**Unity Certification Preparation:**

**C# Programming**

**Orlando Unity3d Development Meetup**

**Date: 3 June 2018**

**All Copyright Kovert Studios © 2018**

**Contents**

[**1.0** **Introduction to Roll-A-Ball** 3](#_Toc515886981)

[**2.0** **Setting up the Game** 3](#_Toc515886982)

[**3.0** **Moving The Player** 5](#_Toc515886983)

[**3.1** **Update v FixedUpdate** 5](#_Toc515886984)

[**3.1.1** **UPDATE** 5](#_Toc515886985)

[**3.2.1** **FIXEDUPDATE** 5](#_Toc515886986)

[**4.0** **Moving the Camera** 6](#_Toc515886987)

[**5.0** **Setting up the Play Area** 7](#_Toc515886988)

[**6.0** **Creating Collectibles** 7](#_Toc515886989)

[**7.0** **Counting Points & Player Collisions** 8](#_Toc515886990)

[**8.0** **Displaying Score & Text** 9](#_Toc515886991)

# **Introduction to Roll-A-Ball**

Hello, and WELCOME! My name is XXX. I am going to be your instructor for the next hour or so. I appreciate you all taking time out from your busy schedules to be here today.

***WHETHER you are 1) New to C#*** *- at the start of your game development journey* ***or 2)"Stuck-in-a-rut****" and want to get over that "coding" hump* ***…. or 3)"Struggling"*** *to apply the knowledge you learned from a previous course…then this* ***Unity Certification Programmer’s Course*** *is for you?*

*You will be programming using C# “inside” of the* ***unity game engine (IDE)*** *editor on your own laptop or pc either via visual studio 2015/2017 for Windows. Or xcode for the Mac.*

*Before we begin. I do expect you to have downloaded & installed (on your laptops or PCs)* ***1) visual studio 2015 or 2017 or XCODE…, and 2) Unity version 5 or higher.***

We will be doing full code walkthroughs. In other words, ... “As-I-type-You-Type” …So you can follow in the exact same steps & processes as me. Feel free to interrupt, if you have any questions. Also, If I am going to fast let me know. And I will sloooowwww down! ….*Let's* ***#DoDahUnityTing!***

# **Setting up the Game**

https://youtu.be/W\_fAidYRGzs

**CREATE A 1 x NEW PROJECT > 2 x OBJECTS, 3 x MATERIALS, 3 x SCRIPT**

**CLICK** on your **Unity Icon** > **CREATE** a NEW project > and **NAME** the project **"ROLLABALL"**

**CREATE 4 x ASSET / PROJECT FOLDERS > \_MATERIALS, \_PREFABS, \_SCRIPTS, \_SCENES**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Inside ASSETS folder > **(RIGHT CLICK)** Select CREATE > Select folder > F2 > Rename = **"\_MATERIALS"**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Inside ASSETS folder > **(RIGHT CLICK)** Select CREATE > Select folder > F2 > Rename = **"\_PREFABS"**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Inside ASSETS folder > **(RIGHT CLICK)** Select CREATE > Select folder > F2 > Rename = **"\_SCRIPTS"**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Inside ASSETS folder > **(RIGHT CLICK)** Select CREATE > Select folder > F2 > Rename = **"\_SCENES"**

**SAVE SCENE as 1 x NEW SCENE = "LEVEL1"**

**(NAVIGATE to Toolbar TOP LEFT)** FILE > NEW SCENE **(CTRL + N)** > SAVE SCENE as **"LEVEL1.unity"** to your windows explorer file path

**CREATE 1 x OBJECT > "THE PLANE" (F2/Rename GROUND)**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select 3D OBJECT > **PLANE**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select 3D OBJECT > **PLANE** (With PLANE Selected) > **NAVIGATE** to INSPECTOR > TRANSFORM > **Set TRANSFORM.POSITION.X=Y=Z=0. Check SET TRANSFORM.SCALE X=Y=Z=2, Set ROTATION X=Y=Z=0.**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select 3D OBJECT > PLANE > **F2 NAME/RENAME** = **"GROUND"**

Plane should be centered at the ORIGIN or center of our scene. With a length=breadth=2

**SHOW FULL VIEW of PLANE**

**PRESS SHIFT + F** to FULL VIEW entire PLANE (current gameObject) in our SCENE view

**SWITCH OFF GRID**

For purposes of this project. **NAVIGATE** to SCENE window > Gizmos > **UNCHECK** Show Grid

**CREATE 1 x OBJECT SPHERE > “PLAYER”**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > **SELECT 3D OBJECT > SPHERE**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > SELECT 3D OBJECT > SPHERE > **F2 NAME/RENAME** = **"PLAYER"**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > **PLAYER**

Select SPHERE > **PRESS** F to FULL VIEW entire PLANE (current gameObject) in our SCENE view

See how the **PLAYER (SPHERE)** is buried in the PLANE. That’s because both gameobjects are located at the same co-ordinates in the SCENE.

