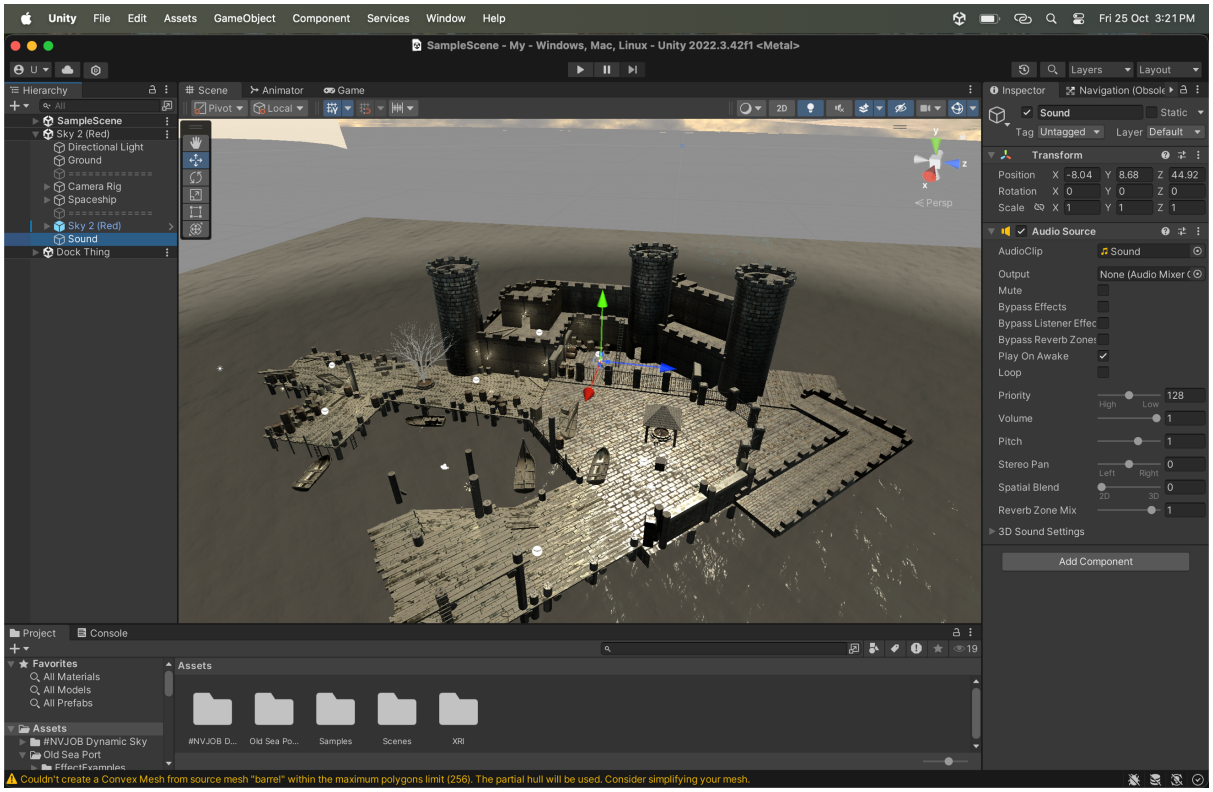
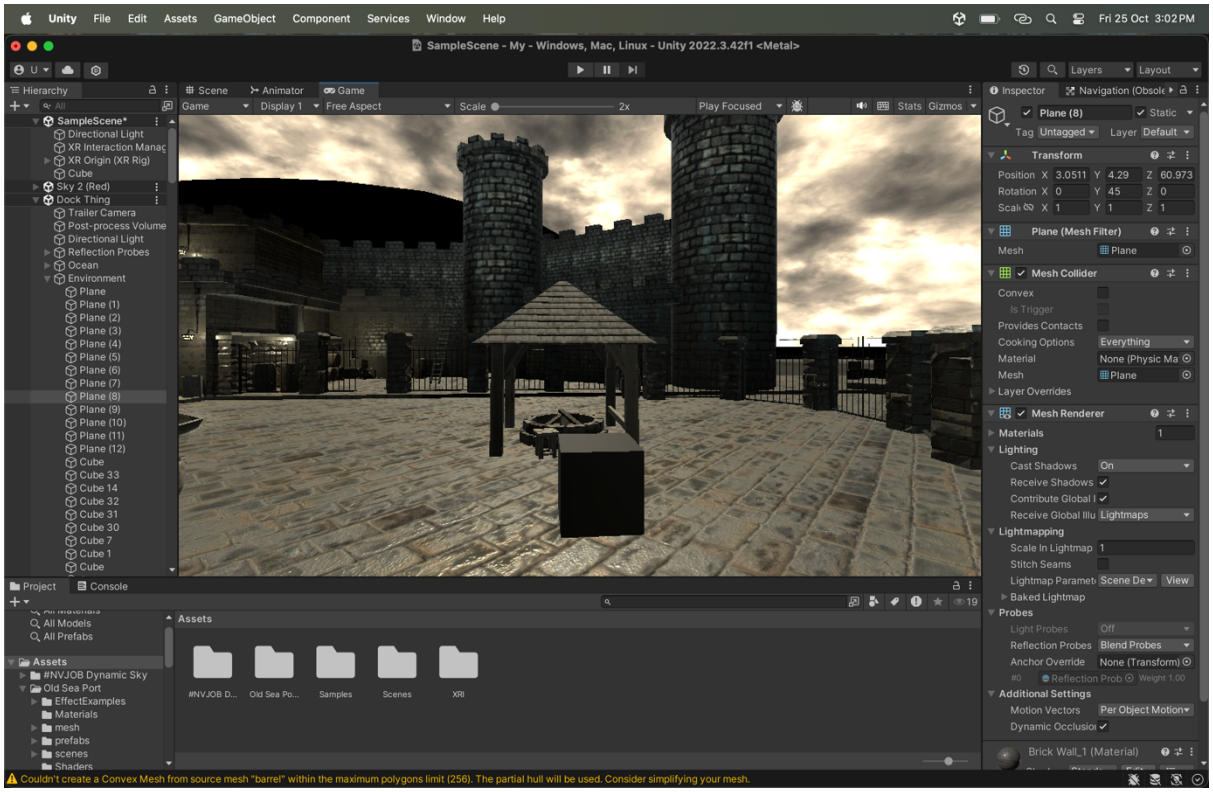
