

**ASSIGNMENT**  
**of**  
**Augmented Reality and Virtual Reality**  
**CS 437**

**Bachelor of Technology (CSE)**  
By

Dhruv Sharma ( 21124012)

Last Year, Semester 7

Course In-charge: Prof. Darshan Parmar



Department of Computer Science and Engineering  
School Engineering and Technology Navrachana University,  
Vadodara  
Spring – 2024.

Github Repo Link: <https://github.com/OPSDrag/LavaZone-VR>

Demo Video Link: [DEMO VIDEO](#)

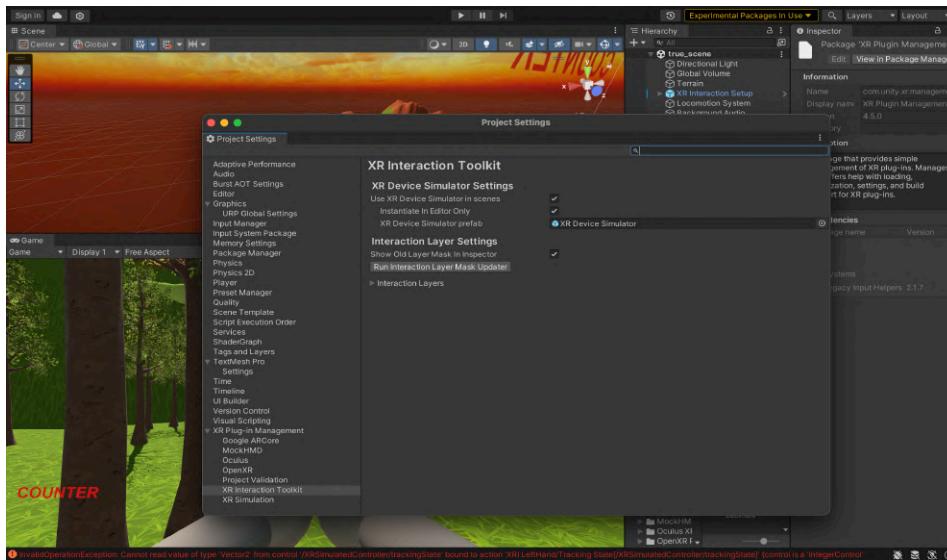
Objective : To create a basic VR game with a virtual environment in Unity that includes a ground plane, a skybox, environmental objects, lighting, and simple VR interaction. The player should be able to grab and move the grabale objects in the environment to score points.

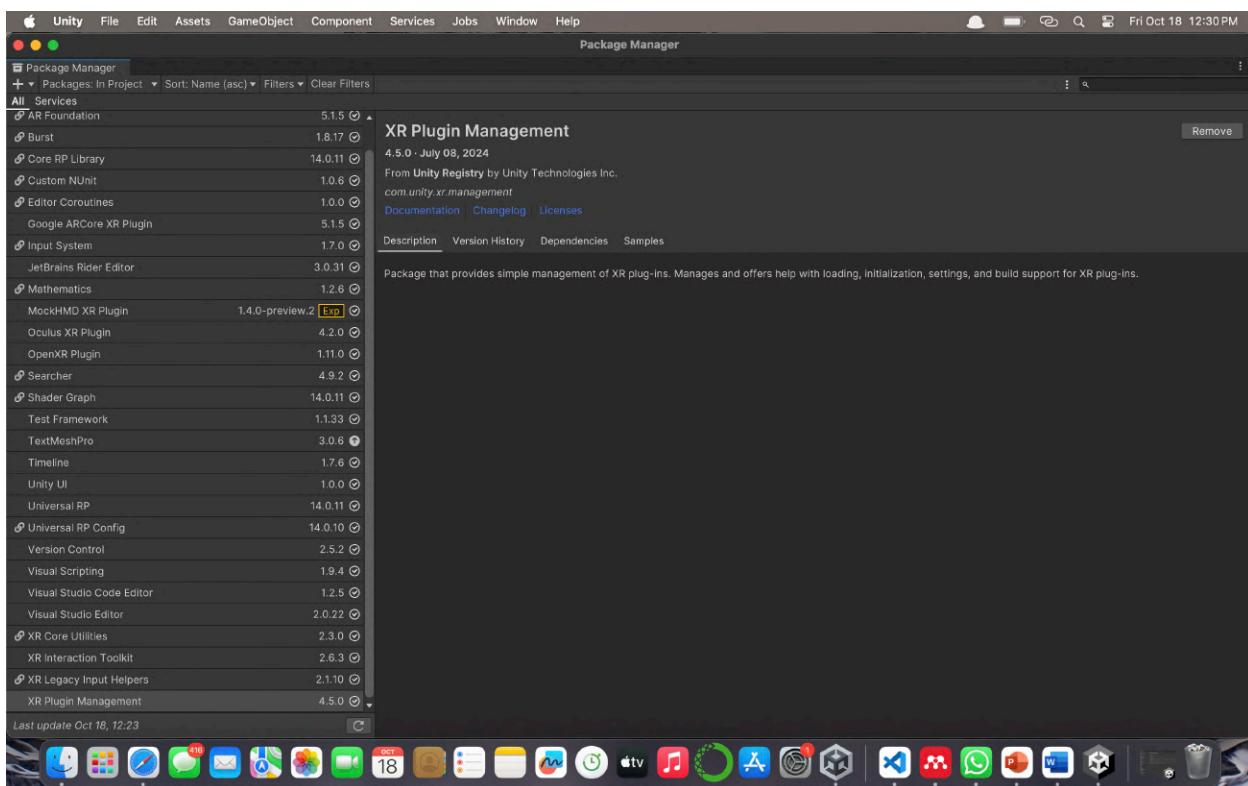
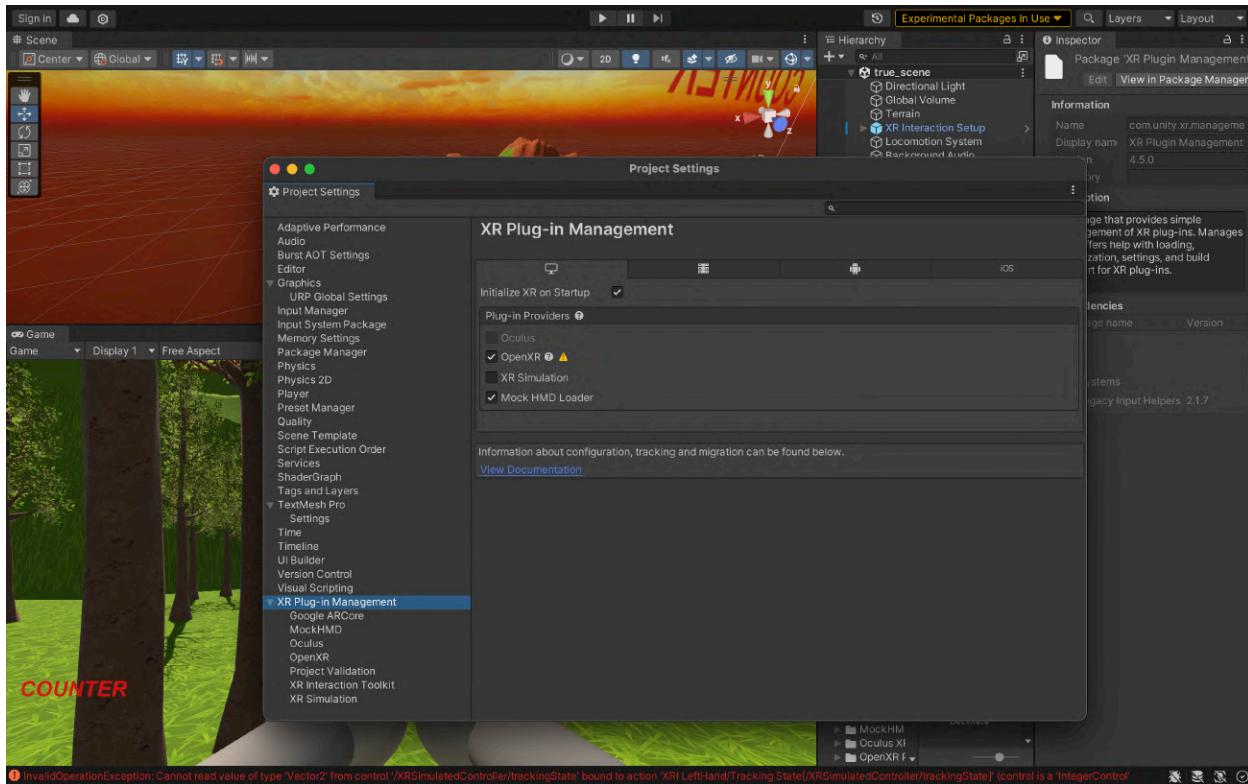
## **TASK 1 : Setting unity project and configuring vr environment**

Steps :-

- Create a new unity project in URP {universal rendering pipeline} 3d project.
- Installing XR interaction toolkit
- Installing XR plugin management
- Setup the XR origin (VR)

