ARQuest: A Tangible Augmented Reality Approach to Developing Computational Thinking Skills

Anna Gardeli
Department of Product & Systems
Design Engineering
University of the Aegean
Syros, Greece
agardeli@aegean.gr

Spyros Vosinakis
Department of Product & Systems
Design Engineering
University of the Aegean
Syros, Greece
spyrosv@aegean.gr

***Abstract*—Computational thinking is considered an important skillset for 21st century learners and became a subject of focus in K-12 education in the recent years. It cultivates problem-solving and algorithmic thinking and can be helpful in wider aspects of everyday life, besides programming and computer science. There have been various approaches to teach computational thinking in early education stages that utilize modern technologies, such as physical computing and robotics, and adopt a playful manner. Mobile Augmented Reality is a promising technology along this path that lets users collaborate and interact with both physical and digital elements. However, the technology is still relatively new and has not been studied extensively with students, especially in formal educational settings. In this paper we present ARQuest, a collaborative mobile augmented reality game for developing computational thinking skills of primary school students. The game combines a physical board and tangible tokens with animated 3D content, and students use them to create and solve challenges in a gamified environment. We have performed an exploratory study in two primary schools in order to explore the impact of ARQuest on students' understanding of algorithmic concepts, engagement and collaboration. The study led to a number of interesting conclusions regarding the game, as well as more generic aspects of interacting and collaborating in mobile augmented reality.**

Keywords—computational thinking, mobile augmented reality, tangible augmented reality, game-based learning, collaborative learning

I. INTRODUCTION

Computational Thinking (CT) is the use of computer science concepts and practices to formulate problems and plan their solutions [1]. It is considered an important skillset for everyone, not just computer scientists, and is becoming a key element of STEM education. CT involves the analysis of complex problems into manageable sub-problems, the production of a sequence of clearly defined steps to solve a problem, the transfer of a solution to other similar cases, and the use of computers to automate a solution [2]. CT cultivates algorithmic thinking and problem solving abilities that are applicable not only to information technology, but also to a much wider range of real-world problems that may or may not involve computers. J. Wing claims that CT is as important as reading, writing and arithmetic for children's analytical abilities [3].

Various technological and instructional approaches have been introduced so far to support the development of CT skills in early educational stages, e.g. [4, 5]. Some notable works focus on simplifying the code and syntax and giving the opportunity to easily create appealing programs, such as the popular visual programming environment Scratch [6]. Students can develop simple programs by manipulating visual code-blocks instead of manually typing text commands, making programming easier and preventing syntactical errors. On the other hand, there are approaches that attempt to disassociate computational thinking from computers and engage students in problem-solving activities that take place in the real world, e.g. using pen and paper or everyday objects, also called 'unplugged' activities [7]. An interesting combination of those two ends is the use of tangible interfaces and physical computing. In this case the students create their code in the physical world through manipulation of tangible interactive objects and can observe the results of their solution, either in the real world [8] or on a screen [9].

At the same time, researchers have explored the application of playful design approaches to learning CT, to increase student motivation. Following the widely established tendency of game-based learning [10], a number of environments for early programming or STEM have been introduced, where students have to use code or command blocks to solve a game challenge, e.g. guide the robot to the exit of a maze [11], or even to create their own games by programming the game rules [12].

One promising path in the above-mentioned directions is the use of playful mobile augmented reality (AR) environments in combination with tangible parts [13]. In this case the interaction takes place in the physical as well as the digital environment, carrying some of the advantages of both worlds. Students can collaborate in an intuitive and playful manner in the physical world through manipulating physical objects, and at the same time be able to view a gamified digital version of their actions through the devices' screen. An additional benefit is that it is a low-cost approach and does not require any additional costly or specialized hardware to operate. As such it is easier to introduce in formal class settings. Last but not least, it is an open-ended solution, for which arbitrary gamified environments and challenges of various difficulties may be developed. However, being a relatively new technology, there have been only a few user studies about efficient ways of interacting and collaborating with mobile AR environments in the classroom, usability issues that might emerge, and the effect of the medium on developing CT skills. Our research attempts to contribute towards this end.

In this paper we present a collaborative mobile AR environment for developing CT skills of primary school students following a game-based learning approach. We introduce the design and development of ARQuest, a game played in teams using a physical board, tangible tokens and mobile devices, which engages students in designing and solving challenges in a digital maze-like game environment using action commands. Compared to related AR approaches [13, 14], ARQuest combines a number of features in a novel

978-1-7281-4540-2/19/$31.00 ©2019 IEEE

way: the synchronization and sharing of the same environment among multiple devices, the possibility of creating custom mazes and challenges using tangible tokens, and the real-time feedback on command execution whilst the solution is running.

We have performed an exploratory study of using ARQuest in the classroom, aiming to identify critical issues regarding its use and its effectiveness, as well as to shed more light in more generic usability and collaboration aspects of tangible mobile AR approaches for CT. The study has been carried out with 26 students aged 9-10 years in two primary schools, and we have measured aspects such as enjoyment, collaboration, usability and effectiveness using a mixture of measures. The results revealed interesting findings regarding student motivation, emerging collaboration patterns, the effect of mobile devices' number and screen size in performance, and the game in general.

