AR-BASED LEARNING RESOURCE: A DIFFERENT WAY TO LEARN DATA-STRUCTURES

Lázaro Vinicius de Oliveira Lima[1], Luis Gonzaga Magalhaes[2], Pedro Rangel Henriques[2] and Alexandre Cardoso[3]

[1]*Instituto Federal de Brasılia (IFB) Brasılia – DF – Brazil*

[2]*Algoritmi Research Centre/ LASI Universidade do Minho Braga - Portugal*

[3]*Universidade Federal de Uberlandia Engenharia Eletrica – Uberlandia, MG – Brazil*

ABSTRACT

Considering the difficulties in the teaching/learning process of computerprogramming, the development of skills related to Computational Thinking (CT) toprepare people to better solve problems stands out. Augmented Reality (AR) allows for the insertion and interaction with virtual information in the real world. AR is becoming ordinary to humanity, including the design of new teaching/learning strategies that enable the creation of metaphors with virtual information. In this way, it can be said that AR provides a framework for the creation of analogies of complex concepts facilitating their comprehension. This article describes an interactive way of understanding computer instructions (a program) to manipulate data structures (lists, stacks, and queues). The objective of this study is to useAugmented Reality to visualize high-level program statements through a mobile animation system. User instructions are animated in real-time creating virtual 3D representations to help understand operations that handle abstract data-structures used in classic programming. This approach contributes substantially to the construction of educational resources aimed at supporting novice programmers that face critical difficulties to learn adequately Computer Programming. The work described here shows that with effective AR-based Learning Resources, it is possibleto train skills such as analyzing, representing information, planning strategies, andsystematizing resolution schemes, that are crucial to solve problems.

KEYWORDS

Augmented Reality, Learning Resource, Learning Active, Mobile, Computational Thinking, Experiment

1. INTRODUCTION

In the Age where computational knowledge is a keyword, the community should be concerned with developing the skills related to programming. There is a concern to help peopledeveloping skills related to Computational Thinking. Therefore, there is also the concern of the hermeneutic dialogue of teaching-learning in the construction of adequate learning resources (LR). Technologies help us to build appropriate resources. One technology that has been showing great impact on students motivation is Augmented Reality (AR). AR is defined as the insertion of virtual information (created by computer) in a real environment inreal time (Azuma 1997). In line with the great popularity of Augmented Reality, we identified potential to create interactive environments that allow students to absorb programming concepts and how instructions manipulate data structures.

We present LVL (Learning Virtual Language), a language and an AR-based artifact to explore data structures (lists, stacks and queues), to help students in the introductory steps to learn computer programming. Our goal in this work is to show a different process of visualizing high-level program statements using handle-mobile AR to manipulate data structures, a new approach to support novice programmers. We want to demonstrate that a tool can help in visualizing the effects that a code and its operations have on a system.

This paper is organized as follows: Section 2 discusses the ideas and motivations that lead to the project; then Section 3 presents an overview of the system we built aiming at resorting to AR to develop a tool that can help novice programming students; Section4 presents the LVL language processor developed; Section 5 introduces the programming language LVL designed to be used in our tool and demonstrate how the proposal can be realized; Section 6 illustrates with examples the tool actually built; Section 7 discusses an experiment conducted in a classroom; Section 8 concludes the paper with a summary, main lessons learned and future work.

2. THINKING ABOUT DATA-STRUCTURES

First, we need to talk about Learning Resources and new educational strategies that support us. This work have intention to create new Learning Resources to help programmers. Work by (Neamtiu et al. 2005) presents a tool to compare source code from different versions of a C program using the partial abstract syntax tree matching approach. The author explores an interesting approach to show us how important it is to follow software evolution issues. In the evolution of technologies, in the paper (Teng et al. 2018), the authors present an AR learning system for visual representation and interaction in 3D programming... In that work it was demonstrated that AR-based resources, the students have better learning efficiency than with common tools. The authors of (Dass et al. 2018) evaluatedtwo interactive AR coding environments:(1) head-mounted AR with Microsoft HoloLens, (2) mobile AR with ARKit on an iPhone. The author reported that participants enjoyed using mobile AR the most, emphasizing the easy access and manipulation of the system. Another author (Agrahari and Chimalakonda 2020) proposes an Augmented Reality-based software application to teach data structures with the help of abstract syntax trees. Using Unity with Vuforia to show concepts of data structures, Narman and colleagues developed an AR application that visualizes some operations in Array list, Linked list, and Stack (Narman et al. 2020). The author creates a learning environment for students based on AR. The author (Card 1999) says: ”Visualization holds great promise for computational science and engineering, provided we can meet the immediate and long-term needs of both toolmaker and tool users.” But there are different types of visual representation of this information. This information is usually classified into 3 categories: one-dimensional, two- dimensional and three-dimensional (Freitas et al., 2001). In the work (Lima et al. 2021), the authors search for ways to visualize different types of data, their connections and relationships. Using family tree analogies and text structures to present information and its relationships, this work allowed to investigate information arrangements and organizations in a 3D environment.

