COMSATS University
Islamabad
Abbottabad Campus

LAB MID

SUBMITTED BY:

ZAROON TAHIR

FA21-BSE-168

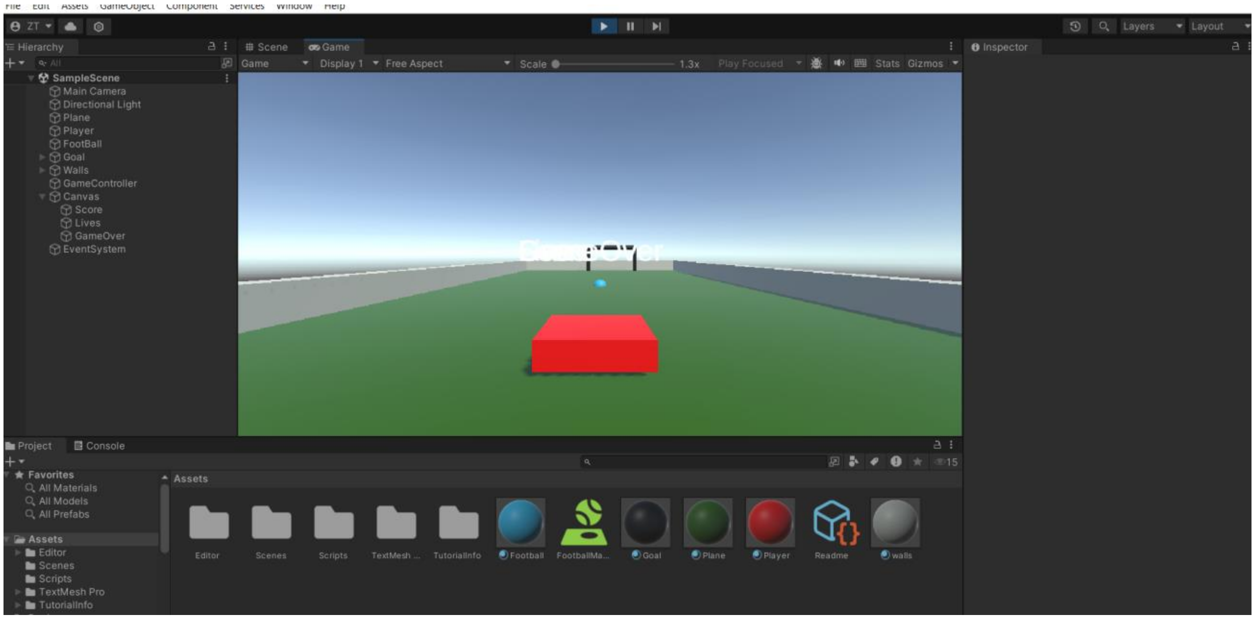
SECTION:

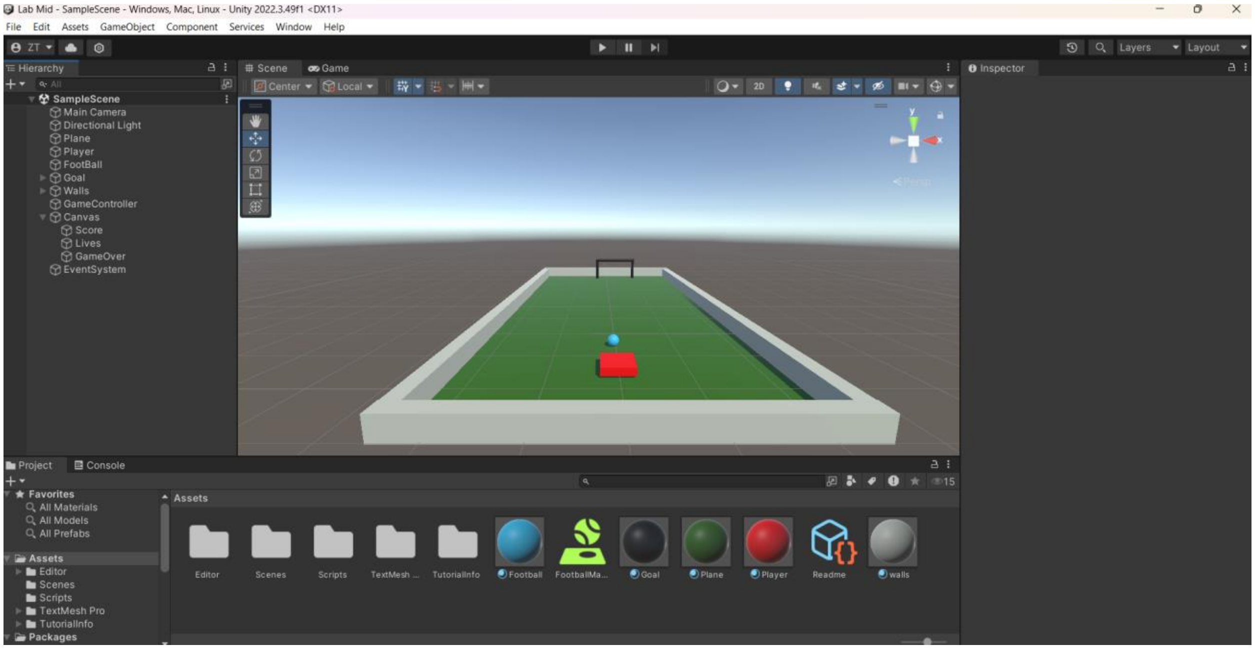
BSE-7C

SUBMITTED TO:

SIR MAZHAR BUKHARI

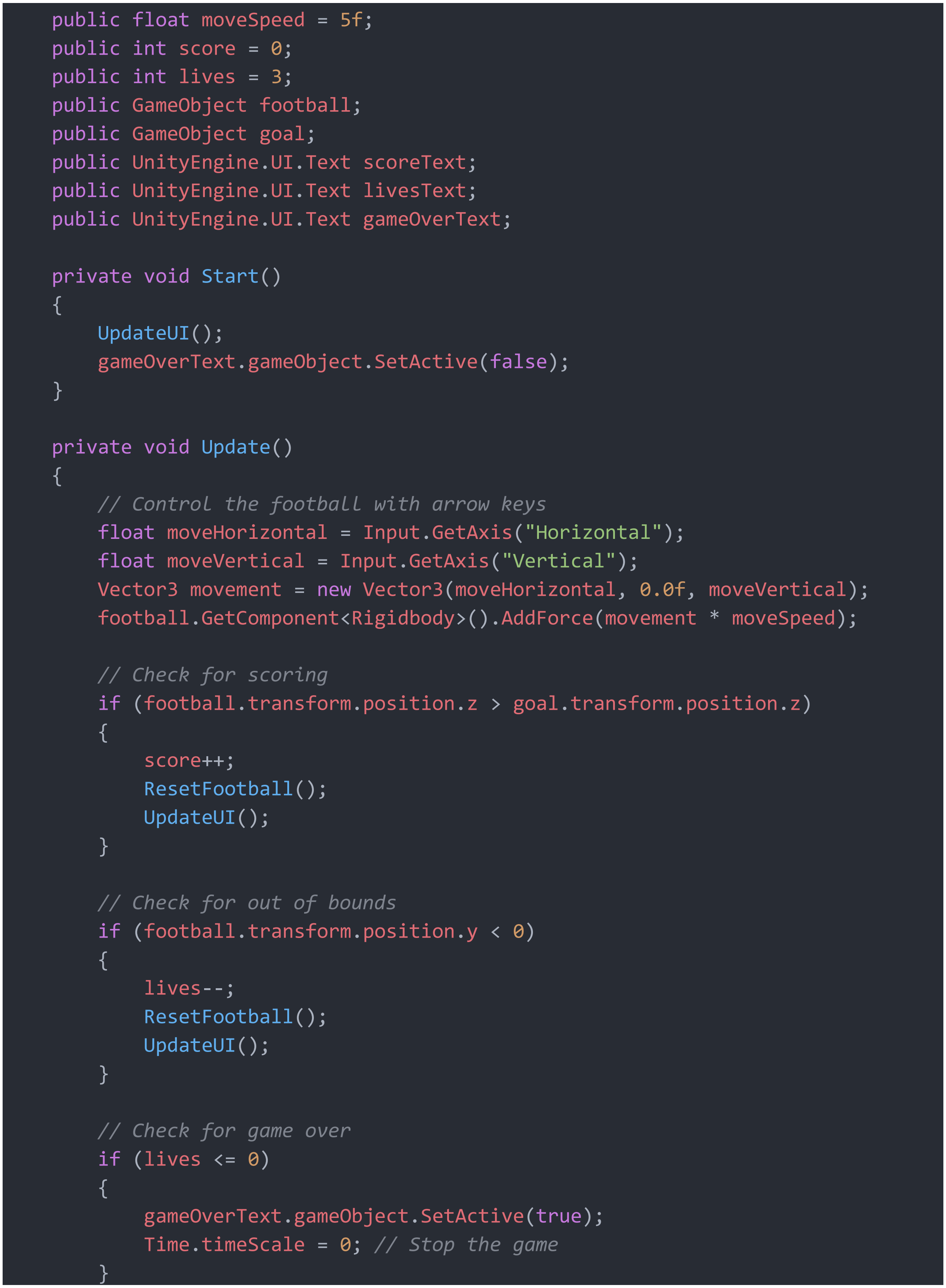
ScreenShoots:



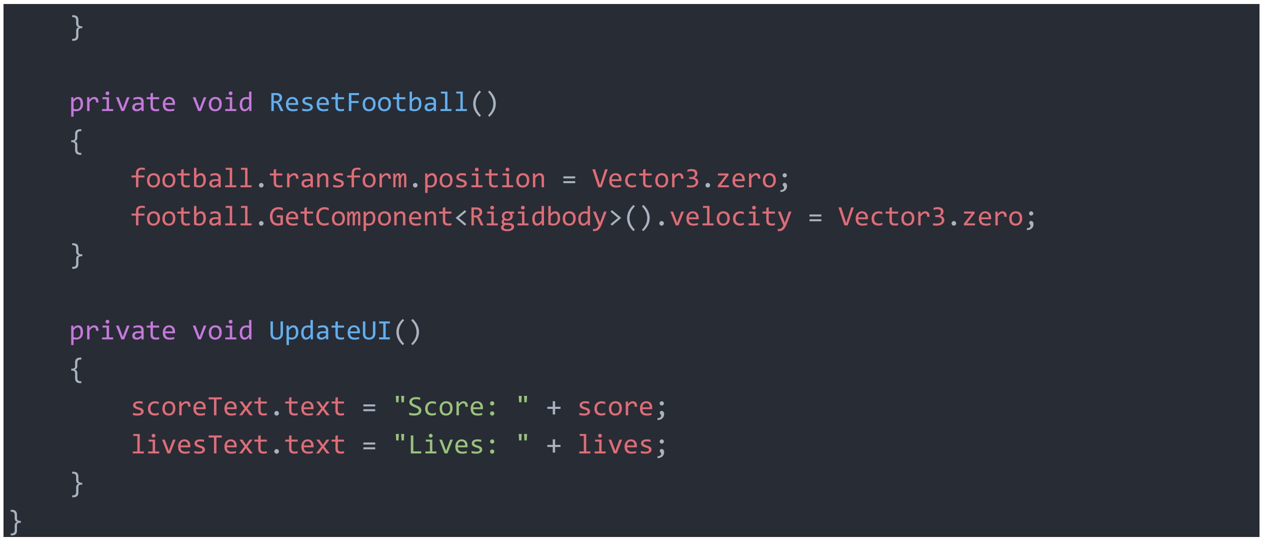


CODE:

```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
```



```
    public float moveSpeed = 5f;
    public int score = 0;
    public int lives = 3;
    public GameObject football;
    public GameObject goal;
    public UnityEngine.UI.Text scoreText;
    public UnityEngine.UI.Text livesText;
    public UnityEngine.UI.Text gameOverText;

    private void Start()
    {
        UpdateUI();
        gameOverText.gameObject.SetActive(false);
    }

    private void Update()
    {
        // Control the football with arrow keys
        float moveHorizontal = Input.GetAxis("Horizontal");
        float moveVertical = Input.GetAxis("Vertical");
        Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);
        football.GetComponent<Rigidbody>().AddForce(movement * moveSpeed);

        // Check for scoring
        if (football.transform.position.z > goal.transform.position.z)
        {
            score++;
            ResetFootball();
            UpdateUI();
        }

        // Check for out of bounds
        if (football.transform.position.y < 0)
        {
            lives--;
            ResetFootball();
            UpdateUI();
        }

        // Check for game over
        if (lives <= 0)
        {
            gameOverText.gameObject.SetActive(true);
            Time.timeScale = 0; // Stop the game
        }
```



```
    }

    private void ResetFootball()
    {
        football.transform.position = Vector3.zero;
        football.GetComponent<Rigidbody>().velocity = Vector3.zero;
    }

    private void UpdateUI()
    {
        scoreText.text = "Score: " + score;
        livesText.text = "Lives: " + lives;
    }
}
```

Explanation:

1. Environment Setup

Football Field with Boundary Lines

- **Why?**: This sets the stage for the gameplay.
- **How?**
  - A **Plane** object represents the football field. You can apply a grass texture for realism.
  - Boundary lines are either textures or white **Cube** objects placed around the edges.
  - **Box Colliders** around the field act as boundaries to detect when the football goes out of bounds.

Goal

- **Why?**: This is the main objective for scoring.
- **How?**
  - Use a **3D Cube** or a prefab as the goal.
  - Add a **Box Collider** set as a **Trigger**. This allows Unity to detect when the football enters the goal without stopping the ball.

Football

- **Why?**: To simulate realistic physics like rolling and bouncing.
- **How?**
  - Add a **Sphere** object.
  - Attach a **Rigidbody** for movement physics.

    - Use a **Physics Material** with bounciness to make the ball behave like a real football.

2. Player Mechanics

New Input System

- **Why?**: The New Input System provides modern and flexible input handling.
- **How?**
    - Install the package via the **Package Manager**.
    - Define an **Input Action Asset** to map player inputs to movement.
    - Generate a C# class from the asset to integrate it with scripts.

Player Control Script

- **Why?**: To allow the player to move the football using arrow keys or `WASD`.
- **How?**
    - Capture movement input using `InputAction.CallbackContext`.
    - Apply the input to the football's **Rigidbody** as a force for realistic movement.

Trigger Detection

- **Why?**: To update the score or lives based on football interactions.
- **How?**
    - Use **Trigger Colliders**:
        - When the football enters the goal, increase the score.
        - If the football exits the boundary, decrease lives.

3. Game UI

Score and Lives Display

- **Why?**: Players need feedback on their progress.
- **How?**
    - Use Unity's **UI Text** elements in a Canvas.
    - Dynamically update the text in a script when the score or lives change.

Game Over

- **Why?**: To indicate the end of the game.
- **How?**
    - Display a "Game Over" message when lives reach zero.

    - Pause the game using `Time.timeScale = 0`.

4. Bonus Features

Sound Effects

- **Why?**: To enhance gameplay feedback and immersion.
- **How?**
    - Play sounds when:
        - The ball bounces (via `OnCollisionEnter`).
        - A goal is scored (via `OnTriggerEnter`).
        - The ball goes out of bounds.

Timer

- **Why?**: To add an additional challenge.
- **How?**
    - Use a `gameTime` variable to count down from 120 seconds.
    - Update the timer display in real-time.
    - End the game when the timer reaches zero.

Submission Checklist

The checklist ensures all the required features are implemented and working:

1. **Environment**: A football field with boundaries, a goal, and a bouncy football.
2. **Player Mechanics**: Smooth control of the football with the new input system.
3. **Game UI**: Dynamic score and lives display, with a Game Over message.
4. **Bonus Features**: Optional sound effects and a game timer.