NDSU

CSCI 488 Project Final Submission

Team Members:

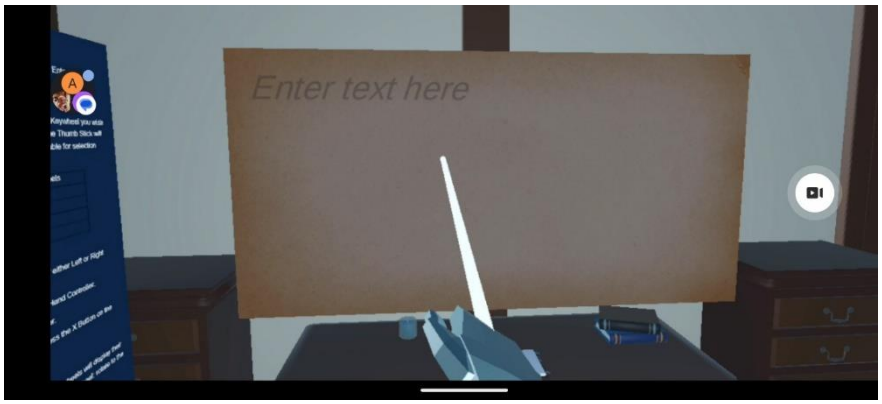
Alex Tjellesen,
Jean Eckelberg

Tasks

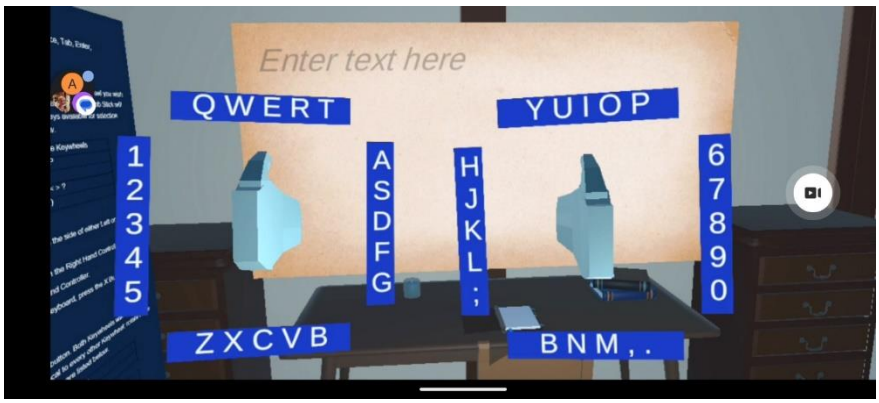
Below you will find a list of the tasks we have implemented as well as an explanation of their implementation. Due to the novelty of our project, we have included an in-depth Tutorial of the Keydial System which can be found at the end of this document.

Select/Deselect Text Entry Object

Providing a seamless transition between our Keydial System and the virtual environment without breaking the player's immersion was a core goal of this project. We wanted the player to be able to enter and exit Text Entry Mode without pulling them out of the environment. We began by making the start of text entry as simple as selecting the object where text will be entered (referred to as a "Text Entry Object"). Rather than bringing up a separate interface upon selection, Text Entry Mode takes place in the same virtual environment.



Above: Pointing at Text Entry Object prior to selecting.



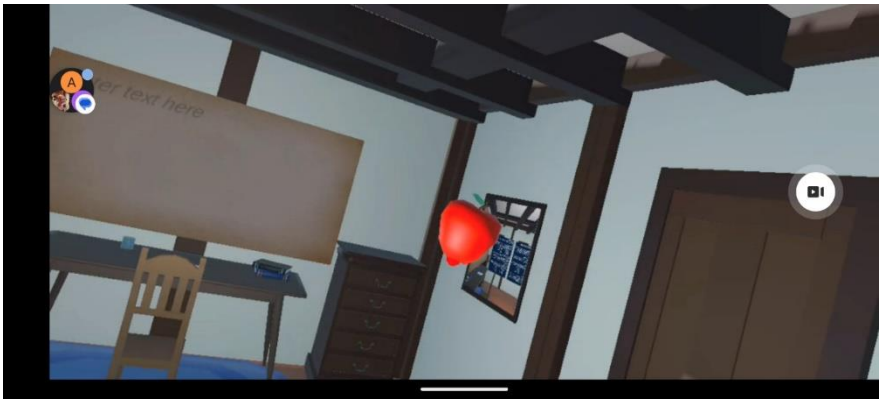
Above: Text Entry Mode – after selecting Text Entry Object.

