SSPOT-VR: An immersive and affordable mobile application for supporting K-12 students in learning programming concepts

SSPOT-VR: An Immersive and Affordable Mobile Application for Supporting K-12 Students in Learning Programming Concepts

Abstract

High-resolution displays on mobile devices, accurate motion sensors, and efficient mobile processors have taken virtual reality (VR), essentially employed in laboratory, to everyday environments, including homes, workplaces, and classrooms. Regarding programming education, it has been investigated in conjunction with various educational strategies, such as block-based programming (BBP), metaphors, and storytelling. However, studies that adopt VR predominantly employ high-end head-mounted displays (HMDs) and powerful computers to deliver interactive and immersive learning experiences. Conversely, investigations involving mobile platforms and low-cost HMDs often lack user interactivity. Towards filling that gap, this study introduces SSPOT-VR (Space Station for Programming Training in Virtual Reality), a cost-effective solution tailored for children and teenagers that integrates interactive methods for the teaching and learning of programming concepts and the simulated experience of an immersive digitally created world. Three surveys, namely S_1, S_2 and S_3, involving SSPOT-VR and K-12 students were conducted. S_1 and S_2 focused on user acceptance ($n_1 = 124$ and $n_2 = 16$) and S_3 centered on knowledge retention ($n_3 = 31$). The results indicate students are inclined to accept SSPOT-VR as a valuable educational tool, since it effectively facilitates the retention of programming knowledge through its engaging and interactive learning experiences. By choosing more cost-effective equipment, this research supports the existing body of knowledge while also providing a detailed description of how an effective solution is designed, developed, and used. The approach enhances both affordability and potential applications of immersive VR in programming education.

Keywords: teaching and learning of programming concepts, mobile immersive virtual reality, block-based programming, storytelling, K-12 education

1 Introduction

Programming skills have become a commonly required asset for most college and university students (Luxton-Reilly et al, 2018), as well as a typical subject among children and teenagers worldwide (Kalelioglu, 2015). Among European education systems and in many developed countries in North America and Asia, programming skills are currently part of the learning outcomes for mostly primary and secondary schools (European Commission/EACEA/Eurydice, 2019; Guo and Ottenbreit-Leftwich, 2020). Concomitantly, computers and programming have been slowly inserted in schools of developing countries (Osín, 1998; Apiola and Tedre, 2012).

The obtaining of programming skills requires both understanding and practice of several computer programming concepts (Sigayret et al, 2022); therefore, the use of block-based programming (BBP) is a distinguished introductory approach that supports learning and practice of those concepts (Weintrop, 2019). BBP enables users to develop and share interactive applications they have created through a visual programming language (VPL) (Weintrop and Wilensky, 2015), which is accessible across a wide range of devices, including computers and smartphones. Notably, Scratch[1] stands out as a high-level BBP language and platform. Additionally, BBP is actively exploring the integration of emerging technologies, such as virtual reality (VR), towards enriching the user experience and offer new possibilities for the teaching and learning of programming.

Studies have explored the use of VR for enhancing the teaching and learning of programming (Vosinakis et al, 2014; Stigall and Sharma, 2017) often by combining immersive VR with high-end head-mounted displays (HMDs) and robust computer systems, among other learning strategies (Vincur et al, 2017; Segura et al, 2020; Jin et al, 2020). Teaching and learning solutions designed through VR enable students to experience learning in a high level of immersion and presence (Radianti et al, 2020). The use of HMDs and immersive VR can improve knowledge retention and recall when compared to a screen (Krokos et al, 2019). Additionally, in comparison with traditional teaching and learning methods, immersive VR provides active learning experiences driven by challenge, engagement, and playfulness, fostering situated knowledge and improving knowledge retention (Shi et al, 2022).

Despite technological advancements and benefits to students and educators, there still are gaps related to immersive VR for the teaching and learning of programming (Pellas et al, 2021; Sukirman et al, 2022). As an example, studies that use mobile platforms to develop immersive VR applications for programming education (Tanielu et al, 2019; Sunday et al, 2022; Agbo et al, 2022) often lack user interactivity, hampering an active knowledge acquisition through practical tasks. According to Sukirman et al (2022), approximately half of the research of VR on supporting the teaching and learning of programming adopts only high-end HMDs and powerful computers. Apart from

[1] https://scratch.mit.edu

investigations on VR based on high-end HMDs, the way children and teenagers engage with mobile immersive VR applications that use low-cost HMDs and smartphones must be analyzed.

The present study is important especially because it considers: (i) differences in affordability between powerful computers/high-end HMDs and smartphones/low-cost HMDs (Jerald, 2016); (ii) the increasing availability and use of mobile devices among children and teenagers (Rideout and Robb, 2020), and (iii) that most contemporary smartphones include appropriate hardware for running immersive VR applications (Jerald, 2016). As cited by Pellas et al (2021) and Sukirman et al (2022), the literature lacks studies of a mobile and low-cost immersive VR application that: (i) supports children and teenagers in the teaching and learning of programming concepts, (ii) uses low-cost HMDs to provide an immersive experience, and (iii) enables users to experience programming concepts through practical tasks based on BBP methods. On the other hand, advances in mobile connectivity and hardware are expected to continue transforming education (Pelletier et al, 2022).

This study reports on the planning, creation, and evaluation of SSPOT-VR (Space Station for Programming Training in Virtual Reality), a low-cost and modern application that supports children and teenagers in the teaching and learning of programming concepts through mobile immersive VR, BBP, and storytelling. The study was guided by the following research questions: *(R. Q. 1) Would K-12 students accept the use of our mobile immersive VR application for supporting their learning of programming concepts?* and *(R. Q. 2) Do K-12 students retain programming knowledge after using our application?*

The goal was to create an engaging, intuitive, and attractive solution for supporting the teaching and learning of simple programming commands and basic programming control structures, such as definition and operation of an algorithm, variables, and loop commands. The study offers two main contributions, of which the first refers to the building of expertise by bridging the gap on the use of mobile devices, immersive VR, BBP, and storytelling towards supporting the teaching and learning of programming concepts to children and teenagers. The second concerns the effectiveness of the application, which is expected to be accepted by children and teenagers and to support them in retaining programming knowledge. The results are expected to correspond with those from related studies in terms of user experience and learning outcomes. More affordable equipment of similar quality was used, thus helping mitigate digital divide.

2 Literature review

2.1 Teaching and learning of programming to K-12 students

Computer Science (CS) and its associated technologies perform an important role in shaping our lives and influencing the global economy (Computer Science Teachers Association, 2017; ACM and IEEE, 2020). To be literate citizens in a computing intensive world and to be prepared for careers in the 21st century, everyone should have a clear understanding of the principles and practices of CS (Computer Science Teachers Association, 2017; ACM and IEEE, 2020). Despite the significance of CS in national economies and societal institutions, it was only in the past decade that innovations such as social networks, online news, and internet commerce have made information technology an integral part of daily life for many worldwide (Kong and Abelson, 2022). Such growing ubiquity has increased the demand for CS education, even at the primary school level, regarding its incorporation as an essential subject for supporting students for a world molded by information technology (Kong and Abelson, 2022).

Advances in CS have expanded people's ability to solve problems on a higher scale, using strategies and resources that were not available until recently (Computer Science Teachers Association, 2017; ACM and IEEE, 2020). Towards benefiting from those evolutionary changes promoted by the rapid advances in technology, primary and secondary education students must learn and practice new skills, such as programming skills (Computer Science Teachers Association, 2017; ACM and IEEE, 2020), which encompass the capacity to write, design, and create computer programs (Kalelioglu, 2015). Developing proficiency in programming demands several prerequisites, such as application of in-depth reading skills and use of metacognitive abilities for an effective and clear solution of programming problems (Kalelioglu, 2015).

Wing (2011) emphasized that, in addition to reading, writing, and arithmetic, programming skills are crucial for cultivating analytical abilities in children and teenagers. They also support certain attitudes, including confidence in dealing with complexity, persistence in addressing challenging problems, tolerance of ambiguity, and proficiency in handling open-ended issues (Computer Science Teachers Association, 2017; ACM and IEEE, 2020). Therefore, programming is not merely a singular skill but a collection of resources, competencies, and experiences associated with a complex cognitive activity (Computer Science Teachers Association, 2017; ACM and IEEE, 2020).

According to the Computer Science Teachers Association (2017), the core concepts of CS include: (i) computing systems; (ii) networks and the internet; (iii) data and analysis; (iv) algorithms and programming, and (v) impacts of computing. The core concept "algorithms and programming" encompass different programming subconcepts, such as algorithm composition, loops, conditionals, functions, variables, decomposition, abstraction, among others (Computer Science Teachers Association, 2017). In this study, programming concepts refer to the subconcepts nested under the core concept of "algorithms and programming" outlined in the K-12 CS Standards developed by the Computer Science Teachers Association (2017).