Works like those discussed above have inspired the work presented in this paper. Inthe next section, we will present the proposed system and its architecture.

3. PROPOSED SYSTEM

A first attempt used the idea of presenting queue, list and stacks and their concepts related to data structure. Each data structure has its own set of methods to perform operations suchas Inserting or deleting elements to introduce the different Data Structure concepts. So, it was possible to choose the type of structure, and the values to be placed in each position. Asimilar work was done by (Narman et al. 2020), but we believe that it is necessary and feasible to go further. In the investigation, we look for greater freedom in textual writing and animmersion in the visual representation of interaction in structures. Bearing these important points in mind, LVL, a small language capable of supporting the declaration and operation with data structures was designed. That language allows the Programmer to declare structures such as lists, stacks and queues, as well as it provides special operators to handle thesestructures.

Figure 1 shows the architecture of the overall system that can recognizes a program given as input (the ’code.txt’ block in the image) and creates an Augmented Reality interactive scenario to help people comprehending the semantics of that input program.

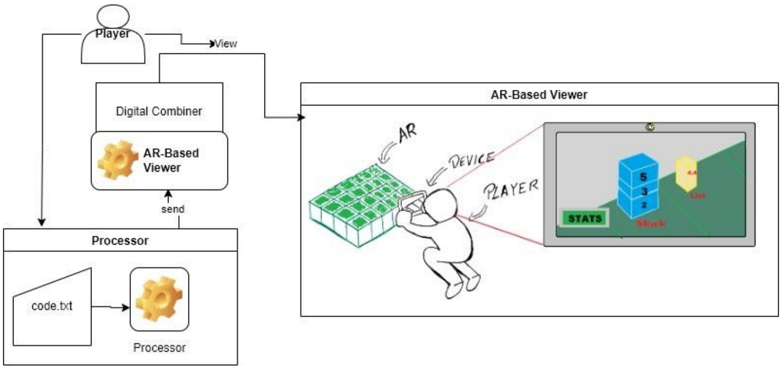

Figure 1. System Architecture

In the sequence of our previous work (see (Lima et al. 2022, Lima et al. 2021, Saraiva et al. 2021) for details) we have decided to use of mobile devices because of their great popularity and the effective power of their technological components. Moreover, that choice offers a low cost solution when compared to Head-mounted display devices. The developed solution can be used on cell phones and tablets that have a camera. The developed mobile APP can be compiled for different platforms simply by configuring the Unity compiler. The system is split into two parts: (1) the analyzer plus translator; and (2) the AR-viewer App.

To implement the ideas, we used Unity 3-D® v.2018.4, that builds Augmented Reality systems supported by the Vuforia® version 9.8 library. Vuforia is a robust library that enables rapid prototyping. The parser and translator were built with Lark, a parsing library for Python. The work aims to create an immersion in textual programming similar to immersion in the use of virtual systems. In the next sections, the implementation of the proposed system is discussed.

4. LVL PROCESSOR

As said above, the overall system is divided into two parts: the input Language Processor (a Translator composed of a lexical-syntactic analyzer plus a code generator); and the AR-based Viewer to display the animation. This section and the next one deal with the language and its processor. LVL is a simple, imperative (or procedural) programming language, specially designed for this project aimed at supporting the creation of sequential one-dimensional Data-structures.

A parser is capable of transforming the input text (the language sentences) into a syntax tree that synthesizes the meaning of the input program and is available for the APP to use this information. After the code is analyzed, the tree is traversed and the information of the input program relevant for the animation (variable declarations and variable manipulations) is extracted and converted to an internal representation in JSON (JavaScript Object Notation) format. After that, JSON description is then transformed into a QR-Code. In that way it can be said that the generated QR-code contains the input statements that the AR viewer can understand and recreate. Pointing the camera of the mobile device to the QR-Code image, the system is able to collect the information extracted by the processor.