To leave Text Entry Mode, the player can simply select the Text Entry Object again. From here, the player is free to continue interacting with the virtual environment.

Interact with Grabbable Objects

To further show the seamless integration made possible by our Keydial System, we have built a small virtual environment for the player to explore. The player can walk around and interact with simple objects such as fruit on a table. These “Grabbable Objects” can be picked up by the player using the Hand Trigger on either Controller and moved around the room – just like in the real world and many other virtual worlds.

Above: Looking at Grabbable Object prior to interaction.



Above: Holding Grabbable Object.

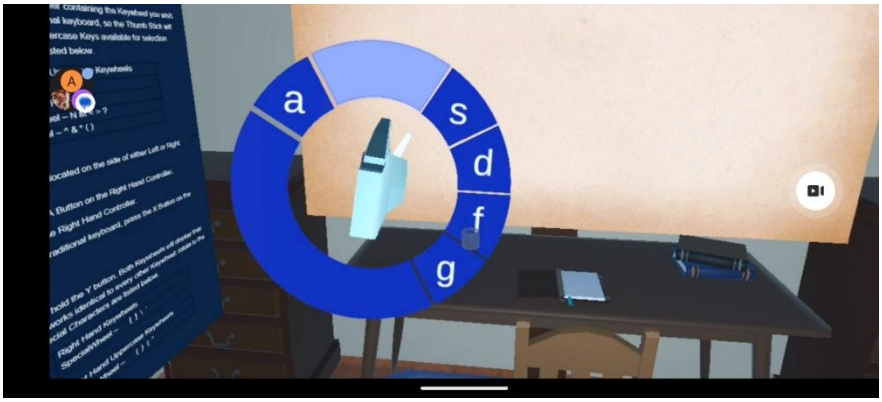
Text Entry

Text entry is where the majority of our project is centered. Our Keydial System replaces the traditional “point-and-click” method of text entry found in nearly every other virtual environment. Once the player has entered Text Entry Mode, they will temporarily lose

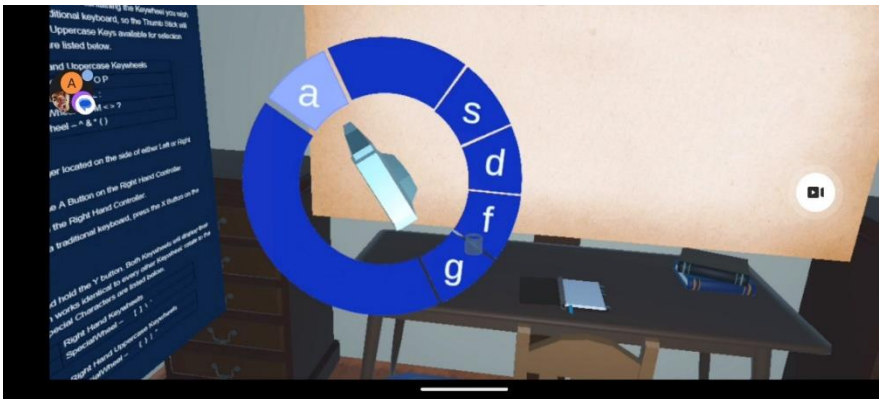
the ability to move around the environment with the Thumb Sticks. The player may still look around and move throughout the environment by turning their head or walking around in the real world. The Thumb Sticks are used to select Keywheels – like rows on a traditional keyboard – while in Text Entry Mode. The steps of text entry are as follows:

Select desired Keywheel with Thumb Stick → Rotate Controller to desired Key → Release Thumb Stick