***We need to move the SPHERE up, so it now rests on the plane. Because we know the gameobjects have a standard thickness of X=Y=Z=1. If we move the SPHERE up by 1 in the Y direction. It will sit perfectly in the plane***.

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select 3D OBJECT > PLAYER (SPHERE) **NAVIGATE** to TRANSFORM > **Set TRANSFORM.POSITION.X=Z=0. TRANSFORM.POSITION.Y = 1**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select 3D OBJECT > PLAYER. **NAVIGATE** to TRANSFORM > **Check SET SCALE X=Y=Z=1, Set ROTATION X=Y=Z=0.**

**CREATE 3 x MATERIALS**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select MATERIALS folder > **(RIGHT CLICK)** Select CREATE > Select MATERIALS > F2 Rename OR NAME = **"RED"**

WITH RED MATERIAL Selected > **NAVIGATE** to INSPECTOR > MAIN MAPS > ALBEDO > **LEFT CLICK** > CHOOSE PALETTE COLOUR = **"RED"**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select MATERIALS folder > **(RIGHT CLICK)** Select CREATE > Select MATERIALS > F2 Rename OR NAME = **“GREEN”**

WITH RED MATERIAL Selected > **NAVIGATE** to INSPECTOR > MAIN MAPS > ALBEDO > **LEFT CLICK** > CHOOSE PALETTE COLOUR = **"GREEN"**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select MATERIALS folder > **(RIGHT CLICK)** Select CREATE > Select MATERIALS > F2 Rename OR NAME = **“YELLOW”**

WITH RED MATERIAL Selected > **NAVIGATE** to INSPECTOR > MAIN MAPS > ALBEDO > **LEFT CLICK** > CHOOSE PALETTE COLOUR = **"YELLOW"**

**NAVIGATE** to PROJECT tab > ASSETS > MATERIALS > **Select RED.mat > DRAG n DROP onto "PLAYER"**

**NAVIGATE** to PROJECT tab > ASSETS > MATERIALS > **Select GREEN.mat > DRAG n DROP onto "GROUND"**

**DIRECTIONAL LIGHTING**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select **DIRECTIONAL LIGHT**

**NAVIGATE** to INSPECTOR tab > Select TRANSFORM > **Set TRANSFORM.ROTATION.X = 50 > CHECK/SET TRANSFORM.ROTATION.Y=60. > CHECK/SET TRANSFORM.ROTATION.Z=0**

# **Moving The Player**

<https://youtu.be/7C7WWxUxPZE>

**Add: 1 x RIGID BODY COMPONENT to [PLAYER (gameObject)**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > **Select "PLAYER"**

**NAVIGATE** to INSPECTOR > **(SCROLL DOWN)** ***Add COMPONENT > PHYSICS > RIGID BODY > SELECT "USE GRAVITY" CHECK=”YES”> UNCHECK "IS KINEMATIC”.*** i.e. By default, we have given it a mass. So, on play the capsule should just drop to the floor.

**CREATE A NEW 1 x SCRIPT**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select SCRIPTS folder > **(RIGHT CLICK)** Create > C# Script > F2 **RENAME/NAME** = **"PlayerController"**

If you use ADD COMPONENT to add a script. Creates & Adds a Script in 1 step

OPEN the Script:

# **3.1 Update v FixedUpdate**

# **3.1.1 UPDATE**

If the framerate were to drop but you need a certain calculation to keep executing, like if you were updating a chat or voip client, you would want regular old update. However, because update is not called at a regular timeline. Therefore, the function calls can vary significantly 1)Called only once for every Frame 2)Used for regular updates only 2)Moving NON-PHYSICS Objects, Simple Timers, Receiving Input etc. 2)Time between function calls IRREGULAR

# **3.2.1 FIXEDUPDATE**

However, If rigidbody is added it the event needs to occur in FixedUpdate. Immediately after the FIXEDUPDATE function is called, any necessary PHYSICS calculations are made**. 1)Called every PHYSICS STEP 2)Has the same TIME between function calls i.e. REGULAR 3)Used for adjusting PHYSICS (RIGIDBODY) objects**

