CodeCubes: Coding with Augmented Reality

Bárbara Cleto
Escola Superior de Tecnologia, Instituto Politécnico do Cávado e do Ave (IPCA), Barcelos, Portugal
a13993@alunos.ipca.pt

Cristina Sylla
Research Centre on Child Studies Universidade do Minho, Braga, Portugal
cristina.sylla@ie.uminho.pt

Luís Ferreira
2Ai, School of Technology, Instituto Politécnico do Cávado e do Ave (IPCA), Barcelos, Portugal
lufer@ipca.pt

João Martinho Moura
Escola Superior de Tecnologia, Instituto Politécnico do Cávado e do Ave (IPCA), Barcelos, Portugal
jmoura@ipca.pt

Abstract

CodeCubes is interface that uses Augmented Reality to stimulate Computational Thinking in young students. The visual programming blocks are replaced by paper cubes that have an Augmented Reality marker on each face. Each marker represents a programming instruction. The game is composed of three levels. It consists of programming a car course in a racetrack, driving from the start to the final goal.

Code Cubes takes advantage of the physicality offered by Augmented Reality technology. We present the design and development of the game, focusing on its main characteristics and describing the various development stages. We also present the first results obtained by exploring Code Cubes. The results were positive, showing the potential of Augmented Reality interfaces in learning scenarios.

2012 ACM Subject Classification Computing methodologies → Mixed / augmented reality

Keywords and phrases Tangible Interfaces, Augmented Reality, Computational Thinking, Games

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.7

Funding This work was partly founded by Portuguese national funds (PIDDAC), through the FCT – Fundação para a Ciência e Tecnologia and FCT/MCTES under the scope of the project UIDB/05549/2020.

Acknowledgements We are thankful to the students who collaborated in the creation and implementation of CodeCubes, as well as the students of the Club of Programming and Robotics for testing Codecubes and for the improvement suggestions that they presented.

1 Introduction

Digital games are increasingly gaining relevance in educational contexts, their educational potential has been acknowledged, and consequently, their use as an educational tool inside the classroom has increased [29]. The same phenomenon is seen when it comes to Augmented Reality (AR), and although it is not a new technology, it is now commonly used within educational contexts, challenging traditional education [7]. The combination of augmented reality interfaces with educational content, offers students new possibilities of interacting with the real and the virtual worlds. The possibility of overlapping virtual elements, generated by a computer, and the real world, allows the learning experience to be less static and the interaction to be made in real-time, increasing the efficiency and the appeal of teaching and learning [16]. Adding an AR component to the games that are already used within the classroom, will offer students new resources that will allow them to work and cooperate in order to solve problems and come up with their own solutions, narratives, and connections [29].

CodeCubes is a digital game with an AR interface developed to teach scientific principles of computational thinking, through a problem-solving approach. In order to solve the tasks, the player needs to explore, experiment and interact with CodeCubes putting to practice a trial /error method. The use of AR and Virtual Reality (VR) change the process of learning from passive to active, allowing active, real-time interaction with the learning contents [22].

This approach aims at introducing basic programming concepts to children through experimentation. The combination of the AR technology with paper cubes, provides a new and interactive way of exploration, creating a motivating and engaging experience that allows students to learn while playing.

In this article, we present the development process of CodeCubes, detailing its conceptualization and planning. The paper is structured as follows: section 2 presents the theoretical framework that sustains the research, section 3 describes the development of the various prototypes. section 4 presents results obtained in the first tests. Finally, section 5 presents final observations and suggestions for future work.

2 Background and related work

Children and young adults, the so called “digital natives” [22], interact most of the time and (apparently) fluidly with digital media, feeling comfortable in sending messages, playing online, and browsing the internet. However, only a few of them can create their own (digital) content, either games, animations, or simulations. It is as if they can “read” but not “write” [27]. Digital fluency demands not only the ability to talk, browse and interact with digital content but also the capacity to project, create and invent with the new media [27, 26] which implies solving problems, projecting systems, and understanding human behavior [35]. Computational thinking [8] does not only mean knowing how to program, but to think and find solutions for the arising problems using the fundamentals of computer sciences, skills that everyone should acquire, just like reading, writing, and arithmetic [35].

