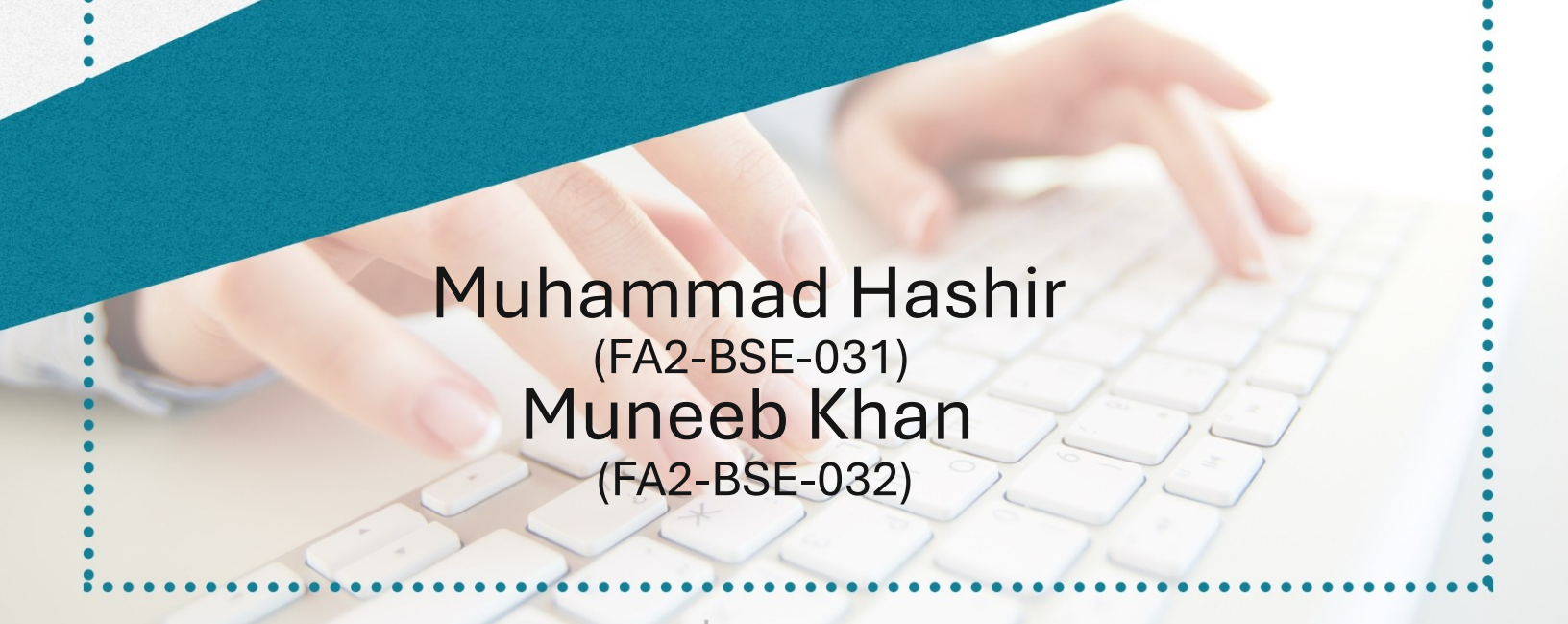
GAME DEVELOPMENT

Assignment – 01

SUBMITTED TO: Sir Ahsan Khan

2025



Muhammad Hashir
(FA2-BSE-031)
Muneeb Khan
(FA2-BSE-032)

Part 1: Project Overview

1. What is the name of your game and what is the main goal for the player?

Name:

CRICKET RIVALRY

Main Goal:

The player selects either **Pakistan** or **India** to play a short-format match (2–4 overs). The goal is to win the iconic rivalry match by scoring the most runs or taking all the opponent's wickets. The gameplay includes batting, bowling, and real-time AI-controlled fielding..

2. What type of game are you creating (e.g., platformer, maze runner, adventure)?

Answer: It's a **short-format, rivalry-based cricket simulation game** focused on delivering intense cricket action between Pakistan and India, using simple controls and AI-driven fielding mechanics.

3. What core Unity components are you planning to use in this assignment?
Answer:

- **Rigidbody & Colliders** (for ball physics)
- **Animator Controller** (for player movements: batting, bowling, catching)
- **NavMeshAgent** (to control AI fielders' chasing behavior)
- **Plane or Terrain** (for cricket ground design)
- **Cinemachine Camera** (for smooth dynamic match view)
- **UI Canvas** (for team selection, scoreboard, and match info)
- **Scripts & Input System** (for player controls and game logic)
- **Prefabs** (players, bat, ball, stumps, stadium elements)

4. How does the player interact with the environment in your game?

Answer:

- The player chooses **Pakistan** or **India** and controls batting and bowling using keyboard input.
- **Batting:** Timed keypress to hit the ball in different directions.
- **Bowling:** Player selects speed and direction with keys.
- **AI-controlled fielders** use NavMesh to chase the ball and attempt catches/runouts.

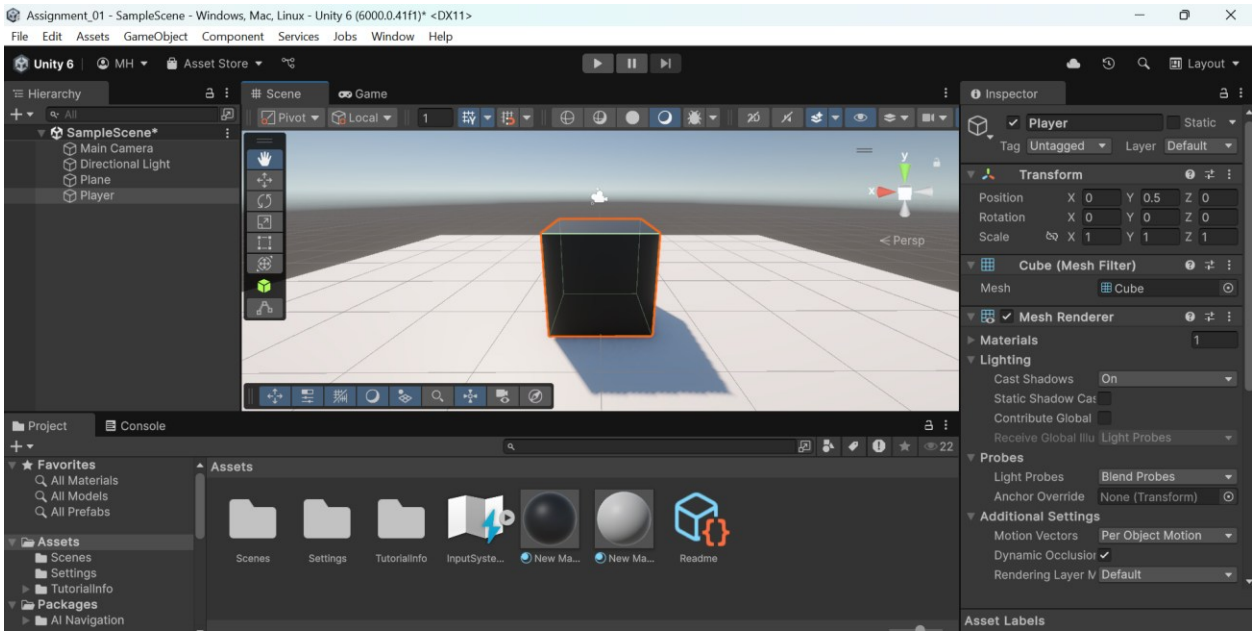
- Overs, wickets, and runs are tracked in real-time. The match ends when all overs are completed or the batting team is out. The player's goal is to beat the rival team and win the match.

Muhammad Hashir's work:

Part 2: Player Movement with Rigidbody

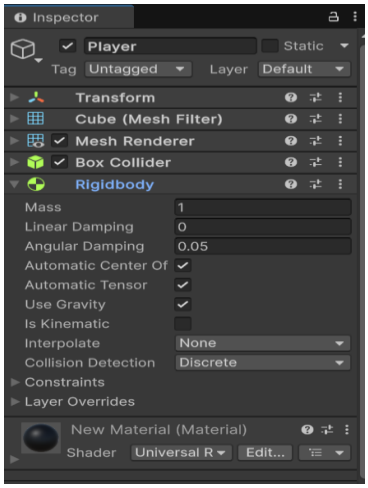
5. Have you created a 3D player GameObject (such as a Capsule or Cube)?

Answer: Yes, I created a 3D player GameObject using a **Cube** to represent a cricket player (batsman).



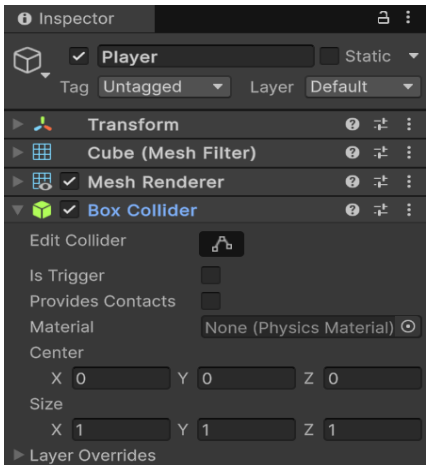
6. Did you attach a Rigidbody component to the player to enable physics?

Answer: Yes, I added a **Rigidbody** component to enable gravity and physics-based movement for the player.



7. Have you added a Collider (e.g., CapsuleCollider or BoxCollider) to the player?

Answer: Yes, I used the **CapsuleCollider**, which is ideal for a standing player-like shape and helps detect collisions with the ball or ground.