Since the development of programming skills in early childhood continues to progress, evidence of the various tools and approaches that have emerged in recent years is required for informing policy and practice (Bers et al, 2022).

Strategies, tools, curricula, and frameworks aimed at promoting CS in K-12 education must be designed and implemented in a manner that actively engages students and supports learning through playing. It is important to recall that younger K-12 students typically have shorter attention spans and more limited working memory, despite their inherent curiosity (Bers et al, 2022).

As stated by Pears et al (2007), an educational tool that supports teaching and learning of programming according to its strategy can be classified as: (i) visualization tool; (ii) automatic evaluation tool; (iii) programming support tool; (iv) microworld, and (v) another tool that encompasses strategies not included in the previous categories. Among visualization tools, those that adopt simplified programming methods (e.g., BBP) must be highlighted. Visualization strategies support mainly the learning of programming concepts, which is achieved through visually engaging activities that serve as intermediaries between students and programming content, subsequently enhancing students' motivation (Weintrop, 2019).

BBP is a VPL based on a jigsaw metaphor and its languages, tools, or environments enable users to assemble functional applications using only drag-and-drop interactions (Weintrop and Wilensky, 2015). Combining the advantages of component-based programming and end-user programming, BBP promotes the integration of several pieces of code in a form of blocks to develop applications in a short period of time, within estimated cost, and without any programming skills needed (Mohamad et al, 2011). As reported by Weintrop et al (2017), BBP tools are effective to enable novices to write successful programs with none or little prior experience, as well as they can serve as an accessible introduction to the learning of traditional programming languages.

As previously noted, BBP primarily operates on computers and mobile devices, with prominent platforms like Scratch. Nevertheless, BBP is increasingly branching out into emerging technologies, including VR, which combination efficiently presents: (i) abstract concepts, such as programming ones; (ii) behavior of elements that can not be seen, such as a computer memory; (iii) behavior of very large or very small elements, such as galaxies or atomic structures; and (iv) other educational and Science concepts (Jerald, 2016). Numerous studies have investigated into the potential of VR for enriching the teaching and learning of programming (Chandramouli et al, 2014; Sharma and Ossuetta, 2017). These investigations often integrate immersive VR experiences with high-end HMDs and robust computer systems, in conjunction with BBP methodologies, gaming elements, storytelling, analogies, metaphors, and so on (Vincur et al, 2017; Segura et al, 2020; Jin et al, 2020).

2.2 Virtual reality and high-end head-mounted displays

Although the term VR is commonly used to describe imaginary worlds that only exist in computers and our minds, the VR technology enable us to create and use computer-generated digital environments that can be interactively experienced as a real environment (Jerald, 2016). Therefore, an ideal VR environment enables users to navigate around objects and touch them as if they were real, despite its complexity (Jerald, 2016). Real and virtual elements are separately handled, since the objective of VR is to remove user's perception of the real world and make available only the perception of the virtual environment (Milgram et al, 1995).

VR defines environments consisting exclusively of virtual objects, such as computer graphic simulations, virtual environments, or virtual worlds, which may or may not represent an existing environment, either based on a display, or using immersive equipment (Milgram et al, 1995). Since VR involves the experience of being in a location that differs from one's physical surroundings, this virtual environment can either replicate the world or create entirely imaginary environment (Jerald, 2016). In this context, HMDs play a fundamental role by projecting images of the virtual environment directly into the user's eyes through two small monitors, using the principle of stereoscopy (Jerald, 2016).

Although VR is an alternative reality created by computer, it is perceived by human's sensory systems almost in the same way as the real world and emotionalizes, teaches, entertains, and responds to external actions without the need to exist in a tangible way (Jerald, 2016). Besides that, tangibleness has already been part of a few sophisticated virtual worlds and environments, enabling those applications to be further indistinguishable from reality (Jerald, 2016). The area of education can substantially benefit from VR applications, since VR provides users a new way of visualizing concepts and experiencing situations (Jerald, 2016). Many researchers have argued VR environments exert a positive effect on the motivation for programming and students are more engaged in the activity by exploring and solving problems in the virtual and realistic environments (Seralidou and Douligeris, 2021).

When used in conjunction with powerful computers, high-end HMDs offer users a profound level of immersion and interaction within VR applications (Radianti et al, 2020). This immersive experience allows users to engage in what is termed "six-degrees-of-freedom" (6DoF) (Radianti et al, 2020). 6DoF implies the simultaneous tracking of user's physical position and rotation within the virtual environment (Jerald, 2016). In other words, when users move in the real world, their corresponding movement is replicated within the virtual one, resulting in a shift in their position as they walk and an adjustment in their visual perspective.

Recent studies have presented immersive virtual worlds based on different sophisticated VR devices to support students in learning programming. To make BBP more immersive and enable users to experience the knowledge through immersive VR, the studies entitled Cubely (Vincur et al, 2017) and VR-OCKS (Segura et al, 2020) adopted the same high-end HMD (HTC Vive plugged to a computer) to support children, teenagers, and adults in experiencing some basic programming commands and concepts for the first time (e.g., what is an algorithm, the algorithms composition, loops, and conditionals structures). Inside their immersive virtual worlds created with the Unity 3D game engine, those studies share the goal of moving users through a path from an initial position to a final one while requiring them to solve a different puzzle with pre-established blocks that symbolize simple programming commands and basic programming control structures.

The studies aimed at using both appeal and potential of VR for creating an engaging and intuitive method for teaching programming, thus attracting users to the world of programming. Students well-rated Cubely and VR-OCKS through questionnaires that were created by the authors and reported preferring to learn using the VR application rather than conventional methods (Vincur et al, 2017; Segura et al, 2020). As stated by Radianti et al (2020), the majority of studies tend to concentrate on VR's development potential rather than providing comprehensive reports on real-world teaching and learning experiences or the lessons acquired through its application.

VWorld (Jin et al, 2020), a third study, was also created using Unity 3D; however, it adopted a different high-end HMD (Oculus Quest standalone) to boost creativity and computational thinking skills of children and teenagers through a BBP method. It aimed to support the learning of programming by enabling children to create and manipulate the behavior of virtual objects by constructing algorithms using pre-established blocks that symbolize motion movements and sound recordings. Although VWorld is intended for children, authors evaluated the pilot version of their VR system with a reduced sample of undergraduate students.

Since VR can adopt game technology to be developed and are inevitably based on 3D graphics, authors usually describe their studies simply as games, although no gamification aspects or game theories are mentioned, such as scores, rankings, and so on (Susi et al, 2007). Nevertheless, when the application is game-like and uses game technology, game can be a way to conversationally name the application (Susi et al, 2007). Nowadays, a large part of this can be currently done at a low cost, being only necessary to have a desktop computer, laptop, or a mobile device together with a HMD or VR adapter to be able to interact with virtual objects in non-immersive or immersive environments with high-definition graphics.

2.3 Low-cost virtual reality head-mounted displays

Given the rise of mobile devices among children and teenagers (Rideout and Robb, 2020) and the educational advantages of VR (Pellas et al, 2021; Pimentel et al, 2022), research must investigate the way such public interacts with those technologies and educational strategies. Until recently, immersive applications were costly and reserved for specialized use cases (Pimentel et al, 2022). However, with the advent of high-resolution mobile screens, precise motion sensors, and highly efficient mobile processors, the panorama has changed (Pimentel et al, 2022).

On the other hand, there are concerns that VR tools, apps, and platforms might not be affordable to all users, thus exacerbating digital divide (Wang et al, 2022), which is a socio-economic gap between those with access to high-quality, open-ended software and technology that promotes creative learning and those without such access (Bers et al, 2022). The unequal access to education would potentially place those students at disadvantage, since technology continues to advance, especially considering programming skills are currently regarded as crucial literacy skills worldwide (Bers et al, 2022). With the rise of metaverses over the last years, costs associated with high-end VR equipment have led to a new type of digital divide, commonly referred to as "metaverse divide" (Wang et al, 2022).

Research on low-cost alternatives is imperative for ensuring global access to technology. Low-cost VR HMDs for mobile devices, such as Google Cardboard (Google, 2023), offer smartphone users an opportunity to engage in interactive and immersive VR experiences with three degrees of freedom (3DoF), similarly to high-end VR HMDs (Powell et al, 2016; Google, 2023). 3DoF means only users' rotational movement can be tracked, i.e., only users' visual perspective can be replicated into the virtual world as they move in the real world (Jerald, 2016).

