Experiment 0

Objective : To create a Unity scene with primitive objects (Cube, Sphere, and Cylinder) and assign distinct bounce behaviours to each object. Additional features, such as prefabs and camera configuration, are also implemented as part of this experiment.

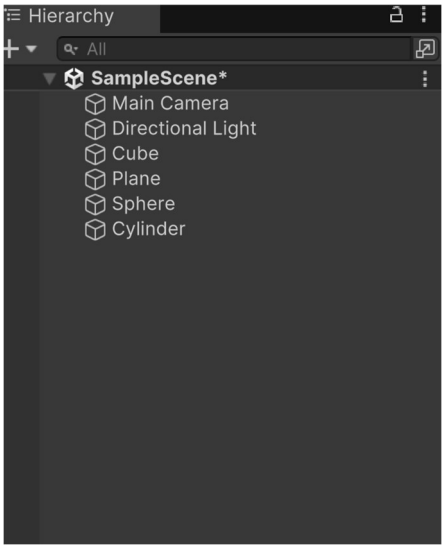
Procedure

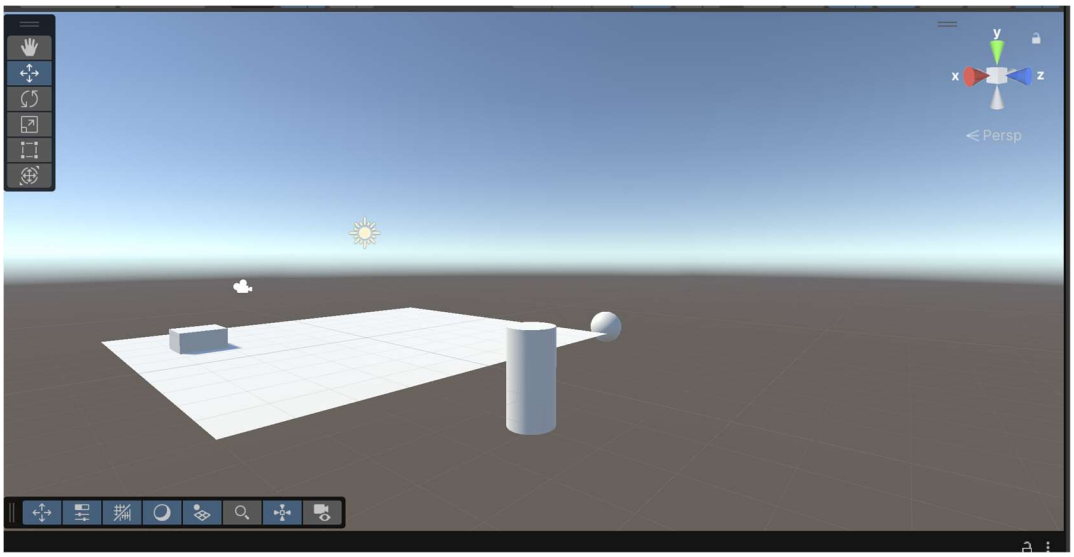
Step I: Adding a Plane

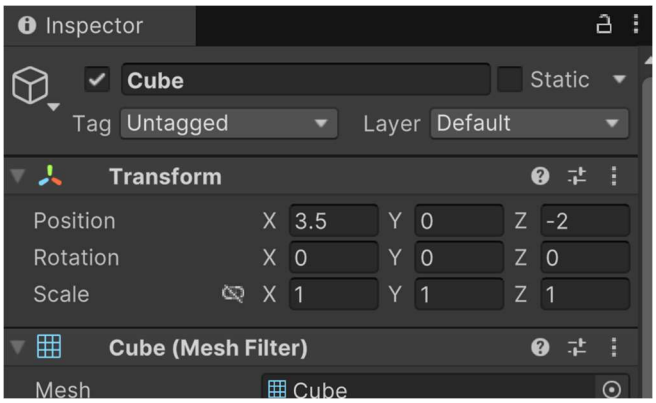
- A **Plane** was added to the Unity scene to serve as the base.
- The **Directional Light** was adjusted to provide proper illumination, ensuring a well-lit environment.
- The **Main Camera** was repositioned and rotated to provide a clear view of the entire scene. The view was adjusted to ensure all objects were visible either at the center or off to the side.