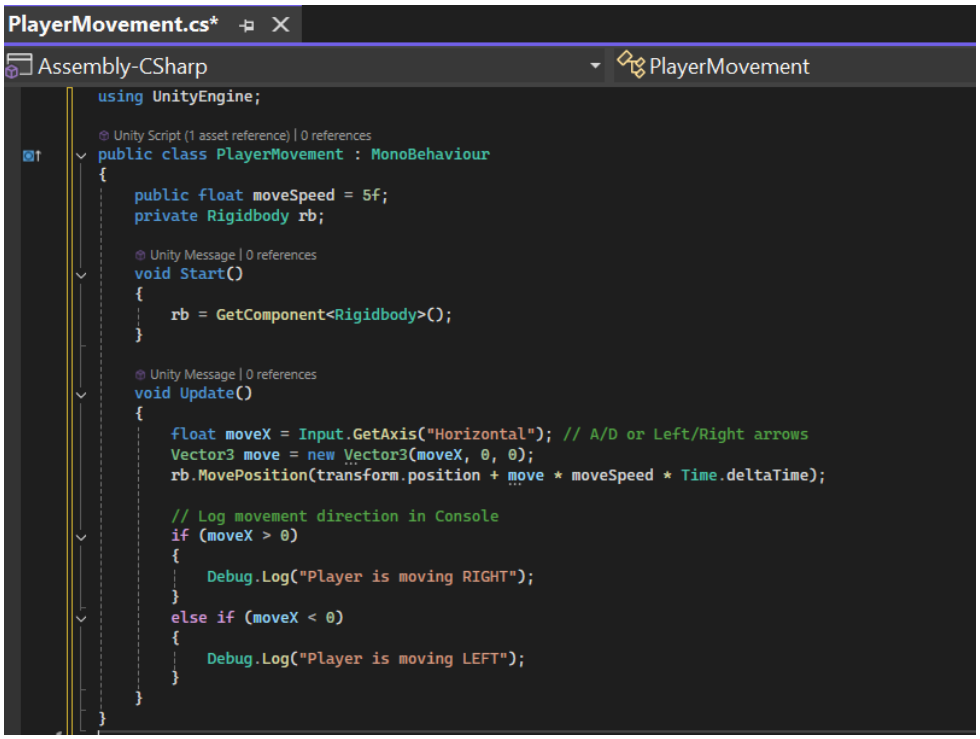


8. Did you write a script using the old Input system (Input.GetAxis) to control the player with WASD or Arrow keys?

o Show the code snippet you used for movement.s

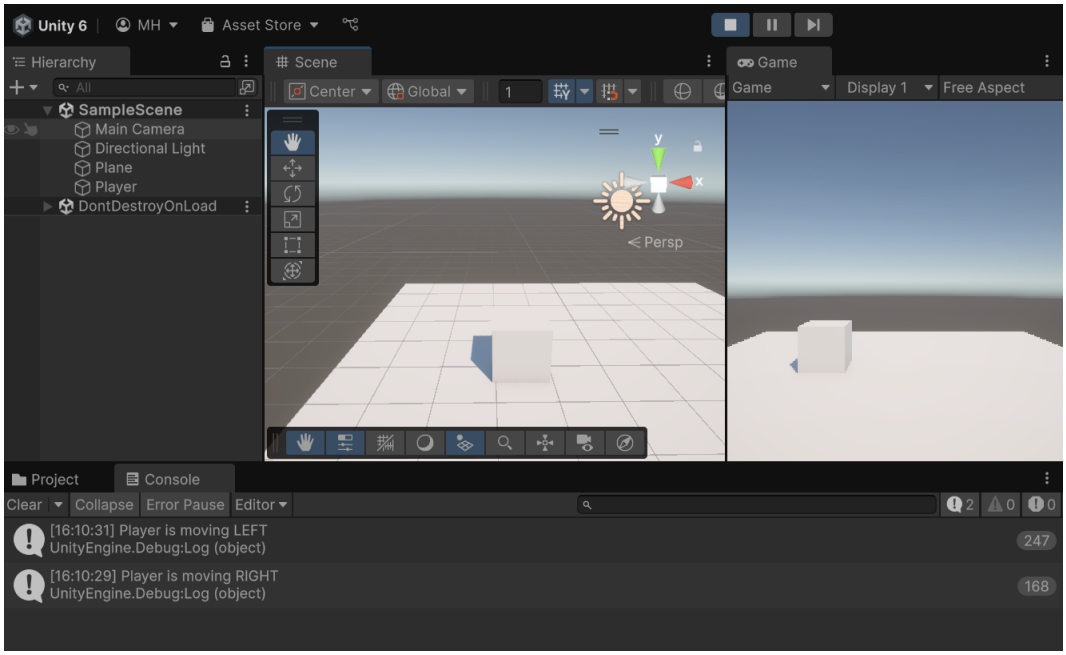
Answer: Yes, I wrote a custom movement script using the **old Input system (Input.GetAxis)** to move the player left or right — useful for small batsman movement or positioning.

SCRIPT:



9. Is your player movement smooth and responding to user input correctly?

Answer: Yes, the movement is smooth and responsive. The player moves left and right across the pitch using **A/D or Left/Right arrow keys**, and the Rigidbody makes the movement feel realistic.

