Experiment No. 5	
Develop a scene in Unity that includes a sphere and plane. Apply Rigid body component, material and Box collider to the game Objects.Write a C# program to grab and throw the sphere using vr controller	
Date of Performance:	
Date of Submission:	
Marks:	
Sign:	

Aim: Develop a scene in Unity that includes a sphere and plane. Apply Rigid body component, material and Box collider to the game Objects. Write a C# program to grab and throw the sphere using vr controller

Theory:

Creating the Scene:
1. Create a plane, sphere & a cube; scale them as the picture.
2. Apply different materials to them.
3. The cube will be used as a table on which the sphere will be placed upon.
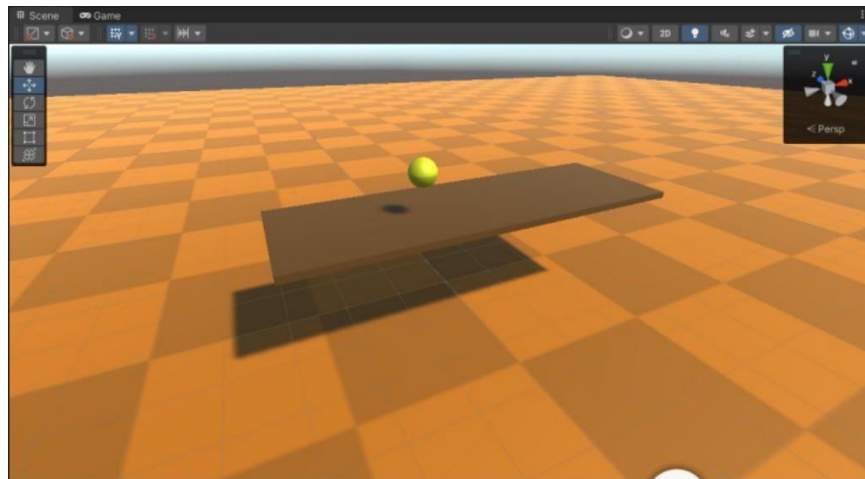
Physics: Colliders
1. Unity handles collision between GameObjects with Colliders, which attach to GameObjects and define the shape of a GameObject for the purposes of physical collisions.
2. A collider is invisible, and does not need to be the exact same shape as the GameObject's mesh.
3. The simplest (and least processor- intensive) colliders are primitive collider types. In 3D, these are theBox Collider, Sphere Collider and Capsule Collider.
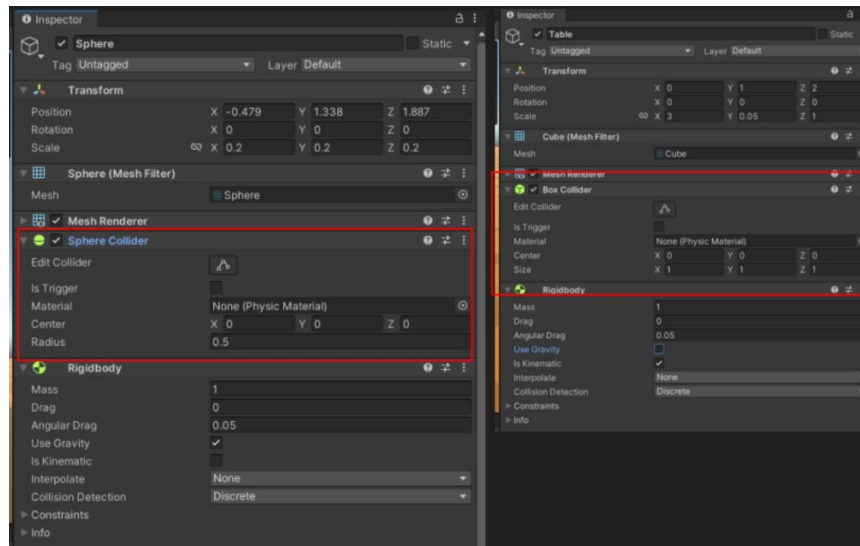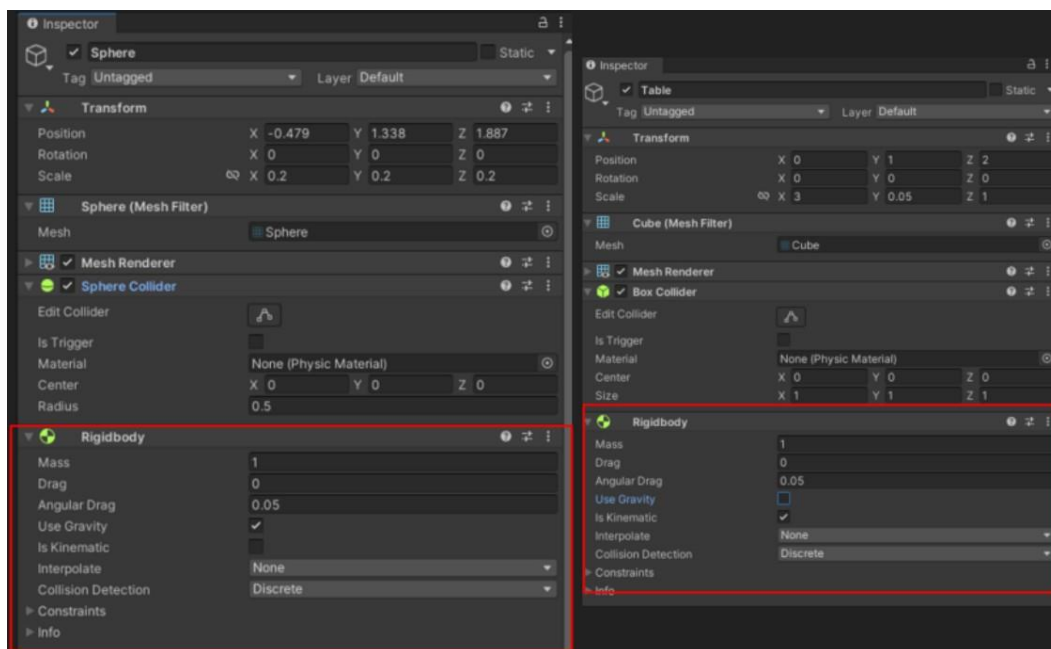