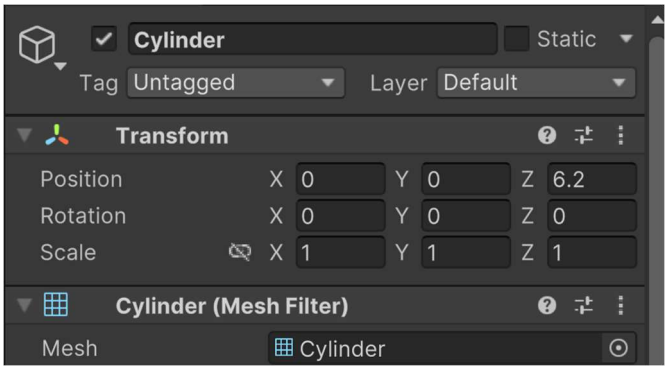
Step II: Adding and Positioning Primitive Objects

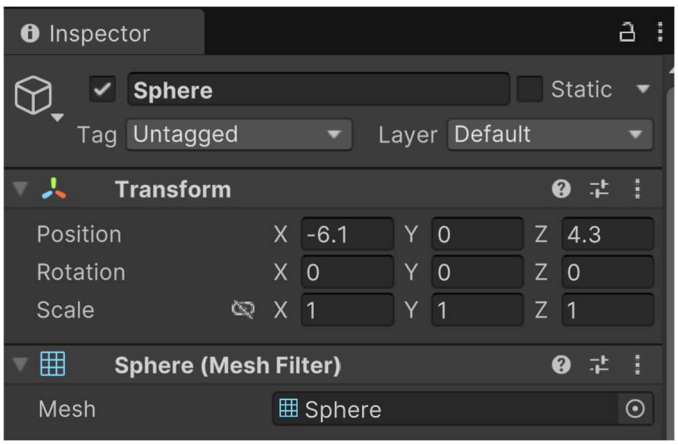
- **Primitive shapes** (Cube, Sphere, and Cylinder) were added to the scene.
- The objects were positioned at the specified coordinates:
  - **Object 1 (Cube):** (X: 3.5, Y: 0, Z: -2.0)
  - **Object 2 (Sphere):** (X: -6.1, Y: 0, Z: 4.3)
  - **Object 3 (Cylinder):** (X: 0.0, Y: 0, Z: 6.2)



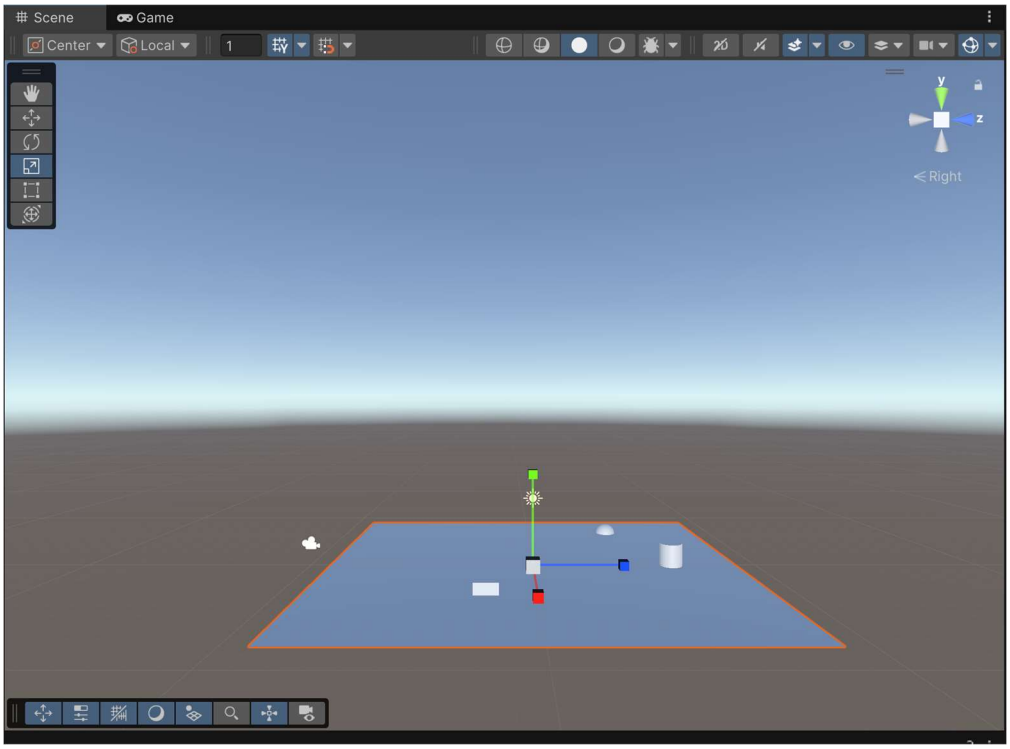

y
x
z
Persp


Inspector
Cube
Static
Tag
Untagged
Layer
Default
Transform
Position
X 3.5
Y 0
Z -2
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Cube (Mesh Filter)
Mesh
Cube

