

# Make OS Home: Home-Like Operating System in Virtual Reality

Wanwan Li

Department of Computer Science  
George Mason University  
Fairfax, Virginia, US  
wli17@gmu.edu



Fig. 1. Demo of Make OS Home: an example to show a digital transformation from a traditional desktop in Windows 10 operating system (a) towards a desktop in a home-like operating system via virtual reality (b) where the desktop on a 2D screen is transferred into a 3D desk model in VR.

**Abstract**—With the rapid development of Virtual Reality (VR) technologies in recent years, VR displays have become more popular, democratic, and attractive for users. Therefore, academic researchers are exploring the way to employ advanced VR technologies to achieve a smooth digital transfer from traditional user interfaces towards VR interfaces. In this paper, we present **Make OS Home: a digital transformation implementation prototype from a traditional 2D-screen-based operating system towards a home-like immersive 3D operating system in virtual reality.** We will present implementation details about the interactions between users, VR controllers, and the operating system to achieve such digital transformation on operating systems.

**Keywords**—digital transformation; operating system; virtual reality; virtual desktop.

## I. INTRODUCTION

With the rapid development of Virtual Reality (VR) technologies, there are more and more chances to transform the traditional user interfaces into immersive user interfaces in VR. Such evolution in digital transformation provides researchers with lots of opportunities to explore how different user interactions between 2D screens-based platforms and 3D immersive VR platforms affect user experiences. Therefore, many research works on user VR experiences have been proposed in diverse aspects of life such as VR aided shape modelings [1], VR education and training [2], VR entertainments [3], VR sports [4], VR performance arts [5], VR music [6], VR agriculture [7], VR business management [8], etc. As one of the most fundamental studies of computer science, operating system design is a very important topic

that can affect people's daily work and life. VR technologies provide infinite chances to change people's feelings about the operating system and computer-related working experiences. Therefore, it is a promising topic to show how the evolution of the operating systems can affect users' experiences and how to further extend such evolution through VR technologies.

During the 1980s, early ages of operating systems had no Graphical User Interfaces (GUI), such as Microsoft's MS-DOS [9]. Since 1985, Windows 1.0 first introduced GUI. Through various generations of the graphical Microsoft Windows operating systems [10]. We have Windows 10 as the most popular operating system today. As shown in Figure 2, the earliest GUIs of Windows operating systems mainly consisted of simple shapes, such as rectangles, rough vector icons, and monotonous colors themes, such as green and grays. Until Windows XP (2001), pictures were introduced into the main theme of OS. At the same time, icons were more photo-realistic than before which were drawn with simple vector graphics. Since Windows VISTA (2006), 3D effects were introduced when switching between windows using win+tab keys. This was a great evolution in Windows OS and gave users great visual effects. Later in Windows 7 (2009), while the 3D components were being kept, transparent icons were introduced. This was the peak masterpiece of work for windows with great visual effects and impressive user experiences. But unfortunately, due to the large amount of computation efforts that were put on graphical computing, the system was tending to be slow and wasting more electricity. Therefore, in Windows

