we present additional data metrics for each challenge, which we will use to calculate the participants' performance presented in the results section: average success time (time required to solve a challenge), average number of attempts that each participant needed to solve it (how many times participants pressed the "play" button to execute the program in the workspace), and average number of interactions with the workspace in which each participant engaged (blocks added or deleted in the workspace).

From the results shown in Tables 2 and 3, we infer that the last challenges of the sequences exceeded the participants' capabilities in many cases. This was not influenced by the time available as 60 minutes was sufficient and an appropriate duration for this study. In the case of sequence 1, 100% of the participants were unable to solve the last challenge, so we could identify a "floor effect" that might be affected by the participants' age. Considering this, Study 3 replicated Study 2 but increased the age of participants from 9-11 to 15-16 years old.

### C. STUDY 3

This study 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 twelve challenges in Kodetu. The first five challenges were training challenges with no block limit and were not considered for analysis. (The first challenge was explained step by step so that the functioning of Kodetu was understood.) The last seven challenges corresponded to one of the six challenge sequences prepared before and randomly assigned to each participant.

Table 4 shows the percentage of participants who started the study and failed to overcome each challenge, and Table 5 presents additional data metrics regarding the performance of the participants in Study 3. As the tables show, the vast majority of participants (85%) succeeded in the first challenges and became stuck in the last challenge, where they consumed the rest of the available time. With the results of Studies 2 and 3, we infer the difficulty associated with each challenge because the young participants of Study 2 suffered from a "floor effect" (low success rate in challenges too complex for their level of competence) while the participants of Study 3 suffered from a "ceiling effect" (high success rate in challenges too simple for their level of competence).

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 [$F(1,82) = 8.437$, $p = 0.005$].

### D. PERFORMANCE CALCULATION

As we saw in the previous analysis, the characteristics of each challenge had an influence on the participant's success rate. However, we must distinguish between the success rate and the participant's performance. *Success* means having solved a challenge, while *performance* involves more parameters, such as the time required to succeed, the number of attempts to solve a challenge, and the interactions with the workspace.

To calculate the performance based on the data collected, we created a Fuzzy Rule-Based System (FRBS). The FRBS works by using rules to encode knowledge from a broad area into an automated system [40].

We use the FRBS to solve the ambiguity of defining "low" or "high" performance in a tool such as Kodetu automatically. Using our FRBS, we obtain the value of performance (a number between 0-100) in each challenge as an output variable given the values of the input variables.

To define our FRBS, we take four variables as the input: the number of attempts per participant, the number of interactions per participant, the loss per challenge, and the average success time on a challenge. The output is the participant's performance in each challenge. Then, we map a given input to an output using fuzzy logic. To build the FRBS, we used the frbs R package. We created two identical FRBSs with the only difference being the input data. In the first system, we used the data from Study 2; while in the other system, we used the data from Study 3.

The FRBS consists of four functional parts. First, the fuzzification interface (fuzzifier) transforms the crisp inputs into degrees of membership functions (MFs) of the linguistic label of each variable. The MFs are shown in Fig. 3. All fuzzy input variables contain four MFs for each of the four associated linguistic labels: low, medium, high, and any. The linguistic label "any" contains all values of the variable and is used when, in a particular fuzzy rule, the corresponding variable is not significant and changes in the value should not be considered as an important factor to determine the performance.

Second, the knowledge base consists of the database and the rulebase. The database shown in Table 6 includes the

**TABLE 3.** Data metrics used to calculate performance, Study 2.

| SEQUENCE | DATA METRICS[a] | CHALLENGE 1 | CHALLENGE 2 | CHALLENGE 3 | CHALLENGE 4 | CHALLENGE 5 | CHALLENGE 6 | CHALLENGE 7 |
|---|---|---|---|---|---|---|---|---|
| 2.1 | AST | 00:06:09 | 00:01:42 | 00:01:15 | 00:03:43 | 00:00:27 | 00:08:45 | N/S |
| | ANA | 4.11 | 2.09 | 1.91 | 2.10 | 1.11 | 12.56 | 8.50 |
| | ANI | 99.83 | 59.20 | 47.84 | 133.29 | 14.89 | 212.04 | 209.30 |
| 2.2 | AST | 00:05:56 | 00:02:08 | 00:01:14 | 00:04:56 | 00:03:54 | 00:01:45 | 00:12:44 |
| | ANA | 4.22 | 3.35 | 2.00 | 3.65 | 6.67 | 2.33 | 8.30 |
| | ANI | 102.26 | 86.13 | 52.09 | 168.22 | 108.22 | 61.58 | 144.40 |
| 2.3 | AST | 00:02:30 | 00:01:31 | 00:01:02 | 00:04:27 | 00:06:55 | 00:06:29 | 00:07:35 |
| | ANA | 3.84 | 2.26 | 2.15 | 8.41 | 9.96 | 9.94 | 7.23 |
| | ANI | 53.57 | 57.91 | 41.73 | 157.91 | 194.29 | 169.50 | 123.38 |
| 2.4 | AST | 00:05:50 | 00:01:10 | 00:00:38 | 00:04:04 | 00:06:42 | 00:00:25 | 00:03:41 |
| | ANA | 4.52 | 2.13 | 1.13 | 2.61 | 7.11 | 1.08 | 4.55 |
| | ANI | 92.00 | 42.26 | 28.26 | 135.00 | 138.32 | 15.42 | 152.91 |
| 2.5 | AST | 00:06:43 | 00:00:41 | 00:04:16 | 00:04:38 | 00:04:01 | 00:03:32 | 00:02:14 |
| | ANA | 5.47 | 1.04 | 3.68 | 5.65 | 8.90 | 6.89 | 10.00 |
| | ANI | 123.81 | 28.85 | 152.32 | 113.10 | 129.50 | 123.56 | 146.25 |
| 2.6 | AST | 00:02:01 | 00:02:41 | 00:01:21 | 00:07:05 | 00:01:58 | 00:00:37 | 00:15:26 |
| | ANA | 2.53 | 4.62 | 2.37 | 9.83 | 1.86 | 1.84 | 10.67 |
| | ANI | 36.65 | 99.03 | 54.61 | 184.65 | 39.67 | 18.37 | 205.39 |

