VWorld: an Immersive VR System for Learning Programming

Qiao Jin
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55414, USA
jin00122@umn.edu

Yu Liu
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55414, USA
liu00885@umn.edu

Ye Yuan
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55414, USA
yuan0191@umn.edu

Lana Yarosh
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55414, USA
lana@umn.edu

Evan Suma Rosenberg
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55414, USA
suma@umn.edu

Interaction Design and Children (IDC '20 Extended Abstracts), June 21–24, 2020, London, United Kingdom
ACM 978-1-4503-8020-1/20/06.
https://doi.org/10.1145/3397617.3397843

Abstract
The growing development and commercialization of Virtual Reality (VR) allow more children to get access to this technology. VR features a new, more emotional relevant experience with a sense of presence and high interactivity. In this paper, we present VWorld, an immersive VR system designed to boost children's creativity and computational thinking skill. VWorld enables children to create their own virtual world by putting 3D objects on a miniature map, then explore the world and control the chosen objects by constructing program sequences. We present the design and implementation of VWorld system, with the design considerations of children in our VR environment, and conduct the preliminary evaluation and the future plan of the study.

Author Keywords
Virtual reality; programming languages; creation; children; computational thinking.

CCS Concepts
•Human-centered computing → Virtual reality; Interactive systems and tools; *Systems and tools for interaction design;* **•Social and professional topics → Children;**

Introduction
Programming learning has been proven to have many benefits for children, especially on fostering computational

thinking at an early age. However, Text-based programming platform usually shares a sharp learning curve, which makes it difficult to interact with children. Researchers have studied leveraging block-based languages for programming learning. It can simplify the understanding of complex syntax, enhance the learning effect cause it helps students focus on the problems by using a computational kit [7].

Virtual Reality (VR) with Head-mounted Displays (HMDs) is emerging in the market rapidly and provides new opportunities to optimize programming experience. Most HMDs contain stereophonic displays and tracking systems, that give the ability to present a virtual environment that resembles the real world, allowing users to immerse themselves in and interact with the virtual world. VR technologies also have the potential to facilitate the acquisition of higher-order thinking and problem-solving skills [2]. What's more, new technologies apply in education can contribute to increase motivation, engagement and critical thinking in students, and positively support knowledge transfer [8]. Consequently, VR shows its power as an innovative technology that can favor student-learning.

In this paper, we present VWorld, a VR system for learning programming. This system attempts to integrate the existing VR technologies naturally to achieve two major contributions: **1) Immersive experience for innovation and programming learning:** Leverage VR to create an immersive platform for children to create and learn programming, enabling children to be innovative and to create digital learning environments for their own use. **2) Customized nonlinear learning experience:** Engage children in the learning process by allowing children to create and customize their learning experience. Children can follow the non-linear path at different learning paces, for different learning styles.

Related Work

Technology for Children's Programming Learning

There are many block-based creation and programming system which support children creativity and computational thinking. For the programming tool which based on GUI (Graphic User Interface), Scratch [4] provides a programming learning platform for children to create and run their programs on the computer. The tools based on TUI (Tangible User Interface) [3, 5] use tangible language like wooden blocks or electronic slices to produce flow-of-control chains to teach children computer programming knowledge.

There are few works done in combining fully immersive VR and programming for children. The most related work to ours is VR-OCKS [7]. It is a VR system to teach the basic programming concepts. The program in VR-OCKS is created by choosing among a set of pre-established action blocks, which controls a virtual character to finish a maze. However, the system is task-driven in the given game background and users need to follow a linear learning instruction which limits creating activities.

This paper explores the approach of learning program in customized immersive environment, in which different learning styles can be catered for, allowing for greater detail and depth of learning.

Immersive VR Technology for children

Immersive VR relies on a 3D, stereoscopic head-tracker displays, enabling hand/body tracking and stereo sound. There are many researches have investigated how children engage and interact with a range of VR content on different headsets. The sample areas include but not limited to virtual field trips, language learning, skill training, art design, special education, and collaboration.

In the review of research and applications on 3D VR learning environments from Dalgarno and Lee [1], five learning affordances are provided that indicates VR learning environment that could enhance the spatial knowledge, realize the task which is impractical in the real world, increase motivation and engagement, improve transfer of knowledge and skills to real situations and more effective in collaboration learning. Dubit [8] investigated how children interact with VR content from 20 children aged 8 to 12 years old. The results also show high appeal and enjoyment when children trying VR in different applications.

