File: PlayerController.cs

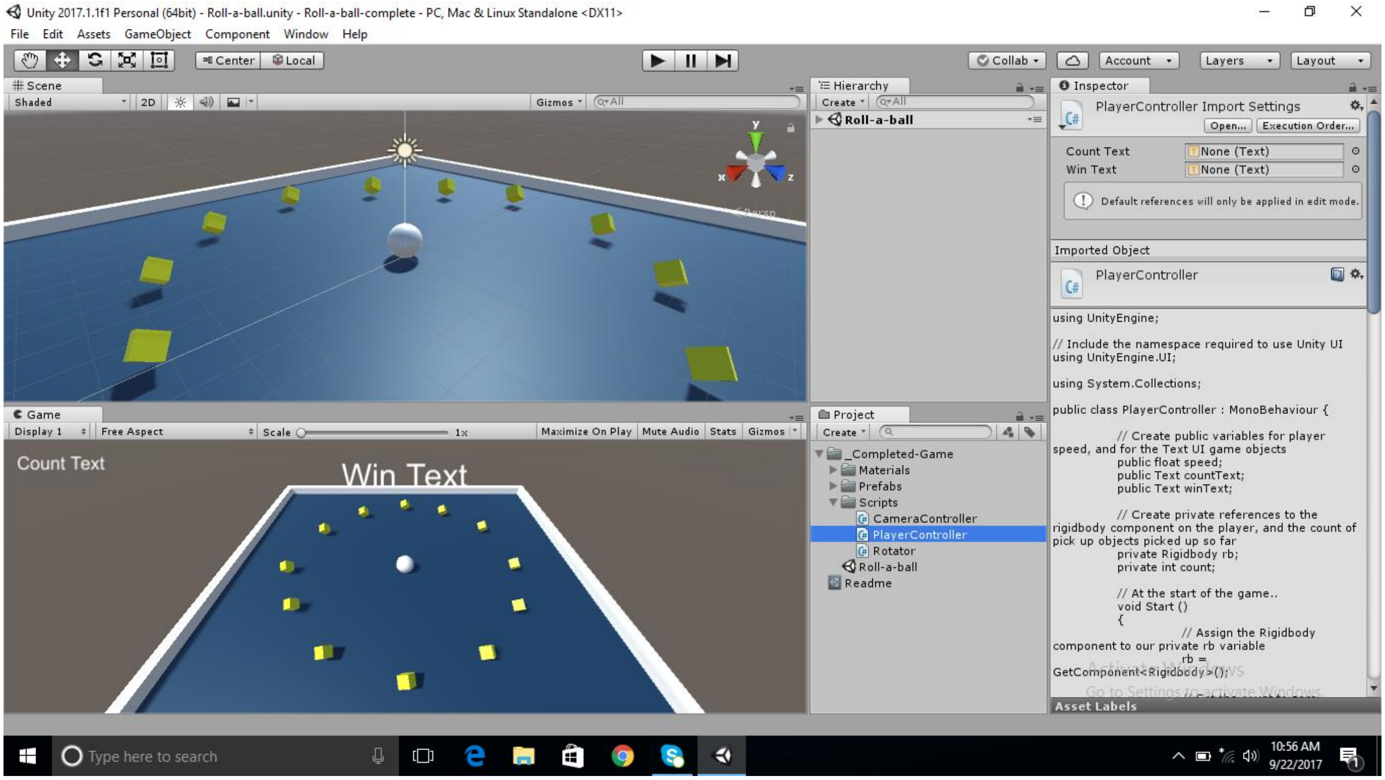
```
using UnityEngine;

// Include the namespace required to use Unity UI
using UnityEngine.UI;

using System.Collections;

public class PlayerController : MonoBehaviour {

	// Create public variables for player speed, and for the Text UI
game objects
	public float speed;
	public Text countText;
	public Text winText;

	// Create private references to the rigidbody component on the
player, and the count of pick up objects picked up so far
	private Rigidbody rb;
	private int count;

	// At the start of the game..
	void Start ()
	{
		// Assign the Rigidbody component to our private rb variable
		rb = GetComponent<Rigidbody>();

		// Set the count to zero
		count = 0;

		// Run the SetCountText function to update the UI (see below)
		SetCountText ();

		// Set the text property of our Win Text UI to an empty string,
making the 'You Win' (game over message) blank
		winText.text = "";
	}

	// Each physics step..
	void FixedUpdate ()
	{
		// Set some local float variables equal to the value of our
Horizontal and Vertical Inputs
		float moveHorizontal = Input.GetAxis ("Horizontal");
		float moveVertical = Input.GetAxis ("Vertical");

		// Create a Vector3 variable, and assign X and Z to feature our
horizontal and vertical float variables above
		Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
moveVertical);

		// Add a physical force to our Player rigidbody using our
'movement' Vector3 above,
		// multiplying it by 'speed' - our public player speed that
appears in the inspector
		rb.AddForce (movement * speed);
```

```
    }

    // When this game object intersects a collider with 'is trigger' 
checked, 
    // store a reference to that collider in a variable named 'other'..
    void OnTriggerEnter(Collider other) 
    {
        // ..and if the game object we intersect has the tag 'Pick Up' 
assigned to it..
        if (other.gameObject.CompareTag ("Pick Up"))
        {
            // Make the other game object (the pick up) inactive, to 
make it disappear
            other.gameObject.SetActive (false);

            // Add one to the score variable 'count'
            count = count + 1;

            // Run the 'SetCountText()' function (see below)
            SetCountText ();
        }
    }

    // Create a standalone function that can update the 'countText' UI 
and check if the required amount to win has been achieved
    void SetCountText()
    {
        // Update the text field of our 'countText' variable
        countText.text = "Count: " + count.ToString ();

        // Check if our 'count' is equal to or exceeded 12
        if (count >= 12) 
        {
            // Set the text value of our 'winText'
            winText.text = "You Win!";
        }
    }
}
```