[a] AST= average success time, ANA= average number of attempts, and ANI= average number of interactions

**TABLE 4.** Percentage of participants who failed to solve a challenge, Study 3.

| SEQUENCE | CHALLENGE 1 | CHALLENGE 2 | CHALLENGE 3 | CHALLENGE 4 | CHALLENGE 5 | CHALLENGE 6 | CHALLENGE 7 |
|---|---|---|---|---|---|---|---|
| 3.1 | 0% | 0% | 11% | 0% | 0% | 13% | 57% |
| 3.2 | 13% | 0% | 0% | 14% | 0% | 0% | 17% |
| 3.3 | 0% | 0% | 0% | 0% | 0% | 10% | 44% |
| 3.4 | 0% | 0% | 8% | 0% | 0% | 0% | 27% |
| 3.5 | 0% | 0% | 13% | 14% | 17% | 0% | 20% |
| 3.6 | 0% | 0% | 0% | 0% | 0% | 0% | 58% |

fuzzy set definitions while the rulebase in Table 7 contains twelve fuzzy IF-THEN rules. These rules express the experts' knowledge in a form that the system can understand.

Third, the Mamdani inference engine performs the inference operations on the fuzzy IF-THEN rules. The Mamdani engine was selected because systems that use the Mamdani engine are designed to incorporate the form of the rulebase that we used in part 3, expressed in natural language.

Fourth, the defuzzification process (defuzzifier) center of gravity (COG), which is the standard method by which Mamdani systems obtain crisp values from linguistic values, is used.

**TABLE 5.** Data metrics used to calculate performance, Study 3.

| SEQUENCE | DATA METRICS[b] | CHALLENGE 1 | CHALLENGE 2 | CHALLENGE 3 | CHALLENGE 4 | CHALLENGE 5 | CHALLENGE 6 | CHALLENGE 7 |
|---|---|---|---|---|---|---|---|---|
| 3.1 | AST | 00:02:33 | 00:01:05 | 00:01:08 | 00:03:57 | 00:00:21 | 00:04:36 | 00:14:28 |
| | ANA | 3.22 | 2.22 | 1.89 | 3.75 | 1.00 | 7.75 | 12.29 |
| | ANI | 73.78 | 66.78 | 54.44 | 151.88 | 11.50 | 158.13 | 327.29 |
| 3.2 | AST | 00:02:56 | 00:01:15 | 00:00:46 | 00:01:47 | 00:02:04 | 00:00:33 | 00:09:40 |
| | ANA | 2.25 | 2.00 | 1.57 | 1.14 | 3.67 | 1.00 | 13.17 |
| | ANI | 74.50 | 77.86 | 46.43 | 128.57 | 66.33 | 21.67 | 233.83 |
| 3.3 | AST | 00:04:32 | 00:00:45 | 00:00:42 | 00:02:20 | 00:04:26 | 00:02:06 | 00:09:24 |
| | ANA | 3.10 | 2.10 | 2.40 | 5.00 | 10.10 | 8.50 | 12.56 |
| | ANI | 39.40 | 41.30 | 37.10 | 94.80 | 171.20 | 144.20 | 266.67 |
| 3.4 | AST | 00:02:15 | 00:00:47 | 00:00:37 | 00:02:07 | 00:02:19 | 00:00:16 | 00:06:32 |
| | ANA | 1.33 | 1.33 | 1.25 | 1.09 | 3.55 | 1.00 | 8.18 |
| | ANI | 69.33 | 41.58 | 35.08 | 154.45 | 59.73 | 7.36 | 244.82 |
| 3.5 | AST | 00:02:56 | 00:00:46 | 00:02:22 | 00:03:27 | 00:04:47 | 00:01:28 | 00:03:41 |
| | ANA | 1.88 | 1.13 | 1.75 | 3.86 | 6.83 | 2.00 | 7.40 |
| | ANI | 66.63 | 36.00 | 116.38 | 98.14 | 141.50 | 60.80 | 131.00 |
| 3.6 | AST | 00:00:57 | 00:01:39 | 00:00:45 | 00:02:47 | 00:01:42 | 00:00:31 | 00:15:25 |
| | ANA | 2.00 | 3.25 | 1.83 | 3.25 | 1.67 | 1.83 | 17.33 |
| | ANI | 28.42 | 81.75 | 49.75 | 83.33 | 49.42 | 19.75 | 372.25 |

