LCSMAR, an AR Based Tool to Inspect Imperative Programs

Luis Carlos Martins
ALGORITMI Research Centre / LASI, DI-University of Minho, Braga, Portugal

Lázaro Vinicius Lima
Department of Informatics, Instituto Federal de Brasília, Brasil

Pedro Rangel Henriques
ALGORITMI Research Centre / LASI, DI-University of Minho, Braga, Portugal

Abstract

LCSMAR is a Learning Resource that takes advantage of Augmented Reality in order to promote the development of Computational Thinking among students who are starting to dive in to the world of computer programming. Students can write code in a imperative programming language and, with the help of their mobile phone, they can visualize and analyze the execution of the code they developed, seeing how variable and data structures change over time with each instruction. Augmented Reality tools allow the visualization of abstract concepts that are often misunderstood and that cause misconception among students, which in term should help students develop the abilities to understand and use these abstract concepts, such as data structures, in other areas of application.

2012 ACM Subject Classification Computing methodologies → Mixed / augmented reality

Keywords and phrases Augmented Reality, Learning Resources, Computer Programming, Computational Thinking

Digital Object Identifier 10.4230/OASIcs.ICPEC.2023.3

Funding This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

1 Introduction

Augmented Reality (AR) is a technology that has often been associated with the world of entertainment and gaming[8]. However, in recent years, its has gained more attention for its educational purposes and has started to being used as a Learning Resource (LR)[2]. The use of an AR in an educational environment gives students new tools and possibilities that a normal, tradition classroom cannot give, thus making their new learning experiences less stale and mundane, creating opportunities for new AR tools to be created.

Computational Thinking (CT) is the thought process that going into formulating problems and the solution for them and its an essential skill to have in the 21st Century[9]. CT enclose a range of concepts that define what it is, such as algorithmic thinking, decomposition, patter recognition and abstraction. One of the core concepts of CT is abstraction, that involves focusing on the essential features of a problem or system while ignoring irrelevant details, but it can be a challenging concepts to grasp, specially for students learning any new concepts that doesn't have a concrete representation and that may cause confusions and misconceptions between students.

Since AR displays computer generated information onto the real world, it is possible to the same with abstract concepts, displaying them in a virtual environment that students could visualize and interact with them in a way that isn't possible in tradition ways, thus being an important technology to be used when teaching in the modern world.

Language-Specific Code Simulation with a Mobile using AR (LCSMAR) is a mobile application with AR where students can develop their own pieces of code in a imperative language that then can be visualized in AR environment step-by-step, thus visualizing the execution of the code, the changes to the variables and data structures in real-time. This approach aims to help students understand the underlying functionalities of basic data structures, such as lists, queues and stacks and how to manipulate them in a correct way, while at the same time visualizing and learning how they work.

In this article, we present the development of a AR-based Learning Resource called LCSMAR (Section 5) after discussing in detail its architecture and desired functionalities Section 4). Before that the main areas of research involved in this project, Computational Thinking and Augmented Reality, are discussed in Sections 2 and 3 respectively. Section 6 closes the paper.

2 Computational Thinking

As defined by Wing [9], Computational Thinking is the thought process that goes into formulating problems, being a collection of skill based on the principles of Computer Science, therefore being one of the most essential skills of the 21st century.

It is a very important skill that, such as reading or writing, must be taught to the younger population at an early age, in order to allow them to develop the necessary skills, not only for CS but also for other areas of science, being mainly focused on the aspect of systematizing, representing, analyzing and problem solving skills.

Just like any other skill, it is important to develop and training it and to do so it is needed specialized tools and LR to help teachers thought their students to develop CT.

One of the ways to effectively train CT is through the use of Learning Resources (LR) that, as the name suggests, are tools designed to assist teachers teaching and students learning. These resources, which can either be physical or digital devices, allow students to not only gain knowledge during classes but also allow them to develop said knowledge, therefore making the learning experience interesting, appealing and modern, thus stimulating their interests in acquiring new skills [3].

