BINUS UNIVERSITY

People
Innovation
Excellence

Course : 2D Game Programming

Effective Period : September 2016

2D Game Programing
LAB 02

BINUS UNIVERSITY

People
Innovation
Excellence

Acknowledgement

These slides have been adapted from:

Pereira, V. (2014). Learning Unity 2D Game Development by Example, Packt Publishing, Inc. San Francisco. ISBN: 9781783559046

Chapter 4

BINUS UNIVERSITY

Learning Objectives

LO 1 : Create 2D game for PC platform
LO 3 : Design 2D game for PC platform

People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Let’s get on with Physics

Apply Unity's 2D physics to our game

Unity's 2D physics engine is very similar to its 3D one. Almost all of the physics components have been integrated into the 2D engine with a slight difference in names (Box Collider 2D, Circle Collider 2D, Rigidbody 2D, and so on…).

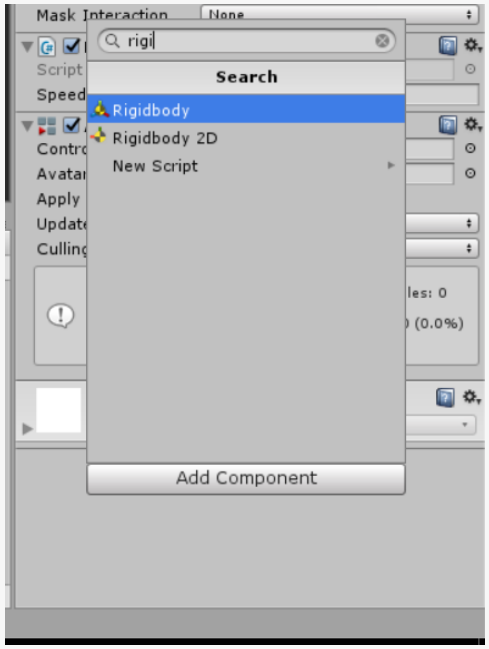
It is important to understand that 2D physics components will not interact with 3D physics components if both exist within the same scene. Even though they both share a lot of similarities, 2D physics only occur on the X and Y axis, and rotating an object using physics will only occur on the X axis.

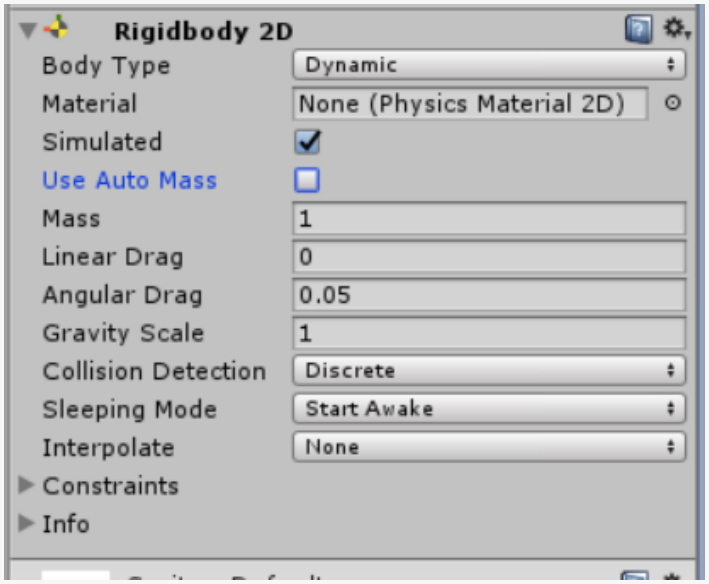
BINUS UNIVERSITY

People
Innovation
Excellence

Add rigid body 2D

Let's add Rigidbody 2D component to the player





You will see above panel in the inspector

BINUS UNIVERSITY

People
Innovation
Excellence

Rigidbody?

Rigidbody2D component controls the physical behaviour of any game object it is attached to inside the scene. It also defines its physical properties, such as mass, and how gravity should affect an object.

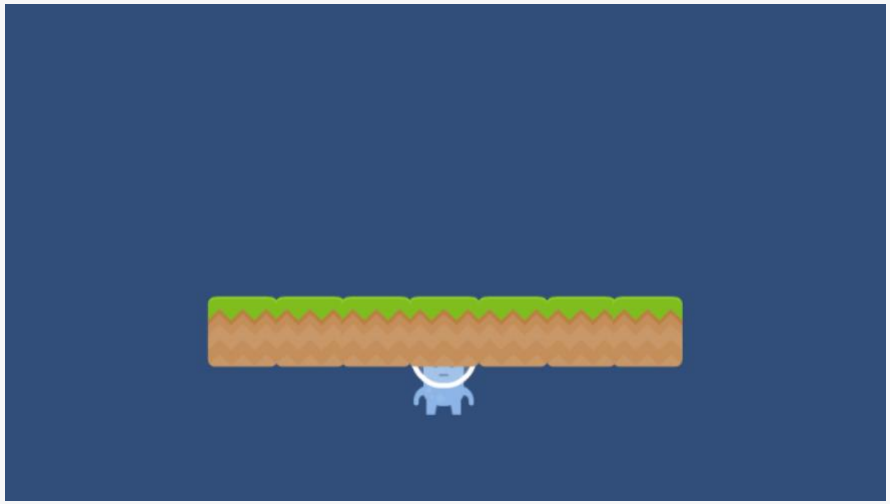
- **Mass**: Mass of the rigid body; the greater the value the greater the force that will be required to move the object.
- **Liner Drag**: This is the amount of friction that a force has to work against to make an object move.
- **Angular Drag**: This is the amount of rotational friction that a force has to work against to make an object rotate.
- **Gravity Scale**: This is the amount of gravity that affects a game object. The greater the value, the stronger the gravitational force.
- **Fixed Angle**: When marked true, the rigid body will continue to respond normally to the physics forces, but without rotating.
- **Is Kinematic**: When marked true, the object will not respond to any physics forces around it. This is typically used when an object needs a special type of physics behavior that can be created by a custom script.

BINUS UNIVERSITY

People
Innovation
Excellence

It falls down

Press play and see what happens



BINUS UNIVERSITY

People
Innovation
Excellence

Colliders 2D

In order to define the dimension of an object in the scene, we need to add a collider component that defines its shape and that reacts if a rigid body is attached to it.

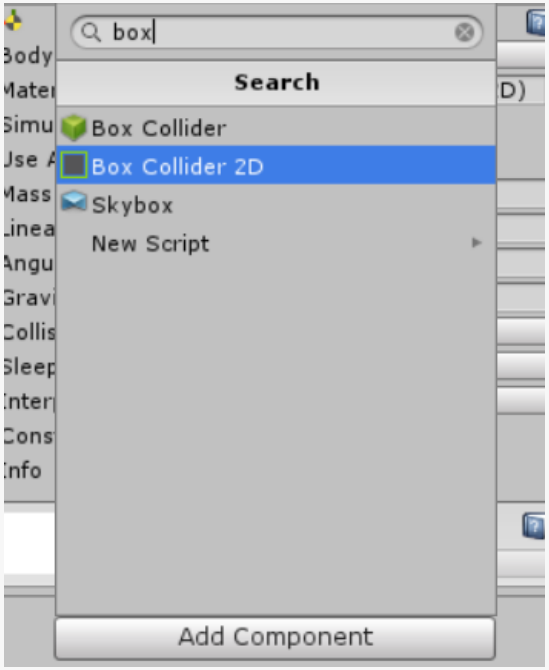
- Box Collider 2D: This type of collider works better with rectangular objects.
- Circle Collider 2D: As given by its name, this collider works better with circular objects.
- Polygon Collider 2D: Its shape is defined by a freeform edge made by line segments that surround the sprite
- Edge Collider 2D: It is used to define a surface without using a series of other colliders

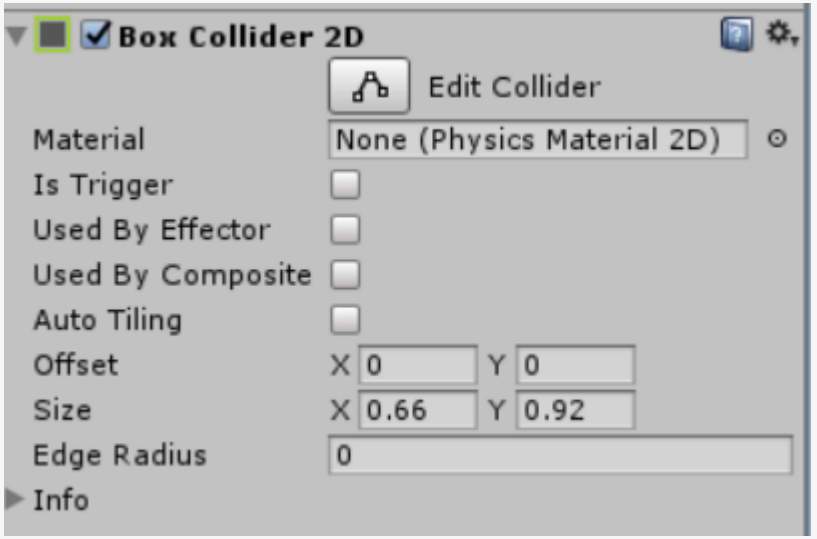
BINUS UNIVERSITY

People
Innovation
Excellence

Box Collider 2D

Let's try to add a box collider to the player & individual floors





You will see above panel in the inspector

BINUS UNIVERSITY

People
Innovation
Excellence

BoxCollider 2D

A type of collider that works better with rectangular objects

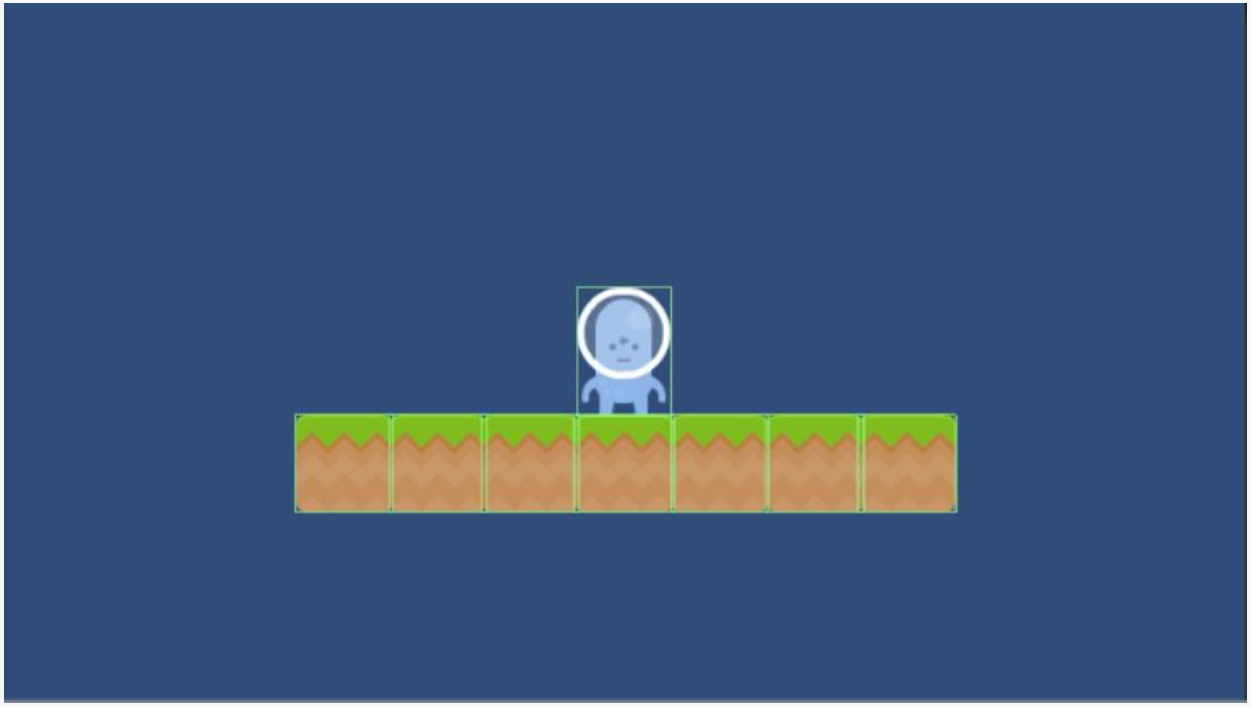
- **Edit Collider**: This is a button that allows us to manually adjust the shape of the collider. This can also be done in the **Scene** view.
- **Material**: This is a reference to the 2D physics material that will define the object's behaviour when colliding with other objects.
- **Is Trigger**: When checked, the collider will act as a trigger to fire events from the code.
- **Used By Effector**: If marked true, this collider will be used by an attached effector to define it. We will see them later in the chapter.
- **Offset**: This allows us to change the centre or pivot point of the collider.
- **Size**: This allows us to change the size of the collider for each axis.

BINUS UNIVERSITY

People
Innovation
Excellence

It stays

Press play and see what happens

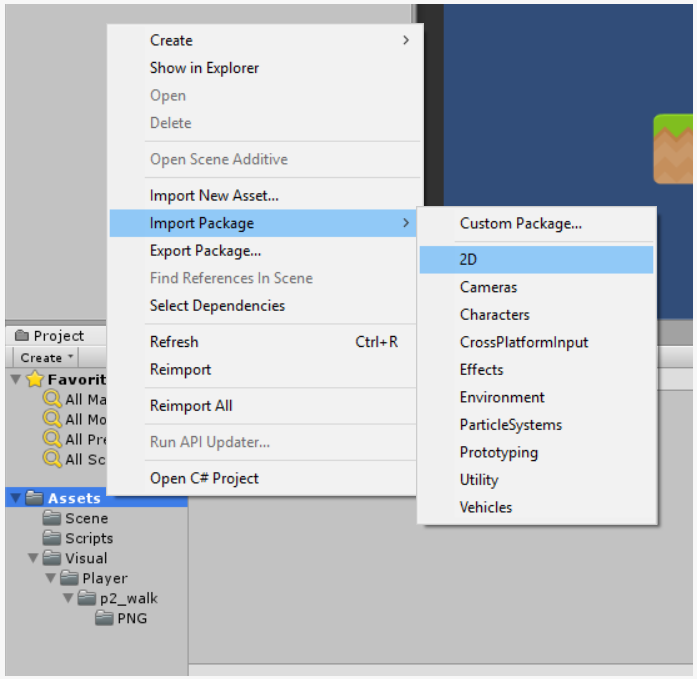


BINUS UNIVERSITY

People
Innovation
Excellence

So...