II. Related Work

A. Computer Science Education and Computational Thinking

Today, there is an apparent shift towards teaching computational thinking (CT) in the fields of STEM and computer science (CS) education. Dijkstra [15] argued that "it becomes clear what 'programming' really amounts to, viz. designing algorithmic solutions, and that activity requires the ability to think efficiently more than anything else", and after three decades, Wing [3] drew the attention to that term and define it as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science". CT is a notion that has taken different dimensions over the years and the skills encompassing CT vary in the literature, though critical thinking and problem solving have been reported as the most commonly related skills to CT [16]. Later, ISTE[1], described CT to encompass the skills of creativity, algorithmic thinking, critical thinking, problem solving, establishing communication and establishing cooperation.

The learning benefits of CT extend beyond the fields of CS and STEM, as it refers to the development of higher-level skills, instead of focusing on the syntax and semantics of computer programming. Several studies argue that teaching computational concepts in the school curriculum provides a range of learning outcomes which can be applied to areas other than computer science [17, 18]. Popat et al. [19] presented an overview of the perceived learning outcomes of learning computational thinking in school, alongside with programming, categorized into five (5) themes, (a) critical thinking, (b) problem-solving, (c) social skills, (d) self-management and (e) academic skills or knowledge. Therefore, educational systems around the world are shifting towards including computational thinking as part of a mandatory or not school curriculum, such as England (2013)[2], which become possible by the availability of programming tools appropriate for younger students.

Nowadays there is a research trend towards more natural approaches on semantics or languages that enhance the computational techniques and the understanding of computation in generally [20, 21], as it is widely accepted that the complex syntax of traditional programming languages is a main obstacle for novices. Towards that, a question is raised; whether traditional programming environments consist the most efficient medium in order to teach computational thinking to novices, and even whether computers are required [22, 7].

Nonetheless, as being noted in several reviews, there is an apparent gap in evaluating educational systems for computational thinking for K-12 students as most studies are conducted as after-school activities [5]. This research field requires further exploration through classroom-based interventions.

B. Game-based Learning for Introductory Computer Science

A number of studies used game-based learning (GBL) environments to support the education of CS, as it has become a prevalent instructional approach in both formal and informal educational settings. GBL provides education with opportunities of intuitive and engaging learning, since it encompasses skills like analyzing, synthesizing, evaluating, and performing higher-order thinking skills [23]. This is due to the attributes and affordances that are built into games.

Zhang et. al [24] present four main types of implementing games as educational tools in computer science (CS) education: (a) using games to motivate students, (b) making games to teach CS topics, (c) using games as environments to teach CS topics, and (d) using games as examples to teach CS topics. Two of those types of tendencies in teaching CT and introductory programming using games in learning; teaching through game design and teaching through gameplay. Towards the former, the most widely known, Scratch [18] and Alice [12], are visual programming tools that allows children to create interactive stories, games, and animations. However, building a game from scratch can be challenging for novices, and especially younger students, as it requires a spectrum of knowledge beyond programming [25, 26]. In addition, students are not able to complete an interesting project on their own from the beginning and it takes relatively long time to master. Therefore, as being noted by Kazimoglu [10], this approach addresses a more instructivist style of learning as an expert instructor is required. On the contrary, the game play approach supports a more constructivist approach as the goal of the game is matching the learning objective defined by specific game rules and knowledge acquisition emanates from trial and error, exploration, increasing difficulty etc. The users learn computational concepts such as sequence algorithms, loops, functions and more by programming a character towards his goal through a visual set of commands, such as in Light-Bot [27] and in Program Your Robot [28].

C. Mobile Augmented Reality and Tangible Approaches

Until recently, there have been two main approaches on teaching CT and introductory programming in K-12 education, as mentioned in introduction; block-based digital environments, such as Alice [12], Scratch [18] for game design and unplugged activities [7]. One of the main benefits of unplugged activities is that they are lowering the access barriers to the required equipment and infrastructures of

[1] ISTE. (2015). CT leadership toolkit. Available at: http://www.iste.org/docs/ctdocuments/ct-leadershipt-toolkit.pdf

[2] Department for Education, National curriculum in England: Computing programmes of study, Tech. rep., Department for Education, London. URL https:// www.gov.uk/government/publications/national-curriculum-in-englandcomputing-programmes-of-study.

educational tools [28] which results in easier implementation in field settings, consequently more concrete evidence concerning their use and effectiveness.

Towards that direction, a third emergent approach is activities that combine the physical and the digital world, where students create programs by manipulating physical objects. This approach is being implemented in two ways, either using AR technology or with microelectronics, such as Arduino. The main element in physical programming is the use of a tangible user interface to construct algorithms, while debugging can take place in physical (e.g. through robots) or digital space (screen display or through AR camera). The tangible interface is perceived more convenient for younger students, who prefer to hold toys in their hands to operate [29].