In order to train CT it is necessary to use LR to develop this thought process. As CT requires a wide range of different skill sets, including abstraction and problem decomposition (Figure 1), it is crucial to have appropriate resources to develop each one of these skills, facilitating students' development of CT skills. These tools must be easy to understand and to use but also must have high potential for students to develop their skills.

There are two main kinds of LRs: *plugged* or virtual resources that must be used with electronic equipment and devices such as smartphones and cameras and *unplugged* resources that don't. Each one of these types offer different advantages that depend on the environment that they are being used.

The initiative CS Unplugged`https://www.csunplugged.org/en/` is a collection of many different *unplugged* activities like card games, puzzles and challenges that aim to build the ground work necessary for students to dive into Computer Science (CS). This initiative not only provides materials for student to learn concepts about CS but also provides teachers with content to teach during classes, thus providing students with a favourable environment to develop and train their CT skills.

One *plugged* LR that's very widely known and used in the work is *Scratch* ([7]) that is used to introduce student to programming and also to CT. There is a big community around *Scratch*, where students can share their creations with other students around the world, as well as tinker and modify other's projects to learn how they work. This tool allows students

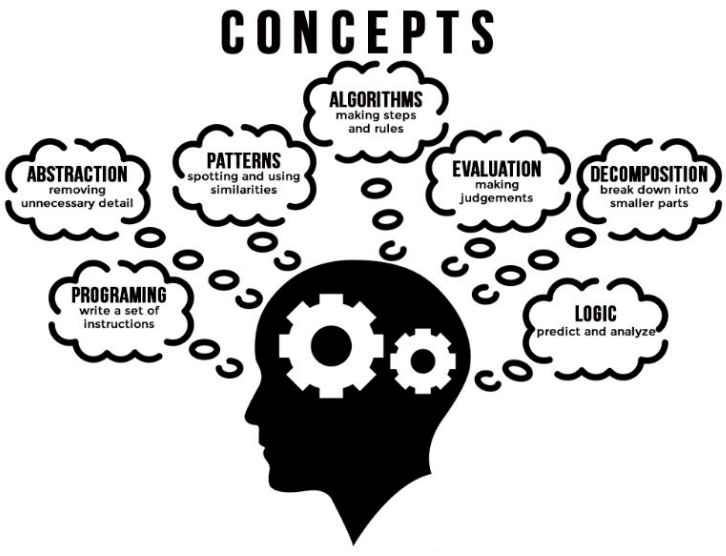

Figure 1 Computational Thinking Concepts that can be trained.

to use blocks of code that can be used interchangeably to make their own creation, while developing and using computer programming concepts such as conditionals, loops and events, while at the same time that they develop CT related skills.

Analyzing the success and impact that *Scratch* has in students, where they get motivated and interested in developing their own applications and games, its possible to conclude that *unplugged* activities have the power to make students more engaged in learning, showing a need for new pieces of technology that provide students with material to learn new sets of skills, starting with CT.

Thus there is a need for new and innovative LR that allow students to learn new concepts and skills that are easy and intuitive to use, while at the same time being effective at delivering its promise of teaching and engaging students.

3 Augmented Reality

Virtual Environment (VE) technologies are becoming more prominent in our world, becoming a key technology for the near future and already having a lot of uses in our society, helping and enhancing our daily tasks.

One of the most popular and used VE technologies is Augmented Reality, being one of the spectrums of the Mixed Reality continuum, as presented by [6]. When using VR the user is completely immersed in a fully virtual environment, separate from the real world, through the use of head-mounted displays and sensory devices, while when using AR devices, the user experiences the real world enhanced by information generated by these devices, adding digital information to the real world. While VR provides a complete virtual experience, AR allows for a mix between real and virtual world.

AR is the enrichment of a real environment by bringing virtual information into the world using technological devices, providing users with computer generated texts, images and/or virtual objects that are superimposed in the real objects that they see, bridging these distinct worlds [1]. AR aims to explore the use of vision and hearing in these environments but it is not exclusive to these senses, as presented by Azuma [1], where all the others senses, such as tact and smell can also be used and explored.