However, the use of low-cost mobile HMDs and mobile VR applications in a few lectures might represent the first steps towards a broad adoption of VR in education (Radianti et al, 2020). Among the different models of low-cost HMDs, some may bring together a Bluetooth joystick, which, with the press of a button, simulates touching the screen. On the other hand, Google Cardboard provides the lowest cost and enables users to interact with the application through a lever-shaped button, which touches a tactile tissue on the screen when pressed.

Regarding supporting the teaching and learning of programming using mobile immersive VR and low-cost HMDs: (i) OOPVR (Tanielu et al, 2019) enables users to create and manipulate virtual objects based on analogies with object-oriented programming (OOP) concepts, such as class, objects, methods, etc; (ii) Imikode (Sunday et al, 2022) provides quiz questions inside a virtual world also about basic OOP concepts, and (iii) iThinkSmart (Agbo et al, 2022) enables users to solve programming puzzles inside the virtual world (e.g., such as the river crossing puzzle and Tower of Hanoi) to support the learning of problem-solving and algorithmic thinking.

Whereas OOPVR provided no details of its design and development, Imikode and iThinkSmart were developed with Unity 3D and used Google Cardboard XR plugin and low-cost HMDs to promote an immersive VR experience to users. Sunday et al (2022) also considered Imikode a game; however, they provided no further explanations, highlighting the point made by Susi et al (2007), that game is a way to call the application.

Users well-rated OOPVR, Imikode, and iThinkSmart using questionnaires. They preferred VR-based learning over traditional textual learning and reported being able to retain the knowledge which is experienced through VR

(Tanielu et al, 2019; Sunday et al, 2022; Agbo et al, 2022). Specifically, OOPVR provided no details about data gathering (Tanielu et al, 2019), Imikode used its own questionnaire, entitled USE (Sunday et al, 2022), containing questions about usefulness, ease of use and learning, and satisfaction, and iThinkSmart also adopted its own questionnaires with CS undergraduate students for evaluating students' learning interests, attitudes and knowledge retention (Agbo et al, 2022).

2.4 Current study

This study aims to contribute to the existing body of research by enhancing the teaching and learning of programming among children and teenagers through the use of immersive and interactive VR. SSPOT-VR was developed from the integration of mobile immersive VR, BBP, and storytelling elements. We evaluated SSPOT-VR using a survey research method involving its target audience. SSPOT-VR offers K-12 students an affordable, immersive, and interactive experience for supporting their understanding of programming concepts through BBP, presenting an alternative to applications that require high-end equipment. Although we did not adopt game theories, we use storytelling elements, which enrich the user experience and motivation and support the interaction within an immersive virtual world (Ferguson et al, 2020).

While all the applications similar to SSPOT-VR offer immersive VR experiences, the main differences emerge regarding interactivity and affordability (Table 1). Whereas Cubely (Vincur et al, 2017), VR-OCKS (Segura et al, 2020), and VWorld (Jin et al, 2020) are highly immersive and interactive, their use require costly equipment. Since they are not compatible with mobile platforms, affordability and access are affected.

On the other hand, OOPVR (Tanielu et al, 2019), Imikode (Sunday et al, 2022), and iThinkSmart (Agbo et al, 2022) are mobile applications, making them comparatively more affordable. These applications maintain a degree of immersion; however, achieve it through a reduced level of interactivity, relying primarily on the use of analogies, metaphors, and the visual appeal of the VR environment. Therefore, among these studies, only SSPOT-VR and OOPVR (Tanielu et al, 2019), Imikode (Sunday et al, 2022), and iThinkSmart (Agbo et al, 2022) are accessible via mobile devices.

However, SSPOT-VR differs from them through the use of storytelling and BBP elements. Therefore, SSPOT-VR incorporates both storytelling and BBP to mobile immersive VR for enhancing the user experience. Finally, all of these applications support a diverse audience, ranging from children to adults, highlighting the varied landscape of the teaching and learning of programming and educational VR applications.

Table 1: Comparative analysis of SSPOT-VR and its related studies

Application	Immersive VR	Mobile	Storytelling	BBP	Equipment Cost	Target Audience
SSPOT-VR	Yes	Yes	Yes	Yes	Affordable	Children and teenagers
iThinkSmart (Agbo et al, 2022)	Yes	Yes	No	No	Affordable	Adults
Imikode (Sunday et al, 2022)	Yes	Yes	No	No	Affordable	Adults
OOPVR (Tanielu et al, 2019)	Yes	Yes	No	No	Affordable	Adults
VWorld (Jin et al, 2020)	Yes	No	No	Yes	High	Children
VR-OCKS (Segura et al, 2020)	Yes	No	No	Yes	High	Children, teenagers and adults
Cubely (Vincur et al, 2017)	Yes	No	No	Yes	High	Children, teenagers and adults

3 Method

3.1 Research model and hypotheses

This quantitative study is primarily aimed at detailing the development and evaluation of SSPOT-VR, a mobile immersive VR application designed to support the learning of programming concepts. The evaluation consists of three distinct surveys, labeled as S_1, S_2, and S_3, which were specifically designed for K-12 students who experienced SSPOT-VR during workshops conducted. S_1 and S_2 assessed user acceptance, whereas S_3 evaluated knowledge retention. The next section provides a user-centered description of SSPOT-VR and its development, usage, and narrative.

Since the main idea of survey methodology is to collect information from a group of people by sampling individuals from a large population, it should be executed through a systematic sequence of steps (Linaker et al, 2015). Therefore, the research design for these surveys was structured in accordance with the guidelines provided by Linaker et al (2015).

The independent variable, namely, use of SSPOT-VR was first defined. Regarding dependent ones, the first refers to the acceptance of SSPOT-VR, assessed by analyses of students' responses to a user acceptance questionnaire after their use of the application. The second dependent variable examined the retention of programming knowledge by assessing whether K-12 students can retain it following their interaction with SSPOT-VR. The evaluation relies on a comparison of students' responses to a quiz applied before and after their engagement with the application. We defined a null hypothesis and an alternative hypothesis for each of our research questions, which are as follows:

- *(R. Q. 1) Would K-12 students accept the use of our mobile immersive VR application for supporting their learning of programming concepts?*
  - **Null Hypothesis** (H_0): K-12 students will not accept the usage of our mobile immersive VR application for learning programming concepts.
  - **Alternative Hypothesis** (H_a): K-12 students will accept the usage of our mobile immersive VR application for learning programming concepts.

- *(R. Q. 2) Do K-12 students retain programming knowledge after using our application?*
  - **Null Hypothesis (H_0)**: K-12 students will not retain programming knowledge after using our mobile immersive VR application.
  - **Alternative Hypothesis (H_a)**: K-12 students will retain programming knowledge after using our mobile immersive VR application.

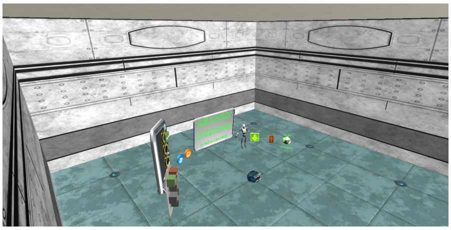
We hypothesize a positive relationship between the independent variable, which is the usage of SSPOT-VR, and both dependent variables, namely, the acceptance of the application and the retention of programming knowledge subsequent to its use. This implies that as students engage with SSPOT-VR, we anticipate both their acceptance of the application and their ability to retain programming knowledge through application use. Since the target population is usually extensive, the next step involved the selection of a representative sample population (Linaker et al, 2015).

3.1.1 SSPOT-VR: Space Station for Programming Training in Virtual Reality

Description and use of the application: SSPOT-VR, or Space Station for Programming Training in Virtual Reality, is an immersive VR and mobile application created to support children and teenagers in the teaching and learning of programming concepts. Its use is based on low-cost and widely accessible devices, requiring only a smartphone and low-cost HMD to provide an immersive learning experience. Users interact with the application by clicking or touching on the mobile device screen or through pressing a button of a joystick connected to the mobile device. The application has three levels for students actively experience several programming concepts using a BBP method based on the manipulation of cubes with different functions (Figure 1).

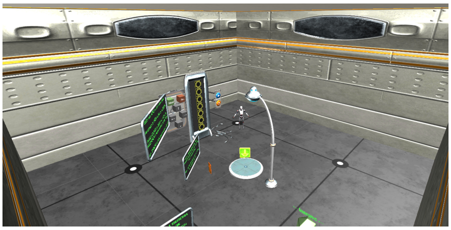
In the *Introduction Level*, all components essential for interaction within the application are exhibited (Figure 1a). They include programming cubes, a large canvas-like computer for cube placement, buttons for running and restarting the algorithm execution, movement anchors, and a button for switching between immersive and non-immersive modes. When users place the circular white crosshair over any of those objects, they receive a corresponding explanation on its function and usage (Figure 1b).