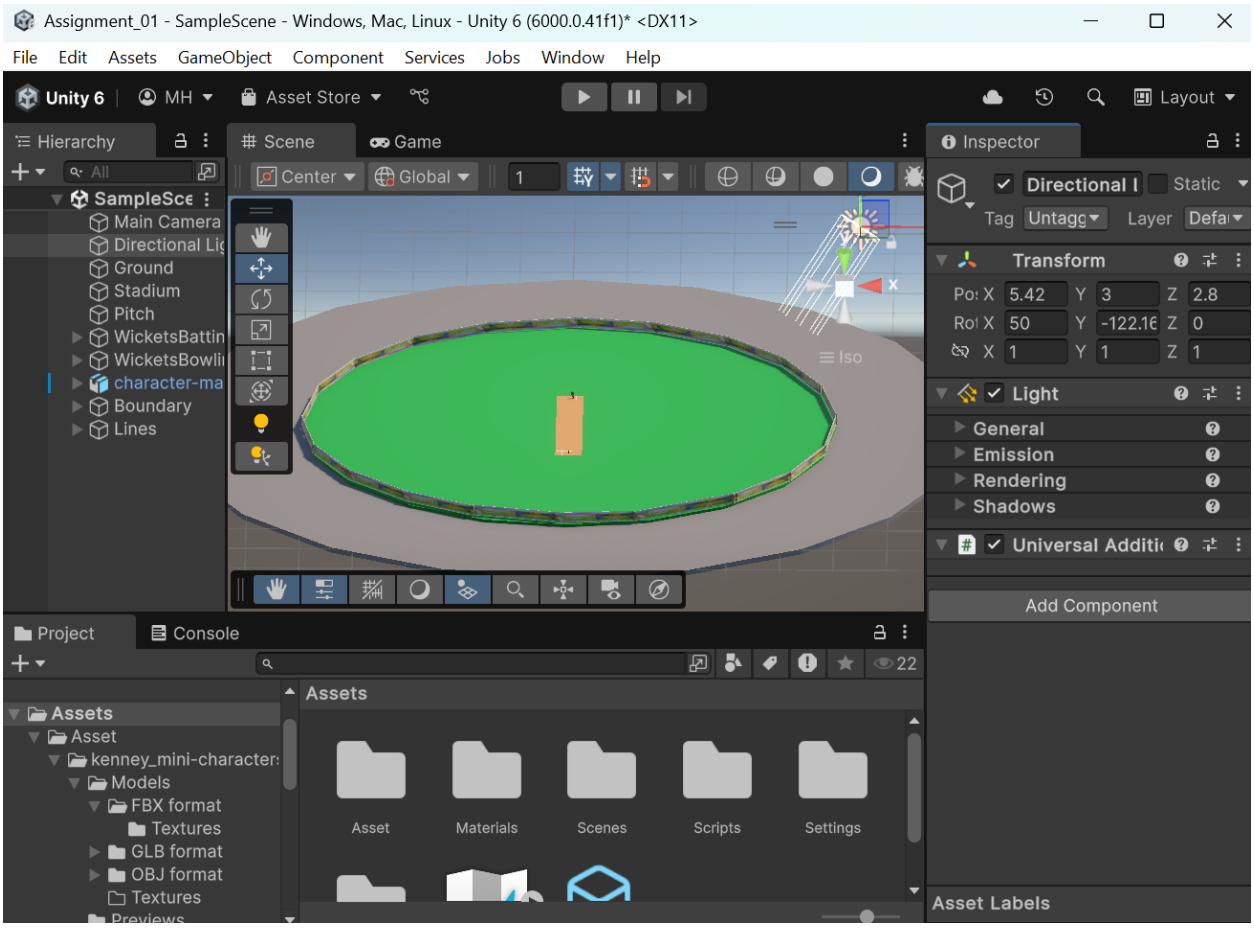


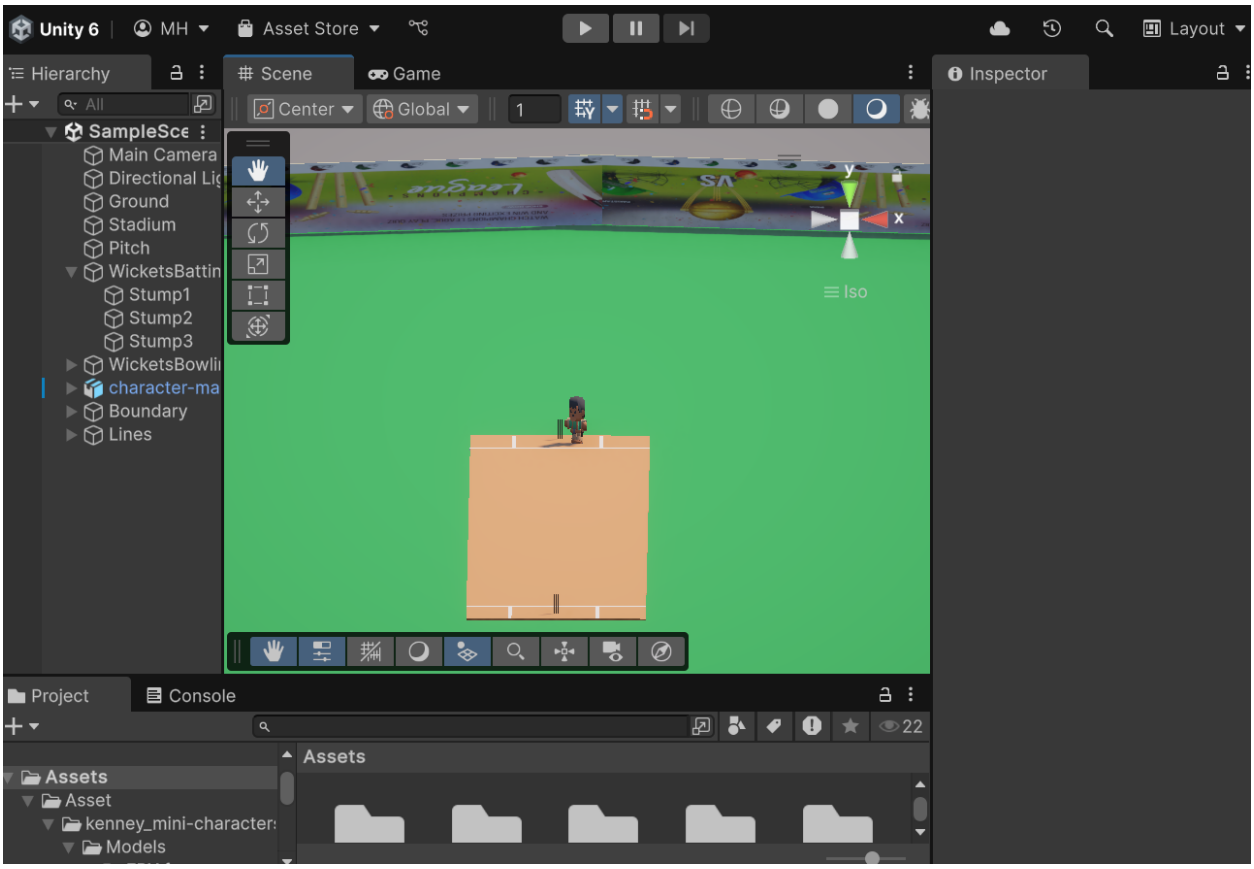
Part 3: Basic Level Design

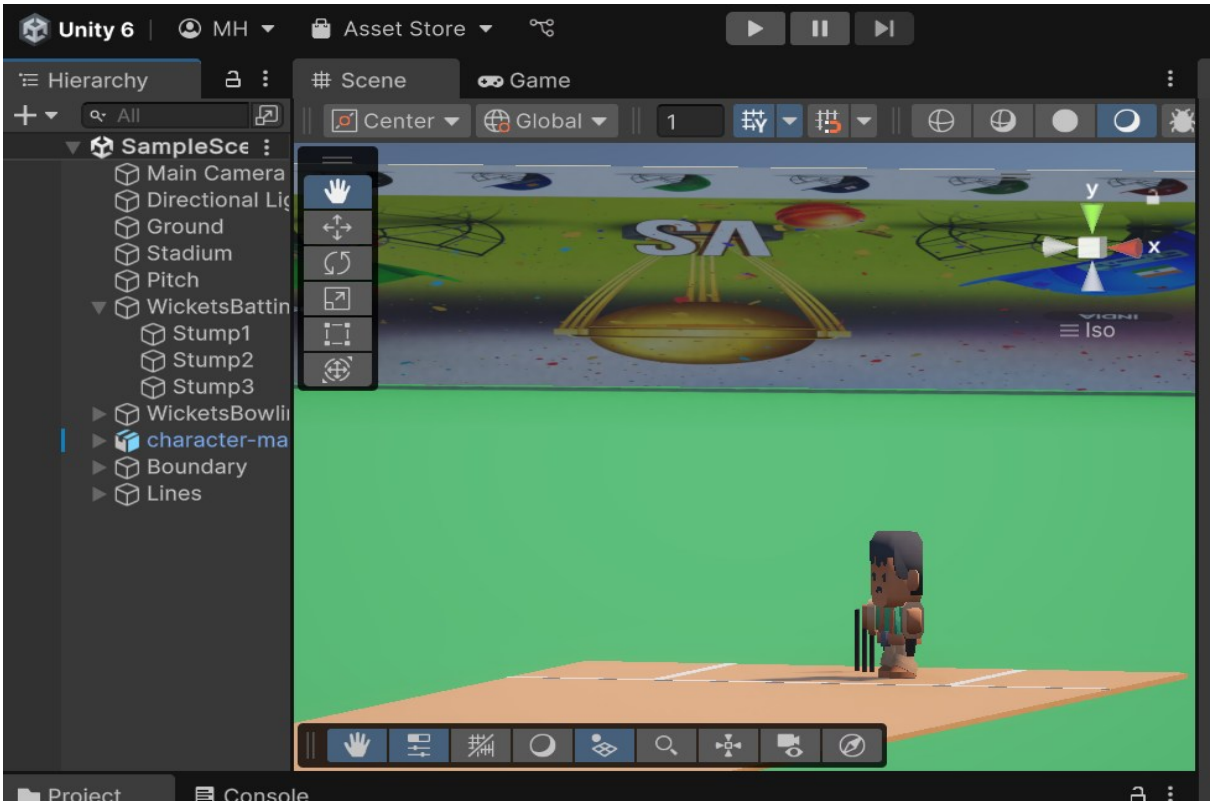
10. Have you designed a simple level layout using 3D objects (cubes, planes, etc.)?

Answer: Yes, I designed a basic cricket ground using 3D objects:

- A **Cylinder** was used as the cricket pitch/ground.
- **Cubes** were used for wickets and boundaries.
- This simple layout helped simulate the environment for player testing.


Assignment_01 - SampleScene - Windows, Mac, Linux - Unity 6 (6000.0.41f1)* <DX11>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
MH
Asset Store
Layout
Hierarchy
Scene
Game
Inspector
SampleSce
Main Camera
Directional Lig
Ground
Stadium
Pitch
WicketsBattin
WicketsBowli
character-ma
Boundary
Lines
Center
Global
Iso
Directional l
Static
Tag
Layer
Transform
Light
General
Emission
Rendering
Shadows
Universal Additi
Add Component
Project
Console
Assets
Asset
kenney_mini-character
Models
FBX format
Textures
GLB format
OBJ format
Textures
Materials
Scenes
Scripts
Settings
Asset Labels

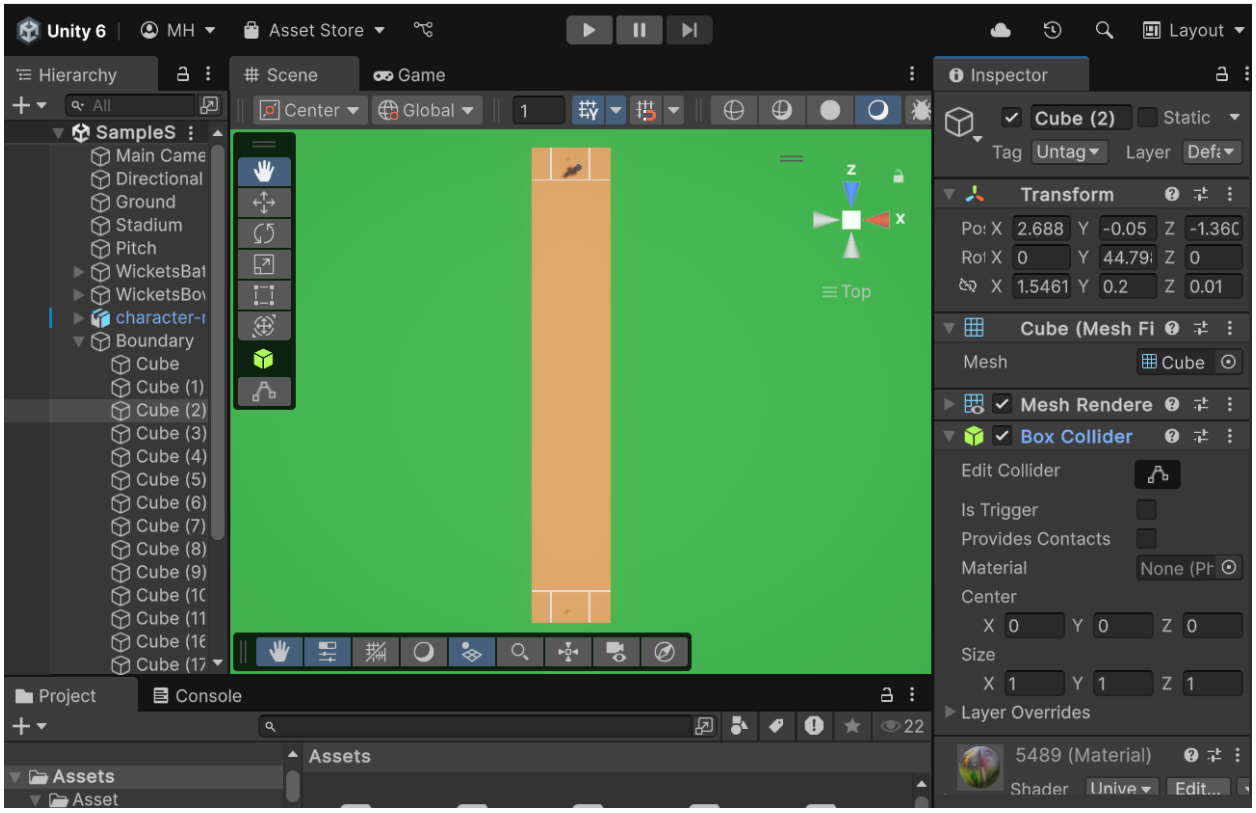

Unity 6
MH
Asset Store
Layout
Hierarchy
Scene
Game
Inspector
Center
Global
SampleSce
Main Camera
Directional Lig
Ground
Stadium
Pitch
WicketsBattin
Stump1
Stump2
Stump3
WicketsBowli
character-ma
Boundary
Lines
Iso
Project
Console
Assets
Asset
kenney_mini-character
Models


Unity 6
MH
Asset Store
Hierarchy
Scene
Game
Center
Global
SampleSce
Main Camera
Directional Lig
Ground
Stadium
Pitch
WicketsBattin
Stump1
Stump2
Stump3
WicketsBowli
character-ma
Boundary
Lines
Iso
Project
Console

11. Did you apply Colliders to all platforms and obstacles so that physics interactions work correctly?

Answer: Yes, **BoxColliders** were added to:

- The ground plane (to prevent falling)
- Wickets and boundary objects
- Other elements like the ball and bat.

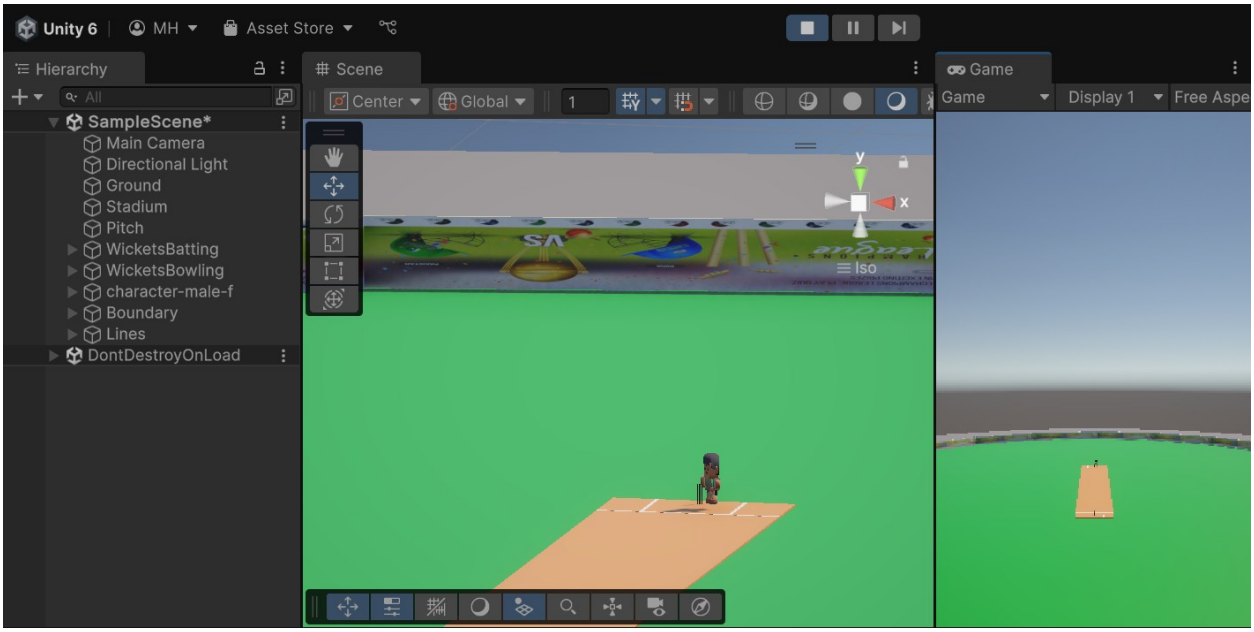


12. Have you tested that your player doesn't fall through the floor and collides properly with level

objects?