**https://unity3d.com/learn/tutorials/topics/scripting/update-and-fixedupdate**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

public float speed = 200f;

private Rigidbody rb;

void Start()

{

rb = GetComponent<Rigidbody>();

}

/\*

Check every frame for player input

Update called before rendering a frame. Only run 1 per frame

FixedUpdate called before any physics calculations. Runs several times a frame

\*/

void FixedUpdate()

{

//Set y-axis = 0.0f i.e. No jump

float moveHorizontal = Input.GetAxis("Horizontal"); //x - axis

float moveVertical = Input.GetAxis("Vertical"); // z-axis

Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

rb.AddForce(movement \* speed \* Time.deltaTime);

}

}//End of PlayerController

# **Moving the Camera**

https://youtu.be/Xcm5H2J95iI

**Set CAMERA 3RD PERSON TO PLAYER gameObject**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > Select "Main Camera"

**NAVIGATE** to INSPECTOR tab > Select TRANSFORM > **Set 1)TRANSFORM.POSITION.X = 0, 2)TRANSFORM.POSITION.Y=10, 3)TRANSFORM.POSITION.Z=-10 > Set TRANSFORM.ROTATION.X=45, TRANSFORM.ROTATION.Y=Z=0**

**CREATE A NEW 1 x SCRIPT**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select SCRIPTS folder > **(RIGHT CLICK)** Create > C# Script > F2 RENAME/NAME = "CameraController"

DRAGnDROP > CAMERACONTROLLER Script onto MAIN CAMERA

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraController : MonoBehaviour

{

public GameObject player; //Drag n Drop Player to player slot in Unity

private Vector3 offset;

// Use this for initialization

void Start()

{

////offset = transform(Camera-Player)

**offset = transform.position - player.transform.position;**

}

// Update is called once per frame

void LateUpdate()

{

//Late Update = guaranteed to run after all other items have been processed

**transform.position = player.transform.position + offset;**

}

}//End of CameraController

# **Setting up the Play Area**

<https://youtu.be/dahT0wRVO1Q>

In the scene gameobjects, can hold other gameobjects

Create a parent Wall gameobject to hold our child Wall gameobject

**CREATE 1 x PLAYER OBJECT > CUBE F2/Rename = WALLS**

**NAVIGATE** to HIERARCHY tab > (RIGHT CLICK) > CREATE EMPTY > F2 RENAME = "WALLS"

**SEELCT** WALLS > **NAVIGATE** to INSPECTOR tab > Select TRANSFORM > Select GEAR > Reset Component

**CREATE WALLS (NORT, SOUTH, EAST and WEST)**

**NAVIGATE** to INSPECTOR tab > Select TRANSFORM > Select GEAR > Reset Component

(Check/Select empty GameObject) NAVIGATE to HIERARCHY tab > (RIGHT CLICK) > CREATE > 3D OBJECT > CUBE

**NAVIGATE** to HIERARCHY tab > Select CUBE > F2 RENAME = "WESTWALL"

**NAVIGATE** to HIERARCHY tab > Select WESTWALL > Select TRANSFORM > **Set TRANSFORM.SCALE.X = 0.5, Y=2, Z=20.5, Set TRANSFORM.POSITION X= -10,Y=Z=0**

**NAVIGATE** to HIERARCHY tab > Select WESTWALL > **(RIGHT CLICK)** DUPLICATE "WESTWALL" > F2/RENAME "WESTWALL" = "EASTWALL"

**NAVIGATE** to HIERARCHY tab > Select WESTWALL > Select TRANSFORM > Set TRANSFORM.SCALE.X = 0.5, Y=2, Z=20.5, Set TRANSFORM.POSITION X= -10,Y=Z=0

**NAVIGATE** to HIERARCHY tab > Select EASTWALL > Select TRANSFORM > [CHECK] Set TRANSFORM.SCALE.X = 0.5, Y=2, Z=20.5, Set TRANSFORM.POSITION X= +10,Y=Z=1

**NAVIGATE** to HIERARCHY tab > Select EASTWALL > (RIGHT CLICK) DUPLICATE "EASTWALL" > F2/RENAME "EASTWALL" = "NORTHWALL"

**NAVIGATE** to HIERARCHY tab > Select NORTHWALL > Select TRANSFORM > Set TRANSFORM.SCALE.X = 0.5, Y=2, Z=20.5, Set TRANSFORM.POSITION X=Y=0, Z=10. Set TRANSFORM.ROTATION X=Z=0, Y=90