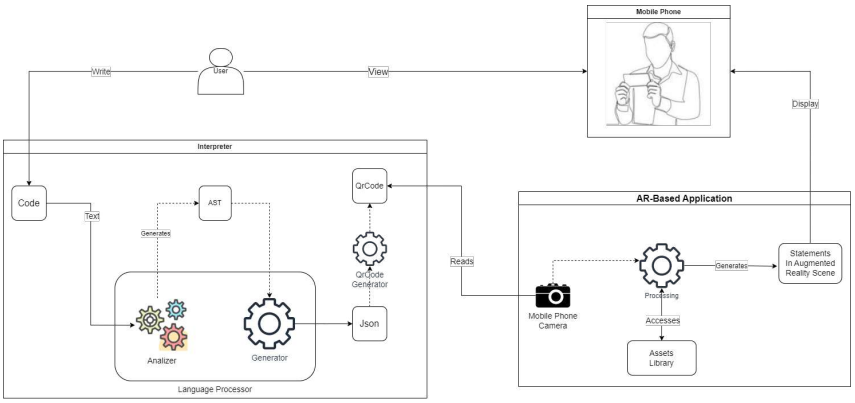

Figure 2 System Architecture.

To take advantage of AR, there are multiple types of devices available to be used that can be subdivided in the following categories. Firstly, head-mounted displays (HMD) are designed to be worn on the head and a projection is displayed in the user's field of view. This type of AR is mainly used in a industrial setting where worked need hand-free access to information. Secondly, handheld devices that are much more portable than the one above since it can be used with the help of a smartphone to display, using the phones cameras, AR information superimposed into the real world. Lastly, using spatial displays, the computer generated information can be displayed directly into the real world without the need for any display. Some examples of these kinds of displays are video-projector or holograms, allowing the virtual world to extend to a wider amount of people simultaneously.

Even tho that at first sight AR look more like a piece of entertainment over anything else, it is a very useful tool to help teachers and students at the same time. With AR its possible to captivating the students attention and motivation while at the same time promoting their own development of skills. The work of [4], CodeCubes, is a game with an AR interface developed to teach scientific principles of computational thinking. In order to solve the tasks, students need to explore, experiment, and interact with CodeCubes putting to practice the trial-and-error method. This approach aims at introducing basic programming concepts to children through experimentation. It combines physical paper cubes with AR technology for teaching basic programming concepts.

Another tool that uses AR in the objective to teach students is the work of [5], stating that programming is an effective way to promote the development of CT, presenting a tangible programming tool that utilizing AR technology for kids, helping them create their own programs by positioning programming blocks and executing code with a mobile device. Through this, kids can learn fundamental programming concepts, such as parameters, loop logic, debug and so on. Following the authors' tests in children, they found that they were engaged and enjoying the game, indicating that these tools are conducive for them to understand several computer concepts. The authors also realized that children are more likely to ignore feedbacks that are received through plain text.

4 Architecture

The developed Learning Resource consist of two main components (Figure 2, each serving a unique purpose and important role in the environment of the system.

The first component Language-Specific Code Interpreter that's written in Python and utilizes the Lark library to do so. This component is responsible for analyzing and interpreting code in a custom imperative language. This library was chosen because it is very flexible and

powerful parsing library for Python, allowing to create the custom imperative language with ease, making it an ideal choice for the project. This component take the code that a user writes, analyzes and executes it and create information that will be transmitted through a Qr-Code and then processed by the second component in the mobile app.

The second component is an AR mobile app that utilizes the Vuforia AR software. In order to gather the processed information from the first component, the ZXing library is used in order to read the generated Qr-Code, which will then be processed in order to layout the initial virtual environment, where the users can see the execution of their written code step by step. The choice to use Vuforia as the AR platform for the mobile app was driven by the fact its a robust piece of software that allows for easy development of AR based programs.

In order to create the Virtual Environment a the AR mobile app uses the data created by the Interpreter in order to display the statement in the AR scene. This data represents a JSON with all the necessary information to be able to advance the code step-by-step while at the same time allow for the user to step back into the execution of the code, only needing to follow the flow of information represented in the JSON itself, thus being a solution not very computationally intensive.