Physics: Rigidbody
1. Rigidbodies are components that allow a GameObject to react to real-time physics.
2. This includes reactions to forces and gravity, mass, drag and momentum.
3. The 2 vital flags of RigidBody are: Use Gravity & Is Kinematic.
4. If isKinematic is enabled, Forces, collisions or joints will not affect the rigidbody anymore. The rigidbody will be under full control of animation or script control by changing transform.position.
5. Use Gravity controls whether gravity affects this rigidbody. If set to false the rigidbody will behave as in outer space.
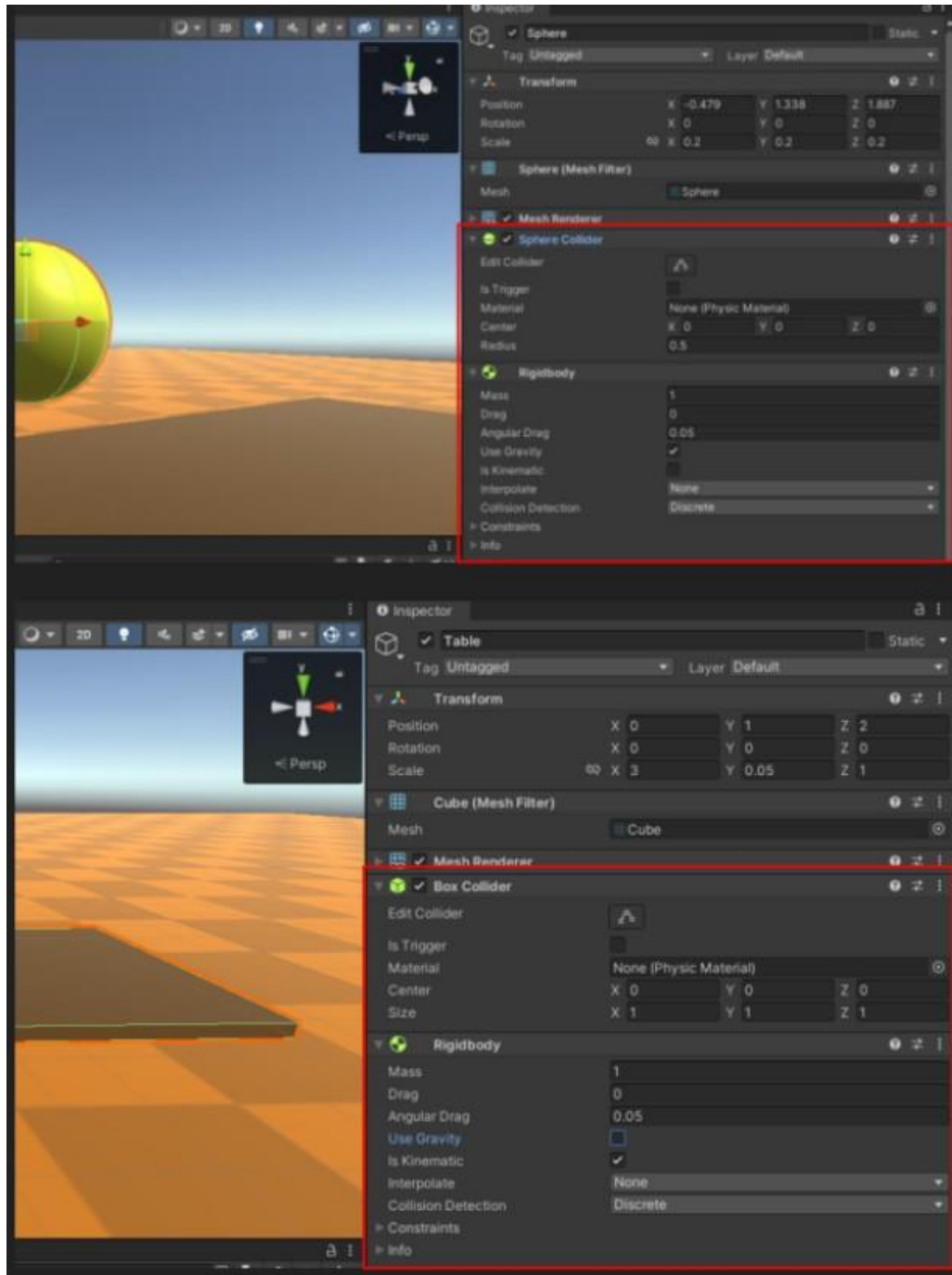
Physics Application:

1. Setup the Transform, Colliders & Rigidbody components of the sphere & cube as following images & Play the scene.
2. The sphere will fall down on the table & stay there.
3. The physics calculations are done by Unity engine.

VR by using XR Interaction Toolkit:

1. Unity has created a package called "XR Interaction Toolkit" by which we can create VR enabled applications for a wide range of VR Headsets.
2. To install the XR Interaction Toolkit package go to Window -> Package Manager.
3. Click on the "Packages:Unity Project" dropdown in the top left corner & select "Unity Registry".
4. Click on the plus icon on the top left corner of the package manager window & click "Add package by name" & type the following:
5. com.unity.xr.interaction.toolkit
6. After installation, update the package if necessary.
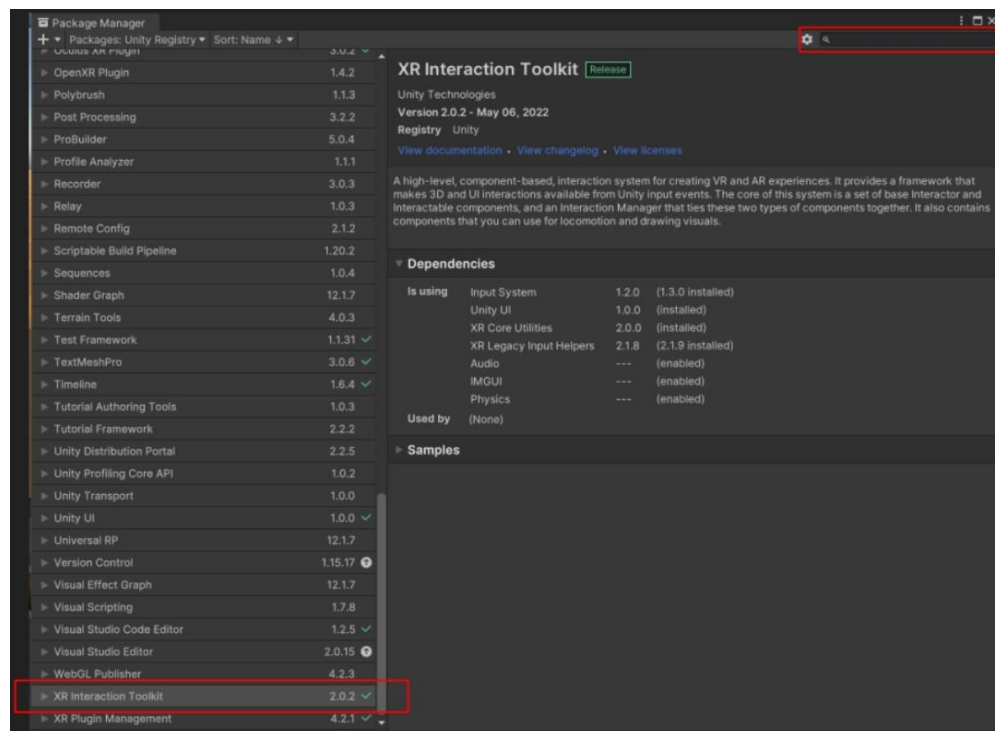
1. Additionally search & install these packages if they aren't already installed:
2. ○ XR Plugin Management
3. ○ Oculus XR Plugin
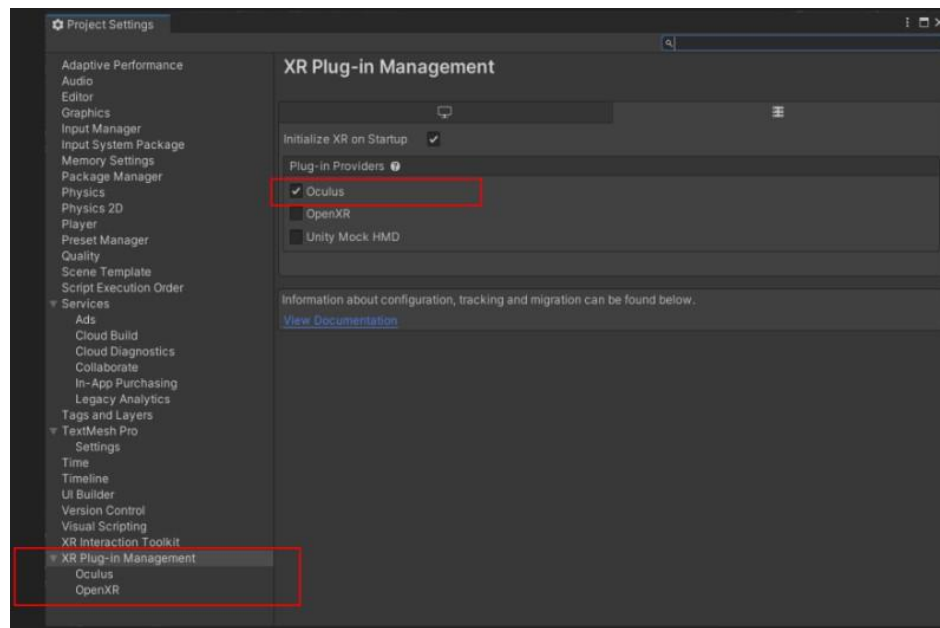4. Unity might ask to restart the project, click yes if it does.

VR by using XR Interaction Toolkit:

1. Now close the package manager window & go to Edit -> Project settings.
2. Click on XR Plug In Management Tab in the bottom left.
3. On the right side, Check the oculus checkbox as said in the Image.

Creating the XROrigin:

1. The XR Origin represents the center of worldspace in an XR scene.

2. The purpose of the XR Origin is to transform objects and trackable features to their final position, orientation, and scale in the Unity scene. It specifies an Origin, a Camera Floor Offset Object, and a Camera.

3. In the hierarchy, right click -> XR -> Device Based -> XROrigin.

4. Set the Transform position of the XROrigin at 0, 0, 0. Set the Tracking Origin Mode field of the XR Origin component to Floor.

5. Expand the hierarchy of XROrigin (Tip: left arrow besides the object in Hierarchy).

6. Select the LeftHand Controller & remove the XR Ray Interactor, XR Interactor Line Visual, Line Renderer components from it. Add XR Direct Interactor & Sphere Collider to it. Set the Sphere Collider values acc to the image. Do the same for the RightHand Controller.