With immersive, interactive and imagination advantages, VR has the potential to boost children's creativity and imagination. With particularly concern on combination of the VR technology and programming, we implement VWorld, hoping to introduce programming concepts to the children with more engagement and immersive, as well as enrich their creating and programming experiences with VR scenes.

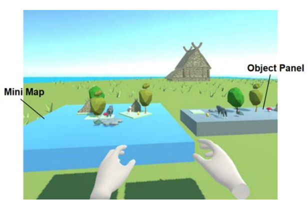

Figure 1: VWorld creation mode.

Design Considerations

One major challenge of VR for children is the size of the headset, both in terms of the optics and ergonomics. The headsets for adults usually has a higher interpupillary distance (IPD) than children. This means children have to adjust the IPD on the very low end of the spectrum, since high IPD causes eye strain and difficulty fusing the stereoscopic images and may results in double vision. Our system targets children at 13-15 years old, that their average IPD [6] meets the requirements for most adjustable headsets and more likely to wear the massive VR device. We also consider the potential negative effects (e.g., visual fatigue) of exposing children to near screen viewing, trying to make the whole game process much smoother with acceptable time duration.

Our primary objective was to create a VR system to be used as a platform to explore the benefits of VR-Auxiliary programming learning. Based on related works we have three considerations when designing our system:

- **Support programming learning:** The primary goal for our system should focus on supporting children's programming learning. Activities in the system should frame around such learning goal.
- **Provide immersive creating and learning experience:** The design of the virtual environment and interface should support an immersive VR experience. We should also work with technical constrains to achieve an immersive experience.
- **Customized learning environment and process:** Our system should allow children customize their learning environment and non-linear learning process. This allows our system to better engage children in programming learning.

Based on these considerations, we propose the *VWorld* system, which supports creating and programming learning in the VR-based 3D scenes with arranging virtual coding blocks. In the following section, we present how the concerns above are reflected in the system design.

VWorld System Design

VWorld is a system for children to learn programming in a customized immersive VR environment. There are two modes of the system: Creation Mode and Programming Mode. The creation mode allows children to construct their learning environment (the virtual world) from manipulating the objects. The programming mode let children learn programming knowledge and write programs to control certain objects in the virtual environment at their own learning pace. In other words, users can customize their learning

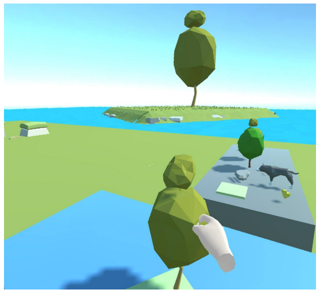

Figure 2: Grab the objects from object panel and put them on the mini map. A corresponding real-world size object will appear in the VWorld.

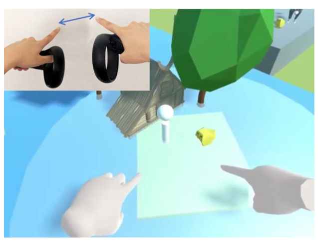

Figure 3: Use index finger of two hands to change the scale of map element.

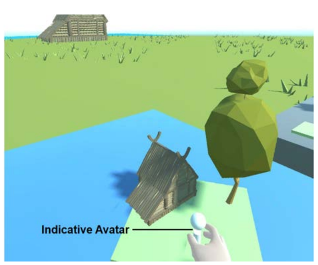

Figure 4: Relocate user's position by moving the indicative avatar in the mini map.

environments under creation mode, then explore and learn programming follow non-linear paths.

Creation Mode
VWorld system starts with the creation mode, which is a pre-defined spatial scope where children can create their own virtual environment from scratch, with provided virtual objects. In the creation mode, an object panel and mini map will show up in front of children for creation (Figure 1).

Objects Panel: The objects panel includes virtual objects that children can use to construct their virtual environment, such as land, trees, rocks and animals. Children can grab virtual objects from the objects panel and place them on the mini map with their right hand controller.