One of the key goals of this project it to develop and AR app that is able to create a tool that can be used in schools that is both easy to use and highly intuitive while also being engaging for the user allowing them to easily visualize the execution of the code they've written step-by-step. The interface was designed to be as easy to use as possible with a clear and simple interface to navigate. Creating an engaging experience was one of the main priorities during the development of the app, with the objective of creating visual that are easy to understand while at the same time being engaging. Despite being able to use HMD displays to take advantage of AR, we decided to stick with the use of smartphones for the fact that they are easier to use, more accessible, an already build-in camera and many students already have one, so it can be used by a larger number of the population.

In order to transmit information from one component to another, many solutions where considered to do so, but the one that made more sense was through the use of QR Codes. QR Codes are two-dimensional barcodes that can easily be read by smartphones. They can stored any type of information from links to website, product details, etc, but can also save binary data that then can be used. So with their help, its possible to transmit information from one component to another with the need to setup additional infrastructure. Since QR Codes have a limited amount of data that can be stored, in order to fit the most amount of data in it as possible, the instructions generated by the interpreter component are compressed before being transformed into a QR Code, thus allowing to storage large amounts of data into it.

5 LCSMAR

The developed LR, as said before, is divided into two components, the first one, the interpreter, it takes the code that the students write in a imperative programming language, interprets it and generates information that can then be used by the second component to lay-out the execution of the code written in a AR environment.

In order to make these two components more independent from each other by not creating a direct way of communication between them, there was a need to find another possible way to transmit information from one component to another, and to do so we used QR Codes, since they are easy to generate and scan, and can be read by almost every modern smartphone. Thus, by using QR Codes there is no need to create server making it an LR that can be used without the need for an internet connection.

Listing 1 Example code of the developed imperative language.

```
VARS
i,evenSum,oddSum:int;
listOdd,listEven:List(int);
STATS
evenSum = 0;
oddSum = 0;
for (i = 1;i<=10;i++){
    if (i % 2 == 0){
        listEven.insert(i);
    } else {
        listOdd.insert(i);
    }
}
while(listEven.len > 0){
    evenSum += listEven.remove();
    oddSum += listOdd.remove();
}
END
```

The main focus of this project was developing an AR application that its easy to use, engaging, appealing and that allowed students to understand better how their code works and how to manipulate and use data structures. Some of the features of the developed application are, visualizing the written code, advancing or stepping back each line of code, see the changes that occur in each variable and data structure, know if the current line of code is inside a conditional block, knowing if the condition is either true or false and know if a line is inside a cycle.

The first step to use this LR is to write the code that will generate a QR Code, that then can be read using the mobile phone. To do so, the users will write their code in a web-browser based application that is able to generate the QR Code thanks to the information generated by the Interpreter (Figure 3).

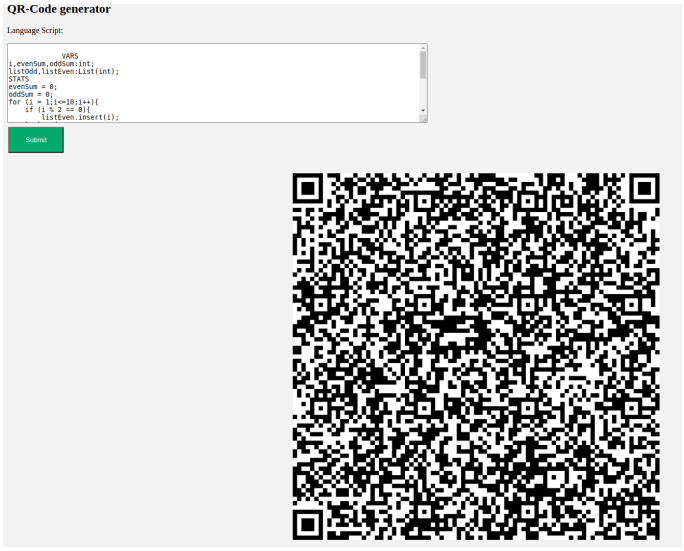

Figure 3 Web App where students can write their code.

