

A 3D Window Management System Using Viewpoint Manipulation

Rin Ohsugi¹ Yuki Yamashita² Miki Katsuragi² Yuji Sasaki³

Abstract: In window management systems of Windows and macOS, simultaneous display of multiple windows and switching displayed windows are implemented as separate operations. In this paper, we propose Air Manager, a window management system that can perform both functions with a single operation using a viewpoint shift. Through this system, we have devised an operation method combining the simultaneous display of multiple objects and switching displayed objects, which are the two separate operations in conventional window management systems. We found a possibility to solve the problem of window management systems of Windows and macOS that require a separate operation each time to change the way windows are displayed between single-tasking and multi-tasking.

1. Introduction

As the performance of personal computers has improved, it has become common for users to launch many applications simultaneously and work with them, either displaying them simultaneously or moving back and forth between them. When users launch many applications, there are several ways to display and control the multiple windows they use to work with them. Each of these operations is based on one of the following two concepts. One is to display multiple objects at the same time. The other is to switch objects to be displayed.

The tiled window method is a conventional display method in window management systems that employs the concept of simultaneous display and can display multiple windows simultaneously. In this tiled window method, windows are laid out on the desktop with no overlap. Thus, users can browse multiple windows simultaneously. The overlapping window method employs the concept of switching to enable changing the windows to be displayed. In this overlapping window method, windows can be arranged with overlapping on a desktop. This allows more windows to be placed on a desktop. Windows hidden in the back can be brought to the front by clicking on them, and the windows displayed on the screen can be switched. In this method, the size and position of the windows can also be adjusted. Thus, it is possible to arrange multiple windows without overlapping each other and display them simultaneously. However, it requires many operations since window switching and window size and position adjustment are two different operations. Comparing the tiled window method and the overlapping window method, it has been demonstrated that the tiled window method works more efficiently in some situations due to the difference in the number of operations [1].

The concept of switching is also applied to desktops. Virtual desktops are designed to duplicate desktop screens to expand the workspace to switch tasks between them.

In Windows and macOS, operation methods based on the concept of simultaneous display and those based on the concept of switching are implemented separately. For example, switching between single windows on the desktop is mainly performed using the overlapping window method, while simultaneous display of multiple windows during multi-tasking is performed using the tiled window method [2][3]. Virtual desktops are implemented to switch multiple windows together [4][5]. However, since the operation method for simultaneously displaying windows and for switching between displayed windows are implemented differently, additional steps are required to switch them. For example, suppose that when changing windows on the desktop and using several windows, users desire to refer to them simultaneously instead of switching between them. In this case, a separate process of dividing the screen or adjusting the window size for simultaneous reference would be necessary. Thus, when users want to work with multiple windows, they have to choose either simultaneous display or switching. At this point, users must switch between different operations each time.

This paper proposes a new concept in window management systems that eliminates the need to select whether to display windows simultaneously or to switch between them. The concept is viewpoint moving, which combines displaying multiple objects simultaneously and switching what is displayed. In this paper, we propose a window management system, Air Manager, that places all windows in a single desktop space and enables both to display simultaneously and to switch them by a single viewpoint control.

2. Previous Work

A number of previous research on window management systems involved approaches to allow several windows to be displayed simultaneously while allowing them to be switched collectively. There are several methods to achieve this. One of them is to apply the idea of switching to desktops, whereby desktops are duplicated and can be switched between them. The other is allowing multiple windows to be switched once within a desktop.

1 N High School

2 The University of Tokyo

3 Keio University

Rooms [6] and Task Gallery [7] are the systems that employ the first method. Rooms enable users to create multiple desktops and switch between them. This system can switch between several windows at once, grouped by desktops. Thus, users can switch between screens with windows displayed simultaneously by switching between desktops with multiple windows arranged without overlap. Task Gallery enables switching between tasks through a three-dimensional space that uses a gallery as a metaphor. Users can duplicate desktops and place them on walls or other locations within the space, making it easier to manage tasks through the user's ability to recognize space. In addition, by placing frequently used applications, in easily accessible locations, windows can be easily moved onto the desktop, facilitating the transition to multitasking.

Stage Manager [8] and Docking Window Framework [9] are the systems that employ the second method. Stage Manager is a window management system implemented in macOS. It enables users to place a group of windows as tasks in a thumbnail view at the left edge of the screen, and by clicking on the thumbnail, users can switch windows by task. This allows multiple windows to be switched with the same click operation as a single window. The Docking Window Framework allows multiple windows to be moved together by docking them. This allows several windows to be operated as a single window within the desktop, and multiple windows can be switched together.

