Exploring the User Experience and Effectiveness of Mobile Game-based Learning in Higher Education

Nicos Kasenides[1][0000−0002−1562−3839], Andriani Piki[1][0000−0003−0376−1713], and Nearchos Paspallis[1][0000−0002−2636−7973]

University of Central Lancashire – Cyprus, Larnaca, Cyprus
nkasenides@uclan.ac.uk

Abstract. Recent technological and societal changes have heightened the uptake of mobile educational games across all subjects and levels of education. Nevertheless, developing mobile games with a set of design guidelines in mind, does not necessarily imply they will naturally engage learners and improve the learning experience. Building on this tenet, this study aims to explore how higher education students experience learning programming principles by playing a maze-solving mobile educational game; evaluate how effective the game's features are in terms of engaging students; and inquire how effective mobile devices are as a learning platform for teaching programming and algorithmic thinking. The overarching theme that emerged by analysing the gathered data is the need to promote 'engagement by design', indicating that subtle synergies are required between the technological qualities and design features of a mobile educational game, the pedagogical context, and human factors, towards achieving the desired level of learner engagement.

Keywords: Educational Games · Mobile Learning · Mobile Game-Based Learning · Learner Engagement · Higher Education

1 Introduction

The ongoing advances in mobile digital technology and the widespread availability and flexibility of mobile devices have increased the application of mobile learning [5, 58] and educational games [7, 27] across all levels of education. The utilisation of mobile devices for teaching and learning was further accelerated during Covid-19 pandemic, revealing their wide-ranging capabilities as learning devices [48]. Mobile learning and Game-Based Learning (GBL) approaches have the potential to provoke learner autonomy [3], positive learning outcomes, and experiential learning [7, 26]. Nevertheless, the exploitation and evaluation of mobile educational games for learning programming and algorithmic thinking is limited. The importance of developing problem-solving, critical thinking, and coding skills in higher education is well documented in the literature [6,63]. Several initiatives are launched towards promoting coding skills, such as hackathons and coding competitions (e.g., Code Week [39] and Hour of Code [62]), and using computer games as learning activities [7,11,18,26,33]. Computer games, including Massively Multiplayer Online Games (MMOGs), are increasingly employed as agents of motivation in higher education [7]. Even though GBL and educational gamification approaches are popular in the context of STEAM (Science, Technology, Engineering, Arts, and Mathematics) subjects, including programming and algorithmic thinking [23, 30, 37], such endeavors focus predominantly on desktop and laptop computer games (such as the Code Combat educational game [30]), while only a few games target mobile devices (like the Lightbot [23] and Run Marco [37] mobile games). Furthermore, there is a limited number of studies evaluating the user experience and effectiveness of mobile games for learning programming in higher education. In this paper, we aim to fill these gaps by conducting an empirical study to explore the learner experience with a mobile educational game, namely aMazeChallenge, and evaluate how effective it is in terms of learning and engagement. aMazeChallenge [27, 43] is an interactive, multiplayer, mobile educational game developed to introduce learners to the fundamental principles of programming and algorithmic thinking. The following research questions are addressed:

1. How do higher education students experience learning programming through a mobile game-based educational activity?
2. Which aspects and game features affect engagement with mobile game-based learning?
3. How effective are mobile devices as a platform for learning programming and algorithmic thinking?

2 Related Work

2.1 Game-based learning

Amongst prevalent innovating pedagogies, playful learning emerges first in terms of immediacy and timescale to widespread implementation, compared to other technology-mediated pedagogical methods, such as learning with robots, drone-based learning, and virtual environments [18]. Educational or serious games, game thinking, and GBL approaches are widely employed in learning and skills development initiatives [8, 11, 20, 34, 59]. GBL is commonly associated with positive outcomes including increased learner engagement with learning content; active participation in educational activities; enhanced understanding, knowledge acquisition, skills development [52]; and improved emotional and motivational outcomes [7, 11, 20, 26, 33]. GBL triggers learners' interest and leverages curiosity-driven learning [1,7,18,20,59], hence offering an alternative to the conventional focus on memorisation, assessment, and performance traits in education, which often counteract active exploration, collaboration, and understanding [18]. Through GBL activities, learners are engaged in playful quests which present genuine opportunities for developing critical thinking, problem-solving, analytical and communication skills [18]. By blending entertainment and learning elements, serious games constitute a powerful educational medium. Games like virtual environments and educational escape rooms [31, 34, 51] activate learning, intensify learner involvement and blend formal and informal learning experiences through dynamic, seamless, and multimodal interactions [56].

2.2 Mobile game-based learning

Mobile Game-Based Learning (mGBL) fuses the unique and distinguishing features of mobile devices (such as portability, social interactivity, context sensitivity, ubiquity, autonomy, and flexibility) [3, 58] with the captivating and engaging nature of games [7, 11, 20, 59] transforming mobile devices into an appealing learning medium. Contemporary research has shown that the ubiquity of mobile devices and the seamless integration of mobile technology in both formal and informal learning contexts is conducive to learner engagement [3, 24, 53]. Nevertheless, formulating an effective mGBL approach is not straightforward. Introducing 'any' mobile game or gamified application as a supplementary tool for enriching the instructional process will not automatically engage students [49]. Even if the game initially captures students' attention, there are several factors which need to be considered for the successful design, development, and deployment of mobile educational games [50]. These are discussed next.

2.3 Challenges and Gaps in Game-Based Learning

While the benefits associated with educational games are evident, recent research findings highlight several challenges in designing, developing, evaluating, and using educational games [18, 20, 27]. The following paragraphs discuss key pedagogical and technological considerations.

Pedagogical considerations. Firstly, although GBL is entwined with increased motivation and engagement, very few studies have examined the influence of gamification on the varied dimensions of engagement [1]. Learner engagement is a complex concept that dynamically transpires through the interactions of the learner with a technology-mediated learning activity [45, 46]. Recent literature suggests that despite innovative applications and technological advancements, situations like student disengagement, rising levels of dropouts [8], course withdrawals, surface learning, and students' decreasing motivation to learn are still present and peaked following the intricate effects of Covid-19 pandemic [36, 47].