The introductory teaching of computer sciences in schools is changing the existing paradigm and shifting the acquisition of programming language to the acquisition of more generic computational thinking skills [14]. At the same time, there is a greater emphasis on playful approaches that increase the motivation and involvement of the students [14]. Currently, there are many kits, robots and/or toys and digital platforms targeting children and young adults that teach concepts related to logic, algorithms, and programming and that can be divided into three major groups: i) physical or tangible interfaces (in which every component is tangible), ii) digital interfaces (for computers or mobile devices that do not have any physical component) and iii) hybrid interfaces (composed of both virtual and physical components) [36]. The physical/tangible kits can be subdivided into two groups, with or without electronics. The hybrid kits can be subdivided into blocks of tangible programming or blocks of virtual programming [36]. Regarding the digital platforms, there are examples such as LOGO [21], one of the first tools developed to teach the basic concepts of computational thinking; Scratch [30], and SratchJr [31] for younger children (ages 5 to 7 years), two interfaces developed by the Lifelong Kindergarten Group at the MIT Media Lab. Other, such as Blockly (Google developers) [6] or the platform Code.org [11] offer a set of games, whose concept is to make the characters execute a specific task, using visual programming blocks, to teach concepts linked to computational thinking [35]. MIT App Inventor [3], initially created by Google and maintained by the MIT Media Laboratory, is a tool for mobile computing, which also uses block-based visual programming. There are also games for iOS and Android systems, such as Lightbot [18] (there is a Web version as well). Regarding the hybrid kits or interfaces that use tangible programming blocks, one example is the Coding Awbie Game by OSMO [12]. Other well-known examples of tangible programming environments are Lego® Mind-storms [17] and Lego® WeDo [34], Project Bloks [24] or littleBits [19] but although these environments promote creativity and collaboration, they are generally expensive as they are built, using electronic and mechanical components [28]. There are also activities, such as Computer Science Unplugged [5] for introducing the basic concepts of computational thinking as well as computer science concepts in a playful way, without having to use the computer and instead using "unplugged" activities. Some projects such as AR Spot or AR Scratch [25], T. Maze [33] or AR-Maze [15]add augmented reality to hybrid interfaces. The next section presents the CodeCubes development process.

3 CodeCubes

The development of CodeCubes is aligned with the hybrid approach using tangible programming blocks. In the following we present its development process, describe aspects related to usability and interaction, and present the implementation of the various prototypes.

3.1 First Idea

CodeCubes targets students that are starting to learn programming. It combines (physical) paper cubes with AR technology for teaching basic programming concepts. The users manipulate the CodeCubes AR markers, which are glued on physical programming blocks (tangible interface) that represent the instructions of programming. The development process followed a user centered design methodology, involving a small group of students, aged between 13 and 14 years old that collaborated in both, the creation and in the testing of the application, giving suggestions and feedback. This allowed us to adjust different design aspects and make the necessary changes that resulted in the creation of several prototypes.

The prototypes described here were based on games from the Code.org platform [11] and followed the game mechanics of Code.org – Angry Birds – Classic Maze [1], in which the player uses "drag and drop" blocks of visual programming in order to program and overcome the proposed challenges[9].

In the developed prototypes, visual programming, drag and drop blocks were replaced by 3D paper cubes, which have an AR marker on each face that represents basic instructions for programming. Prototype 1 supports programming six instructions: start, up, right, left, down, and end, whereas prototype 2 supports programming four instructions: left, right, up, and down. In order to execute and visualize the programmed actions, the user clicks on the play button, (prototype 1), and on the SPACE key (prototype 2,) if playing on a computer, or by tapping the screen when using a mobile device.