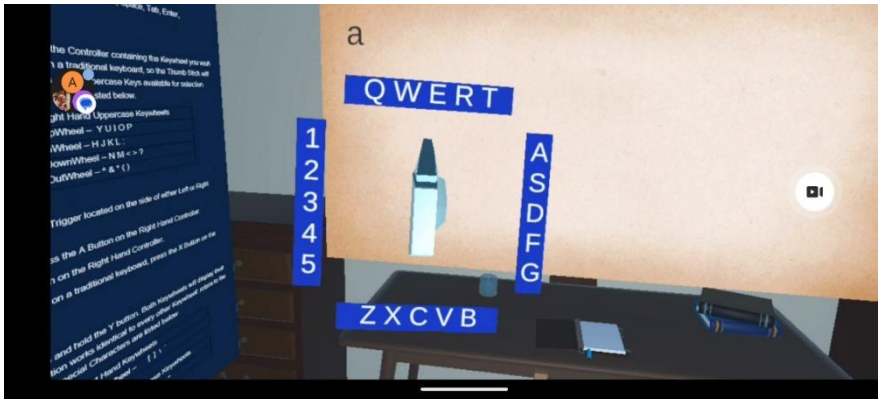
By moving either the Left or Right Thumb Stick up, down, left, or right, the corresponding Keywheel will be displayed around the player's left or right wrist. A list of available Keywheels can be found in the Tutorial at the end of this document. Below is an example of entering the character "a" using the Keydial System.



Above: Holding Thumb Stick in Up position on Left Hand Controller.



Above: Rotating Left Hand Controller to the "a" Key while still holding Thumb Stick.



Above: Releasing Thumb Stick inputs "a" character.

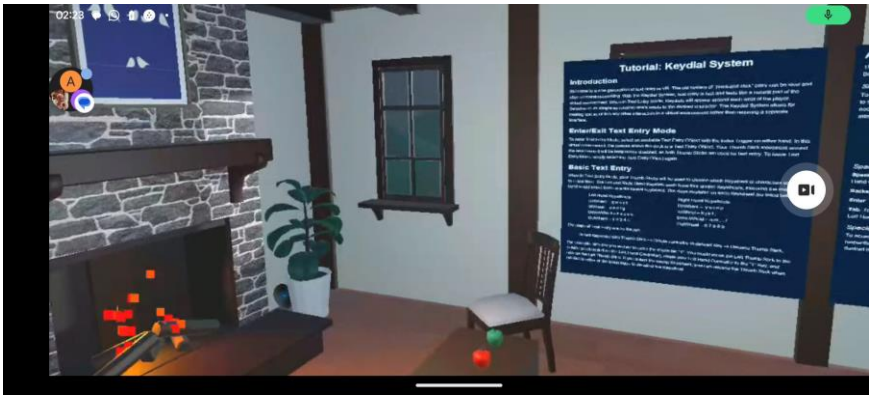
If the user has selected the wrong Keywheel for their desired Key, they can simply rotate to either of the blank Keys on the Keydial and release the Thumb Stick. The Keydial System also provides the functionality for common text entry controls such as Shift, Space, Tab, Enter, and Backspace. Each of these functions are accessed by pressing a button on the Left or Right Hand Controllers, and are further explained in the Tutorial at the end of this document.

Storyboard

Below is an example of a simple interaction a player might have with our environment.

Moving Around the Room

Let's assume the player wants to first explore their surroundings after loading into our virtual environment.

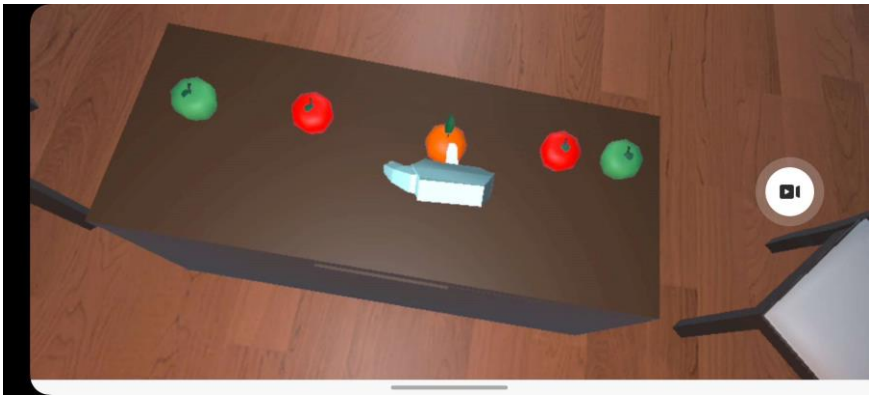


Above: The player has loaded into our virtual environment.