Left hand controller's Inspector
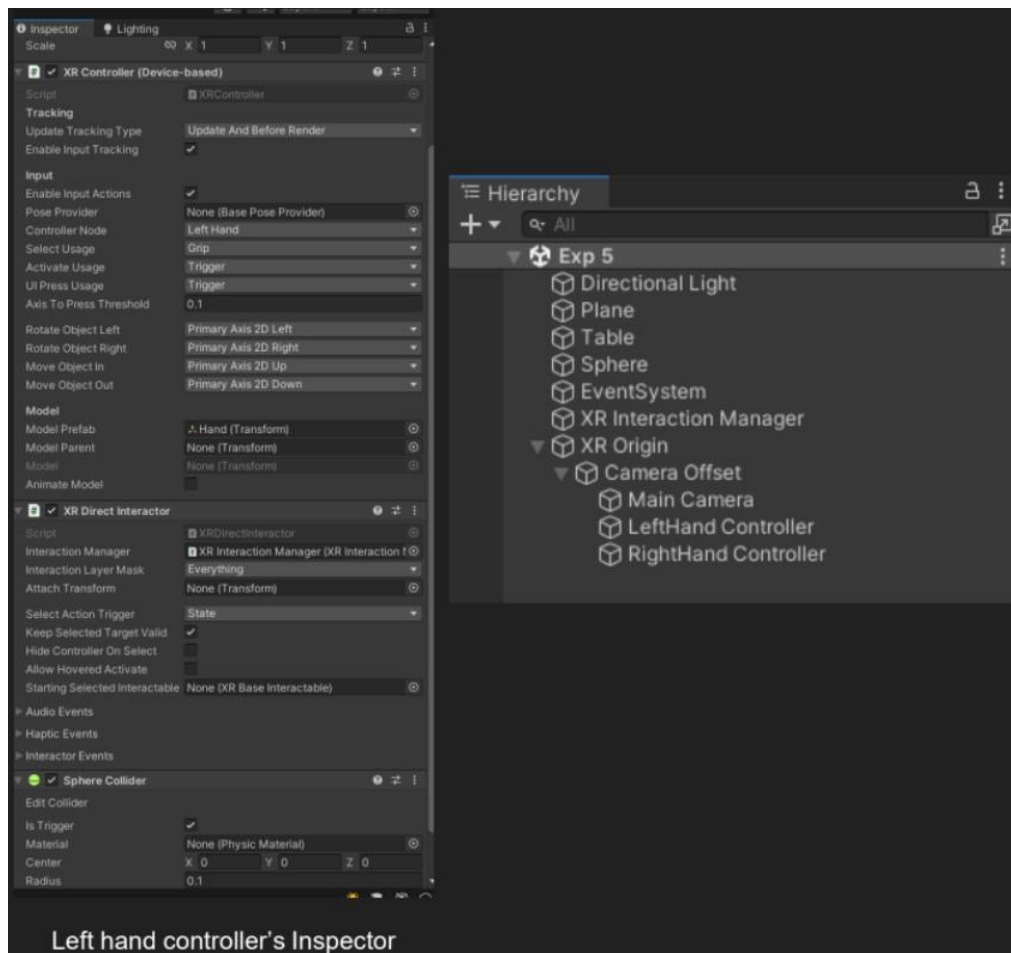
Play & Check:
● Do the setup of the oculus link software. Software Link - https://store.facebook.com/quest/setup/
● Connect the Oculus Headset to the PC with a wire & Play the game.
● If everything is setup correctly, you should be able to see the camera move in Game scene when you wear the Oculus headset.
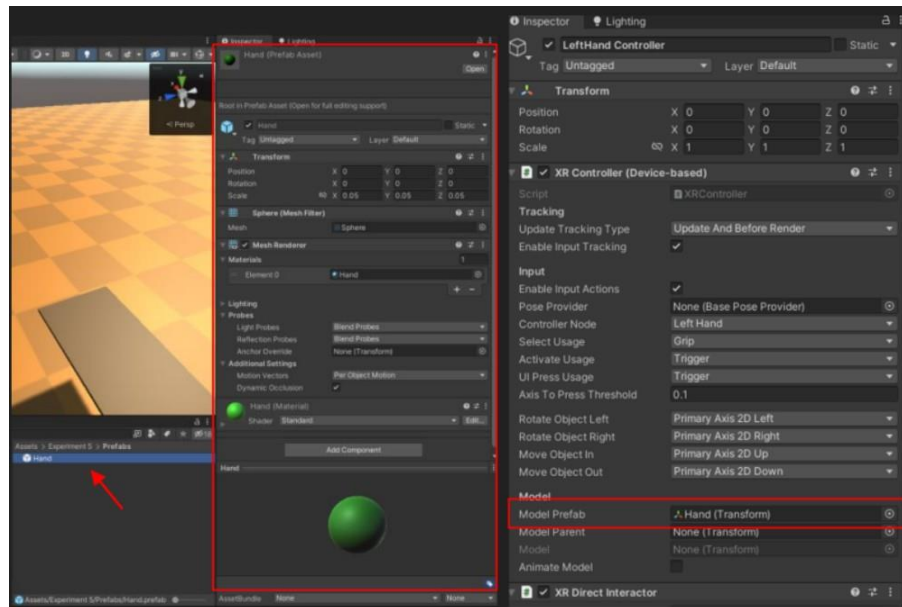
Adding Hands:
1. As of now we can't see our controllers (hands) so we will solve that next.