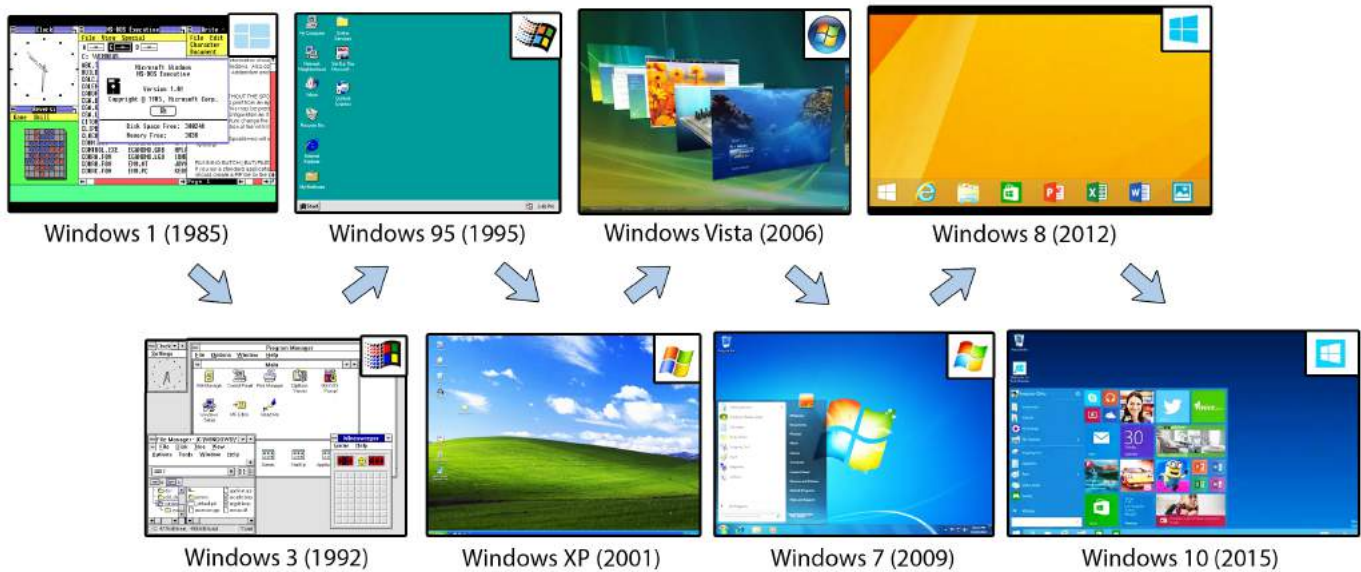


Fig. 2. History of windows operating systems from Windows 1 (1985) to WIndows 10 (2015).

8.1 (2012) and Windows 10 (2015), the 3D and transparent effects were removed. Then, system became faster and less power-demanding. The advantage of Windows 10 was the introduction of photo icons in the start menu. As we can see, within these 30 years, there were more than 10 different versions of Microsoft Windows operating systems. An obvious trend can be seen that Windows operating systems are becoming more and more user-friendly and environmental-friendly while achieving stronger visual performance and optimizing the system's efficiency.

As advanced technologies in virtual reality are emerging, a notion of "working in VR" is appearing as well. Lots of recent works are focused on moving workspace from the real world to virtual space in VR. For example, as shown in Figure 3 (a), *immersed* [11] presents a novel VR interface for creating private working space, supporting multiple virtual monitors without adding additional hardware. Also, *immersed* is portable as it supports mobile and wireless headsets so that it can be taken anywhere. As shown in Figure 3 (b), Virtual Desktop [12] is an application developed for the Oculus Rift / Rift S, HTC Vive, Valve Index, and WMR (Windows Mixed Reality) headsets that can visualize the screen of a computer as a virtual screen in VR. As shown in Figure 3 (c), VR-OS [13] is an operating system that replaces the monitor with a head-mounted display in an immersive virtual environment. It provides the virtual mouse and virtual keyboard that can match with the real ones on the desk. VR-OS allows powerful workflows in virtual reality platforms. As shown in Figure 3 (d), Falkengren et al. [14] presented a concept VR OS User Interface (UI) that was developed with basic OS features, such as viewing files and opening programs. As the closest work to ours, Falkengren et al. [14] developed a prototype virtual operating system that considers 3D icons and file browsers through VR operations, such as dragging and dropping. But

unfortunately, all of those proposed works are actually pseudo-3D visualizations where all displays, icons and browsers are visualized through flat or curved virtual 3D surfaces which look like 2D plat or curved screens hanging in the 3D virtual environment. This visual effect lacks immersiveness.

Therefore, given these observations, we propose Make OS Home: a home-like operating system in virtual reality. In Make OS Home, all 3D icons can be created, dragged, and clicked in full 3D space that is not restricted to a flat surface of the pseudo-3D space. In this interface, users are able to manipulate 3D icons as if manipulating 3D things at home. During each operation on 3D icons in the virtual environment, operating system APIs will be invoked simultaneously to modify the file system within the operating system. At the same time, we provide a screen of the operating system through a virtual monitor, so that users can identify the operations that happened in the operating system while they are manipulating the 3D icons using VR controllers. Our work reveals a novel aspect and a new level of control degree for operating system user interactions and GUI, which opens a new dimension of how to interact with the operating system in an entirely immersive way. In the future, by regarding our interactive interface of Make OS Home as a prototype, extending our interfaces with other types of operating system operations, such as hardware settings, network settings, visualizing image and video within 3D volumes, and so forth, the GUI of the future operating system will be entirely transferred in virtual reality. As shown in Figure 1, a desktop of standard Windows 10 operating system is shown in (a); Through the implementation presented in this paper, the desktop will be transformed into a virtual desktop in home-like environment through virtual reality as shown in (b). The full demo of the video can be accessed at [15] which shows manipulating folders and documents with our devised VR interface.

