VR Escape Room: partial project brief

Application concept

Overview

In this project, you will implement the last portion of a VR escape room. In these final interactions, the user must be able to:

- Enter a code on a number pad to receive a keycard
- Swipe that keycard through a card reader to unlock a door
- Slide the door open to exit the room

These user experiences are more complex than the out-of-the-box interactions provided by the XRI toolkit and therefore require custom code that extends the API.

Reference examples

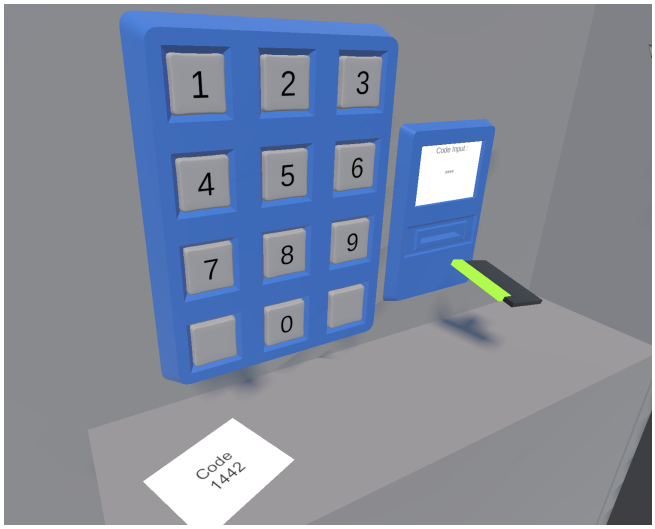
Some examples of VR applications with similar interactions that you might want to explore include:

- **Keep Talking and Nobody Explodes** - a multiplayer game where one player must use instructions provided by partners to diffuse a bomb.
- **The Room VR: A Dark Matter** - A single player puzzle game where the user must solve riddles in order to unravel a larger mystery.

Prioritized Checklist

Below is a check-list with all of the requirements of each feature, categorized as either “must have” or “nice to have”

Number Pad



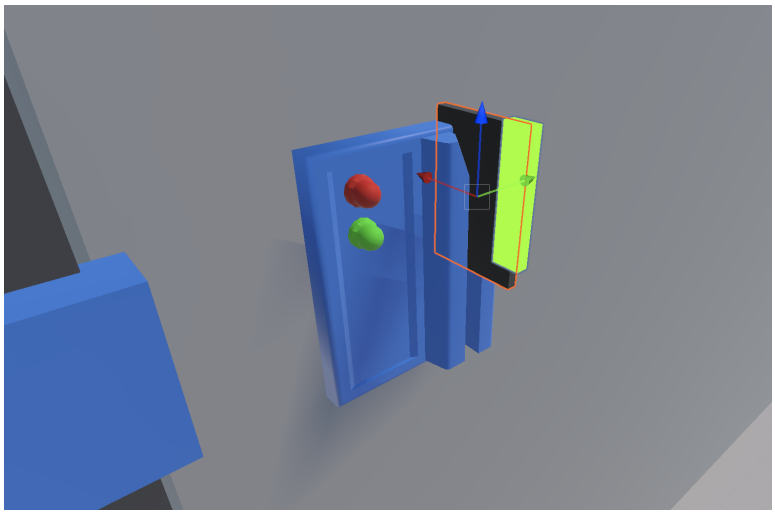
Must have

- [ ] There should be a script named TouchButton.cs, which extends XRBaseInteractable.
- [ ] OnHoverEntered should be used to detect button presses.
- [ ] The color of the button should change when you touch it.
- [ ] The code should be displayed as the buttons are pressed.
- [ ] If the sequence is correct, a keycard should be spawned at the location of the dispenser.
- [ ] If the sequence is incorrect, the number pad should be reset to its original state, allowing the user to enter another code.

Nice to have

- [ ] Differential feedback should be provided for correct and incorrect entries on the keypad.
- [ ] When you have one hand on a button and put a second hand on it, the button should not register a second button push.
- [ ] The buttons should be push buttons, rather than simply touch buttons.

Card Reader



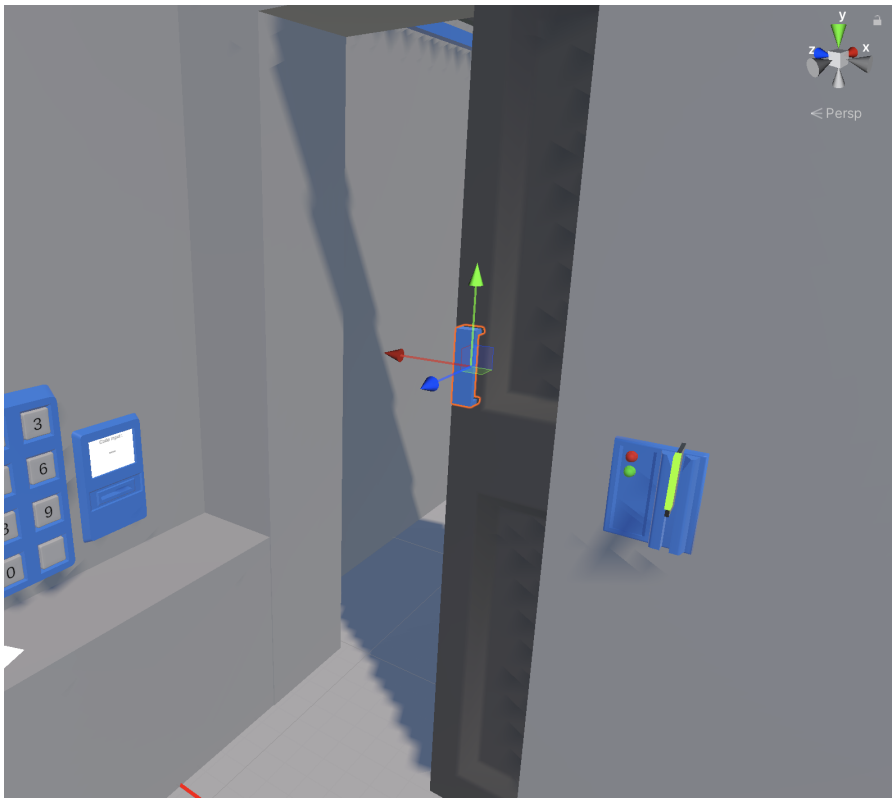
Must have

- [ ] The keycard should be grabbable
- [ ] The CardReader.cs script must extend the XRSocketInteractor class, but the keycard should not snap into the socket like a typical socket.
- [ ] The swipe should only register if the card is swiped down a minimum distance through the reader.
- [ ] When a swipe is successful, the lock and bar across the door should disappear.

Nice to have

- [ ] When the swipe is valid, the green light should come on. When the swipe is invalid, the red light should come on.
- [ ] The swipe should only be valid if it moves through the reader at an appropriate speed.
- [ ] As long as the user's hand is close enough to the reader, the card could snap to the track within the reader and follow that path.

Sliding Door



Must have

- [ ] The door handle should extend XRBaseInteractable.
- [ ] The door should only be movable after the lock has been opened or removed.
- [ ] The door should feel heavy: it speeds up as you pull farther from it (almost as if you're pulling it with a rubber band).
- [ ] The door should only move if you're pulling in the appropriate direction. For example, nothing should happen if you pull directly perpendicular to it.

Nice to have

- [ ] Include a sound effect with the sliding of the door to enhance the feeling of heaviness.
- [ ] Dynamic haptic feedback while the door is opening: stronger vibrations as the door is beginning to move and lighter vibrations once it is already moving and there's less friction.