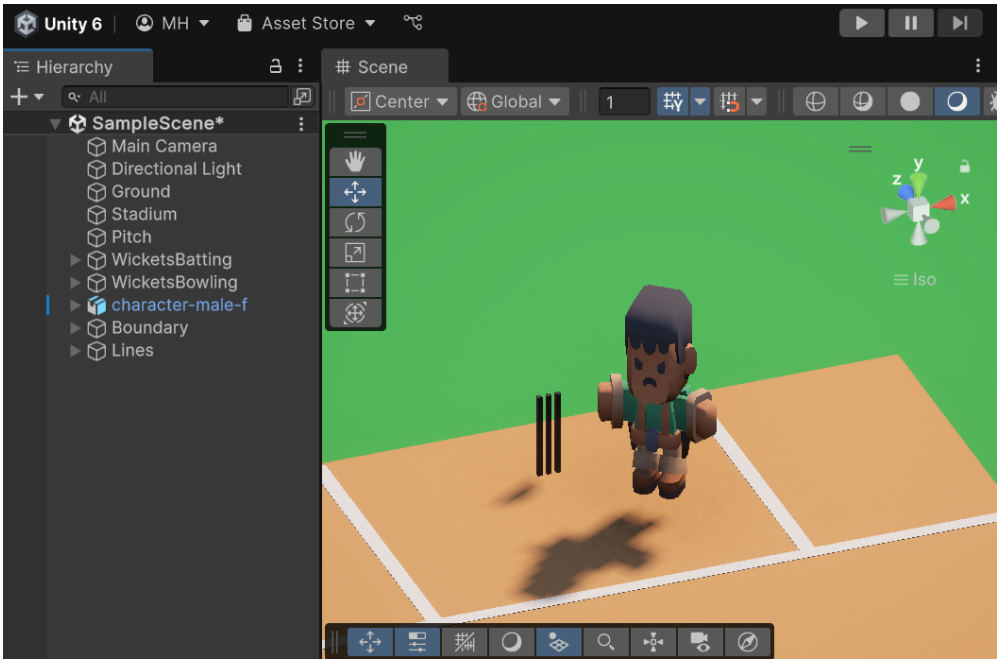
Answer: Yes, I tested the game and confirmed:

- The player stays on the pitch (doesn't fall through)
- Collisions with obstacles like stumps and boundaries are working correctly
- Rigidbody and Collider settings are properly configured



13. Have you added at least one obstacle or challenge element to the level?

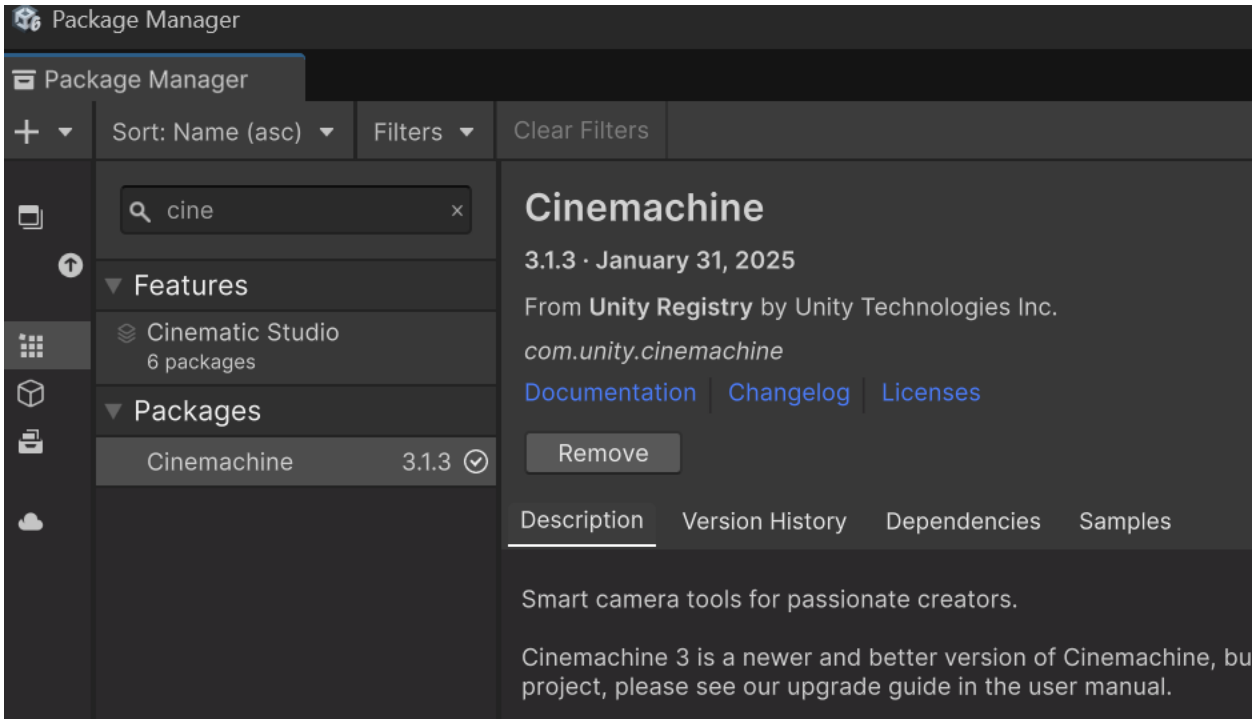
Answer: Yes, I added wickets as **obstacles** in front of the batsman and **fielders** as challenge elements.
If the ball hits the stumps or is caught by a fielder (controlled by AI), the **level restarts or an event triggers**, adding challenge to the gameplay.



Part 4: Cinemachine Camera Setup

14. Have you imported the Cinemachine package from the Unity Package Manager?

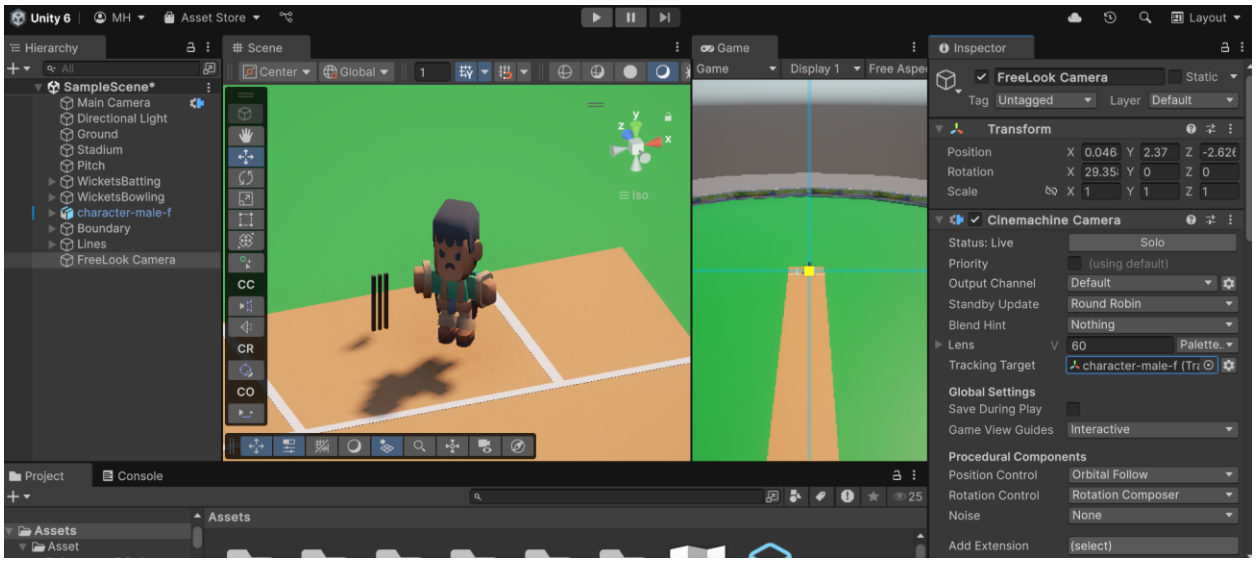
Answer: Yes, the Cinemachine package was successfully imported using the Unity Package Manager. It adds powerful camera control tools ideal for tracking players or action dynamically.



15. Did you add a Cinemachine Virtual Camera to your scene?

Answer:

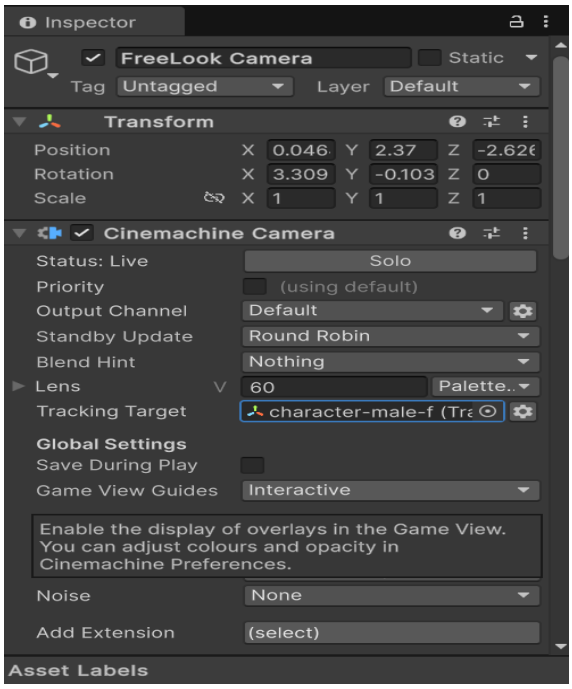
- Yes, I added a **Cinemachine Virtual Camera** to the scene.
- The camera **follows the player (batsman)** during normal gameplay.
- When the player hits a shot, the camera **switches to follow the ball** to show its trajectory.
- This adds a **dynamic and realistic feel** to the gameplay.



16. Have you assigned your player GameObject as the Follow and Look At targets in the virtual camera?

Answer:

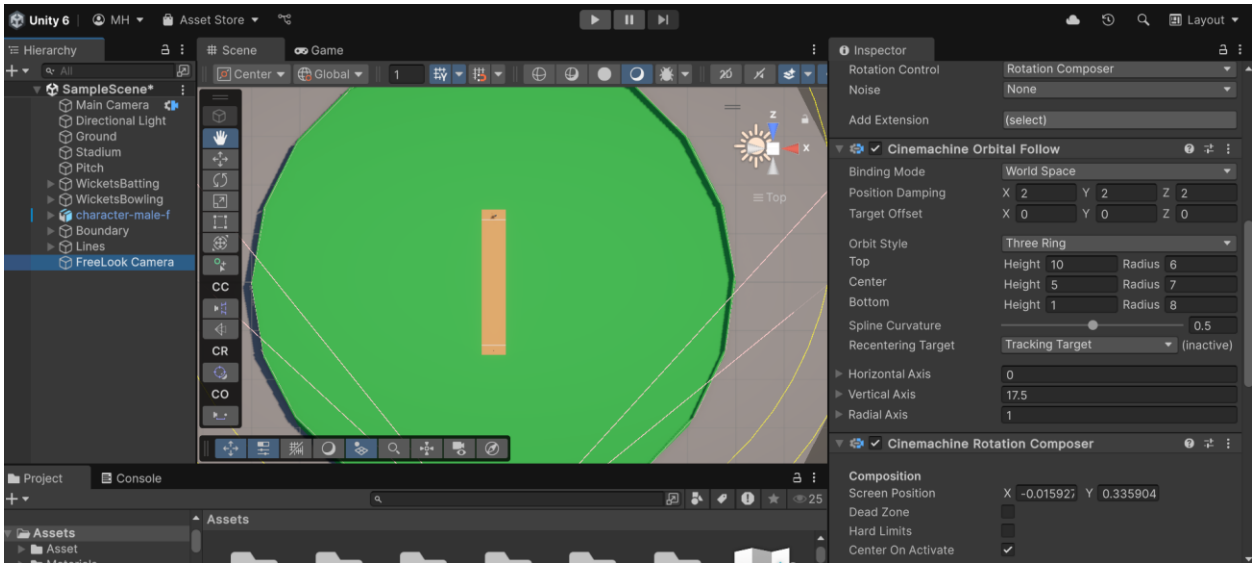
- Yes, I used a **Cinemachine FreeLook Camera** in my scene.
- The **Batsman GameObject** is assigned as both the **Follow** and **Look At** target.
- This allows the camera to smoothly orbit around the player and give a more **interactive and immersive view** during gameplay.