[b] AST= average success time, ANA= average number of attempts, and ANI= average number of interactions

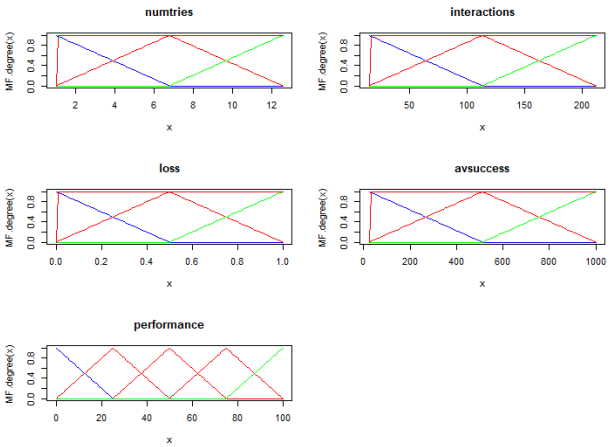


**FIGURE 3.** Plot of membership functions-FRBS.

The loss rate was one of the input variables for the FRBS that we used to measure the performance. However, there is a problem when the loss rate of challenge n-1 is larger than the loss of challenge n. Many factors could cause this issue that prevents us from using the loss rate of challenge n as an indicator of the performance in that challenge. Accordingly, the decision was made to set a low boundary: for the challenges in which the value of the loss rate was equal to or less than 20% of the value of the loss rate at the previous challenge, we reran the FRBS but without using the loss rate as an input variable. In Study 2, three challenges (2.4.6, 2.5.5, and 2.6.5) were affected by this issue.

## V. RESULTS

In order to answer our research questions, we have to calculate the performance of the participants on the maze-based programming challenges using the FRBS system presented above. Using the Mamdani inference method, the COG for defuzzification (output processor), and the rules based on

**TABLE 6.** Fuzzy database (input/output set variables).

| VARIABLE | TYPE | LINGUISTIC LABEL |
|---|---|---|
| Number of attempts per participant | Input | Low<br>Medium<br>High<br>Any |
| Interactions per participant | Input | Low<br>Medium<br>High<br>Any |
| Loss (percentage) | Input | Low<br>Medium<br>High<br>Any |
| Average success time | Input | Low<br>Medium<br>High<br>Any |
| Performance | Output | Very high<br>High<br>Average<br>Low<br>Very low |

the 12 linguistic propositions presented above, we obtained the crisp output values for each of the 42 challenges, which represent the performances of the participants on each challenge. The crisp output values ranged from 0 to 100, with 0 being the lowest performance and 100 being the highest (Table 9 and Table 10).

Although the one-way ANOVA to compare the effect of age on performance in Studies 2 and 3 is statistically significant [$F(1,82) = 4.674$, $p = 0.034$), in the overall ranking of the challenges based on the performance of participants in both studies, we observed that 52% of the challenges differ by less than two positions in the ranking, 31% differ by three to seven positions, and the rest differ by more than eight positions (For instance, the success rate of challenge 2.4.7 is 27% and the success rate of 3.4.7 is 73%; however, both challenges rank 5$^{th}$ on the overall ranking of the challenges).

Once the performance of the challenges is calculated, we conducted a simple regression analysis to determine the correlations between the dependent variable performance and the independent variables defined in the methodology section and investigated them in Studies 2 and 3 (width, height, total number of steps of the maze, length of the optimal path, turns, x-crosses, t-crosses, blocks available and block limit). The relationships between performance and the variables enable us to answer the research questions set at the beginning of the investigation.

Several variables were excluded from the analysis because they did not meet the assumptions of the regression analysis according to [41] (see Table 8). Thus, the independent variables that were used were the following: the number of steps, turns, and blocks available.

**TABLE 7.** Fuzzy rulebase specified from experts' knowledge.

| RULE NUMBER | RULEBASE |
|---|---|
| 1 | IF number of attempts is high and interactions is high and loss is high and average success time is high THEN performance is very low |
| 2 | IF number of attempts is high and interactions is high and loss is medium and average success time is high THEN performance is very low |
| 3 | IF number of attempts is medium and interactions is high and loss is medium and average success time is high THEN performance is low |
| 4 | IF number of attempts is medium and interactions is any and loss is high and average success time is any THEN performance is low |
| 5 | IF number of attempts is medium and interactions is medium and loss is medium and average success time is medium THEN performance is average |
| 6 | IF number of attempts is low and interactions is medium and loss is medium and average success time is medium THEN performance is average |
| 7 | IF number of attempts is high and interactions is medium and loss is medium and average success time is medium THEN performance is average |
| 8 | IF number of attempts is low and interactions is medium and loss is medium and average success time is low THEN performance is average |
| 9 | IF number of attempts is low and interactions is medium and loss is low and average success time is medium THEN performance is high |
| 10 | IF number of attempts is medium and interactions is medium and loss is low and average success time is low THEN performance is high |
| 11 | IF number of attempts is low and interactions is low and loss is medium and average success time is low THEN performance is very high |
| 12 | IF number of attempts is low and interactions is low and loss is low and average success time is low THEN performance is very high |

We conducted a simple regression analysis using the data of Study 2 and another analysis with those of Study 3. The regression analysis will determine whether and how much these variables affect the performance, described in the form of a function.