Secondly, educational games need to establish a symbiotic relationship between educational and entertainment requirements [35] which are not always aligned [27]. Furthermore, entertainment is not always prioritised in traditional instructional approaches which are guided by strict curricula, fixed timelines, and formal teaching and assessment methods, which constitutes some of the key reasons why the use of games has not yet had a profound influence on education [18]. Therefore, new modes of assessment and delivery need to be developed which are more relevant and aligned to today's needs [44].

Another challenge lies in the fact that, despite the increasing utilisation of GBL approaches and the rising number of educational games available, no comprehensive policy exists for the use of games in education [60]. Furthermore, the connection between gamification and actual learning still appears to be vague [20]. This can be attributed to the complexity inherent in measuring the benefits of GBL compared to traditional learning approaches [18], and the increased time and effort required for assessing learners' performance, which may inevitably affect educators' willingness to incorporate technology in education [44]. Moreover, educators may need to be trained in devising efficient monitoring, feedback, and assessment strategies appropriate for mGBL [18,28,31,44,49,60], while also contemplating the mobile technologies at hand and their inherent constraints [50]. Thus, beyond the pedagogical challenges, manifold technological considerations emerge for the successful deployment of mGBL.

Technological considerations. Students tend to find mobile learning less convenient and more frustrating [21], compared to learning using a personal computer [15], for several technology-oriented reasons, including: small screen size constraining the text and controls that can be displayed [2, 4, 55]; poor user interface (UI) design hindering information transfer and making learning more cognitively demanding [2]; operating system compatibility [41]; variability in Internet speed [34,50]; and short battery life affecting resource-demanding games, all of which are crucial for the successful deployment of mGBL. Ensuring high UI design quality [2,16] and creating mobile-friendly content [28] are important aspects to consider when promoting mobile learning. These aspects also bring forward various human factors. The effects of mobile learning on lowering the learners' cognitive load and increasing their achievement depend heavily on the quality of the mobile learning design [2, 9, 57]. The design itself can influence user acceptance, adoption, and use of a mobile learning application. Hence, usability and user experience (UX) goals must be thoughtfully considered along with pedagogical objectives [42, 49]. Despite recent developments, and the fact that most higher education students describe themselves as technologically savvy and active on social media mobile apps, many students still lack essential digital literacy skills to embark on playing a new game [41] or participate online [21]. Developing digital skills can reduce negative experiences, such as frustration while trying to understand how the game works [27], or disappointment while attempting to complete advanced tasks through the game [29].

There is also a paradox regarding whether mobile learning enables or hinders learning and engagement. On one hand, the literature shows that well-designed mobile apps featuring multimedia or gamified elements can increase learner participation, contextualise learning, promote inquiry-based learning, and improve learners' achievements compared to traditional, formal learning approaches [17, 32, 57]. On the other hand, when mobile apps are not developed with learning objectives and learner engagement in mind, this may act as a barrier to learning [49]. For certain learning tasks, presenting dense information with small fonts on a single screen [2, 13], or displaying redundant information and animations [32], may be perceived as a distraction by students, hence inhibiting their concentration and learning [28], increasing their cognitive load [2, 13, 32] and creating mental overheads, which may negatively impact learners' affective and cognitive involvement [16] and the learning outcomes altogether [8]. These findings suggest that further research is needed on the learning benefits of mobile apps, including competitive and collaborative games for learning programming [38, 43].

3 Mobile Game-Based Learning with aMazeChallenge

To address the research questions, we utilise aMazeChallenge[1] - an interactive, multiplayer, mobile game developed for teaching programming and algorithmic thinking to higher education students. This section describes the key learning objectives, features, game rules and components of aMazeChallenge.

3.1 Learning objectives

The learning objectives of aMazeChallenge focus on the skills that higher education students must acquire on their journey to become successful programmers.

(i) Read and understand existing code. Before writing their own code, students need to understand the concept of programming statements and be capable of reading and interpreting code. This objective is achieved via a tutorial environment in aMazeChallenge, which links the actions of an avatar (such as moving and turning) to programming commands or statements.

(ii) Improve or fix existing code. Students must be able to identify syntactic and semantic errors in pre-written code and fix them. To achieve this, aMazeChallenge provides players with a set of incrementally harder challenges for training, enabling them to develop their skills and confidence.

(iii) Write code from scratch. When students feel confident in reading, understanding, and improving the existing code, aMazeChallenge reinforces code writing by allowing students to easily form their code by dragging blocks in the workspace while minimising syntactic errors by accepting only valid connections between the blocks.

(iv) Write clean and efficient code. This slightly more advanced programming skill is a significant aspect of software development. aMazeChallenge rewards players who create more efficient code, as these players may generally exit the maze faster and thus score more points than other players.

The above skills enable students to develop problem-solving, algorithmic, and critical thinking skills, which are not limited to Computer Science, but are transferable to multiple domains. Through the use of graphics, sound effects, interactive gameplay, learn by try-and-fail, playful learning, unlimited attempts, competition, and other game-like elements, aMazeChallenge aims to engage students with the learning process and achieve these objectives.

[1] aMazeChallenge can be downloaded from the Google Play Store: https://play.google.com/store/apps/details?id=org.inspirecenter.amazechallenge

