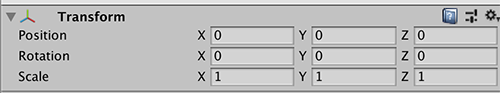
Workshop 3 Lab

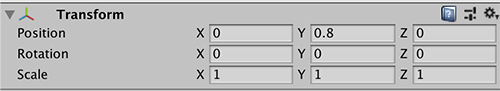
In this lab, we will create a ***Roll a Ball*** game with 3D transformation, camera, prefabs and lighting.

In this game, the player uses the keyboard arrow key to control the ball to collect rotating cubes for points.

Procedure:

1. Create a new Unity 3D project name Roll A Ball.
2. Create a 3D plane name “Ground” in the game with the following properties:



1. Create a 3D Sphere with name “Player” in the game with the following properties:
2. Set the Directional light to the following properties:

Graphical user interface, application

Description automatically generated

1. Duplicate the Directional light, rename it to “Fill Light” with the following properties:

Graphical user interface, application

Description automatically generated

(Note: set the Color to a tint of blue)

You may add in a point or spot light in the middle of the game scene.

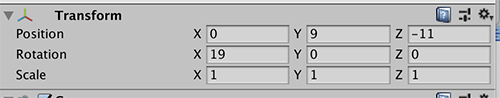
Graphical user interface

Description automatically generated

1. Select the Player. In the inspector, “Add Component” Select Physics->Rigid Body.
2. Attached a new C# script PlayerControl to the Player game object.



In the Inspector, set the speed to 500.

1. Test the game out. You should be able to control the ball using the arrow key
2. Create a new camera that will follow the Player game object.
   1. Create a new camera game object with the following transform properties:
   2. Create a new C# Script and attach it to the camera. Attach the player object to the script



* 1. Test the game again, this time you will observe that the game view will follow the Player game object.

1. To prevent the Player object from falling off the ground, create four walls around the ground plane using four cubes.
   1. Create four cubes and group them under Walls.

Text

Description automatically generated

* 1. Set each cube with the following properties:

Table

Description automatically generated

1. You should get something like this:

A picture containing graphical user interface

Description automatically generated

1. Create a new cube name “Pickup”.



* 1. Create a new C# Script RotatingCube and attach it to Pickup.



* 1. Create a new folder Prefabs under Assets. Drag the Pickup game object into the Prefabs folder.

1. Create an empty game object and drag in three Pickup prefab.

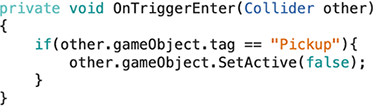


Adjust the cubes position to something like this:

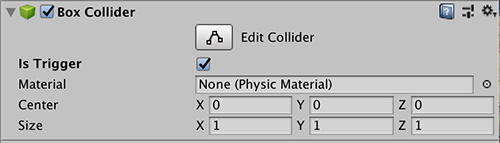


You may add more cubes to the game scene.

1. Open up the PlayerControl C# Script and add in this method.



1. Add a new tag “Pickup” and tag all Pickup game object as “Pickup”.
2. Enable Is Trigger in all Pickup game objects.



1. Test the game. When the player manoeuvres the Player game object to the Pickup game object, the Pickup game object will disappear.
2. You may add in an area light at this point.
3. Continue the game with the following enhancement:
   1. Detect the ending of the game when all Pickup objects disappear.
   2. Score.
   3. Introduction and ending screen.