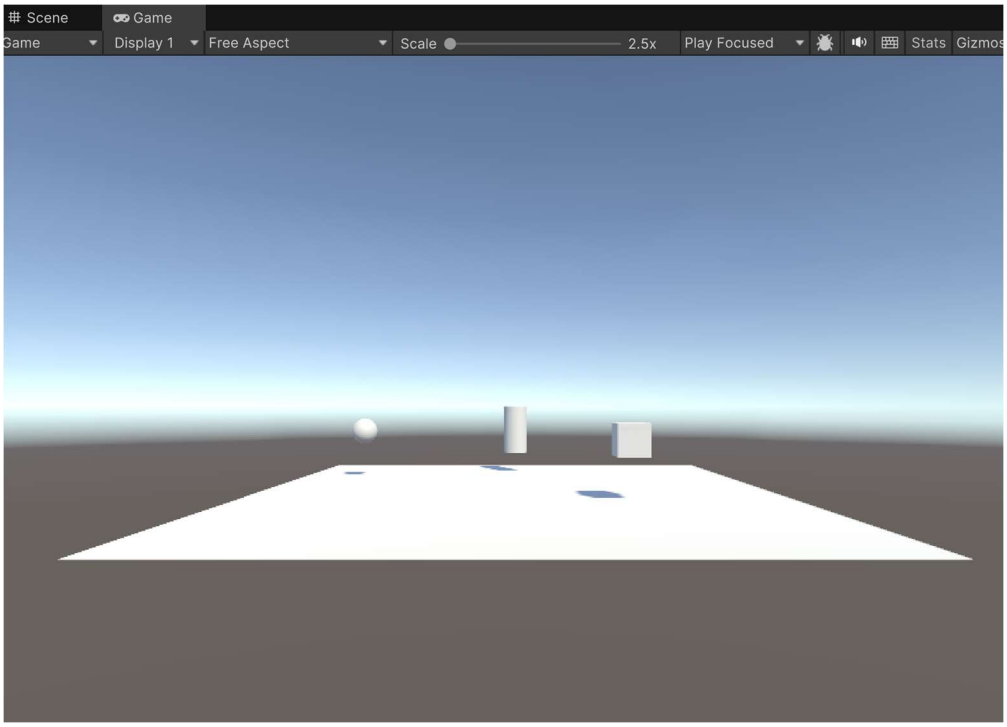

Cylinder
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 0
Z 6.2
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Cylinder (Mesh Filter)
Mesh
Cylinder


Inspector
Sphere
Static
Tag
Untagged
Layer
Default
Transform
Position
X -6.1
Y 0
Z 4.3
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Sphere (Mesh Filter)
Mesh
Sphere