With an understanding of SSPOT-VR components and their use, users can progress to the *First Challenge Level* (Figure 1c), where they are provided with guidance through informative rectangular gray panels with textual instructions. Simultaneously, the movement anchors in the shape of green arrows become visible, enabling users to navigate through the scenario (Figure 1d). Those informative panels also provide step-by-step instructions on the creation of algorithms with available cubes, symbolizing directional movement commands and algorithm initiation (**begin**) and termination (**end**) (Figure 1e). Users are guided on the use of the algorithm run button, as well as with a

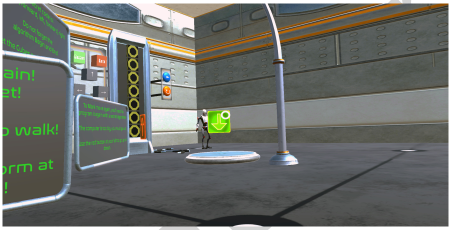


(a) Overhead view of the *Introduction Level* scene

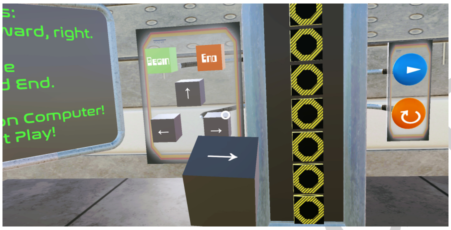
(b) One of users' first interaction with the *Introduction Level*



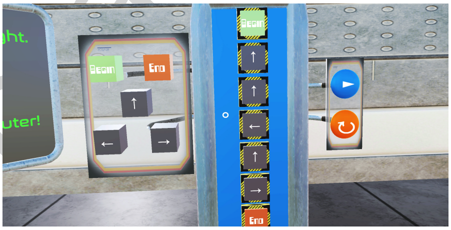
(c) Overhead view of the *First Challenge Level* scene



(d) Users' view of the *First Challenge Level*

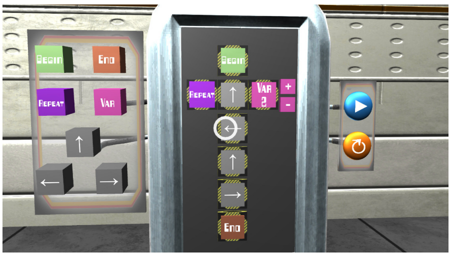


(e) Towards creating algorithms, students must click to hold cubes and click again to release them



(f) SSPOT-VR showing cubes were successfully compiled in the *First Challenge Level*

(g) Algorithm of the *Second Challenge Level* using a repeat command



(h) Algorithm of the *Third Challenge Level* with a repeat command and variable

Fig. 1: SSPOT-VR screenshots

separate restart button that resets the created algorithm. A blue or red background color signalizes, respectively, the success or failure of the compilation of the cube-created algorithm (Figure 1f).

In the *Second Challenge Level* (Figure 1g) and *Third Challenge Level* (Figure 1h), new programming concepts are introduced, presented by the *repeat* and *variable* cubes. All levels use as a basis the programming activity proposed in the *First Challenge Level*, which is to create an algorithm with cubes to make a robot accordingly walk.

Therefore, SSPOT-VR covers: (i) concept and composition of algorithms; (ii) use of a visual programming method (BBP); (iii) concept and use of a repetition structure, and (iv) concept and use of variables. Such programming concepts are considered essential for any novice in programming and can also be approached by children at an early age (Computer Science Teachers Association, 2017; ACM and IEEE, 2020). Although both *Introduction Level* and *First Challenge Level* have been used by over 100 children and teenagers, SSPOT-VR was designed to go over the concept of algorithm. Therefore, two different levels, namely *Second Challenge Level* and *Third Challenge Level*, are currently under development and have not been used by users yet.

Storyline: The *First Challenge Level* aims at connecting students with SSPOT-VR by using elements of storytelling and a programming method based on BBP. Among educators, storytelling is often considered as an advantageous teaching and learning strategy, especially regarding recall and retention of knowledge (Ferguson et al, 2020). When combined with immersive VR, storytelling enhances the potential of the application for providing a fully immersive and interactive experience (Ferguson et al, 2020). Therefore, the objective is to engage students in the programming task and assist interaction among students, the virtual world, and educators.

A robot called Mark (Figure 1b) is the main character and the space station tutor that guides students during their journey on the learning of programming concepts. In our plot, Mark's computer crashes and his memory has to be reset, which make him unable to move and proceed with the programming training. Therefore, in the *First Challenge Level*, students must rewrite Mark's memory by programming his computer with cubes.

SSPOT-VR enables students to create algorithms composed of cubes that represent the directions Mark can step, namely, *forward*, *left*, and *right*. Students must create their algorithms using the direction cubes and the *begin* and *end* cubes. If no compiling errors occur, the students can observe the output of the algorithm as Mark accordingly walks.

Design and development: Although SSPOT-VR can be experienced at a standard immersion degree, on the digital screen of mobile devices as non-immersive displays, i. e., a magic window to the virtual world, SSPOT-VR was created to be experienced at a higher degree of immersion (3DoF) using a low-cost HMDs. The creation of SSPOT-VR had been assisted by the free software development platforms and documentation provided by the Google Cardboard XR plugin[2] and the Unity 3D game engine[3][4]. As mentioned,

[2] https://github.com/googlevr/cardboard-xr-plugin
[3] https://unity.com
[4] https://assetstore.unity.com

SSPOT-VR was also built based on guidelines for teaching programming to children and teenagers (Computer Science Teachers Association, 2017).

3.2 Research procedure and sampling

Sampling involves the selection of a subset of observation units from a sampling frame (Linaker et al, 2015). In this study, we employed an accidental sampling approach, a non-probability technique that relies on the accessibility and convenience of the sample (Linaker et al, 2015). Accidental sampling involves selecting individuals or items for a sample based on their ease of access or occasional presence (Linaker et al, 2015) and is often chosen due to its efficiency and cost-effectiveness, especially when time and resources are limited (Linaker et al, 2015).

To recruit students for our study, we conducted free programming workshops, both online and in-classroom sessions, targeting K-12 students. We advertised these workshops to promote the teaching and learning of programming concepts, and students, irrespective of their prior programming knowledge, from public and private schools across Brazil, voluntarily registered for these sessions.

We intentionally included students from all age groups within the K-12 system to explore whether there is an age-related threshold where the usage of a mobile immersive VR application becomes impractical or less engaging for students. Additionally, we aimed to provide an engaging and novel activity for students, especially during the COVID-19 pandemic when many were exclusively attending online classes. Consequently, student selection was primarily based on their own interest in participating.

It is imperative to underscore the preservation of survey participants' privacy. All collected data were acquired with the informed consent of the guardians, and no sensitive personal information was gathered. Furthermore, all collected and anonymized data can be provided upon request.

Each workshop session followed a five-step procedure (Figure 2): (i) providing concise explanations about VR and BBP; (ii) only for S_3, students completed a pre-test survey in the form of a quiz to establish their baseline knowledge of programming concepts; (iii) students manually installed SSPOT-VR on their smartphones, and set up low-cost HMDs, if applicable; (iv) students had an unrestricted 40-minute experience with SSPOT-VR; and (v) following the experience, students completed web-based survey questionnaires.

As previously mentioned, SSPOT-VR was specifically designed to deliver an immersive experience utilizing a low-cost HMD. During the workshops, students engaged with SSPOT-VR in two distinct ways: (i) as an online activity (S_1 and S_3), where each student accessed the application directly on their mobile device display, and (ii) as an in-classroom activity (S_2), using low-cost HMDs and Bluetooth joysticks provided to them.

S_1 was conducted online, involving a sample of $n_1 = 124$ participants, while S_2 took place in a classroom setting with $n_2 = 16$ students. These two

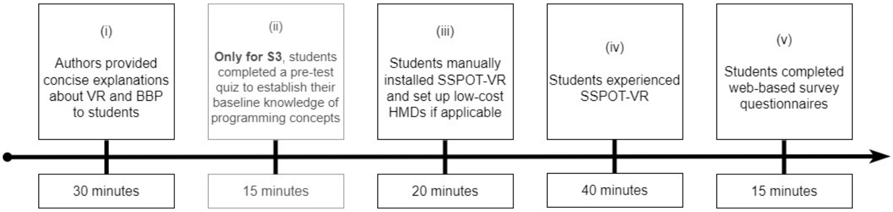


Fig. 2: Research procedure

groups experienced SSPOT-VR under different usage settings (S_1 being non-immersive and S_2 immersive). However, both groups subjected to the same data collection regarding the acceptance and usage of SSPOT-VR. The user and acceptance questionnaire was answered at the conclusion of the workshop session (Table 2).

