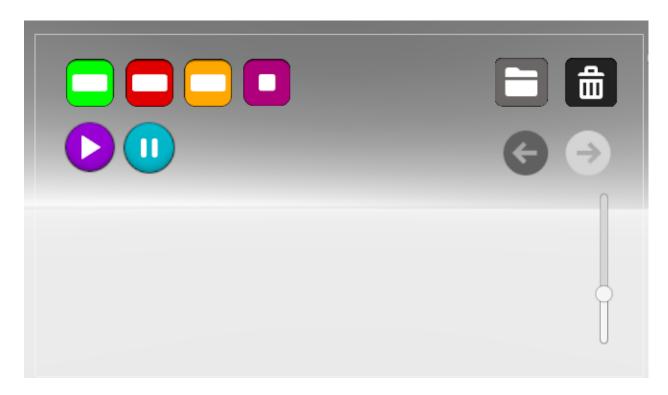
# Part 1:



### 1. UI

- a. Click on the square colorful buttons to create maze blocks
  - i. Click on the round colorful buttons to create start and end tiles
  - ii. Upon clicking, the headworn device's transform.forward will be used to spawn an object right in the user's line of sight.
- b. The "file" button will export all of the maze blocks and start/end tiles into a csv file
  - i. Prior to building the game, I already defined an array of maze blocks
  - ii. The export button will get an array of all the active maze blocks
  - iii. It will also create a dictionary that maps maze prefab → array index
  - iv. Each block is written into a csv file in the following format: "arrayIndex, x.localPos, y.localPos, z.localPos, x.localRot, y.localRot, z.localRot"
- c. To delete an object, hover your right controller over the object and use your left controller (the pointer) to click on the "garbage" button
- d. The "left arrow" and "right arrow" buttons will undo and redo actions
- e. Use the slider to rescale the maze
- 2. Selecting objects (2 ways)
  - a. Hand: Hover your right or left controller over the object
  - b. Pointer: Point the ray in your left controller at the object you want to select. If it is selectable, the ray will turn white
  - c. Hovered objects will turn bright pink. Upon hover exit, they return to their original color

d. To easily showcase hand-driven and pointer-driven selection, I decided to make one controller work as a hand and the other to work as a pointer. This offers a huge benefit, allowing the user to select objects and interact with the far away UI at the same time.

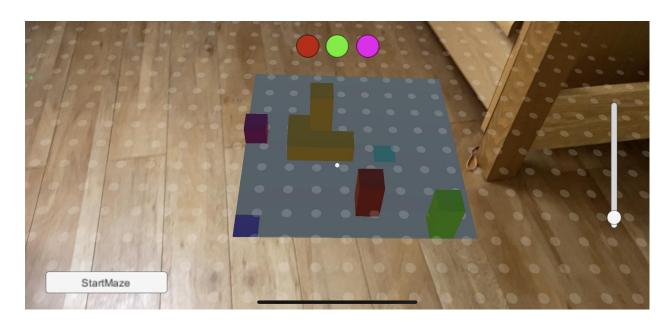
# 3. Grabbing objects (2 ways)

- a. Hand: With your right or left controller, hover over an object and then press the trigger button. This will allow you to pick it up
- b. Pointer: After hovering the ray over the object you want to pick up, press the grip button. This will allow you to pick up the object
- c. To show the user that the object has really been selected, the controller will vibrate. The use of haptics tries to mimic how a real "grab" would work in a real life.

# 4. Transforming objects (2 ways)

- a. Freeform: Grab the object. Then, move/rotate it however you want. When you drop the object, it will snap into a discretized position and rotation.
- b. Controller:
  - i. Right hand: Use the A, B buttons to translate along the Y axis. Use the joystick to translate along the X and Z axis.
  - ii. Left hand: Use the X, Y buttons to rotate along the Y axis. Use the joystick to rotate along the X and Z axis.
- c. The user is technically able to rotate/translate the start and end tiles along to look wacky and unreachable. However, that is discouraged because it will make the AR portion look funky.

Part 2:



1. Importing the CSV file

- a. Once done with the VR portion, plug the device into your computer and find the Android folder
- b. Locate the exported csv file within the Assignment3 subfolder, titled "export.csv"
- c. Drag file into your desktop
- d. Open the AR portion of the assignment and import "export.csv" into Unity
- e. In the hierarchy, navigate to the Grid element
- f. Find the "ReadCSV" component in Grid. It takes a TextAsset argument
- g. Drag "export.csv" into the TextAsset argument
- h. Now you can build and run the game

## 2. Creating the right maze blocks

- a. Prior to running, I already programmed the Grid element with an array of maze blocks. The array is identical to the one created in the VR app
- b. OnStart, the Grid element will run a script called ReadCSV
- c. For each line read, the game object at the correct index will be spawned as a child of the Grid
- d. The block will be assigned the corresponding localPosition and localRotation

#### 3. Transforming the maze

- a. Translate: tap your finger on any areas covered by white polka dots. These are the areas registered as horizontal planes. The maze will respawn there
  - i. Initially, the maze will not appear. This is because the assignment required that the user place the maze on a horizontal plane
  - ii. As a result, the maze will only be spawned on tap
- b. Rotate: drag your finger left or right. This will rotate the maze along its local origin
- c. Scale: Toggle the slider to the right.

#### 4. Selecting objects

- a. Angle the camera so that the selector dot (white dot in the middle of the screen) hovers over a maze block. The block will turn orange, indicating that it's been selected
- b. To deselect, simply angle the camera so that the dot is no longer on the maze block
- c. Using a selection dot rather than more touch/swipe features avoids cross-contamination of information

### 5. Changing color

a. While selecting a maze block, tap on one of the colored circles on the top of the screen. The object will still be a bright orange if the selector dot remains on it. After the dot is moved away, however, the block will change to its new color

# 6. Traversing the maze

- a. Click on the button on the bottom left corner. All UI elements will disappear
- b. A bright orange player object will appear over the purple "start" tile
- c. Tap parts of the maze to move the player to that location
- d. When the player has reached the "end" tile, a panel will reappear prompting the user to restart the game