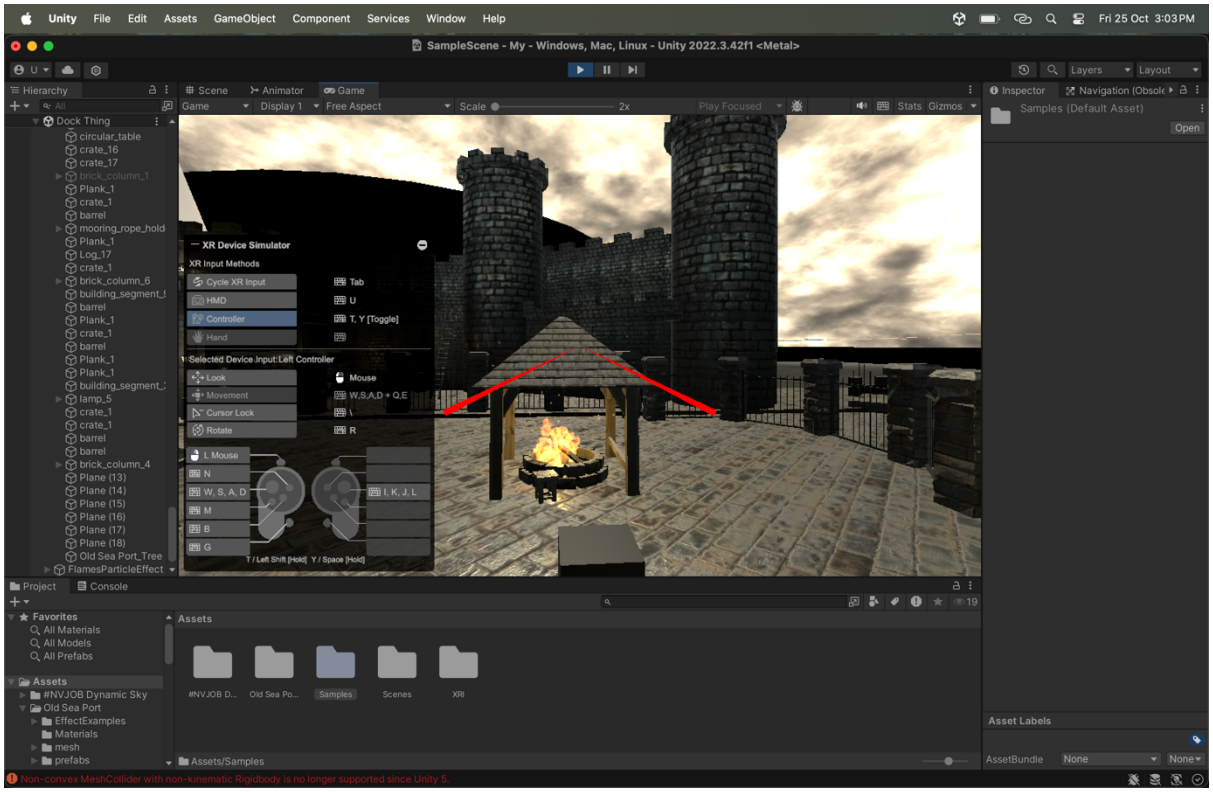
CMP510 (AR and VR Technologies)
Assignment