3.2 Implementation of the first idea

The first implementation was carried out in Scratch [30] and aimed at building a house with geometric figures. The players had to program a previously defined path moving a square form to reach a triangle form located at the end of the path in order to build a house [9].

The movement of the square form was programmed, using drag and drop visual programming blocks. These blocks were later replaced by physical (tangible) blocks that represented the programming instructions. Additionally to the conception and implementation of the this idea, the students drew the figures (square and triangle) to be codified so that, after their recognition, they can fulfill the expected instructions, which, in this case, are of creating movement.

It is intended that CodeCubes allows its users to create their markers, print them and build the cubes using paper, scissors, and glue, a process that will give them a sense of ownership. Therefore, we are encouraging hands on learning, in which users interact with physical objects instead of being limited to a screen with digital content [13].

The object recognition is made by using the camera of a cellphone/tablet or the webcam of a computer. After tracking the objects, the image is captured and processed, and the instructions are executed. The movement of the square is shown according to the sequence in which the objects were placed, allowing the users to visualize and receive feedback in real time of the programming.

3.3 First prototype

The first prototype of CodeCubes was created and developed on the game engine Unity 2017.4.0f1 (64 bits) [32] and the platform AR Vuforia [4] to the operating systems Windows and Android.

The prototype allows carrying out six instructions: start, end, right, left, down and up. Each instruction associated with a face of the cube; each face as an AR marker which is detected by the camera of a mobile device.

For the markers to be easily detected by the vision system, several textures with different patterns were used and associated in each instruction. Therefore, the app will more acutely detect the different markers in situations that might be more adverse when it comes to luminosity [8].

This differentiation makes it possible for the simultaneous visualization of different markers at the same time. The platform Vuforia also contains a detector of the marker's image features, with a quality scale for detection. The different textures and patterns, applied to the several markers, allow for a higher number of visual characteristics, in a superior number to 90%, in the scale of the qualification of the platform Vuforia [4].

The game consists of programming a sequence of actions, setting them in the correct order. The cube's face that represents the intended instruction is placed in front of the camera. To execute and see the programmed actions, the user presses the play button.[9]

Figure 1 shows the execution of the right instruction. The player starts by placing the cube, with the START side up, placed towards the camera; after the recognition of the AR marker, an AR cube can be seen. By approximating a cube with a new instruction, the AR cube changes color, indicating its recognition. By pressing PLAY, the instruction is executed, and the user sees the virtually simulated programming carried out with the tangible cubes.

Figure 1 Instructions and execution with the app CodeCubes.

In this phase, the tests were focused on usability; the first results of the tests were positive, and they were related to the motivation of the use of technology. The most limitation that was found was related to the use of the app under certain light conditions, which made it more difficult for the recognition of the markers. It is important to pay attention to the light conditions and to be careful with the surface of the cubes in order to avoid the unwanted reflection that might interfere in the marker's recognition.

3.4 CodeCubes Game

The CodeCubes Game was another prototype created using the NyARToolkit [20] library for Processing [23], an open source library of augmented reality for Java, was used; the library Ani [2] was also used, in order to create the animations and transitions used in the program.

The game developed consists of three levels and in each of them, only the course that the car must go through (Figure 2) changes.

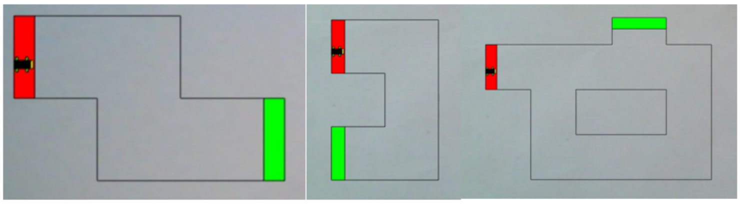

Figure 2 Game levels: Level 1 (left), level 2 (center) and level 3 (right).