17. Did you adjust camera settings like damping or offset for a smooth follow effect?

Answer: Yes, I adjusted the FreeLook camera settings to achieve a smooth and cinematic follow effect.
I fine-tuned the Orbits (Top, Middle, Bottom) individually:

- Top Rig: Height = 10, Radius = 6
- Middle Rig: Height = 5, Radius = 7
- Bottom Rig: Height = 1, Radius = 8
  Additionally, I set the Damping values to smoothen camera motion, ensuring a natural feel when following the player or the ball during gameplay.



18. Is the camera tracking the player properly as they move through the level?

Answer: Yes, the FreeLook camera is properly tracking the player during gameplay. The camera smoothly rotates and follows the player's movement across the pitch, maintaining a dynamic and cinematic perspective throughout the match.

Github Link:

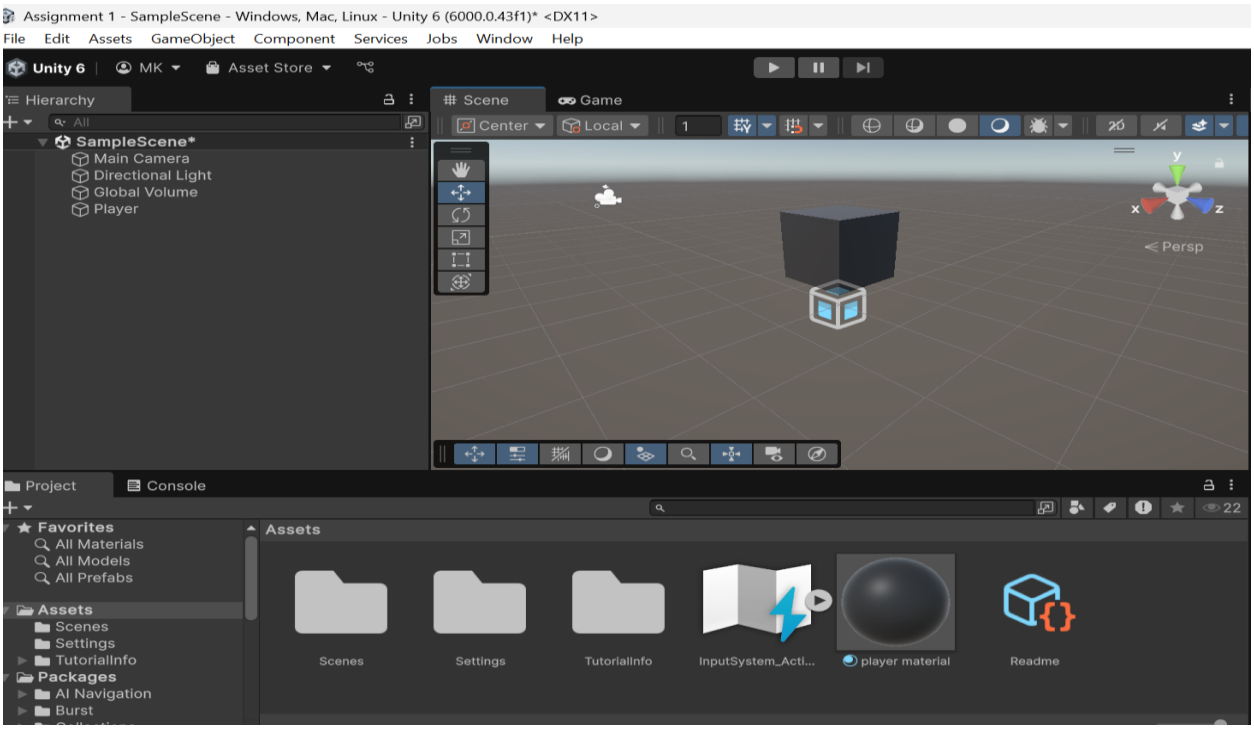
https://github.com/hashir015/Game_Semester_Project

Muneeb Khan's work:

PART 2: Player Movement with Rigidbody

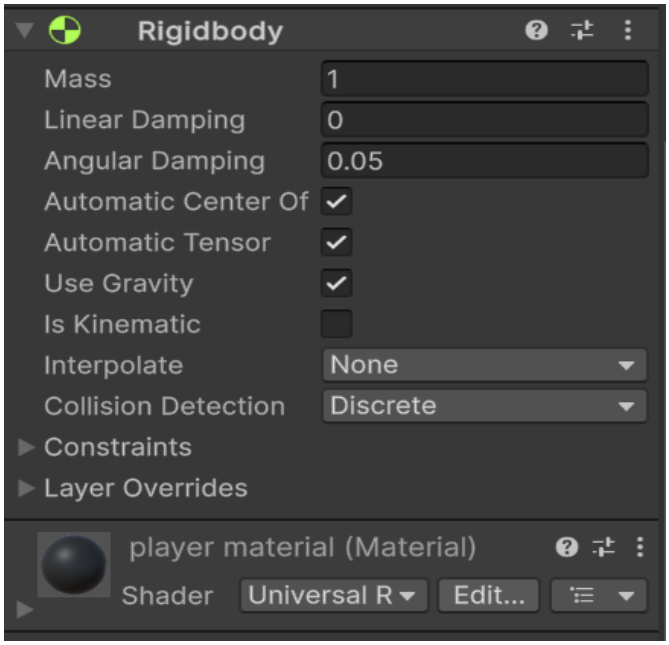
5. Have you created a 3D player GameObject (such as a Capsule or Cube)?

Answer: **Yes**, I created a 3D player GameObject using a Cube to represent a cricket player (batsman).

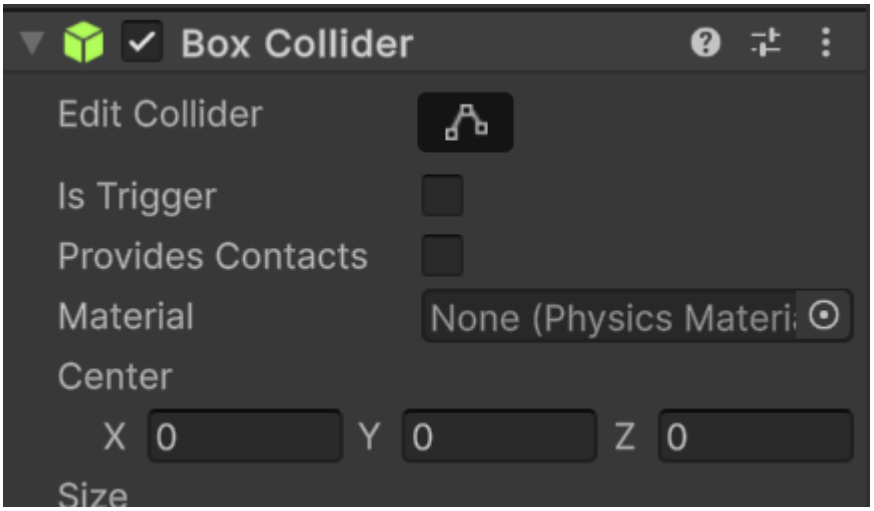


6. Did you attach a Rigidbody component to the player to enable physics?

Answer: Y**es**, I added a Rigid body component to enable gravity and physics-based movement for the player.



7. Have you added a Collider (e.g., CapsuleCollider or BoxCollider) to the player?



8. Did you write a script using the old Input system (Input.GetAxis) to control the player with WASD or Arrow keys?

Show the code snippet you used for movement.s

Answer: Yes, I wrote a custom movement script using the old Input system (Input.GetAxis) to move the player left or right — useful for small batsman movement or positioning.

SCRIPT:

PlayerMovement.cs

Assets > PlayerMovement.cs

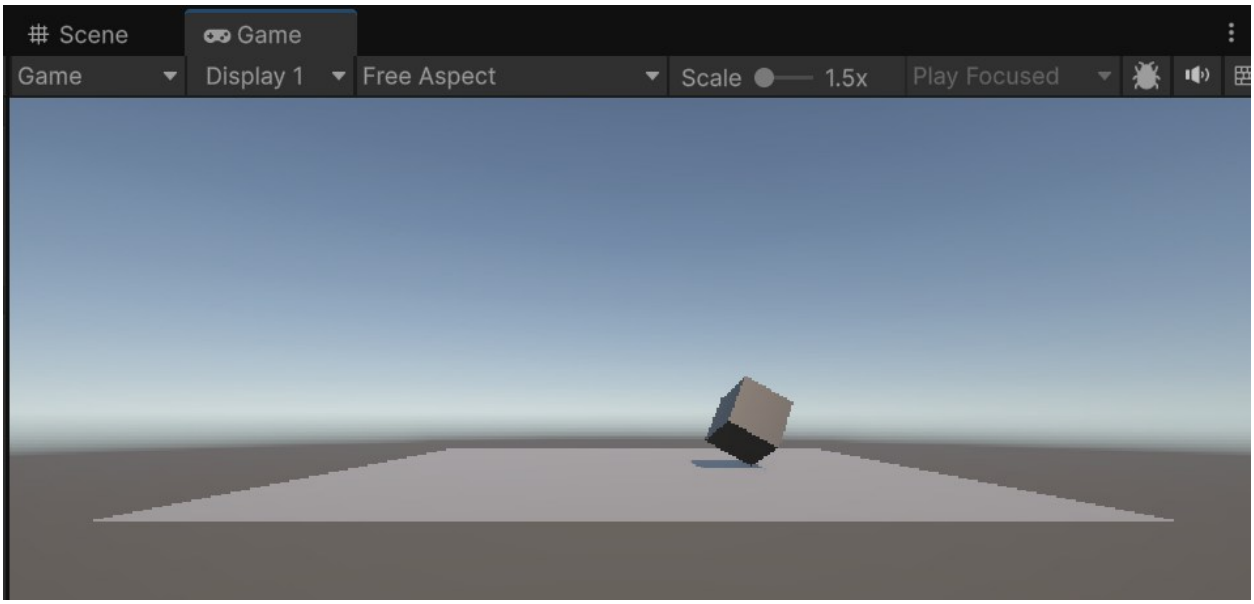
```
1   using UnityEngine;
2
3   public class PlayerMovement : MonoBehaviour
4   {
5       public float moveSpeed = 5f;  // Speed of movement
6       private Rigidbody rb;
7
8       void Start()
9       {
10          rb = GetComponent<Rigidbody>();  // Get the Rigidbody attached to the player
11      }
12
13      void FixedUpdate()
14      {
15          float move = Input.GetAxis("Horizontal");  // A/D or Left/Right
16          rb.velocity = new Vector3(move * moveSpeed, rb.velocity.y, 0);
17      }
18  }
```