S_3 was conducted online with a separate group of participants, consisting of $n_3 = 31$ students. Its goal was to assess changes in students' knowledge of programming concepts before and after using SSPOT-VR. For this purpose, we employed a questionnaire (pre-test) before the workshop started, establishing a baseline for students' knowledge. Following the workshop session, an identical questionnaire (post-test) was administered, featuring the same content but with different phrasing (Table 2).

Table 2: Surveys' structure

	Setting	SSPOT-VR Setup	User Acceptance or Knowledge Retention	Questionnaire Filling	Sample Size
S_1	Online	Non-immersive	User acceptance	After experiencing SSPOT-VR	124
S_2	In-classroom	Immersive	User acceptance	After experiencing SSPOT-VR	16
S_3	Online	Non-immersive	Knowledge retention	Before and after experiencing SSPOT-VR	31

Prior to reaching out to the sample population, it is necessary to design a data collection instrument, often in the form of a questionnaire. This process involves initially formulating the questionnaire and subsequently evaluating it on a subset of the sample population to assess its effectiveness (Linaker et al, 2015).

3.3 Data collection instruments

A survey instrument, typically in the form of a questionnaire, plays a crucial role in surveys and requires careful consideration. The quality of the questionnaire used directly influences the outcomes and conclusions of a survey (Linaker et al, 2015). Such questionnaires primarily include (Linaker et al, 2015): (i) descriptive questions for gathering demographic information; (ii) attitudinal questions aimed at eliciting the respondents' attitudes and opinions, and (iii) behavioral questions intended to assess differences or changes in the respondents' behaviors.

In this study, we employed only closed-ended questions within self-administered questionnaires, utilizing ordinal data collected via a 5-point Likert scale for both S_1 and S_2. For S_3, we gathered interval data from quiz questionnaires. Across all three surveys conducted, quantitative data were collected from K-12 students via web-based questionnaires.

To assess user acceptance of SSPOT-VR, we tailored 20 questions into a single questionnaire (Table 3), drawing from three established studies: (i) the Unified Theory of Acceptance and Use Of Technology (UTAUT) (Venkatesh et al, 2003), which aimed to elucidate users' intentions to use an unfamiliar application; (ii) the EGameFlow scale (Fu et al, 2009), which introduced a scale for gauging students' perceptions related to categories such as concentration, goal clarity, feedback, challenge, autonomy, immersion, social interaction, and knowledge improvement, and (iii) the path model proposed by Bourgonjon et al (2010), which provided items categorized by usefulness, ease of use, learning opportunities, experience, and preference.

To validate the tailored questionnaire administered to K-12 students via a 5-point Likert scale, we employed Cronbach's Alpha to assess its scale reliability or internal consistency levels (Hair et al, 2010). Cronbach's Alpha quantifies the correlation among responses in a questionnaire by assessing the average correlation of responses, indicating how closely related the set of items is as a whole (Landis and Koch, 1977). The calculated result for our data was Cronbach's Alpha = 0.691, affirming the questionnaire's reliability. As per Landis and Koch (1977), an Alpha value exceeding 0.6 is judged satisfactory for evaluating questionnaire reliability.

To determine if SSPOT-VR contributes to knowledge retention among K-12 students, we designed a quiz consisting of eight questions focused on the concept of an algorithm (Table 4). The concepts of algorithm is the programming concept that characterize the main activity of SSPOT-VR. For the questionnaire composition, we drew upon the programming education guidelines for children and teenagers outlined by the Computer Science Teachers Association (2017). These guidelines define the elements and characteristics of an algorithm. The eight questions remained consistent between the pre and post-test, although being differently phrased.

Pilot surveys were conducted using identical instruments. However, these pilot surveys involved a small number of participants from the target population. Conducting pilot surveys is essential to identify areas for improvement in survey instrumentation before implementing the full-scale study (Linaker et al, 2015). After the sample population has provided responses to the questionnaire, the researcher proceeds to analyze the collected data and draw conclusions based on this analysis (Linaker et al, 2015).

Table 3: Tailored user acceptance questionnaire

	Source	Description
1	Fu et al (2009)	I can remain concentrated in the application.
2	Fu et al (2009)	I am burdened with tasks that seem unrelated to learning programming.
3	Fu et al (2009)	Overall application's goals were presented clearly in the beginning.
4	Fu et al (2009)	Workload in the application is too much.
5	Fu et al (2009)	I receive immediate feedback on my actions.
6	Fu et al (2009)	I forget about time passing while using the application.
7	Fu et al (2009)	I become unaware of my surroundings while using the application.
8	Venkatesh et al (2003)	I could complete a job or task using the application if there was no one around to tell me what to do as I go.
9	Venkatesh et al (2003)	The application is somewhat intimidating to me, I hesitate to use it for fear of making mistakes.
10	Fu et al (2009)	I understood the basic idea of algorithms.
11	Fu et al (2009)	I want to know more about algorithms and programming.
12	Bourgonjon et al (2010)	Applications like this one offer opportunities to learn new things in a cool way.
13	Venkatesh et al (2003)	I like working with the application.
14	Venkatesh et al (2003)	I have the resources necessary to use the application.
15	Bourgonjon et al (2010)	Using applications like this one in the classroom would help me to achieve better grades.
16	Venkatesh et al (2003)	A specific person or group is available for assistance in case I have difficulties with the application.
17	Bourgonjon et al (2010)	I would know how to handle applications like this one in the classroom.
18	Bourgonjon et al (2010)	It would be easy for me to use applications like this one in the classroom.
19	Venkatesh et al (2003)	My family and professors think that I should use applications like this one.
20	Bourgonjon et al (2010)	If I had to vote, I would vote in favor of using applications like this one in the classroom.

Table 4: S_3 algorithm quiz questionnaire

	Question	Correct answer
1	What is an algorithm?	A description for how to solve a problem or how to do some task
2	What are the main pieces of an algorithm?	Input, output and processing
3	Can algorithms only be used on computers?	No
4	Complete the following sentence: "Algorithms are written as ... "	A list of steps
5	What is the orientation and direction which we write an algorithm?	From top to bottom (vertically)
6	Mark the true statements	Algorithms explain real life processes, Algorithms must have outputs
7	What is an algorithm for?	For describing how to solve a problem or how to do something
8	When can we use algorithms?	All the time for anyone

3.4 Data analysis

To quantitatively and descriptively analyze the data collected from the two surveys focused on students' acceptance of SSPOT-VR (S_1 and S_2), we computed a questionnaire score for each student, which ranged from zero to 100. Additionally, we compared the results from these two surveys to gain insights into the relationship between non-immersive (S_1) and immersive (S_2) SSPOT-VR usage. For the data collected from S_3, structured as a pre-test and post-test, we conducted both quantitative and descriptive analyses. We calculated each student's scores based on the eight quiz questions and used descriptive statistics to assess these scores before and after their interaction with SSPOT-VR.

Given that our data consisted of discrete scores, and considering the appropriateness of the Wilcoxon signed-rank test for analyzing repeated measures when the same subjects are evaluated under two different conditions (Linaker et al, 2015), we applied this statistical test to determine whether a significant difference existed between pre-test and post-test quiz scores in S_3 (Scheff, 2016). To assess whether our data from S_3 was suitable for non-parametric statistical testing (Scheff, 2016), we conducted the Shapiro-Wilk test. The results indicated that both pre-test and post-test samples were not normally distributed, as evidenced by $p_{\text{pre}} = 0.002353$ and $p_{\text{post}} = 0.001361$, where $p < 0.05$ in both cases.

3.5 Threats to validity

As this study is subject to threats to validity, we discuss the validity and reliability concerns based on a classification presented by Kitchenham and Pfleeger (2008) and Linaker et al (2015).

Face and content validity: To assess students' acceptance (S_1 and S_2), we designed a questionnaire consisting of 20 questions, drawing from insights from three established studies (Venkatesh et al, 2003; Fu et al, 2009; Bourgonjon et al, 2010). We ensured the reliability and internal consistency of this questionnaire using Cronbach's Alpha (Landis and Koch, 1977). Similarly, for our pre-test and post-test questionnaires (S_1), which evaluated students' retention of knowledge, we adhered to guidelines established for K-12 programming education by the Computer Science Teachers Association (2017). Although the eight questions were the same in both tests, they were phrased differently for the pre-test and post-test. Additionally, pilot studies were conducted using both questionnaires to ascertain their suitability for the target audience (Linaker et al, 2015). No issues or difficulties were reported during these pilot studies.