AR shows evidence for effectively supporting educational activities in a dynamic and interactive way in many disciplines, including computer science education. Studies present positive results on students' academic performance, motivation and engagement, while improving the building of their own knowledge. Billinghurst and Duenser [30] argue that AR interfaces are significantly different from other computer interfaces, as they enhance the real-world experience and draw users away from the screen. They provide a more intuitive way of interaction, resulting in higher levels of usability, learnability and engagement.

Regarding computer science education, AR appears to have a great potential concerning the difficulties novices face due to three-dimensional visualization, the virtual-real and real-time interaction [31]. There is the case of AR Scratch [32]; the Scratch environment is being enhanced with AR features enabling children to create their own marker-based AR application, although neither AR technology nor tangible interaction is actively involved in the learning process. On the other hand, tangible AR systems supporting learning through game play are more prevalent. They use physical blocks that should be placed in sequence to create programs which can be visualized and executed through the camera of a device [33, 14]. Most of them aim to teach algorithmic sequences, although some aim to teach different concepts of CS, such as networks [14]. Only one study, to the best of our knowledge, suggests a tangible AR system that provides real-time feedback during the debugging of the program [13], although it is visualized on top of the tangible tokens resulting in covering a lot of screen space and the execution of the program is being diminished.

Notwithstanding the potential of AR technology is being noted and supported by positive evidence concerning the motivation and engagement of students, there are still limited user studies found in bibliography about AR's educational value in class settings.

D. *The Scope of our Work*

Our proposed system shares the objective of the aforementioned studies in teaching basic CT concepts through tangible interaction with an AR enhanced environment. Nevertheless, ARQuest combines both attributes of teaching through game design and through game play in class setting. In addition, it provides real-time feedback during execution and collaborative settings through multiple devices. In contrast to other related works that are limiting their focus to CT concepts [5], ARQuest addresses some computational practices as well. Finally, our study aims to contribute to the evident gap in bibliography regarding evaluating educational AR systems in classroom-based interventions.

III. Design and Implementation of the Game

The ARQuest game has been designed to offer an introductory problem-solving and programming experience to students between 9 and 10 years old in a motivating and enjoyable manner. Its aims to help them cultivate their computational thinking skills through a gamified activity, where students collaborate in teams, create challenges and try to solve them through action sequences. Fig. 1 presents an overview of the game.

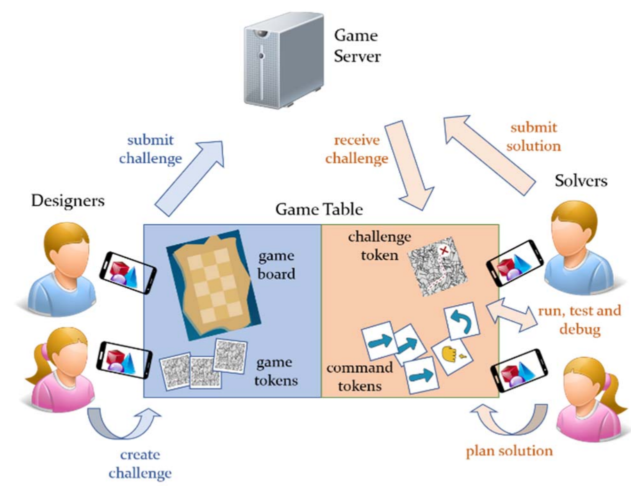

Fig. 1. An overview of the ARQuest game, its elements and actions.

A. *Game Overview*

ARQuest supports multiple 'game tables' on which independent games can be played simultaneously. Each game involves two competing teams:

- the *designers*, whose goal is to create a challenge for their opponents, and
- the *solvers*, who try to overcome the challenge

Accordingly, the table is arranged in two opposing sides: the designer side and the solver side, where the respective teams are collaborating. Each player regardless of the team can use a mobile device to view the digitally augmented parts of the game.

The designer side includes a *physical board* and a set of *game tokens*. The board shows a grid that represents the game stage, and the designers have to select and place a number of game tokens on it to create the challenge. Each token represents a game element, which can be seen through the mobile device in AR. There are three different types of game elements; elements that should be picked up, elements that should be avoided and obstacles that could be overcome by using the first type of elements. When the designers agree on the challenge, they can submit it to the solvers.

On the solvers' side there is a single *challenge token*, and multiple *command tokens*. The players may look at the whole game stage and its elements through their mobile devices if their camera points at the challenge token. They can select the appropriate commands to plan their solution and arrange them on the table. Each command is directly visible on the token, so they do not have to look at them through their mobile devices. The use of commands aims to guide the hero towards his goal, by using the right sequence of actions. There are four types of actions; move, turn (left or right), pickup and throw. When they are ready, they can scan their solutions, by having the phone's camera identify the tokens. All solvers' devices are then automatically updated with the latest solution.

Finally, each of the solvers may execute his/her solution to see if it is successful or not. The code sequence starts executing on the digital board and they can see the evolution of their plan, whilst getting visual feedback about their commands. Commands that have been successfully executed are displayed in green, the currently executed command is displayed in glowing white, and if a command fails it is displayed in red and the execution stops. Finally, the solvers get feedback about the success of their plan.