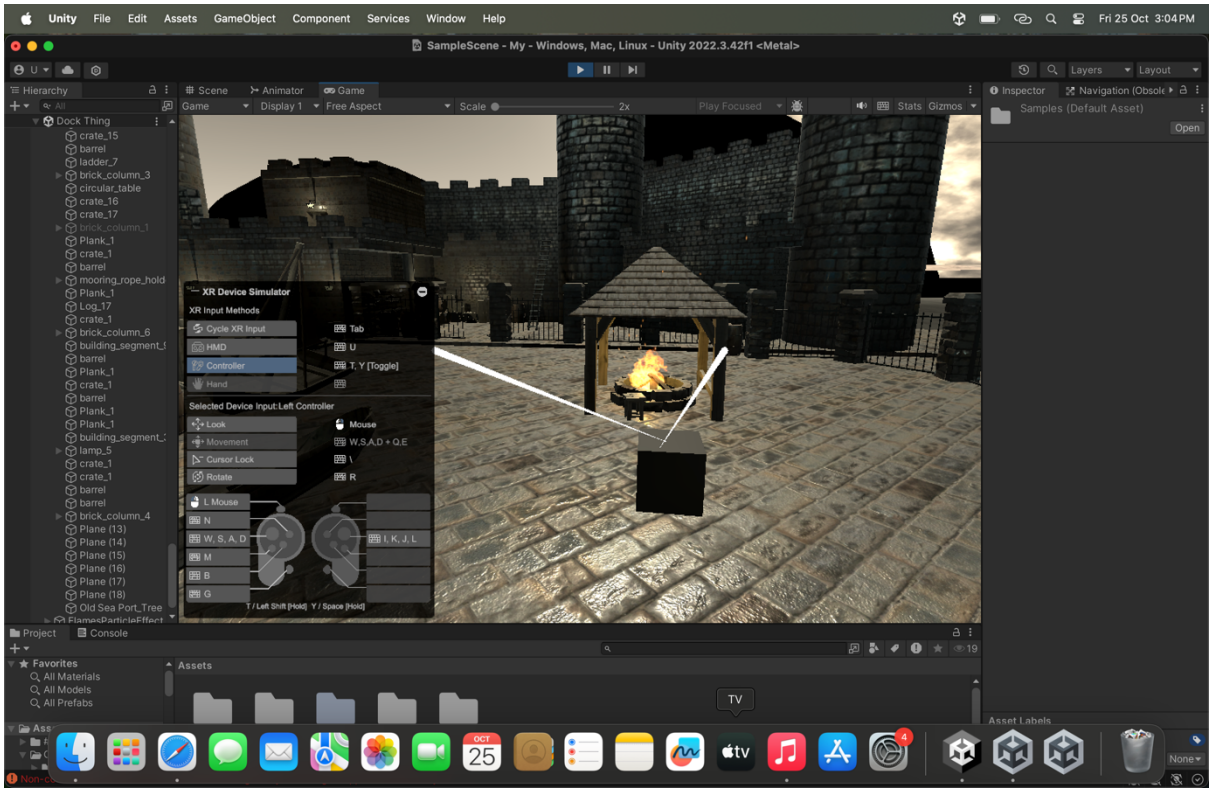
Navrachana University

OUTPUTS

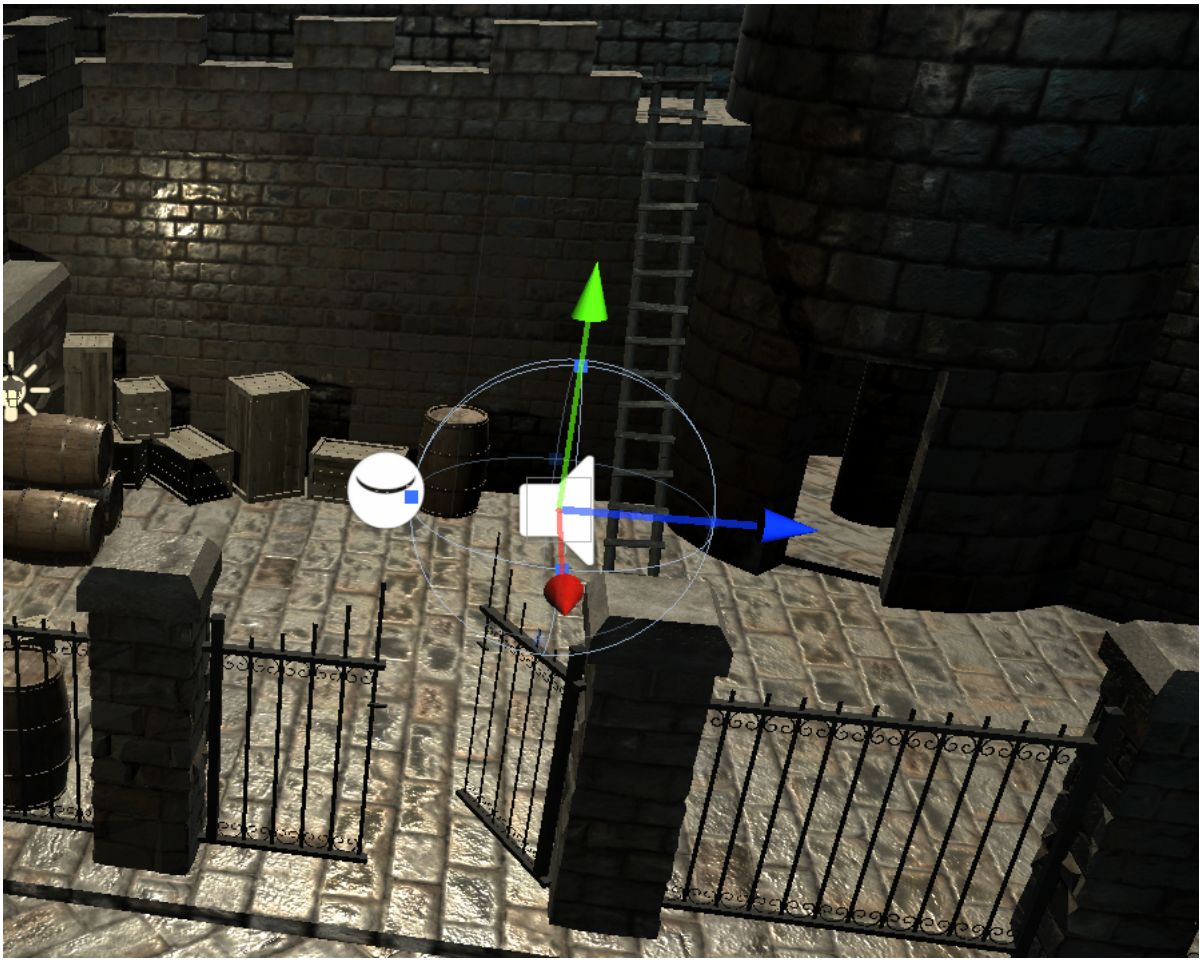

Unity File Edit Assets GameObject Component Services Window Help
Fri 25 Oct 3:21 PM
SampleScene - My - Windows, Mac, Linux - Unity 2022.3.42f1 <Metal>
Hierarchy
Scene
Animator
Game
Inspector
Project
Console
Assets
Couldn't create a Convex Mesh from source mesh "barrel" within the maximum polygons limit (256). The partial hull will be used. Consider simplifying your mesh.