Above: The player moves to inspect the fireplace.

Interacting with Objects

After having a look around the room, the player decides to interact with the fruit on the coffee table.



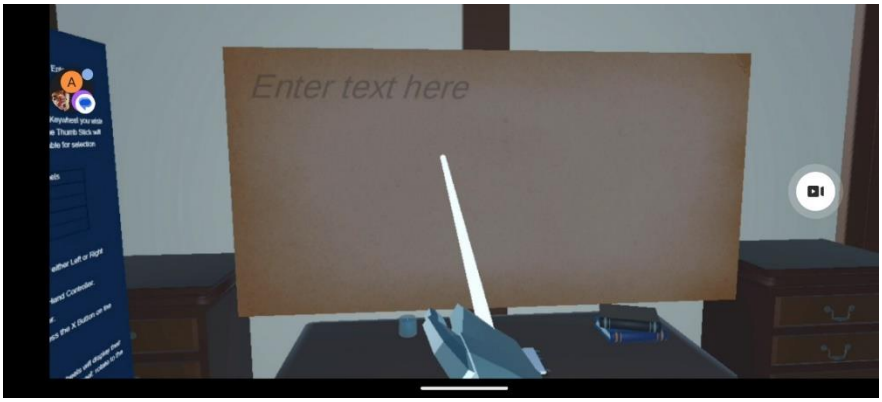
Above: The player prepares to grab the orange from the table.

Above: The player looks at the orange in their hand.

Above: The player throws the orange across the room.

Text Entry

The player now wishes to explore the novel Keydial System!



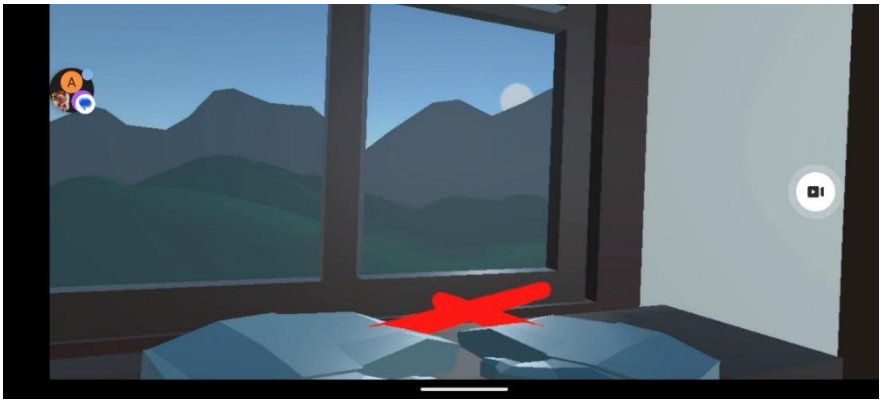
Above: The player points at the Text Entry Object prior to selection.

Above: The player begins to type a message.

Above: The player has finished typing their message.

Back to the Room

Overjoyed with their experience using the Keydial System, the player looks through a window and imagines the possibilities of this new method of text entry!



Above: The player has left Text Entry Mode and looks through a window.

Complexity of Implementation

To implement our Keydial System, we had to entirely rebuild text entry in the virtual environment. We began by creating three canvases in the environment. The first canvas contains a text field object to display the inputted text. This object is shown above the desk in the environment. The other two canvases are used to display the Left and Right Keydials. These canvases are attached to the player's in-game wrists and are enabled when in Text Entry Mode. There are approximately 100 image objects, each containing a text mesh object displaying a different character, for the various Keys available to

input. These objects are split across the left and right canvases and are disabled until needed. We also wrote six custom scripts called LeftCustomController, RightCustomController, LeftCustomCharacter, and RightCustomCharater, LeftWheelController, and RightWheelController. The CustomCharacter scripts are used to disable the continuous movement provider (Thumb Stick movement) when in Text Entry Mode. The CustomController scripts are used to enable the features found on the controller buttons such Select, Grip, Space, Backspace, etc. as well as collect controller rotation information. Finally, the WheelController scripts receive the controller rotation information and enable features such as hovering over characters, selecting Keywheels, and sending selected characters to the main text field object.