The game rules for an ARQuest game are the following:

1. The designers create a challenge that must be solvable using no more than twelve commands.
2. The solvers try to plan an effective solution for the challenge. They may test and revise their solution up to three times.
3. If the solvers succeed, they ask the designers if they can solve it using fewer commands. If not, the solvers win.
4. Otherwise, the designers have to solve their own challenge, to prove that it is indeed solvable or to do so in a smaller number of steps. If they succeed, they win.

B. *Design Rationale*

The main design goals of ARQuest were: a) to be entertaining as a game, b) to afford student collaboration, and c) to support the development CT skills

Regarding the entertaining features of the game, we have adopted the Mechanics-Dynamics-Aesthetics (MDA) approach [34] to decide about its main design directions. We have selected the following aesthetics of MDA that we consider suitable for the specific activity, technology and target group: *Challenge*, *Fantasy*, *Fellowship*, *Expression* and *Submission*. Challenge is inherent in the game, as students have to be engaged in problem solving activities to win. The Fantasy aspect is supported through the storyline and the animated 3D objects of the game, 'dressing up' the logical challenges as a pirate adventure. Fellowship is also emerging during the game because of the need for the team members to work together to effectively design or solve a problem. The aspect of Expression is supported through the teams' freedom to design their own game challenge and arrange the stage as they wish. Finally, aiming at Submission, we decided to place more emphasis in tangible interactions instead of using a digital interface, because children of this age are familiar with board games and manipulatives, and may find this modality more natural and fun.

To satisfy the goal of affording student collaboration, we took a number of design decisions regarding technology and game rules. First, we decided to make the game multi-user, so that each student can have his/her own view of the same digital environment through the mobile phone. Second, we created the physical board and tokens in large enough sizes to be easily accessible to small student teams, so they can effectively work together. And finally, the rules and interface of the game encourage students to actively cooperate to solve a challenge, as they need to look at the digital board to understand the problem and simultaneously construct their own solution using the markers.

With respect to the educational affordances of the game, we took into account the key dimensions of CT as proposed by Brennan and Resnick [6] and targeted at some *Concepts* as well as *Practices* suitable for the learners' age and background. Regarding Concepts, the game introduces students to sequential programming and to variables: they have to produce a sequence of commands to solve the challenge, and the hero is able to 'pickup' or 'throw' an element in his single free hand (the other one holds a sword!) demonstrating the idea of assigning a value to a variable and using it when needed. Regarding Practices, students learn about optimization and debugging. They have to optimize their solution in order to win the opponent team, as they need not to simply plan a sequence of commands, but to plan it effectively, in the minimum number of steps. Similarly, designers have to make the challenge difficult, but also make sure that it is solvable to have a chance to win the game. Finally, students can see the execution of their solution and get real-time visual feedback about the success of each command. This is expected to help them mentally connect their symbolic solution expressed in commands with the actual game animation, and more importantly introduce them to the concept of debugging.

C. *Story and Challenges*

The story in the current prototype of ARQuest is simple: a young pirate has to explore the island, find and open a treasure box, avoiding any possible obstacles that may stand in his path.

The island environment is represented as a 3 x 5 grid and there are 12 game objects that may be optionally placed on the board. Each grid cell may have exactly one game object, with the exception of the pirate. The latter may in the course of his actions find himself in a grid already occupied with another object, if it is not considered an obstacle. Each game object may occupy only a single grid cell and may have one of four possible orientations (facing north, east, south or west). In the current prototype, only the orientation of the pirate is important for solving the challenge, whilst for all the other objects the orientation only affects their visual representation in the scene.

The game objects are: the pirate, 5 trees, a key, a bucket with water, a plank of wood, a fire, a lake and the treasure box. Three of these objects can be picked up by the pirate: the key, the bucket and the plank. The trees, the lake and the fire are obstacles and the pirate cannot move to their position. However, the fire can be eliminated if the bucket of water is thrown into it, and the lake can be turned into a passable tile if the wooden plank is thrown to it. Thus, the pirate has the chance to overcome these obstacles if he picks up and uses the appropriate objects.

The available commands are: *forward*, *turn left*, *turn right*, *pickup* and *throw*. Forward moves the pirate to the next cell in the direction he is facing, unless it is outside the board or there is an obstacle there. Turn left and right changes his direction respectively. Pick up allows the pirate to take and hold the element found in his current cell. As mentioned before, the pirate can hold only one item each time. Finally, the throw command makes the pirate throw the item he is holding to the next cell forward.

Fig. 2. Various stages of the game: a. the physical board with the edge markers, b. a challenge created by the designers on the board, c. the same challenge as seen by the solvers, d. a sequence of actions sumbitted as a solution to the challenge, e. the solution is executed on the digital board and the commands (displayed as icons) are visualized based on their status.

The challenge terminates successfully if at the end of the command execution the pirate finds himself in the position of the treasure box holding the key. Figure 2 displays images and screenshots of various stages of a single game.