The goal is to control the movement of the car, which is over the starting line of a racing track, reaching the end of the road, using an appropriate programming sequence of actions. To move the car, the player must place the AR marker that represents the instruction intended to be carried out in front of the camera [10].

The progress in the game is made by moving the car over the course and reaching the finish line; to do so, the player can execute instructions one after the other or program and execute an instructed sequence.If the player does not execute the instructions or the sequence of correct instructions, but if the car reaches the finish line, it is possible to change levels [10]. This strategy has the goal of motivating students to play while they learn by the process of trial and error and experimenting with different solutions. Therefore, the performance of the students depends solely on the time of execution of each task and allows the student to interact, experiment, and make mistakes without the concern of earning points or losing lives. At any moment, the player can start the level again; the game ends when the three levels are concluded.

The interaction is made by touching the screen, in mobile devices, or with a click of the mouse pad, in the computer version. Four screens have been developed; the starting one, the end one, the gaming screen (in AR mode), and the one to change levels.

In the bottom left corner, there are the buttons to restart the level or to activate and see the execution of the given instructions.

The racing track on the bottom left corner was developed for the pieces to be placed in the upper area. Each time that an object is placed and recognized, the arrow that corresponds to the action appears on the upper left corner. By associating arrows to the objects, the interaction with the game is improved. After objects placement and recognition, instruction processing starts, and the car moves across the track.

The system executes the instructions sequentially, by the physical order they were placed. There is a waiting time between the execution of the given instructions for the user to understand what is being executed and what the user is seeing.

After the sequence execution, and as soon as the car gets to the finish line, a screen appears, one that congratulates the player and that allows the user to play a new level.

One must stress the fact that the level always changes when the car reaches the finish line, whether the programmed sequence is correct or not. The player can restart the level and program each of the instructions one by one or the whole sequence.

4 Exploratory study

The tests were carried out with nine students from the Programming and Robotics Club, RoboESAS – Clube dos Pequenitos, aged between nine and thirteen years old. The average age was ten years old, four boys and five girls, and they are currently in the second year of the programming and robotics club.

With these tests, one intended to detect eventual difficulties of interaction with the game and the interest, motivation, and impact that the use of technology of Augmented Reality (AR) may have in learning.

The activities proposed to students were to perform the same challenge (level 1 of the game CodeCubes). That consists of programming the path of a car on a track so that it can reach the goal, using the Scratch platform and programming one of sever-al robots used in the club. Also, students used the Code.Org platform to play Classic Maze - Angry and explored the developed AR game – CodeCubes. [10]

The instruments and technics for gathering data were: i) direct observation and photo/audio records of the tasks that the participants had to carry out; and ii) a questionnaire and interviews with the participants. The questionnaire was divided into four parts: 1) characterization of the participants, 2) opinion of the participants about the app that they tested, 3) evaluate the impact that AR technology may have in the motivation for learning, and 4) evaluate and classify the activity that they preferred to carry out. All those who participated in the sessions responded to the questionnaires [10].

The results show that, although the participants did not have contact with this type of technology and did not know what it was, they did not show any difficulties in the interaction with the AR game, nor in playing with the AR markers intuitively and autonomously, as pieces of programming [10].

The participants were interested and curious; they were not resistant to participating in the study, nor in using CodeCubes, which turned out to be intuitive and easy to use, without being necessary to explain how it should be used previously.

The activity of programming with AR was the one that the participants enjoyed the most, side by side with programming robots. It was also verified that, on the first level, the students initially preferred to execute the code one instruction after the other; in the following levels, they would set several objects, keeping in mind the course that the car had to go through while manipulating the objects [10]

The enthusiasm of the participants was visible, and it was clear that they were receptive to the use of this type of technology within the space of the classroom, through books, and that it would motivate them in learning the content of other subjects.