A simple-to-use interface was created to work in a web-browser. The user writes the code and presses the button to process the code and generate the corresponding QR- code. With this QR-code, the user points his AR viewer to visualize the result of his program.

5. LVL LANGUAGE

At the present moment LVL system only supports three data structures—LIST, QUEUE and STACK—to represent in the program visualization. Of course, variables of atomic data types (integer, real, and string) can be declared and used, and so will be displayed as components of the system state. We use colored cubes to represent those structures. After declaring the program variables and their respective data types, it is possible to write statements to assign them values. For List, we have the operations Insert(n) and Remove(). Stacks are based on the LIFO principle (Last In First Out), the element inserted at the last, is the first element to come out of the list. The language provides operators Push(n) and Pop().

The language also allows the use of Queues. The queues are based on the FIFO (First In First Out) principle, the first element inserted, is the first element to come out of the list. To work with Queues, the language provides the operators Enqueue(n)and Dequeue().

Listing 1 illustrates a program written in LVL language that declares a variable Stack, and operates with it. Figure 2 shows the visual environment produced by LVL system for the stack code listed.

```
VARS
    a, b, c: INT;
    pilha : STACK( int )
STATS
    a=2;b=4;c=6;
    pilha.push(2);
    pilha.push(4);
    pilha.pop();
    pilha.push(6)
END
```

Listing 1. Example of a program using a Stack structure

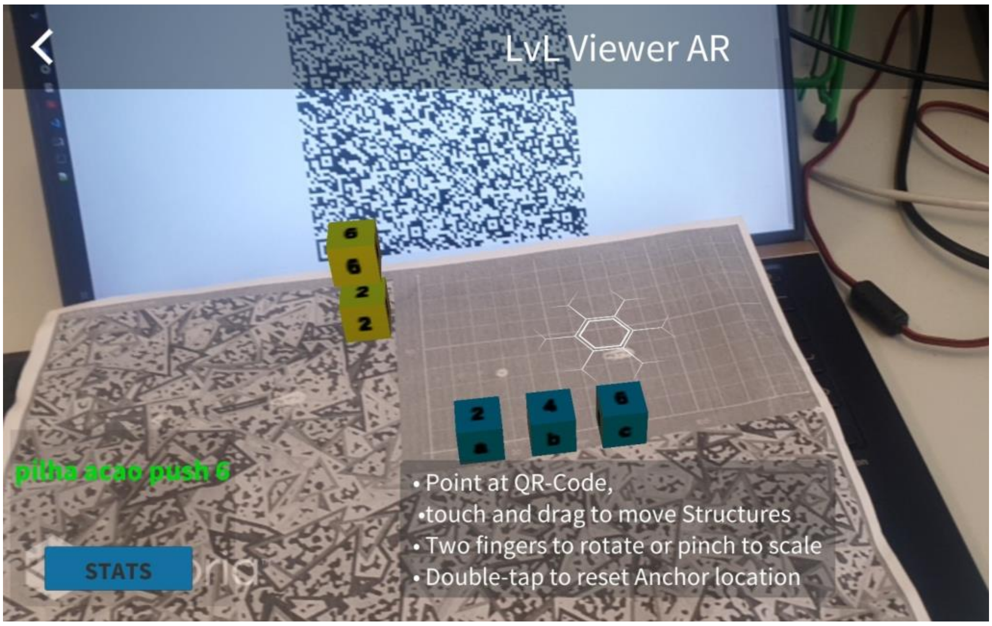

Figure 2. Screen capture of LVL animating the Stack program example

Next, more examples of LVL programs and a description of the AR-Viewer will be provided.

6. AR-BASED VIEWER

AR Viewer uses the QR-code to read the information processed by the Compiler/Processor. This section explains how the user views the information through the App. Figure 3, illustrates how to use the mobile.

The user detects with mobile the ground and then reads the QR-code. The App is very simple to use.