After reading the generated QR Code, the starting scene is generated, with the starting variables and the empty data structures, which can then be advanced step by step using the UI. The piece of code 1 will be used to show the features of the AR application. The code puts the numbers between 1 and 10 in a corresponding List of either even or odd numbers, and then it calculates the sum of each, storing these values in their corresponding variable. In the Figure 4 the variables and their states can be seen represented in the blue cubes, that have the name of the variable as well as its state.

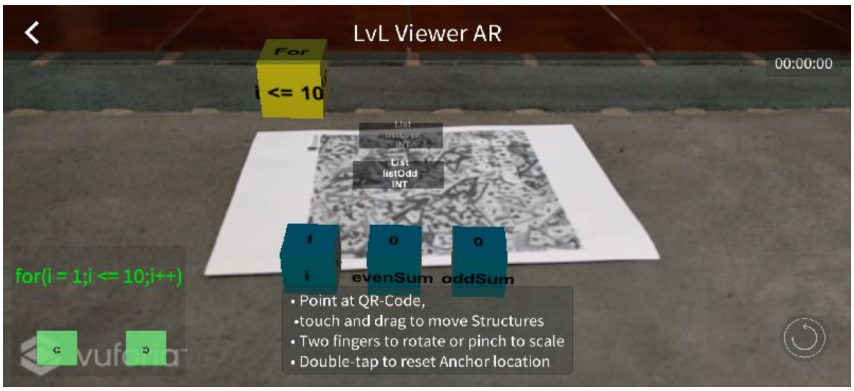

Figure 4 Code execution representing a for cycle.

When the code reaches the for cycle a new block appear representing it, showing the loop condition. In a similar way a block that represents the if condition can either be red (Figure 5)when the condition is false or green if the condition is true.

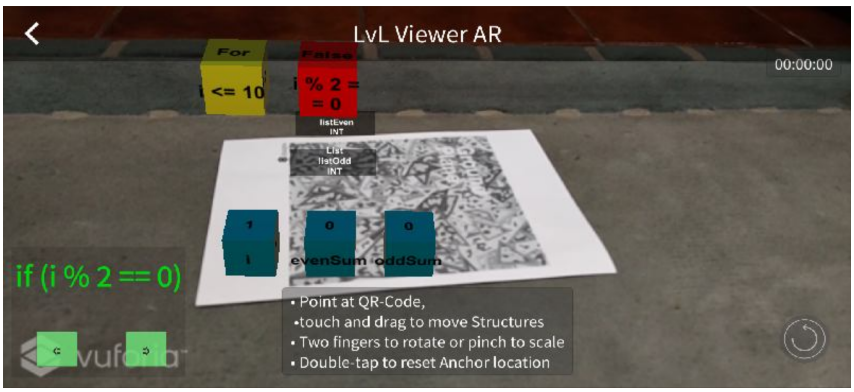

Figure 5 Code execution representing an if condition.

As the code is executed the content of each data structure changes, adding more blocks to each one, representing each element as can be seen in the Figure 6.

Then inside a while loop (Figure 7) that is represented by another colored block, the elements of each data structure will be removed from them and then they are added to the respective variable, thus calculating the sum of the numbers.

This LR allow for students to have a different approach into learning how code works and the intricacies that are the manipulation and use of data structures. By using a variety of teaching methods and tools, such as visual aids, interactive exercises, and real-world examples, students can gain a deeper understanding of the concepts and develop problem-solving skills that they can apply in their future careers. Additionally, a well-designed LR can help students stay engaged and motivated throughout the learning process, which can improve their overall retention and mastery of the material.