Regarding the performance of the participants in Study 2, a significant regression equation was found [$F(3,38) = 25.408$, $p < 0.000$] with an $R^2$ of 0.667. Considering this, the participants' predicted $performance_{study2}$ in Study 2 is defined by the following equation:

$$performance_{study2} = 120.848 - 0.985 * num\_steps - 8.289 * turns - 4.076 * blocks \quad (1)$$

**TABLE 8. Variables not used in the regression analysis.**

| Variable name | Rejection reason |
|---|---|
| Width | No linear relationship between the dependent and independent variable. |
| Height | No linear relationship between the dependent and independent variable. |
| Optimal path | Strong correlation with independent variable number of steps (Pearson coefficient = 0.96) |
| x-crosses | No linear relationship between the dependent and independent variable. |
| t-crosses | No linear relationship between the dependent and independent variable. |
| Block limit | Strong correlation with independent variable blocks (Pearson coefficient = 0.83) |

**TABLE 9. Crisp output of frbs - performance values, Study 2.**

| SEQUENCE | CHALLENGE 1 | CHALLENGE 2 | CHALLENGE 3 | CHALLENGE 4 | CHALLENGE 5 | CHALLENGE 6 | CHALLENGE 7 |
|---|---|---|---|---|---|---|---|
| 2.1 | 78.23 | 91.76 | 95.53 | 70.06 | 100 | 0 | 20.66 |
| 2.2 | 69.08 | 81.42 | 97.04 | 58.33 | 59.12 | 80.17 | 22.26 |
| 2.3 | 89.85 | 93.73 | 97.07 | 69.86 | 58.33 | 56.78 | 51.79 |
| 2.4 | 78.87 | 97.1 | 99.60 | 69.39 | 54.48 | 99.87 | 32.61 |
| 2.5 | 56.62 | 96.81 | 67.59 | 50.18 | 44.25 | 71.21 | 50 |
| 2.6 | 94.85 | 77.63 | 95.95 | 51.68 | 86.55 | 98.36 | 7.22 |

**TABLE 10. Crisp output of frbs - performance values, Study 3.**

| SEQUENCE | CHALLENGE 1 | CHALLENGE 2 | CHALLENGE 3 | CHALLENGE 4 | CHALLENGE 5 | CHALLENGE 6 | CHALLENGE 7 |
|---|---|---|---|---|---|---|---|
| 3.1 | 95.23 | 97.5 | 88.5 | 79.4 | 99.87 | 62.66 | 18.13 |
| 3.2 | 83.38 | 97.6 | 98.93 | 66.62 | 95.16 | 99.5 | 36.62 |
| 3.3 | 95.81 | 98.09 | 97.76 | 92.01 | 75 | 71.03 | 31.98 |
| 3.4 | 95.88 | 99.01 | 93.52 | 82.11 | 95.69 | 100 | 51.18 |
| 3.5 | 94.99 | 99.05 | 69.44 | 79.68 | 61.82 | 97.5 | 62.03 |
| 3.6 | 98.46 | 95.28 | 98.44 | 94.45 | 97.26 | 99.09 | 0 |

Regarding the performance of the participants in Study 3, a regression equation was found [$F(2, 39) = 15.61$, $p < 0.001$] with an $R^2$ of 0.445. The learners' predicted $performance_{study3}$ in Study 3 is defined by the following equation:

$$performance_{study3} = 119.122 - 1.274 * num\_steps - 2.752 * blocks \quad (2)$$

To evaluate both models, we calculated the Mean Absolute Error (MAE) [52], [53] for (1) ($MAE_1 = 10.528$) and for (2) ($MAE_2 = 11.989$). Since the lower the MAE is the better the model ($MAE_1 < MAE_2$) and the low $R^2$(0.445) of (2), we proceeded to define the difficulty function using (1).

Considering (1) and the results from the research of Latham, Seijts, and Crim [50] indicating that the higher the complexity of a task is, the lower the person's performance in that specific task, we infer that performance can be used to estimate difficulty:

$$dif = -performance \Rightarrow dif = -120.848 + 0.985 * num\_steps + 8.289 * turns + 4.076 * blocks \quad (3)$$