**NAVIGATE** to HIERARCHY tab > Select NORTHWALL > [ RIGHT CLICK] DUPLICATE "NORTHWALL" > F2/RENAME "NORTHWALL" = "SOUTHWALL"

**NAVIGATE** to HIERARCHY tab > Select SOUTHWALL > Select TRANSFORM > Set TRANSFORM.SCALE.X = 0.5, Y=2, Z=20.5, Set TRANSFORM.POSITION X=Y=0, Z=-10

**NAVIGATE** to PROJECT tab > ASSETS > MATERIALS > **Select GREEN.mat > 4 x DRAG n DROP onto NORTHWALL, SOUTHWALL, EASTWALL, WESTWALL**

# **Creating Collectibles**

<https://youtu.be/HlDGSStxuHI>

**CREATE COLLECTIABLE OBJECTS > 1 x CUBE =”PICKUP”**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > CREATE > 3D OBJECT > CUBE

**NAVIGATE** to INSPECTOR tab > Select TRANSFORM > Select GEAR > Reset TRANSFORM Component (to Origin)

**NAVIGATE** to HIERARCHY tab > Select CUBE > F2 RENAME = "PICKUP"

**NAVIGATE** to HIERARCHY tab > Select PICKUP > **Select TRANSFORM > Set TRANSFORM.SCALE.X=Y=Z=0.5, Set TRANSFORM.POSITION X=Z=0,Y=1, Set TRANSFORM.ROTATION X=Y=Z=45 degrees**

**NAVIGATE** to PROJECT tab > ASSETS > MATERIALS > Select YELLOW.mat > DRAG n DROP onto PICKUP

**CREATE A NEW 1 x SCRIPT**

**NAVIGATE** to PROJECT tab > Select ASSETS folder > Select SCRIPTS folder > **(RIGHT CLICK)** Create > C# Script > F2 **RENAME/NAME** = "RotateScript"

**DRAGnDROP** "RotateScript" onto PICKUP object

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class RotateScript : MonoBehaviour

{

// Update is called once per frame

void Update ()

{

transform.Rotate(new Vector3(15, 30, 45) \* Time.deltaTime);

}

}//End of RotateScript

**CREATE PICKUP PREFAB**

**NAVIGATE** to HIERARCHY tab > Select PICKUP object > DRAG n DROP to ASSETS/ PREFABS folder = > **NAVIGATE** to PROJECT tab > ASSETS > PREFABS folder.

**NAVIGATE** to HIERARCHY tab > Select PICKUP > **(RIGHT CLICK)** Delete PICKUP object

**CREATE "PICKUPS" empty game object**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** > CREATE EMPTY > F2 RENAME "PICKUPS"

**NAVIGATE** to INSPECTOR tab > Select TRANSFORM > Select GEAR > Reset TRANSFORM Component (to Origin)

**NAVIGATE** to HIERARCHY tab > Select PICKUPS > Select TRANSFORM > Set TRANSFORM.SCALE.X=Y=Z=0.

**DRAGnDROP** "PICKUP PREFAB" onto PICKUP gameobject

**NAVIGATE** to HIERARCHY tab > Select PICKUPS > PickUp > **DUPLICATE** Pickup > 12 x PickUp(1)...DRAG and DROP

# **Counting Points & Player Collisions**

https://youtu.be/XtR29MmzuT0

**Set up PLAYER COLLISIONS**

**NAVIGATE** to PROJECTS > ASSETS > \_SCRIPTS > Select "PLAYERCONTROLLER" script > DOUBLE CLICK > edit >

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

public float speed = 200f;

private Rigidbody rb;

void Start()

{

rb = GetComponent<Rigidbody>();

}

/\*

Check every frame for player input

Update called before rendering a frame. Only run 1 per frame

FixedUpdate called before any physics calculations. Runs several times a frame

\*/

void FixedUpdate()

{

//Set y-axis = 0.0f i.e. No jump

float moveHorizontal = Input.GetAxis("Horizontal"); //x - axis

float moveVertical = Input.GetAxis("Vertical"); // z-axis

Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

rb.AddForce(movement \* speed \* Time.deltaTime);

}

void OnTriggerEnter(Collider other)

{

//Destroy(other.gameObject);

if (other.gameObject.CompareTag("PICKUP"))

{

other.gameObject.SetActive(false);

}

}

}//End of PlayerController

# **Displaying Score & UI Text**

https://youtu.be/bFSLI2cmYYo

**Create the SCORECOUNTER**

Declare and Define count = 0 in void Start() function

**CREATE UI/UX Text Elements 1#**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** UI > Text > **F2/RENAME** = “**CountText”**

