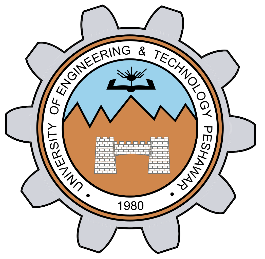
**Unity Animations and Animator**

**LAB # 9**



**Fall 2024**

**CSE-411L Intro to Game Development Lab**

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Abdullah Hamid**

Date:

**25th December 2024**

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objective:**

In this lab we further explored the Unity API.

**Tasks:**

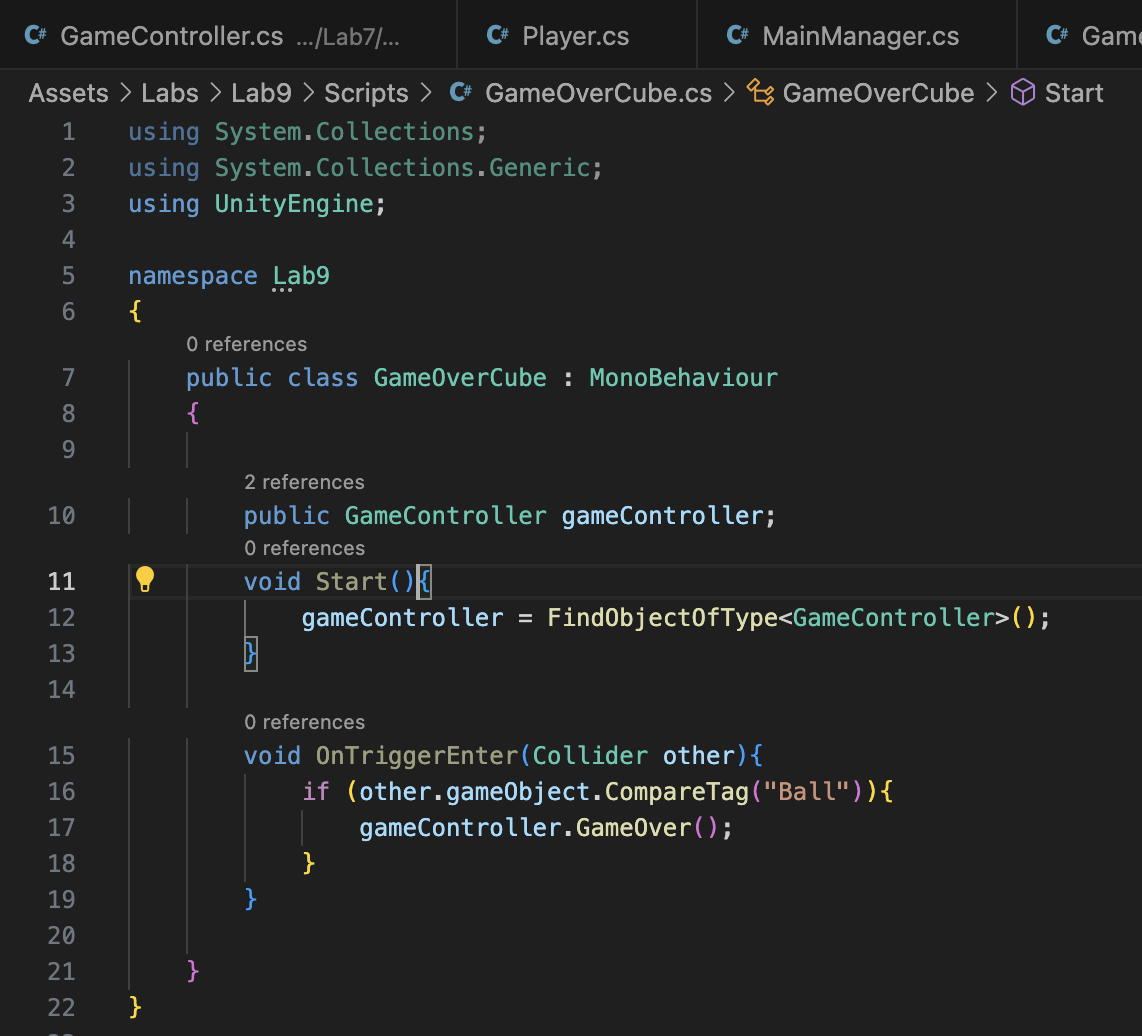
1. **Set Up the Scene**:
   1. Open or create a new scene in Unity.
   2. In this scene, create your game layout and add a main panel with a “Play” button.
2. **Pause the Game on Start**:
   1. Ensure the game is paused when it starts.
   2. When the “Play” button is clicked:
   3. Animate the main panel to slide up and out of the main camera’s view.
   4. Resume the game.
3. **Download Assets**:
   1. Go to Mixamo and download a character model along with basic animations like **walk** and **idle**.
4. **Convert to Humanoid**:
   1. Convert the character and animations to **Humanoid** rig in Unity.
   2. (See conversion method on the next page).
5. **Set Up Animator Controller**:
   1. Create a new **Animator Controller** for the character.
   2. This character will act as the main player in your game.
6. **Idle and Walk Animation**:
   1. Set the default animation state of the player to **Idle**.
   2. Configure the animator so that pressing W, A, S, or D will:
   3. Trigger the **Walk** animation.
   4. Move the player in the corresponding direction.
7. **Design the Game Scene**:
   1. Add a **Plane** as the ground.
   2. Place the player and a **Ball** on the plane.
   3. Apply a **Rigidbody** and **Bouncy Physics Material** to the ball.
   4. Ensure the player can push the ball towards a goal on the plane.
8. **Goal Mechanic**:
   1. Add a **Goal** object to the scene.
   2. The goal should have a **color-changing animation**.
   3. When the ball reaches the goal:
   4. Display the UI message: **“Game Complete!”**
   5. End the game.
9. **Game Over Mechanic**:
   1. Add an **Invisible Collider** underneath the plane.
   2. If the ball falls off the plane and collides with this object:
   3. Display the UI message: **“Game Over, you lose!”**
   4. End the game.