Moreover, given that previous efforts to define difficulty in games [11], [18], [33]–[35] are time dependent and that research has shown that limiting the time of an activity affects performance [56]–[59], we suggest that difficulty also depends on the time given to solve the challenge. Using a linear regression, we predicted the average success time (ASt) given the maze and coding characteristics of a challenge. A regression equation was found ($F(2, 39) = 14.79$, $p < 0.000$) with an $R^2$ of 0.0.5012:

$$ASt = 14.303 * num\_steps + 23.891 * blocks \quad (4)$$

Having predicted an estimation of the average success time (ASt), we infer that by introducing a time limit that equals the ASt, half of the participants will be able to succeed at that level. Therefore, if we grant more time, the percentage of participants who succeed in the challenge will be higher and vice versa. However, if only 50% of the participants succeeded in a challenge, we considered it to be difficult. Similarly, we assume that if the time limit corresponds with an extra time of 25% of the ASt, we consider the time limit to have no negative impact on the difficulty of the challenge; in addition, if participants have more than 25% extra time regarding the ASt, the difficulty of the challenge will be lower. Considering this, we moderate the factor of time limit with a coefficient of 0.8. Consequently, our estimation for the difficulty function that considers time is:

$$diftime = \frac{dif * ASt}{0.8 * time\_limit} \quad (5)$$

## VI. 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. By measuring this effect, we propose an estimation for the difficulty of block-based maze programming challenges. Our results are based on the analysis

of the platform log data gathered from 326 learners during three studies in which participants were tasked with solving maze-based programming challenges in the online platform Kodetu.

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. Furthermore, consider the fact that maze loops have not affected the participants' performance, we propose that the high failure rate was caused by the limited time given to complete the sequences, as well as the effect of the rest of the variables present in the challenges.

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, are demanding in terms of the time to succeed, the number of interactions with the platform, and the number of attempts to solve them. This confirms the results from prior research [18], [42] as the use of blocks of conditionals and loops to solve a challenge requires challenging CT competences [43], [44]. In addition, the difficulty added by the block limit is because the learners are forced to use conditional and loop blocks to solve the maze instead of using only the sequence of movement blocks. 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.

We analyzed the data in more depth by measuring the success rate, the loss rate, the average success time, the number of interactions in a challenge, and the number of attempts to solve a challenge. We also found that the sequences of challenges that we created were too complex for most of the participants (''floor effect''). Considering this, we conducted the same experiment with older participants (Study 3).

Unlike what happened in Study 2, we observed that the success rate in Study 3 during the first challenges was very high, and almost every participant was able to solve them (''ceiling effect''). However, in the last challenges of the sequences, there was a remarkable increase in the time necessary to succeed, the number of interactions, and the number of attempts to solve a challenge. This indicates that age affects the success rate and that the challenges requiring higher CT competencies are also demanding for older participants.

We developed an FRBS to calculate the performance based on the data metrics of the average success time, loss, interactions, and number of attempts. We noted that although the value of the performance in each challenge differs depending on the learners' age [45], the overall ranking of challenges based on performance is similar, showing that the CT competencies required to solve a maze-based programming challenge are difficult to achieve for both younger and older learners.

Finally, the main finding of this research was the definition of an estimation of the difficulty for block-based maze programming challenges. According to the results of the regression analysis and previous literature, a challenge is difficult when it contains turns, when the total number of steps is substantial, and when movement, conditionals and loop blocks are provided to the learner; thus, our estimation of difficulty is presented in (3).

Considering that we wanted to focus our research on investigating the effect of the maze characteristics and the coding limitations of the game, we designed our three studies without setting a time limit for the completion of each challenge. However, putting a time limitation to solve a challenge is considered a factor that affects the difficulty of the challenge [56]–[59]. Therefore, we estimated a time-dependent function of difficulty as (5). We provide an estimation of the average time to succeed in a challenge based on the characteristics of the maze and the coding limitations, and we suggest that this estimation can be used as a threshold for choosing an adequate time limitation. The use of time limitations does not mean that increasing the time limit for a difficult challenge makes it easier; nevertheless, we suggest that the time limit should be considered an additional limitation to the learner.

The limitations of each study were mentioned in the methodology section because they motivated the main changes in the design of the next study. However, some overall limitations of the research presented here should be highlighted. First, due to the lack of specific rules for creating sequences of challenges in block-based maze games, the design of the sequences was conducted mostly in an empirical way that may have affected some participants' performance in the studies. However, we considered this limitation when defining the performance with the FRBS and minimized its further effect in our research. Second, we identified only ten variables that may affect the difficulty of a maze-based game. We consider this research to be the first step in identifying and defining a proper estimation of difficulty in this type of game. However, more variables may affect the difficulty of this type of game, and they should be investigated and added to the function in order to increase the function's accuracy.

The results of this research should be considered when considering how to design learning paths to develop and enhance CT competences via maze-based programming challenges. The data gathered in our three studies contribute to a clearer understanding of the maze characteristics and coding limitations that affect the difficulty of these challenges. While previous research has focused on calculating the difficulty of challenges postplay, our results show that difficulty can be predicted while designing the learning paths. Considering the prospects and limitations of this research, future work should focus on automatically generating programming challenges with adaptive difficulty that will offer personalization of the learning paths. Our research results encourage the design of

new experiments to explore the effects of providing personalized learning on the acquisition of CT skills by learners.

## ACKNOWLEDGMENT

A number of institutions have backed and cofinanced this project.

## REFERENCES

[1] *Computer and Information Technology Occupations?: Occupational Outlook Handbook: U.S. Bureau of Labor Statistics*. Accessed: Sep. 30, 2020. [Online]. Available: https://www.bls.gov/ooh/computer-and-information-technology/home.htm

[2] *ICT Specialists in Employment–Statistics Explained*. Accessed: Sep. 30, 2020. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_in_employment

[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*, 2020, p. 11.