Screenshots :-



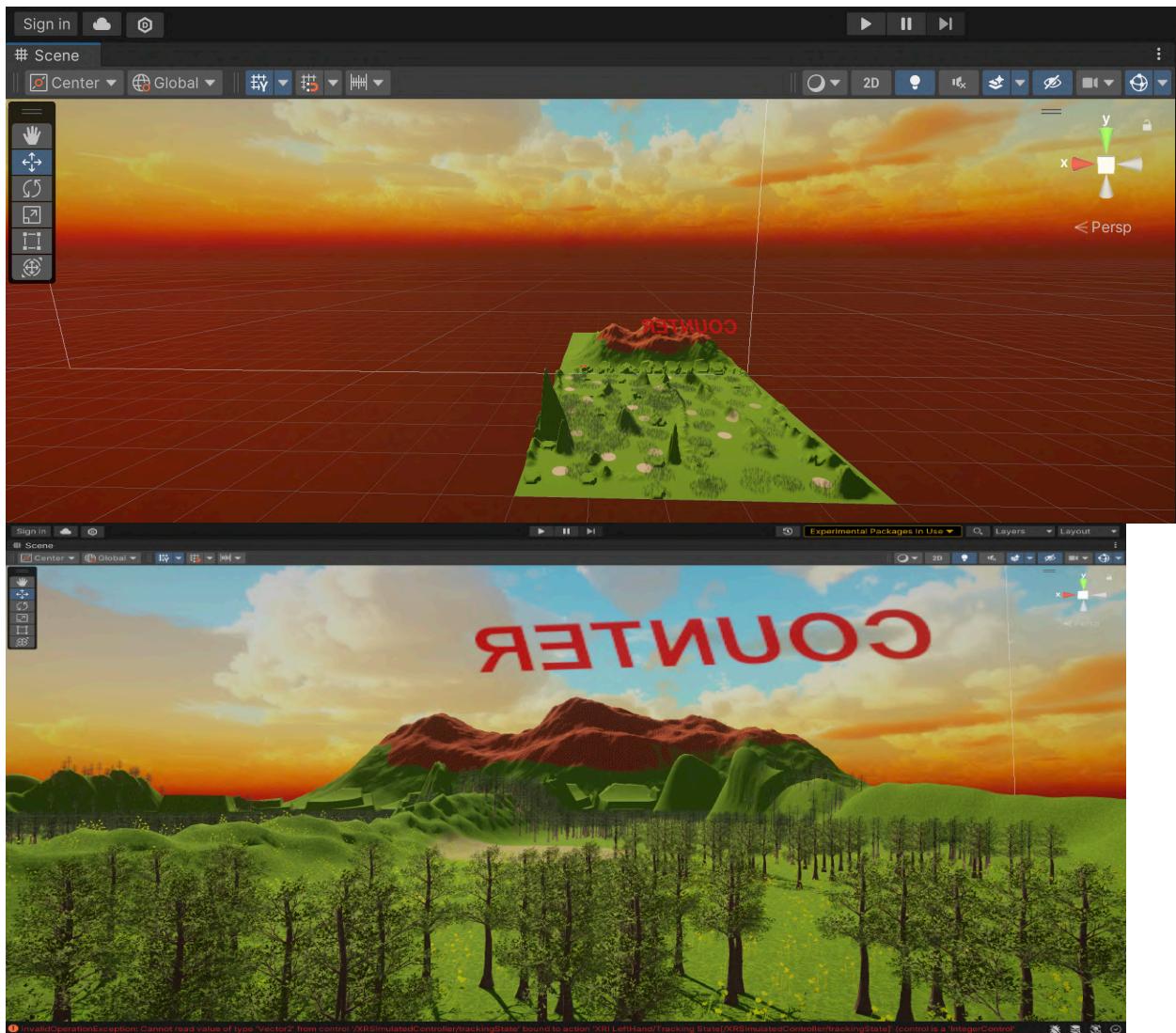


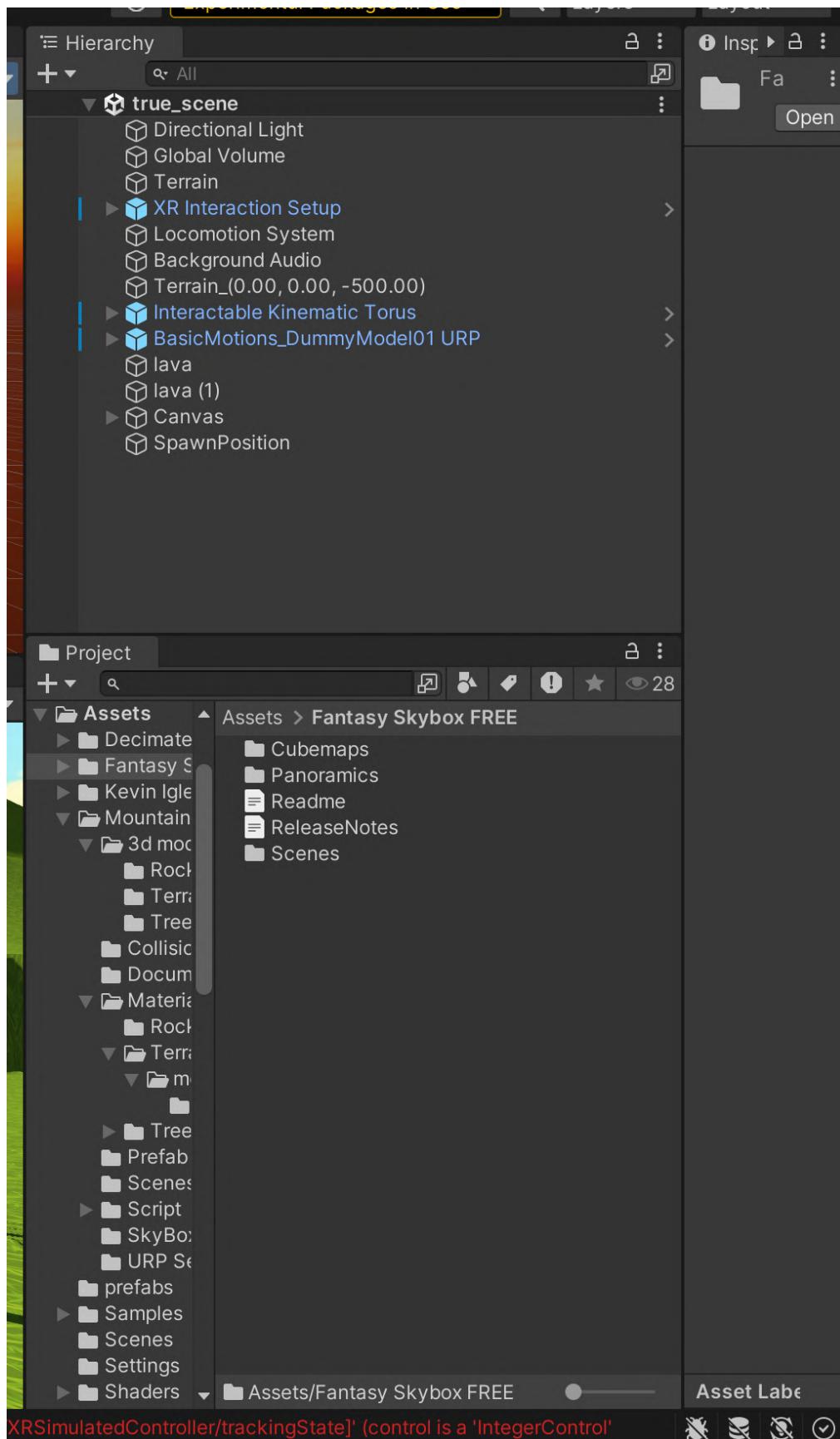
## TASK 2 : Create the ground plane and add skybox

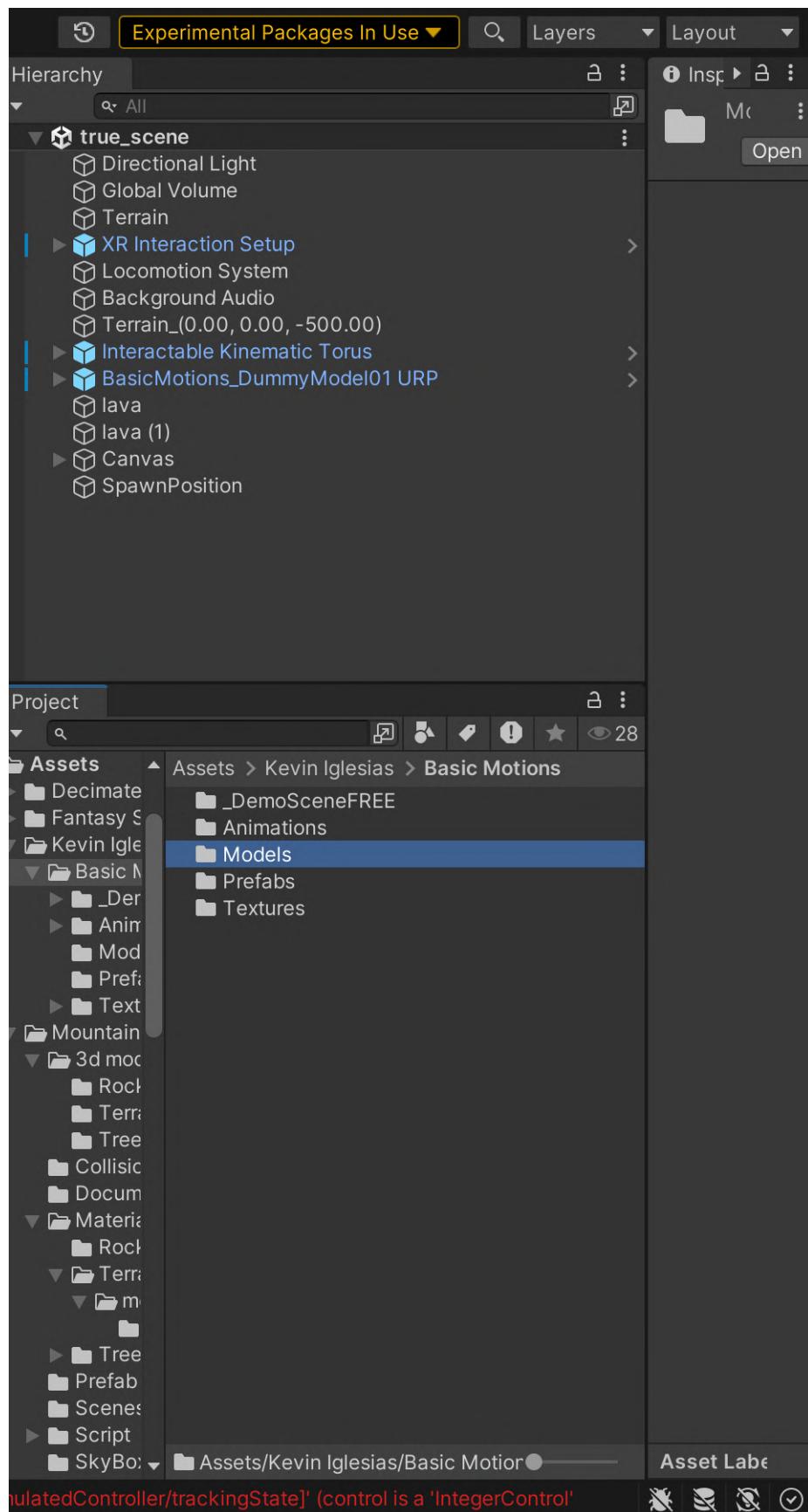
Steps:-

- Importing free asset of terrain from online unity's asset store.
- Importing free asset of skybox from online unity's asset store.
- With the help of terrain brush made the bulges on the terrains to look like mountains and used the hole painter of unity to create a hole for lava.