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM > GEAR > RESET Component

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > Text(Script) > Color > WHITE R=G=B=A=256

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > Text(Script) > **TYPE** Text ="Count Text"

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM > **Click** ANCHOR/RESET (Middle/Center) Menu > **SHIFT + ALT + SELECT (TOP n LEFT)**

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM **> Set POS X = 10, POS Y = -10, POS Z=0.**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** PLAYER > **NAVIGATE** to INSPECTOR > **DRAGnDROP** CountTEXT (HIERARCHY > CANVAS > into PLAYER > INSPECTOR > “PLAYER CONTROLLER SCRIPT” > CountText

**Add UnityEngine. UI namespace to SCRIPT**

Now **NAVIGATE** to the PROJECT tab > ASSETS > \_SCRIPTS > Select **(DOUBLECLICK)** PlayerController > Add/type using UnityEngine.UI;

**Add variable to hold scoreText**

public Text countText;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class PlayerController : MonoBehaviour

{

public float speed = 200f;

private Rigidbody rb;

public Text countText;

private int count;

void Start()

{

rb = GetComponent<Rigidbody>();

count = 0;

SetCountText();

}

/\*

Check every frame for player input

Update called before rendering a frame. Only run 1 per frame

FixedUpdate called before any physics calculations. Runs several times a frame

\*/

void FixedUpdate()

{

//Set y-axis = 0.0f i.e. No jump

float moveHorizontal = Input.GetAxis("Horizontal"); //x - axis

float moveVertical = Input.GetAxis("Vertical"); // z-axis

Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

rb.AddForce(movement \* speed \* Time.deltaTime);

}

void OnTriggerEnter(Collider other)

{

//Destroy(other.gameObject);

if (other.gameObject.CompareTag("PICKUP"))

{

other.gameObject.SetActive(false);

//count=count+1;

count += 1;

SetCountText();

}

}

void SetCountText()

{

countText.text = "Count: " + count.ToString();

}

}//End of PlayerController

**CREATE UI/UX Text Elements 2#**

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** **UI > Text > F2/RENAME** = **"WINTEXT"**

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM > GEAR > RESET Component

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > Text(Script) > Color > WHITE **R=G=B=A=256**

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM > FONT = 24

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > PARAGRAPH > ALIGNMENT = CENTER & MIDDLE

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > (Add PLACEHOLDER text) Text(Script) > Text ="WinText"

(With Text Selected/Highlighted in Blue) > **NAVIGATE** to INSPECTOR > RECT TRANSFORM > Set POS X = Z = 0 POS Y = 75

**NAVIGATE** to HIERARCHY tab > **(RIGHT CLICK)** CANVAS > WinText into SLOT

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class PlayerController : MonoBehaviour

{

public float speed = 400f;

private Rigidbody rb;

public Text countText;

private int count;

public Text winText;

void Start()

{

rb = GetComponent<Rigidbody>();

count = 0;

SetCountText();

winText.text = "";

}

/\*

Check every frame for player input

Update called before rendering a frame. Only run 1 per frame

FixedUpdate called before any physics calculations. Runs several times a frame

\*/

void FixedUpdate()

{

//Set y-axis = 0.0f i.e. No jump

float moveHorizontal = Input.GetAxis("Horizontal"); //x - axis

float moveVertical = Input.GetAxis("Vertical"); // z-axis

Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

rb.AddForce(movement \* speed \* Time.deltaTime);

}

void OnTriggerEnter(Collider other)

{

//Destroy(other.gameObject);

if (other.gameObject.CompareTag("PICKUP"))

{

other.gameObject.SetActive(false);

//count=count+1;

count += 1;

SetCountText();

}

}

void SetCountText()

{

countText.text = "Count: " + count.ToString();

if(count > 9)

{

winText.text = "You Win !";

}

}

}//End of PlayerController

**CREATE BUILD & RUN the game**

**NAVIGATE** to TOOLBAR TOP LEFT > Select FILE > BUILD SETTINGS > PLATFORM > PC, MAC & Linux STANDALONE

**NAVIGATE** to PROJECT > ASSETS > \_SCENES folder > DRAG n DROP to BUILD SETTINGS / SCENES IN BUILD

**NAVIGATE** to TOOLBAR TOP LEFT > Select FILE > BUILD SETTINGS > Click BUILD > Name file = "Rollaball" (will be saved as an .EXE)

C:\Users\SeanA\Desktop\\_UNITYPROJS\\_MHJ\ROLLABALL

..TO RUN THE GAME…YOU run it as an Executable application.

/END