File: Rotator.cs

```
using UnityEngine;
using System.Collections;

public class Rotator : MonoBehaviour {

    // Before rendering each frame..
    void Update () 
    {
        // Rotate the game object that this script is attached to by 15 
in the X axis,
        // 30 in the Y axis and 45 in the Z axis, multiplied by 
deltaTime in order to make it per second
        // rather than per frame.
        transform.Rotate (new Vector3 (15, 30, 45) * Time.deltaTime);
    }
}
```

OUTPUT:


Unity 2017.1.1f1 Personal (64bit) - Roll-a-ball.unity - Roll-a-ball-complete - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component Window Help
Center Local
Collab Account Layers Layout
Scene
Shaded 2D Gizmos
Hierarchy
Create
Roll-a-ball
Inspector
Game
Display 1 Free Aspect Scale 1x Maximize On Play Mute Audio Stats Gizmos
Count Text
Win Text
Project
Create
_Completed-Game
Materials
Prefabs
Scripts
CameraController
PlayerController
Rotator
Roll-a-ball
Readme
Asset Labels
Type here to search
10:56 AM
9/22/2017

PlayerController Import Settings

Open... Execution Order...

Count Text: None (Text)

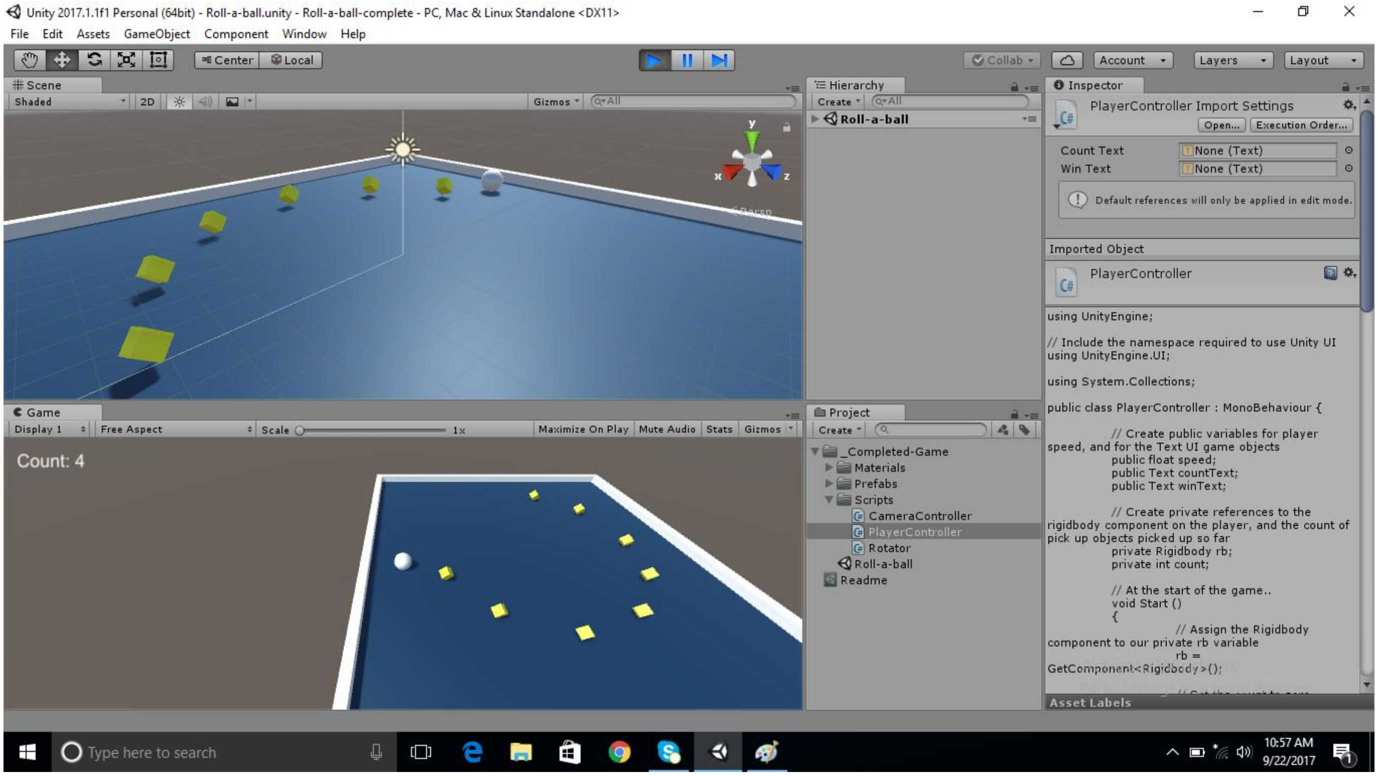
Win Text: None (Text)