Screenshots:-





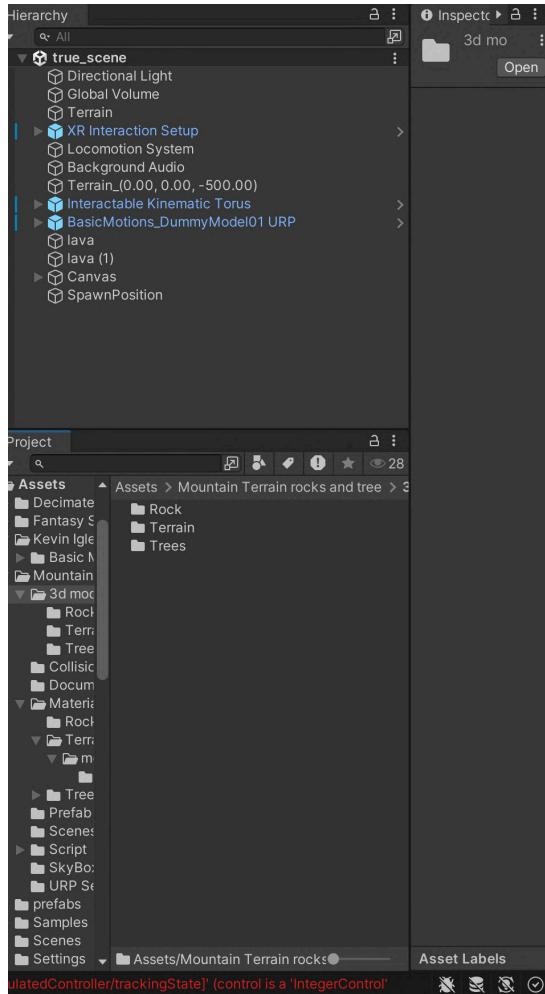


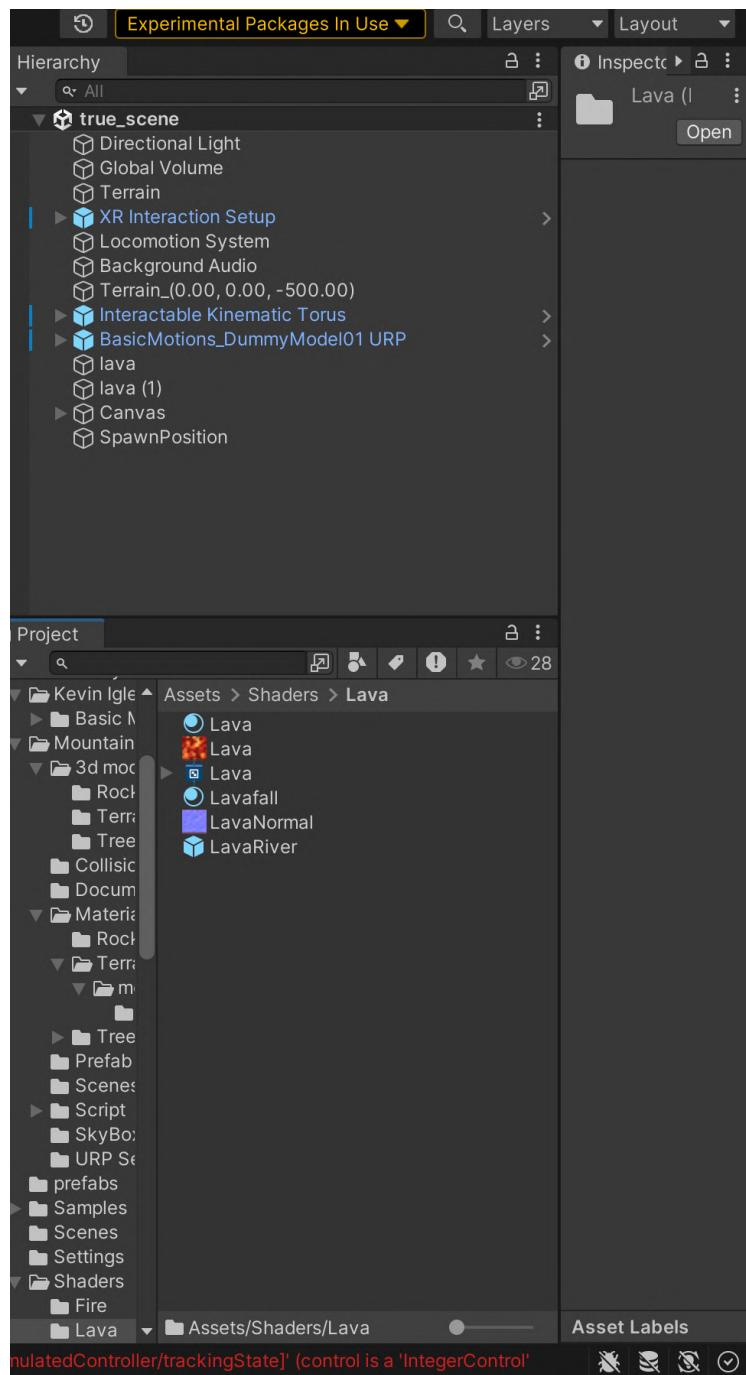
## TASK 3 : Add environment object

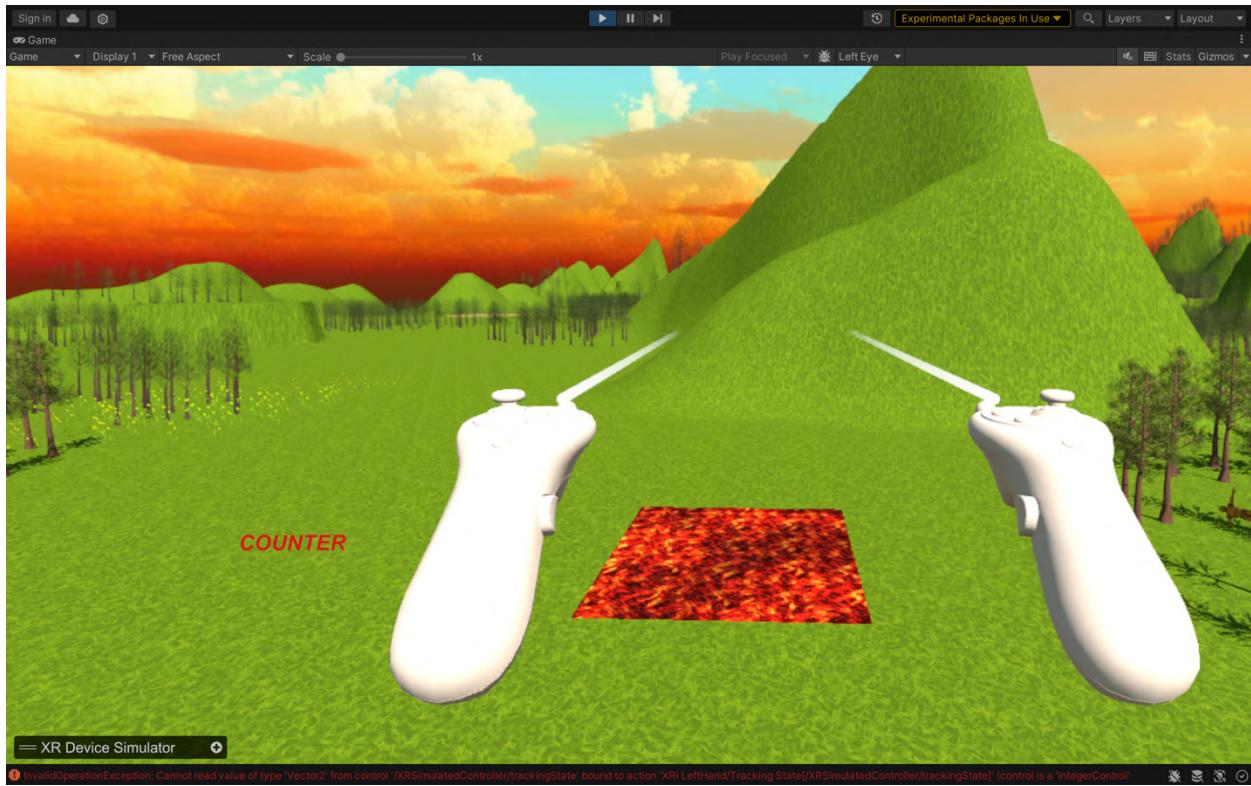
Steps:-

- Expanded the size of terrain x:100 | y:1| z:100 so that there is enough place for the player to walkthrough.
- Imported free assets of trees, flowers and rocks to make terrain look more appealing.
- Added free assets of lava to increment in the game logic and added it's flowing animation.