Mini Map: The mini map serves as an unifying metaphor for children to create the virtual world by allowing to manipulate the distant and large virtual objects without moving too much, also giving children a better sense of what the world is like. Once children place an object on the mini map, a corresponding real-world size object will be placed in the VWorld environment (Figure 2). The location of the real-world object is determined by the position of the virtual object on the mini map. The virtual objects placed on the mini map are scalable as well. To scale a virtual object, children can point at an object by putting their right hand in a pointing gesture and touch the object with two hands' finger tips (Figure 3). When an object is selected (indicated by a surrounded transparent light effect), children can scale the object by changing the distance between two index fingers or rotate the object by wheeling with two hands. Such interaction designs follow symmetric bimanual interaction technique in VR systems.

Navigation: To navigate in the virtual world, children can walk physically or use the joystick on the controller to move forward, backward, left or right. They can also use the indicative avatar on the mini map to help them relocate their position and navigate in the virtual environment, especially for long-distance movement (Figure 4). The objects panel and mini map will move with children once their location changes in the virtual environment.

Programming Mode
Once children leave the creation space, the programming panel will appear in the left hand indicating the programming mode (Figure 6). Children can go back to the creation mode and recall the object panel and mini map by pressing button in the controller (Figure 5). When the creation mode is triggered, the creation space will transfer to children's current location.

Programming Panel: The programming panel shows up on children's left hand when they entered programming mode. It contains two types of programming blocks: motion blocks and audio blocks. Motion blocks include instructions for programmable objects: move forward, turn left and turn right. Audio blocks include three types of sound effects. We also include visuals for the programming blocks so children can easily figure out the relative functions. Children can use these programming blocks from the panel to construct programs for programmable objects.

Programmable Object and Control Panel: Programmable objects are virtual objects in the environment that children can program with. Each programmable object has an associated control panel that has eight open slots for programming blocks. Children can program the objects by grabbing blocks from the programming panel and place them in the control panel to construct a code sequence (Figure 7a). The arrows between blocks provide visual indication on the logistics between programming blocks and the sequence of execution. We limit the number of programming slots and

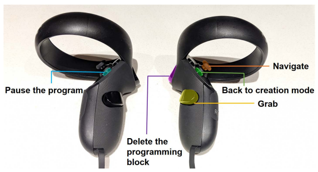

Figure 5: Controller input functions

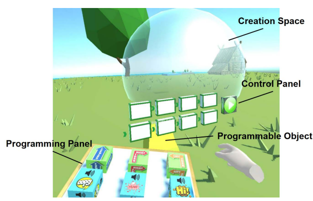

Figure 6: VWorld programming mode.

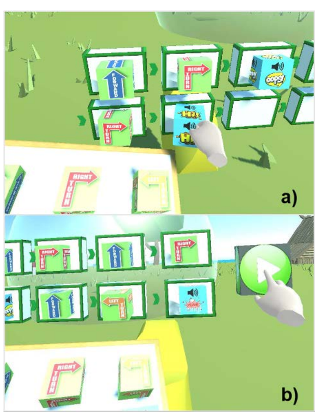

Figure 7: a) Put the programming block on the control panel; b) touch the run button to execute the program.

keep the logistics on the control panel simple to prevent potential cognitive overload for children.

Program Execution and Debugging: To execute the written programs for the objects, children can press the run button on the control panel (Figure 7b). The chosen programmable object will execute each programming blocks on the control panel in sequence. Children can follow the programmable object to observe the outcomes of written program. They can also pause the program to debug if the outcomes don't meet their expectation.

Implementation

We implemented the system on Unity3D for the Oculus Quest platform. We choose Oculus Quest as the targeted platform since it is an all-in-one VR gaming system that can track users' motion without cables. Also it's IPD range is adjustable in the range of 58-72mm. Low-poly models and low frame rate animation for virtual objects were used in VWorld, to make sure our VR system can run smoothly on a variety of configurations, and ensure an immersive learning experience.

We realized two 3D interaction techniques in VR environment: world-in-miniture (WIM) and symmetric bimanual interaction (SBI). WIM offers user a second dynamic viewport onto the virtual environment, acting as a single uniform metaphor for object selection, navigation, path planning, and visualization. Our system utilized WIM in creation mode, allowing children to manipulate the virtual world directly from the mini map. It also provides children with a visualization of the virtual environment that lets children plan their world easily and visually. SBI lets children use two hands to complete the task where each hand is assigned an identical function. Such technique is used to scale the virtual objects on the mini map. We chose SBI over asymmetric bimanual interaction as our system's input method, to reduce task difficulties and improve performance.