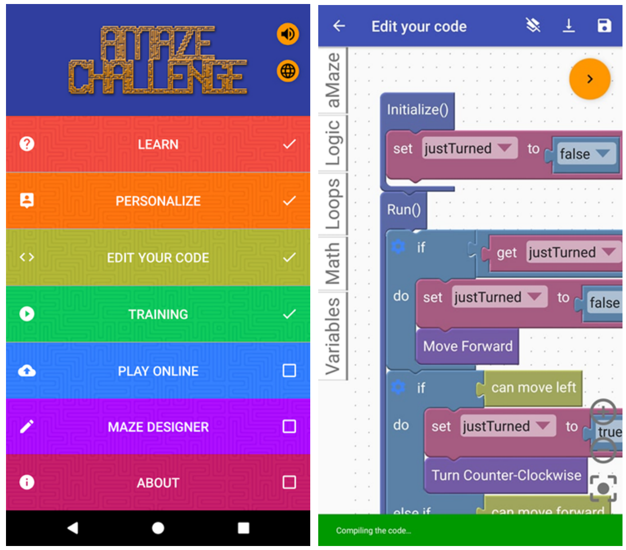
3.2 Features

aMazeChallenge is introduced to the players through the *Learning* section (fig. 1(a)) featuring a text-based tutorial enhanced with screenshots of each game feature along with explanations on how it works. Players can optionally personalise their character by selecting an avatar icon and color. To play online, players must also set their name and email address, through the *Personalization* screen, which serves as an engagement medium. They are then directed to create the first version of their code by using the workspace found in the *Code Editor* screen (fig. 1(b)). In this screen, players drag blocks to create their code, or load pre-defined samples of maze-solving algorithms. When ready, players can see how their code affects the behavior of an in-game avatar by visiting the *Training* section. This section includes a set of challenges that are initially very simple but get progressively more difficult, aiming to challenge players to produce more complex code by making incremental changes to adjust to the requirements of each challenge. When the players advance sufficiently through the tutorials, they may opt to compete against others in the *Online* mode. This mode makes a set of challenges available online, which players can join in and play against each other in real-time. Alternatively, players can also form teams and cooperate to create code for an online challenge, competing against other teams. Finally, aMazeChallenge also includes a *Maze Designer*, which allows players to create their own mazes choosing from a diverse set of settings such as the maze type, size, background image and audio, and wall color. These custom mazes can be added to the training mode, allowing players to practice in their own mazes to improve their code, while also aiming to increase learner engagement.

3.3 Game rules and components

aMazeChallenge features highly dynamic maze arenas (fig. 2), presenting a grid that is divided into cells representing the valid positions within the maze. Player avatars are positioned in these cells and can move to adjacent cells using commands issued by the players before the start of the game, using Google's block-based graphical language Blockly [22,54]. Players program their avatar to escape the maze by moving from the start to the finish cell. These cells are often located at opposite ends of the grid to allow for more challenging gameplay and are colour-coded to distinguish them from the rest (red for start and green for finish). The cells also have walls that prohibit the players from arbitrarily moving from one cell to another. The gameplay is based on commands the players have previously specified in their code for their avatar to execute. Once their code is compiled, each player can either choose to train solo or participate in an online challenge. In either case, players joining a challenge will play in turns with the game executing a single iteration of their instructions in each turn. The outcome of the player's code in each turn needs to result in a valid move (i.e., moving forward or turning toward a specified direction).

Moving through and interacting with the maze can be achieved by using game-specific functions such as `moveForward`, `turnClockwise`, `look`, and so on.



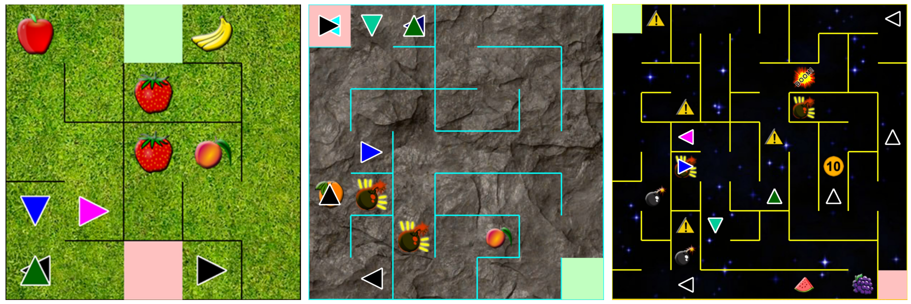
(a) (b)

Fig. 1. (a) aMazeChallenge Main menu (b) Code Editor

The maze includes objects that are randomly spawned inside the game arena, allowing a more dynamic and interactive experience, and also complicating the task of exiting the maze. For instance, objects like coins and fruits are beneficial to the players allowing them to gather points and increase their health when collected. On the other hand, objects such as traps and bombs have a negative effect, slowing players down or causing them to lose health. Players lose when their health falls to 0, but they can always start over. Such objects add an element of randomness and luck, aiming to make the game more interesting by allowing the players to find ways to either pursue or avoid objects. From an educational perspective, they encourage the use of decision-making programming constructs which can be used to decide whether an encountered object is beneficial or not. The winner of a challenge is the player who manages to exit the maze in the shortest time. Various maze-solving algorithms may have similar performance, and players may also utilise identical algorithms causing them to exit the maze within the same turn. To differentiate among the players who have exited the maze within the same turn, points are used as a secondary criterion, with those having more points being ranked higher [19].

3.4 Programming and code execution

The players can program their avatars using specific block-based instructions. Block-based languages have been successfully utilised in many educational games, especially at an introductory level [10]. To leverage Blockly assets, we



(a) (b) (c)

Fig. 2. Three different aMazeChallenge arenas, containing player avatars (colored triangles) and various types of interactive objects, during a multiplayer competition.

created a specialised Blockly library that contains specific commands related to aMazeChallenge. Coupled with existing Blockly commands for basic programming constructs like expressions, variables, math, and logic, this enables players to create a diverse set of programs. The custom-defined library includes blocks for moving, turning, navigation and directionality, and logical functions to retrieve the state of adjacent cells. For instance, the `Direction` class enumerates the possible directions an avatar could be facing. Similarly, the `Item` type enumerates the possible types of items a player's avatar could be facing (i.e., a penalty item, a reward item, or no item). This is typically utilised in the `look` command, which allows players to determine if one of such items is in their way and decide if they want to avoid or attempt to retrieve it.

The code structure is guided by the need to first initialise several values and then run a set of commands continuously at each turn until the exit is reached. To allow players to define these two distinct parts in their programs, we introduce two special functions: `Initialize` and `Run`. These functions are then called at specific points during the game to implement the functionality defined by the players. The `Initialize` function is executed only once, right before the gameplay starts. Conversely, the `Run` function is executed at each turn of the game by their avatar. This code is executed continuously until the player either loses, exits the maze, or the game is stopped. This function must return a single valid move to be performed at each turn by the player's avatar. If players place any blocks outside these two functions those blocks are ignored.

Players can create their code using Blockly's workspace, which provides an environment to edit code blocks by dragging them on the screen and joining them to form meaningful programming instructions. To make it easier for users to interact with this workspace, Blockly organises blocks into different panels containing sets of blocks based on their functionality. This also enables users to view only a certain number of blocks at a time, which is useful, especially when working with the limited screen size of a mobile device. In addition, blocks are colour-coded and have a specific shape based on their functionality, which makes it easier for players to choose an appropriate block. For instance, expression blocks only fit within the context of an expression (e.g., using them as the condition in an `if` statement). Using the metaphor of puzzle pieces, this limits the scope in which they can be used, and therefore prevents a wide set of errors that would occur during the code writing process. Hence, this satisfies the principle of error prevention. Furthermore, Blockly's workspace allows users to adjust the zoom level to accommodate different screen sizes and orientations. This makes the process of editing code significantly easier and more flexible as users can select to zoom out when reading code. On the contrary, users can zoom in when editing specific parts of the code. In addition to these, Blockly also allows users to delete blocks by dragging them toward a delete icon.