Screenshots:-





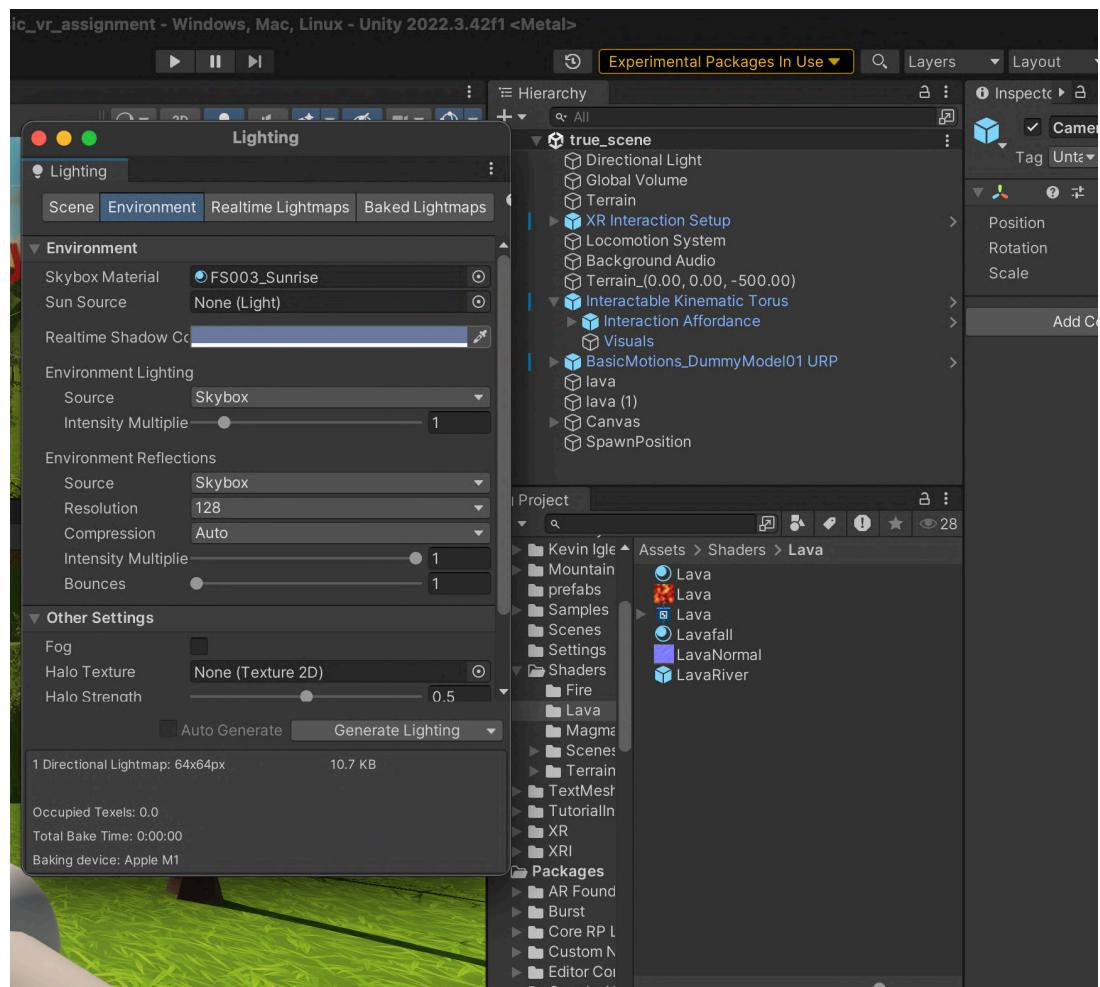


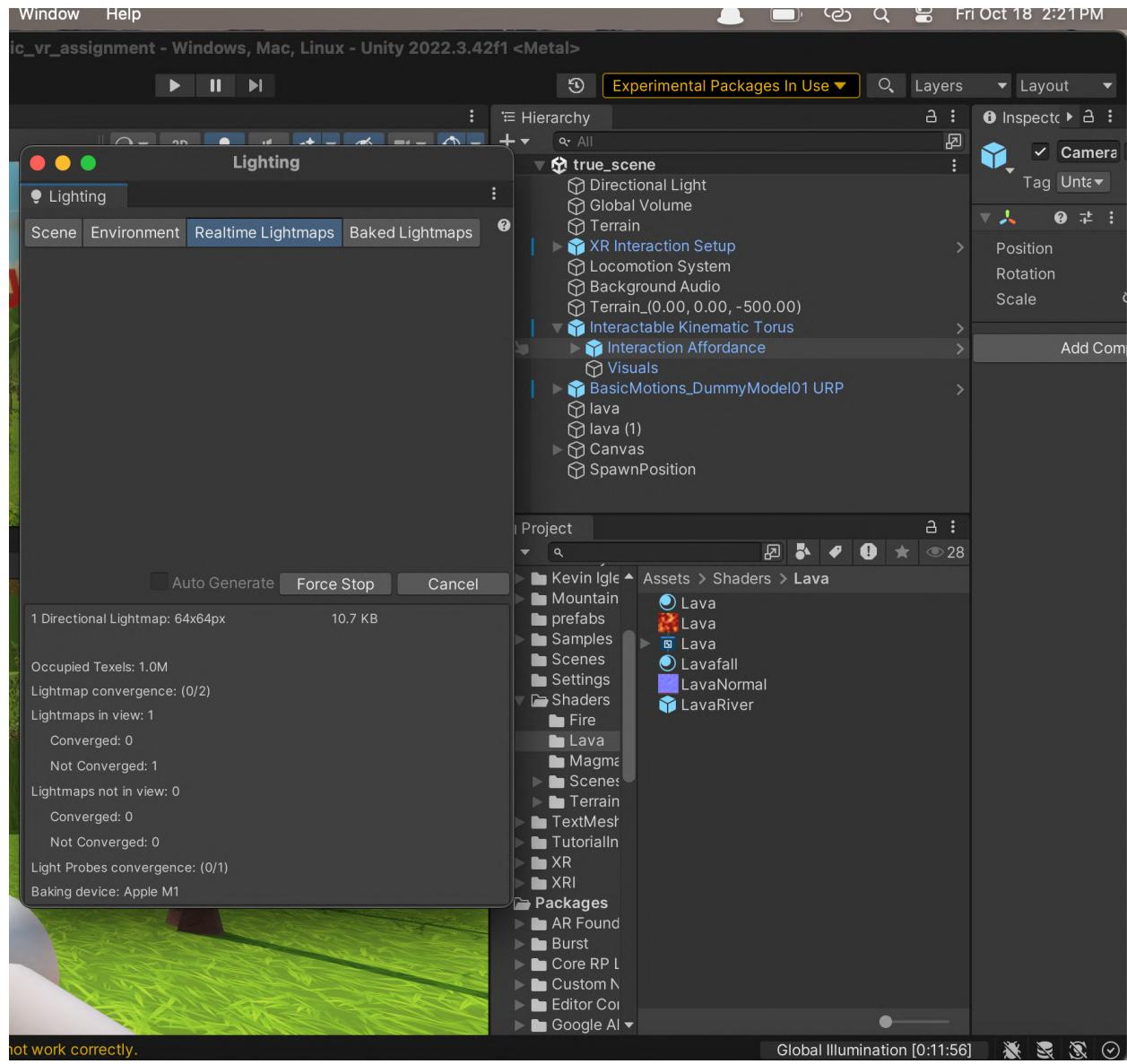
## TASK 4 : Configure lighting and shadow

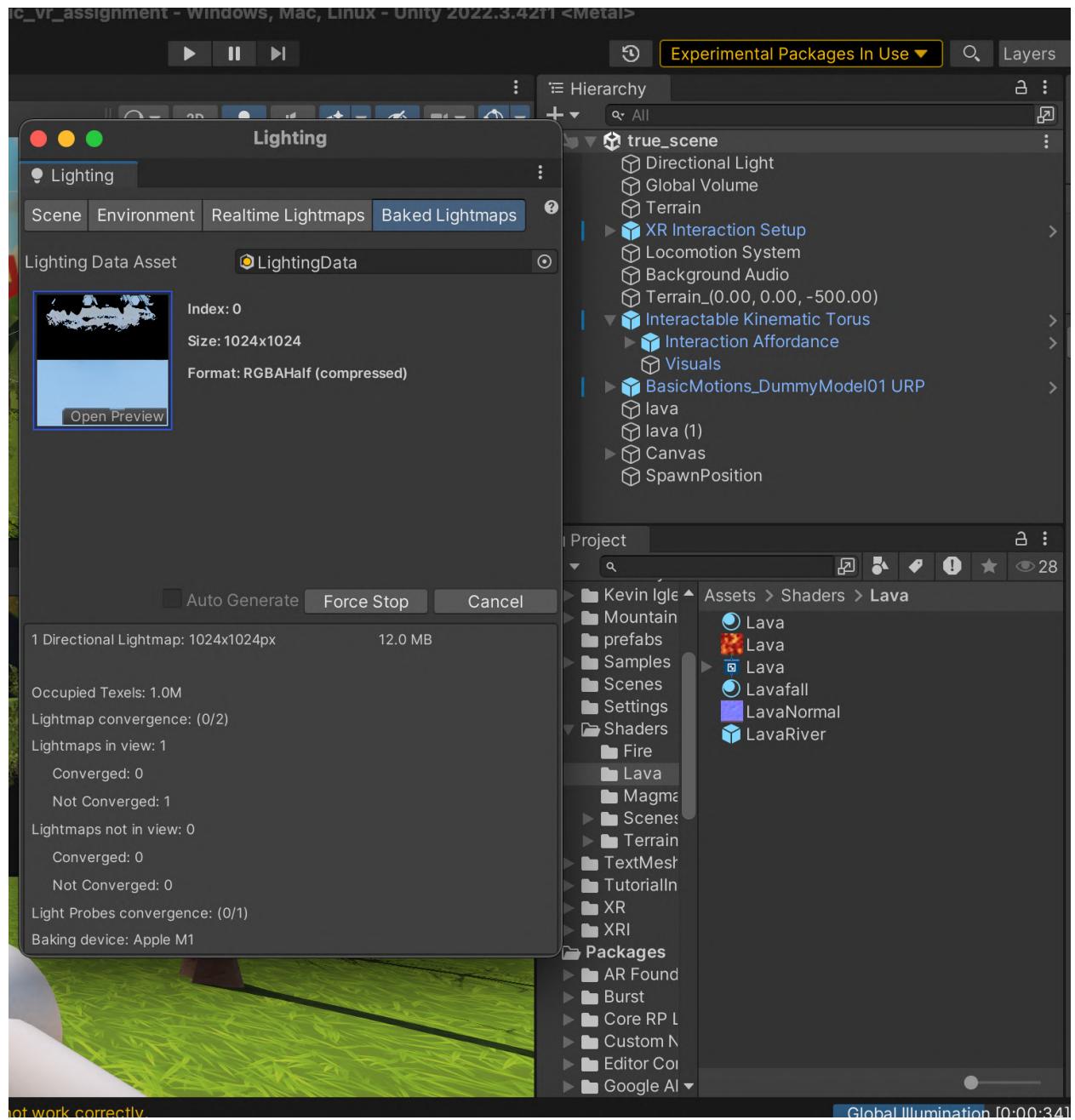
Steps:-

- Added directional light to simulate the sunlight
- Baked lightmaps
- Generated and configured real-time lightmaps