Pilot Testing and Ongoing Work

We have conducted pilot testing sessions with our current system prototype to gather feedback and iterate on system design. These pilot sessions focused on gathering feedback on tasks completion and ease of use from participants who were not experts in programming. We recruited 3 university students who have little programming knowledge and never use VR technology before. During the sessions, we asked participants to meet some requirements with our system (e.g., creating a virtual scene with at least 10 virtual objects, add programs to at least three virtual objects). All participants understood the interactions and could complete the tasks without help from researchers during the sessions, and engaged with our system prototype. Two of them use audio blocks to create a conversation plot between two programmable objects. Participants suggestions include more diverse objects and programming blocks, also prolonging the control panel for future improvement.

We plan to conduct more thorough evaluation sessions with targeted group of children (age 13-15). Besides task completion and ease of use, we will also evaluate perspectives include ease of creation, ease of programming, immersiveness, likeability and engagement. We plan on working with some local coding camps and programs to recruit potential participants for our evaluation. We also want to explore the opportunities for adding social components to our system, since VR has advantages for social support and collaborative learning. To understand these opportunities, we want to observe the usage of our system in co-located group settings, to understand interactions happen when using such systems which can be transferred into our system design. Then, we are going to add more functions and logic structures, allowing children to create more complex program and build more dynamic interactions among the programmable objects.

Finally, We interest in designing VR environment for younger children. Because young children have smaller heads, their IPDs are going to be lower than the minimum setting of the Quest. We believe it would be a good idea to prototype a custom headset for them using a 3D printed housing as our future work.

Conclusion

We present VWorld, a virtual reality programming system that supports children's programming learning. Our system allows children to customize their virtual environment and learning process by creating their own virtual world and writing programs to control objects in such world. Compared to previous works on designing and building systems for children to learn programming, our system design has two contributions: 1) leveraging VR technology to provide an immersive creating and programming learning experience and 2) allowing children to customize their learning environment and process. To evaluate our system design and iterate, We have conducted pilot testing with our system prototype. We are also planning on running further evaluation with children.

REFERENCES

[1] Barney Dalgarno and Mark JW Lee. 2010. What are the learning affordances of 3-D virtual environments? *British Journal of Educational Technology* 41, 1 (2010), 10–32.

[2] Hunter G Hoffman, David R Patterson, Jeff Magula, Gretchen J Carrougher, Karen Zeltzer, Stephen Dagadakis, and Sam R Sharar. 2004. Water-friendly virtual reality pain control during wound care. *Journal of clinical psychology* 60, 2 (2004), 189–195.

[3] Qiao Jin, Danli Wang, Xiaozhou Deng, Nan Zheng, and Steve Chiu. 2018. AR-Maze: A Tangible Programming Tool for Children Based on AR Technology. In *Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18)*. New York, NY, USA, 611–616. DOI: http://dx.doi.org/10.1145/3202185.3210784

[4] MIT Media Lab. n.d. Scratch. https://scratch.mit.edu/. (n.d.).

[5] Paulina Mun-Yee Lam, Carlo Ka-Ho Lai, Yeseul Tracy Choi, Brianna Jean Huxtable, Jan Rainier Castro, Andrew Hawryshkewich, and Carman Neustaedter. 2014. Loopo: A Tangible Programming Game for Kids. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. New York, NY, USA, 179–180. DOI: http://dx.doi.org/10.1145/2559206.2579484

[6] Colleen MacLachlan and Howard C Howland. 2002. Normal values and standard deviations for pupil diameter and interpupillary distance in subjects aged 1 month to 19 years. *Ophthalmic and Physiological Optics* 22, 3 (2002), 175–182.

[7] Rafael J Segura, Francisco J del Pino, Carlos J Ogáyar, and Antonio J Rueda. 2019. VR-OCKS: A virtual reality game for learning the basic concepts of programming. *Computer Applications in Engineering Education* (2019).

[8] Dylan Yamada-Rice, Faisal Mushtaq, Adam Woodgate, D Bosmans, A Douthwaite, I Douthwaite, W Harris, R Holt, D Kleeman, J Marsh, and others. 2017. Children and virtual reality: Emerging possibilities and challenges. (2017).