- After that scaled the plane so that all the primitives are on the plane

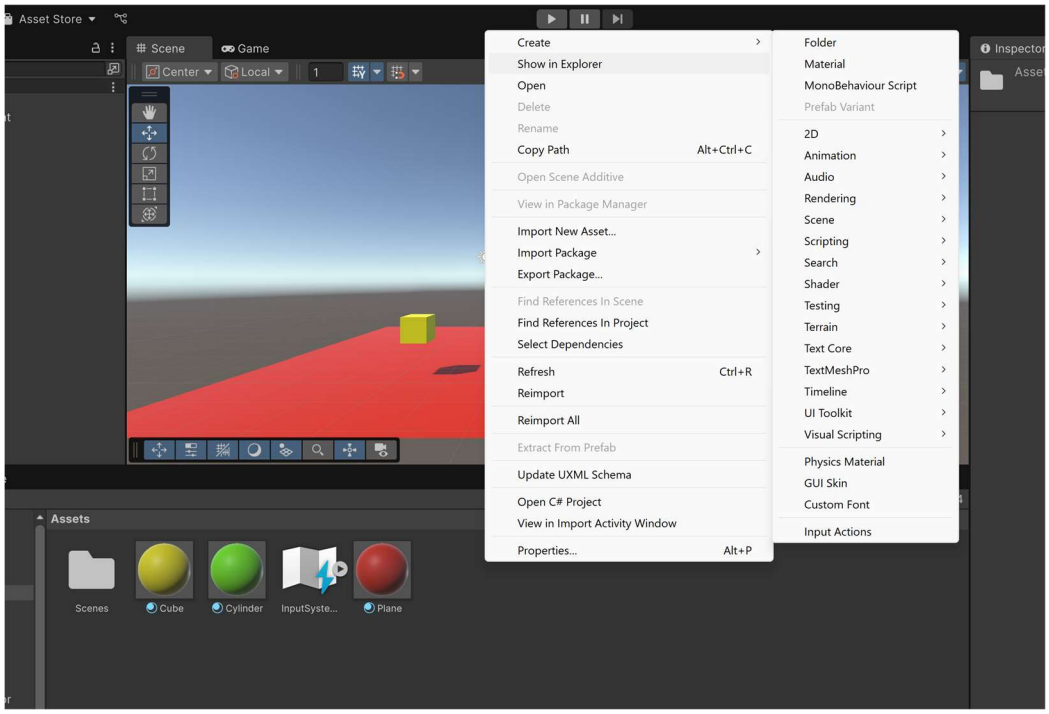


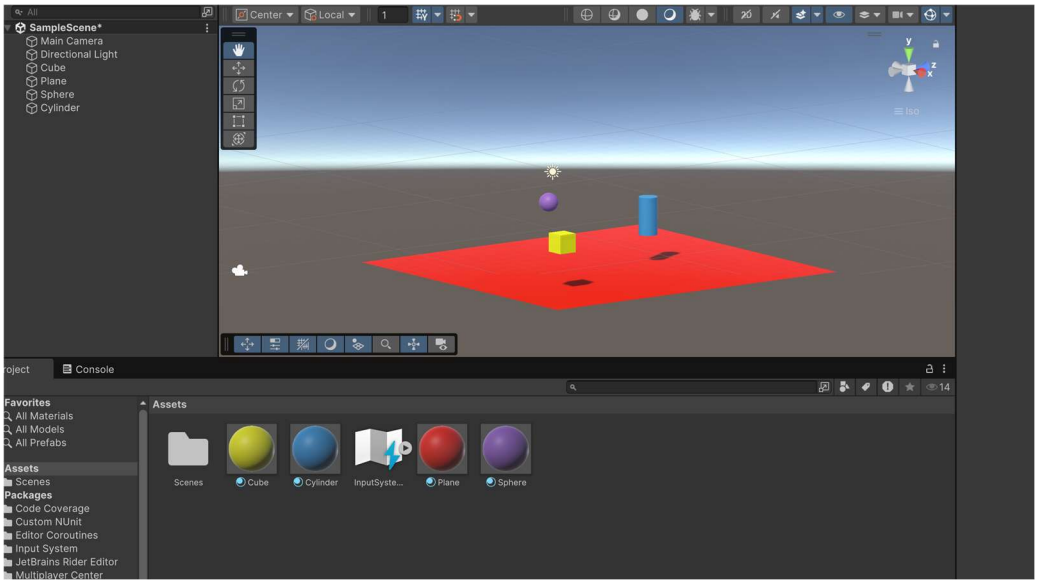
- Game view looks like below



Step III: Creating materials to modify the primitives

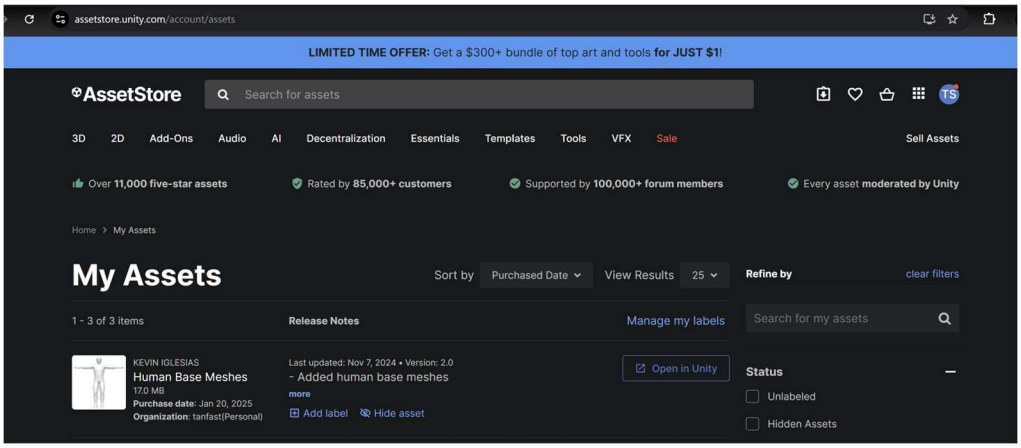
- Created material to add colors to the objects. Yellow color was assigned to cube, purple to sphere and blue to cylinder