The default Blockly workspace is enhanced with additional features adding the capability of saving and loading code, as well as loading several sample algorithms (i.e., pre-written pieces of code that can be loaded by players, allowing them to see full solutions of specific maze problems, and inspiring them to write their own code). The aMazeChallenge code editor (fig., 1), includes the default Blockly workspace coupled with customised navigation and buttons that enable players to use these functionalities. Additionally, it includes options that allow the players to go back to the main menu or proceed through in-app controls.

When players finish their code, the next step is to convert it into a high-level language that can be executed by an interpreter. This process triggers a static checker which analyses statements in the program, detects errors (i.e., empty statements, run function returning an invalid move, cases where the code may execute indefinitely, etc.), and warns players to fix them before the code can be compiled. When such issues are detected, the code editor's interface displays colour-coded dialogs (red for errors and gold for warnings) containing relevant information and potential solutions. This approach improves the user experience by catching problematic circumstances and handling them, before causing any catastrophic situations such as the entire server crashing. Most importantly, code checking is part of the learning process, helping novice programmers understand different types of errors, how these are communicated, and how to fix them, all of which are crucial skills in the field of software development. Once the static checks have passed, a confirmation message is shown indicating that the code has been successfully compiled. The player is then automatically transferred back to the main menu and can opt to either train or play online.

4 Research Methodology

To explore the students' experiences with learning programming through a mobile educational game, and evaluate its effectiveness, we conducted an empirical study utilising mixed methods.

4.1 Study context and participants

The study involved the planning and execution of two data gathering phases during the first semester of the academic years 2021-22 and 2022-23. A total of 112 first-year undergraduate students were involved in the study, including both female and male students studying Computing, Electrical engineering, and Computer engineering. The students had varying backgrounds and prior experiences with programming, but they were all enrolled on the same introductory programming module and hence exposed to basic programming constructs in Java before participating in the study. User feedback was gathered through observation of live play sessions during which students competed online using their personal mobile devices, as well as through two questionnaires. Both phases were performed during the same period (respective teaching weeks) in each academic year, following an identical process to ensure that the study design remained as consistent as possible and that the presented results are comparable. The UI and features of aMazeChallenge, the content presented during the demonstrations, the difficulty of the maze challenges, and the conditions under which students engaged with the game also remained unchanged.

4.2 Data gathering

Prior to their participation in the study, students were asked to provide their consent by carefully reading and signing an informed consent form. In both phases of the study, data collection took place during class time. The first part of the study involved asking students to provide their responses to a background questionnaire including basic demographics, academic background, familiarity with programming, and opinion about programming. A 20-minute session was then delivered introducing aMazeChallenge to students in which the game's objectives and mechanics were discussed, and the students were encouraged to download the game during their own time to try it out. Predictably, only a small number of students downloaded the app and interacted with it outside class time.

A second session took place a week later as a physical event and involved a live demo of aMazeChallenge, during which the game was presented to the students, illustrating how to use the various features of the game, including the code editing workspace, personalisation options, executing code in training mode, and joining and playing in online challenges. Students were then given time to interact and familiarise themselves with the game UI and engage with various game features (e.g., read the tutorial, personalise their avatar, create code, load code samples, and test them in training mode). After letting students become acquainted with the game, the first online challenge was published and students were asked to join in. Students participated in a total of three online challenges with varying difficulty (easy, moderate, difficult). To further motivate students to participate in this event, multiple winners (such as the first to exit the maze, or the one to exit with the most points) were rewarded with coupons for the university's cafeteria. While the students interacted with the online challenges, we took screenshots during the live online gameplay (through the app), photographs in the classroom space (with participants' consent), and field notes based on observed behaviour. Participant observations took place in a real-life classroom setting placing emphasis on students' feelings and mood (such as frustration, boredom, joy, or excitement), level of involvement in the activity, performance in the game (through the shared leaderboard displayed on the projector screen), winners' responses after successfully exiting the maze, and other noteworthy facts. This data highlighted key issues pertinent to learning and engagement which constitute the focus of inquiry, and further contributed to our understanding of user experience, feelings, and actions during data collection and analysis [12, 25].

Following the experimental session, participants were asked to fill out a post-event questionnaire, focusing on their experience with aMazeChallenge in both training and online modes, their opinion about programming following this experience, their perception regarding the game's effectiveness, and the features they liked and disliked the most. Furthermore, the questionnaire included two open-ended questions, the first asking students to indicate any problems they faced, and the second inviting students to provide their recommendations for potential improvements. The background and post-event questionnaires included the name of the students in order to relate their responses to their academic performance. After the data was collected and linked to overall academic performance, responses were anonymised. Following a data cleaning process, out of 112 registered students, a total of 81 background and 68 post-event questionnaires (across both phases) were used for further data analysis.

4.3 Study results

Background questionnaire. The majority of participating students were male (79%). While this confirms that Computer Science constitutes a male-dominated field [61], the results appear promising as the female students increased from only 4 in 2021, to 12 in 2022. The overwhelming majority (94%) of participants use a smartphone, with the most popular being Android (51%) and iOS devices (44%). This was a barrier for our study since aMazeChallenge is available only for Android-enabled devices. To circumvent this, and in an effort to engage all students, we asked students with iOS devices to pair with one of their peers so they could participate as a team.

In terms of their prior experience in programming, approximately half of the students (52%) responded that they had previously attended a programming course, while the rest (48%) had no programming experience before joining the University. Amongst those with previous experience, the results show that C++ is by far the most popular language, followed by Python and Java. Furthermore, students were asked to self-assess their programming level. Data from both groups yielded similar results. The majority of students considered themselves to be Beginners (57%), followed by Intermediate (20%) and Confident (11%), while some students reported that they were not sure (12%). The latter is not surprising given that almost half of the students were just starting to familiarise with core programming concepts as part of their degree. When asked to report on their perception of programming, the vast majority of students (83%) reported they consider it an 'Interesting' or 'Very interesting' activity, with the remaining students (17%) were neutral. None of the participants reported finding programming either 'Boring' or 'Very boring'.