D. Implementation

ARQuest has been implemented in Unity using Vuforia engine. It is a multiuser application following a typical client-server approach, where each app running in a mobile device exchanges messages with a single server using TCP/IP connections. The game server supports multiple simultaneous tables, and for each table it stores the latest challenge (the game stage and the elements arranged on it) and the solution. Whenever a new challenge or a new solution is submitted, all the connected devices that join the same table are updated accordingly. The mobile phones run an Android app, which can operate in two modes: designer or solver.

In the designer mode, the app can detect the game tokens as AR markers and displays the associated 3D game objects (Fig 2b). When a user decides to submit the challenge, it identifies the markers' position and orientation on the grid. To do so, the app takes advantage of four special markers placed at the four edges of the board (edge markers), see Fig. 2a. If at least one edge marker is detected by the camera, the app: a) transforms the positions and orientations of all game tokens to the egde marker's local coordinate space, b) it calculates their horizontal and vertical distance from the marker in grid cells using their transformed position c) calculates their final grid position, based on the known position of the edge marker with respect to the grid, and d) it calculates their orientation as north, west, south or east, based on the edge marker's orientation. Finally, what is sent to the server is a list of all game objects that participate in the board with their unique id, row and column on the grid, and orientation.

In the solver mode, the device can detect the challenge token and the command tokens as AR markers. It reconstructs the whole board and its elements based on incoming server messages, and when the challenge token is detected, the board is displayed on screen (Fig. 2c). If a user wishes to send a solution, the app examines the position of all identified command tokens and sorts them based on their detected position from top to bottom and from left to right (Fig. 2d). The sequence of commands is sent to the server. Finally, when a solution is executed, the digital board attached to the challenge token begins animating and the respective commands are highlighted (Fig. 2e).

IV. Exploratory Study

Our research motivations are the following: (a) what are the factors and the characteristics of a tangible AR game that affect the engagement and involvement of students in an educational activity, (b) which collaboration patterns emerge within a team in class and how do they affect involvement, engagement and performance, and (c) what are the perceived difficulties and obstacles arising in integrating tangible AR-based systems in typical education settings, in terms of technological equipment and collaborative settings. We aim to explore how students respond to playing with it and manage to collaborate within the given settings, in order to achieve a better understanding of the problem.

A. Materials and Settings

The evaluation process took place in the context of typical education, in class settings. The study was held during regular school hours in two elementary schools in Syros, Greece, as a special activity, in collaboration with the educator, who attended the process.

It was facilitated at the computer lab of each school, where we set three large tables as workstations. Each table represented one game, so it accommodated two groups of students opposed to each other. For each game, a physical board, two different set of tokens - one for the designers and one for the solvers - and at least two portable devices were used. Concerning the technological equipment, we used two different settings; *screen-sharing* setting and *individual-screen* setting. In the former, students' groups shared one tablet device within the team. In the latter, each member individually controlled their one smartphone device using a selfie stick, in order to support one-hand holding.

B. Participants

The participants were 26 students in total (13 from each school) almost gender balanced (11 girls and 15 boys) aged 9 to 10 years old. As expected, most of them were familiar with mobile devices, such as smartphones and tablets. Most students were novices in computer science concepts, with some basic prior experience with block-based programming tools. They had no prior experience with mobile AR technology.

C. Measures

The data collection methods adopted during the study aim to collect mainly qualitative data. The goal of our study is to evaluate our approach, in terms of four factors (a) collaboration, (b) enjoyment, (c) interactivity and (d) comprehensibility. Our study included think-aloud feedback, observation of participants' interactions, performance recording, and anonymous questionnaires. More specifically, we observed and recorded the perceived collaboration patterns, such as turn-taking, talking and non-talking interactions, discussion, task management inside the groups etc. In addition, both problem designs and solutions of each game match were recorded by utilizing server's logging system, in order to inspect and analyze students' mistakes and performance.

Regarding the questionnaire, responses were given in terms of (Q1) enjoyment, (Q2) collaboration, (Q3) interactivity, and (Q4) learnability, using a Likert scale from 1 (strongly disagree) to 5 (strongly agree), adapted for children, i.e. using simple language and graphic representations (smileys) instead of numbers. The questionnaire presented the following statements: (Q1) enjoyed playing the game, (Q2) enjoyed playing the game with my classmates, (Q3) I enjoyed playing, using smartphone or tablet and paper blocks, and (Q4) I learned something new from the game.

D. Procedure

The study was conducted in two different sessions, one for each school. In each session participants were clustered in five (5) groups of two (2) students and one (1) group of three (3). The groups were paired randomly. In each session, we applied the screen-sharing setting on 4 groups and the individual-screen setting on two groups.

The first session lasted 45 minutes in total: 10 minutes for explanation and demonstration of the game, the goal, the commands and the execution process, 25 minutes for the actual game, and 10 minutes for feedback through a questionnaire. The second lasted 90 minutes in total, so we had 60 minutes for the actual game and 20 minutes for feedback.

Fig. 3. Photos of students playing with ARQuest during the study.