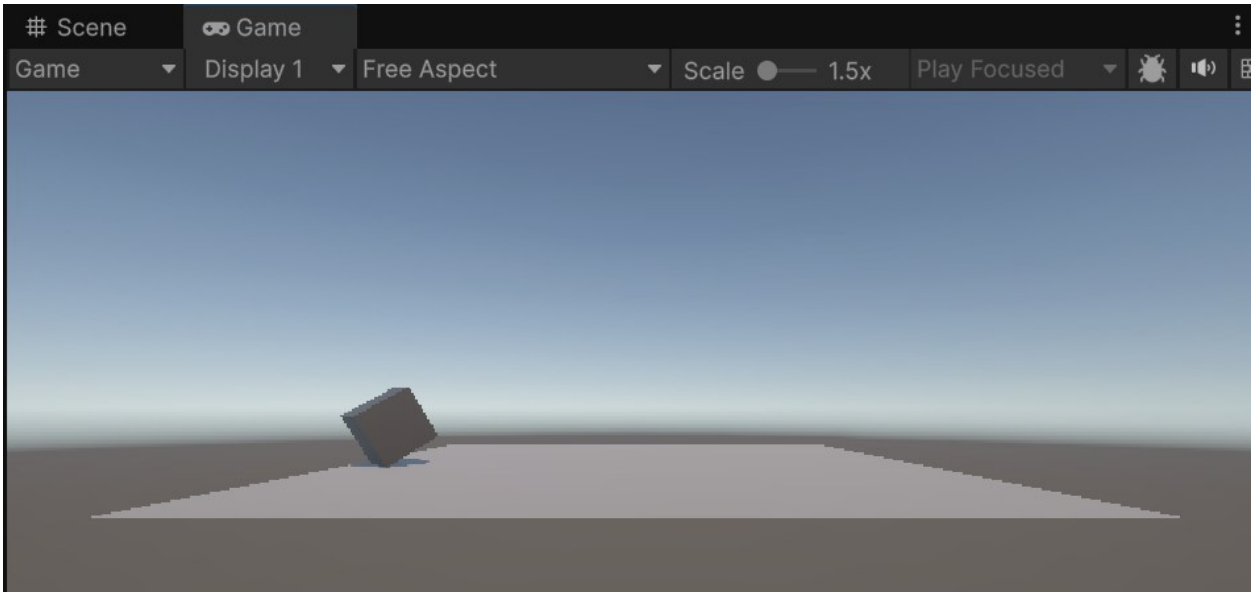
Code Snippet:

```
float move = Input.GetAxis("Horizontal");  // A/D or Left/Right
rb.linearVelocity = new Vector3(move * moveSpeed, rb.linearVelocity.y, 0);
```

9. Question: Is your player movement smooth and responding to user input correctly?

Answer: Yes, the movement is smooth and responsive. The player moves left and right across the pitch using A/D or Left/Right arrow keys, and the Rigidbody makes the movement feel realistic.





PART 3: Basic Level Design

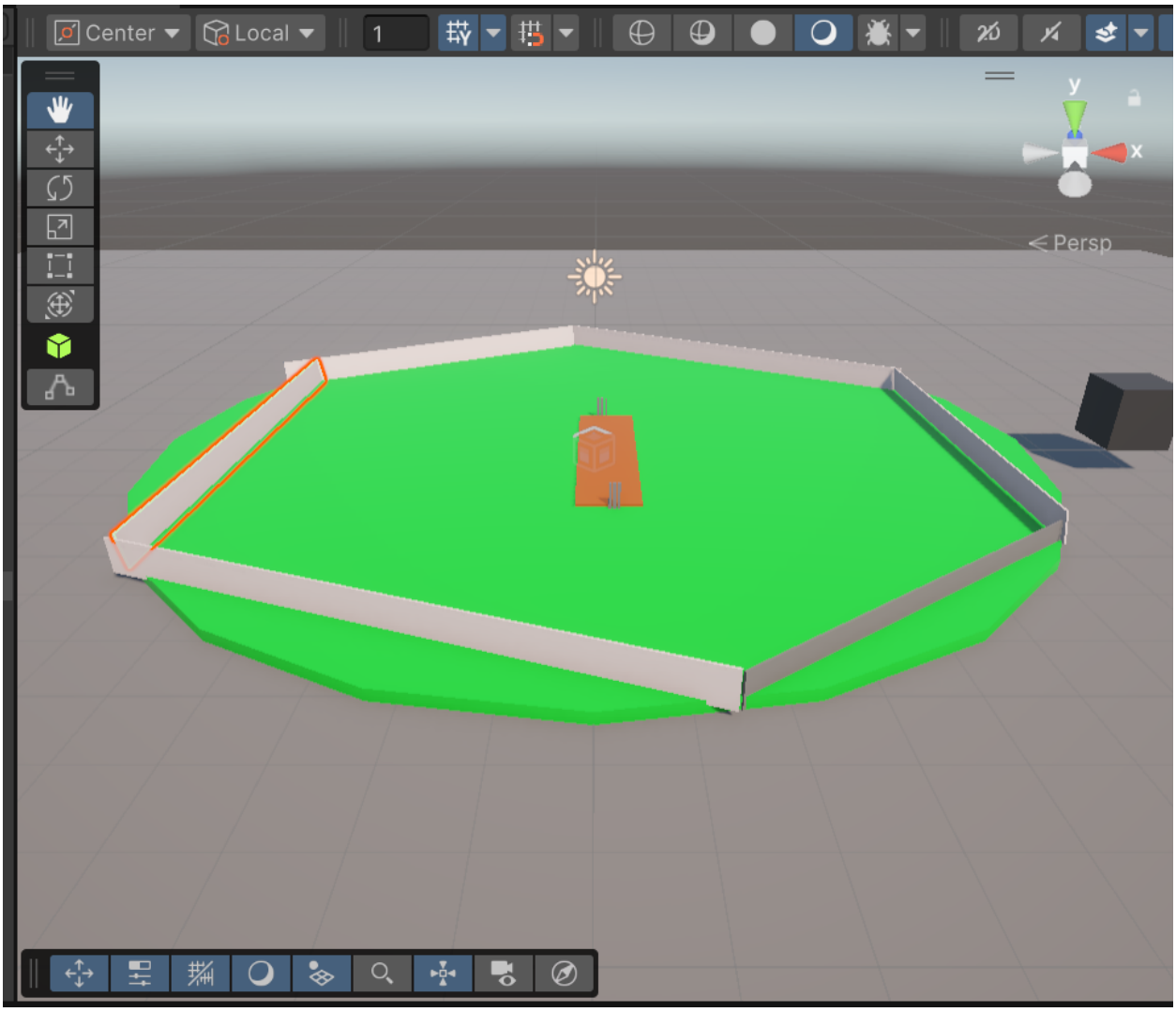
10. Have you designed a simple level layout using 3D objects (cubes, planes, etc.)?

Answer: Yes, I designed a basic cricket ground using 3D objects:

- A Cylinder was used as the pitch to give a circular shape to the play area.
- Cubes were used as wickets placed at both ends of the pitch.

- To match the round pitch, I created a circular boundary by placing multiple cubes (named B1, B2, B3, etc.) in a circular formation around the ground.
- All boundary pieces were grouped under a parent GameObject named "Boundary" for better organization.

This simple layout helped create a playable environment for testing movement and collision.