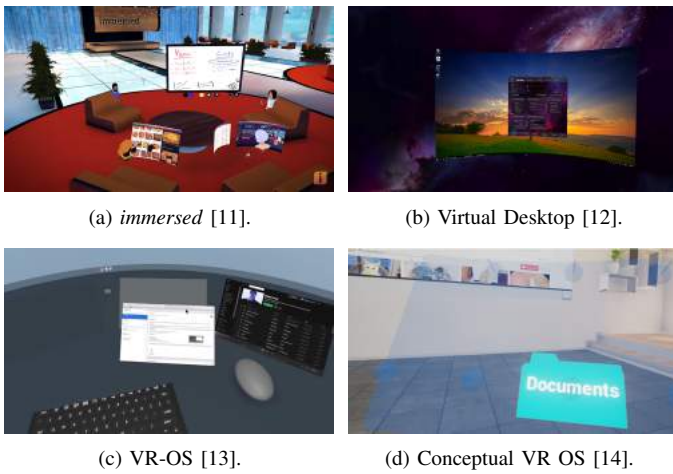


Fig. 3. Examples of related works on virtual desktops or conceptual virtual reality operating systems.

## II. IMPLEMENTATION DETAILS

In this section, we present the implementation details of Make OS Home. In our implementation, we are mainly focusing on how to connect the file systems in an operating system with 3D virtual objects as manipulable icons in virtual reality and how to display such files system through to two views (1) a virtual monitor that records the screen of the operating system and (2) the user's 3D view in a virtual environment. Our work extends an operating system with VR operations through the proposed two views. We have implemented the Make OS Home system using Unity 3D with the 2019 version. We have implemented the VR interactive interface using the Steam VR 2.0 plugin. The hardware configurations include Intel Core i5 CPU, 32GB DDR4 RAM, and NVIDIA GeForce GTX 1650 4GB GDDR6 Graphics Card. The VR program is configured on Oculus Quest 2.0.

**File System.** Given our purpose to connect the file system with VR objects of 3D icons, we propose an extended version of the data structure to represent the file system. In our implementation, we mainly consider two types of files (1) a folder and (2) a document. A folder can have an arbitrary number of subfolders, a document has no subdocument. Therefore, a file system is typically a tree data structure. Let us call a node representing a folder or a document a file node, then, the file system is the tree consisting of a root node and its descendants. In our case, the root node is the desktop folder, and other subfolders on the desktop are its children. As shown in Figure 4, a detailed representation of the file node is presented. We store the file information, such as file name and file type (folder or document) within a class called FileValue. FileValue also contains the icon object which is a game object in Unity and the icon behaviour which is a C# class defining the interactions between icon and VR controllers. FileValue is always paired with a string called FileKey. The reason is we have maintained all file nodes not only in a file tree, but also in a Hash table for easy access.

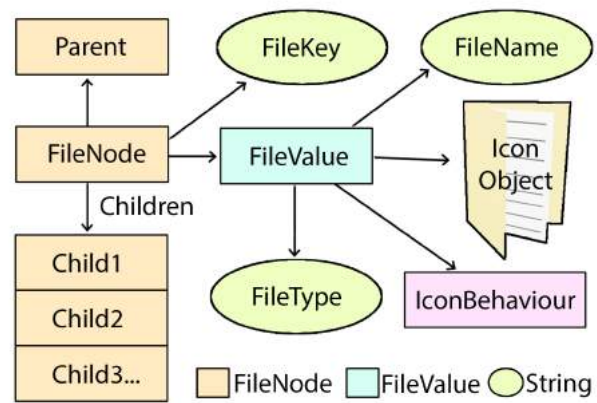


Fig. 4. Data structure of a file node which is the basic component of a file system in our Make OS Home interface. Different colors represents different C# classes in Unity.

**User Interactions.** We have implemented several different file operations, such as creating file icons, dragging file icons, scaling file icons, and delete file icons. Each operation is followed by a selection detection, which is achieved by calculating a line intersection between the lightsaber shooting out from the controller and the selected object. After the selected object is determined, we will check whether there is a click on the right controller. If so, we will react according to which icon is selected: if the selected icon is the create folder or create document icon as shown in the figure above where there are green plus signs attached, then, we will call the system API to create that file at the current path. Otherwise, if the selected icon is a subfolder of the current folder, we will switch to that subfolder where all of the children files or documents in that subfolder will appear.