Criterion and construct validity: Despite the potential for selection bias associated with accidental sampling (Linaker et al, 2015), our study involved K-12 students from various public and private schools across different locations who voluntarily enrolled. This diverse sample allowed us to evaluate SSPOT-VR in authentic educational settings, collecting data from 171 students across nearly all grade levels, from kindergarten to 12th grade. Our analysis employed descriptive statistics, statistical tests, reliability assessments, and normality tests. All results were analyzed by all authors to avoid interpretation errors.

Reliability: Although workshop sessions occurred on different dates and one of the three surveys used a pre-test and post-test approach, the treatment of the 171 participants remained consistent. All students engaged with the application for 40 uninterrupted minutes after installing it on their smartphones and attending an introductory class. During the workshop sessions, students exclusively performed the tasks outlined in the workshop. Furthermore, we offered SSPOT-VR for usage on mobile device displays and low-cost

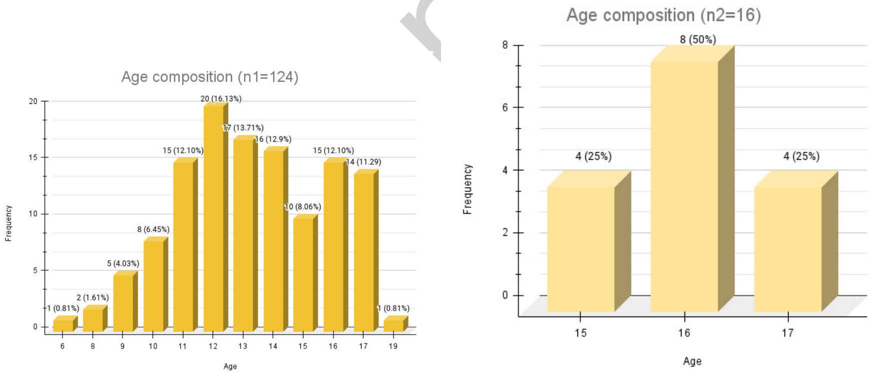
HMDs providing both levels of immersion. The results from the surveys formats were comparable, indicating that students' perceptions of the application remained consistent, regardless of the degree of immersion. To facilitate future work and international use, translating the virtual world instructions into different languages is the only requirement.

4 Results

A total of 171 K-12 students experienced SSPOT-VR throughout the workshop sessions offered. As we conducted three surveys: S_1 and S_2 aimed to collect data from students about user acceptance and use; and S_3 aimed to collect data about retention of programming knowledge, thus we present results according to these two perspectives.

4.1 Students' acceptance

In S_1, the age composition of 124 students ranged from 6 to 19 years old ($M_{\text{age}} = 13.26$). In S_2, the ages of 16 students ranged from 15 to 17 years old ($M_{\text{age}} = 16$) (Figure 3). As we mentioned, the goal of S_1 (online) and S_2 (in-classroom) was to identify the levels of students' individual acceptance of SSPOT-VR.

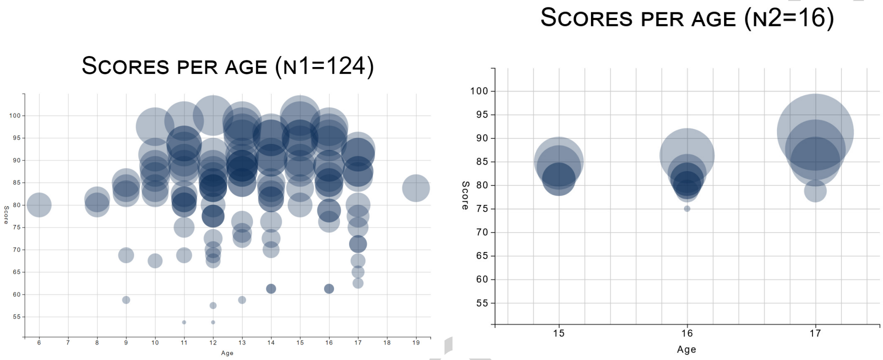


(a) Age composition in S_1 (b) Age composition in S_2

Fig. 3: Students' age composition in S_1 and S_2

In order to perform a descriptive analysis of the collected data, we calculated the questionnaire score for each student. Reducing the questionnaire answers to a score enable us to measure students' perceptions about SSPOT-VR in a scale from zero to 100. We assembled the 20 questions from our questionnaire into scores and distributed these scores to students according to their age.

The graphs (Figure 4) displaying the scores of the user acceptance questionnaire reveal a trend. Regardless of the age group or the specific educational context in which SSPOT-VR was utilized, the majority of scores fall within the 80-point range. Although some minor variations are observed, most scores are clustered in this range, suggesting a high level of scores, nearly approaching a score of 100 across all subjects. Specifically in S_1, the average for students' scores was 82.785, while the standard deviation was 10.788. There were 72 students that scored higher than the average and 51 that scored lower than the average, resulting that more than 58% of the sample of 124 students scored equal or higher than the average score of 82.785.



(a) Scores distribution in S_1 (b) Scores distribution in S_2

Fig. 4: Students' scores in S_1 and S_2

Similarly, the data collected during S_2 indicates that the average for students' scores is equal to 82.42, whereas the standard deviation is 3.94. On the other hand, there were 7 students in S_2 that scored higher than the average and 9 students that scored lower than the average. Resulting that 43.75% of students' scores for SSPOT-VR were above the average. Bearing in mind that most students' scores are close to the upper limit in both S_1 and S_2, it is possible to identify the tendency of students to accept the use of SSPOT-VR both as an online (S_1) and in-classroom activity (S_2).

4.2 Retention of knowledge

In S_3, the ages of 31 students ranged from 6 to 17 years old ($M_{\text{age}} = 12.74$) (Figure 5). S_3 aimed to identify retention of knowledge in students after they used SSPOT-VR. With respect to students' pre-test scores, measured on a scale from 0 to 8, it is evident that students exhibited a similar baseline level of knowledge regarding programming concepts. The average pre-test score was 3.9, with a standard deviation of 1.16. This average score of 3.9 represents only 48.75% correctness on our quiz (Figure 6). While in the post-test, which students also showed similar improvement of knowledge level, the average score of the students was equal to 6.55, with a standard deviation equal to 1.23. Now, the average score of 6.55 is equivalent to 81.88% of correct answers in the post-test quiz.

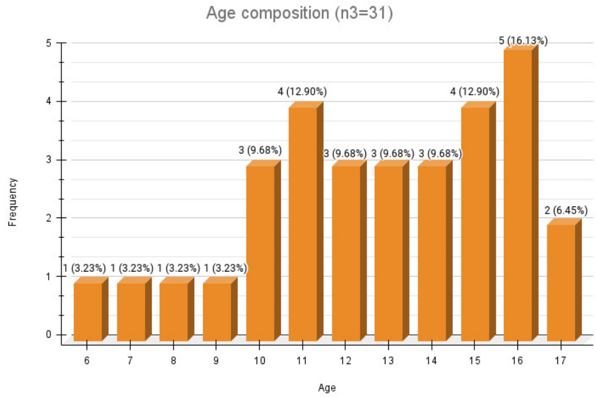


Fig. 5: Age composition for S_3 with $n_3 = 31$

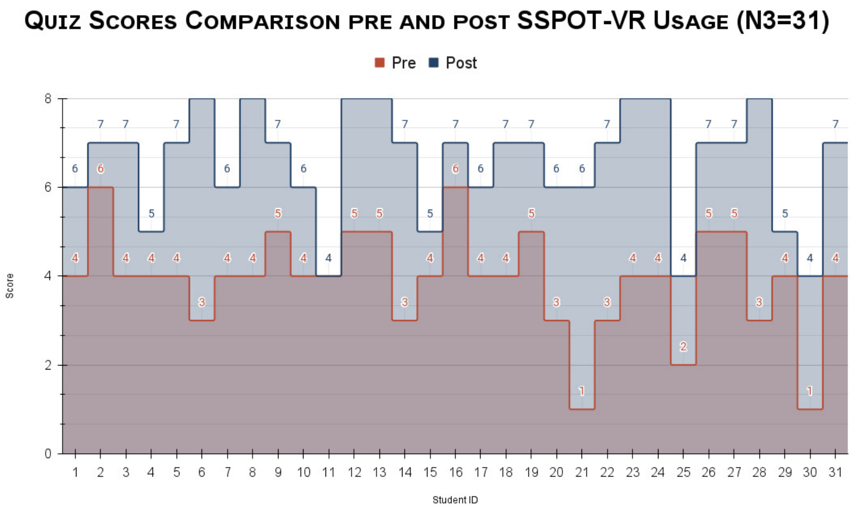


Fig. 6: S_3 quiz scores comparison

Comparing the results of pre-test and post-test, the 31 students who used SSPOT-VR had improvements in their score. The contrast between the low scores on the pre-test and the high scores on the post-test indicates that students are able to retain knowledge about the programming concepts seen in SSPOT-VR. It also shows that students are able to express this knowledge afterwards. Additionally, a Wilcoxon signed rank test revealed significant difference between the students' scores in the pre-test and post-test quiz, $Z = -4.782139$ and $p = 0.000002$. Since $p < 0.05$, the difference between pre and post-test are enough to be statistically significant (Scheff, 2016).