[4] OECD. (2012). *Better Skills, Better Jobs, Better Lives*. [Online]. Available: https://www.oecd-ilibrary.org/content/publication/9789264177338-en.

[5] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006, doi: 10.1145/1118178.1118215.

[6] I. Jeon and K.-S. Song, "The effect of learning analytics system towards Learner's computational thinking capabilities," in *Proc. 11th Int. Conf. Comput. Autom. Eng. (ICCAE)*, New York, NY, USA, 2019, pp. 12–16, doi: 10.1145/3313991.3314017.

[7] 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.

[8] B. Bontchev and R. Panayotova, "Towards automatic generation of serious maze games for education," *Serdica J. Comput.*, vol. 11, nos. 3–4, pp. 249–278, Nov. 2018.

[9] Ž. Ternik, A. Koron, T. Koron, and I. N. Šerbec, "Learning programming concepts through maze game in scratch," in *Proc. Eur. Conf. Games Based Learn.*, Reading, U.K., Oct. 2017, pp. 661–670. Accessed: Feb. 2, 2018. [Online]. Available: https://search.proquest.com/docview/1967728949/abstract/573F8E28B0344D95PQ/1

[10] A. Tlili, M. Denden, F. Essalmi, M. Jemni, Kinshuk, N.-S. Chen, and R. Huang, "Does providing a personalized educational game based on personality matter? A case study," *IEEE Access*, vol. 7, pp. 119566–119575, 2019, doi: 10.1109/ACCESS.2019.2936384.

[11] F. J. Gallego-Durán, R. Molina-Carmona, and F. Llorens-Largo, "Measuring the difficulty of activities for adaptive learning," *Universal Access Inf. Soc.*, vol. 17, no. 2, pp. 335–348, Jun. 2018, doi: 10.1007/s10209-017-0552-x.

[12] J. Guggemos, "On the predictors of computational thinking and its growth at the high-school level," *Comput. Educ.*, vol. 161, Feb. 2021, Art. no. 104060, doi: 10.1016/j.compedu.2020.104060.

[13] L. Zhang, J. Nouri, and L. Rolandsson, "Progression of computational thinking skills in swedish compulsory schools with block-based programming," in *Proc. 22nd Australas. Comput. Educ. Conf.*, New York, NY, USA, Feb. 2020, pp. 66–75, doi: 10.1145/3373165.3373173.

[14] S. Trilles and C. Granell, "Advancing preuniversity students' computational thinking skills through an educational project based on tangible elements and virtual block-based programming," *Comput. Appl. Eng. Educ.*, vol. 28, no. 6, pp. 1490–1502, 2020, doi: 10.1002/cae.22319.

[15] I. Fronza, L. Corral, and C. Pahl, "Combining block-based programming and hardware prototyping to foster computational thinking," in *Proc. 20th Annu. SIG Conf. Inf. Technol. Educ.*, New York, NY, USA, Sep. 2019, pp. 55–60, doi: 10.1145/3349266.3351410.

[16] 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. Rijeka, Croatia: InTech, 2018.

[17] D. Weintrop, "Block-based programming in computer science education," *Commun. ACM*, vol. 62, no. 8, pp. 22–25, Jul. 2019, doi: 10.1145/3341221.

[18] R. Pelánek and T. Effenberger, "Design and analysis of microworlds and puzzles for block-based programming," *Comput. Sci. Educ.*, pp. 1–39, Oct. 2020, doi: 10.1080/08993408.2020.1832813.

[19] W. Min, B. Mott, K. Park, S. Taylor, B. Akram, E. Wiebe, K. E. Boyer, and J. Lester, "Promoting computer science learning with block-based programming and narrative-centered gameplay," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 654–657, doi: 10.1109/CoG47356.2020.9231881.

[20] A. Smith, B. Mott, S. Taylor, A. Hubbard Cheuoua, J. Minogue, K. Oliver, and C. Ringstaff, "Designing block-based programming language features to support upper elementary students in creating interactive science narratives," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, Portland, OR, USA, Feb. 2020, p. 1327, doi: 10.1145/3328778.3372653.

[21] C. Gleasman and C. Kim, "Pre-service teacher's use of block-based programming and computational thinking to teach elementary mathematics," *Digit. Experiences Math. Educ.*, vol. 6, no. 1, pp. 1–39, 2020.

[22] D. Weintrop and U. Wilensky, "To block or not to block, that is the question: Students' perceptions of blocks-based programming," in *Proc. 14th Int. Conf. Interact. Design Children*, New York, NY, USA, Jun. 2015, pp. 199–208, doi: 10.1145/2771839.2771860.

[23] T. Effenberger and R. Pelánek, "Towards making block-based programming activities adaptive," in *Proc. 5th Annu. ACM Conf. Learn. Scale*, New York, NY, USA, Jun. 2018, pp. 1–4, doi: 10.1145/3231644.3231670.