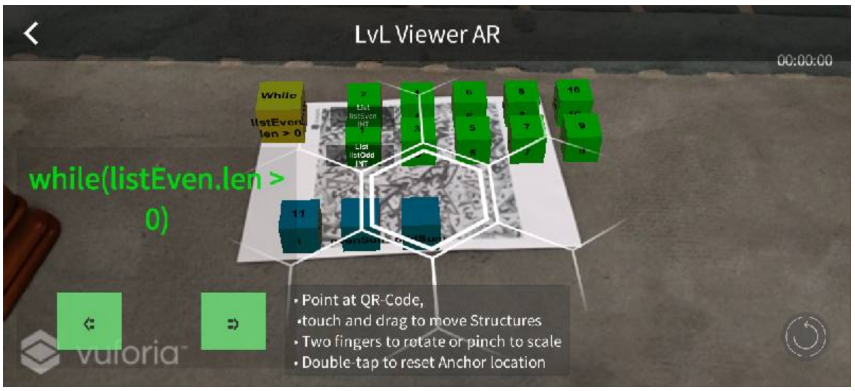

Figure 6 Code execution representing the state of the data structures.

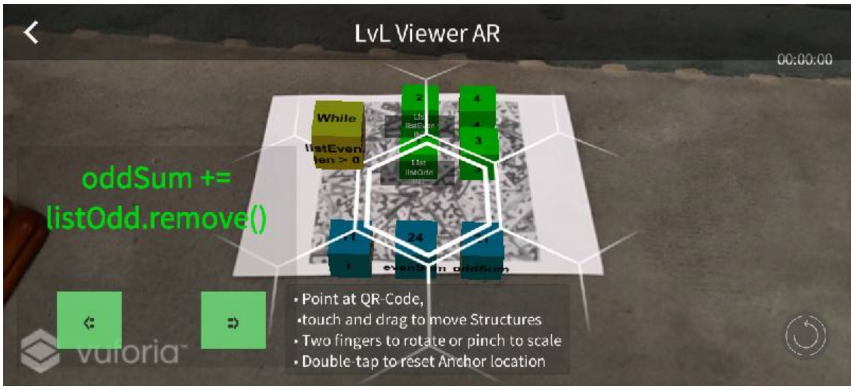

Figure 7 Code execution representing removal of elements from data structures during a while loop.

6 Conclusion and Future Work

In this article, we have presented LCSMAR, an innovative LR that uses AR to enhance students' understanding of important programming concepts and data structures. We have described the two main components of LCSMAR: the interpreter, which generates information about the code written by students; and the AR application, which uses the information generated by the previous component to lay out the execution of the code in an AR environment. At the best of our knowledge, this approach is new and can pave the way to the development of original tools to help the learning process enabling the production of nice visualizations of formal specifications or programs provided at the input. We believe that we can aid Programming teachers to improve the success of their courses. The initial development phase of LCSMAR focused on creating an easy-to-use, engaging, and visually appealing AR application that allows students to visualize step-by-step the execution of his source program emphasizing the data structures manipulation.

While the tests conducted in classrooms have shown promising results, we plan to conduct further experiments with students and educators to refine and improve LCSMAR. By incorporating feedback from users, we aim to create an effective and comprehensive learning resource that can be used to teach programming and data structures in a more engaging and interactive way.

Figure 8 AR application being used.

LCSMAR here discussed is a component of a bigger project that aims at the creation of AR-based Learning Resource to train Computational Thinking. Our idea is to define a strategy to develop in a systematic way, guided by the description of the skills to train, such kind of tools.

References

1 Ronald T Azuma. A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385, 1997.

2 Matt Bower, Cathie Howe, Nerida McCredie, Austin Robinson, and David Grover. Augmented reality in education–cases, places and potentials. *Educational Media International*, 51(1):1–15, 2014.

3 Rona Bušljeta. Effective use of teaching and learning resources. *Czech-Polish historical and pedagogical journal*, 5(2), 2013.

4 Bárbara Cleto, Cristina Sylla, Luís Ferreira, and João Martinho Moura. Codecubes: Coding with augmented reality. In *First international computer programming education conference (ICPEC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

5 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, 2018.

6 Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. Spie, 1995.

7 Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

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

9 Jeannette M Wing. Computational thinking benefits society. *40th anniversary blog of social issues in computing*, 2014:26, 2014.