2. Create a sphere in the scene.
3. Give it a material.
4. Remove the sphere collider from it.
5. Set it's position at 0,0,0 & Scale the game object down to 0.1, 0.1, 0.1 (Tip:-
Transform)
6. Create a Prefab of it & delete it from the scene.
7. Now assign the prefab to the Model Prefab field of Left Hand Controller & Right Hand Controller.
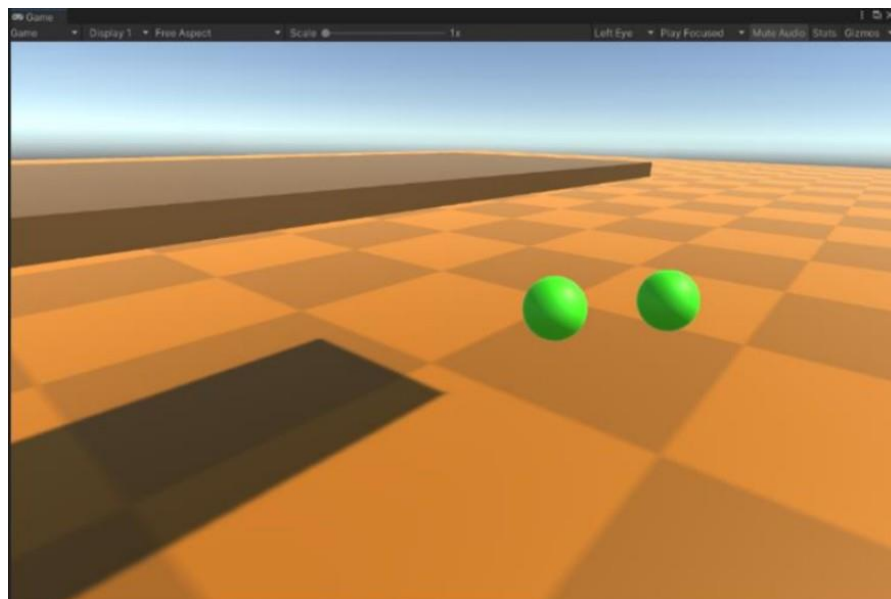
Play the Game:
● Play the Game & watch the hands in game.
● Notice how they move when you move the controllers in real time.