We are going to use one of Unity Standard Assets



Since we are going to use the Character Controller inside the Standard Assets, we need to have it in our project. If you haven't downloaded it yet, you can do so by going in the Asset Store: Window | Asset Store or alternatively, Ctrl + 9.

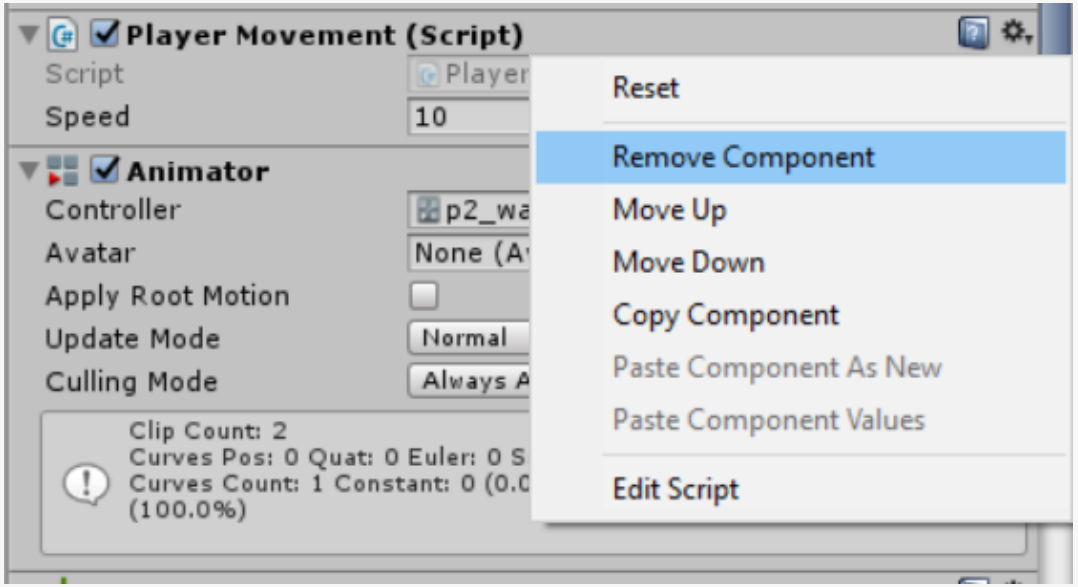
If you have already downloaded them, but not imported them into this project, you can do so by by clicking on Assets | Import Package or again, using the Asset Store.

If you are doing this in the lab class than you should have 2DUnityStandardAssets.unity in your class FTP folder

BINUS UNIVERSITY

People
Innovation
Excellence

Remove your old script

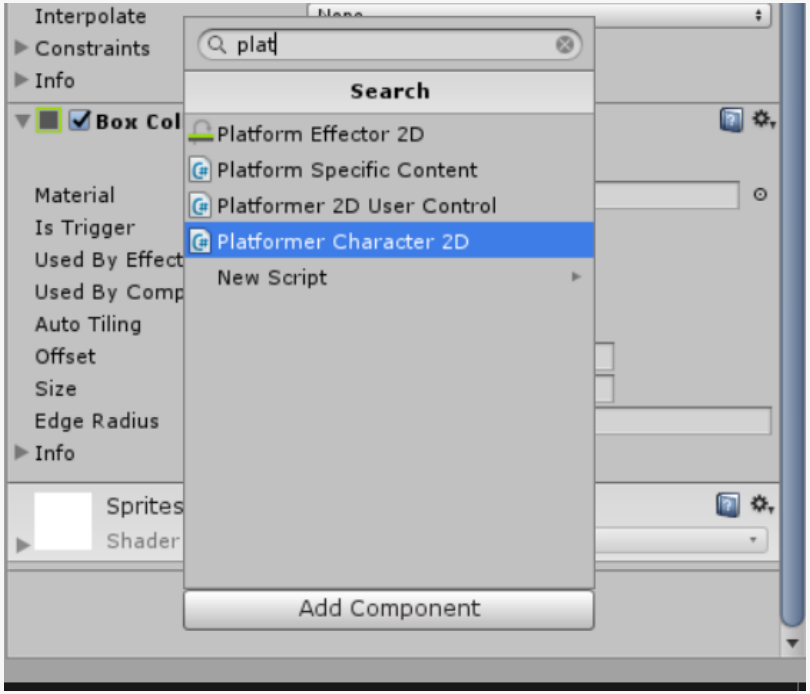


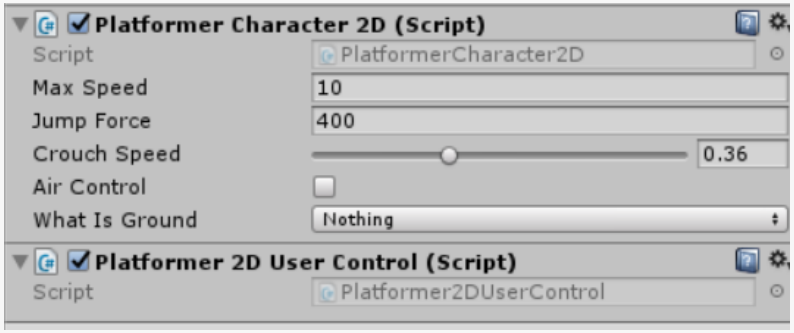
BINUS UNIVERSITY

People
Innovation
Excellence

Add the unity standard asset script

Add the new PlatformerCharacter 2D and Platform 2D User Control Scripts to the player object





You will see above panel in the inspector

BINUS UNIVERSITY

People
Innovation
Excellence

Platform Character 2D

The PlatformerCharacter2D component makes use of the rigid body attached to the object to move the character in a Platformer fashion. It still allows us to use the other physical forces in the scene, making the character even more believable.

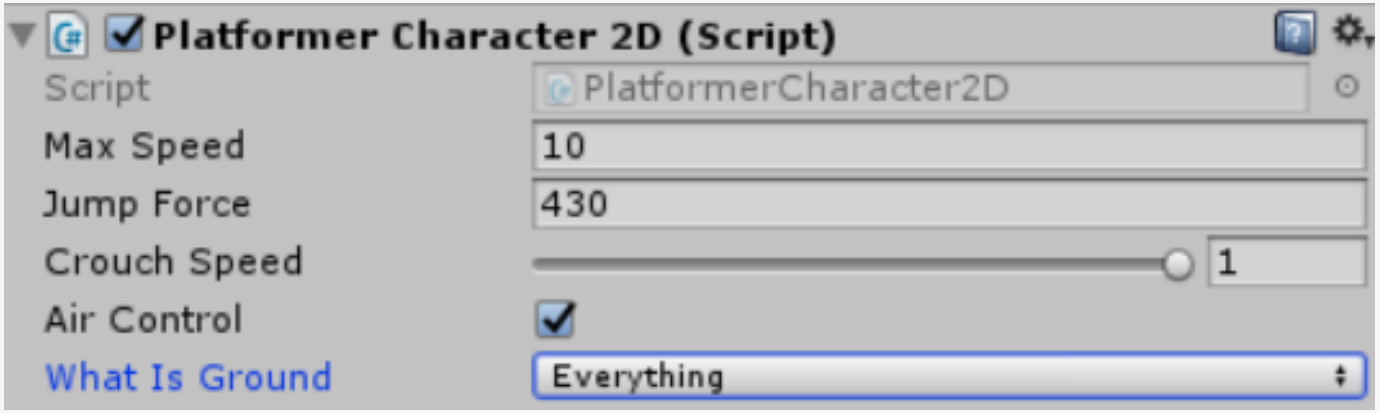
- **Max Speed**: This is the maximum speed that the player can reach when moving along the *X* axis
- **Jump Force**: This is the force that will be applied to the player's rigid body when he jumps
- **Crouch Speed**: This is the player's speed while crouching (crouching is done by pressing *Ctrl*)
- **Air Control**: If marked true, the player can also be controlled in air, when he is not being grounded
- **What Is Ground**: These are the layers that will be treated as the ground.

BINUS UNIVERSITY

People
Innovation
Excellence

Do some changes

Let's edit the parameter

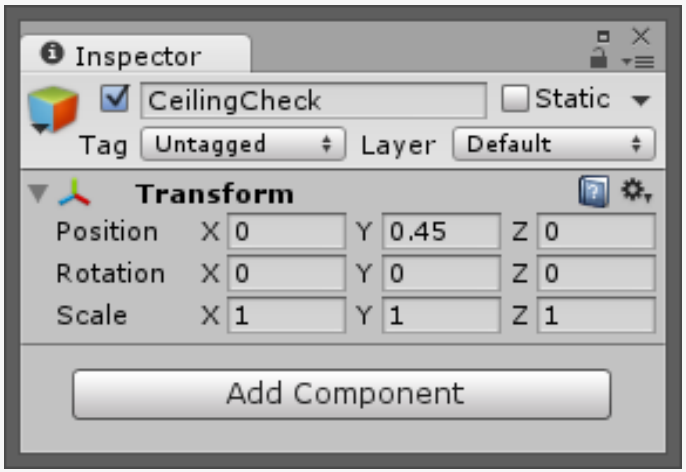


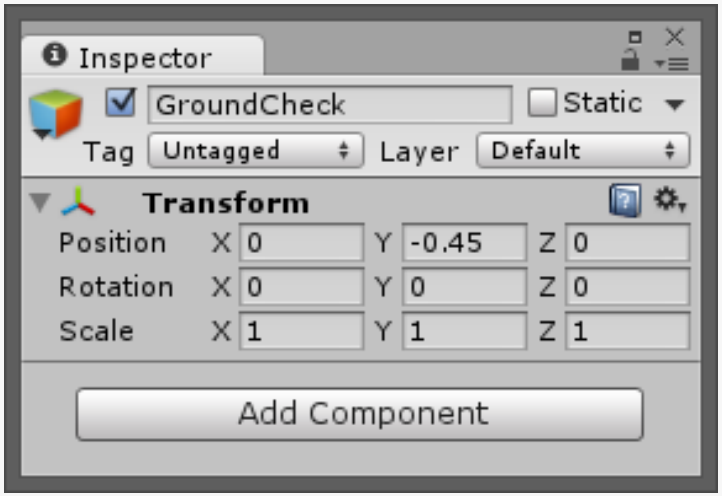
BINUS UNIVERSITY

People
Innovation
Excellence

Add detection

Now we need to define which are the top and bottom ends of the Player object. This is because they are used by the PlatformerCharacter2D component to understand the dimensions of the character. We can easily do this by adding two empty child game objects to the Player and adjust their positions