Screenshots:-





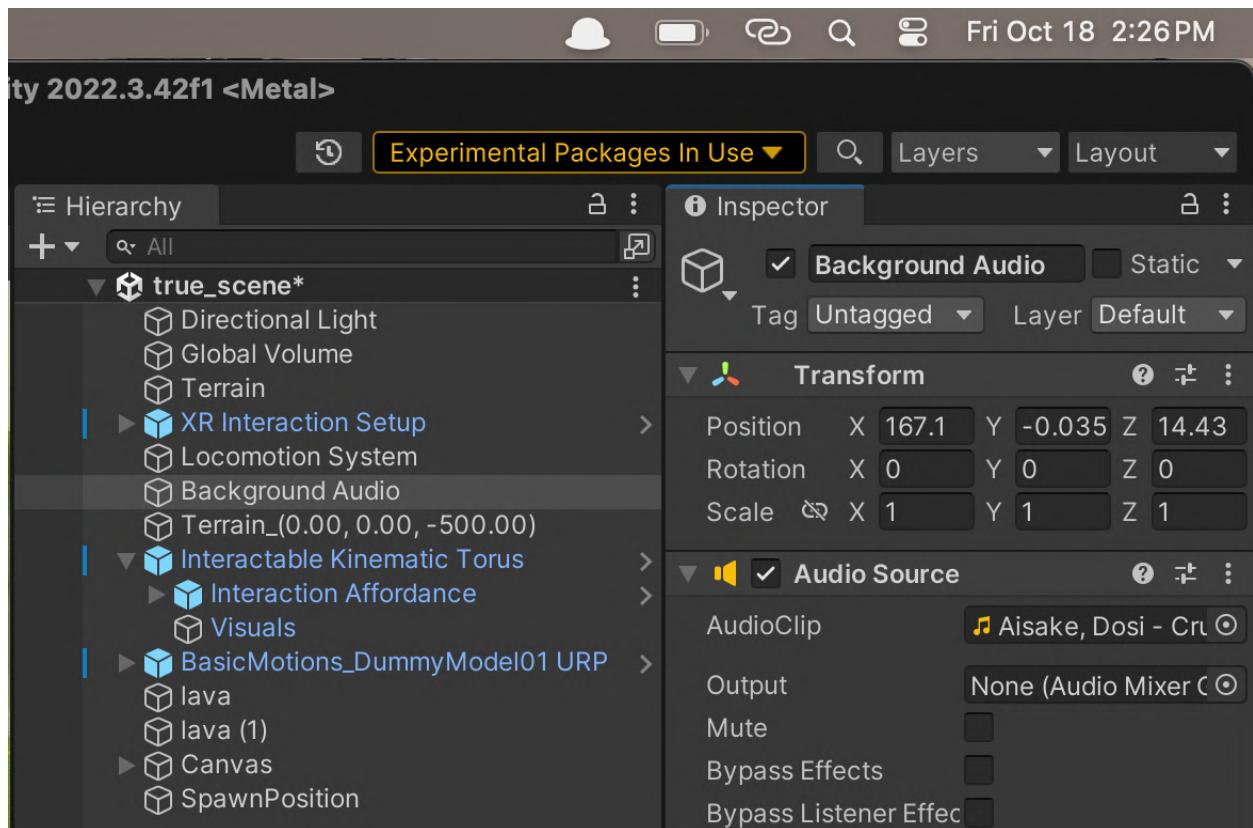


## TASK 5 : Add Audio

Steps:-

- Got a NCS (non-copyright sound) from internet
- Created an empty game object name background-music
- Added audio component in that game object and then assigned the audio to the component

Screenshots:-



Fri Oct 18 2:27 PM

2022.3.42f1 <Metal>

Experimental Packages In Use ▾

Hierarchy

- All
- true\_scene\*
- Directional Light
- Global Volume
- Terrain
- XR Interaction Setup
- Locomotion System
- Background Audio
- Terrain\_(0.00, 0.00, -500.00)
- Interactable Kinematic Torus
  - Interaction Affordance
  - Visuals
- BasicMotions\_DummyModel01 URP
  - lava
  - lava (1)
  - Canvas
  - SpawnPosition

Inspector

Cartoon, Daniel Levi, Jeja - C

Force To Mono

Normalize

Load In Background

Ambisonic

Default  iOS  Mac

Load Type Decompress On Load

Preload Audio Data\*

Compression Format Vorbis

Quality 100

Sample Rate Setting Preserve Sample Rate

\* Shared setting between multiple platforms.

Original Size: 3.2 MB

Imported Size: 3.0 MB

Ratio: 95.57%

Revert Apply

Assets

- Decimate
- Fantasy S
- Kevin Iglesias
- Mountain
- prefabs
- Samples
- Scenes
- Settings
- Shaders
  - Fire
  - Lava
  - Magma
- Scenes
- Terrain
- TextMesh
- TutorialInn
- XR
- XRI

Assets

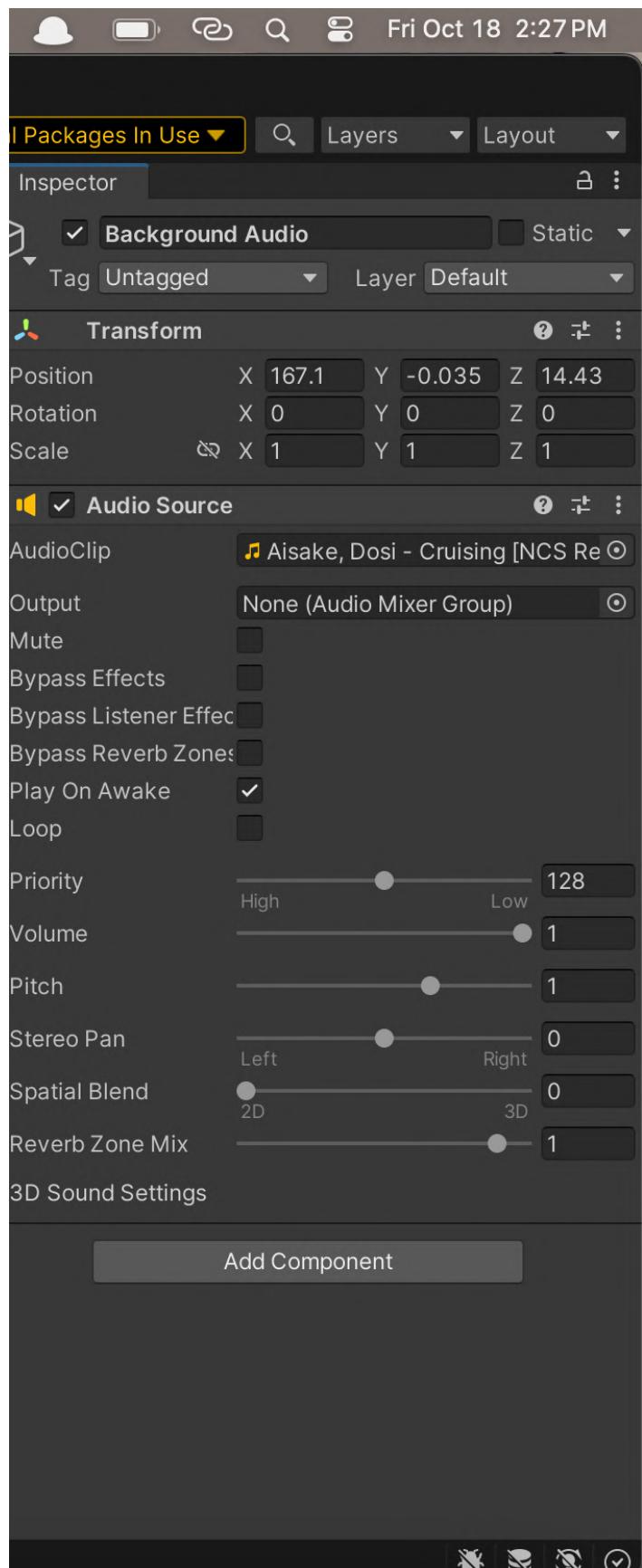
- Aisake, Dosi - Cruising [N]
- Cartoon, Daniel Levi, Jeja
- cube\_m
- Decimate
- Fantasy Skybox FREE
- Kevin Iglesias
- LavaZone
- Mountain Terrain rocks a
- New Terrain
- New Terrain 1
- New Terrain 2
- NewLayer
- prefabs
- Readme
- Samples
- Scenes
- Settings
- Shaders
- TerrainData\_a9807e6e-9
- TextMesh Pro
- TutorialInnInfo