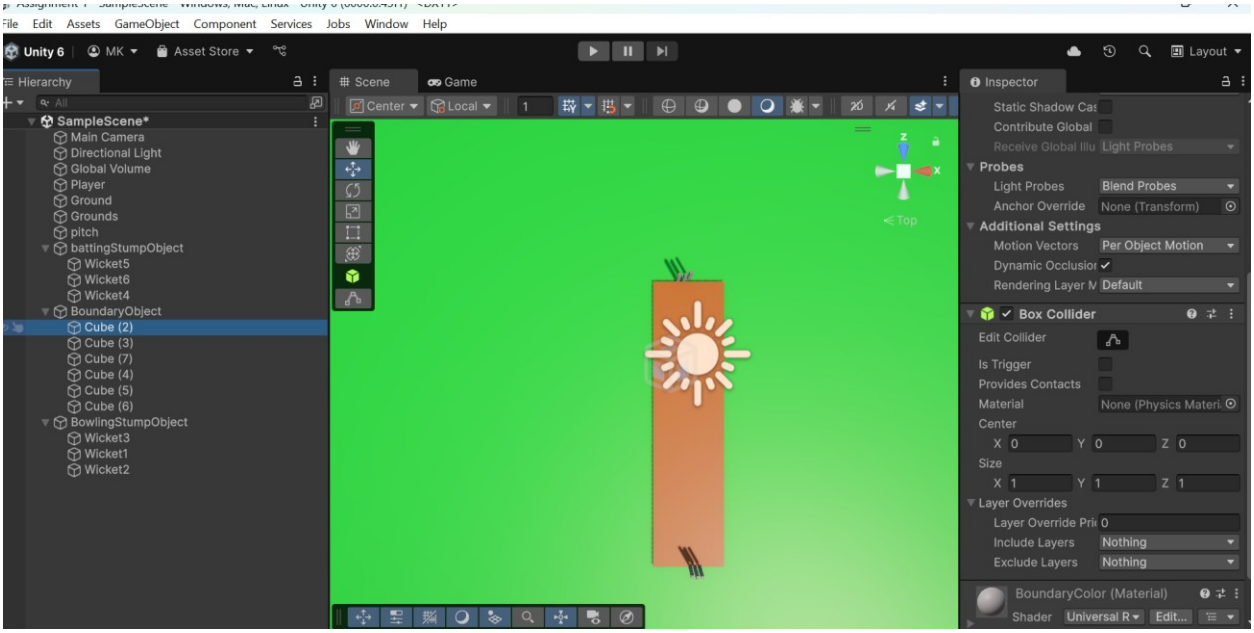


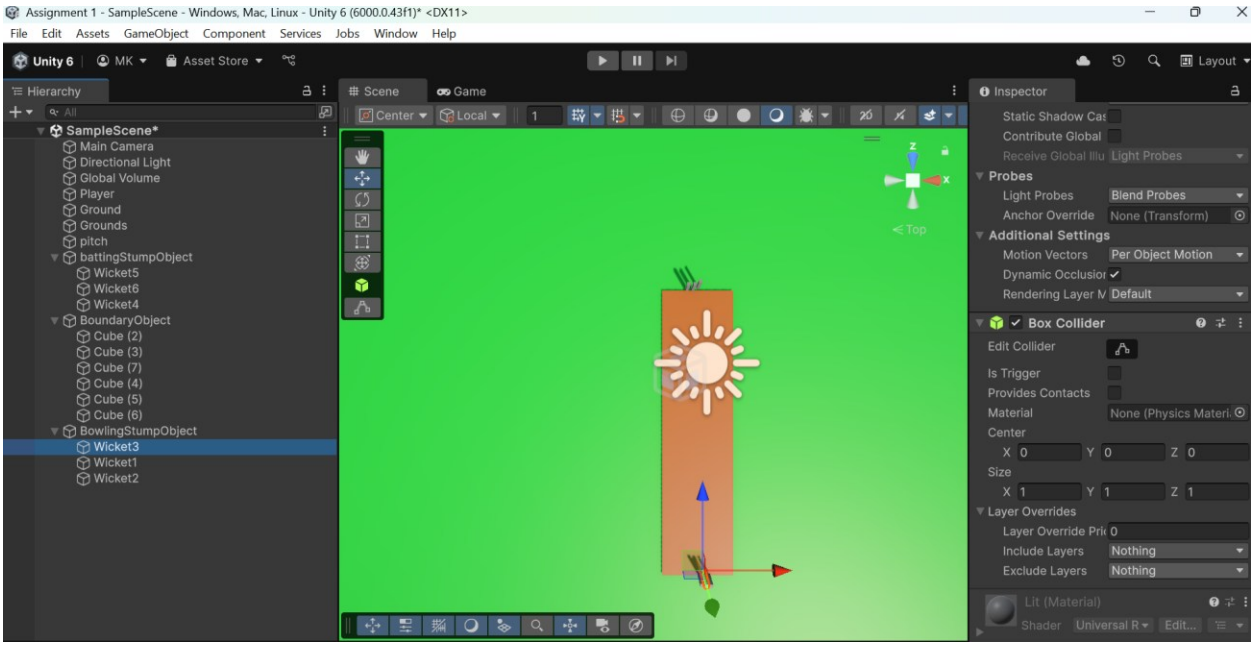
11. Did you apply Colliders to all platforms and obstacles so that physics interactions work correctly?

Answer: Yes, BoxColliders were added to:

- The ground plane (to prevent falling)
- Wickets and boundary objects
- Other elements like the ball and bat

This ensures proper physics interactions in the scene.



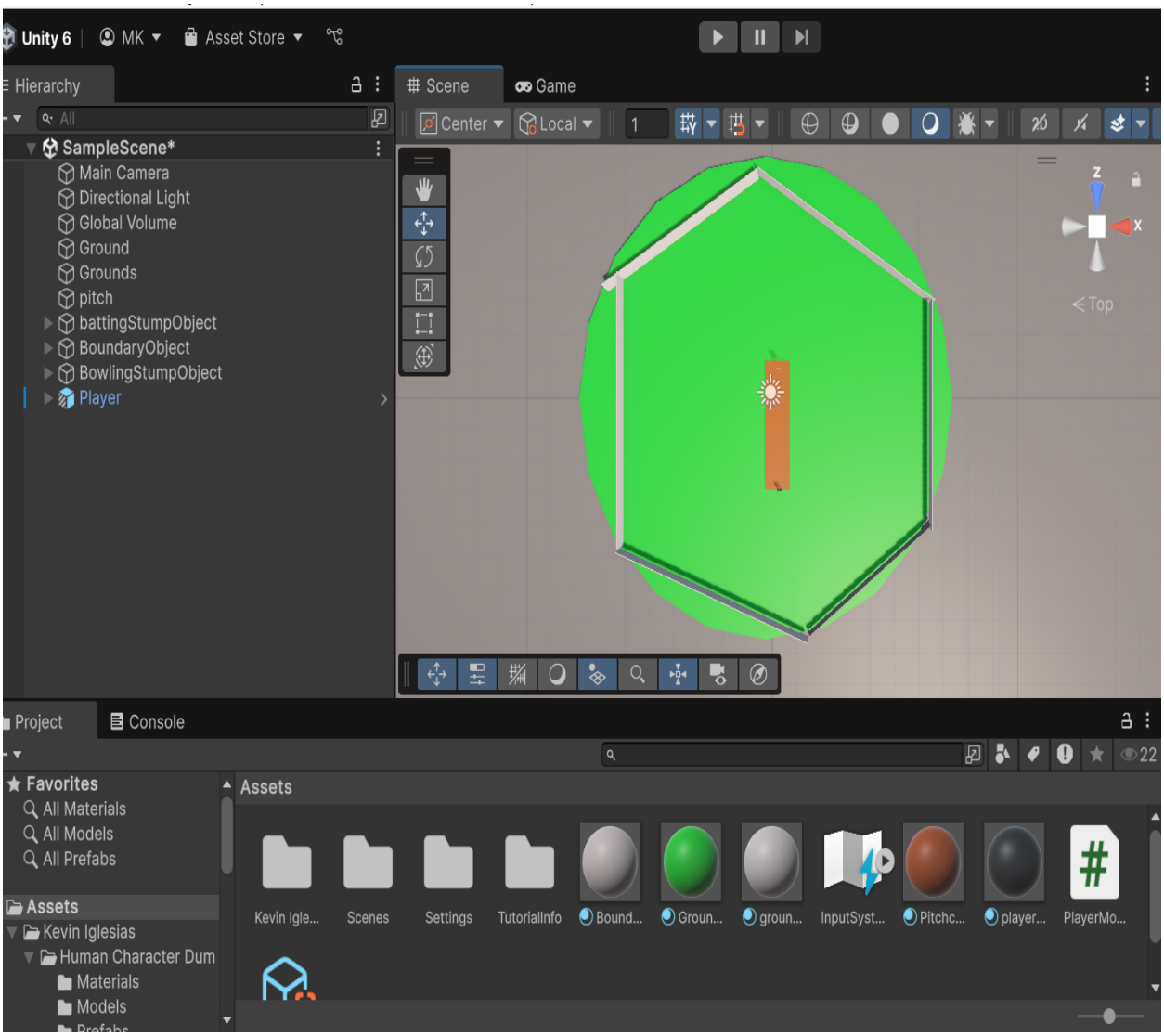


12. Have you tested that your player doesn't fall through the floor and collides properly with level

Answer: Yes, I tested the game and confirmed:

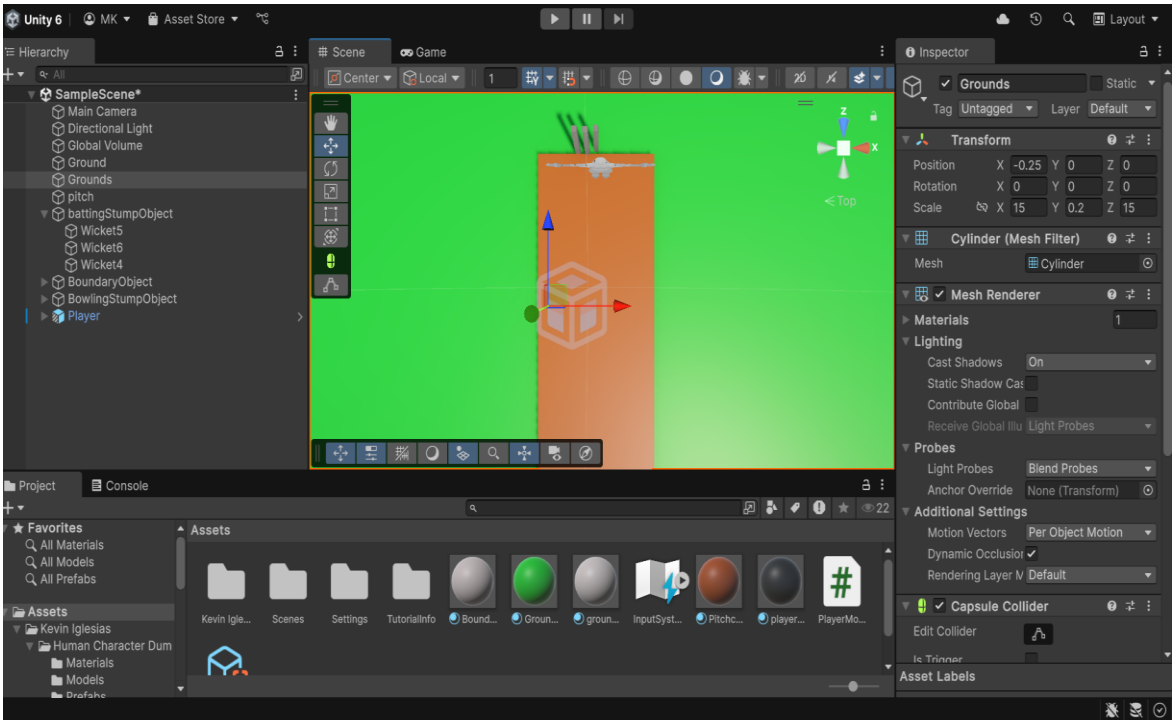
- The player stays on the pitch (doesn't fall through)
- Collisions with obstacles like stumps and boundaries are working correctly

- Rigidbody and Collider settings are properly configured



13. Have you added at least one obstacle or challenge element to the level?

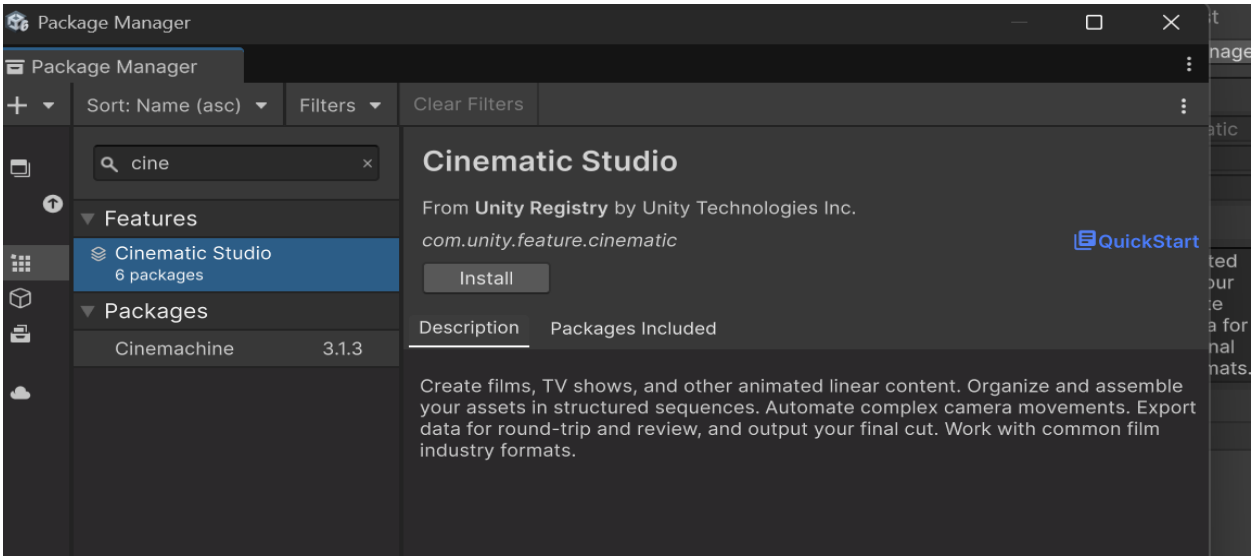
Answer: Yes, I added wickets as obstacles in front of the batsman and fielders as challenge elements. If the ball hits the stumps or is caught by a fielder (controlled by AI), the level restarts or an event triggers, adding challenge to the gameplay.