In the first session three (3) game matches were played, due to time limitations. In the second session six (6) game matches were played, where each paired group played two (2) game matches. For the first game match, the one group was asked to design a problem on the board and the other to solve that problem. In the second game match the groups switched roles. At the end of both sessions all the participants were asked to fill in a questionnaire about their experience. Finally, we held an open discussion with the participants and their teacher, in order to gain more insights about their experience. Figure 3 shows photos of students playing the game during the second session.

V. Results

The two sessions were generally successful, as most of the students thought ARQuest was a very fun game and we did not experience any technical problem or other critical issue.

Our results suggest high level of engagement and enjoyment. Some students asked where they can find the game, so they can play again. Questionnaires confirm the high level of enjoyment (Table I), as only 1.9 % of the answers were negative or neutral, while 89.4 % of the answers were five out of five in Likert scale. Most students pointed out the designing of the challenge, as their favorite part. Through the think aloud and observation method it was obvious that students were intrigued by the "blind" tokens, i.e. they could identify the game elements only through the mobile device. That become more apparent after the design groups submitted their challenge to the solvers and they usually continued playing with the game tokens on the board by changing their position trying out new challenges and watch them through the screen. In addition, they thought it was more challenging than solving the problem, as they had to calculate the steps in their mind. One student said, "the solution seams easy, I am more worried about the design", although he was eager to try.

TABLE I. QUESTIONNAIRE RESULTS

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Q1	0	0	0	2	24
Q2	0	1	0	3	22
Q3	0	0	0	3	23
Q4	0	0	1	1	24

Regarding the comprehensibility and usability of the system, two minor issues arose in relation to familiarity with the technology and semantics. Students needed more time to familiarize with the system, as it was their first encounter with AR technology. It became apparent especially in our first session, which lasted shorter. Fortunately, they all could master it shortly after some practice. A few students were confused due to the semantics used in our system. They could not completely conceive the concept of rotating as a single action. Our hypothesis is that this was due to prior experience with similar environments, where "turn" was conceptualized into a relatively different action, e.g. change direction and move towards that new direction. In addition, some students commented on the difference between pick and throw. As mentioned in Story and Challenges section pick up allows the character to hold the element found in his current cell, while the throw command makes the character throw the item he is holding to the next cell forward.

Through our server logs we were able to record the performance of the students in the activity. Table II shows the following data for each game played in the study: objects used, minimum steps to solve the challenge, number of steps of the solution, number of tries and total time take to solve. As seen in the table, most groups manage to achieve their goal. Only two groups exceed the defined maximum number of steps, resulting in a non - solvable solution in given settings. One group exceeded the defined maximum number of steps by one, which was supported by the system but against the rules of the game. Two groups exceed the permissible number of test and revise of their solution; one group by one and one group by four. The average time for solving the challenge was 10.2 minutes, while the maximum time needed was 14 minutes and the minimum was 6 minutes. The feature of real-time visual feedback during execution was proven to be very helpful, as expected. Students were able to understand the reason their solution failed and identify the exact step where things went wrong. By identifying the error, they were able to update their solution fast by changing one or two commands.

TABLE II. SERVER LOGS OF THE GAMES PLAYED IN THE STUDY

#	Objects used	Min. steps	Solution	Tries	Time
1	9	12	12	3	9 min
2	11	**20**	-	1	12 min
3	9	7	7	2	12 min
4	7	12	12	2	10 min
5	9	**16**	-	0	10 min
6	9	9	9	**4**	7 min
7	8	12	12	**7**	14 min
8	8	**13**	**13**	3	12 min
9	10	11	11	3	6 min

One more interesting finding is that even though the groups in each game were supposedly opponents, most groups – if not all – worked as a team. It became evident that the game rules created an environment where collaboration towards a common goal is prevailing in relation to competitiveness, as designers were aiming primarily towards a solvable challenge and secondly towards winning the solvers. Therefore, both designers and solvers wanted the pirate to win the treasure and act accordingly. One girl of the design group, said with enthusiasm "We won!", when the solvers' group manages to achieve their goal. However, the recorded challenges indicate that they did not try to make it easy in order to win, in contrast they created demanding problems, as possible in given settings, aiming to challenge each other.

Concerning collaboration patterns and involvement, students can take two roles in the activity: (a) the role of the *observer*; the one handling the mobile device and overlooking the augmented physical space and (b) the *operator*; the one manipulating the tokens in the physical space. The collaboration patterns observed refer to role-management within the groups, where students either hold, switch or share one or more roles, as depicted in Fig. 4. Role management depends on the following three factors: device handling, tokens manipulation and decision making. In the case of screen – sharing setting, where one tablet device was provided, we observed the following patterns: (C1) one member was the observer and led the decision-making for the whole group, (C2) one member was the observer but the decision-making was held by the whole group, and (C3) each member of the group took turn either as observer or operator. In the last case, the decision-making was held either by the observer (C3-1) or by the whole group (C3-2), referring back to cases (C1) and (C2). In the case of , where two smartphone devices were provided, the following patterns: (C4) the whole group facilitated both roles of observer and operator at the same time by making use of both devices and the decision-making was held by the whole group, and (C5) the group decided to utilize only one of the given devices leading to similar patterns as described in the sharing-screen setting (C1).