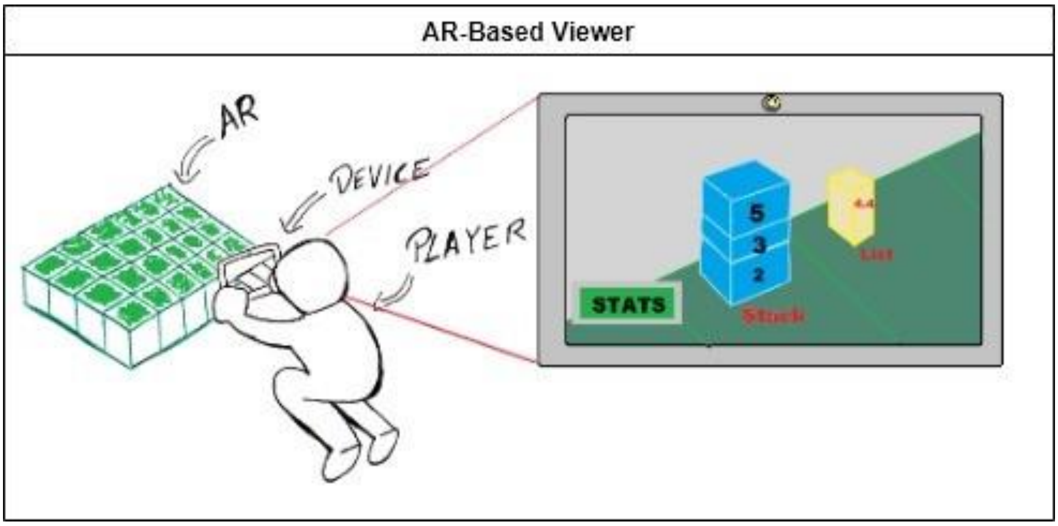

Figure 3. AR viewer app

First access the menu to turn on the camera, then point to the generated QR-code. The viewer reads the instructions and projects the structures without a marker. The system was built to be able to show the effects of each instruction manipulating the data.Examples of a short program and visual results are shown in Listings 2 and 3 and respectivelyin Figure 4 and Figure 5.

```
VARS
    a, b, c: INT;
    lista:list(int)
STATS
    a=2; b=4; c=6;
    lista.insert(2);
    lista.insert(4);
    lista.remove();
    lista.insert(6)
END
```

Listing 2. Example of a program using a List

```
VARS
    a, b, c: INT;
    fila: queue(str);
    pilha: stack(int)
STATS
    a=2;    b=4;
    c=6;
    pilha.push(5);
    pilha.push(a);
    pilha.push(6);
    fila.enqueue(”a”);
    pilha.pop()
END
```