Unity File Edit Assets GameObject Component Services Window Help
Fri 25 Oct 3:02 PM
SampleScene - My - Windows, Mac, Linux - Unity 2022.3.42f1 <Metal>
Hierarchy
Scene
Animator
Game
Inspector
Project
Console
Assets
Couldn't create a Convex Mesh from source mesh "barrel" within the maximum polygons limit (256). The partial hull will be used. Consider simplifying your mesh.


Unity File Edit Assets GameObject Component Services Window Help
Fri 25 Oct 3:03 PM
SampleScene - My - Windows, Mac, Linux - Unity 2022.3.42f1 <Metal>
Hierarchy
Scene
Animator
Game
Inspector
XR Device Simulator
XR Input Methods
Cycle XR Input
HMD
Controller
Hand
Selected Device Input:Left Controller
Look
Movement
Cursor Lock
Rotate
L Mouse
Mouse
W,S,A,D + Q,E
Tab
U
T, Y [Toggle]
R
N
M
B
G
I, K, J, L
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Samples
Scenes
XRI
Asset Labels
AssetBundle
Assets/Samples


Unity File Edit Assets GameObject Component Services Window Help
Fri 25 Oct 3:04 PM
SampleScene - My - Windows, Mac, Linux - Unity 2022.3.42f1 <Metal>
Hierarchy
Scene
Animator
Game
Inspector
XR Device Simulator
XR Input Methods
Cycle XR Input
HMD
Controller
Hand
Selected Device Input:Left Controller
Look
Movement
Cursor Lock
Rotate
L Mouse
Mouse
W,S,A,D + Q,E
Tab
U
T, Y [Toggle]
R
N
M
B
G
I, K, J, L
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Asset Labels
TV


XR Device Simulator
XR Input Methods
Cycle XR Input
Tab
HMD
U
Controller
T, Y [Toggle]
Hand
Selected Device Input: Left Controller
Look
Mouse
Movement
W,S,A,D + Q,E
Cursor Lock
\
Rotate
R
L Mouse
N
W, S, A, D
I, K, J, L
M
B



Objective:

To create a basic virtual environment in Unity that includes a ground plane, a skybox, environmental objects, lighting, and simple VR interaction. The player should be able to grab and move the grabbable objects in the environment.

Task 1: Set Up Your Unity Project & Configure the VR Environment [5 marks]

- Download Unity Hub from the Unity website.
- Install the Unity Editor version compatible with your project (LTS versions are stable).
- Open Unity Hub, click "Install Editor", and choose the necessary version.
- Click "New Project".
- Select 3D or VR (depending on your template options).
- Name your project and choose a save location, then click Create.
- In Unity, go to Edit > Project Settings.
- Scroll down to XR Plug-in Management.
- Click Install XR Plugin Management.
- After installation, go to the XR Plug-in Management settings.
- In the XR Plug-in Management settings, check the platform you are building for (e.g., Oculus, OpenXR, etc.).
- Install the respective plug-ins (like OpenXR for cross-platform VR support).
- Go to Edit > Project Settings > Player.
- Under the XR Settings, check Virtual Reality Supported if required.
- Adjust other settings like resolution, and ensure the project is set to target your VR platform.

- Open Package Manager (Window > Package Manager).
- Search for and install XR Interaction Toolkit and Input System to manage VR interactions.
- In Unity's Hierarchy, right-click and choose XR > XR Origin (VR) to create your VR rig.
- This includes the camera and controllers, providing a basic VR setup.
- Under XR Origin, check for Camera and Controllers.
- If using a specific headset (e.g., Oculus), map the input controls for actions like moving, grabbing, or interacting with objects.

- Connect your VR headset to your computer.
- Ensure Oculus Link or SteamVR is running (depending on the headset).
- Press Play in Unity to test the VR environment.
- Adjust lighting, performance settings, or add objects to interact with.
- When ready to test on the device, go to File > Build Settings, select your platform, and hit Build.