Regarding their favourite methods for learning programming, the most popular responses were ‘Learning and practicing on my own’ (89%), ‘Watching instructional videos’ (72%), and ‘Attending lectures’ (65%). Alternative learning methods, including ‘Playing educational games on the web’ or ‘Playing educational games on a mobile device’ were less frequent, but still relatively popular (37% and 31%, respectively). When asked whether they have previously used an app to learn a new skill, only 32% of students responded ‘Yes’, 53% responded ‘No’, and 15% were ‘Not sure’. When asked to rate how good a competitive environment is for learning programming, most students responded either ‘Definitely good’ or ‘Good’ (56%), many responded they are ‘Neutral’ (41%), and a few found it either ‘Bad’ or ‘Definitely bad’. In addition, participants were asked to indicate their opinion on how helpful games are for learning programming. As shown in figure 3, the majority of participants believe that games are either ‘Very helpful’ or ‘Helpful’ for learning programming (79%), while only a few believe that they are either ‘Not helpful’ or ‘Completely ineffective’ (5%). 16% of students indicated that they are ‘Not sure’.

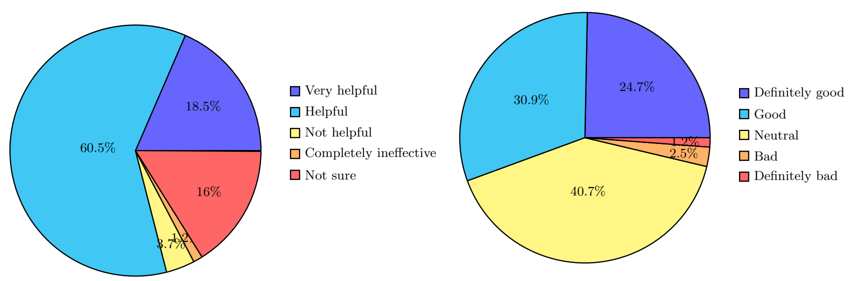


Fig. 3. Perceived usefulness of games (left) and competitive environments (right) toward learning programming (right).

Students were also asked to report which features of an educational game they consider as important. According to the responses, Performance is the most important feature in an educational game (72%), followed by User Interface (68%), Graphics (61%), and Multiplayer mode (58%). Other features, like Rewards and Battery-friendliness, were also recorded. These results are aligned with key considerations reported in recent literature, in relation to mobile learning [2,28] and mGBL [50]. The background questionnaire also included an open-ended question asking students to report the expectations from playing the aMazeChallenge mobile educational game. Students responses related to gaining experience in programming (58%), entertainment (18%), and improving critical thinking (10%), while 14% reported that they did not have any expectations.

Observation in the field. Observing students provided additional insights into their experiences with aMazeChallenge. Common observations across both phases included frustration, especially at the beginning when they were trying to figure out how to play the game, and a positive response to the competitions and the prizes for winners. Typical questions posed by students focused on whether their participation counts towards their marks for the module and whether it is compulsory to participate (neither of which was the case). Overall, students spent less time than expected on the learning and training mode, and were more eager to proceed to the live online competitions. Although overall the findings were similar across the two phases of the study, in the second phase students demonstrated more enthusiasm, their actions and reactions were more energetic, and they appeared to be more engaged during gameplay. These observations provided evidence that they also exerted themselves to try and win the challenges; they appeared more concentrated, they were asking more specific questions or requested hints more often compared to students in the first phase, indicating their eagerness to play ‘for the win’ and their willingness towards making a mental effort. A parallel observation is the fact that in the second group most students participated as teams rather than individually, which made competitive play more thrilling and fun. In addition to questionnaire data, these observations enriched our understanding of how students experienced learning programming through this mGBL activity.

Post-event questionnaire. After their experience with aMazeChallenge, students were asked to provide their responses to the post-event questionnaire. The questionnaire items were aligned to our research questions. Participants were first asked to rate their experience with aMazeChallenge indicating their satisfaction with the game. The majority of participants (64%) reported that their experience was either ‘Good’ or ‘Excellent’, 29% reported having a ‘Neutral’ experience, while 6% of the participants reported having a ‘Bad’ experience. Students were also asked to rate their experience in the single-player and multi-player modes of the game, which yielded analogous results. When asked to rate their perceived level of programming experience after engaging with aMazeChallenge, 41% of participants reported feeling ‘Confident’, 32% ranked themselves as ‘Intermediate’, and 12% as ‘Beginner’. Furthermore, the vast majority of students (79%) responded that programming is either ‘Very interesting’ or ‘Interesting’, with 19% responding with ‘Neutral’, and 1% finding this activity ‘Very boring’.

The post-event questionnaire also asked students to report how difficult they found aMazeChallenge to be. As shown in figure 4, 43% of participants found aMazeChallenge either ‘Very easy’ or ‘Easy’, and 40% found it either ‘Very difficult’ or ‘Difficult’. The remaining 18% reported that they were not sure. In addition, when asked about how helpful aMazeChallenge is when learning programming, most participants responded with ‘Neutral’ (49%), while 37% found it ‘Helpful’, and 15% found it either ‘Not helpful’ or ‘Completely ineffective’. None of the participants found aMazeChallenge ‘Extremely helpful’. Similarly, most participants (62%) indicated that their skills were unchanged after playing aMazeChallenge, 34% reported an improvement, and 5% reported a reduction.

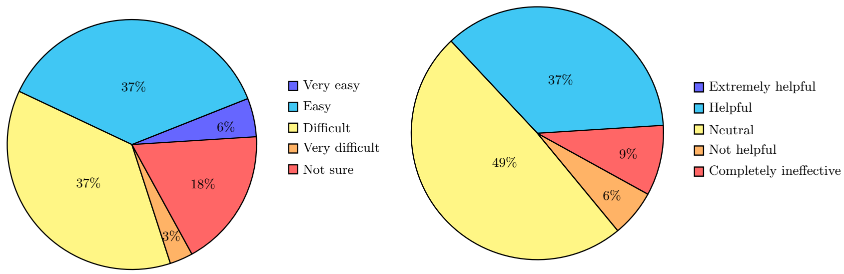


Fig. 4. Perceived difficulty when playing aMazeChallenge (left) and its helpfulness in terms of learning programming (right).