Perhaps, the selected icon is a document in the current folder. In this case, we will call a system API to run a notepad app to open such text file. Or, if the selected file icon is the current folder itself, then, we will hide all of the children subfolders and switch to its parent folder. If there are no clicks from the grab button on the right controller, then, we will detect whether the pinch button is pressed. If so, a dragging operation is applied on that file icon. If that file icon is dragged into a recycle bin icon as shown in the figure above, then, we will call a system API to remove that folder or file recursively. This operation can be achieved by adding a specially tagged collider onto the recycle bin so that every game object that hit the recycle bin will disappear. During the dragging operation, the user can also push the joystick up with the thumb to zoom in the 3D icon or push the thumb down to zoom out the 3D icon in the virtual environment.

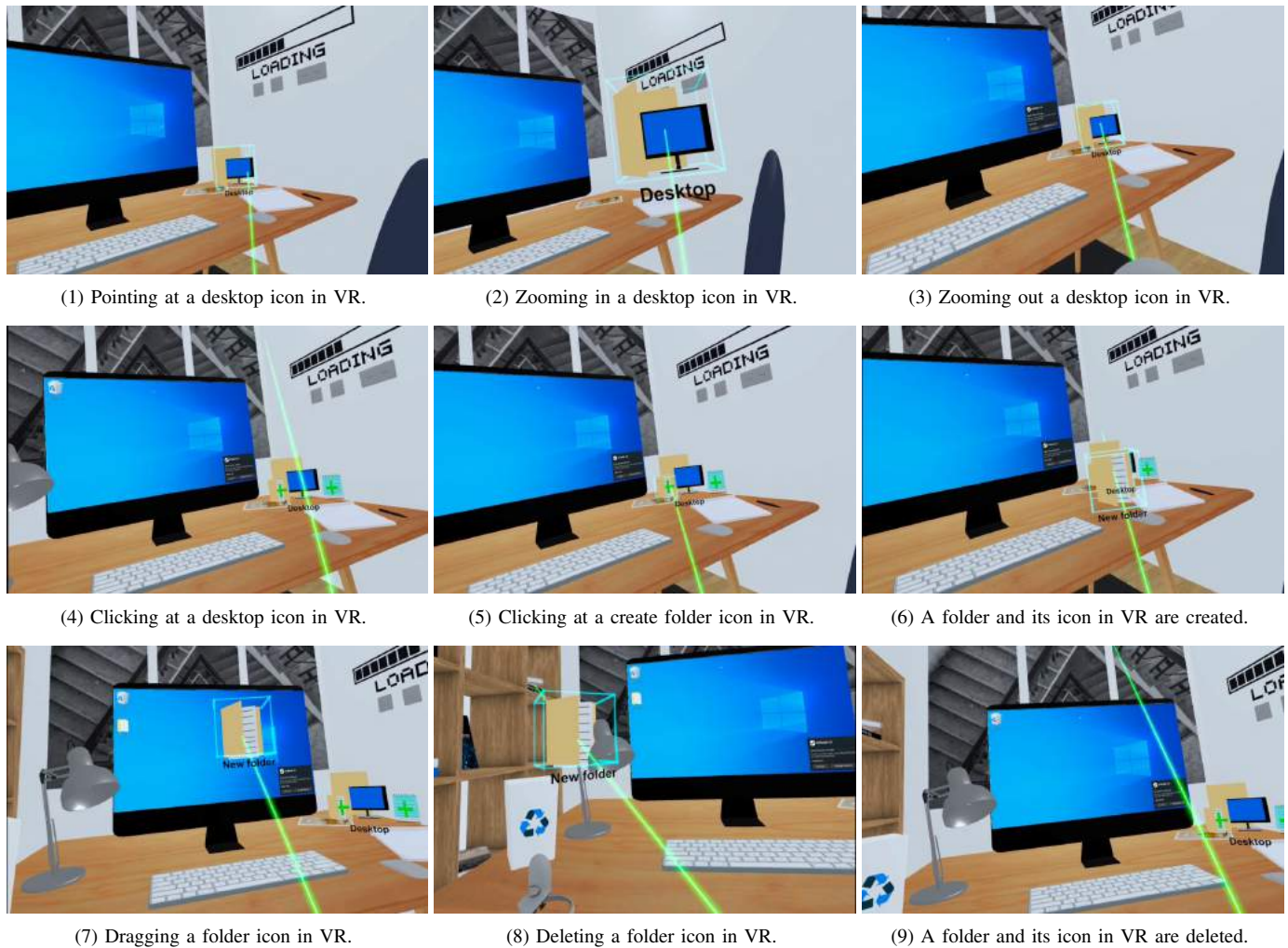


Fig. 5. A running example of using the Make OS Home system (Part I).