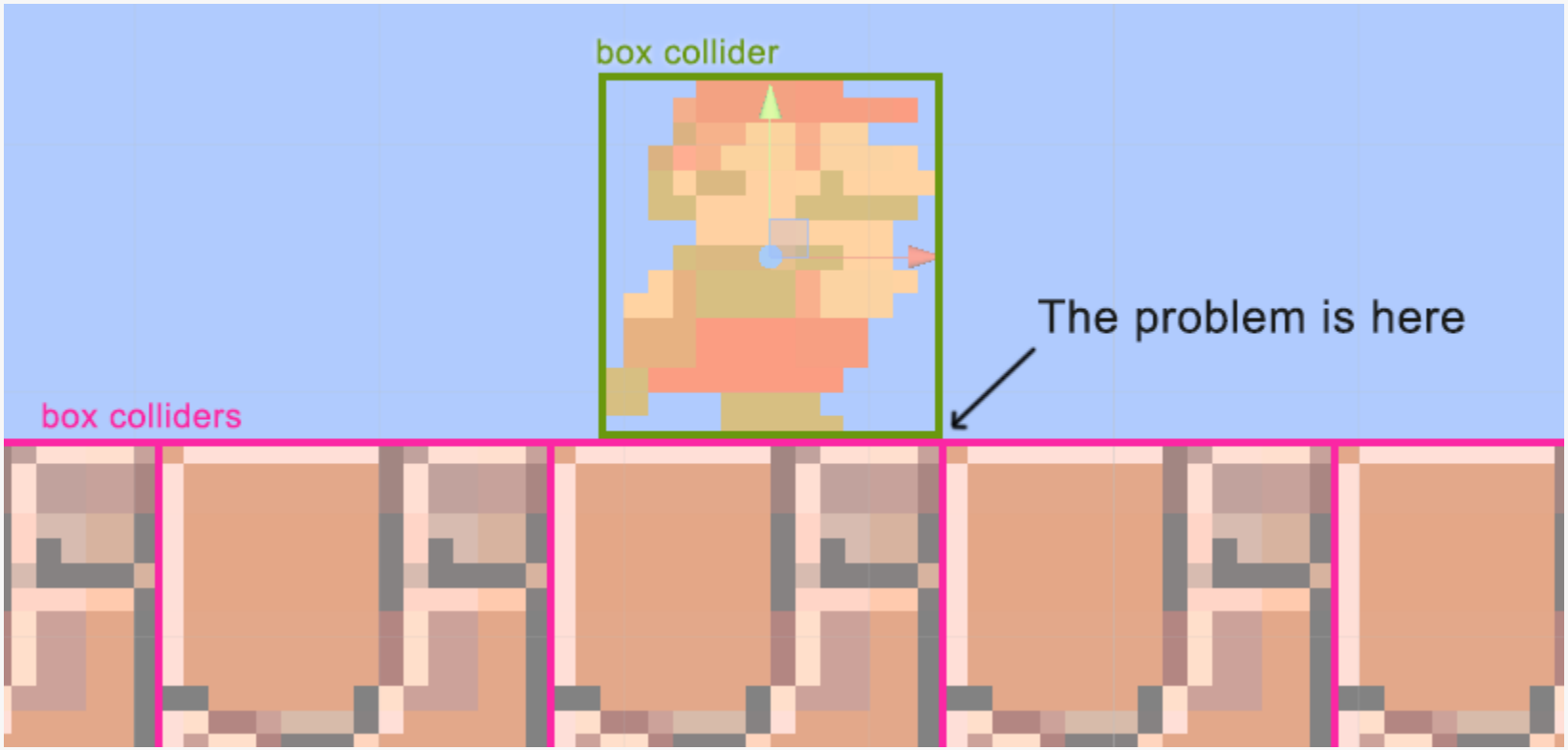




BINUS UNIVERSITY

People
Innovation
Excellence

The problem with box colliders

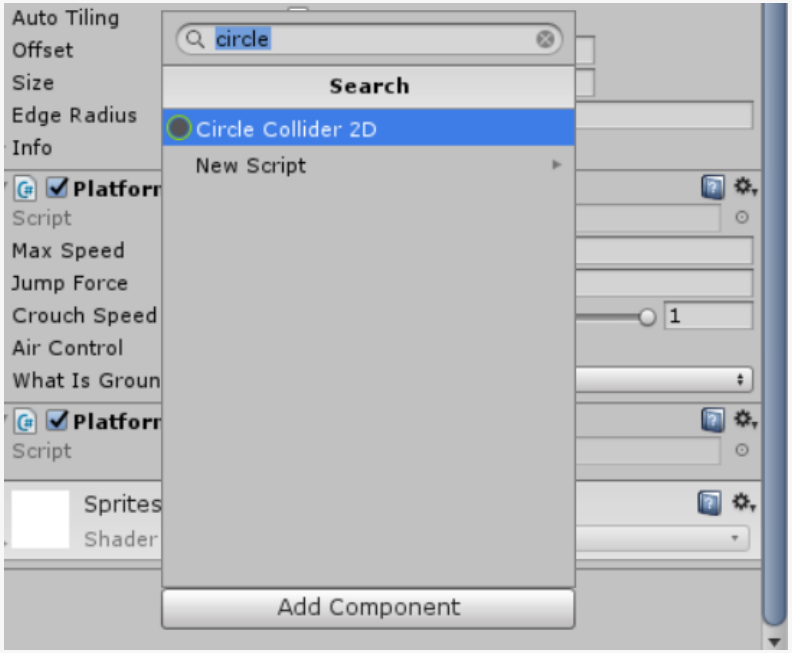


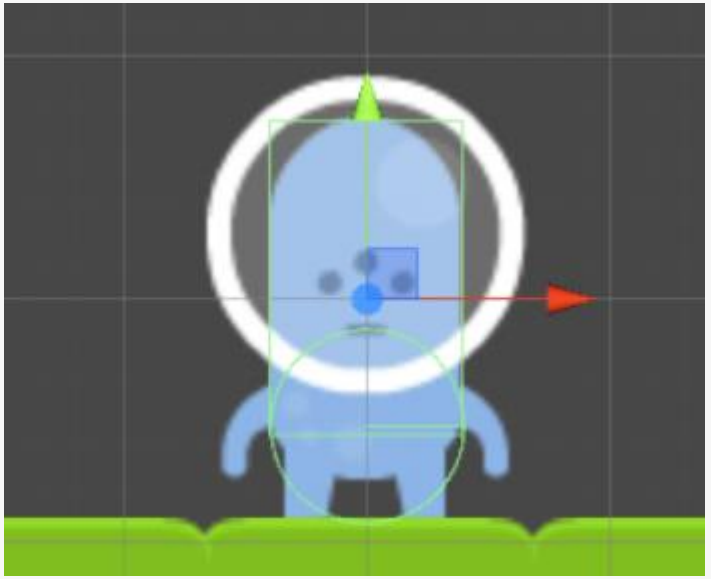
BINUS UNIVERSITY

People
Innovation
Excellence

How to fix it?

Add circle collider!





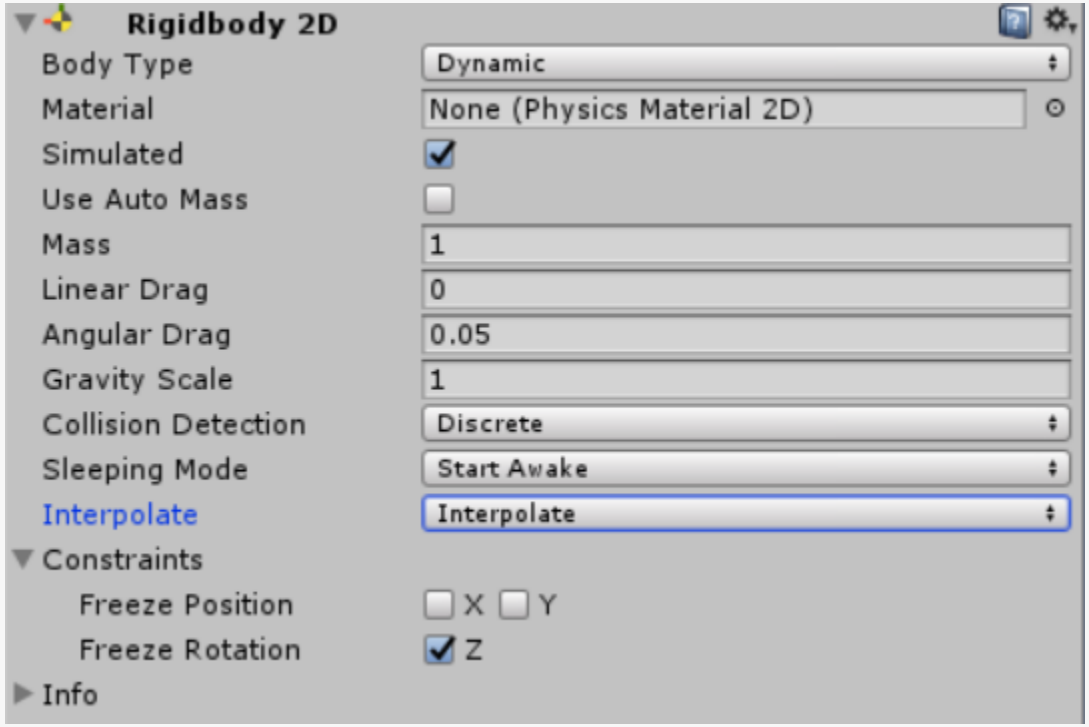
Edit the collider to look like this
(Check the green lines)

BINUS UNIVERSITY

People
Innovation
Excellence

Edit rigid body setting

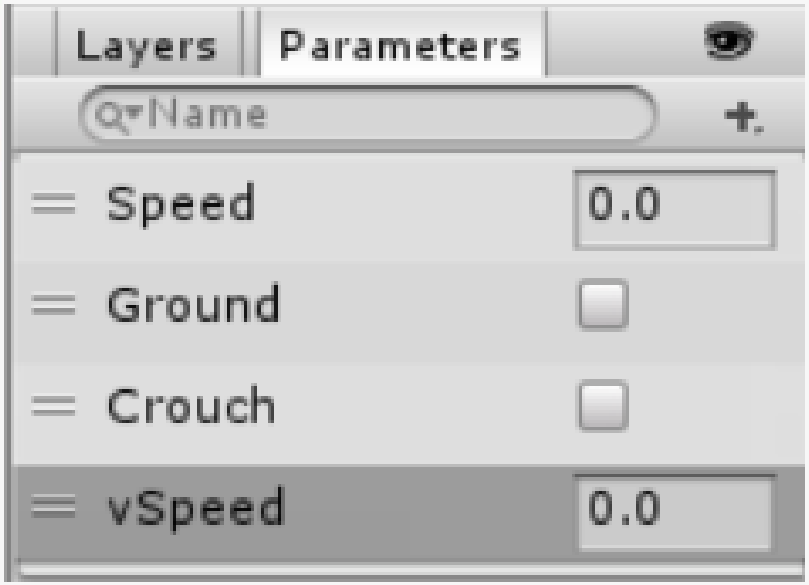
Finally, in the RigidBody2D component, we need to change Constraints > Freeze Rotation > Z to true to True. As a result, our character will not rotate. Also, in the Interpolate variable, choose the Interpolate method in order to smooth the character movement.



BINUS UNIVERSITY

People
Innovation
Excellence

Get back to your animator

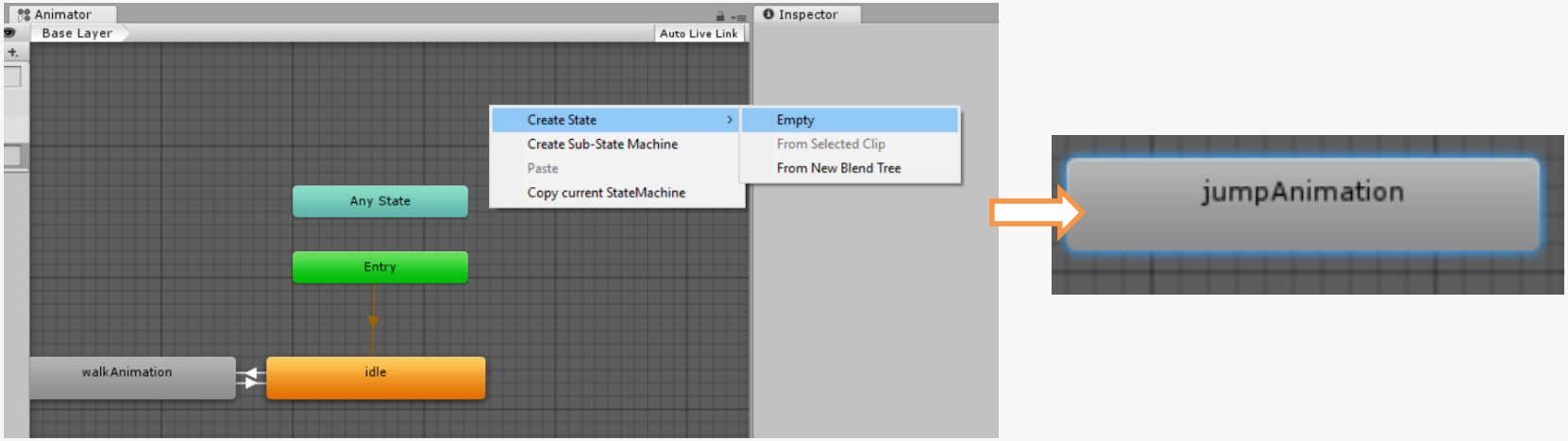


1. Rename the parameter PlayerSpeed to Speed. This is the speed of the player.
2. Create two Boolean parameters and name it Ground and Crouch. This is an indicator whether the player is in the ground or crouching.
3. Add a new parameter of type float and name it vSpeed. This is the vertical speed.

BINUS UNIVERSITY

People
Innovation
Excellence

Create a new state

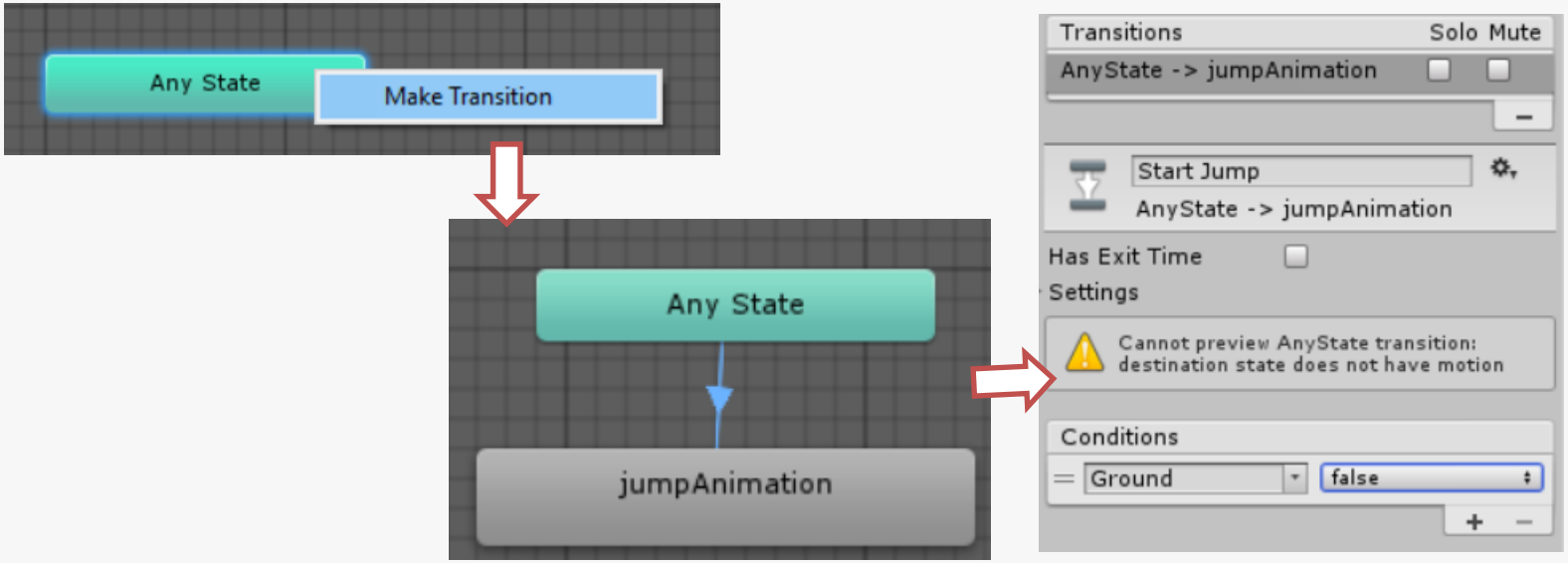


Create a new empty state and name it “jumpAnimation”