Listing 3. Example of a program using a Queue

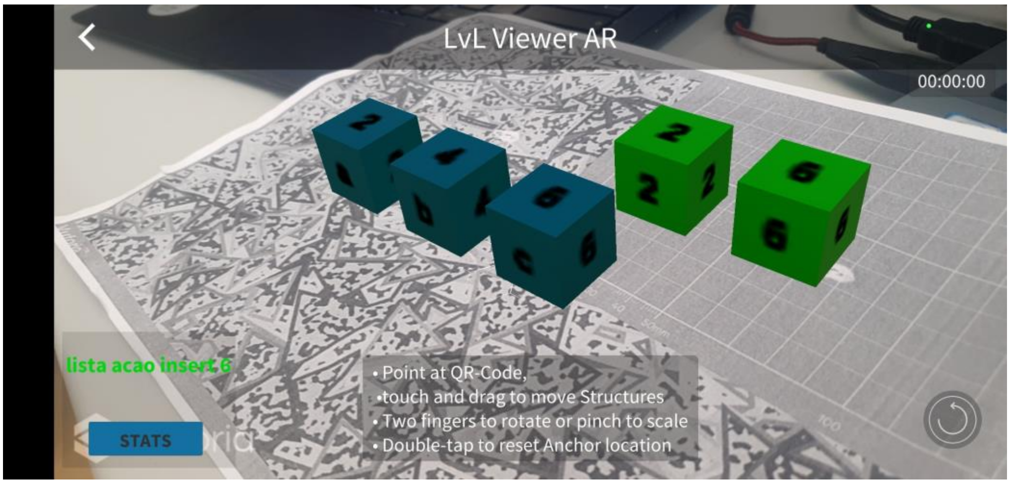

Figure 4. Screen capture of LVL animating the List program example

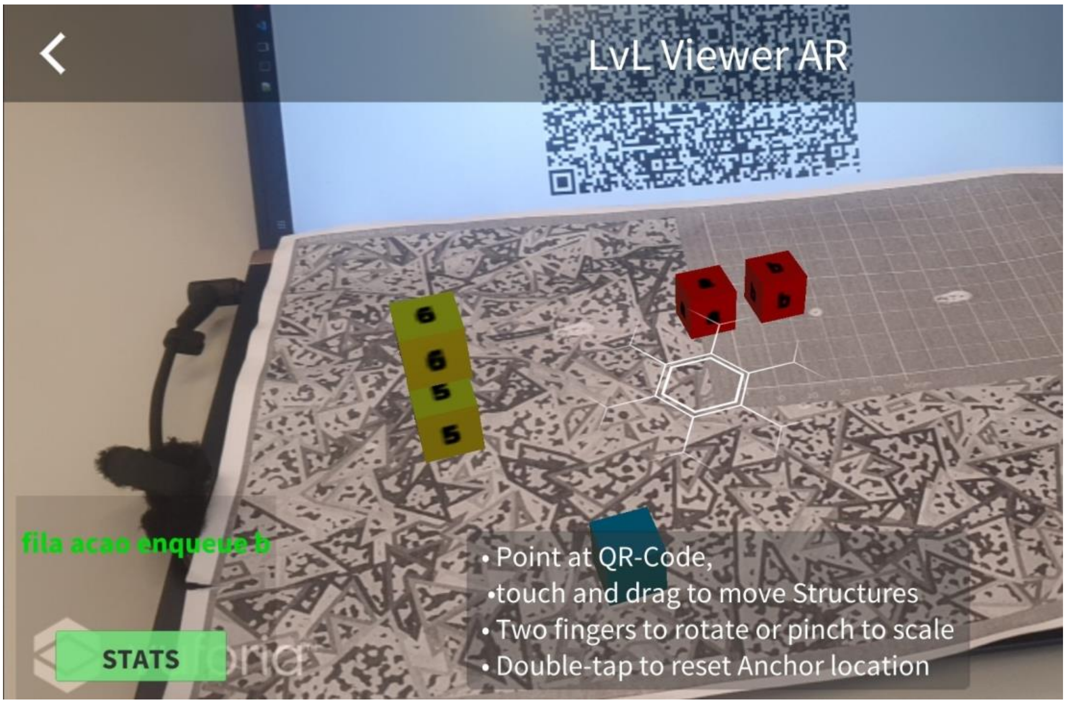

Figure 5. Screen capture of LVL animating the Queue program example

Using the screen of the mobile device, the user can touch and navigate through the program statements, observing how the content of the various data structures (the values stored in each one) evolve.

7. RESULTS AND DISCUSSIONS

To validate the AR-based Learning Resource created, an experiment in a real classroom was conducted. That experiment took place with a group of students graduating from the IT course at University in Brasilia. A total of 7 male students participated in the experiment, aged between 20 and 24 years. The presentation was carried out in the classroom. After introducing the Language LVL, a link was shared to download the activity and the app. The activity consisted of using the LVL language to write a small program manipulating the 3 different structures available in the system (List, Stack and Queue). To carry out the activity, the students spent an hour writing the script and compiling it with the LVLprocessor available online. At the end of that one hour task, the students answered a questionnaire prepared/designed according to Evaluation of a Mobile Augmented Reality Application (MAREEA) (Pombo et al. 2022) as an Outdoor Learning Tool. Four evaluation factors are defined in the MAREEA evaluation model: *Usability, engagement, motivation, and active learning*. All students were able to successfully complete the activities. As response format, the use of the Likert scale is defined. Student feedback indicates that all students agree with the statements contained in the MAREEA questionnaire. In the open question, one student wrote that the system was "Good for Learning". Corroborating with the thesis that an Augmented Reality tool can help in understanding complex problems. The experiment had good feedback showing that all students were able to understand the language and manipulate the visualization tool. Interface improvements were suggested to improve the visualization of subtitles in structures. It is necessary to expand the experiments with a more significant number of students.