**Code:**

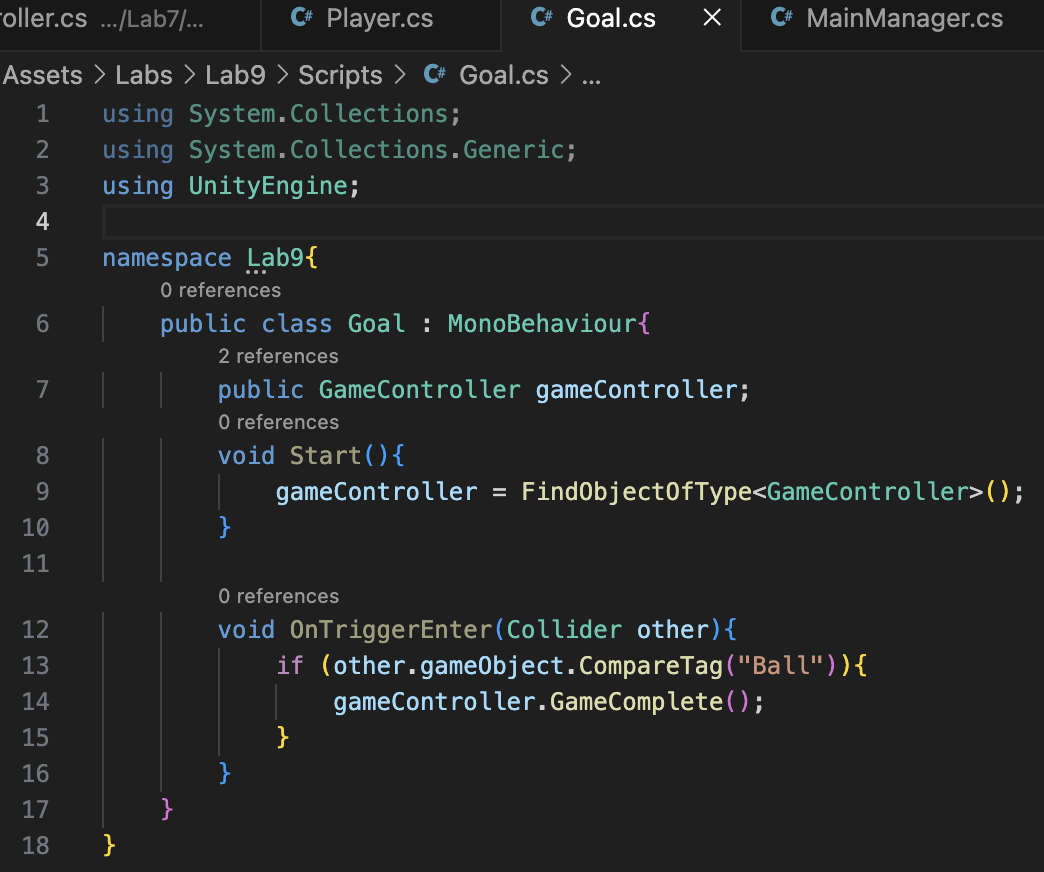
GameController class