BINUS UNIVERSITY

People
Innovation
Excellence

Create a new transition

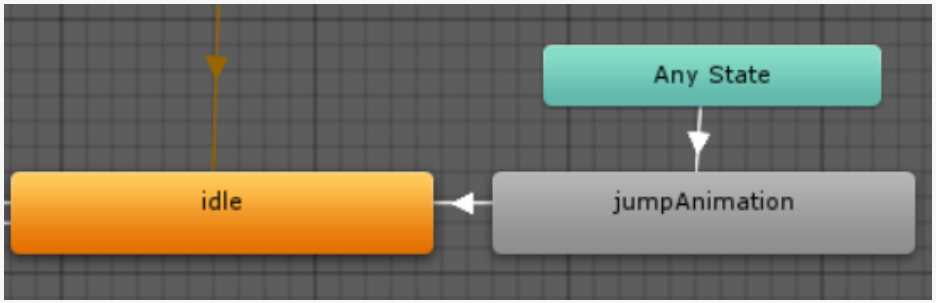


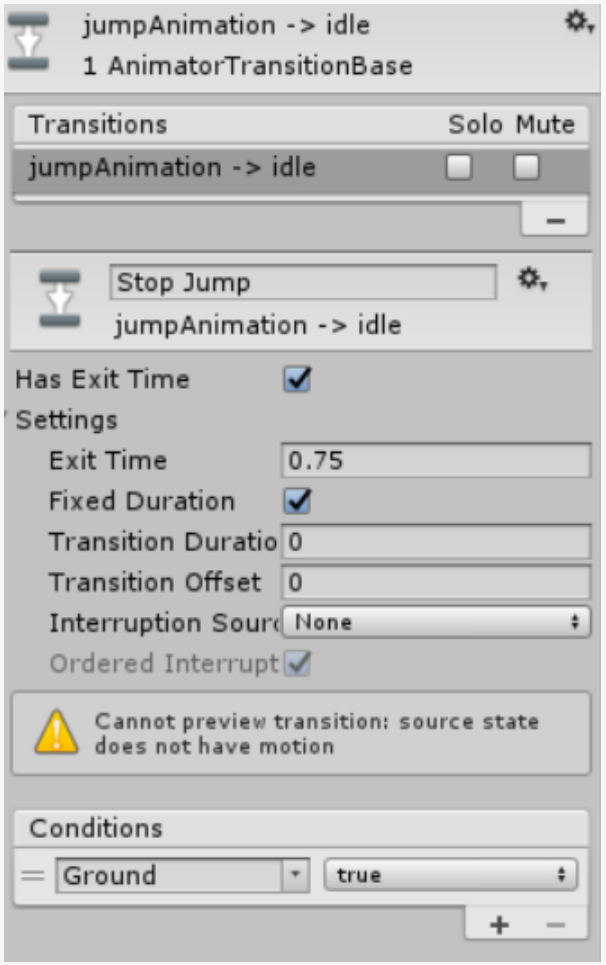
Create transition from AnyState to jumpAnimation. Name it Start Jump and create a Conditions where Ground is false

BINUS UNIVERSITY

People
Innovation
Excellence

Another transition



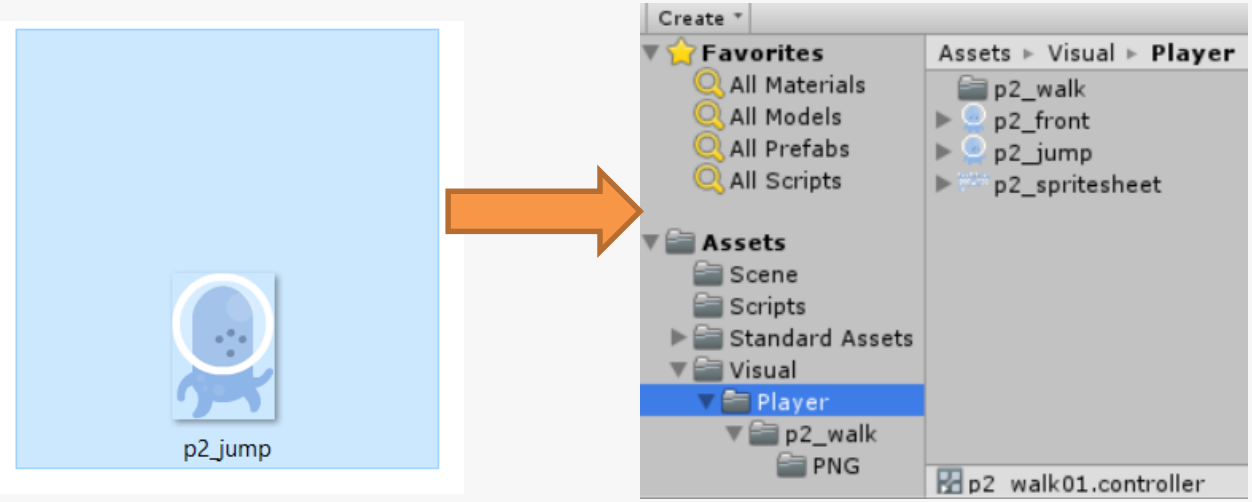


Create transition from jumpAnimation to idle. Name it Stop Jump and create a Conditions where Ground is true and set Transition Duration to 0.

BINUS UNIVERSITY

People
Innovation
Excellence

The sprite



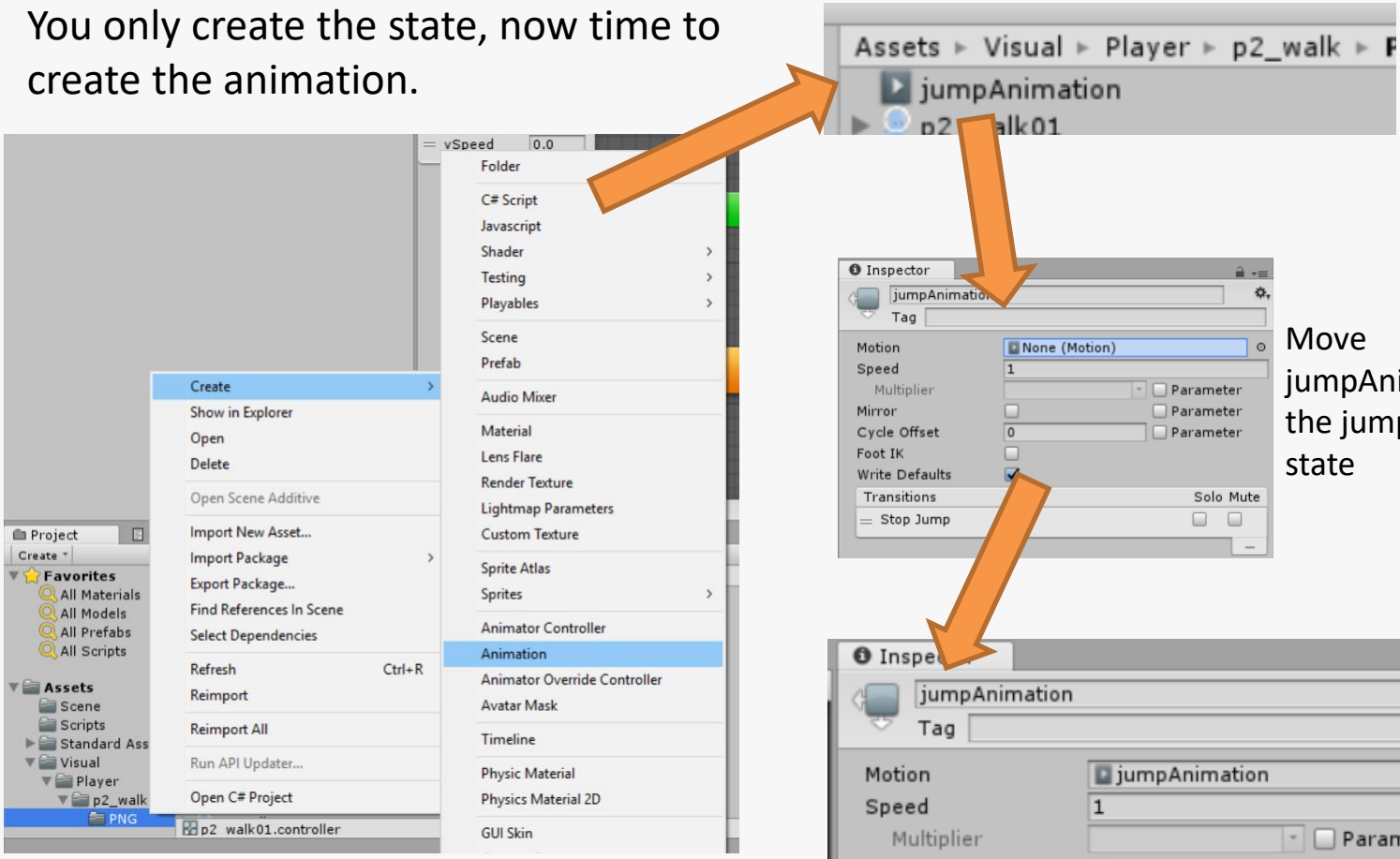
Grab jump sprite from the kenney's asset folder and put it on our project folder.

BINUS UNIVERSITY

People
Innovation
Excellence

Create the jumpAnimation

You only create the state, now time to create the animation.



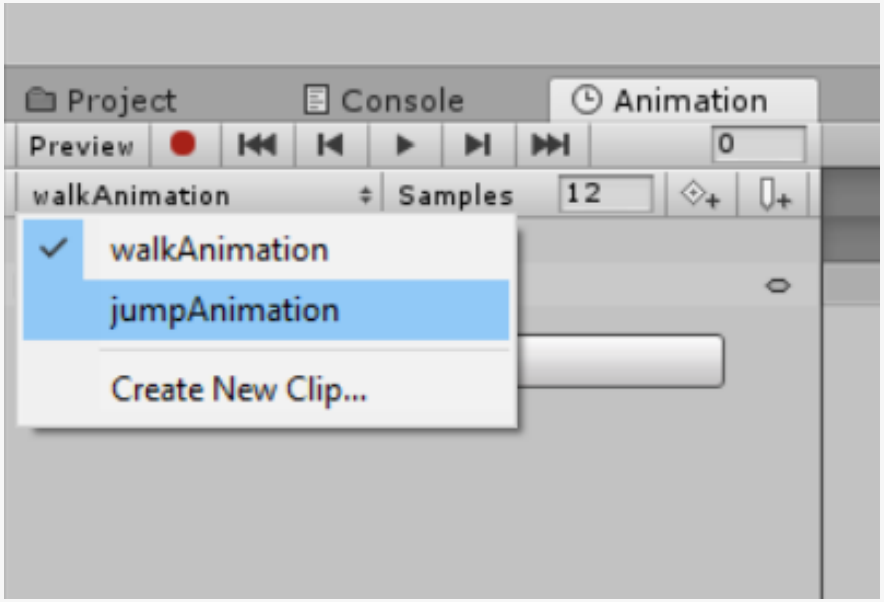
Move jumpAnimation to the jumpAnimation state

BINUS UNIVERSITY

People
Innovation
Excellence

Open the animator

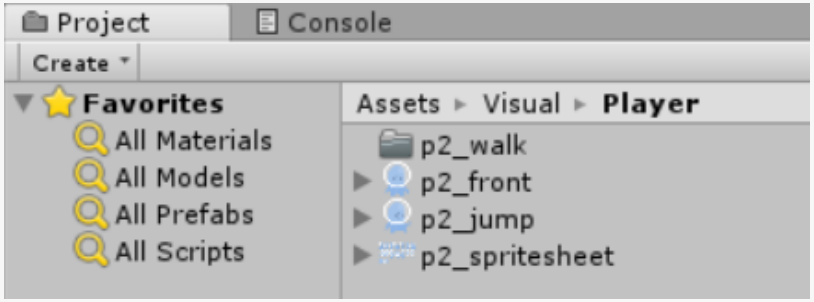
Pick the jump animation

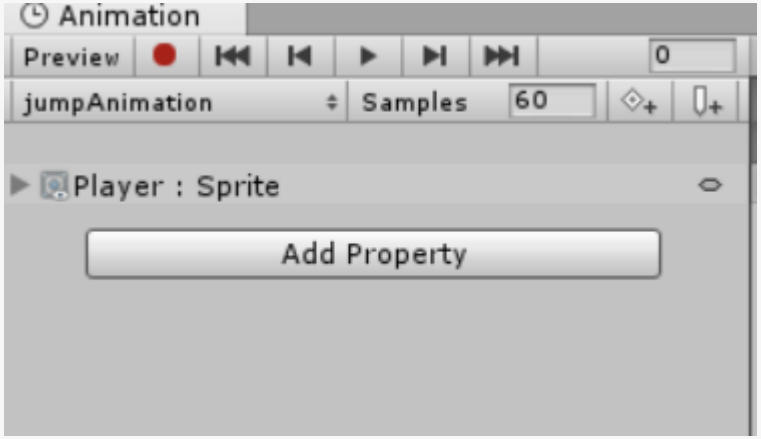


BINUS UNIVERSITY

People
Innovation
Excellence

Create the simple animation



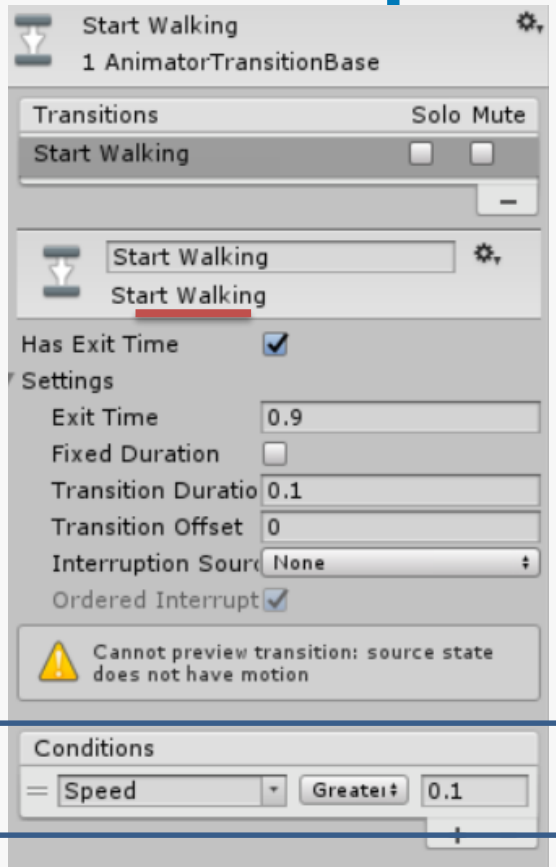


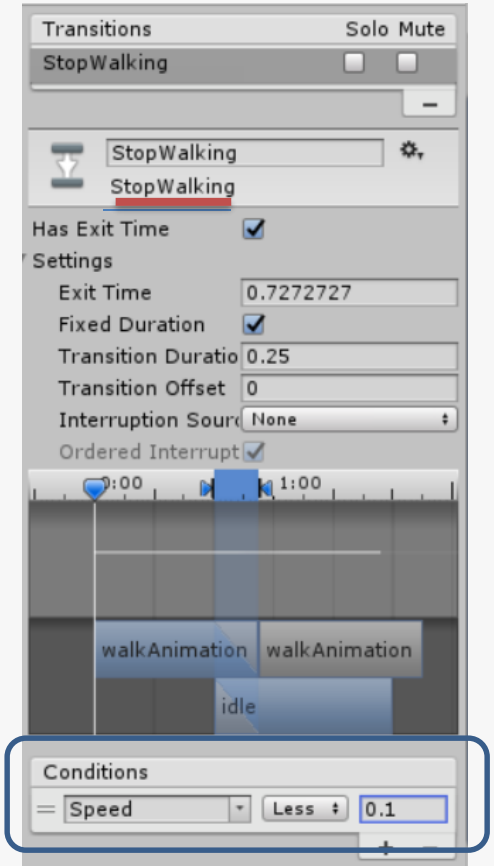
Drag p2_jump to the jumpAnimation Animation