In this context, it is possible to highlight the retention of knowledge for Student 6 and Student 28, who obtained a score equal to 3 in the pre-test and a maximum score of 8 in the post-test, increasing their scores by 5 points, which is more than 60% of progress from the pre-test to the post-test. Similar performance improvements of approximately 40% are also observed in Student 8, Student 14, Student 21, Student 22, Student 23, and Student 24. Few students had scores close to the maximum score of 8 both in the pre-test and in the post-test, as is the case of Student 2 and Student 16. Low pre-test scores indicate that students had little or no prior knowledge about programming and algorithms.

5 Discussion

S_1 and S_3 collected different data from students (S_1 is about acceptance and S_3 is about retention of knowledge), both were online and therefore had broad student participation. On the other hand, S_2 was conducted to collect information on acceptance, however as a in-classroom activity, which reduced the number of students who participated due to COVID-19.

Analyzing the average scores for students and questions in S_1 and S_2, it is possible to identify in both cases a trend towards optimal values. This trend indicates perceptions of students about their usage of SSPOT-VR. The predominance of scores close to the optimal value reveals that a mobile application which is based on immersive VR, BBP, and storytelling to support the teaching and learning of programming concepts for children and teenagers is suitable for development and usage. Additionally, although students differently experienced SSPOT-VR in S_1 and S_2, the score distributions are similar. This indicates that students understand the benefits associated with SSPOT-VR under different setups of use.

Furthermore, we noticed that *knowledge improvement* potential and *preference* are key points for students acceptance and use of our application. *Knowledge improvement* potential is theoretically related to *usefulness* and defines that user acceptance depends on the identified possibilities of learning something (Bourgonjon et al, 2010). On the other hand, *preference* is subjective and involves users natural interests (Venkatesh et al, 2003). According to Venkatesh et al (2003) and Bourgonjon et al (2010), students naturally reveal *preference* for tools that capture their attention, address their mindsets, enable control, interaction, and experimentation with no limitations or usage problems.

Regarding S_3 (focused on the comparison of students programming concepts knowledge before and after using SSPOT-VR), the post-test scores with higher values for all students in contrast to the low pre-test scores indicates that students retained knowledge about the concept of algorithms through SSPOT-VR. The low average score of the pre-test quiz, equivalent to less than 50% of correctness, revealed that students had little knowledge of programming and algorithms before participating in the workshop session, in contrast to the post-test, whose results demonstrated an average of approximately 80% of correctness. Such results are consistent with the notion that immersive VR and HMDs support knowledge retention and recall (Krokos et al, 2019).

Compared to conventional instructional methods, immersive VR provides active learning experiences characterized by challenges, engagement, and playfulness, fostering situated knowledge and improving knowledge retention (Shi et al, 2022). Given that VR integrates various media formats and presents capabilities that surpass those of other media, we must assess the aspect of cost-effectiveness when incorporating VR applications into pedagogical frameworks (Pimentel et al, 2022).

In comparison to the studies of Vincur et al (2017), Tanielu et al (2019), Jin et al (2020), Segura et al (2020), Sunday et al (2022), and Agbo et al (2022), SSPOT-VR: (i) requires a smartphone and low-cost VR equipment to provide an educational experience and (ii) uses BBP and storytelling to work together with immersive VR to engage and motivate students on tasks, as well as to present a programming activity in a simplified manner. In addition, as stated by Radianti et al (2020), most studies focus on VR's development potential rather than reporting teaching and learning experiences or lessons learned, therefore, we provided data gathered from 171 K-12 students that used SSPOT-VR in order to analyze user acceptance and retention of knowledge.

The use of mobile devices with low-cost HMDs can also provide quality immersive educational experiences for K-12 students and benefit their retention of programming concepts, therefore, representing the first steps towards a broad adoption of VR in education (Radianti et al, 2020). On the other hand, limitations and difficulties are associated with SSPOT-VR usage. Although mobile VR expands access and reduces the cost of using an emerging technology for education purposes, there are limitations regarding interactions within the virtual world. Mobile VR enables users to explore 3DoF, in comparison with 6DoF that high-end HMDs usually provide. In other words, in comparison with some related studies (Vincur et al, 2017; Jin et al, 2020; Segura et al, 2020), SSPOT-VR does not enable students to freely walk inside the virtual world.

During the workshops, in real teaching and learning environments, one of the first limitations was that students occasionally did not have any smartphone available, or a smartphone compatible with VR applications, which requires accelerometer and gyroscope sensors. Considering that in some cases there are no smartphones available, a high-end VR setup is even further away. Some students also had difficulties in using HMD, which can at times offer slight comfort in its use. Moreover, its weight along with the weight of the smartphone may cause fatigue in students' necks, hampering their experience, making it important to measure the time of immersive experience.

Finally, each and every aspect of the application must be developed and deployed. Since SSPOT-VR is under development, students could experience the only two available SSPOT-VR levels, namely, *Introduction Level* and the *First Challenge Level*. Moreover, since SSPOT-VR is available only for Android mobile operating system, only students with smartphones compatible with the hardware and software required by SSPOT-VR were able to use the application.

6 Conclusions and Future Work

Studies focused on supporting the teaching and learning of programming have been adopting different technologies and educational strategies. We emphasize the usage of: (i) BBP, an approach with low access barrier in comparison to textual programming languages (Weintrop, 2019) and (ii) immersive VR, that brings immersion and presence to the learning process, inserting students in virtual worlds that stimulate their senses in the same way as the real world (Vincur et al, 2017). In the current study, we introduced SSPOT-VR, a mobile immersive VR application for supporting the teaching and learning of programming concepts that provides access to the community by jointly approaching BBP and storytelling with the usage of a low-cost VR HMD.

SSPOT-VR was experienced online or in-classroom by a considerable number of students ($n_1 = 124$, $n_2 = 16$ and $n_3 = 31$). Data were gathered in accordance with theories of user acceptance and retention of knowledge. Regarding our first research question, *"Would K-12 students accept the use of our mobile immersive VR application for supporting their learning of programming concepts?"*, the results point out that the application was well-accepted for use both online and as a in-classroom activity and would be adopted by children, teenagers, and young adults. They also reported it offered enjoyable learning activities and that they are capable to learn using the application. In addition, the quiz results showed students can retain knowledge experienced inside the virtual world. Therefore, regarding the second research question, *"Do K-12 students retain programming knowledge after using our application?"*, students retained knowledge about computer programming after using SSPOT-VR, such as the concept and purpose of algorithms.

According to the results, both null hypotheses were rejected and SSPOT-VR brought together immersive VR, BBP, and storytelling in a low-cost and mobile application for supporting the learning of programming concepts by K-12 students. These results were similar to those from high-end HMDs studies, conducted by Vincur et al (2017), Segura et al (2020), and Jin et al (2020). In contrast, SSPOT-VR adopted mobile immersive VR, low-cost HMDs, and storytelling. The results were also similar to those obtained by Tanielu et al (2019), Sunday et al (2022), and Agbo et al (2022), although SSPOT-VR adopted an active programming method based on BBP. Furthermore, we also evaluated our application with samples of the target audience.

SSPOT-VR can be adopted as a less digital divide solution, which is free, low-barrier of entry, and a technological resource to assist students and educators in the process of teaching and learning of programming concepts. For instance, when students are exposed to basic arithmetic operations in school, educators can present the basic definition of algorithms with a mobile VR application. This enables students to use and follow instructions from different sources to develop algorithms to solve different types of problems, using both natural languages and visual languages. The application can also be translated into any language for usage anywhere.