Our results suggest that, in terms of involvement and interactivity, our system provides positive indications, as all the patterns described above required at least a minimum level of cognitive capacity and active participation of all students. However, in C1 (as in C2-1) only one member is responsible for the decision making. Therefore, for the other members of the team it is uncertain if they benefit of the learning aspect of the game. Our results provide evidence that an individual – screen setting in collaborative AR-based approach has the potential to better support students' active involvement in all parts of the learning process (C4). It is significant in collaborative learning settings to reaffirm that all parties are actually learning. However, an interesting observation is that in one occasion, the group decide to use only one device, although two were provided, for the challenge design (C5), as one of the two members manipulated the physical tokens (C1).

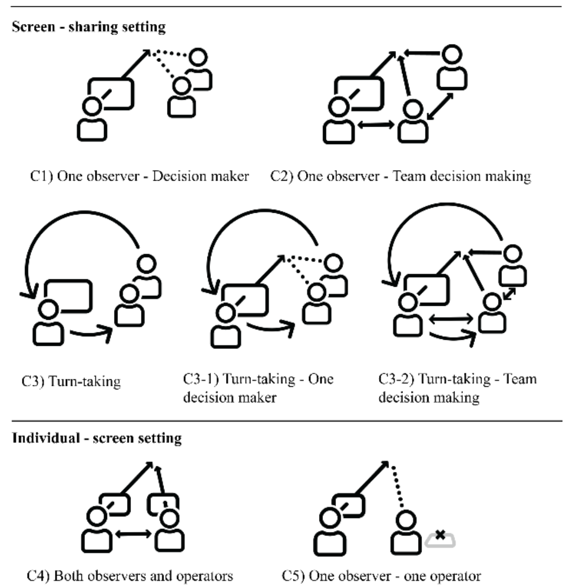

Fig. 4. Collaboration patterns that emerged during the game.

Finally, regarding the perceived difficulties and obstacles arising in integrating tangible AR-based systems in typical education settings, we noticed that the table surface, where the markers were placed, was relatively high for the students to achieve the required height and angle to scan all the markers simultaneously. There was a difference between the proper angle for students to look at the screen and the required angle for all the markers to be scanned. This problem occurred solely during scanning time.

VI. CONCLUSIONS

We have presented the design and implementation of ARQuest, a game for teaching CT to primary school students. It is played in a tangible marker-based AR environment for multiple users and combines both physical and digital elements in a shared gaming world. ARQuest has been designed to be both entertaining and effective as a learning environment. It includes gaming features such as creativity, competition, collaboration and fantasy, whilst it introduces students to sequential programming, variables, code optimization and debugging. This combination of novel affordances and metaphors in ARQuest unavoidably generates a number of challenges regarding the collaboration and interactions taking place in both the physical and digital world.

Our exploratory study attempted to shed more light to these challenges. We have tested ARQuest in typical educational settings in order to discover critical issues and observations regarding its use and to assess its effect on student motivation and learning. The results of our study indicate that mobile AR combined with tangible activities is indeed a promising path for early STEM education, as students were really engaged and motivated during the activity. Furthermore, the distribution and screen size of the mobile devices seemed to affect student collaboration, and a number of interesting patterns have emerged that need to be further studied.

The game style, game rules and types of commands that can be included in a tangible mobile AR game for introductory programming and problem-solving is still an almost uncharted territory. Our approach has been based on simple game rules with a relatively small number of commands. In the future we plan to examine the scalability of this first attempt towards richer and more advanced environments. We intend to explore designs that add more room for student creativity, such as the ability to draw and include their own game elements, as well as to support a wider range of concepts and practices related to CT that can be addressed to later stages of K-12 education.

ACKNOWLEDGMENTS

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE – INNOVATE (project code:T1EDK- 01571)

REFERENCES

[1] V.J. Shute, C. Sun, and J. Asbell-Clarke, “Demystifying computational thinking”, Educational Research Review, vol. 22, pp. 142-158, 2017.

[2] F. Kalelioglu, Y. Gülbahar, and V. Kukul, “A framework for computational thinking based on a systematic research review”, Baltic Journal of Modern Computing, vol. 4(3), pp. 583, 2016.

[3] J. Wing, “Computational thinking”, Commun. ACM, vol. 49(3), p. 33–35, 2006.

[4] A. Yadav, H. Hong, and C. Stephenson, “Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms”, TechTrends, vol. 60(6), pp. 565-568, 2016.

[5] S. Y. Lye and J.H.L. Koh, “Review on teaching and learning of computational thinking through programming: What is next for K-12?”, in Computers in Human Behavior, 41, p.51-61, 2014.

[6] K. Brennan and M. Resnick. "New frameworks for studying and assessing the development of computational thinking.", Proc. of the American Educational Research Association, Vol. 1, 2012.

[7] T. Bell, J. Alexander, I. Freeman, M. Grimley, “Computer science unplugged: school students doing real computing without computers”, N.Z. J. Appl. Comput. Inf. Technol. ,vol. 13 (1), pp. 20–29, 2009.

[8] P. Wyeth, and G. Wyeth, “Electronic Blocks: Tangible Programming Elements for Preschoolers”, Proc. of Interact, pp. 496-503, 2001.