[24] S. Ludi, "Position paper: Towards making block-based programming accessible for blind users," in *Proc. IEEE Blocks Beyond Workshop (Blocks Beyond)*, Oct. 2015, pp. 67–69, doi: 10.1109/BLOCKS.2015.7369005.

[25] K. Park, B. W. Mott, W. Min, K. E. Boyer, E. N. Wiebe, and J. C. Lester, "Generating educational game levels with multistep deep convolutional generative adversarial networks," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2019, pp. 1–8, doi: 10.1109/CIG.2019.8848085.

[26] M.-V. Aponte, G. Levieux, and S. Natkin, "Difficulty in videogames: An experimental validation of a formal definition," in *Proc. 8th Int. Conf. Adv. Comput. Entertainment Technol. (ACE)*, New York, NY, USA, 2011, pp. 1–8, doi: 10.1145/2071423.2071484.

[27] M. Csikszentmihalyi and I. S. Csikszentmihalyi, *Optimal Experience: Psychological Studies of Flow in Consciousness*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[28] 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, Switzerland: Springer, 2016, pp. 417–428, doi: 10.1007/978-3-319-39483-1_38.

[29] 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," *J. Comput. Assist. Learn.*, vol. 36, no. 4, pp. 502–513, 2020, doi: 10.1111/jcal.12416.

[30] K. Kiili, S. de Freitas, S. Arnab, and T. Lainema, "The design principles for flow experience in educational games," *Procedia Comput. Sci.*, vol. 15, pp. 78–91, 2012, doi: 10.1016/j.procs.2012.10.060.

[31] S. Sampayo-Vargas, C. J. Cope, Z. He, and G. J. Byrne, "The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game," *Comput. Educ.*, vol. 69, pp. 452–462, Nov. 2013, doi: 10.1016/j.compedu.2013.07.004.

[32] J. D. Lomas, K. Koedinger, N. Patel, S. Shodhan, N. Poonwala, and J. L. Forlizzi, "Is difficulty overrated?: The effects of choice, novelty and suspense on intrinsic motivation in educational games," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Denver, CO, USA, May 2017, pp. 1028–1039, doi: 10.1145/3025453.3025638.

[33] M.-V. Aponte, G. Levieux, and S. Natkin, "Measuring the level of difficulty in single player video games," *Entertainment Comput.*, vol. 2, no. 4, pp. 205–213, Jan. 2011, doi: 10.1016/j.entcom.2011.04.001.

[34] T. Constant, G. Levieux, A. Buendia, and S. Natkin, "From Objective to Subjective Difficulty Evaluation in Video Games," in *Human-Computer Interaction–(INTERACT)*, vol. 10514, R. Bernhaupt, G. Dalvi, A. Joshi, D. K. Balkrishan, J. O'Neill, and M. Winckler, Eds. Cham, Switzerland: Springer, 2017, pp. 107–127.

[35] M. Szabo, K. D. Pomazi, B. Radostyan, L. Szegletes, and B. Forstner, "Estimating task difficulty in educational games," in *Proc. 7th IEEE Int. Conf. Cognit. Infocommun. (CogInfoCom)*, Oct. 2016, pp. 000397–000402, doi: 10.1109/CogInfoCom.2016.7804582.

[36] M. S. McClendon. (2001). *The Complexity and Difficulty of a Maze*. presented at Bridges: Math. Connections Art, Music, Sci. Accessed: May 30, 2018. [Online]. Available: http://at.yorku.ca/c/a/h/d/25.htm

[37] A. Eguiluz, M. Guenaga, P. Garaizar, and C. Olivares-Rodriguez, "Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 256–261, Jan. 2020, doi: 10.1109/TETC.2017.2768550.