When asked to rate the usefulness of a competitive environment, 66% of students reported that such environments are conducive toward learning programming, 30% were neutral, and 5% said that they did not help the learning process. Asked whether they are likely to play aMazeChallenge again in the future, most participants (48%) responded with either 'Very likely' or 'Likely', 31% responded with 'Unlikely' and 'Very unlikely', whereas 22% were not sure. Finally, the most liked feature of aMazeChallenge was Personalization, followed by Code editing and Multiplayer mode. Paradoxically, the code editor was both a liked and disliked feature for different student groups. In total, 7 participants reported several issues involving app crashes, among other features that were disliked.

5 Discussion of the Findings

The analysis of questionnaire responses, observational data, and field notes were used to extract useful insights about user experience with aMazeChallenge, explore how and whether students' perceptions about programming were affected after their experience, as well as to evaluate the effectiveness and usability of aMazeChallenge (and to an extent of mGBL), in terms of engaging learners.

5.1 User experience with mGBL

The first objective of this study is to explore learners' experiences and perceptions on learning programming through a mGBL activity. Prior to interacting with aMazeChallenge, the majority of the students (56%) considered themselves as beginners in programming, while after the intervention only 11% of the students considered themselves beginners, and the proportion of students who felt confident or at an intermediate level increased to 74%. There was also a slight increase (+3%) in the number of participants who found programming to be an interesting activity after their mGBL experience. A controversial finding was that while most students' satisfaction was 'Good' or 'Excellent' and their self-perceived confidence levels increased, the majority of the students also reported that their skills were unchanged after playing aMazeChallenge and provided low rating in terms of how helpful it is. This may relate to the fact that students did not engage with the game outside the class time, which also brings forward the need to consider engagement from a wider socio-techno-pedagogical perspective.

Additionally, the results confirm the notion that most students consider programming to be a difficult task. Interestingly, a higher percentage of female participants (29%) consider programming to be easy, compared to male participants (21%). This also emerged during the observation of the live gameplay session, especially in the second phase, where female students appeared more intrigued and involved. In contrast with their responses about programming in general, a significantly lower percentage found aMazeChallenge to be difficult. Based on these results, it is evident that the learners' perspectives on learning programming were affected by introducing the game-based learning activity with aMazeChallenge. The vast majority of participants (79%) consider games to be helpful toward learning programming, which provides additional motivation for exploring GBL. The results indicate that students felt more confident in their abilities as programmers, which may suggest that aMazeChallenge, and to a certain extent mGBL, can serve as an alternative method to engage students with programming concepts in higher education.

5.2 Engagement effectiveness

The second objective of the study was to explore which aspects affect learner engagement with mGBL and how effective the game's features are in terms of engaging students. Even though more students found programming with aMazeChallenge easier than those who found it difficult, conversely, only 41% found aMazeChallenge to be helpful, and many students (46%) reported feeling neutral. The latter finding potentially uncovers limitations in terms of engagement effectiveness. Observational data also highlight the multiplicity of aspects that affect learner engagement in mGBL, many of which go beyond the inherent technical or design characteristics of the game. Aspects such as the overall user experience, intrinsic and extrinsic motivation, assessment methods, and feedback strategies highly impact learning experiences and learner engagement. Although further research is needed to explore this in the context of learning programming, the findings confirm that learning engagement is a complex construct that may be affected by a multitude of factors [45, 46]

The results of the final questionnaire also revealed that the most liked features of aMazeChallenge among the participants were personalisation, code editor, and multiplayer gameplay. However, the code editor appears to be both disliked and loved by different groups of participants. During observation it was evident that some students were frustrated with the drag-and-drop mode used in the Blockly workspace. Such languages may appear to be less direct, with more advanced students considering them as less powerful tools of expression. Notably, several participants also reported app crashes during the online gameplay. Such issues must be avoided altogether so that user experience and learner engagement are not negatively impacted.

Another aspect related to engagement has to do with collaborative and competitive game elements. Most students (56%) seem to believe that competitive game environments can be beneficial to the learning process. The responses of the students to the same question after playing aMazeChallenge reveal an increase (+14%) in the number of students who considered competitive environments advantageous toward the learning process. Therefore, incorporating such environments in interactive multiplayer games may intrigue and engage students to get involved in the learning process. Finally, the game also seems to have relatively high long-term engagement, as about half of the students said that they are likely to play aMazeChallenge again in the future. Other studies also support the claim that MMOGs can be employed as agents of motivation enhancing the learning experience due to the manifestation of collaborative and competitive gameplay, rendering them a promising game genre for university students [7].

5.3 Technological readiness

The results also provide insights into the technological effectiveness of aMazeChallenge and can help explore how effective mobile devices are as a learning platform specifically for learning programming and algorithmic thinking. Firstly, mobile operating system usage trends indicate that future mobile educational games may benefit by being developed as cross-platform apps that can be deployed on a variety of operating systems [40]. This may increase their uptake and effectiveness alike, as more students will be able to interact with the game on their own mobile devices. Secondly, technological features (namely, game performance, UI, and quality of graphics) prevail in what students consider as the most important features when playing an educational game. These results highlight the importance of human-factors, game design, usability and UX goals, as all of the top features selected by the participants relate to UX. Performance and UI design are widely regarded are the most critical factors affecting the user's experience in an online game [14]. Similarly, graphics and game mechanics play a significant role in player immersion and contribute to the player's overall experience in games [54]. These results are also supported by relevant literature [4, 34, 41, 50, 55] and therefore indicate the potential areas where developers may need to focus on in the future to create an enhanced user experience and more effective mobile educational games.

6 Conclusion

The outcomes of the evaluation illustrate that aMazeChallenge has the engagement efficacy and technological capability to induce university students into basic programming concepts, yet highlight the multiplicity of aspects that affect learning and learner engagement in game-based learning, many of which go beyond the inherent technical characteristics of the game. Students felt significantly more confident with their programming skills after playing aMazeChallenge; enjoyed the competitive gameplay; but also voiced their frustrations, and assessment-related concerns, and indicated aspects they considered significant, such as performance, user interface, graphics, online gameplay, and personalisation. The findings re-emphasise that aspects such as the overall user experience, intrinsic and extrinsic motivation, assessment methods, and feedback strategies employed, highly impact learning experiences and learner engagement, in addition to the design features of the game. The overarching theme that emerged by collectively addressing the research objectives is what we refer to as 'engagement by design', that is, engagement must be treated as the fusion of pedagogical design, technological design, and game design rather than an external student trait. This theme emphasises the genuine need to achieve a constructive alignment between the inherent technological qualities and design features of a mobile educational game, the broader pedagogical context, and human-centered factors, in order to achieve the desired learner engagement and learning outcomes. The study findings can inform the development of heuristic or theoretical frameworks that can guide the design, development, and evaluation of interactive education mobile games in higher education.