****

GameOverCube class

****

Goal class

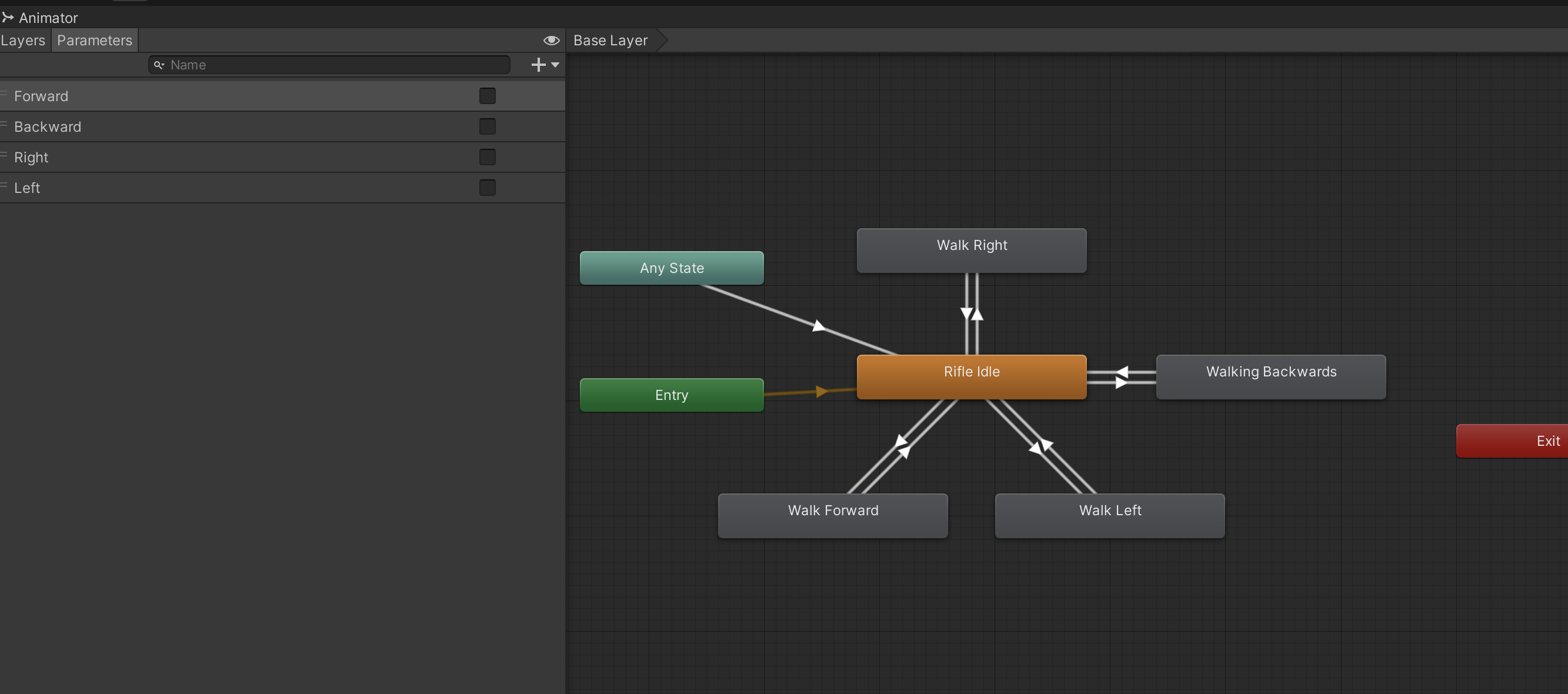
