Experiment No.7	
Create an immersive environment with only static game objects.	
Date of Performance:	
Date of Submission:	
Marks:	
Sign	

Aim: Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects.

Theory:

Unity's GameObject class represents anything which can exist in a Scene. GameObjects are the building blocks for scenes in Unity, and act as a container for functional components which determine how the GameObject looks, and what the GameObject does.

In scripting, the GameObject class provides a collection of methods which allow you to work with them in your code, including finding, making connections and sending messages between GameObjects, and adding or removing components attached to the GameObject, and setting values relating to their status within the scene.

If a GameObject does not move at runtime, it is known as a static GameObject. If a GameObject moves at runtime, it is known as a dynamic GameObject.

Many systems in Unity can precompute information about static GameObjects in the Editor. Because the GameObjects do not move, the results of these calculations are still valid at runtime. This means that Unity can save on runtime calculations, and potentially improve performance.

Problem Statement:

Creating an immersive environment with static game objects involves setting up a scene with various 3D models and environmental elements to create the desired atmosphere. Below, I'll outline how you might create a living room environment using Unity and some basic static game objects:

1. **Setting Up Unity Project:**
   - Open Unity and create a new 3D project.
2. **Importing Assets:**
   - You can use pre-existing 3D models from asset stores or create your own using tools like Blender.
   - Import 3D models representing furniture, decorations, and other elements of a living room environment into your Unity project.

3. **Scene Setup:**
   - Create a ground plane to serve as the floor of the living room.
   - Place furniture models such as sofas, tables, chairs, and shelves in appropriate locations within the scene.
   - Add decorative elements like plants, paintings, lamps, and rugs to enhance the ambiance.
   - Consider adding windows, doors, and lighting to make the environment feel more realistic.

4. **Lighting:**
   - Use Unity's lighting system to set up realistic lighting conditions within the living room.
   - Adjust the direction, intensity, and color of lights to create the desired mood.

5. **Camera Setup:**
   - Set up a camera to serve as the player's viewpoint within the living room.
   - Adjust the camera's position and rotation to provide the desired perspective.

6. **Navigation:**
   - You can allow the player to navigate the environment using a first-person or third-person controller.
   - Alternatively, you can set up fixed camera angles to create a more cinematic experience.

7. **Audio:**
   - Add background music or ambient sounds to further immerse the player in the environment.
   - Consider adding sound effects for interactions with objects or environmental elements.

8. **Optimization:**
   - Optimize the scene by baking lighting and occlusion culling to improve performance.
   - Consider using LOD (Level of Detail) to optimize the rendering of distant objects.

9. **Testing:**
   - Test the environment to ensure that it looks and feels immersive.
   - Make adjustments as needed to improve the overall experience.

By following these steps, you can create an immersive living room environment using static game objects in Unity. You can apply similar principles to create environments for other settings such as a battlefield or tennis court, adjusting the assets and layout accordingly.

Adding Collison Detection

1. Click on the OVRPlayerController in your Hierarchy.
   - In the character controller section change the Radius to 0.2
2. Scroll to the bottom and click Add Component.
   - Add a Character Camera Constraint
   - Check Enable Collision
   - Check Dynamic Height
   - In your hierarchy, expand the OVRPlayerController and find the OVRCameraRig. **Drag and drop it into the Camera Rig field of the Character Camera Constraints script.**

Adding Hands With LocalAvatar

1. In your Hierarchy, expand your OVRPlayerController till you see the TrackingSpace underneath the OVRCameraRig.
2. In your Assets folder, find the LocalAvatar prefab.
3. Drag and drop LocalAvatar on top of the TrackingSpace. Do not place it underneath it. If you get a pop-up about modifying the prefab, you've done it wrong. TrackingSpace should expand and you should see a +LocalAvatar at the bottom of its list.
4. Find the Ovr Avatar (Script) component within +LocalAvatar
5. Ensure that Show Third Person is not checked.
6. Optionally, un-check Can Own Microphone because we won't be using the mic in this tutorial.
7. Build the object and if you can't see your hands then you might need to go to Oculus(Toolbar menu)->Avatars->Edit Settings.
8. You might be required to create an App in your Oculus Developer Dashboard (Link)

9. Create a new App and enter the App id here(Type Oculus Go/Quest if you are working on Oculus Quest). See attached screenshot

Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Now build your project and you will be able to your hands inside the scene.

Making the Sphere Grabbable

Add a Rigidbody component to the sphere after clicking the Add Component button.

Add an OVRGrabbable component

Making your Hands Grabblable :)

Find LeftHandAnchor and RightHandAnchor under the OVRPlayerController in your hierarchy.

Select them both and add a Sphere Collider

Set the radius to 0.05

Check the Is Trigger box

Add an OVR Grabber script to both

Check Parent Held Object or you will see some jumpiness when holding and object and moving your hands.

2. Expand the RightHandAnchor and find the RightControllerAnchor under it.

On the RightHandAnchor, drag and drop the RightControllerAnchor to the Grip Transform field

CSDOL7021: User Experience Design with VR Lab

Under Grab Volumes on the RightHandAnchor do the following

Set the size to 1

Drag the Sphere Collider of the RightHandAnchor in to this field.

In the Controller dropdown, select R Touch

3. Do the same thing for the LeftHandController for the left hand.

When you start up your project now, you should be able to pick up the sphere.

This concludes our post. I will be writing some more posts about creating small games in VR and Unity.

Do you let me know in the comments section if you get stuck somewhere. Happy to help.
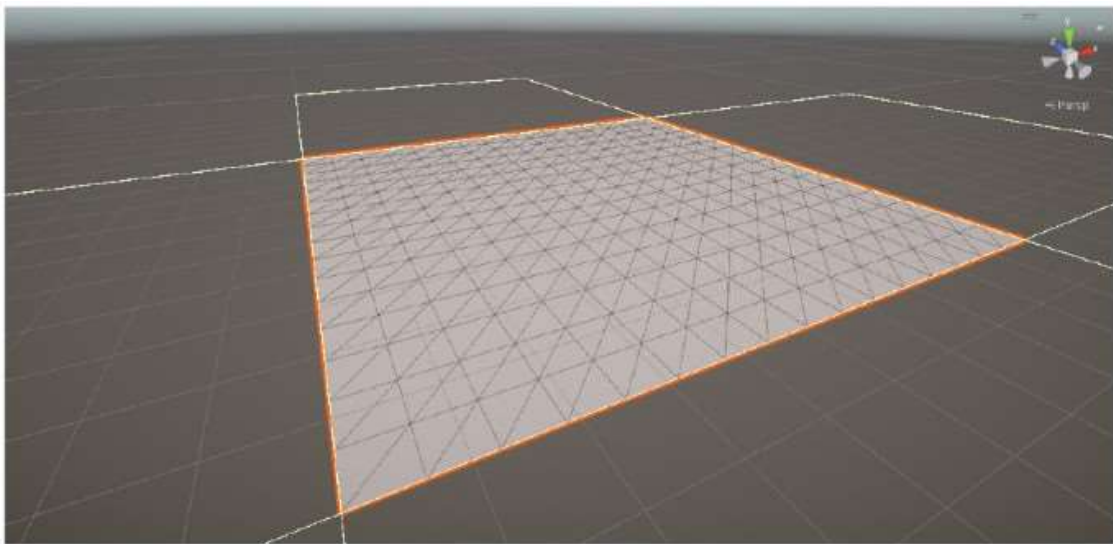
Conclusion:

What are some techniques you can use to enhance the realism and immersion of the environment, such as lighting, texturing, and object placement? How do these elements contribute to the overall atmosphere of the scene?