Develop VR!

Get ready to create your first fully interactive multisensory VR experience!

Getting Started

1. Download Course 3 Starter Project[1]
2. Open the project in Unity and load the scene called Maze.unity

Create a Maze Scene

1. Using the supplied Unity prefabs, make a maze environment.
2. Make sure it's big enough to spend a minute or two moving around in it.
3. Place the door prefab so it blocks the entrance to the building.
4. Pick a spot to hide a key.
5. Light the scene (and be sure to save it!)
6. Optionally, add some ambient sounds, animations, or particle effects.

Add VR Functionality

- If one doesn’t already exist, then add the Event System to support interactions.

Build the waypoint navigation system.

- Find the waypoint prefab in the project folder.
- Position them throughout the maze so the user can move around. Note that the Waypoint prefabs included in starter project are different from the ones used in the course. This will allow you to let users explore the maze without directing them. As such, Waypoints do not have 'neighbors' - you can travel to any waypoint.
- Test the scene (make sure you can get everywhere you need to be without getting stuck.)

Make collectable items.

- Find the Coin prefab and finish the Coin.cs script so that when you click on a coin then it will play a sound and then disappear.

- The best way to do this is to Instantiate a new “Poof” prefab when the Coin is clicked, and then destroy the Coin itself. The “Poof” prefab will play the sound and emit some colorful particles.
- First, find the Poof prefab. Copy it and create a new CoinPoof prefab. Drag-and-drop the coin.wav sound into the CoinPoof prefab’s AudioSource clip slot. Now you have a custom Poof just for coins. We’ll be making other “Poof” objects later in the project.
- Inside the script, customize it so that the CoinPoof prefab is Instantiated at the coin’s position and then destroy the coin.
- Make sure that the poof goes vertical and not horizontal.

Create a key to pick up.

- We want to make sure the door can't be opened until the key is collected.
- Just like the Coin, create a new KeyPoof prefab and assign its AudioSource clip to key.wav
- Also, just like the Coin, make it so that the Key instantiates the KeyPoof prefab and gets destroyed when you click on it.
- Now, add code to the Key so that it call the Unlock() method on the Door when you click on it. We’ve already created this method, but you’ll fill it in later.
- For now, create a reference to the Door object in the Key script and drag-and-drop the door into it.
- Then, make sure to call door.Unlock() when the Key is clicked.

Design a gate blocking the path to the treasure.

- Fill out Door.cs so that it moves up when the Unlock() method is called. Recall the falling coconut lesson for details on how to do it.

Construct and display a UI Canvas

- Find the SignPost prefab and place it inside the building
- Finish the SignPost.cs script so the scene is reloaded when the user clicks on it.

Deploy, Test and Optimize

- Be sure that the scene runs on your device and can be completed.
- Make sure performance is acceptable. (Hint: don't be surprised if optimizations are necessary.)
- Apply what you've learned in prior courses to optimize the scene.

Congrats, you've done it. Time to submit your project!

Links

1. http://www.udacityvr.com/downloads/