8. CONCLUSION

This paper reports the design and development of an immersive tool to explore the manipulation of data structures through an imperative programming language. The proposed system is able to represent, in an interactive virtual environment, variables of different types— atomic, lists, stacks, or queues—defining the state of a program. We believe that this system, with the AR-Based viewer, can be used as a Learning Resource allowing the user to inspect his programs with a mobile device in order to better absorb concepts related to data structures. Vuforia library allows for surface recognition removing the need for markers. Taking advantage of the best practices of AR, one can align the objectives of teaching algorithms and data structures with 3D visualization so that the teacher can write a program with declarations and operations and thereafter the students can visualize their effects, this is the program behavior. Similar to some high-level debuggers, the AR-based Viewer shows all declared variables and allows to evolve step-by-step along the sequence of the program statements, providing a new and attractive environment to understand through the visual aid how statements change the value of the declared structures.

One of the biggest challenges in the development of this project was the question of how to transmit the information extracted from the input program by the LVL processor to the Viewer so that it can render and display that data. Initially, wireless transmission approaches were considered, but as the device camera is already used for the creation of the AR immersive scenario, we decided to resort to QR-code technology to overcome that issue. It was actually a nice solution but we are aware that may pose scalability problems. Another interesting challenge was related to the visual organization of all the objects in the 3D scenario. It was necessary to build specific scripts in Unity to perform the layout of one or many structures of the supported types aiming at the construction of a nice and clear scenario. With the arrangements made, objects are grouped keeping their relative position in a stable image. Moreover, a color mapping is used so that the user can easily identify each data structure type. The ideas and the proof-of-concept presented and discussed along the paper proved that the development of Learning Resources with new technologies such as AR is a promising approach. AR allows total control of the environment. This means that the user can move around the surface of the chosen projection, pause or repeat all simulations to better visualize the structures. Summing up, we can say that Augmented Reality allows us to create analogies of simple or complex concepts enhancing their comprehension. As future work, it is intended to extend LVL language with conditional and cyclic statements, not yet included. Also, improvements in the LVL Processor are planned to decrease the processing effort in the mobile device side. Actually, the JSON description generated can contain more information about the program behavior avoiding its inference by the mobile APP, reducing time and power consumption. The user interface can also be improved using guided audio.

ACKNOWLEDGMENT

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

REFERENCES

Agrahari, V. and Chimalakonda, S. (2020). Ast (ar)–towards using augmented reality and abstract syntax trees for teaching data structures to novice programmers. In *2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)*, pages 311–315. IEEE.

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

Card, M. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.

Dass, N., Kim, J., Ford, S., Agarwal, S., and Chau, D. H. (2018). Augmenting coding: Augmented reality for learning programming. In *Proceedings of the Sixth International Symposium of Chinese CHI*, pages 156–159.

Freitas, C. M. D. S. and et, a. (2001). Introducao a Visualização de Informação. *Revista de Informtica Teorica e Aplicada*, 2:143–158.

Lima, L. V., Sousa, M., Magalhaes, L. G., and Henriques, P. R. (2021). Understanding effects of the algorithm visualized with ar techniques (short paper). In *Second International Computer Programming Education Conference (ICPEC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik.

Lima, L. V. O., Saraiva, F., Magalhães, L. G., Henriques, P. R., and Cardoso, A. (2022). AR- based Resources to train Computational Thinking Skills. In *International Conference in Information Technology Educations - (ICITED 22)*.(to be published).

Narman, H. S., Berry, C., Canfield, A., Carpenter, L., Giese, J., Loftus, N., and Schrader, I. (2020). Augmented reality for teaching data structures in computer science. In *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 1–7. IEEE.

Neamtiu, I., Foster, J. S., and Hicks, M. (2005). Understanding source code evolution usingabstract syntax tree matching. In *Proceedings of the 2005 international workshop on Mining software repositories*, pages 1–5.

Pombo, L., Marques, M., Afonso, L., Dias, P., and Madeira, J. (2022). *Evaluation of a Mobile Augmented Reality Game Application as an Outdoor Learning Tool*, pages 321–343.

Saraiva, F., Lima, L. V. O., Araújo, C., Magalhães, L. G., and Henriques, P. R. (2021). SHREWS: A Game with Augmented Reality for Training Computational Thinking. In Henriques, P. R., Portela, F., Queirós, R., and Simões, A., editors, *Second International Computer Programming Education Conference (ICPEC 2021)*, volume 91 of *Open Access Series in Informatics (OASIcs)*, pages 14:1–14:10, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Teng, C.-H., Chen, J.-Y., and Chen, Z.-H. (2018). Impact of augmented reality on programming language learning: Efficiency and perception. *Journal of Educational Computing Research*, 56(2):254–271.