These window management systems enable task switching by allowing multiple windows to be operated together. However, none of them integrates the window switching operation with the operation required to display multiple windows simultaneously. For example, in Rooms and Task Gallery, there are operations to place specific windows on the desktop and operations to switch between desktops. Stage Manager and Docking Window Framework also require two operations: one to select windows used for multitasking and the other to switch between them. On the other hand, Air Manager can basically shift from single-tasking to multi-tasking with only a viewpoint operation.

3. Proposal of Air Manager

3.1 Concepts

In Air Manager, as the viewpoint can be moved, several windows can be displayed simultaneously and switched between them by simply moving the viewpoint. Thus, the simultaneous display of multiple windows and switching between windows to be displayed are not viewed as separate concepts in Air Manager and can be performed with a single operation. Air Manager is characterized by allowing users to arrange windows in the x- and z-axis directions within a single continuous three-dimensional space, and to switch between these windows by viewpoint. In addition, the viewpoint can be moved along the x- and z-axes, allowing windows to be positioned off-screen and expanding the workspace on a single desktop.

Air Manager's viewpoint operation has the role of three following functions in a conventional window management system.

- Switching between front and rear windows in an overlapping window system.
- Virtual desktops for task switching.
- Screen splitting for multi-tasking.

In Air Manager, to effectively utilize the above-mentioned viewpoint operation features, arranging windows in the following manner during setup is desirable.

- (1) Group the windows related to the work as tasks and align them in the x-axis direction with the same z-axis coordinates as shown in Figure 1. (In Figure 1, window A, B, and C are grouped as one task and window D, E, and F are grouped as another task.)
- (2) Place the different tasks in a row by shifting them in the z-axis direction as shown in Figure 1. (In Figure 1, window D, E, and F are placed behind window A, B, and C.)

By doing so, switching between single windows, task switching (switching between multiple windows), and multitasking can basically be performed only by operating the viewpoint.

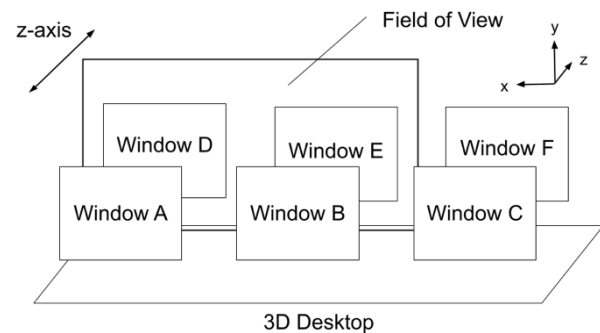


Figure 1. preferable placement of windows in Air Manager

Due to these points, for example, Air Manager does not switch operation methods in the following situations: switching between two overlapping windows on the desktop and users occasionally desire to refer to both windows at the same time. In this case, window management systems in Windows and macOS require a transition to the split-screen or window resizing operation. Both operations require a separate operation different from the window-switching operation on a desktop. Air Manager, on the other hand, does not require such operations. Two windows lined up on the left and right can be switched to the left window by moving the viewpoint to the left (Figure 2) or to the right window by moving the viewpoint to the right (Figure 3). In this case, by moving the viewpoint to a position between the two windows, both can be displayed at the same time. (Figure 4) Therefore, Air Manager has the advantage of being able to simultaneously display multiple windows and switch between them with a single operation. However, some exceptions exist. For example, if windows that users want to display at the same time are not adjacent to each other on the left or right, it is necessary to relocate those windows.

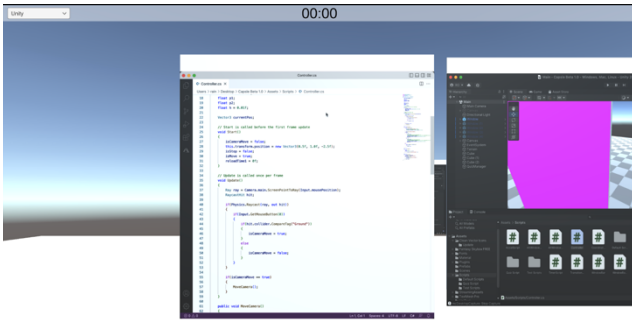


Figure 2. View of the screen when the viewpoint is moved to the left to match the left window