Default references will only be applied in edit mode.

Imported Object

PlayerController

```
using UnityEngine;

// Include the namespace required to use Unity UI
using UnityEngine.UI;

using System.Collections;

public class PlayerController : MonoBehaviour {

		// Create public variables for player
speed, and for the Text UI game objects
		public float speed;
		public Text countText;
		public Text winText;

		// Create private references to the
rigidbody component on the player, and the count of
pick up objects picked up so far
		private Rigidbody rb;
		private int count;

		// At the start of the game..
		void Start ()
		{
				// Assign the Rigidbody
component to our private rb variable
				rb =
GetComponent<Rigidbody>();
```


Unity 2017.1.1f1 Personal (64bit) - Roll-a-ball.unity - Roll-a-ball-complete - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component Window Help
Center Local
Collab Account Layers Layout
Scene
Shaded 2D Gizmos
Hierarchy
Create
Roll-a-ball
Inspector
Game
Display 1 Free Aspect Scale 1x Maximize On Play Mute Audio Stats Gizmos
Count: 4
Project
Create
_Completed-Game
Materials
Prefabs
Scripts
CameraController
PlayerController
Rotator
Roll-a-ball
Readme
Asset Labels
Type here to search
10:57 AM
9/22/2017

PlayerController Import Settings

Open... Execution Order...

Count Text: None (Text)

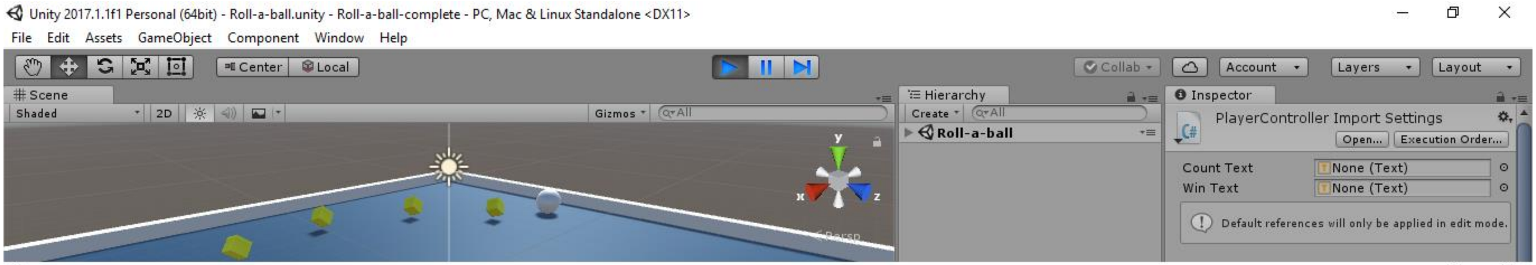
Win Text: None (Text)

Default references will only be applied in edit mode.