Task 2: Create the Ground Plane. [5 marks]
• Create a larger ground area so player can move around
• You can use Terrain object for that

1. **Add a Terrain**:

   Right-click in the Hierarchy panel, go to 3D Object, then click Terrain. This will add a big ground to your scene.
2. **Make the Terrain Bigger**:

   Select the Terrain in the Hierarchy**.** In the Inspector (on the right side), you'll see options to change the Width and Length. Make these bigger, like 500x500, to give your player more space.

3. **Flatten the Ground**:

   In the Inspector, find the Terrain Paint Tools (a brush icon). Choose Set Height, set the height to 0, and click Flatten to make the ground flat.

4. **Add a Texture**:

   If you want the ground to look like grass or dirt, click Paint Texture in the Terrain settings and pick a texture you like.

5. **Test Your Ground**:

   Press Play button to see if the ground is big enough for your player to move around.

Task 3: Add a Skybox [5 marks]

- **You can use sky presets.**
- **Try to make it more detailed and more interesting.**

1. **Open Skybox Settings**:

   Go to Window > Rendering > Lighting and click on the Environment tab.

2. **Download Skybox Presets**:

   Open the Asset Store, search for Skybox, and download a preset you like.

3. **Import Skybox**:

   In Package Manager, import the skybox assets you downloaded from the Asset Store.

4. **Set the Skybox**:

   In the Lighting settings, find Skybox Material, click the circle icon, and select the skybox you imported.

5. **Customize the Skybox**:

   Adjust settings like Rotation, Exposure, and Tint to make it look more interesting.

6. **Test It**:

   Press Play button to see how your new sky looks in the scene.

Task 4: Add Environment Objects [15 marks]

- You can use Assets to create an engaging Environment
- Create a grabbable objects spawning at random locations

1. Use Assets to Create an Engaging Environment

• Go to the Unity Asset Store and search for environment-related assets like “Low Poly Environment” or “Dungeon Assets.”

• Once you find suitable assets, click Download and Import them into your Unity project.

• In the Hierarchy window, right-click and choose 3D Object > Create Empty to organize the objects in a container or add individual objects.

• Drag objects from the Project window (under Assets) to the Scene view to place them
in your environment. Position and rotate them to your preference to create an engaging
setup.

2. Make Objects Grabbable

• Choose which objects in the environment will be interactable (e.g., gems, pots).

• Add a Collider and a Rigidbody component to make the object

• Use the XR Grab Interactable component from XR Interaction Toolkit to enable grabbing functionality.

• Test the objects by running the scene and using your keyboard controls (WASD and Space/Shift) to move and interact with the grabbable objects.

3. Randomize Object Spawning

• In Unity, create multiple spawn points where the objects will randomly appear each time the scene is loaded.
• Configure the spawn points and objects to be placed in various locations to ensure the player has to search for them in different spots each time.

Task 6: Configure Lighting and Shadows [5 marks]

1. Add a Light Source

- Right-click in the Hierarchy window.
- Select Light > Directional Light (simulates sunlight for outdoor scenes).
- Adjust the position and rotation of the light as needed.

2. Enable Shadows for the Light

- Select the Directional Light in the Hierarchy.
- In the Inspector window, under the Light component, find the Shadows option.
- Choose either Soft Shadows or Hard Shadows to enable them.

3. Enable Shadows for Objects

- Select the object in the Hierarchy.
- In the Inspector, go to the Mesh Renderer section.
- Make sure Cast Shadows and Receive Shadows are checked.

4. Adjust Global Lighting Settings

- Open the Lighting Settings by going to Window > Rendering > Lighting.
- Adjust Skybox, Ambient Light, and Reflections to control the overall scene lighting.

5. Bake Lighting for Static Objects

- Select objects that won't move in the scene (like the ground or walls).
- Check the Static box at the top of the Inspector.
- In the Lighting Settings, click Generate Lighting to pre-calculate lighting for these objects.

6. Adjust Shadow Distance and Quality

- Go to Edit > Project Settings > Quality.
- Adjust the Shadow Distance to control how far shadows are visible.
- Modify the Shadow Cascades for better shadow quality at various distances.

