International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2022)
R. Kondo and T. Teo (Editors)

Demo

Programming in the Metaverse Era: VR Empowering the Educational Effects of Tangibles

Kohta Ebihara[1] and Johei Matsuoka[1] and Kiyohiko Hattori[1]

[1]Tokyo University of Technology

Abstract

In this presentation, we propose a new tangible programming interface utilizing Virtual Reality technologies, aiming to improve upon existing methods. Existing tangible programming methods have three major areas which can be improved upon. Limited set of commands, shape and textures not being employed much, and requiring specialized equipment. With VR, all three can be improved, with extra benefits of remote access capabilities and more. We will in the future, perform a through proof of concept experiment, as well as implement multiplayer focused features, and features to visualize debug sequences.

CCS Concepts

• ***Applied computing*** → ***Interactive learning environments;*** • ***Human-centered computing*** → *User interface programming;*

1. Introduction

In recent years, the field of XR and the Metaverse has grown significantly, and it is expected to keep growing in the following decade. It is not far in the future where many of us would regularuly connect to immersive 3D environments to perform daily activities. With such futures in sight and programmers being in ever higher demand, it is now very important to have more children interested in programming and the Metaverse. In this presentation we propose a new tangible programming interface utilizing Virtual Reality (hereinafter called VR) technologies.

There are many approaches in making programming more accecible to children. One such approach is tangible programming. As T. Sapounidis et al. [SSD16] lists, this usually involves physical blocks representing a certain function, which you would connect togeather to create your own programs. The resulting program could then, for example, control a real life robot.

There are many advantages in employing tangible interfaces, notably that it is believed to be easier to learn and use, but it still has a lot of room for improvement in areas such as the scope of creative freedom, and ease of adoption. To note a few: (1) Many systems only have inadequate set of commands and parameters [CB96]. (2) Physical qualities such as shape, and texture may contribute significantly towards learning with tangible interfaces, but has not been employed very much yet [XAM08]. And (3) Many systems require specialized equipment [SSD16], hampering their viability for widespread use.

Improvements to tangible programming interfaces should be possible by addressing each of the 3 points mentioned above. (1) Improving on the scope of creative freedom. With this, children who continuously play around with it have a chance to learn more about programming, before hitting the limits of what is capable with the said interface. (2) Employing more physical qualities to aid in recognition. With this, children could be offered visual and instinctive guides while programming with the interface. (3) Making extensive use of existing hardware and software in the design of the interface. With this, costs of initial adoption should go down, allowing for more widespread use of tangible programming interfaces in the field of education.

2. Aims

Currently existing tangible programming interfaces focus on having physical objects to interact with. Our proposed method brings this to the VR and Metaverse scenes, with the aim of overcoming the shortcomings of existing methods, as well as enhancing the programming experience with features unique to VR, such as infinite space to work with, and dynamic components.

Firstly, unlike physical tangibles, objects in VR can be free from spatial constraints, as well as having the ability to be created and destroyed at will. This allows for much greater freedom in how programs can be assembled. Thus making it possible to have a greater set of commands and parameters to be used. The ability to be freely created and destroyed, also allows easy copying and sharing of assembled programs as well.

Secondly, being fully digital allows dynamically adjusting shapes and sizes of components like is shown in figure1, which would be impractical with physical implementations. This allows for designs centered around shapes and textures, while at the same

DOI: 10.2312/egve.20221305

time being flexible enough to support many different use cases with a single type of command object.

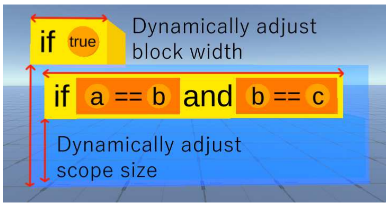

Figure 1: *Example of dynamically adjusting blocks*

Thirdly, being in a virtual space allows multiple participants to be in physically remote locations, while still retaining the ability to collaborate in full. This is unlike physical tangibles which requires all participants to be in the same physical space at the same time.

Lastly, this being one of many VR applications around the world, the equipment used in providing this VR programming experience can also be used for many other educational or non-educational uses. This versitility of equipments should significantly boost the chances of this proposed method to become widespread.

3. Features to be presented in the demo

3.1. Intuitive block manipulation

Users will be able to grab and manipulate parts of the code, like you would a physical object as shown in figure2. Along with the ability to freely create and destroy these objects. This demonstrates that VR applications can bring tangibility to programming, as well as show some features unique to VR and not achievable by traditional physical approaches.

Figure 2: *Manipulating code like you would real life objects*

3.2. A wider array of commands and sequenses

When engaging in programming, users will be able to choose and spawn what they want to use from a sufficiently wide array of commands and sequences. With this, users can experience greater freedom in creating their code, compared to traditional physical approaches to tangible programming.

3.3. Using tangible properties to prevent errors

When engaging in programming, users will come across scenes where they commit mistakes, but realise it immediately due to tangible properties given to the code. For example as in figure3, one might accidentaly try to set non integers as a loop count in "for" statements. This demonstrates how tangible programming interfaces are less prone to erros, and more friendly to beginners.

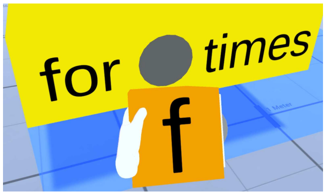

Figure 3: *A possible error prevented due to tangible properties*

4. Future development

As future development goals, apart from fleshing out our existing systems, we have three major areas of development we are interested in persuing. The first is to perform a through proof of concept experiment. The second is implementing proper multiplayer support, allowing for multiple individuals to share the same VR space and collaborate. The third is visualising debug sequences, allowing for deeper understandings of the underlying mechanics of programs, while still retaining the tangibility aspect.

Towards our first goal, we are planning a comparative experiment against other tangible programming interfaces, as well as some other non-tangible but widely used educational programming interfaces. Current plans include testing against KIBO and Scratch. Our initial experiments will be against more mature participants, although we do plan to test against children in the future.

Since collaboration and communication is said to have a key role in the effectiveness of learning, we aim to take advantage of that where possible. Towards our second goal, having features such as unifying codes made by different individuals, copying and handing codes to others for them to save or modify themselves, are planned.

While debugging is a major part of the programming experience, it is rarely focused on in beginner oriented programming systems. We believe it is highly beneficial to visualise this aspect, letting beginners gain insight on how programs get executed. Towards our third goal, to better visualise a program's execution flow, features such as showing animated characters that carry around values and inspect variables are planned. On top of that, since accessing and manipulating values inside arrays are a common stumbling block for beginners, a feature to display the values stored in an array in 3D space, are also planned.

References

[CB96] Cockburn A., Bryant A.: Do it this way: equal opportunity programming for kids. In *Proceedings Sixth Australian Conference on Computer-Human Interaction* (1996), pp. 246–251. doi:10.1109/OZCHI.1996.560017. 1

[SSD16] Sapounidis T., Stamelos I., Demetriadis S. N.: Tangible user interfaces for programming and education: A new field for innovation and entrepreneurship. *Innovation and Entrepreneurship in Education 2*, 1 (Dec 2016), 271–295. doi:10.1108/S2051-229520160000002016. 1

[XAM08] Xie L., Antle A. N., Motamedi N.: Are tangibles more fun? comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. TEI '08, Association for Computing Machinery, p. 191–198. URL: https://doi.org/10.1145/1347390.1347433, doi:10.1145/1347390.1347433. 1