5 Conclusions and Future Work

In this article, we described the development process of CodeCubes, that uses Augmented Reality (AR) and physical blocks for teaching basic programming concepts.

The development of CodeCubes had the primary goal of exploring an educational resource based on AR, on the increase of the potential of Computational Thinking (CT) in children aged between 4 and 14 years old, in formal and informal educational contexts, and thus evaluate if this type of technology can, indeed, improve learning. By exploring a new space for gaming and bringing new paradigms of interaction to the classroom, one hopes to have contributed to a better understanding and utilization of AR technology as an environment for learning.

For future work, it would be interesting to allow users to create their own physical markers and tangible objects to manipulate the content of the virtual game. It would be desirable to implement CodeCubes to AR HMDs (head-mounted display) in order to improve the interaction for the fact that both hands would be free for the paper cubes handling. One possible solution would be to create a version without markers using ARCore or creating a version with Leap Motion, which would allow for a type of interaction made with gestures.

Another possible path would be the application of a multiplayer version in order to expand the interaction between the workgroups by creating a collaborative environment and therefore analyze how augmented reality technology can influence the students' social skills; this multiplayer mode might also promote the collaborative resolution of problems.

The suggestions received from the participants of the study include adding sounds to the games, creating more characters, and adding a scoring system. Another aspect that should be improved is the game's appearance and graphic details, by making the characters (or the car) 3D in AR. Moreover, adding more levels, so that students can learn more and gradually, in a more consolidating way, by inserting a new block of instructions that allows the use of structures of repetition and selection, could also be more productive.

References

1. Angry Birds – Code.org – Classic Maze. `https://studio.code.org/hoc/1`, 2019. [Online; accessed 2019/03/15].
2. Ani library Homepage. `http://www.looksgood.de/libraries/Ani/`, 2019. [Online; accessed 2018/02/15].
3. Appinventor. `https://appinventor.mit.edu/`, 2019. [Online; accessed 2019/06/10].
4. AR Vuforia platform. `https://www.vuforia.com`, 2018. [Online; accessed 2018/09/30].
5. Timothy Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. Computer science unplugged: school students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13, January 2009.
6. Blockly Homepage. `https://developers.google.com/blockly/`, 2019. [Online; accessed 2019/09/30].
7. M. Bower, C. Howe, N. McCredie, A. Robinson, and D. Grover. Augmented reality in education – cases, places, and potentials. In *2013 IEEE 63rd Annual Conference International Council for Education Media (ICEM)*, pages 1–11, 2013.
8. José Cerqueira, Bárbara Cleto, João Martinho Moura, and Cristina Sylla. Visualizing platonic solids with augmented reality. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, IDC '18, page 489–492, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3202185.3210761`.
9. Barbara Cleto, João Martinho Moura, Luís Ferreira, and Cristina Sylla. Codecubes-playing with cubes and learning to code. In *Interactivity, Game Creation, Design, Learning, and Innovation*, pages 538–543. Springer, 2018.
10. Bárbara Cleto, Cristina Sylla, Luís Ferreira, and João Martinho Moura. "play and learn": Exploring codecubes. In *EAI International Conference on Technology, Innovation, Entrepreneurship and Education*, pages 34–42. Springer, 2019.

11 Code.org: What will you create? `https://code.org/learn`, 2019. [Online; accessed 2019/06/10].

12 Coding Awbie. `https://www.playosmo.com/en/coding/`, 2019. [Online; accessed 2019/03/15].

13 Anna Fusté Lleixà. *Hypercubes: learning computational thinking through embodied spatial programming in augmented reality; Hyper cubes; Learning computational thinking through embodied spatial programming in augmented reality.* PhD thesis, MIT, 2018.

14 Anna Gardeli and Spyros Vosinakis. *The Effect of Tangible Augmented Reality Interfaces on Teaching Computational Thinking: A Preliminary Study*, pages 673–684. Springer International Publishing, 2020. `doi:10.1007/978-3-030-11932-4_63`.