Imported Object

PlayerController

```
using UnityEngine;

// Include the namespace required to use Unity UI
using UnityEngine.UI;

using System.Collections;

public class PlayerController : MonoBehaviour {

		// Create public variables for player
speed, and for the Text UI game objects
		public float speed;
		public Text countText;
		public Text winText;

		// Create private references to the
rigidbody component on the player, and the count of
pick up objects picked up so far
		private Rigidbody rb;
		private int count;

		// At the start of the game..
		void Start ()
		{
				// Assign the Rigidbody
component to our private rb variable
				rb =
GetComponent<Rigidbody>();
```

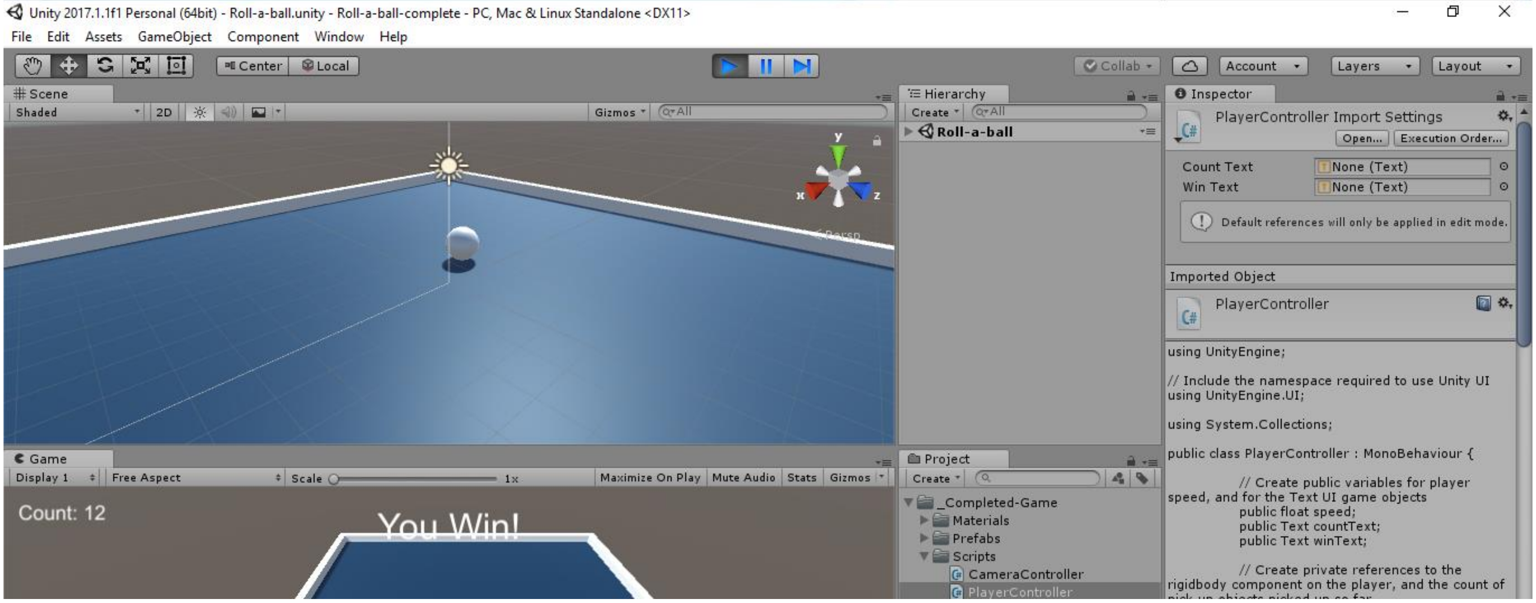


Unity 2017.1.1f1 Personal (64bit) - Roll-a-ball.unity - Roll-a-ball-complete - PC, Mac & Linux Standalone <DX11>

PlayerController Import Settings

Count Text None (Text)
Win Text None (Text)

Default references will only be applied in edit mode.



Unity 2017.1.1f1 Personal (64bit) - Roll-a-ball.unity - Roll-a-ball-complete - PC, Mac & Linux Standalone <DX11>

PlayerController Import Settings

Count Text None (Text)
Win Text None (Text)

Default references will only be applied in edit mode.

Imported Object

PlayerController

```
using UnityEngine;

// Include the namespace required to use Unity UI
using UnityEngine.UI;

using System.Collections;

public class PlayerController : MonoBehaviour {

		// Create public variables for player
speed, and for the Text UI game objects
		public float speed;
		public Text countText;
		public Text winText;

		// Create private references to the
rigidbody component on the player, and the count of
pick up objects picked up so far
```