Asset PostProcessors

UnityEditor.ShaderGraph.ShaderGraphAss

Imported Object

Cartoon, Daniel Levi, Jeja - On

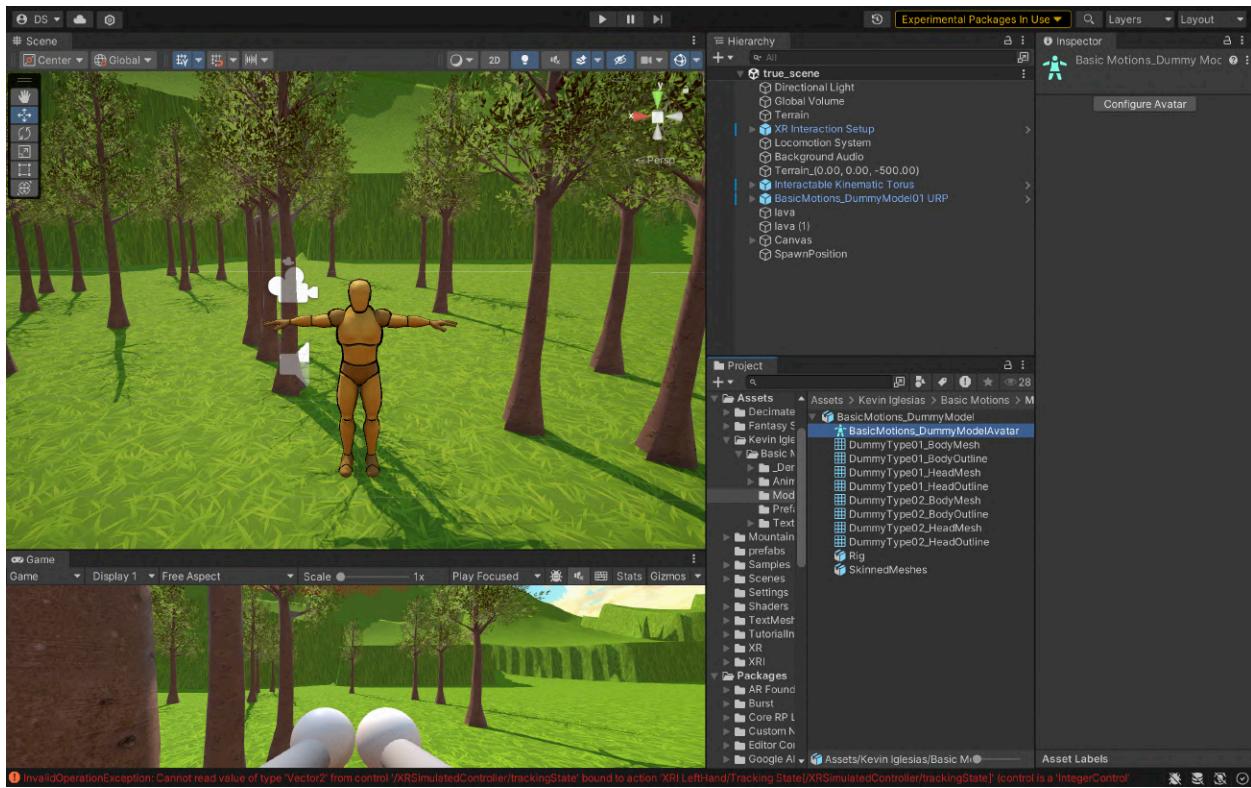


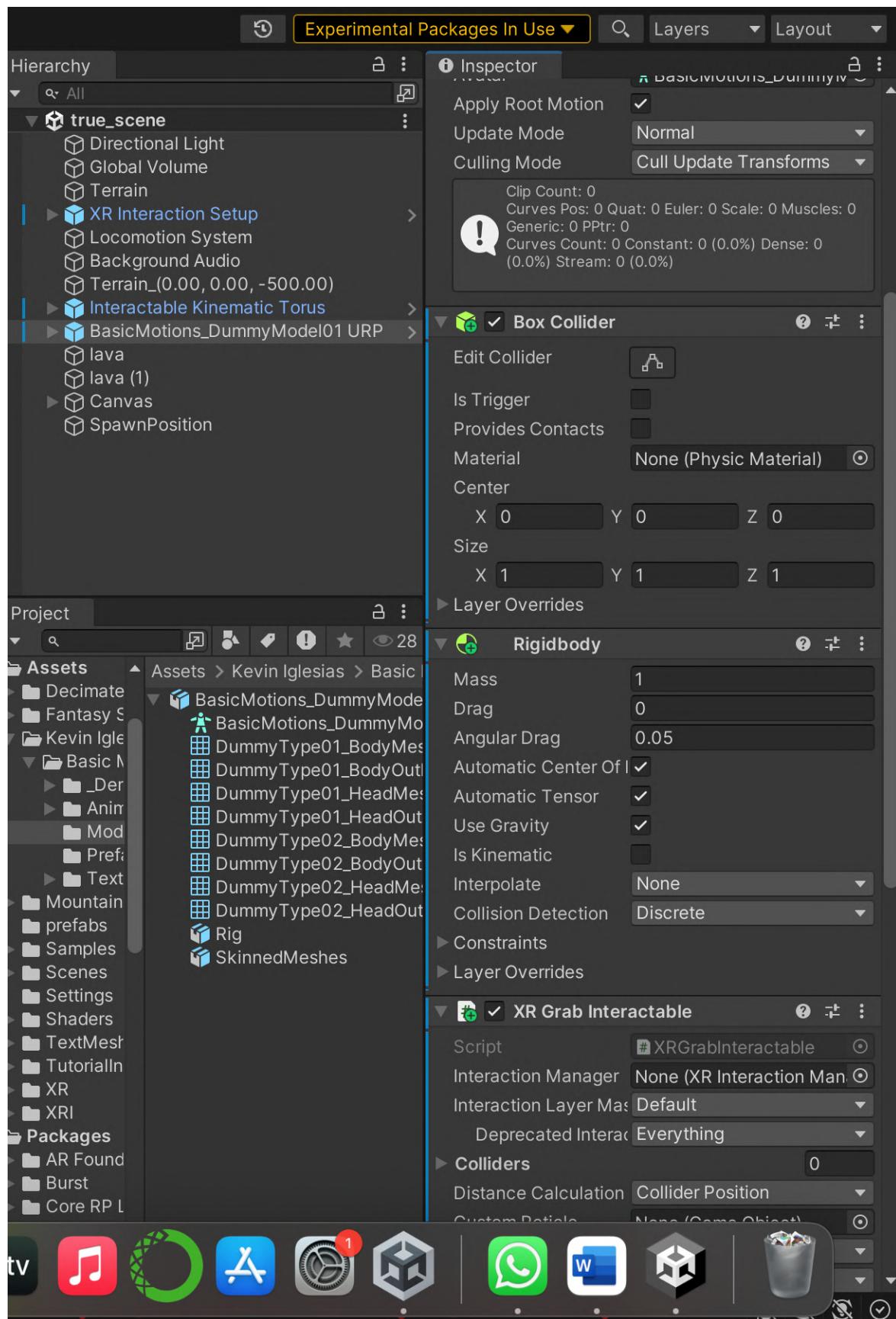
## TASK 6 : Implement basic vr interaction

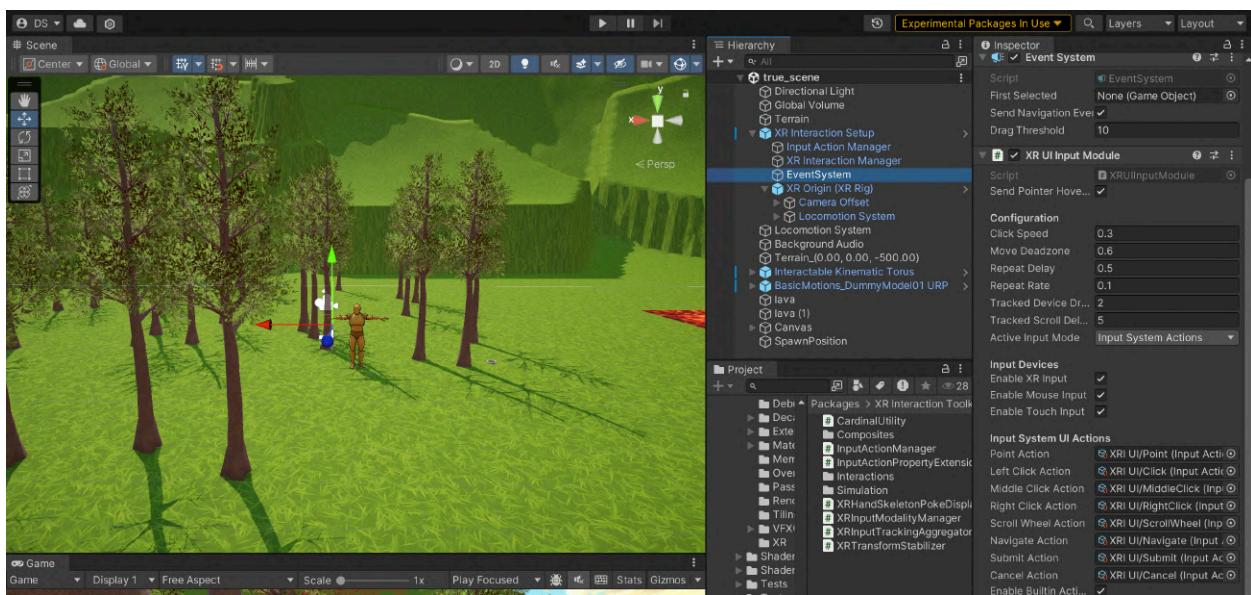
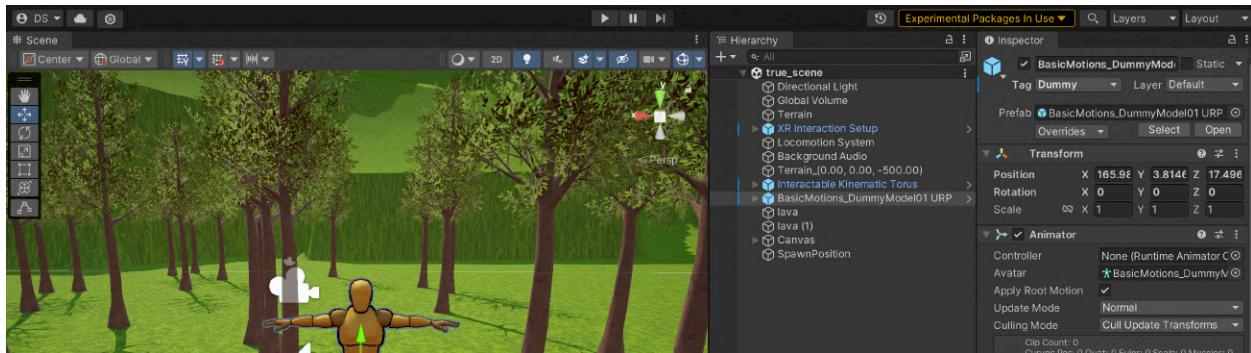
Steps:-

- Configure the grabbable object using xr toolkit's xr grab interactable script.
- Add grabber components to the controllers.
- Importing asset of dummy for free from unity asset store.
- Adding xr grabinteractable component to it.
- Add in the animations on the dummy through animators.
- Configure vr simulator's interaction.