15 Qiao Jin, Danli Wang, Xiaozhou Deng, Nan Zheng, and Steve Chiu. Ar-maze: a tangible programming tool for children based on ar technology. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pages 611–616. Association for Computing Machinery, June 2018. `doi:10.1145/3202185.3210784`.

16 M. Tumerkan Kesim and Yasin Ozarslan. Augmented reality in education: current technologies and the potential for education. *Procedia - Social and Behavioral Sciences*, 47:297 – 302, 2012. Cyprus International Conference on Educational Research (CY-ICER-2012)North Cyprus, US08-10 February, 2012. `doi:10.1016/j.sbspro.2012.06.654`.

17 Lego Homepage. `https://education.lego.com/en-us/support`, 2019. [Online; accessed 2019/06/10].

18 Lightbot Homepage. `http://lightbot.com/flash.html`, 2019. [Online; accessed 2019/03/15].

19 LittleBits Homepage. `https://www.littlebits.cc/getting-started`, 2019. [Online; accessed 2019/06/10].

20 NyARToolkit library Homepage. `https://nyatla.jp/nyartoolkit/wp/?page_id=166`, 2019. [Online; accessed 2018/02/15].

21 Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., USA, 1980.

22 Marc Prensky. Digital natives, digital immigrants part 1. *On the Horizon*, 9:1–6, September 2001. `doi:10.1108/10748120110424816`.

23 Processing Homepage. `https://processing.org/`, 2019. [Online; accessed 2018/02/15].

24 Projectbloks Homepage. `https://projectbloks.withgoogle.com/`, 2019. [Online; accessed 2019/06/10].

25 Iulian Radu and Blair Macintyre. Augmented-reality scratch: A children's authoring environment for augmented-reality experiences. In *Proceedings of IDC 2009 - The 8th International Conference on Interaction Design and Children*, pages 210–213. Association for Computing Machinery, 2009. `doi:10.1145/1551788.1551831`.

26 Mitchel Resnick. Sowing the seeds for a more creative society. *Learning and Leading with Technology*, 35, January 2007. `doi:10.1145/1518701.2167142`.

27 Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, November 2009. `doi:10.1145/1592761.1592779`.

28 Alpay Sabuncuoğlu, Merve Erkaya, Oğuz Turan Buruk, and Tilbe Göksun. Code notes: Designing a low-cost tangible coding tool for/with children. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, IDC '18, page 644–649, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3202185.3210791`.

29 Karen Schrier. Using augmented reality games to teach 21st century skills. In *ACM SIGGRAPH 2006 Educators Program*, SIGGRAPH '06, page 15–es, New York, NY, USA, 2006. Association for Computing Machinery. `doi:10.1145/1179295.1179311`.

30 Scratch Online Homepage. `https://scratch.mit.edu/`, 2019. [Online; accessed 2019/06/10].

31 ScratchJr Online. `https://www.scratchjr.org/`, 2019. [Online; accessed 2019/06/10].

32 Unity (game engine). `https://unity3d.com`, 2018. [Online; accessed 2018/09/30].

33 Danli Wang, Cheng Zhang, and Hongan Wang. T-maze: A tangible programming tool for children. In *Proceedings of the 10th International Conference on Interaction Design and Children*, pages 127–135. Association for Computing Machinery, 2011. doi:10.1145/1999030.1999045.

34 WeDo 2.0 – Support – LEGO Education. https://education.lego.com/en-gb/support/wedo-2, 2019. [Online; accessed 2019/06/10].

35 Jeannette Wing. Computational thinking. *Communications of the ACM*, 49:33–35, March 2006. doi:10.1145/1118178.1118215.

36 Junnan Yu and Ricarose Roque. A survey of computational kits for young children. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, IDC '18, page 289–299, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3202185.3202738.