References

1. Adams, S.: The role of gamification in the facilitation of student engagement: an exploratory industrial psychology application. Ph.D. thesis, Stellenbosch: Stellenbosch University (2019)
2. Ahmad Faudzi, M., Che Cob, Z., Omar, R., Sharudin, S.A., Ghazali, M.: Investigating the user interface design frameworks of current mobile learning applications: A systematic review. Education Sciences **13**(1), 94 (2023)
3. Al Zieni, H.: The effect of mobile learning on learner autonomy: A suggested measurement tool to assess the development of learner autonomy. Journal of Asia TEFL **16**(3), 1020 (2019)
4. Alasmari, T.: The effect of screen size on students' cognitive load in mobile learning. Journal of Education, Teaching and Learning **5**(2), 280–295 (2020)
5. Ally, M.: Mobile learning: From research to practice to impact education. Learning and Teaching in Higher Education: Gulf Perspectives **10**(2), 3–12 (2013)
6. Barr, V., Stephenson, C.: Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? Acm Inroads **2**(1), 48–54 (2011)
7. Bawa, P., Watson, S.L., Watson, W.: Motivation is a game: Massively multiplayer online games as agents of motivation in higher education. Computers & Education **123**, 174–194 (2018)
8. Carrión12, M., Santorum12, M., Aguilar, J., Peréz, M.: iplus methodology for requirements elicitation for serious games (2019)
9. Chu, H.C.: Potential negative effects of mobile learning on students' learning achievement and cognitive load—a format assessment perspective. Journal of Educational Technology & Society **17**(1), 332–344 (2014)
10. CodeMonkey: Coding for kids - game-based programming. `https://www.codemonkey.com/` (2019), last accessed: 2019-12-19
11. Connolly, T.M., Boyle, E.A., MacArthur, E., Hainey, T., Boyle, J.M.: A systematic literature review of empirical evidence on computer games and serious games. Computers & education **59**(2), 661–686 (2012)

12. Creswell, J.W.: Research design: Qualitative, quantitative, and mixed methods approaches. Thousand Oaks: Sage (2003)
13. Curum, B., Khedo, K.K.: Cognitive load management in mobile learning systems: principles and theories. Journal of Computers in Education **8**(1), 109–136 (2021)
14. Einfeldt, L., Degbelo, A.: User interface factors of mobile ux: A study with an incident reporting application. arXiv preprint arXiv:2102.02510 (2021)
15. Elkhair, Z., Abdul Mutalib, A.: Mobile learning applications: characteristics, perspectives, and future trends. International Journal of Interactive Digital Media (IJIDM) **5**(1), 18–21 (2019)
16. Faisal, C.M.N., Fernandez-Lanvin, D., De Andrés, J., Gonzalez-Rodriguez, M.: Design quality in building behavioral intention through affective and cognitive involvement for e-learning on smartphones. Internet Research (2020)
17. Feng, Y., Liao, Y., Ren, Y.: Effects of m-learning on students' learning outcome: A meta-analysis. In: New Media for Educational Change, pp. 115–123. Springer (2018)
18. Ferguson, R., Coughlan, T., Egelandsdal, K., Gaved, M., Herodotou, C., Hillaire, G., Jones, D., Jowers, I., Kukulska-Hulme, A., McAndrew, P., et al.: Innovating pedagogy 2019: Open university innovation report 7 (2019)
19. Fotaris, P., Mastoras, T., Leinfellner, R., Rosunally, Y.: Climbing up the leaderboard: An empirical study of applying gamification techniques to a computer programming class. Electronic Journal of e-learning **14**(2), 94–110 (2016)
20. Giannakoulopoulos, A., Limniati, L., Konstantinou, N.: Examining the contemporary status of gamification in education. In: INTED2020 Proceedings. pp. 6467–6473. IATED (2020)
21. Gikas, J., Grant, M.M.: Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones & social media. The Internet and Higher Education **19**, 18–26 (2013)
22. Google: Blockly - google developers. `https://developers.google.com/blockly` (2019), last accessed: 2020-01-08
23. Gouws, L.A., Bradshaw, K., Wentworth, P.: Computational thinking in educational activities: an evaluation of the educational game light-bot. In: Proceedings of the 18th ACM conference on Innovation and technology in computer science education. pp. 10–15 (2013)
24. Hambrock, H., De Villiers, F., Rusman, E., MacCallum, K., Arrifin, S.: Seamless learning in higher education: Perspectives of international educators on its curriculum and implementation potential: Global research project 2020. International Association for Mobile Learning (2020)
25. Hammersley, M., Atkinson, P.: Ethnography: Principles in practice. New York: Routledge (2007)
26. Kanellopoulou, C., Giannakoulopoulos, A., et al.: Engage and conquer: An online empirical approach into whether intrinsic or extrinsic motivation leads to more enhanced students' engagement. Creative Education **11**(02), 143 (2020)
27. Kasenides, N., Paspallis, N.: amazechallenge: An interactive multiplayer game for learning to code (2021)
28. Kim, J., Choi, Y., Xia, M., Kim, J.: Mobile-friendly content design for moocs: Challenges, requirements, and design opportunities. In: CHI Conference on Human Factors in Computing Systems. pp. 1–16 (2022)
29. Krajcsi, A., Csapodi, C., Stettner, E.: Algotaurus: an educational computer programming game for beginners. Interactive Learning Environments **29**(4), 634–647 (2021)