Screenshots:-

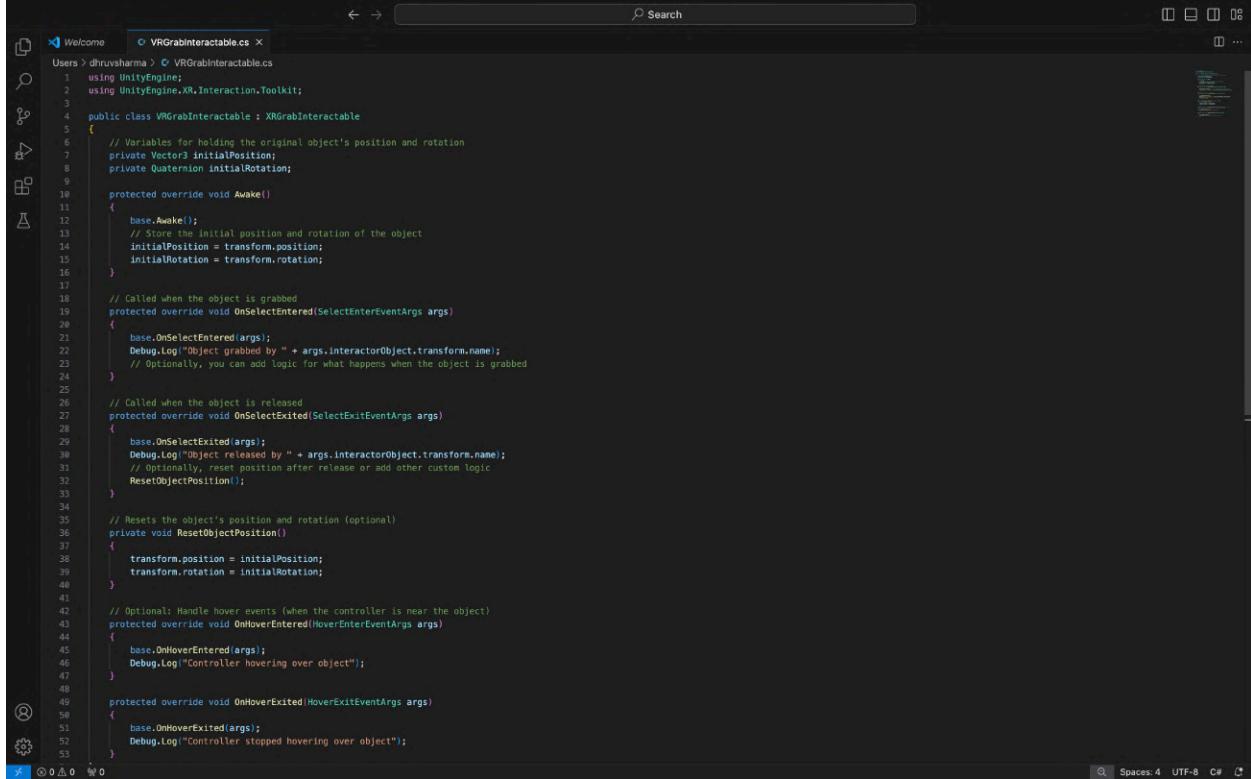






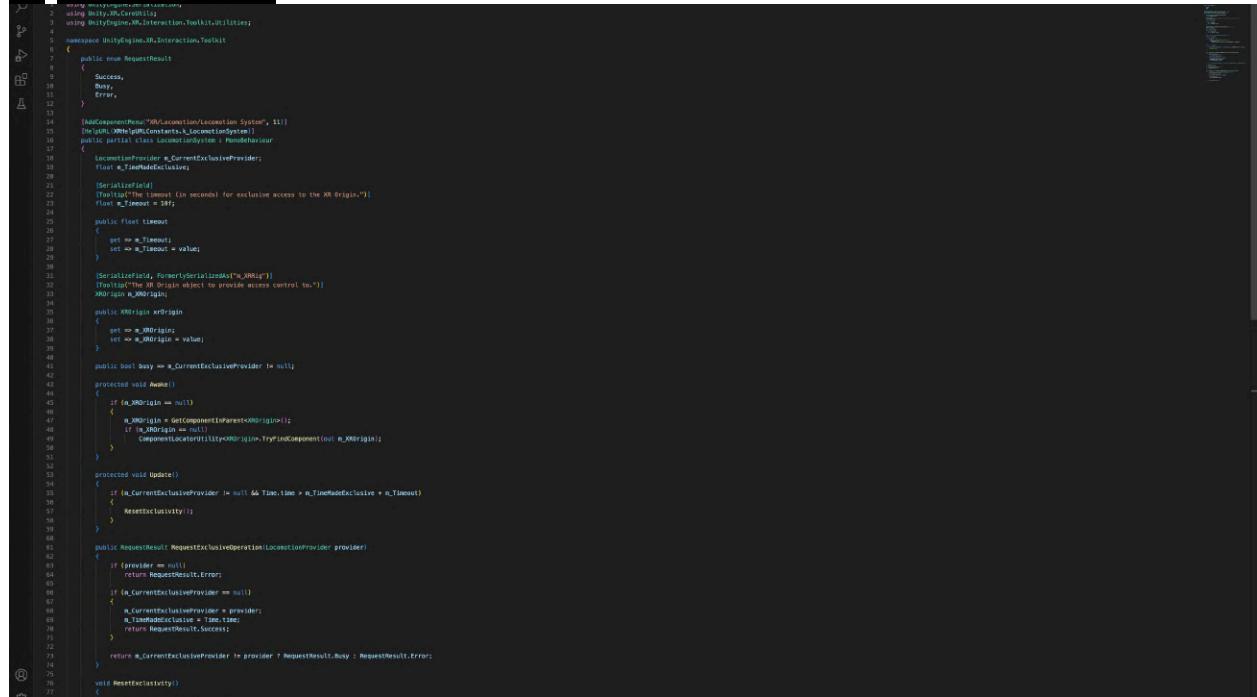
## TASK 7 : Write VR interaction script

### 1) Grab Interactable



```
1 using UnityEngine;
2 using UnityEngine.XR.Interaction.Toolkit;
3
4 public class VRGrabInteractable : XRGrabInteractable
5 {
6     // Variables for holding the original object's position and rotation
7     private Vector3 initialPosition;
8     private Quaternion initialRotation;
9
10    protected override void Awake()
11    {
12        base.Awake();
13        // Store the initial position and rotation of the object
14        initialPosition = transform.position;
15        initialRotation = transform.rotation;
16    }
17
18    // Called when the object is grabbed
19    protected override void OnSelectEntered(SelectEventArgs args)
20    {
21        base.OnSelectEntered(args);
22        Debug.Log("Object grabbed by " + args.interactorObject.transform.name);
23        // Optionally, you can add logic for what happens when the object is grabbed
24    }
25
26    // Called when the object is released
27    protected override void OnSelectExited(SelectExitEventArgs args)
28    {
29        base.OnSelectExited(args);
30        Debug.Log("Object released by " + args.interactorObject.transform.name);
31        // Optionally, reset position after release or add other custom logic
32        ResetObjectPosition();
33    }
34
35    // Resets the object's position and rotation (optional)
36    private void ResetObjectPosition()
37    {
38        transform.position = initialPosition;
39        transform.rotation = initialRotation;
40    }
41
42    // Optional: Handle hover events (when the controller is near the object)
43    protected override void OnHoverEntered(HoverEnterEventArgs args)
44    {
45        base.OnHoverEntered(args);
46        Debug.Log("Controller hovering over object");
47    }
48
49    protected override void OnHoverExited(HoverExitEventArgs args)
50    {
51        base.OnHoverExited(args);
52        Debug.Log("Controller stopped hovering over object");
53    }
}
```

### 2) Locomotion



```
1 using UnityEngine;
2 using UnityEngine.XR.Interaction.Toolkit;
3
4 namespace UnityEngine.XR.Interaction.Toolkit
5 {
6     public enum RequestResult
7     {
8         Success,
9         Busy,
10        Error,
11    }
12
13    [AddComponentMenu("XR/Locomotion/Locomotion System")]
14    [SerializeField] XRMixedRealityConstants _locomotionSystem;
15    public partial class LocomotionSystem : MonoBehaviour
16    {
17        [SerializeField] XRMixedRealityProvider _currentExclusiveProvider;
18        float _timeMadeExclusive;
19
20        [SerializeField]
21        TimeSpan _timeExclusiveRequestTimeout; // Timeout for request (in seconds) for exclusive access to the XR Origin.
22        float _timeout = 3f;
23
24        public float timeout
25        {
26            get => _timeout;
27            set => _timeout = value;
28        }
29
30        [SerializableField, PomerySerializable("m_XROrigin")]
31        [HideInInspector("The XR Origin object to provide access control to.")] XRMixedReality _xrOrigin;
32
33        public XRMixedReality xrOrigin
34        {
35            get => _xrOrigin;
36            set => _xrOrigin = value;
37        }
38
39        public bool busy => _currentExclusiveProvider != null;
40
41        protected void Awake()
42        {
43            if (_xrOrigin == null)
44            {
45                _xrOrigin = GetComponentsInParent<XROrigin>()[0];
46                if (_xrOrigin == null)
47                    ComponentLocatorUtility<XROrigin>.TryFindComponent(out _xrOrigin);
48            }
49        }
50
51        protected void Update()
52        {
53            if (_currentExclusiveProvider != null && Time.time > _timeMadeExclusive + _timeout)
54            {
55                ResetExclusivity();
56            }
57        }
58
59        public RequestResult RequestExclusiveOperation(ILocomotionProvider provider)
60        {
61            if (provider == null)
62                return RequestResult.Error;
63
64            if (_currentExclusiveProvider == null)
65            {
66                _currentExclusiveProvider = provider;
67                _timeMadeExclusive = Time.time;
68                return RequestResult.Success;
69            }
70
71            return _currentExclusiveProvider != provider ? RequestResult.Error : RequestResult.Error;
72        }
73
74        void ResetExclusivity()
75        {
76            _currentExclusiveProvider = null;
77        }
78    }
}
```

