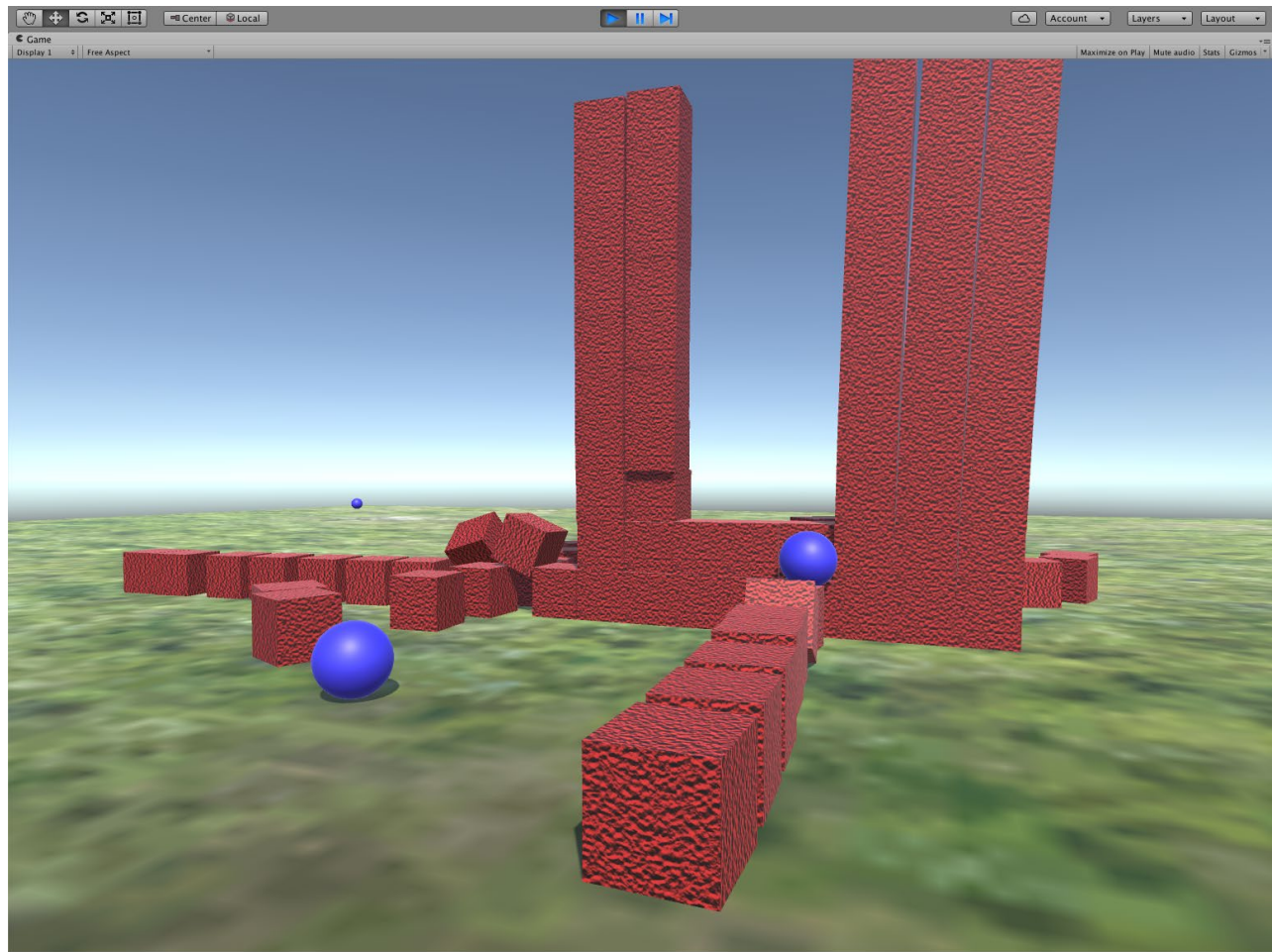


# Tutorial 2

Dr. Osama Halabi

---

For this exercise, you will create a basic game prototype. You will be exposed to various Unity GameObjects and Components. Perform the following steps.

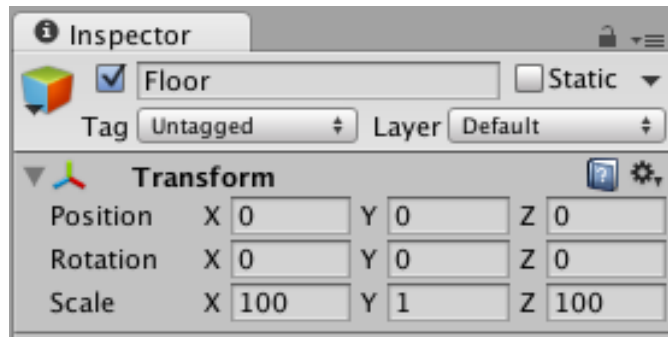


## Create a New Project

1. Create New Project (File - - > New Project. . .)
2. (optional) Download and add Standard Assets from unity assets store:
  - a. Windows → Mange Packages → My assets
  - b. Import Standard Assets → Environment