Adding Locomotion:
1. To move in game we need to add a few components.
2. To the XROrigin gameobject, Add Component Character Controller & Character Controller Driver.
3. To the XR Interaction Manager gameobject add the following
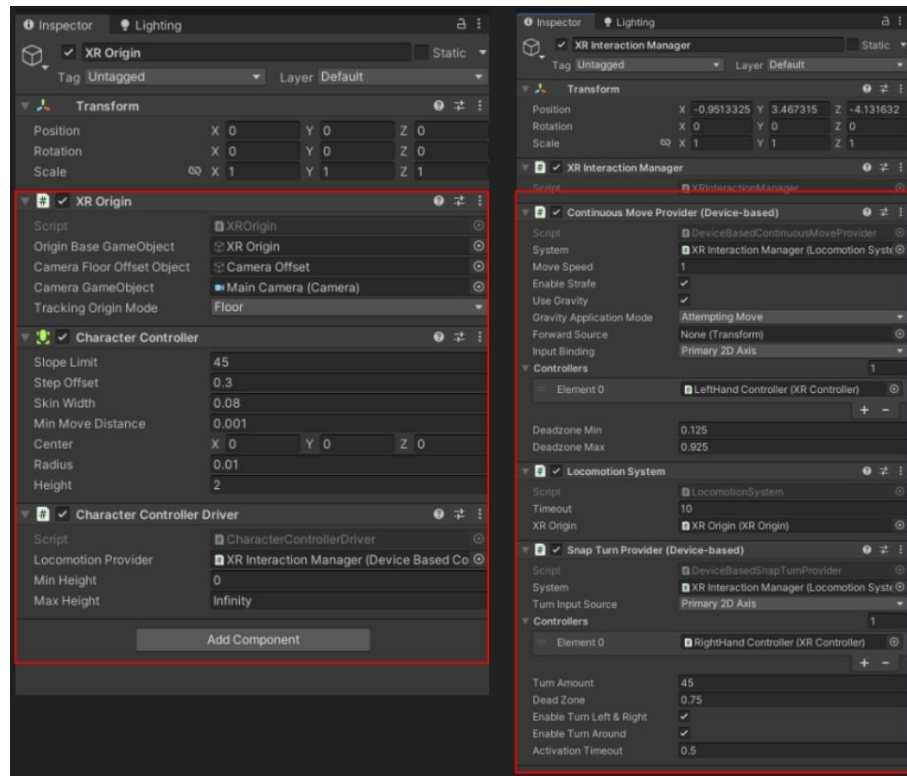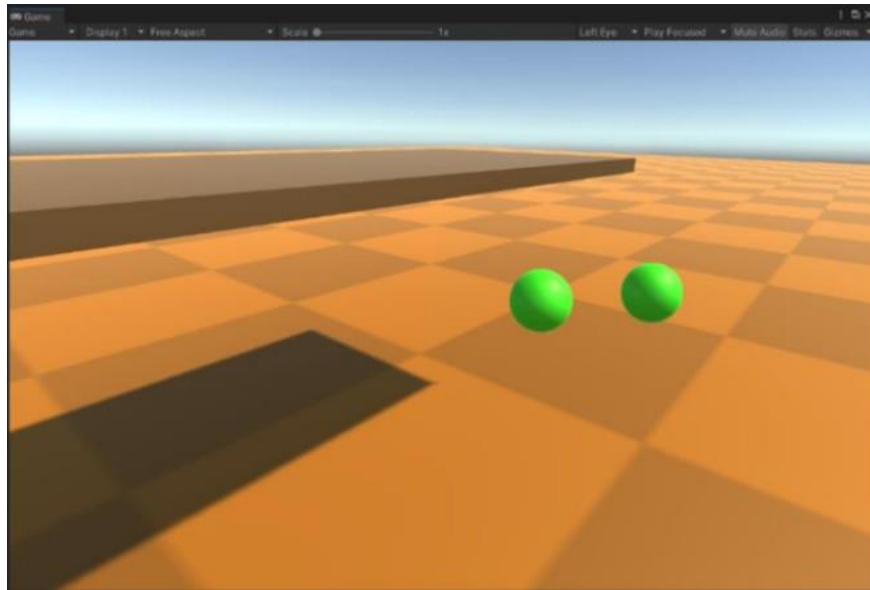components- Continuous Move Provider (Device Based), Locomotion