### 3) Re-spawnner and counter [LAVAZONE]

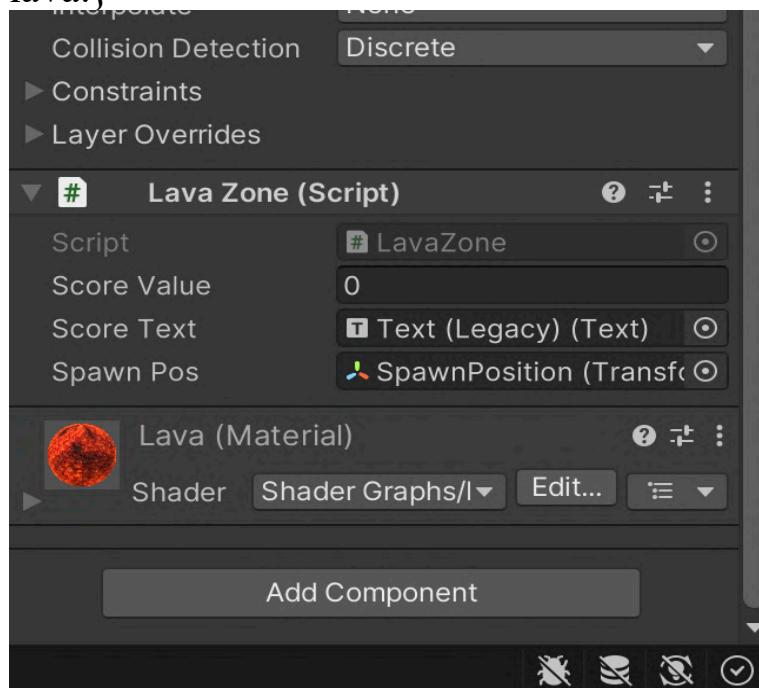
The screenshot shows the Unity Editor interface. The top bar has tabs for Welcome, VRGrabInteractable.cs, InputActionManager.cs, and LavaZone.cs (which is currently selected). The code editor displays the LavaZone.cs script:

```
1  using UnityEngine;
2  using UnityEngine.UI; // Required for UI elements like Text
3
4  public class LavaZone : MonoBehaviour
5  {
6      public int scoreValue = 0; // Score given when dummy hits the zone
7      public Text scoreText; // Reference to the UI Text element
8      public Transform spawnPos;
9
10     private void OnTriggerEnter(Collider other)
11     {
12         if (other.CompareTag("Dummy")) // Check if the object entering the zone is the dummy
13         {
14             scoreValue += 10; // Increment score
15             UpdateScoreText(); // Update the text UI
16             Debug.Log("Dummy entered the lava! Score: " + scoreValue);
17
18             other.transform.position = spawnPos.position;
19         }
20     }
21
22     // Function to update the UI Text element
23     private void UpdateScoreText()
24     {
25         if (scoreText != null)
26         {
27             scoreText.text = "Score: " + scoreValue.ToString(); // Update the UI with the new score
28         }
29     }
30 }
31
32 }
```

The Inspector panel below shows the component settings for the Lava Zone (Script) component:

- Collision Detection: Discrete
- Constraints
- Layer Overrides
- Script: LavaZone
- Score Value: 0
- Score Text: Text (Legacy) (Text)
- Spawn Pos: SpawnPosition (Transform)

{This particular script consists of both respawn at the same location and increment on the counter when the dummy is disposed in the lava.}



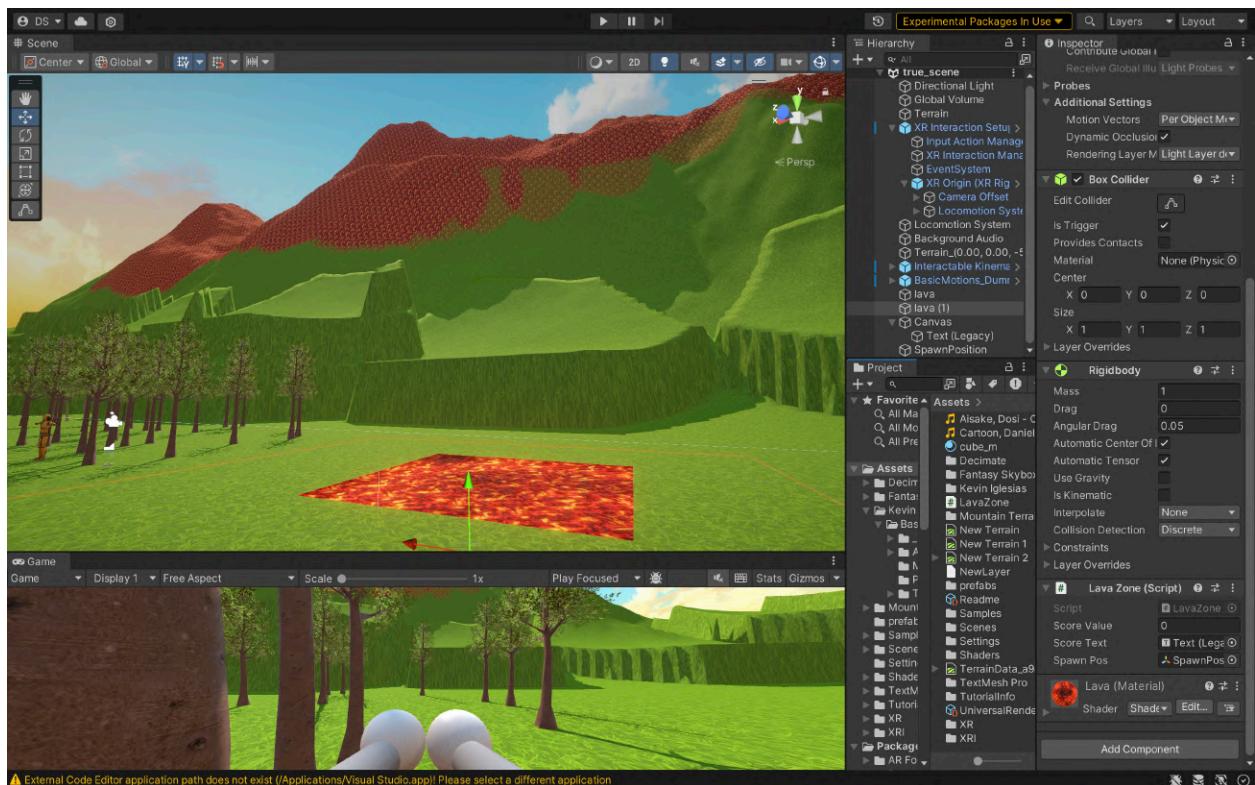
## TASK 8 : Create a scoring mechanism

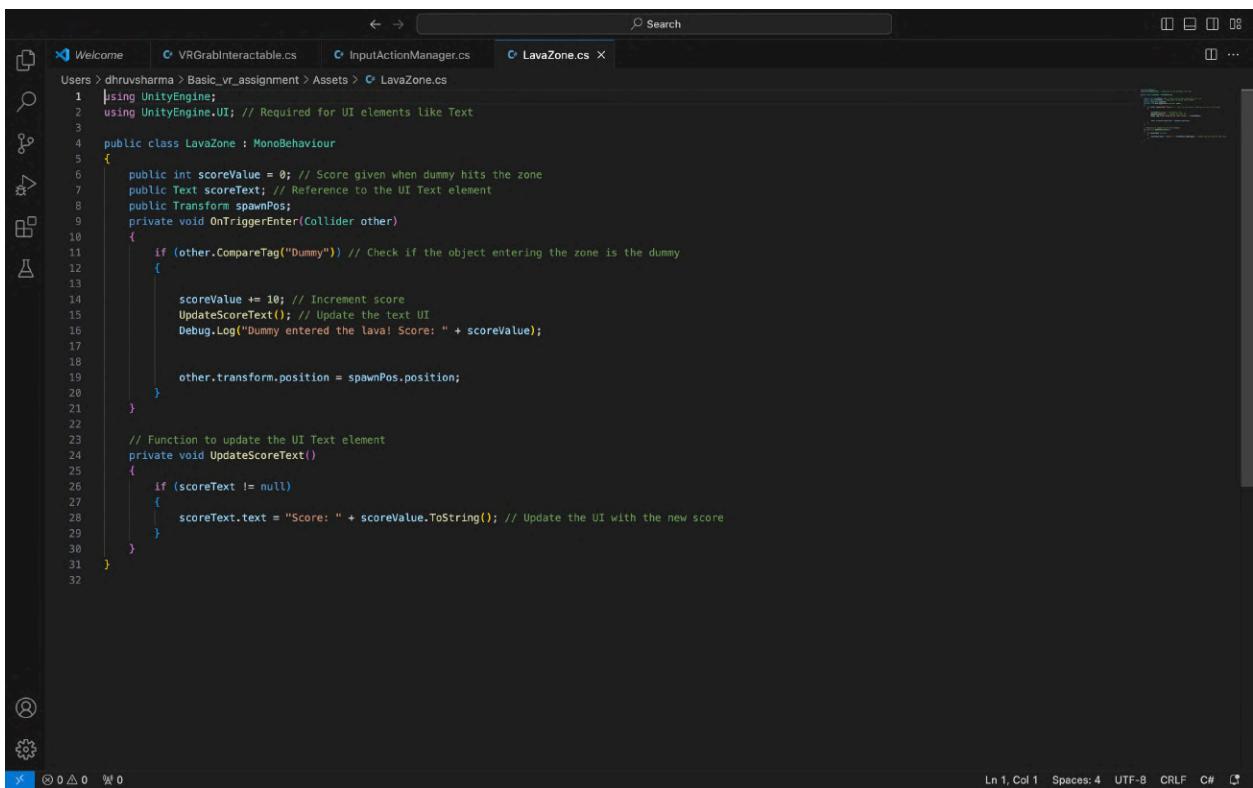
Created a scoring mechanism where when a player grabs the grabbable dummy and throws it through the lava pit the score increments by 10.

Steps :-

- Create a lava pit by erasing some terrain blocks and replacing them with a lava block.
- Keep the box to on trigger and keep the lava passable.
- Adding the script LavaZone as assign the prefabs to the variables created in the script for them.
- Detect the dummy passing through with help of OnTriggere event.
- Display the score to the player using legacy text and add this text to the variable position in the script LavaZone.

Screenshots:-





The screenshot shows a code editor window with the file `LavaZone.cs` open. The code is a C# script for a Unity MonoBehaviour. It includes imports for `UnityEngine` and `UnityEngine.UI`. The script defines a class `LavaZone` with methods for handling collisions and updating a UI text element.

```
1  using UnityEngine;
2  using UnityEngine.UI; // Required for UI elements like Text
3
4  public class LavaZone : MonoBehaviour
5  {
6      public int scoreValue = 0; // Score given when dummy hits the zone
7      public Text scoreText; // Reference to the UI Text element
8      public Transform spawnPos;
9
10     private void OnTriggerEnter(Collider other)
11     {
12         if (other.CompareTag("Dummy")) // Check if the object entering the zone is the dummy
13         {
14             scoreValue += 10; // Increment score
15             UpdateScoreText(); // Update the text UI
16             Debug.Log("Dummy entered the lava! Score: " + scoreValue);
17
18             other.transform.position = spawnPos.position;
19         }
20     }
21
22     // Function to update the UI Text element
23     private void UpdateScoreText()
24     {
25         if (scoreText != null)
26         {
27             scoreText.text = "Score: " + scoreValue.ToString(); // Update the UI with the new score
28         }
29     }
30 }
31
32 }
```