BINUS UNIVERSITY

People
Innovation
Excellence

Go back to the animator and update other parameters

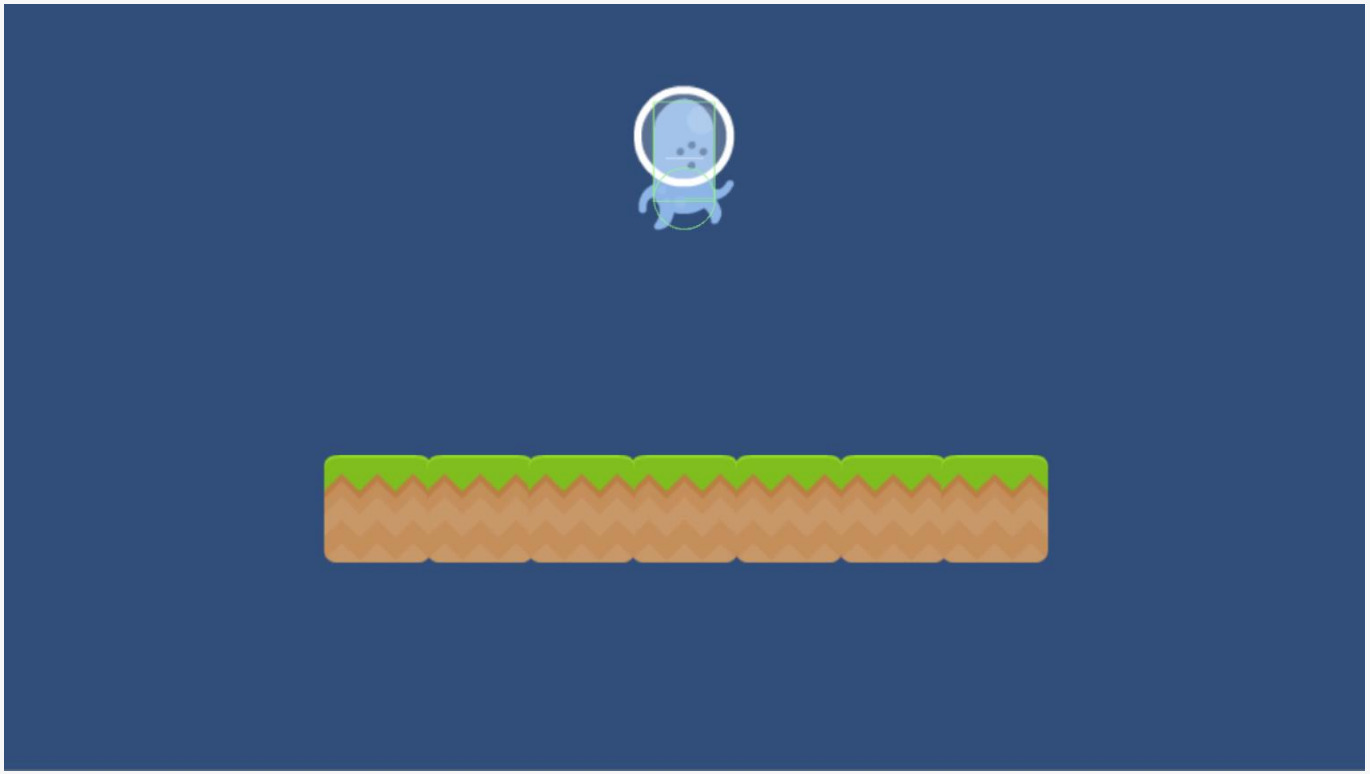




BINUS UNIVERSITY

People
Innovation
Excellence

Press play and tests it out



BINUS UNIVERSITY

People
Innovation
Excellence

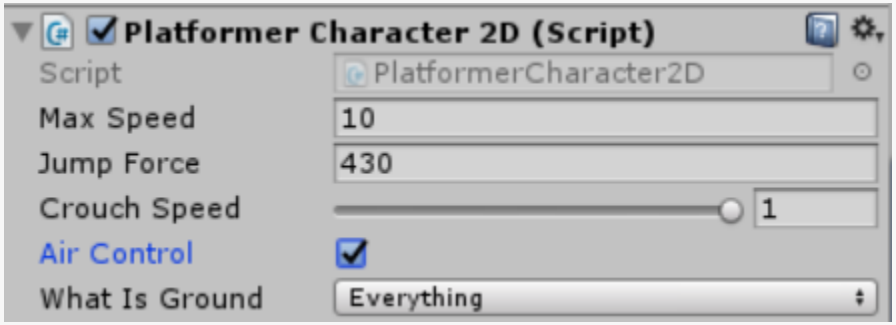
Following camera?

Put the main camera as the sub-game object of the player and again...
Press play!!

BINUS UNIVERSITY

People
Innovation
Excellence

The Platformer 2D User Control



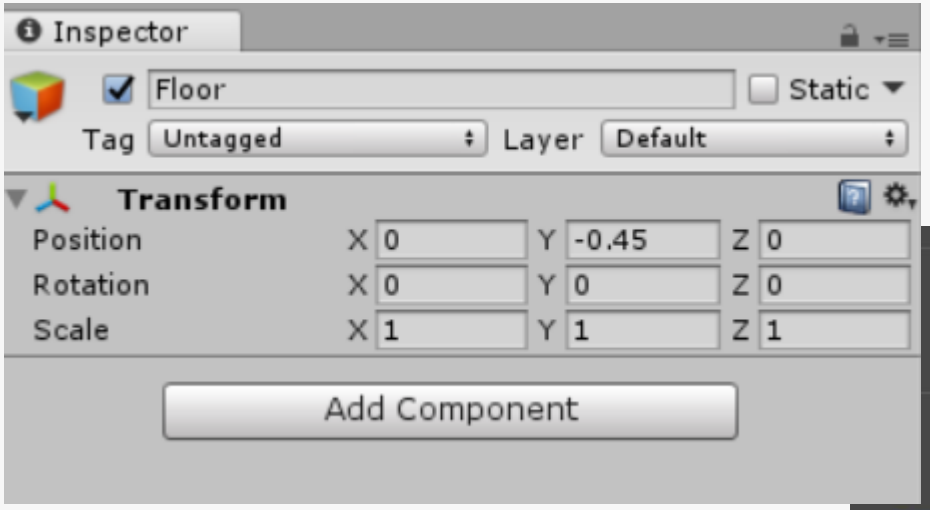
- Max speed -> *The fastest the player can travel in the x axis.*
- Jump Force -> *Amount of force added when the player jumps.*
- Crouch Speed -> *Amount of maxSpeed applied to crouching movement. 1 = 100%*
- Air Control -> *Whether or not a player can steer while jumping*
- What is Ground? -> *A mask determining what is ground to the character*

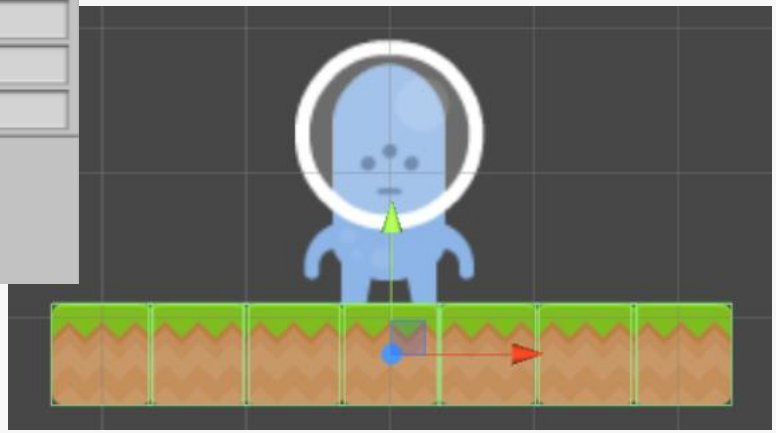
BINUS UNIVERSITY

People
Innovation
Excellence

Scale your platform

Rescale your platform to 1,1,1 and move it directly under your player.

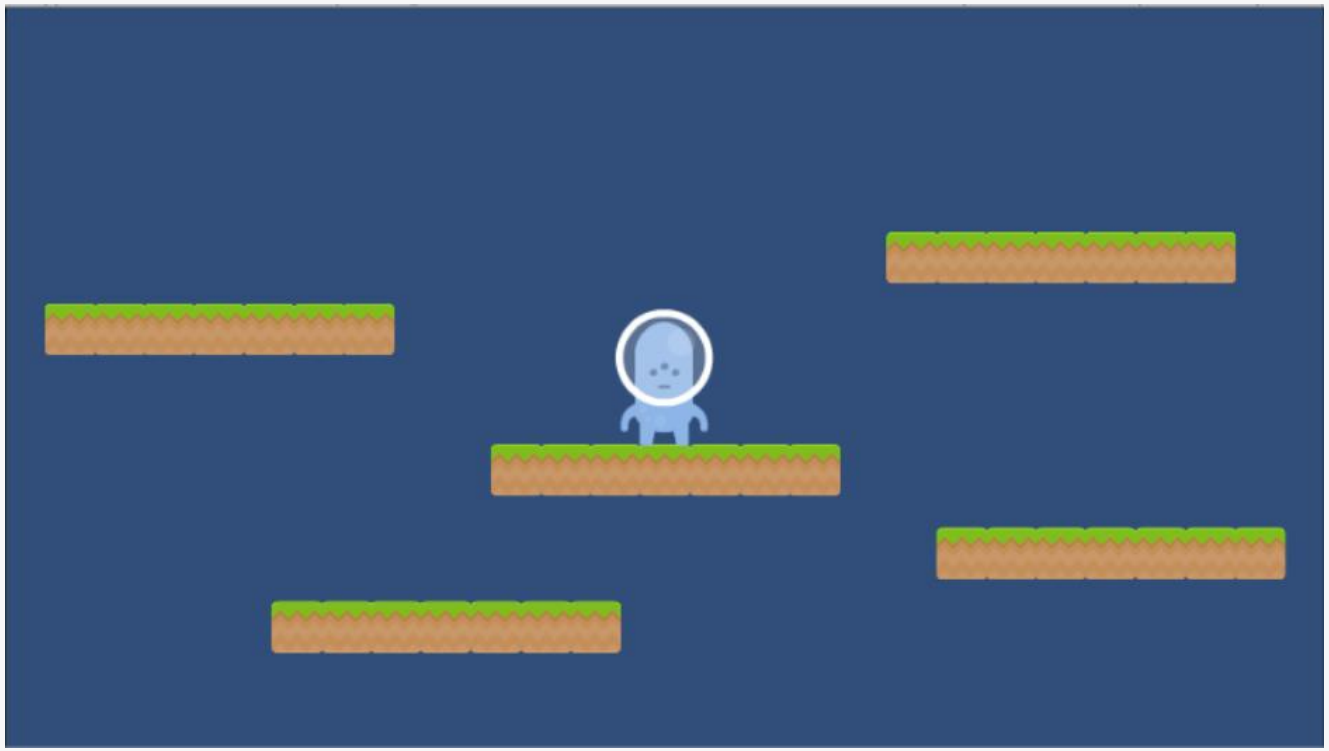




BINUS UNIVERSITY

People
Innovation
Excellence

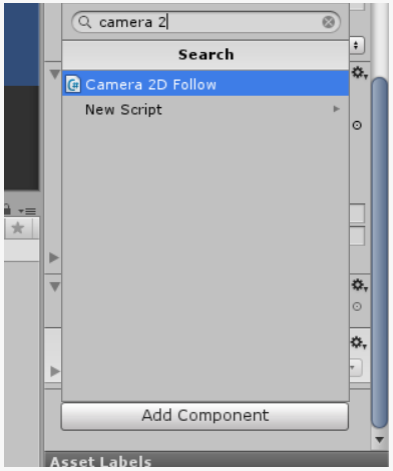
Duplicate and move it as you wanted



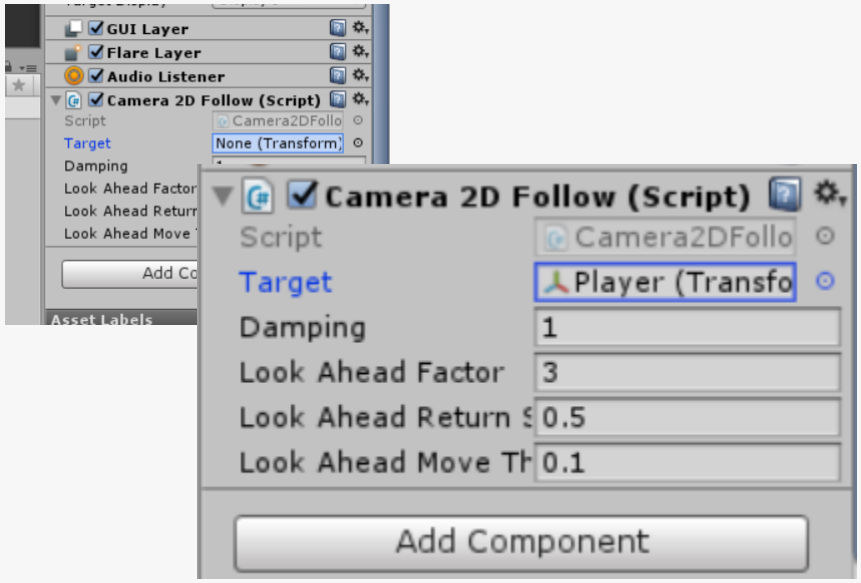
BINUS UNIVERSITY

People
Innovation
Excellence

Camera Follow



Add Camera 2D Follow to the Main Camera Script

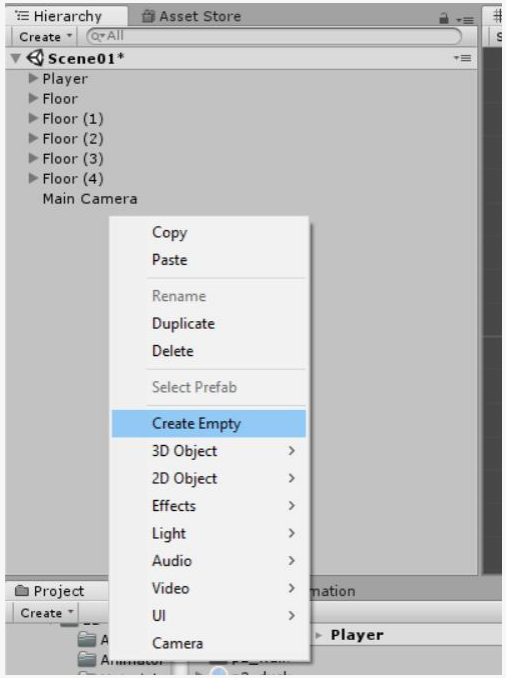


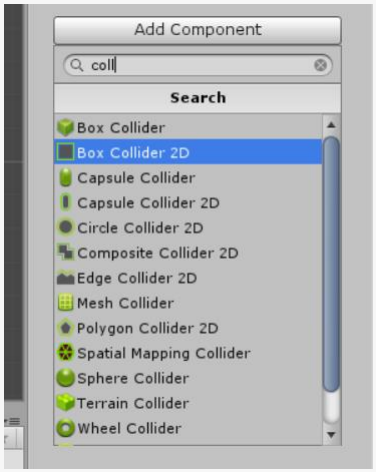
Drag player game object to the target field