### III. EXPERIMENTAL RESULTS

In this section, we give a frame-by-frame demo of a running example to illustrate our proposed system. As shown in Figure 5, subfigures (1)-(3) demonstrate a process to zoom in and zoom out the desktop icon in VR. Users first point their controllers towards the desktop icon in VR. Then, the shining bounding box will pop up (1). When holding the pinch button down, users can push the joystick on the controller up and down to zoom in the desktop icon (2) and zoom out the desktop icon (3). Subfigures (4)-(9) demonstrate a process to create a folder and delete a folder. After a click at the desktop icon with the grab button on the controller, the create folder/document icon will pop up (4). After another click at the create folder icon (5), a new folder icon will be created randomly between the controller's position and the desktop icon position using a randomized linear interpolation (6). Then, users can drag the folder icon in mid-air (7) until it hits the recycle bin icon (8), the folder icon disappears (9). Note that the 2D folder icon on the desktop in the virtual screen appears and disappears together with the 3D folder icon

simultaneously, as the system API is called to create and delete the file along with the user's VR operation automatically. Also, the screen recorder API will update the VR monitor screen's texture to reflect the changes made in the operating system.

There is another testing case in this experiment as shown in Figure 6. This time, the user creates a folder (1) but does not drag the folder to the recycle bin, instead, somewhere else nearby (2). When the user clicks on the folder icon, the user will navigate into that folder (3). If users click the create icon again, the files will be created under that folder. (4)-(6) show the process to create and drag a folder while (7)-(9) are creating and dragging a text document.

### IV. CONCLUSION AND FUTURE WORK

In this paper, we present Make OS Home: a digital transformation implementation prototype from a traditional 2D-screen-based operating system towards a home-like immersive 3D operating system in virtual reality. We demonstrate the functionalities of our proposed interface in the experiments. Through Make OS Home, users can successfully create, navigate, and delete the files when operating 3D icons in VR.