We also created a Debugger canvas to display various logs generated during Text Entry Mode. This canvas proved very helpful in creating the Keydial System.

Citations

3D assets used throughout the project taken from Unity's "Create with VR" tutorial. The tutorial can be accessed at https://learn.unity.com/course/create-with-vr.

NDSU

CSCI 488 Tutorial: Keydial System

Team Members:

Alex Tjellesen,
Jean Eckelberg

Introduction

Welcome to a new generation of text entry in VR. The old system of “point-and-click” entry can be slow and often immersion-breaking. With the Keydial System, text entry is fast and feels like a natural part of the virtual environment. When in Text Entry Mode, Keydials will appear around each wrist of the player. Selection is as simple as rotating one’s wrists to the desired character. The Keydial System allows for treating text input like any other interaction in a virtual environment rather than requiring a separate interface.

Enter/Exit Text Entry Mode

To enter Text Entry Mode, select an available Text Entry Object with the Index Trigger on either hand. In this virtual environment, the canvas above the desk is a Text Entry Object. Your Thumb Stick movement around the environment will be temporarily disabled, as both Thumb Sticks are used for text entry. To leave Text Entry Mode, simply select the Text Entry Object again.

Basic Text Entry

When in Text Entry Mode, your Thumb Sticks will be used to choose which Keywheel of characters you wish to select from. The Left and Right Hand Keydials each have four unique Keywheels, mirroring the rows each hand would select from on a traditional keyboard. The Keys available on each Keywheel are listed below:

Left Hand Keywheels	**Right Hand Keywheels**
UpWheel – q w e r t	UpWheel – y u i o p
InWheel – a s d f g	InWheel – h j k l ;
DownWheel – z x c v b	DownWheel – n m , . /
OutWheel – 1 2 3 4 5	OutWheel – 6 7 8 9 0

The steps of Text Entry are as follows: Select Keywheel with Thumb Stick → Rotate controller to desired Key → Release Thumb Stick. For example, let’s say you wished to

enter the character “1”. You would move the Left Thumb Stick to the outside position (left on the Left Hand Controller), rotate your Left Hand Controller to the “1” Key, and release the Left Thumb Stick. If you select the wrong Keywheel, you can release the Thumb Stick when rotated to either of the blank Keys to deselect the Keywheel.

Advanced Controls

The Keydial System also allows for common text entry controls such as Shift, Space, Tab, Enter, Backspace, and entering Special Characters.

Shift

To shift to Uppercase Keywheels, click the Thumb Stick on the Controller containing the Keywheel you wish to select from. This feature functions similar to CapsLock on a traditional keyboard, so the Thumb Stick will need to be clicked again to return to Lowercase Keywheels. The Uppercase Keys available for selection mirror those available using Shift on a traditional keyboard and are listed below.

Left Hand Uppercase Keywheels	**Right Hand Uppercase Keywheels**
UpWheel – Q W E R T	UpWheel – Y U I O P
InWheel – A S D F G	InWheel – H J K L :
DownWheel – Z X C V B	DownWheel – N M < > ?
OutWheel – ! @ # $ %	OutWheel – ^ & * ( )

Space, Backspace, Enter, and Tab

Space: To enter the Space character, press the Hand Trigger located on the side of either Left or Right Hand Controller.

Backspace: To delete the character before your Cursor, press the A Button on the Right Hand Controller.

Enter: To create a new line of text, press the B Button on the Right Hand Controller.

Tab: To enter whitespace similar to the Tab function on a traditional keyboard, press the X Button on the Left Hand Controller.

Special Characters

To access the Special Character Keywheels, press and hold the Y button. Both Keywheels will display their respective Special Character Keywheel. Key selection works identical to every other Keywheel: rotate to the desired Key and release Y to select. The available Special Characters are listed below.

Left Hand Keywheels	**Right Hand Keywheels**
SpecialWheel – ` - =	SpecialWheel – [ ] \ ‘

Left Hand Uppercase Keywheels	**Right Hand Uppercase Keywheels**
SpecialWheel – ~ _ +	SpecialWheel – { } | “