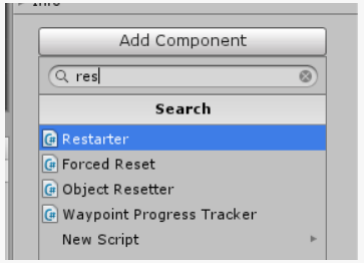
BINUS UNIVERSITY

People
Innovation
Excellence

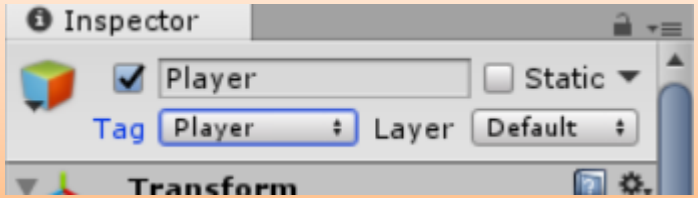
Create a reset zone







Create an empty gameobject and name it Reset Zone. Add component Box Collider 2D and Restarter script

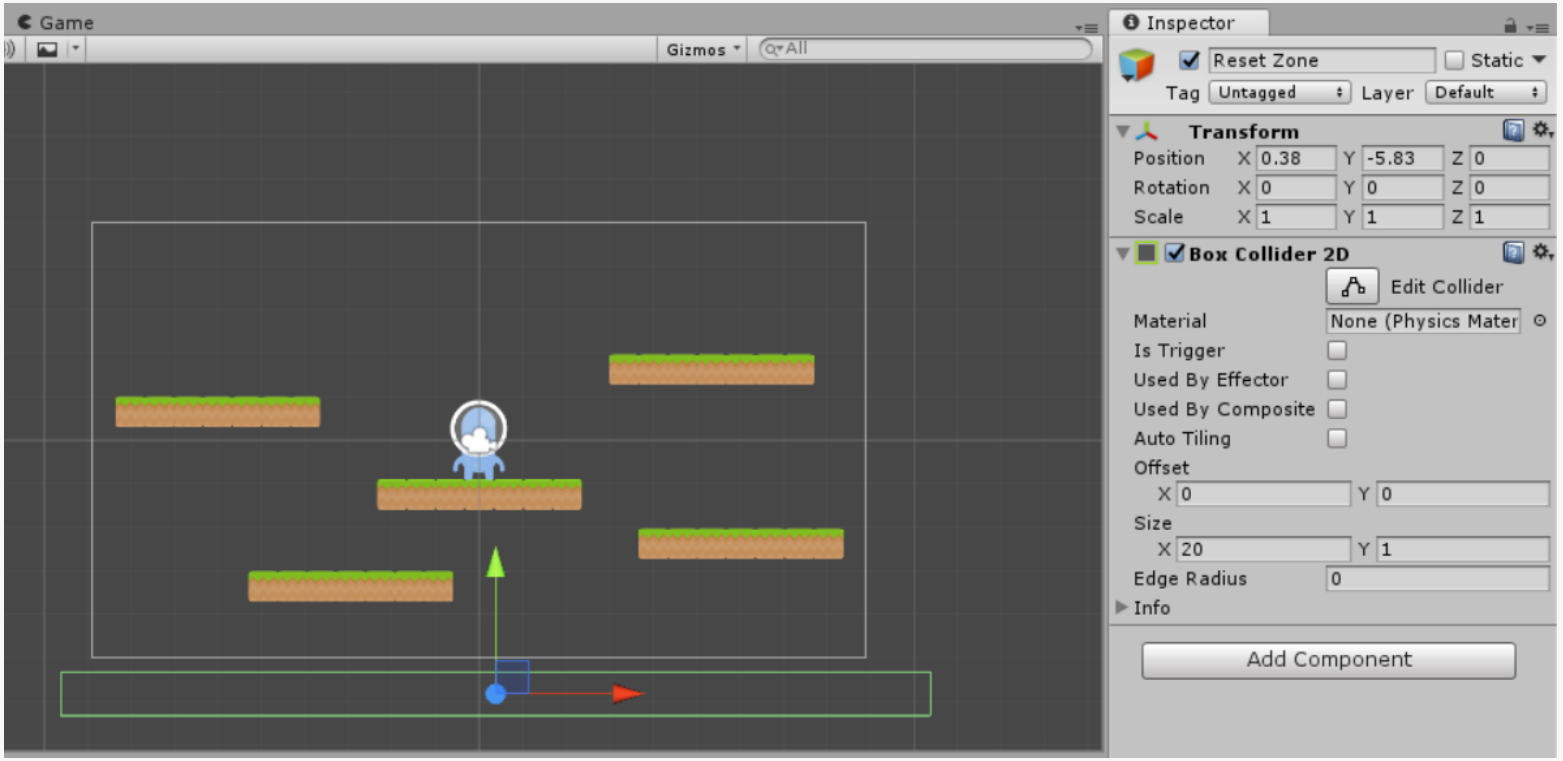


Make sure that you tag your Player object as Player

BINUS UNIVERSITY

People
Innovation
Excellence

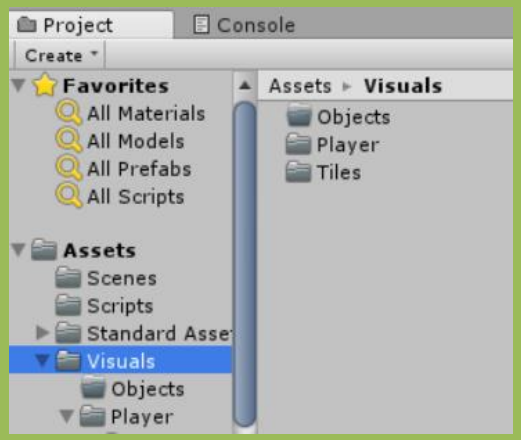
Move it to under the platforms



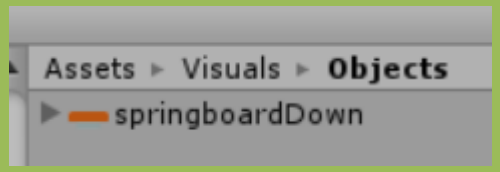
BINUS UNIVERSITY

People
Innovation
Excellence

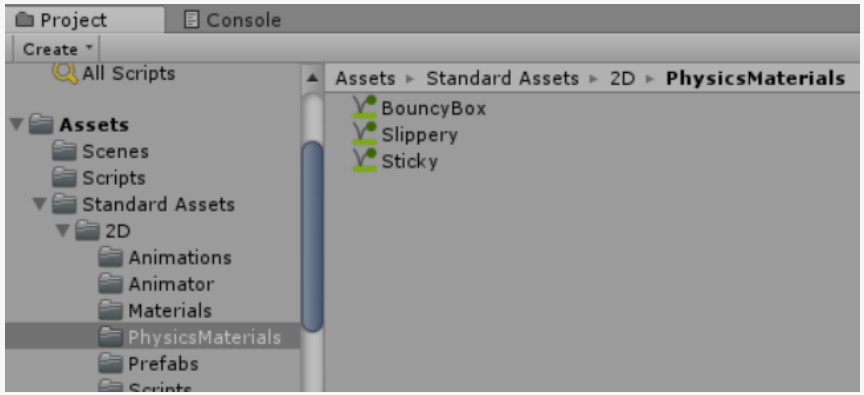
Boing



Make objects folder in the Visuals project folder and Move springboardDown from kenney's asset to the project folder



Move the springboardDown to the game scene and add box collider 2D component

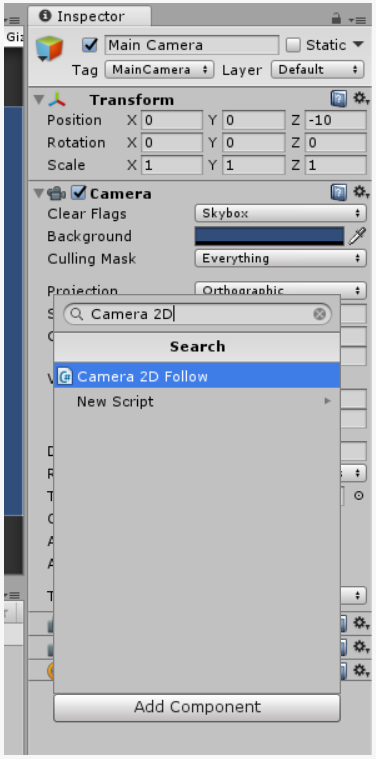


Open BouncBoc physicsmaterials from standard assets and apply it to the game object (drag and drop).

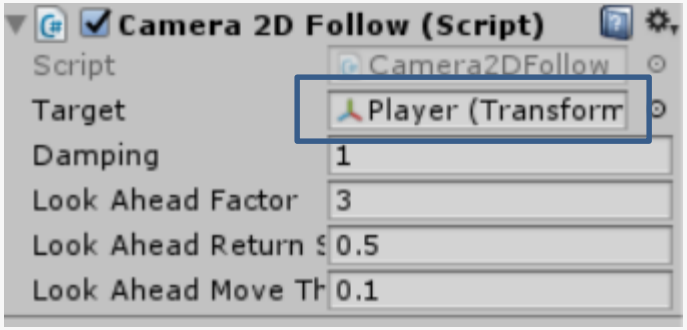
BINUS UNIVERSITY

People
Innovation
Excellence

Make sure the camera follows....



Open “Main Camera” inspector and add Camera 2D Follow Component.