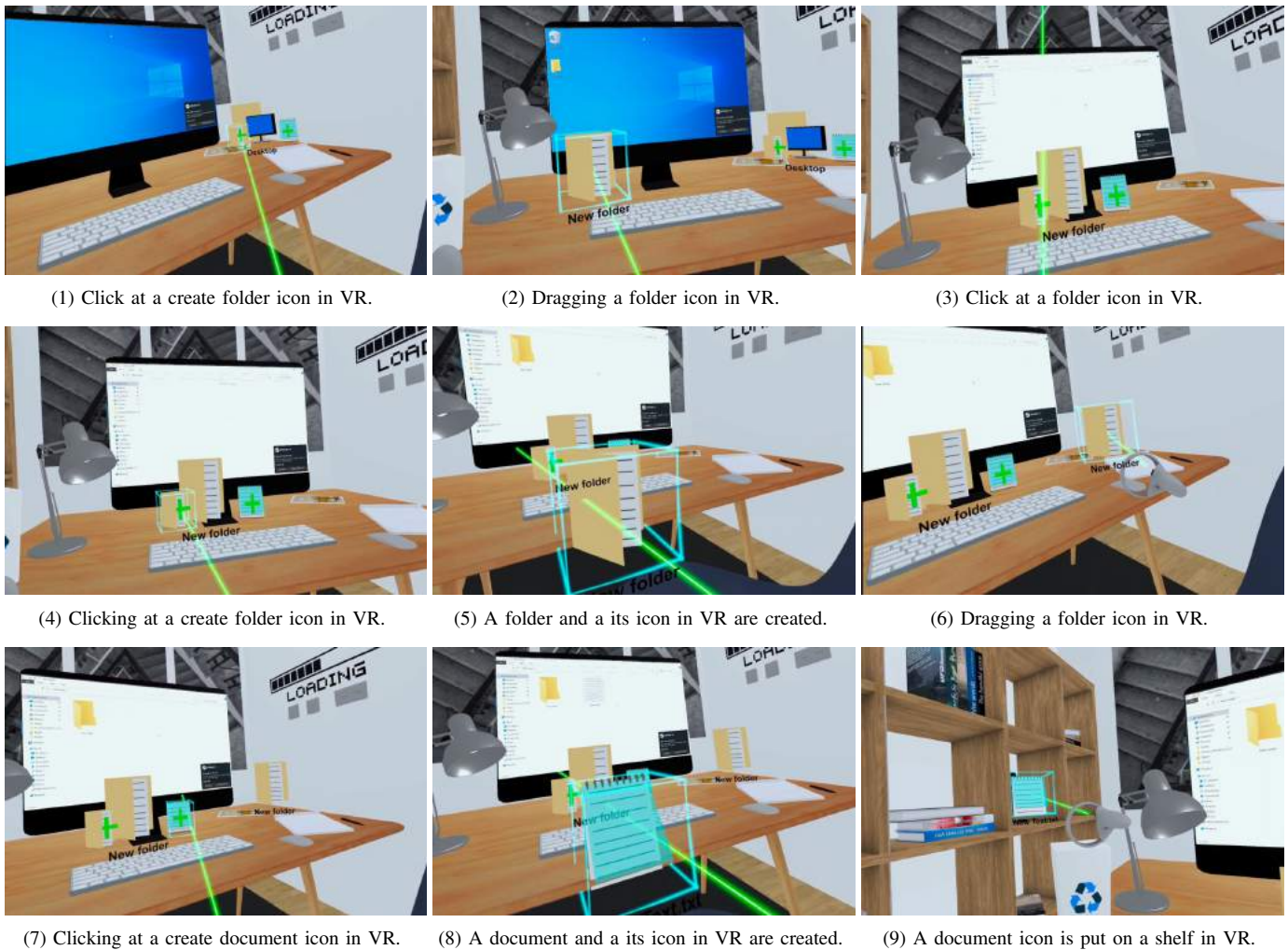


Fig. 6. A running example of using the Make OS Home system (Part II).

In future work, we will extend our proposed system by adding new features, such as moving folders into another through dragging the icons similar to the delete operation, adding a start menu in VR to trigger more apps other than the notepad app, adding hardware configuration settings in VR, adding network connections in VR, and rendering 3D images in VR, etc. In the end, we intend to build an entirely new VR OS without relying on another existing OS, such as Windows. We believe our work can open the new dimension to think about how the future operating systems will look like on a VR platform.

#### REFERENCES

- [1] T. H. Dani and R. Gadh, "Creation of concept shape designs via a virtual reality interface," *Computer-Aided Design*, vol. 29, no. 8, pp. 555–563, 1997.
- [2] J. Psocka, "Immersive training systems: Virtual reality and education and training," *Instructional science*, vol. 23, no. 5-6, pp. 405–431, 1995.
- [3] K. Lancaster, *Warlocks and warpdrive: Contemporary fantasy entertainments with interactive and virtual environments*. McFarland, 1999.
- [4] K. Ahir, K. Govani, R. Gajera, and M. Shah, "Application on virtual reality for enhanced education learning, military training and sports," *Augmented Human Research*, vol. 5, no. 1, pp. 1–9, 2020.
- [5] S. Smith, "Dance performance and virtual reality: an investigation of current practice and a suggested tool for analysis," *International Journal of Performance Arts and Digital Media*, vol. 14, no. 2, pp. 199–214, 2018.
- [6] S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson, and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, vol. 40, no. 3, pp. 22–40, 2016.
- [7] F. Yu, J.-F. Zhang, Y. Zhao, J.-C. Zhao, C. Tan, and R.-P. Luan, "The research and application of virtual reality (vr) technology in agriculture science," in *International Conference on Computer and Computing Technologies in Agriculture*. Springer, 2009, pp. 546–550.
- [8] H. Patel and R. Cardinali, "Virtual reality technology in business," *Management Decision*, 1994.
- [9] T. Paterson, "An inside look at ms-dos." *Byte*, vol. 8, no. 6, p. 230, 1983.
- [10] T. Carpenter, *Microsoft Windows Operating System Essentials*. John Wiley & Sons, 2011.
- [11] "immersed : Work faster in vr than in real life, Immersed Inc. All rights reserved," <https://immersed.com/>, 2021.
- [12] "Virtual desktop, Valve Corporation. All rights reserved." [https://store.steampowered.com/app/382110/Virtual\\_Desktop/](https://store.steampowered.com/app/382110/Virtual_Desktop/), 2021.
- [13] M. Krenn, "Vr-os," <https://matthaeuskrenn.com/vr-os/>, 2019.
- [14] W. Falkengren, M. Lilja, A. Solback, A. Strandman, J. Swanberg, and A. Höglund, "Virtual reality operating system user interface," B.S. thesis, 2017.
- [15] W. Li, "Make os home: Home-like operating system in virtual reality," <https://youtu.be/OyyF0zltUZ0>, Nov 2021.