[38] C. Olivares-Rodríguez, M. Guenaga, and P. Garaizar, "Using children's search patterns to predict the quality of their creative problem solving," *Aslib J. Inf. Manage.*, vol. 70, no. 5, pp. 538–550, Jan. 2018, doi: 10.1108/AJIM-05-2018-0103.
[39] R. Israel-Fishelson, A. Hershkovitz, A. Eguíluz, P. Garaizar, and M. Guenaga, "The associations between computational thinking and creativity: The role of personal characteristics," *J. Educ. Comput. Res.*, vol. 58, no. 8, pp. 1415–1447, 2020, doi: 10.1177/0735633120940954.
[40] H. Jiang, Q. Song, K. Gao, Q. Song, and X. Zhao, "Rule-based expert system to assess caving output ratio in top coal caving," *PLoS ONE*, vol. 15, no. 9, Sep. 2020, Art. no. e0238138, doi: 10.1371/journal.pone.0238138.
[41] J. W. Osbourne and E. Waters, "Four assumptions of multiple regression that researchers should always test," *Practical Assessment, Res., Eval.*, vol. 8, no. 1, p. 2, 2002, Accessed: Jul. 9, 2020. [Online]. Available: https://scholarworks.umass.edu/pare/vol8/iss1/2/
[42] S.-W. Chan, C.-K. Looi, and B. Sumintono, "Assessing computational thinking abilities among Singapore secondary students: A Rasch model measurement analysis," *J. Comput. Educ.*, Oct. 2020, doi: 10.1007/s40692-020-00177-2.
[43] J. Moreno-León and G. Robles, "Dr. Scratch: A Web tool to automatically evaluate scratch projects," in *Proc. Workshop Primary Secondary Comput. Educ.*, London, Nov. 2015, pp. 132–133, doi: 10.1145/2818314.2818338.
[44] A. Wilson, T. Hainey, and T. Connolly, "Evaluation of computer games developed by primary school children to gauge understanding of programming concepts," in *Proc. 6th Eur. Conf. Games Based Learn.*, P. Felicia, Ed. ACPIL, 2012, pp. 549–558.
[45] J.-J. Navarro, J. García-Rubio, and P. R. Olivares, "The relative age effect and its influence on academic performance," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0141895, doi: 10.1371/journal.pone.0141895.
[46] W.-H. Wu, H.-C. Hsiao, P.-L. Wu, C.-H. Lin, and S.-H. Huang, "Investigating the learning-theory foundations of game-based learning: A meta-analysis," *J. Comput. Assist. Learn.*, vol. 28, no. 3, pp. 265–279, 2012, doi: 10.1111/j.1365-2729.2011.00437.x.
[47] M. Prensky, "Digital game-based learning," *Comput. Entertainment*, vol. 1, no. 1, p. 21, Oct. 2003, doi: 10.1145/950566.950596.
[48] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation," *Comput. Educ.*, vol. 52, no. 1, pp. 1–12, Jan. 2009, doi: 10.1016/j.compedu.2008.06.004.
[49] V. Aleven, E. Myers, M. Easterday, and A. Ogan, "Toward a framework for the analysis and design of educational games," in *Proc. 3rd IEEE Int. Conf. Digit. Game Intell. Toy Enhanced Learn.*, Apr. 2010, pp. 69–76, doi: 10.1109/DIGITEL.2010.55.
[50] C. Malliarakis, M. Satratzemi, and S. Xinogalos, "CMX: The effects of an educational MMORPG on learning and teaching computer programming," *IEEE Trans. Learn. Technol.*, vol. 10, no. 2, pp. 219–235, Apr. 2017, doi: 10.1109/TLT.2016.2556666.
[51] S. Xinogalos, M. Satratzemi, and C. Malliarakis, "Microworlds, games, animations, mobile apps, puzzle editors and more: What is important for an introductory programming environment?" *Educ. Inf. Technol.*, vol. 22, no. 1, pp. 145–176, Jan. 2017, doi: 10.1007/s10639-015-9433-1.
[52] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, pp. 79–82, 2005, doi: 10.3354/cr030079.
[53] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?" *Numer. Methods*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmdd-7-1525-2014.
[54] A. Amory and R. Seagram, "Educational game models: Conceptualization and evaluation: The practice of higher education," *South Afr. J. Higher Educ.*, vol. 17, no. 2, pp. 206–217, Jan. 2003. [Online]. Available: https://hdl.handle.net/10520/EJC36981
[55] R. Ibrahim and A. Jaafar, "Educational games (EG) design framework: Combination of game design, pedagogy and content modeling," in *Proc. Int. Conf. Electr. Eng. Informat.*, Aug. 2009, pp. 293–298, doi: 10.1109/ICEEI.2009.5254771.
[56] W. M. Davidson and J. B. Carroll, "Speed and level components in time-limit scores: A factor analysis," *Educ. Psychol. Meas.*, vol. 5, no. 4, pp. 411–427, Dec. 1945, doi: 10.1177/001316444500500408.
[57] J. Mullane and S. McKelvie, "Effects of removing the time limit on first and second language intelligence test performance," *Practical Assessment, Res., Eval.*, vol. 7, no. 1, p. 123, Nov. 2019, doi: 10.7275/ph8y-yz89.
[58] D. E. Powers and M. E. Fowles, "Effects of applying different time limits to a proposed GRE writing test," *J. Educ. Meas.*, vol. 33, no. 4, pp. 433–452, Dec. 1996, doi: 10.1111/j.1745-3984.1996.tb00500.x.
[59] M. A. DeDonno, K. Rivera-Torres, A. Monis, and J. F. Fagan, "The influence of a time limit & bilingualism on scholastic school assessment test performance," *North Amer. J. Psychol.*, vol. 16, no. 2, pp. 211–223, 2014.

**IOANNA KANELLOPOULOU** received the Diploma of Engineering degree in computer engineering and informatics and the master's degree in computational mathematics-informatics in education from the University of Patras, Greece, in 2012 and 2016, respectively. She is currently pursuing the Ph.D. degree with the Faculty of Engineering, University of Deusto. Her research interests include data science and the implementation of information and communications technology in education. She is also the Marie Sklodowska-Curie Fellow.

**PABLO GARAIZAR** received the B.Sc. degree in psychology and the Ph.D. degree in computer engineering. He is currently working with the University of Deusto, Bilbao, Spain, as a Lecturer and a Researcher.

**MARILUZ GUENAGA** received the Ph.D. degree in computer engineering from the University of Deusto, Bilbao, Spain, in 2007. She is currently a Lecturer with the University of Deusto. She is also responsible for the Deusto Learning Laboratory Research Group.