****

PlayerController class

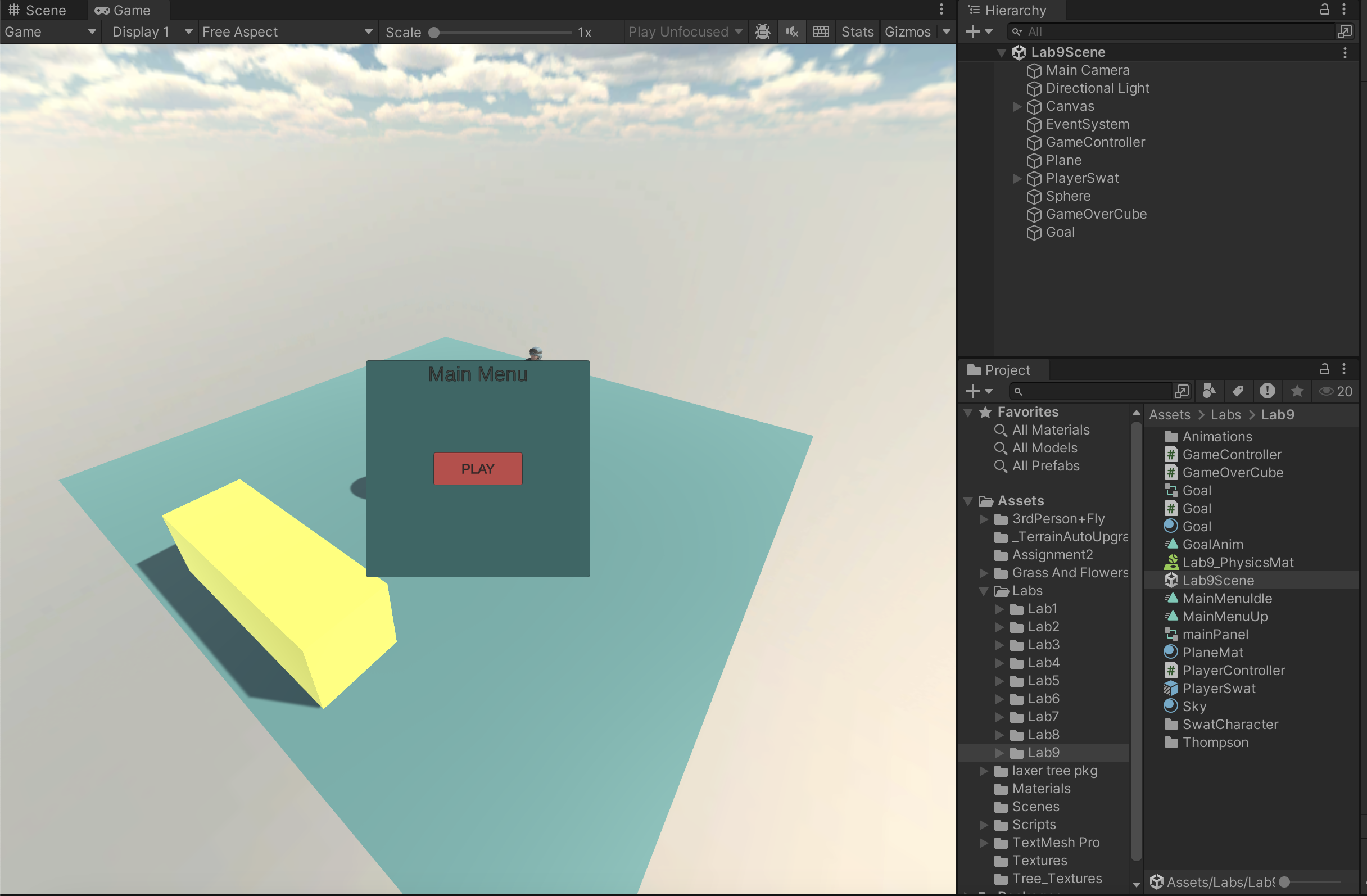
****