7. Test in Play Mode

- Click Play to see how the lighting and shadows appear in your scene.

Task 7: Add Audio [5 marks]

1. Prepare Your Audio Files:

- Obtain the audio files (e.g., .wav, .mp3).
- Import audio files into the Assets folder.

2. Add an Audio Source:

- Right-click in the Hierarchy panel and select Create Empty.
- Name it AudioManager.
- With AudioManager selected, go to the Inspector panel.
- Click on Add Component and choose Audio Source.

3. Set Up the Audio Source:

- Drag your audio file into the Audio Clip field in the Audio Source.
- Adjust Play On Awake, Volume, and Pitch settings as desired.

4. Control Audio in Script:

- Create a new C# script (e.g., AudioManagerScript) and attach it to the AudioManager.

5. Trigger the Audio:

- In the script where you want to play the sound (like LogicScript), add a reference to the AudioManager and call PlaySound() when needed.

6. Test Your Audio:

- Click the Play button in Unity and check if the audio plays when triggered.

7. Adjust as Needed:

- Modify Audio Source settings or add more audio effects as required.

Task 8: Implement Basic VR Interaction [25 marks]
• Create a Grabbable Object
• Add Grabbable and Grabber Components

1. Create a New 3D Object:

- Right-click in the Hierarchy panel.
- Select 3D Object > Cube (or another shape).

2. Add a Collider:

- Select the object.
- Check if Box Collider is present; if not, click Add Component and select Box Collider.

3. Add a Rigidbody Component:

- With the object selected, click Add Component.
- Search for and select Rigidbody.
- Ensure Use Gravity is checked.

4. Create the Grabbable Script:

- Right-click in the Project panel and create a new C# Script named Grabbable.

5. Add the Grabbable Component:

- Select the grabbable object.
- Click Add Component and select the Grabbable script.

6. Create a Grabber Script:

- Right-click in the Project panel and create a new C# Script named Grabber.

7. Add the Grabber Component:

- Create a new empty GameObject to represent the VR hand.
- Select the new GameObject and click Add Component.
- Select the Grabber script.

8. Set Up VR Input:

- Ensure your VR setup is configured correctly (e.g., Oculus, HTC Vive).
- Check input settings for grab actions.

9. Test in Play Mode:

- Press Play in the Unity Editor.

Task 9: Write the VR Interaction Script [25 marks]

To integrate gem grabbing into this script, you need to trigger the AddGem method when the player successfully places a gem into the correct location (like a socket or box).

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.XR.Interaction.Toolkit;
public class GemManager : MonoBehaviour
{
[SerializeField] private int numberOfGems; // Total number of gems required
private int numberOfGemsInSocket = 0; // Current number of gems in the socket
```

```
public UnityEvent unlock; // Event to unlock when all gems are placed
// This function is called when a gem is successfully placed
public void AddGem()
{
numberOfGemsInSocket += 1;
CheckForAllGems();
}
// Check if all gems are placed to trigger unlock event
private void CheckForAllGems()
{
if (numberOfGemsInSocket == numberOfGems)
{
unlock.Invoke(); // Unlock when all gems are in place
}
}
// Optional: Function to remove a gem
public void RemoveGem()
{
numberOfGemsInSocket -= 1;
}
// Optional: Handle grabbing interaction
public void OnGrab(XRBaseInteractor interactor)
{
// You could add audio or visual feedback when the gem is grabbed
Debug.Log("Gem grabbed by: " + interactor.name);
}
// Optional: Handle release interaction
public void OnRelease(XRBaseInteractor interactor)
{
```

```
Debug.Log("Gem released by: " + interactor.name);
}
}
```

Task 10: Demo application [5 marks]

1. VR Environment Setup:

The player moves through the virtual room using keyboard controls (WASD keys) to interact with hidden gems.

2. Object Interaction:

The player picks up three gems placed around the environmentand places them onto a designated box in the scene.

3. Door Mechanism:

Once all three gems are placed, the door unlocks, allowing the player to escape the room.

4. Audio Integration:

Sounds are played during interactions, such as when the player grabs a gem.

5. Lighting and Shadows:

Torches and spotlights are strategically placed in the room,providing immersive lighting effects.