[9] M. Horn, “Tangible Programming in the class with Tern”, ACM CHI Trends Interactivity, 2007.

[10] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, “Learning Programming at the Computational Thinking Level via Digital Game-Play,” Procedia - Procedia Comput. Sci., vol. 9, pp. 522–531, 2012.

[11] K. Sung, C. Hillyard, R. L. Angotti, M. W. Panitz, D. S. Goldstein and J. Nordlinger, “Game-themed programming assignment modules: A pathway for gradual integration of gaming context into existing introductory programming courses”, in IEEE Transactions on Education, vol. 54(3), pp. 416-427, 2011.

[12] S. Cooper, W. Dann, R. Pausch, “Alice: a 3-D tool for introductory programming concepts”, J. Comput. Sci. Coll. Consort. Comput. Sci. Coll., vol. 15, pp. 107–116, 2000.

[13] Q. Jin, D. Wang, X. Deng, N. Zheng, and S. Chiu, “AR-Maze: a tangible programming tool for children based on AR technology”, Proc. of the 17th ACM Conference on Interaction Design and Children (pp. 611-616), 2018.

[14] D. Bouchard and S. Daniels, “Tiles that Talk: Tangible Templates for Networked Objects”, in Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15). ACM, 2015, p.197-200.

[15] E.W. Dijkstra, “On the teaching of programming, i.e. on the teaching of thinking”, in: Language Hierarchies and Interfaces, Springer 1976, p. 1–10.

[16] A. A. Kranov, R. Bryant, G. Orr, S. A. Wallace, and M. Zhang, “Developing a Community Definition and Teaching Modules for Computational Thinking: Accomplishments and Challenges”, 2010

[17] F. Kalelioğlu, "A new way of teaching programming skills to K-12 students: Code. org." Computers in Human Behavior 52, 2015, p. 200-210.

[18] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al., “Scratch: programming for all”, Commun. ACM 52 (11), p. 60–67, 2009.

[19] S. Popat, & L. Starkey, “Learning to code or coding to learn? A systematic review”, Computers & Education, 128, 2019, p. 365-376.

[20] C. Kelleher, R. Pausch, “Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers”, ACM Comput. Surv. 37 (2), p. 83–137, 2005.

[21] J. Good, and K. Howland, “Programming language, natural language? Supporting the diverse computational activities of novice programmers”. Journal of Visual Languages & Computing, 39, p. 78-92, 2017.

[22] T.J. Cortina, “An introduction to computer science for non-majors using principles of computation”, in: ACM SIGCSE Bull., vol. 39, ACM, p. 218–222, 2007.

[23] C. W. Liao, C. Ching-Huei and S. Sie-Jhih, "The interactivity of video and collaboration for learning achievement, intrinsic motivation, cognitive load, and behavior patterns in a digital game-based learning environment." Computers & Education, 2019.

[24] F. Zhang, D. Kaufman, and S. Fraser, “Using Video Games in Computer Science Education”, European Scientific Journal, 10(22), 2014.

[25] T. Turchi, D. Fogli, and A. Malizia, “Fostering computational thinking through collaborative game-based learning”, Multimedia Tools and Applications, p. 1-25, 2019.

[26] K. Sung, C. Hillyard, R. L. Angotti, M. W. Panitz, D. S. Goldstein and J. Nordlinger, “Game-themed programming assignment modules: A pathway for gradual integration of gaming context into existing introductory programming courses”, in IEEE Transactions on Education, 54(3), p. 416-427, 2011.

[27] L. A. Gouws, K. Bradshaw and P. Wentworth. "Computational thinking in educational activities: an evaluation of the educational game light-bot." Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ACM, 2013.

[28] L. Klopfenstein, A. Fedosyeyev, and A. Bogliolo, “Bringing an Unplugged Coding Card Game to Augmented Reality”, INTED PROCEEDINGS, 9800-9805, 2017.

[29] S. Heller, “The meaning of children in culture becomes a focal point for scholars”, in The Chronicle of Higher Education, 1998, A14 -A16.

[30] M. Billinghurst and A. Duenser, “Augmented reality in the classroom”, Computer 45(7):56–63. ISSN 0018-9162, 2012.

[31] S. Zheng, “Research on mobile learning based on augmented reality”, in Open Journal of Social Sciences, 3(12), 179, 2015.

[32] I. Radu and B. MacIntyre, “Augmented-reality scratch: a children's authoring environment for augmented-reality experiences”, in Proceedings of the 8th International Conference on Interaction Design and Children (IDC '09). ACM, 210-213, 2009.

[33] S. Goyal, R. S. Vijay, C. Monga, and P. Kalita, “Code Bits: An Inexpensive Tangible Computational Thinking Toolkit For K-12 Curriculum”, Proc. of the TEI '16. ACM, p. 441-447, 2016.

[34] R. Hunicke, M. LeBlanc, and R. Zubek, “MDA: A formal approach to game design and game research”, Proc of the AAAI Workshop on Challenges in Game AI, vol. 4(1), pp. 1722, 2004