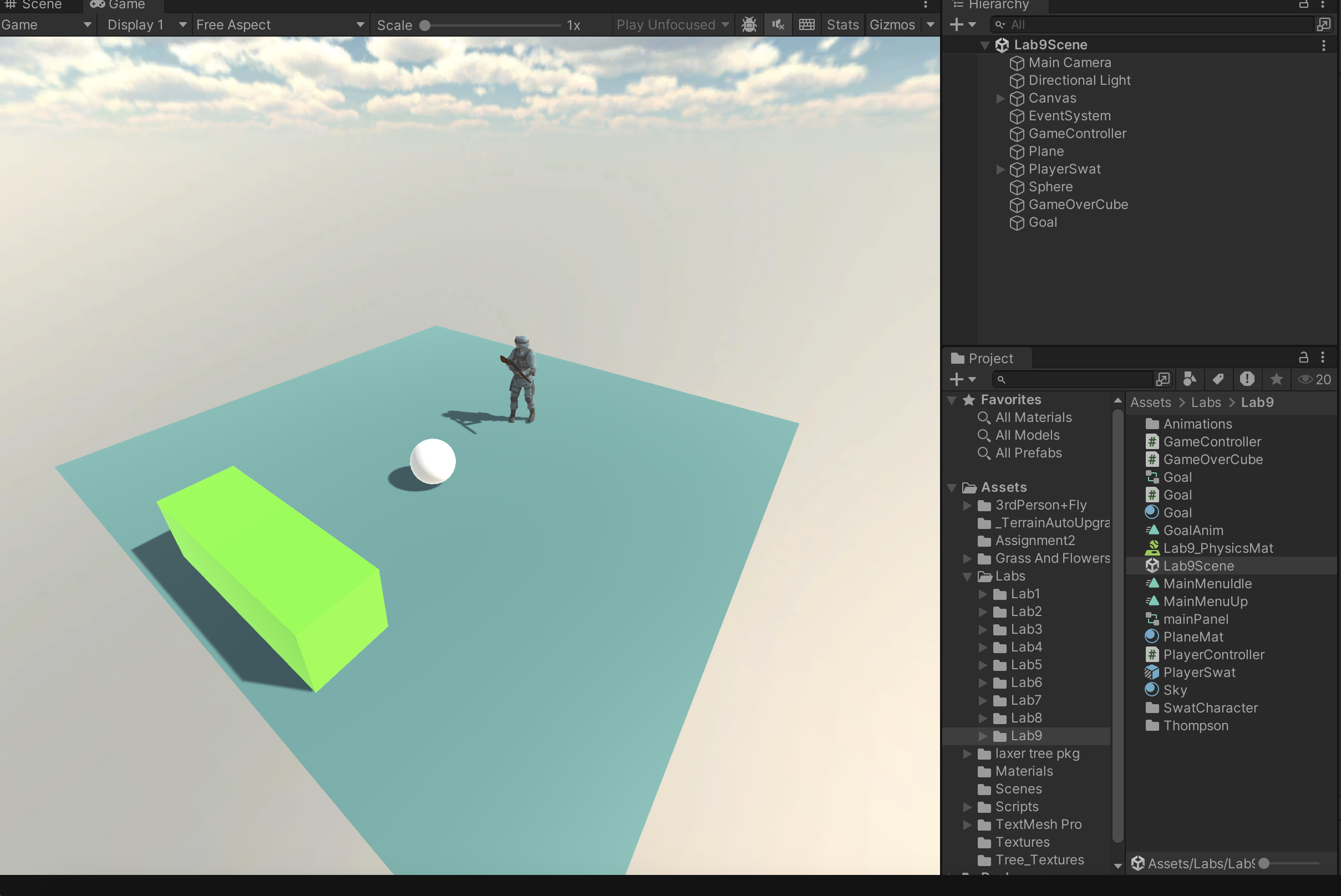
****

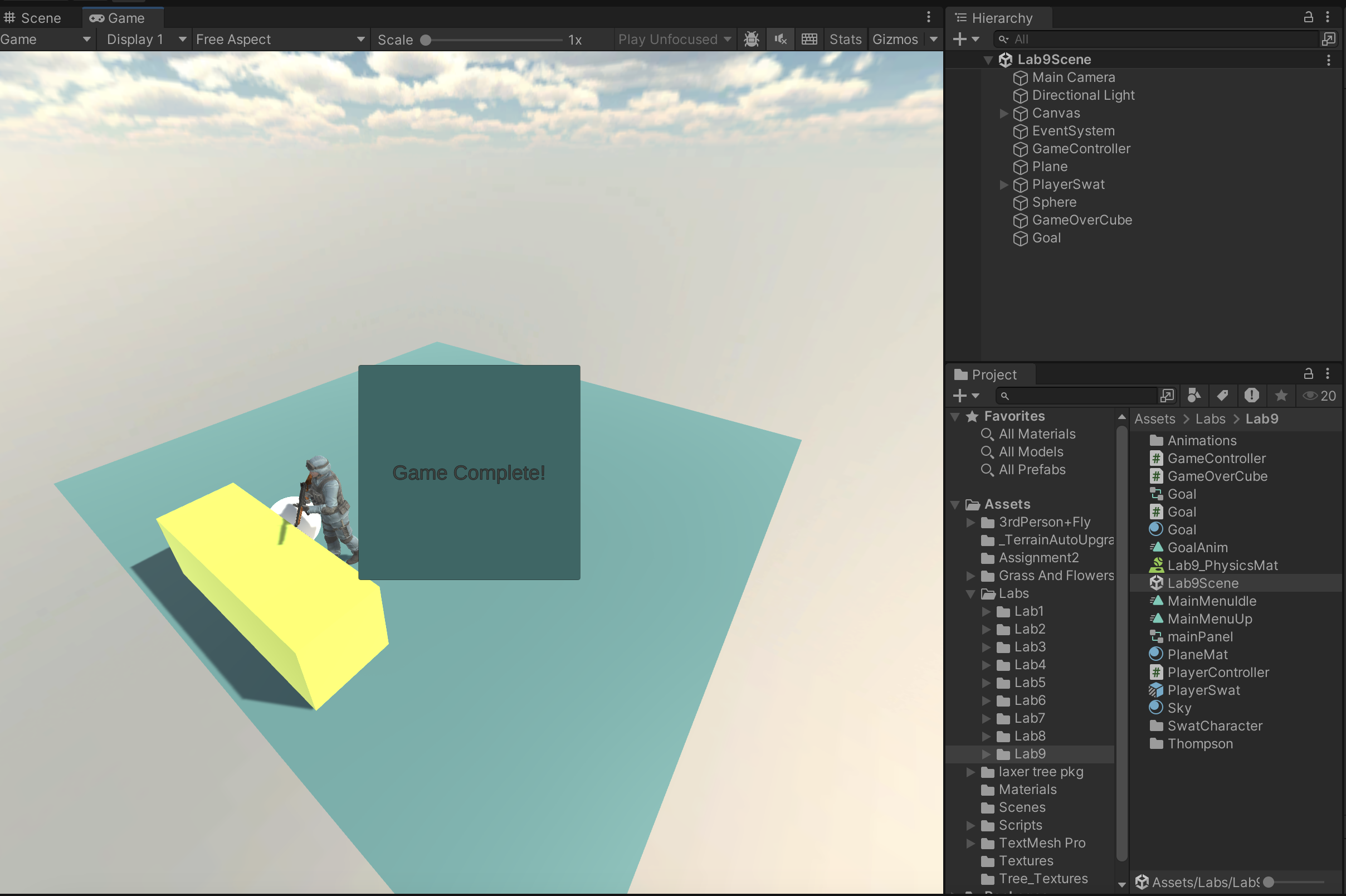
**Animator FSM:**