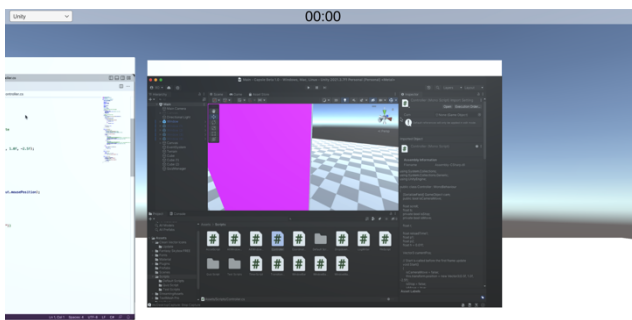


Figure 3. View of the screen when the viewpoint is moved to the right with the window on the right

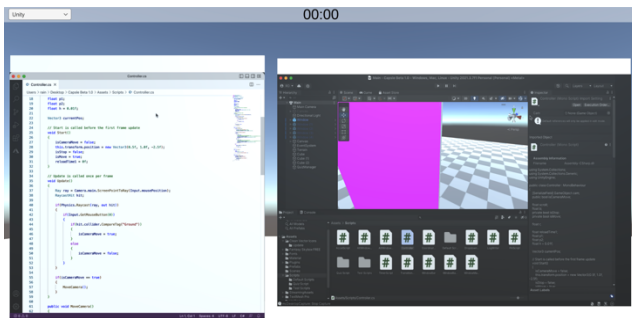


Figure 4. View of the screen when the viewpoint is moved to the nearby area between the two windows to display both windows simultaneously.

3.2 Operation Method

The desktop screen of the proposed system is a projection of a 3D space from a single viewpoint. The viewpoint can be manipulated by dragging and scrolling the ground object displayed at the bottom of the screen. By scrolling, the viewpoint can be moved in the z-axis direction at regular intervals according to the direction of the wheel or trackpad. The ground object has two states: active and inactive. Once clicked, it becomes active and drag and scroll gestures are activated. On the

other hand, when clicking on another window, these gestures are deactivated. This allows the user to use the same gestures as when normally operating a window. Similarly, windows have active and inactive states, and once a window is clicked, it becomes active. A window in the active state moves in the xy-plane to the direction along which the cursor moves and can move along the z-axis at regular intervals in the same way as the viewpoint can be scrolled. These functions allow users to place windows in different positions, including depth, and to switch tasks by viewpoint operation without changing their placement.

4. Development

We developed a prototype using Unity to validate the proposed method. This system runs on macOS and can handle all windows displayed on the desktop. The implementation method is divided into the following two modules: (1) Window Setup Module to display windows, (2) Objects Control Module to move the viewpoint and windows.

4.1 Window Setup Module

Windows in this system are projected onto a game object in Unity, which is captured on the desktop screen. For capturing windows, we used mcDesktopCapture [10], which is based on ScreenCaptureKit. Selecting an application from a dropdown button at the top of the screen generates a new window object for that window. The generated window object is attached with a texture of the captured image of the application window selected in the dropdown and displayed. The object's scale values (x- and y-axis scales) are calculated by dividing the number of pixels of the captured window by 1000 times the size of the object's mesh (a constant based on the ratio of the number of pixels to objects in Unity). In other words, the size and screen ratio of the window object in this system is based on the resolution according to the window size on the macOS desktop.

4.2 Objects Control Module

This section describes the behavior related to object movement. The screen of this system shows the view from a camera in Unity, which is an object used to capture the world (3D space in this case) to be displayed to users. A ray is sent from the center of the camera's viewpoint in the direction of the mouse cursor, and the system identifies objects that are hit by the ray. This allows the mouse cursor to click on a window in the 3D space. The click is determined by the Collider, which is assigned to the window and ground objects. Once the window object detects a click, the coordinates of the window object can be manipulated with gestures. It can move in the x and y-axis directions in response to the up/down and left/right movements read from the mouse position information while users are clicking. (Figure 5) In addition, the coordinates of the z-axis change with the number of times the object is scrolled; one scrolling movement moves the object by 3 in the coordinates in Unity. (Figure 6) (If the value of the scrolling rotation is positive and the scrolling is in the backward direction, the value is +3; if the value of the scrolling

rotation is negative and the scrolling is in the forward direction, the value is -3). When Ray hits an object on the ground, the coordinates of the camera can be manipulated with gestures. The left/right movement is read based on the mouse position while clicking, and the viewpoint moves in the opposite direction of that movement. (Figure 5) This makes it possible to move the viewpoint by dragging objects against the ground. The camera can also be moved in the z-axis direction by clicking on the ground object followed by scrolling, in the same manner as the window. (Figure 6)