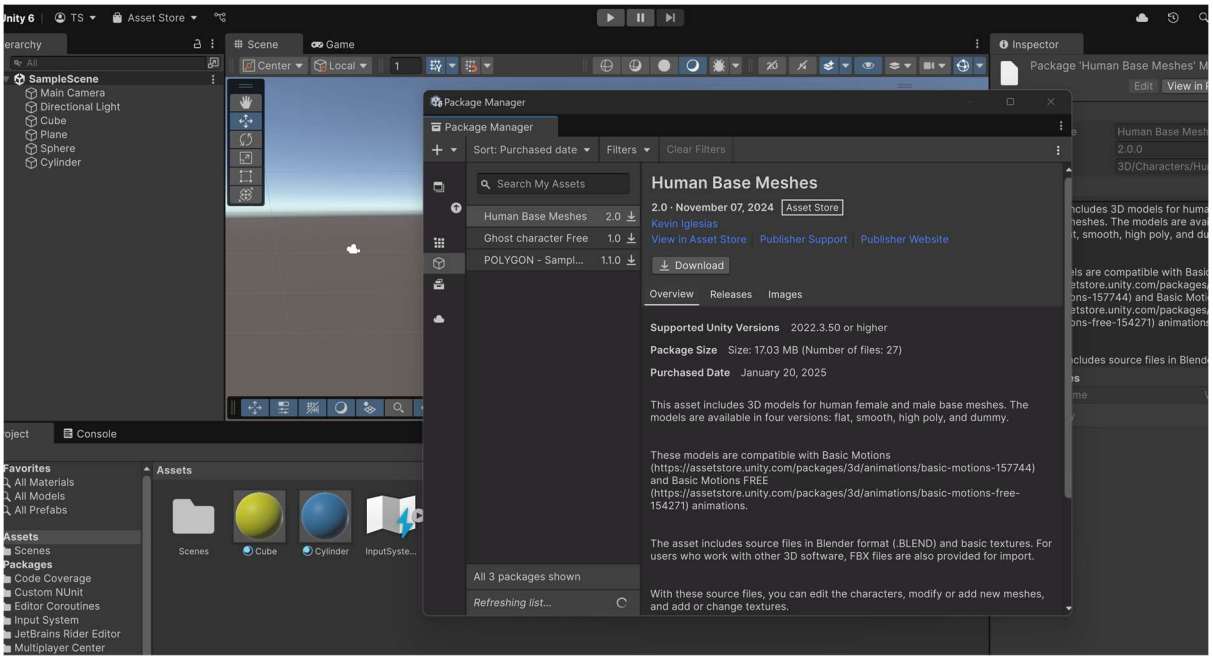


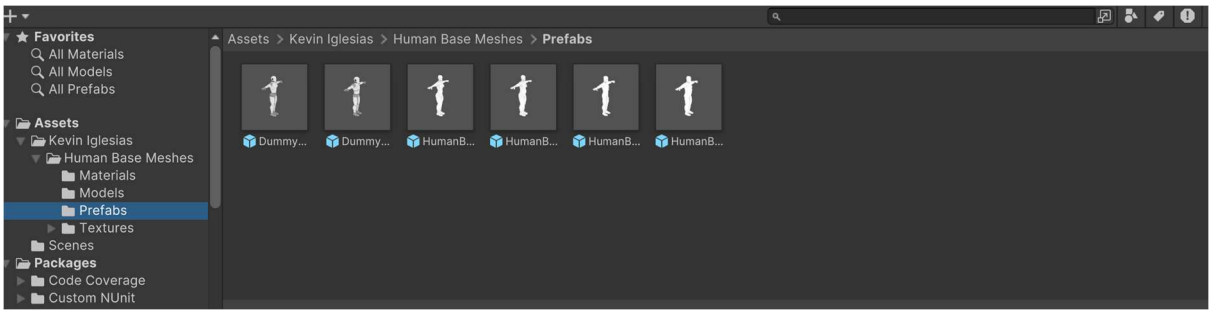


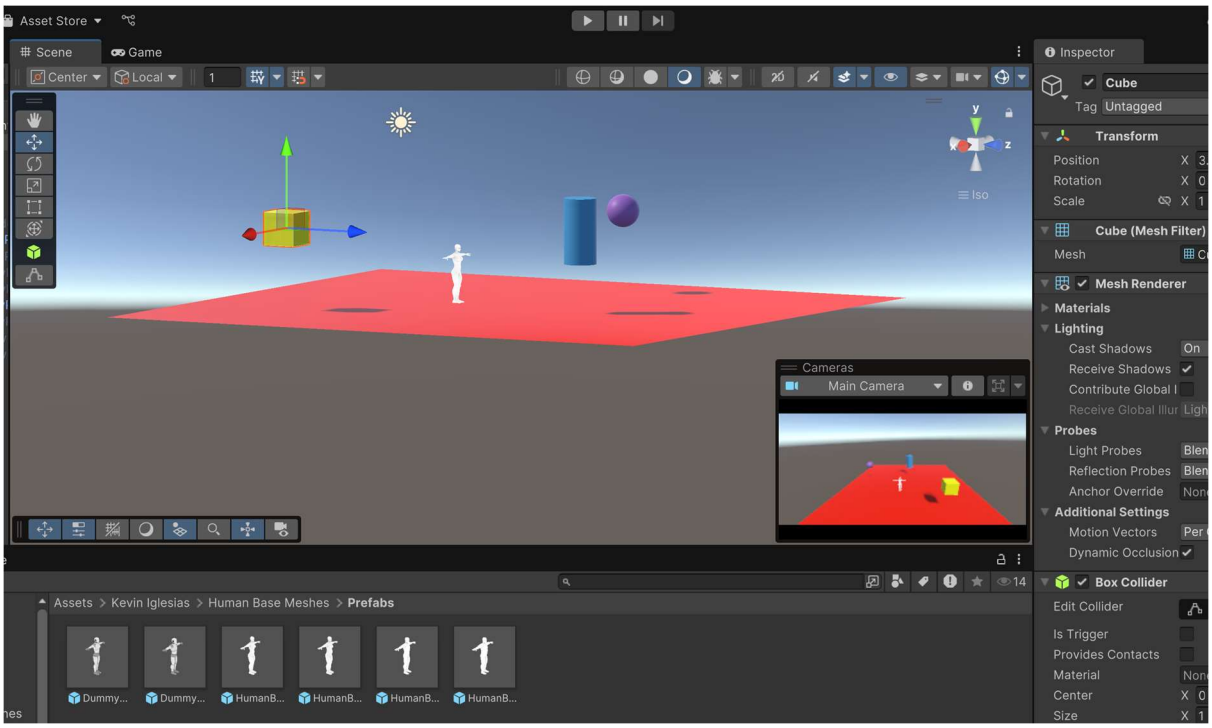
Step IV: Creating Prefabs

- **Prefabs** were created for by downloading human mesh from the Unity Asset Store and added to the scene:
    - Dragged the configured objects (human mesh) into the **Assets folder**.
    - Saved these prefabs for future use.