Part 4: Cinemachine Camera Setup

14. Have you imported the Cinemachine package from the Unity Package Manager?

Answer: Yes, the Cinemachine package was successfully imported using the Unity Package Manager. It adds powerful camera control tools ideal for tracking players or action dynamically.



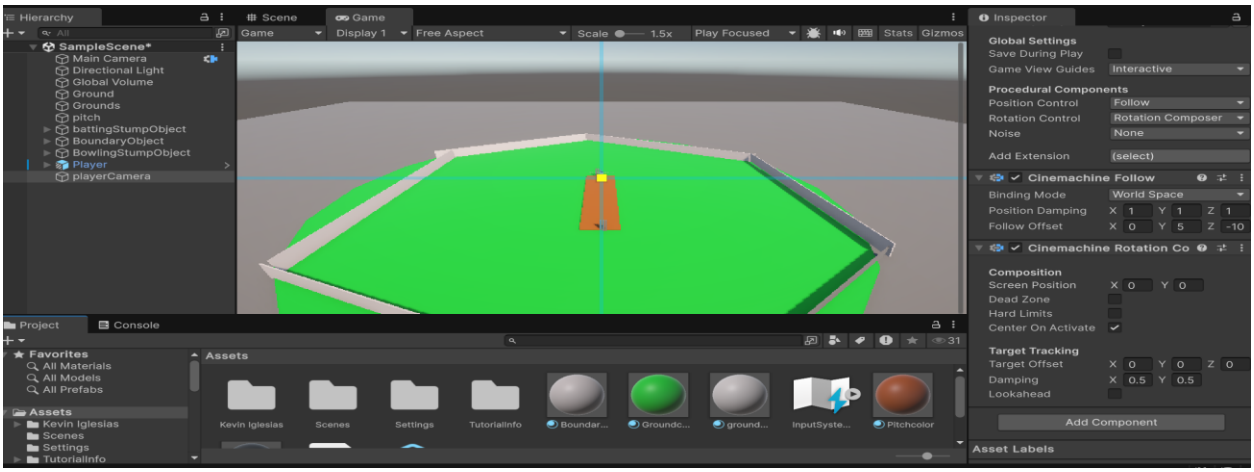
15. Did you add a Cinemachine Virtual Camera to your scene?

Answer: Yes, I added a Cinemachine Virtual Camera to the scene.

The camera follows the player (batsman) during normal gameplay.

When the player hits a shot, the camera switches to follow the ball to show its trajectory.

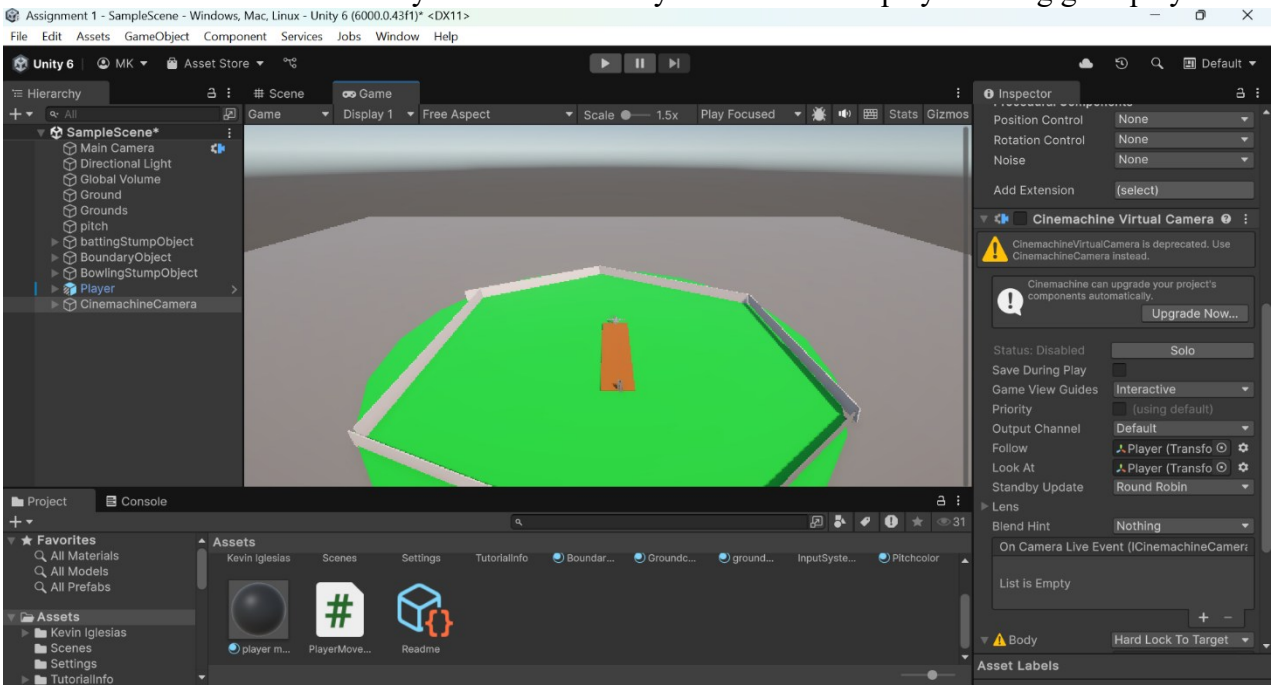
This adds a dynamic and realistic feel to the gameplay.



16. Have you assigned your player GameObject as the Follow and Look At targets in the virtual camera?

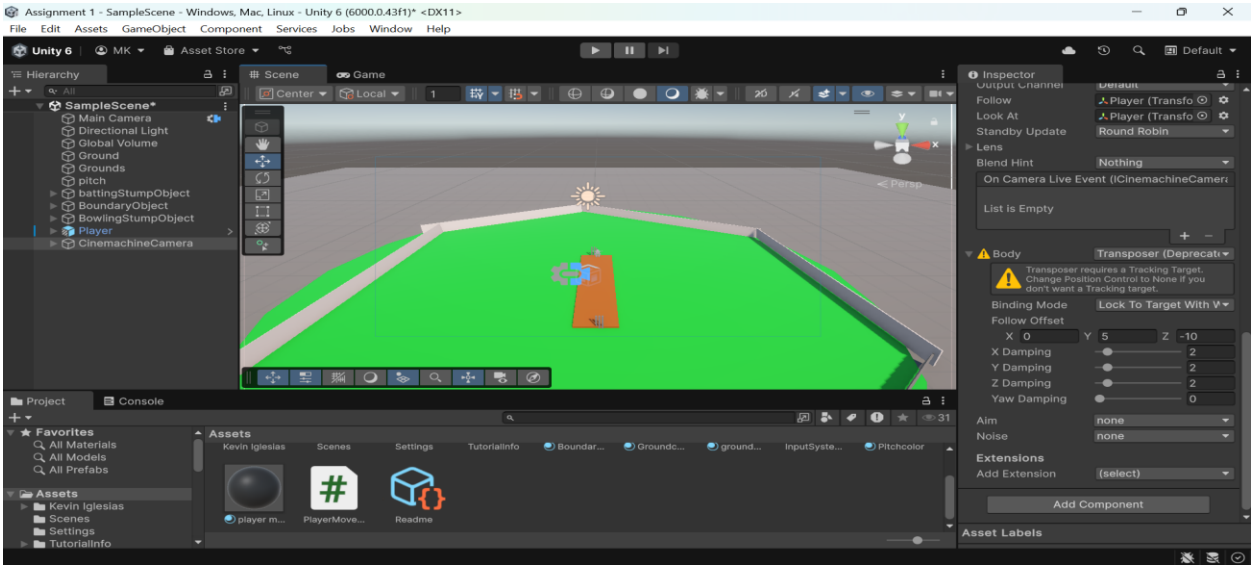
Answer: Yes, I assigned my Player GameObject as both the **Follow** and **Look At** targets in the Cinemachine Virtual Camera.

This ensures the camera smoothly follows and always focuses on the player during gameplay.



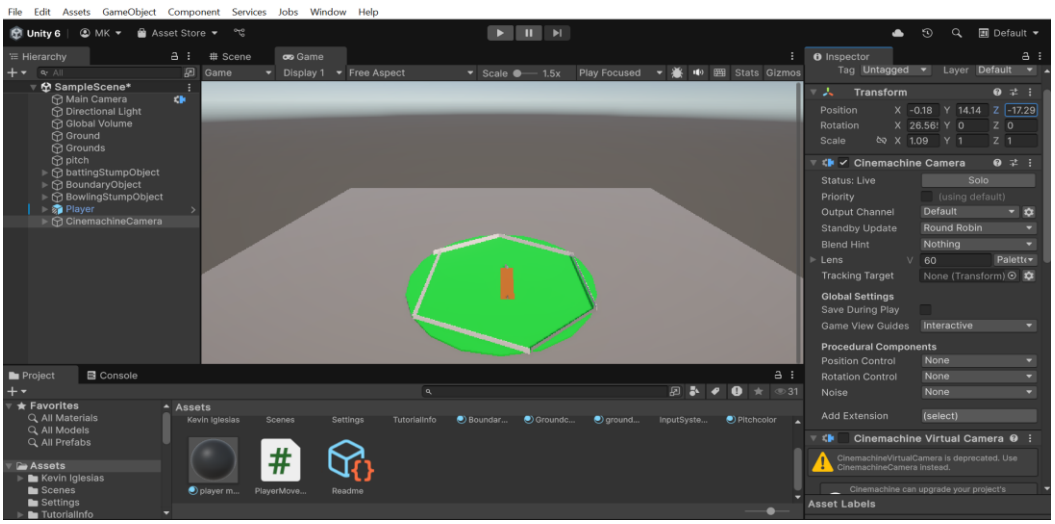
17. Did you adjust camera settings like damping or offset for a smooth follow effect?

Answer: Yes, I adjusted the camera settings to achieve a smooth follow effect.
I set the Damping values (X=2, Y=2, Z=2) and adjusted the Follow Offset (X=0, Y=5, Z=-10). This made the camera follow the player smoothly and kept the gameplay experience natural and cinematic.



18. Is the camera tracking the player properly as they move through the level?

Answer: Yes, the camera is properly tracking the player.
The Cinemachine Virtual Camera follows the player's movement smoothly across the level using the **Framing Transposer** with a configured **Follow Offset**.
As the player moves left or right, the camera adjusts its position without sudden jerks, maintaining a smooth and professional gameplay experience.



Github Link:

https://github.com/Moonkhaan/GameDevelopmentProject