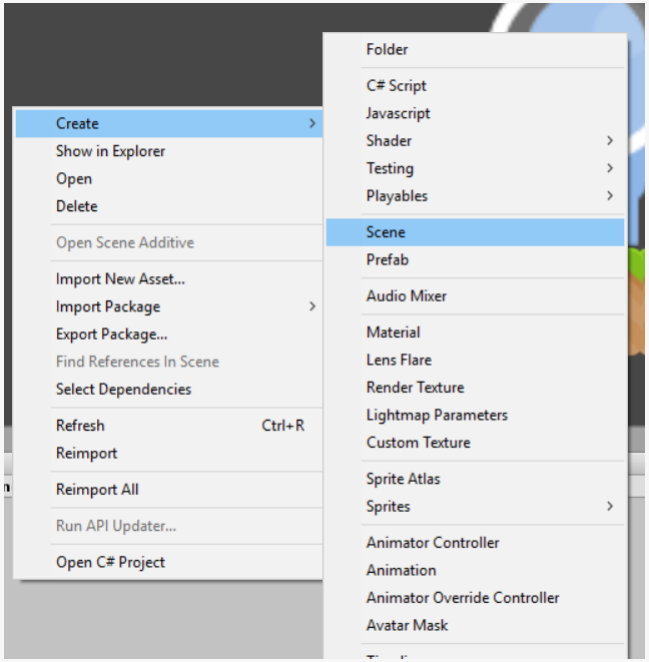


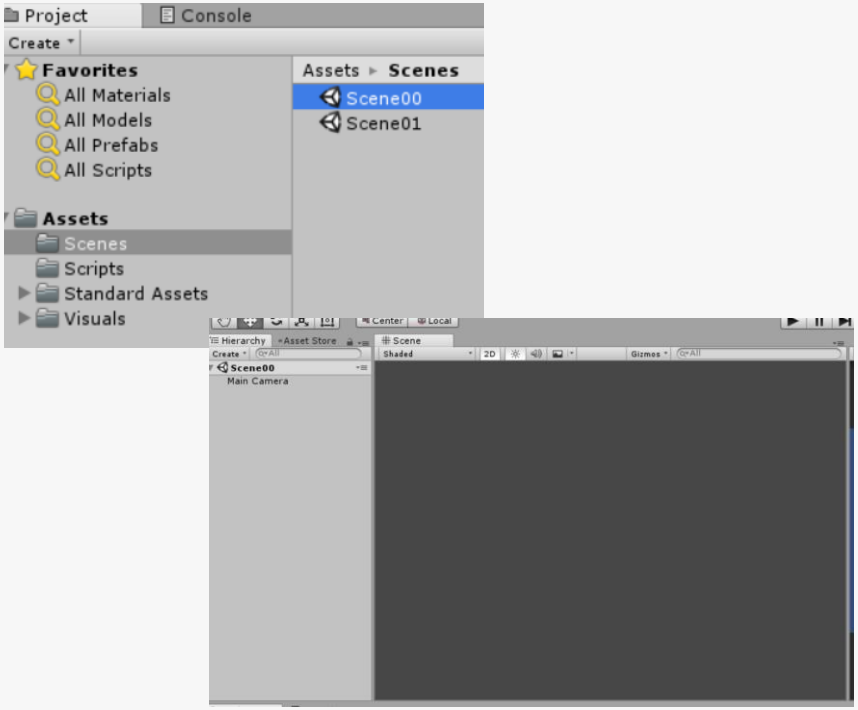
Make sure that the Target is the Player

BINUS UNIVERSITY

People
Innovation
Excellence

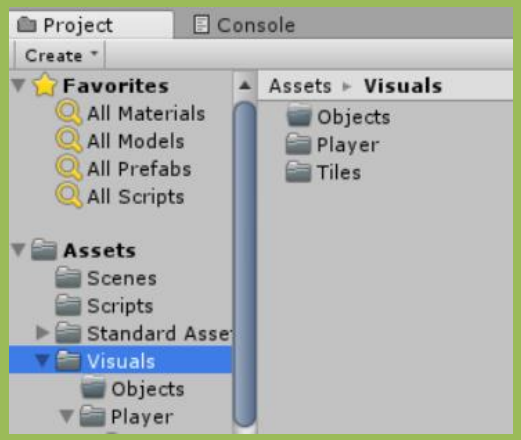
Create a main menu scene



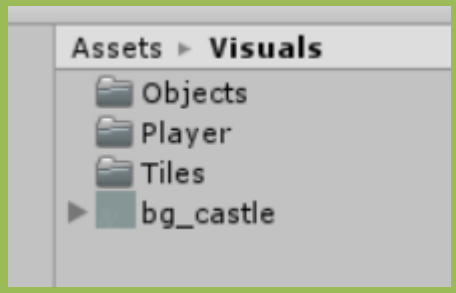


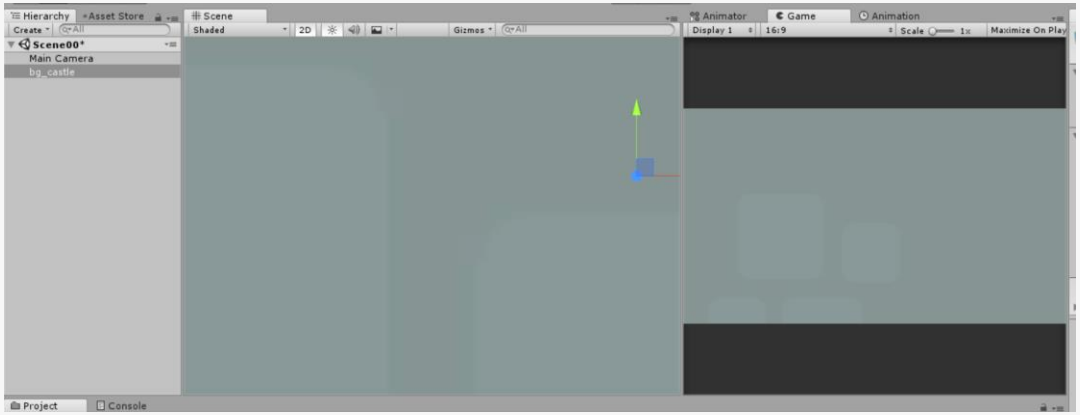
Create a new scene, name it Scene00 and open it!!

Create the Background



Move bg_castle from kenney's assets to the project visual folder



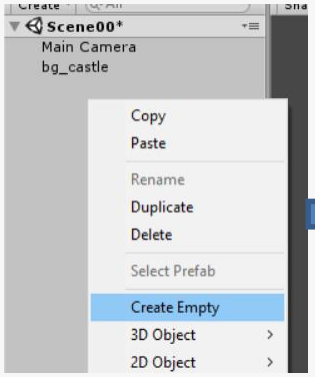


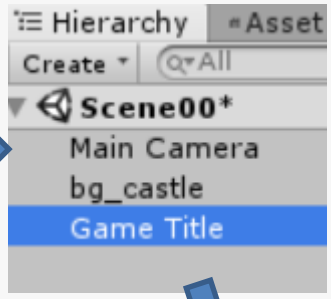
Move the springboardDown to the game scene and resize it (7,7,1)

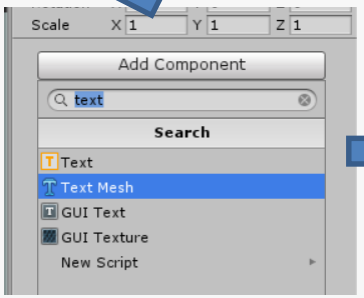
People
Innovation
Excellence

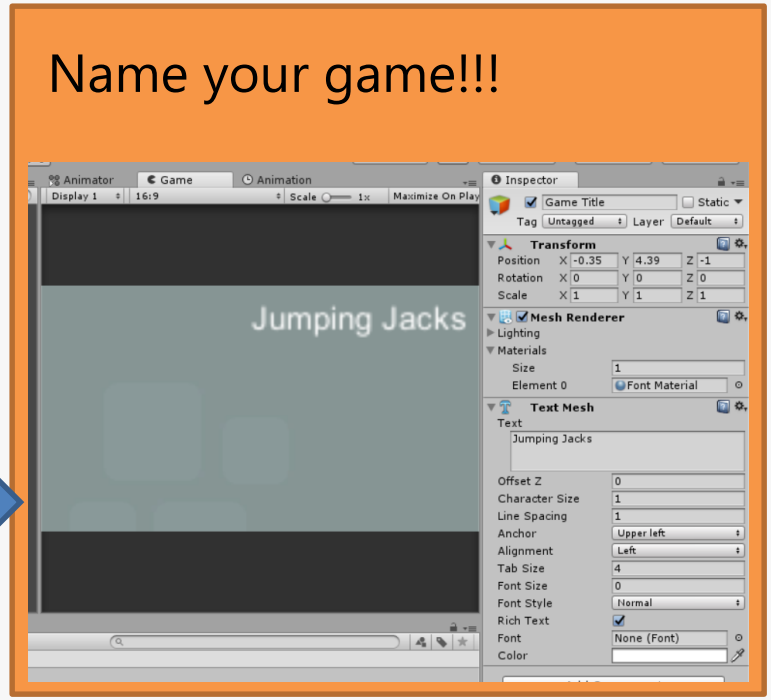
Game Title!

Create new gameobject and name it Game Title and add a text mesh









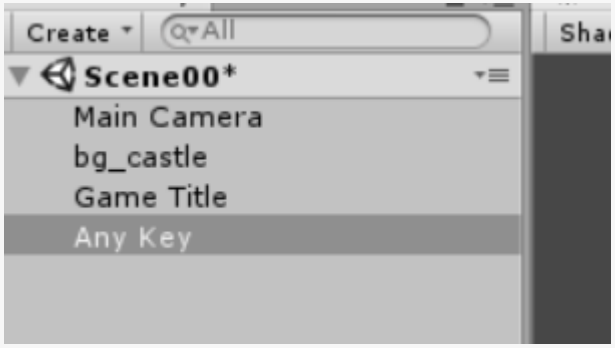
People
Innovation
Excellence

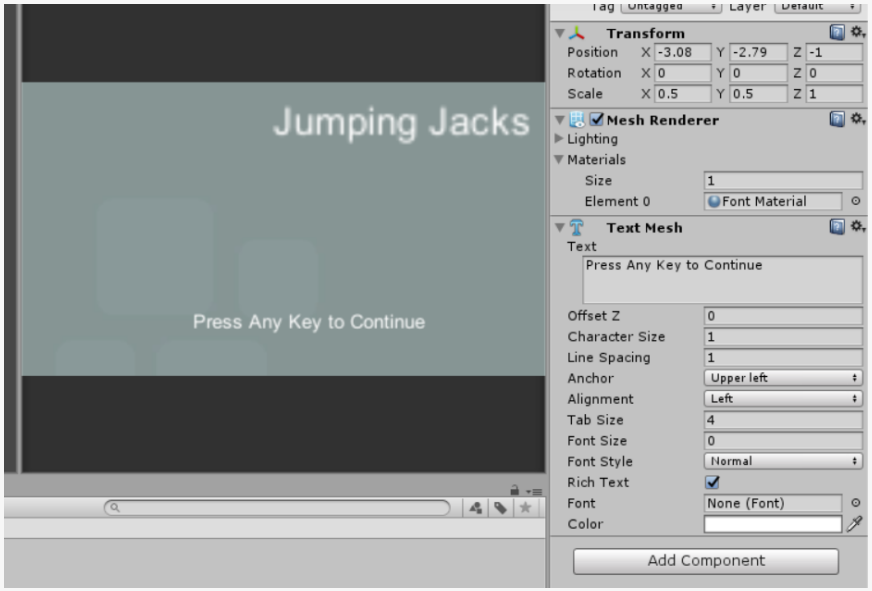
BINUS UNIVERSITY

People
Innovation
Excellence

Any Key Text

Create a game object and name it Any Key



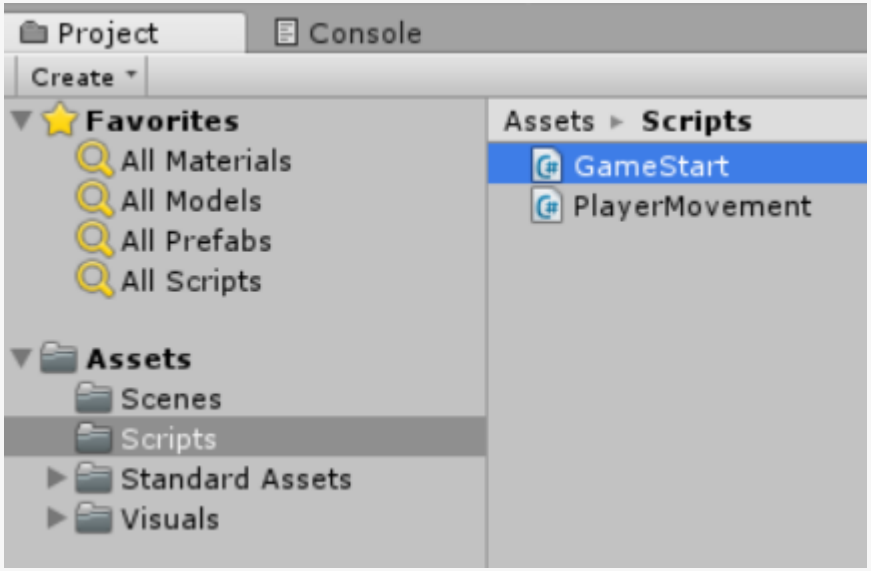


Add text mesh and write “Press Any Key to Continue”