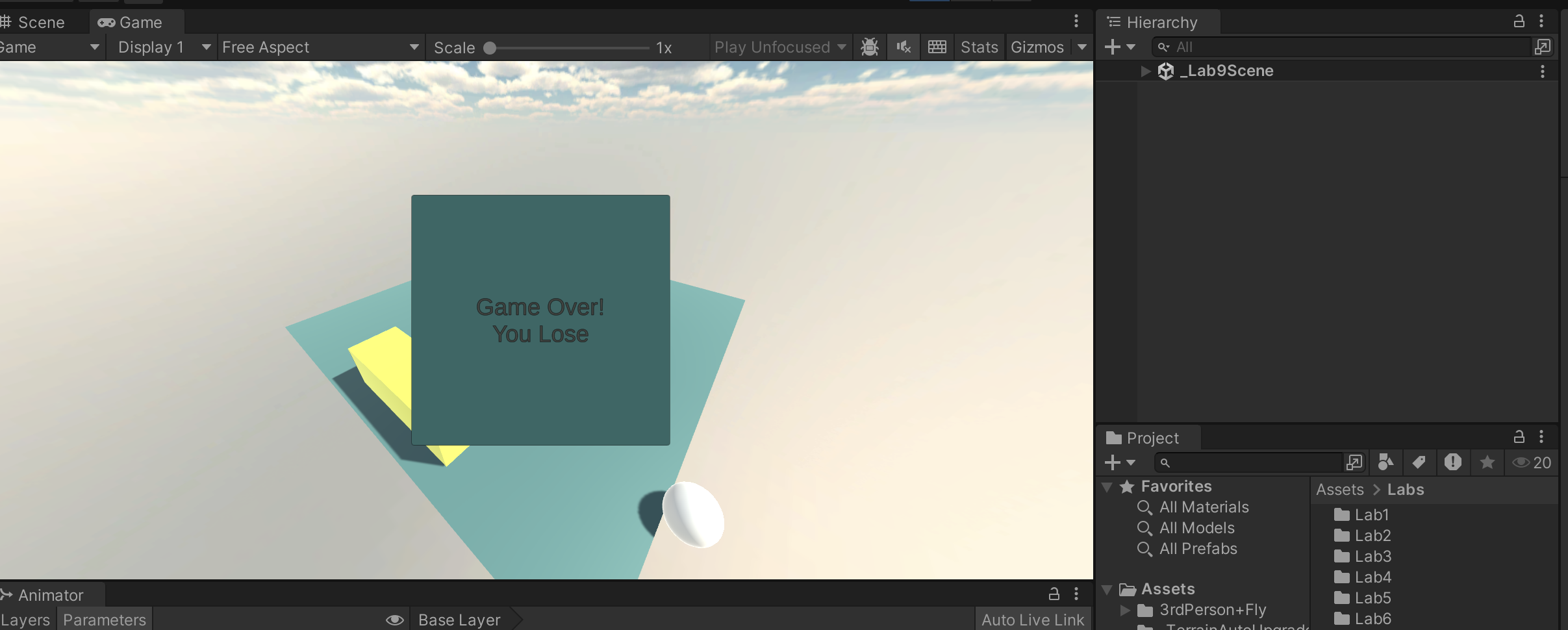
****

**Output:**

****

****

****

****