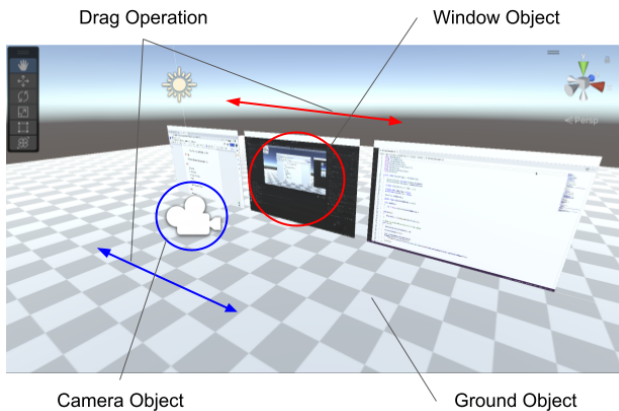


Figure 5. The top image shows the positioning of objects in 3D space.

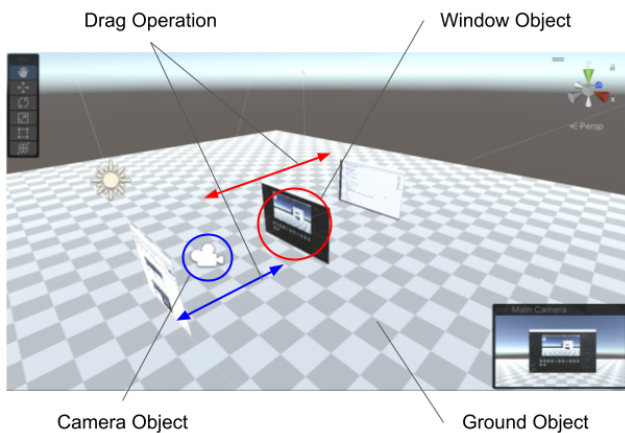


Figure 6. The image shows the positioning of objects in 3D space.

5. Conclusion

This paper proposes Air Manager, a window management system that enables both simultaneous display of multiple windows and switching between them using simple viewpoint operations. Through this system, we have devised an operation method combining the simultaneous display of multiple objects and the switching of displayed objects, two separate operations in conventional window management systems. We found the possibility of solving the problem of Windows and macOS

window management systems that require a separate operation each time when changing the window display between single-tasking and multitasking. Further validation experiments are necessary to evaluate the usability of Air Manager in the future. We also plan to solve problems in Air Manager, such as the problem where users have to move windows when they want to simultaneously display two windows that are not next to each other on the left or right.

Acknowledgments This work is supported by KADOKAWA DWANGO educational institute, Research Club.

Reference

- [1] Bly, B. and Rosenberg J.: A comparison of tiled and overlapping windows, In Proc. CHI '86, 1986, pp.101-106.
- [2] "Snap your windows – Microsoft Support" <https://support.microsoft.com/en-us/windows/snap-your-windows-885a9b1e-a983-a3b1-16cd-c531795e6241> (accessed 2022-12-10)
- [3] "Use two Mac apps side by side in Split View – Apple Support" <https://support.apple.com/en-us/HT204948> (accessed 2022-12-10)
- [4] "Meet Windows 11: Basics – Microsoft Support" <https://support.microsoft.com/en-us/windows/meet-windows-11-the-basics-a7519756-6807-41e4-be66-ed3b2c0abc0d> (accessed 2022-12-10)
- [5] "Work in multiple spaces on Mac – Apple Support" <https://support.apple.com/guide/mac-help/work-in-multiple-spaces-mh14112/mac> (accessed 2022-12-10)
- [6] Henderson, A., Card, S. K., "Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface", ACM Transactions on Graphics, 1986, p.212-242
- [7] Robertson, G., Dantzich, M.v., Robbins, D., Czerwinski, M., Hinckley, K., Ridsen, K., Thiel, D., Gorokhovskiy, V.: The task gallery: a 3D window manager; In Proc. of CHI '00, 2000, p.494-501.
- [8] "Organize your Mac Desktop with Stage Manager" <https://support.apple.com/guide/mac-help/use-stage-manager-mchl534ba392/mac> (accessed 2022-12-10)
- [9] Hirohito Shibata, Kengo Omura. Proposal and evaluation of a window system that enables constructing workspaces, Transactions of the Japanese Society for Artificial Intelligence, 2011, p.237-247
- [10] "Github fuziki/mcDesktopCapture" <https://github.com/fuziki/mcDesktopCapture> (accessed 2022-11-20)