This study contributes to the growing literature on supporting the teaching and learning of programming concepts to children and teenagers, as well as with the report of the design, development, and usage of a low-cost mobile immersive VR application. By using SSPOT-VR students empower themselves with different computing and technology elements experience. The use of mobile devices, the immersive experience with VR, and a BBP method could support the presence and continuity of the fundamentals of programming in students' lives. This introduction to programming also enables them to better understand the world, to have autonomy, flexibility, resilience, and creativity.

Future work will not only extend students' access to the application, but make SSPOT-VR available for other mobile operating systems and platforms and in different languages. New features such as collaborative programming and scaffolding will be added and, regarding experimental tests, traditional teaching methods and other mobile applications that support the teaching and learning of programming will be compared with SSPOT-VR towards evaluations of preference and retention of knowledge. Free programming workshop sessions will continue to be offered to the community as SSPOT-VR progresses.

Data Availability Statements

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

ACM and IEEE (2020) Computing Curricula 2020: Paradigms for Global Computing Education. Association for Computing Machinery, New York, NY, USA

Agbo FJ, Oyelere SS, Suhonen J, et al (2022) Design, development, and evaluation of a virtual reality game-based application to support computational thinking. Educational technology research and development

Apiola M, Tedre M (2012) New perspectives on the pedagogy of programming in a developing country context. Computer Science Education

Bers MU, Strawhacker A, Sullivan A (2022) The state of the field of computational thinking in early childhood education. OECD Education Working Papers (274). https://doi.org/https://doi.org/https://doi.org/10.1787/3354387a-en

Bourgonjon J, Valcke M, Soetaert R, et al (2010) Students' perceptions about the use of video games in the classroom. Computers and Education

Chandramouli M, Zahraee M, Winer C (2014) A fun-learning approach to programming: An adaptive Virtual Reality (VR) platform to teach programming to engineering students. IEEE International Conference on Electro Information Technology

Computer Science Teachers Association (2017) CSTA K-12 Computer Science Standards

European Commission/EACEA/Eurydice (2019) Digital education at school in Europe. Luxembourg: Publications Office of the European Union

Ferguson C, van den Broek EL, van Oostendorp H (2020) On the role of interaction mode and story structure in virtual reality serious games. Computers and Education

Fu FL, Su RC, Yu SC (2009) EGameFlow: A scale to measure learners' enjoyment of e-learning games. Computers and Education

Google (2023) Google vr & ar. cardboard. URL https://arvr.google.com/cardboard/

Guo M, Ottenbreit-Leftwich A (2020) Exploring the K-12 computer science curriculum standards in the us. Proceedings of the 15th Workshop on Primary and Secondary Computing Education

Hair JF, C. BW, Babin B, et al (2010) Multivariate Data Analysis, 7th edn. Pearson Prentice Hall

Jerald J (2016) The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA

Jin Q, Liu Y, Yuan Y, et al (2020) VWorld: An immersive vr system for learning programming. Proceedings of the 2020 ACM Interaction Design and Children Conference: Extended Abstracts

Kalelioglu F (2015) A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior

Kitchenham BA, Pfleeger SL (2008) Personal opinion surveys. Guide to advanced empirical software engineering

Kong SC, Abelson H (2022) (Eds.). Computational thinking education in K-12: Artificial intelligence literacy and physical computing. Cambridge, Massachusetts: MIT Press

Krokos E, Plaisant C, Varshney A (2019) Virtual memory palaces: immersion aids recall. Virtual reality

Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics

Linaker J, Sulaman SM, Höst M, et al (2015) Guidelines for conducting surveys in software engineering. Lund University

Luxton-Reilly A, Simon, Albluwi I, et al (2018) Introductory programming: A systematic literature review. Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education

Milgram P, Takemura H, Utsumi A, et al (1995) Augmented reality: a class of displays on the reality-virtuality continuum. In: Das H (ed) Telemanipulator and Telepresence Technologies, International Society for Optics and Photonics, vol 2351. SPIE, pp 282 – 292, https://doi.org/10.1117/12.197321, URL https://doi.org/10.1117/12.197321

Mohamad SNH, Patel A, Latih R, et al (2011) Block-based programming approach: challenges and benefits. In: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp 1–5, https://doi.org/10.1109/ICEEI.2011.6021507

Osín L (1998) Computers in education in developing countries: Why and how? Education and Technology Team, Human Development Network, World Bank

Pears A, Seidman S, Malmi L, et al (2007) A survey of literature on the teaching of introductory programming. SIGCSE Bull 39(4):204–223. https://doi.org/10.1145/1345375.1345441, URL http://doi.acm.org/10.1145/1345375.1345441

Pellas N, Mystakidis S, Kazanidis I (2021) Immersive virtual reality in K-12 and higher education: A systematic review of the last decade scientific literature. Virtual Reality Journal

Pelletier K, McCormack M, Reeves J, et al (2022) EDUCAUSE Horizon Report: Teaching and learning edition. Educause

Pimentel D, Fauville G, Frazier K, et al (2022) An Introduction to Learning in the Metaverse. Meridian Treehouse

Powell W, Powell V, Brown P, et al (2016) Getting around in google cardboard – exploring navigation preferences with low-cost mobile vr. In: 2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR), https://doi.org/10.1109/WEVR.2016.7859536

Radianti J, Majchrzak TA, Fromm J, et al (2020) A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. Computers and Education

Rideout V, Robb MB (2020) The Common Sense census: Media use by kids age zero to eight. San Francisco, CA: Common Sense Media

Scheff SW (2016) Chapter 8 - Nonparametric Statistics. Fundamental Statistical Principles for the Neurobiologist

Segura RJ, del Pino FJ, Ogáyar CJ, et al (2020) Vr-ocks: A virtual reality game for learning the basic concepts of programming. Computer Applications in Engineering Education

Seralidou E, Douligeris C (2021) Learning programming by creating games through the use of structured activities in secondary education in greece. Education and Information Technologies https://doi.org/10.1007/s10639-020-10255-8

Sharma S, Ossuetta E (2017) Virtual Reality Instructional Modules in Education Based on Gaming Metaphor. Electronic Imaging 2017(3):11–18. https://doi.org/10.2352/ISSN.2470-1173.2017.3.ERVR-090

Shi A, Wang Y, Ding N (2022) The effect of game–based immersive virtual reality learning environment on learning outcomes: designing an intrinsic integrated educational game for pre–class learning. Interactive Learning Environments

Sigayret K, Tricot A, Blanc N (2022) Unplugged or plugged-in programming learning: A comparative experimental study. Computers and Education

Stigall J, Sharma S (2017) Virtual reality instructional modules for introductory programming courses. Proceedings of the 7th IEEE Integrated STEM Education Conference

Sukirman S, Ibharim LFM, Said CS, et al (2022) A strategy of learning computational thinking through game based in virtual reality: Systematic review and conceptual framework. Informatics in Education

Sunday K, Wong SY, Samson BO, et al (2022) Investigating the effect of imikode virtual reality game in enhancing object oriented programming concepts among university students in Nigeria. Education and Information Technologies

Susi T, Johannesson M, Backlund P (2007) Serious games: An overview. Institutionen för kommunikation och information

Tanielu T, 'Akau'ola R, Varoy E, et al (2019) Combining analogies and virtual reality for active and visual object-oriented programming. Proceedings of the ACM Conference on Global Computing Education

Venkatesh V, Morris M, Davis G, et al (2003) User acceptance of information technology: Toward a unified view. MIS Quarterly

Vincur J, Konopka M, Tvarozek J, et al (2017) Cubely: Virtual reality block-based programming environment. Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology

Vosinakis S, Koutsabasis P, Anastassakis G (2014) A platform for teaching logic programming using virtual worlds. Proceedings - IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014 pp 657–661. https://doi.org/10.1109/ICALT.2014.193

Wang M, Yu H, Bell Z, et al (2022) Constructing an edu-metaverse ecosystem: A new and innovative framework. IEEE Transactions on Learning Technologies

Weintrop D (2019) Block-based programming in computer science education. Communications of the ACM

Weintrop D, Wilensky U (2015) To block or not to block, that is the question: Students' perceptions of blocks-based programming. In: Proceedings of the 14th International Conference on Interaction Design and Children. ACM, New York, NY, USA, IDC '15, pp 199–208, https://doi.org/10.1145/2771839.2771860, URL http://doi.acm.org/10.1145/2771839.2771860

Weintrop D, Shepherd DC, Francis P, et al (2017) Blockly goes to work: Block-based programming for industrial robots. 2017 IEEE Blocks and Beyond Workshop (B B)

Wing JM (2011) Computational thinking: What and why? The Link. The magazine of Carnegie Mellon University's School of Computer Science, URL https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why, access in: 2021-03-07