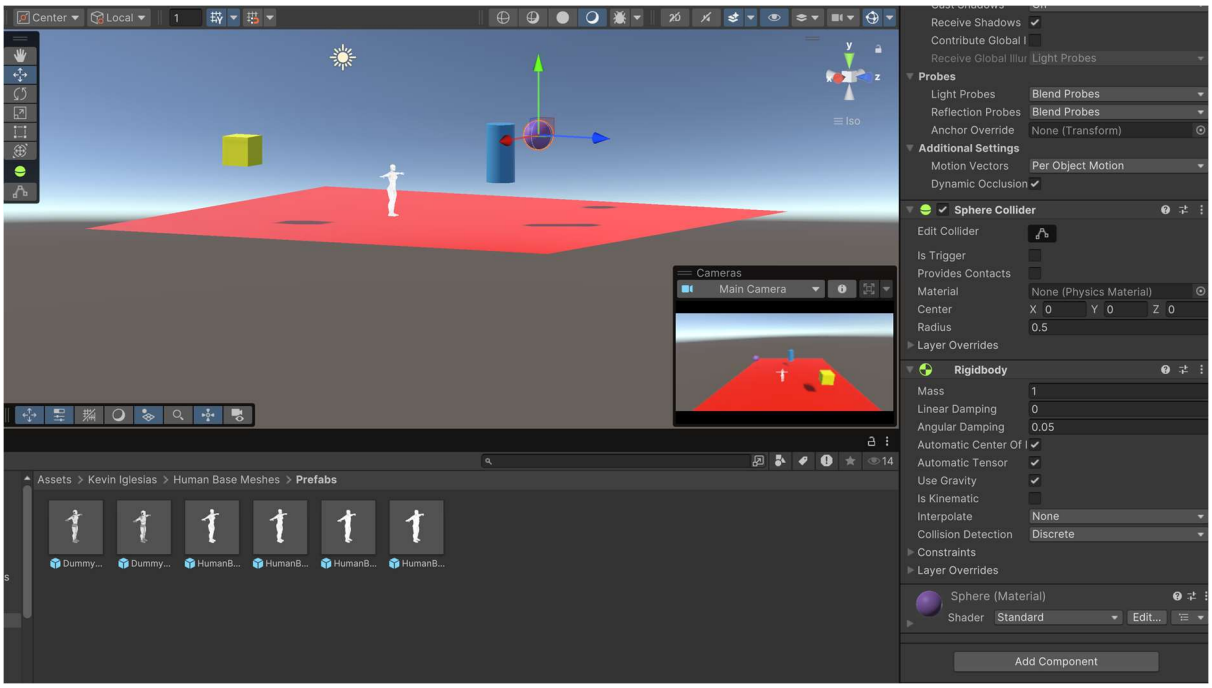




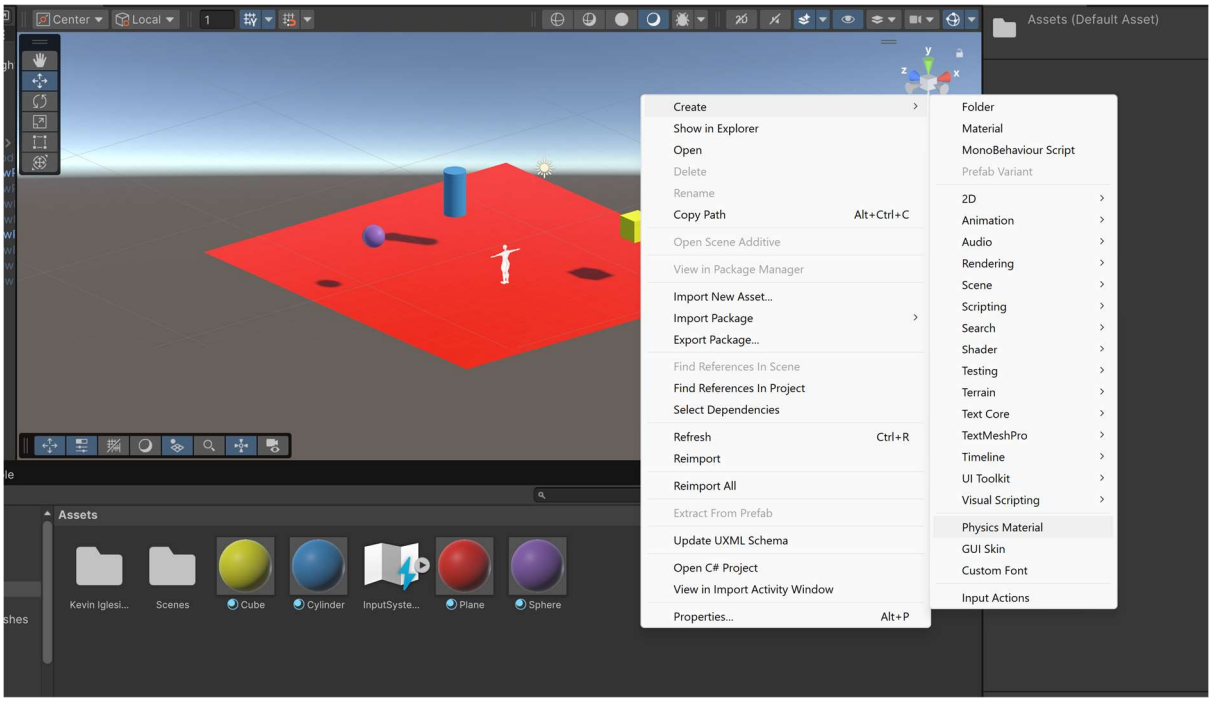


Step V: Configuring Bounce Behaviors

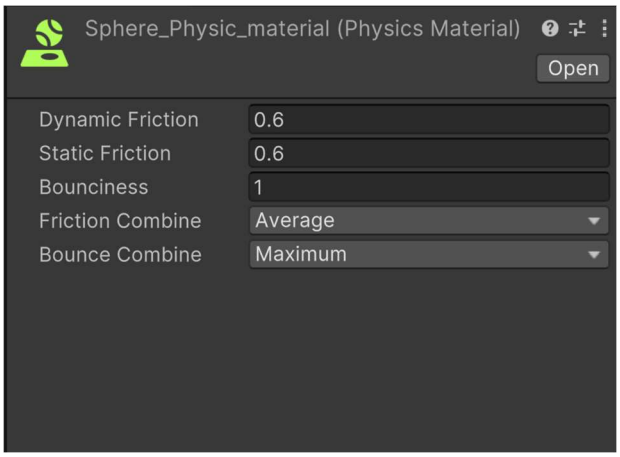
- Added components : collider and rigid body for all the 3 primtives



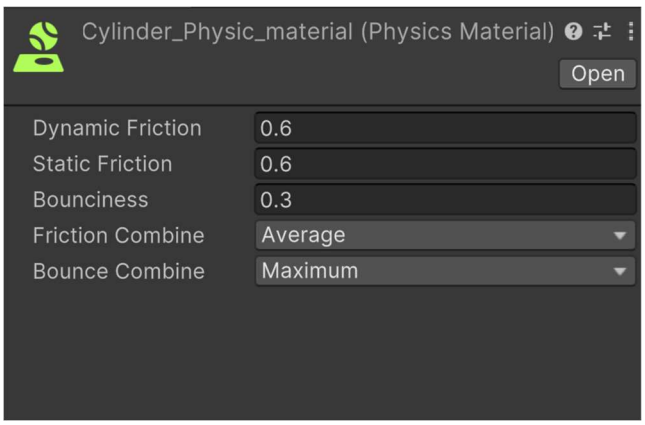
- Creating Physics material



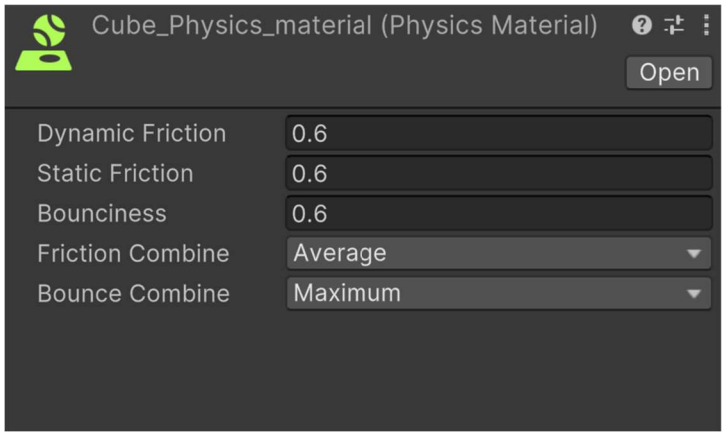
- For sphere, the physic material Bounciness was set to 1 for maximum bounciness



- For Cylinder, the Bounciness was set to 0.3 for least bounciness.



- For Cube, the Bounciness was set to 0.6



Summary of Bounce Behaviors

- **Cube:** Medium bounce behavior with a Bounciness of 0.6. The object bounces moderately when it hits the plane.
- **Sphere:** Maximum bounce behavior with a Bounciness of 1.0. This object exhibits the highest bounce and stays airborne the longest.
- **Cylinder:** Least bounce behavior with a Bounciness of 0.3. The object barely bounces upon hitting the plane.

Recorded Video

- A video of the scene demonstrating the bounce behaviors for each object has been recorded.
- The video highlights:
    - The bounce behavior of each object when dropped onto the plane.
    - The scene setup, including the additional prefab and camera view.

Conclusion

The Unity scene was successfully configured with primitive objects exhibiting distinct bounce behaviors. The inclusion of prefabs and additional scene adjustments enhanced the overall presentation. This lab demonstrated the application of Unity's physics engine and prefab system effectively.