System, Snap Turn Provider (Device Based).
4. Set up the Continuous Move Provider & Snap Turn Provider as specified in the image.

Play the Game:
● Play the Game & move in game with the left controller's joystick.
● Turn in game with the help of right controller's joystick.

Interaction In VR:
1. In XR Interaction Toolkit, there are two main types of categories:
○ Interactors
○ Interactables
2. Interactor: An interactor is a gameobject with the interactor component with some way of interacting with the Interactables.
○ 3 main types of interactors are: Direct Interactor, Socket Interactor, Ray Interactor
3. Interactable: An interactable is an object with the Interactable component which can be interacted with by the Interactors.
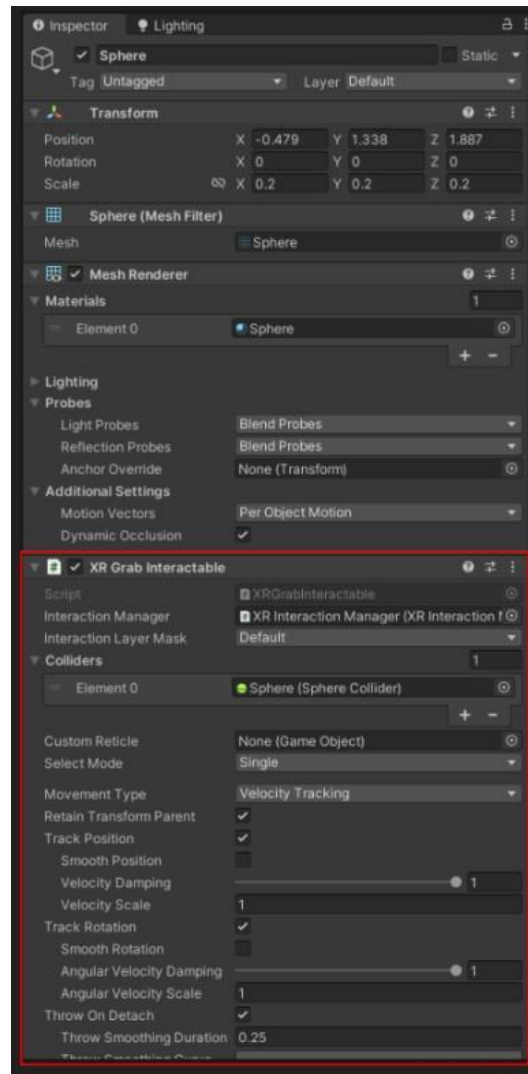○ Eg: Grab Interactable

Making the sphere grabbable:
1. Select the sphere that has gravity & add the XRGrabInteractable
component to it.
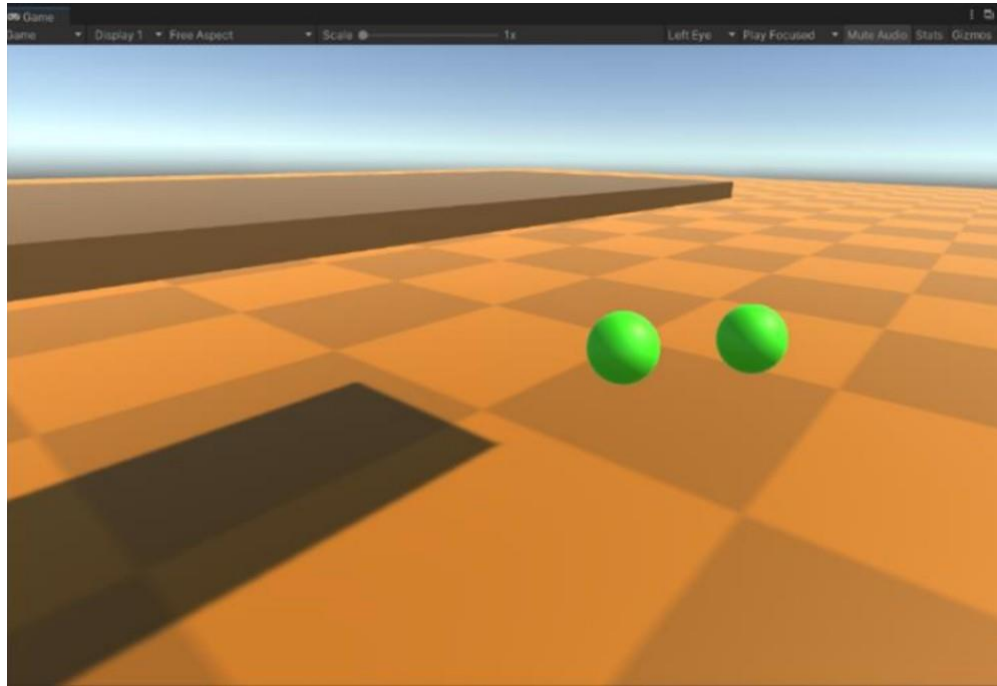2. Set it up like the image suggests.

Play the Game:
● Play the Game & try to grab the sphere from the table with the help of a controller.
● Notice that after grabbing the sphere it doesn't pass through the table.

Output:

Conclusion:-

In this Unity scene, we successfully created a basic virtual reality interaction by placing a sphere and a plane in the virtual environment. Both objects were given Rigid Body components and Box Colliders, and we applied materials to make them visually appealing. Additionally, we developed a C# program that enabled the user to grab and throw the sphere using a VR controller, achieving an interactive and immersive experience within the Unity environment. This experiment showcases the fundamental principles of VR interaction, object manipulation, and physics simulations, serving as a foundation for more complex virtual reality applications and experiences.