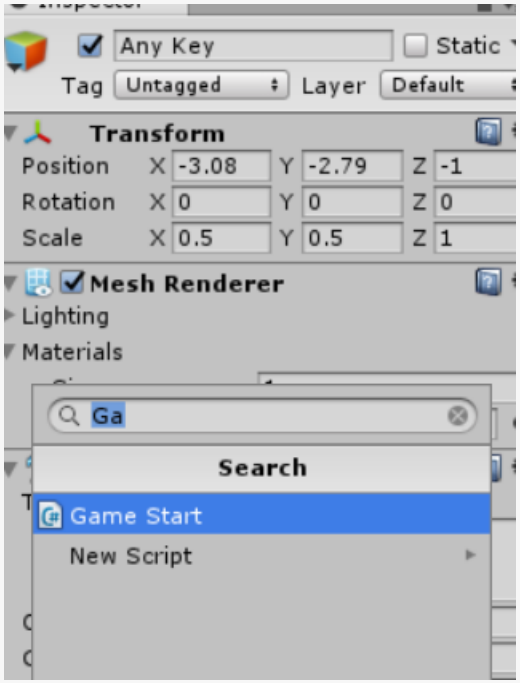
BINUS UNIVERSITY

People
Innovation
Excellence

The main menu script



Create “GameStart” script



Add the script to Any Key gameobject

BINUS UNIVERSITY

People
Innovation
Excellence

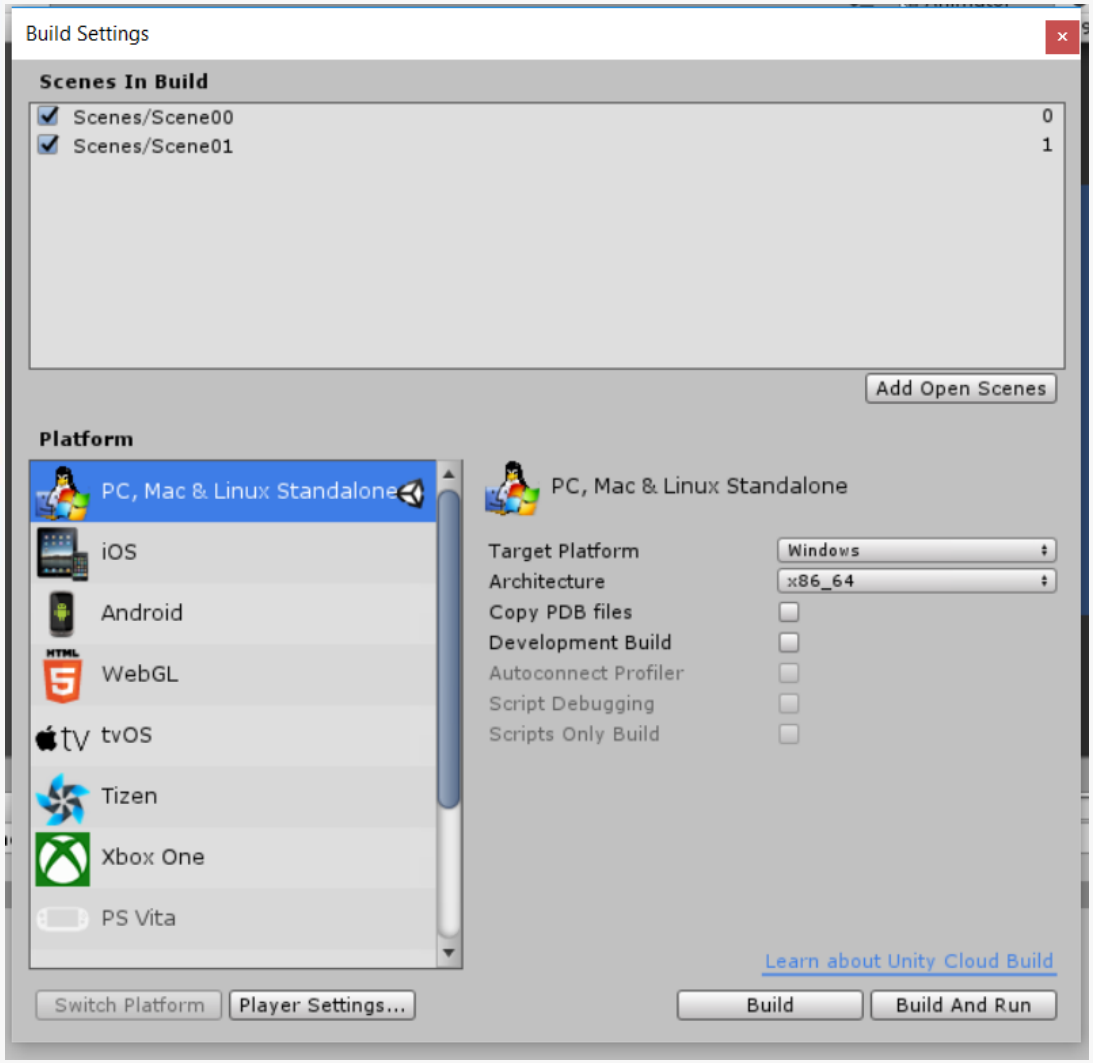
The script

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class GameStart : MonoBehaviour {
7
8     // Use this for initialization
9     void Start () {
10
11    }
12
13    // Update is called once per frame
14    void Update () {
15        if (Input.anyKey)
16            SceneManager.LoadScene("Scene01");
17    }
18 }
```

BINUS UNIVERSITY

People
Innovation
Excellence

Game Scenes

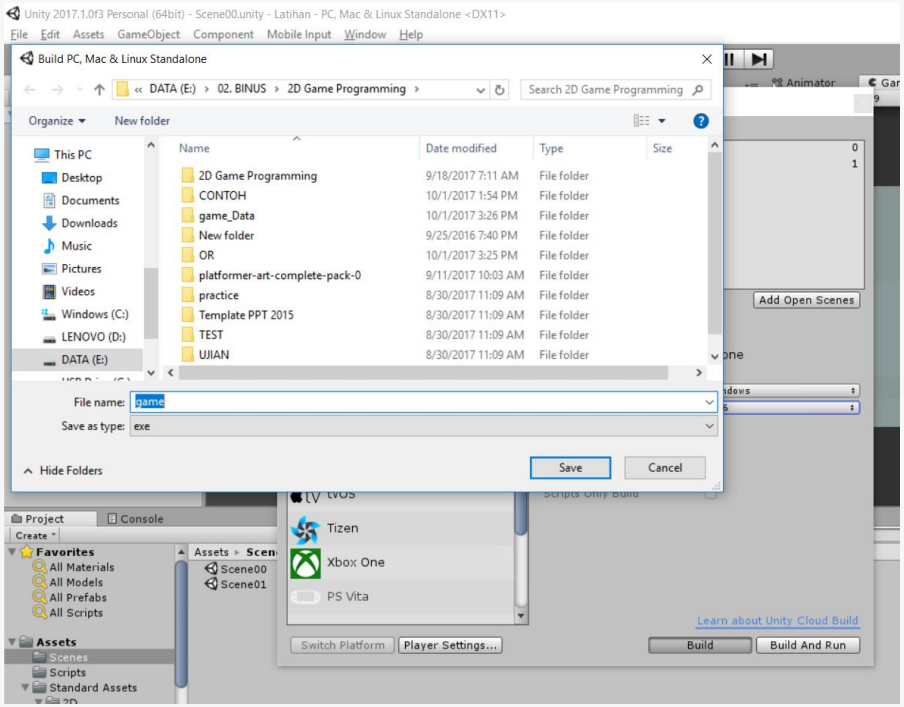


To make sure the game run smoothly you have to add the build to the build settings.

People
Innovation
Excellence

As we are on the build settings…

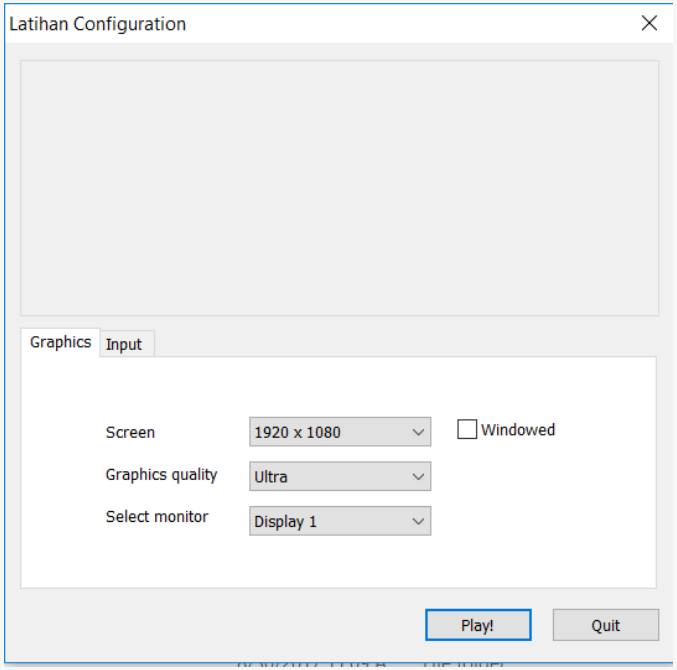
Why don't we just build it…



BINUS UNIVERSITY

People
Innovation
Excellence

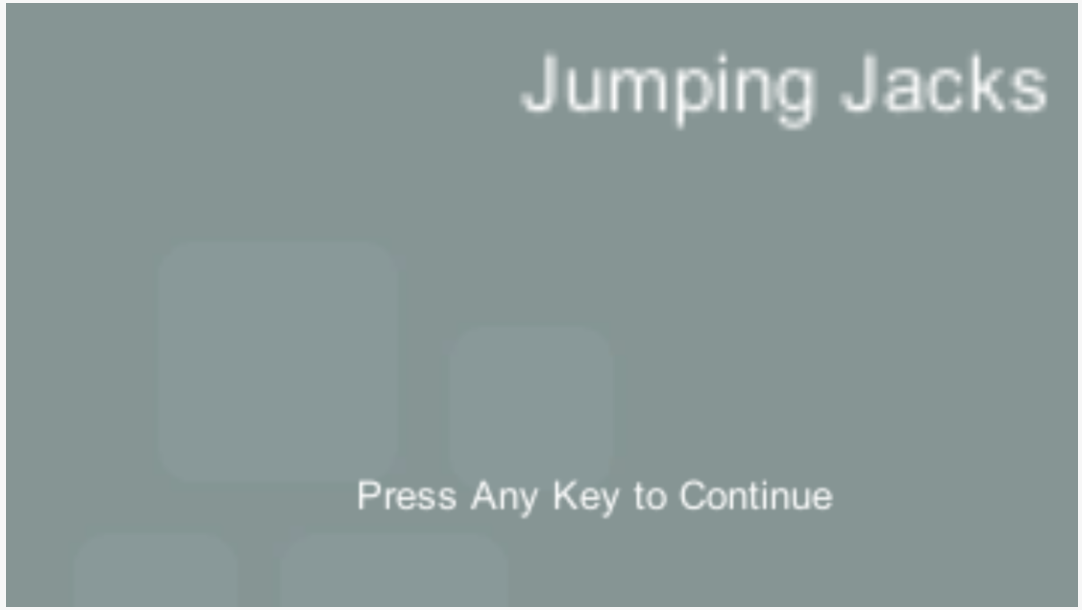
Let's run it...



BINUS UNIVERSITY

People
Innovation
Excellence

And play!!

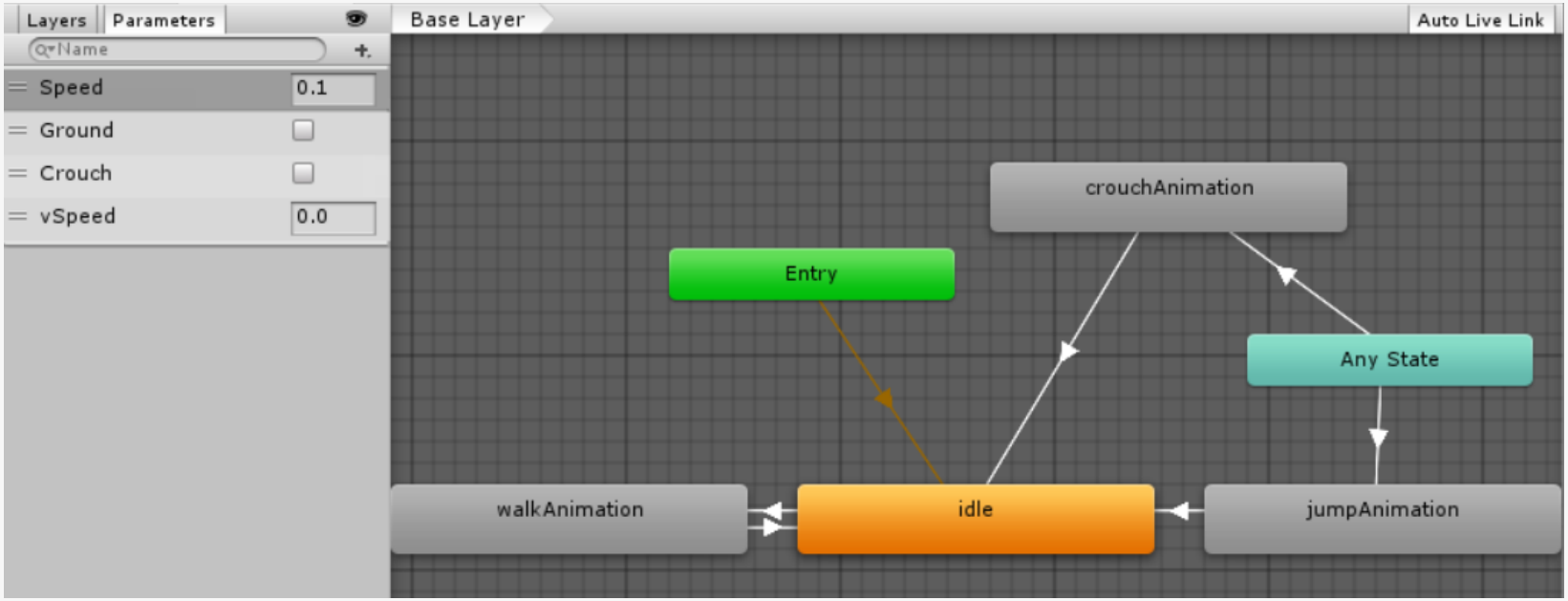


BINUS UNIVERSITY

People
Innovation
Excellence

Tugas Mandiri

Create a crouch animation...

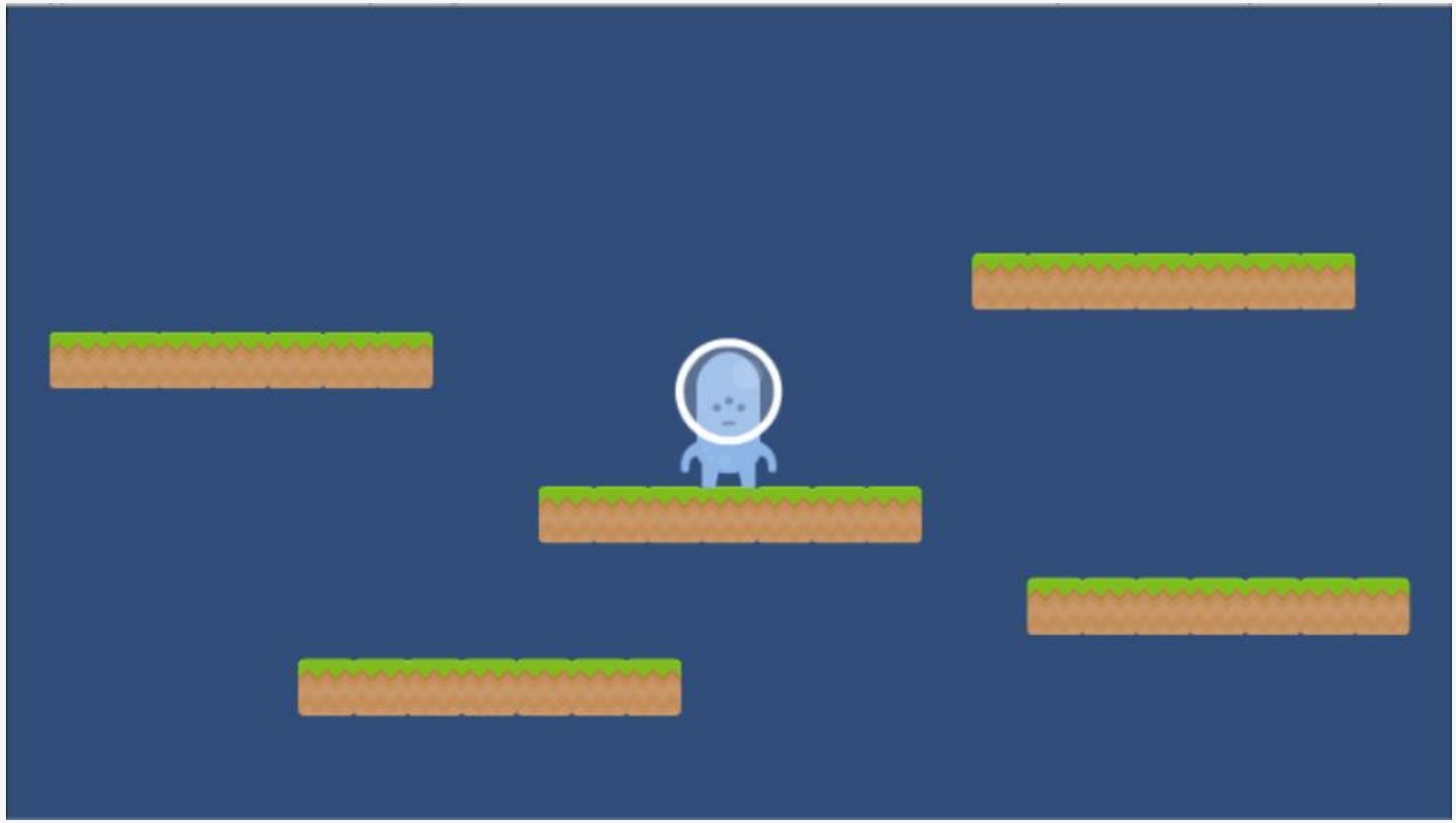


BINUS UNIVERSITY

People
Innovation
Excellence

Tugas Mandiri

Create your own game and add an end scene...



Hint... Modify restarter script

BINUS UNIVERSITY

People
Innovation
Excellence

Using another key?

```
using UnityEngine; using System.Collections;

public class ExampleClass : MonoBehaviour {
void Update()
        {
                if (Input.GetKeyDown(KeyCode.Space))
                print("space key was pressed");
        }
}
```

BINUS UNIVERSITY

People
Innovation
Excellence

References

Freeman, J. (2015). Unity's New 2D Workflow

Vidyasagar. (2014. Unity and C#: Game Loop.CodeProject

Pereira, V. (2014). Learning Unity 2D Game Development by Example. Packt Publishing, Inc. San Francisco. ISBN: 9781783559046