30. Kroustalli, C., Xinogalos, S.: Studying the effects of teaching programming to lower secondary school students with a serious game: a case study with python and codecombat. Education and Information Technologies **26**(5), 6069–6095 (2021)
31. Lathwesen, C., Belova, N.: Escape rooms in stem teaching and learning—prospective field or declining trend? a literature review. Education Sciences **11**(6), 308 (2021)
32. Li, X., Heng, Q.: Design of mobile learning resources based on new blended learning: a case study of superstar learning app. In: 2021 IEEE 3rd International Conference on Computer Science and Educational Informatization (CSEI). pp. 333–338. IEEE (2021)
33. Liu, M.: Motivating students to learn using a game-based learning approach gaming and education issue. Texas Education Review **2**(1), 117–128 (2014)
34. Manzano-León, A., Aguilar-Parra, J.M., Rodríguez-Ferrer, J.M., Trigueros, R., Collado-Soler, R., Méndez-Aguado, C., García-Hernández, M.J., Molina-Alonso, L.: Online escape room during covid-19: A qualitative study of social education degree students' experiences. Education Sciences **11**(8), 426 (2021)
35. Marfisi-Schottman, I., George, S., Tarpin-Bernard, F.: Evaluating learning games during their conception. In: European Conference on Games Based Learning. vol. 1, p. 364. Academic Conferences International Limited (2014)
36. Marinoni, G., van't Land, H.: The impact of covid-19 on global higher education. International higher education (102), 7–9 (2020)
37. Meftah, C., Retbi, A., Bennani, S., Idrissi, M.K.: Evaluation of user experience in the context of mobile serious game. In: 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS). pp. 1–5. IEEE (2019)
38. Miljanovic, M.A., Bradbury, J.S.: A review of serious games for programming. In: Joint international conference on serious games. pp. 204–216. Springer (2018)
39. Moreno-León, J., Robles, G.: The europe code week (codeeu) initiative shaping the skills of future engineers. In: 2015 IEEE global engineering education conference (EDUCON). pp. 561–566. IEEE (2015)
40. Nawrocki, P., Wrona, K., Marczak, M., Sniezynski, B.: A comparison of native and cross-platform frameworks for mobile applications. Computer **54**(3), 18–27 (2021)
41. Ng'ambi, D., Brown, C., Bozalek, V., Gachago, D., Wood, D.: Technology enhanced teaching and learning in south african higher education–a rearview of a 20 year journey. British Journal of Educational Technology **47**(5), 843–858 (2016)
42. Papadakis, S.: Advances in mobile learning educational research (amler): Mobile learning as an educational reform. Advances in Mobile learning educational research **1**(1), 1–4 (2021)
43. Paspallis, N., Kasenides, N., Piki, A.: A software architecture for developing distributed games that teach coding and algorithmic thinking. In: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). pp. 101–110. IEEE (2022)
44. Pauli, M., Ferrell, G.: The future of assessment: five principles, five targets for 2025 (2020)
45. Piki, A.: Learner engagement in computer-supported collaborative learning activities: Natural or nurtured? In: International Conference on Learning and Collaboration Technologies. pp. 107–118. Springer (2014)
46. Piki, A.: Learner engagement in mobile computer-supported collaborative learning contexts: An integrative framework. In: Proceedings of the 16th World Conference on Mobile and Contextual Learning. pp. 1–7 (2017)

47. Piki, A.: An exploration of student experiences with social media and mobile technologies during emergency transition to remote education. In: World Conference on Mobile and Contextual Learning. pp. 10–17 (2020)
48. Piki, A.: Re-imagining the distributed nature of learner engagement in computer-supported collaborative learning contexts in the post-pandemic era. In: International Conference on Human-Computer Interaction. pp. 161–179. Springer (2022)
49. Piki, A., Markou, M., Vasiliou, A.: Learning through play: The role of learning and engagement theory in the development of educational games for intellectually challenged children. In: 2016 International Conference on Interactive Technologies and Games (ITAG). pp. 1–6. IEEE (2016)
50. Piki, A., Ştefan, I.A., Stefan, A., Gheorghe, A.F.: Mitigating the challenges of mobile games-based learning through gamified lesson paths. In: World Conference on Mobile and Contextual Learning. pp. 73–80 (2020)
51. Piñero Charlo, J.C.: Educational escape rooms as a tool for horizontal mathematization: Learning process evidence. Education Sciences **10**(9), 213 (2020)
52. Schell, J.: The Art of Game Design: A book of lenses. CRC press (2008)
53. Schindler, L.A., Burkholder, G.J., Morad, O.A., Marsh, C.: Computer-based technology and student engagement: a critical review of the literature. International journal of educational technology in higher education **14**(1), 1–28 (2017)
54. Shih, W.C.: Mining learners' behavioral sequential patterns in a blockly visual programming educational game. In: 2017 International Conference on Industrial Engineering, Management Science and Application (ICIMSA). pp. 1–2. IEEE (2017)
55. Sophonhiranrak, S.: Features, barriers, and influencing factors of mobile learning in higher education: A systematic review. Heliyon **7**(4), e06696 (2021)
56. Stefan, I.A., Gheorghe, A.F., Stefan, A., Piki, A., Tsalapata, H., Heidmann, O.: Constructing seamless learning through game-based learning experiences. International Journal of Mobile and Blended Learning (IJMBL) **14**(4), 1–12 (2022)
57. Suartama, I.K., Setyosari, P., Ulfa, S., et al.: Development of an instructional design model for mobile blended learning in higher education. International Journal of Emerging Technologies in Learning **14**(16) (2019)
58. Sung, Y.T., Yang, J.M., Lee, H.Y.: The effects of mobile-computer-supported collaborative learning: Meta-analysis and critical synthesis. Review of educational research **87**(4), 768–805 (2017)
59. Swacha, J.: State of research on gamification in education: A bibliometric survey. Education Sciences **11**(2), 69 (2021)
60. Tsekleves, E., Cosmas, J., Aggoun, A.: Benefits, barriers and guideline recommendations for the implementation of serious games in education for stakeholders and policymakers. British Journal of Educational Technology **47**(1), 164–183 (2016). https://doi.org/https://doi.org/10.1111/bjet.12223
61. Vitores, A., Gil-Juárez, A.: The trouble with 'women in computing': a critical examination of the deployment of research on the gender gap in computer science. Journal of Gender Studies **25**(6), 666–680 (2016)
62. Wilson, C.: Hour of code—a record year for computer science. ACM Inroads **6**(1), 22–22 (2015)
63. Wing, J.M.: Computational thinking. Communications of the ACM **49**(3), 33–35 (2006)