## Create a Floor

1. From within the Hierarchy Window, click Create; select Cube.
2. Click on the Cube listing in the Hierarchy Window and rename it to “Floor”.
3. Verify that the Floor is at the center of the world
  - a. With Cube selected, go to the Inspector Window and click the Transform’s Cog icon, then select Reset Position. If it is grey, then it is already centered.
4. Reshape the cube into a Floor
  - a. In the Inspector Window, set the Transform:Scale:X to 100, Transform:Scale:Y to 1, and Transform:Scale:Z to 100



## Build a Brick Wall: The Master Brick

We will make a brick wall that will start with one Master Brick that will then be cloned multiple times to make a wall

1. Build the Master Brick
  - a. From the Hierarchy Window, click on the grey background in the Window to make sure no other objects are selected.
  - b. Select Create - - > Cube
  - c. Move the cube to floor
    - i. In the Inspector Window, set the Transform:Position:X to 0, Transform:Position:Y to 1, and Transform:Position:Z to 0.
  - d. Focus the Scene on the Cube
    - i. Make sure the Cube is still selected in the Hierarchy Window.
    - ii. Hover the mouse over the Scene window.
    - iii. Press the F key.
  - e. Add physics to the Cube
    - i. From the Hierarchy Window, select the Cube
    - ii. From the Top Menu, select Component - - > Physics - - > Rigidbody
  - f. Color the Cube
    - i. Create a new Material. From the Project Window, click Create - - > Material
    - ii. Name this new Material, “Red”
    - iii. In the Inspector Window, click the Color Picker icon

- iv. Set the color to any shade of red.
  - v. Apply the Material to the Cube
    - 1. From the Project Window, drag the Red Material over to the Hierarchy Window and drop it on the Cube
- 2. Test the brick
  - a. In the Hierarchy Window, select the Cube
  - b. Lift and rotate the cube
    - i. In the Inspector Window, change the Transform:Position:Y to 15. Change the Transform:Rotation:X to 40
  - c. Press the Play button
  - d. Verify that the Cube falls from the sky and tumbles a little
  - e. When done testing, press the Play button again (NOT the Pause button)
  - f. Set the Y position back to 1 and the X rotation back to 0

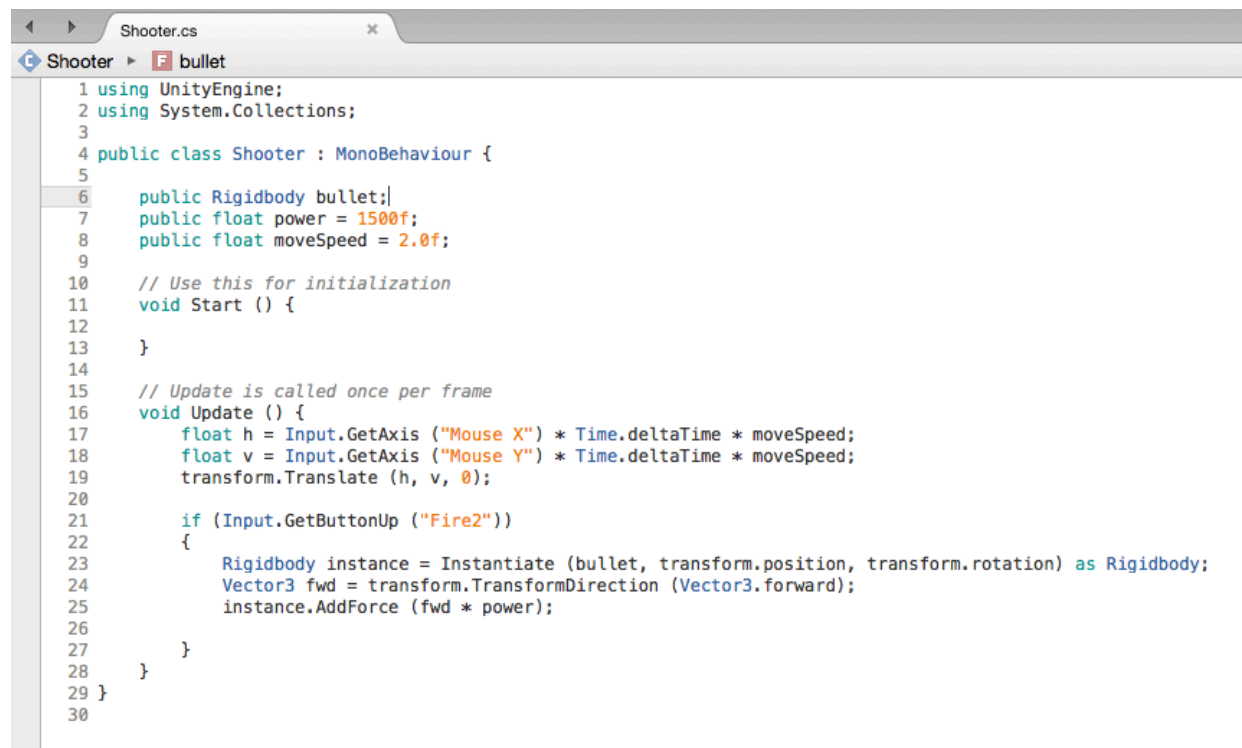
## Building a Brick Wall: the Rest of the Bricks

- 1. Duplicate the Master Brick
  - a. In the Hierarchy Window, click on the Cube
  - b. Right click and select Duplicate
- 2. Move the new Cube using Snapping
  - a. Hold down the Command key
  - b. In the Scene Window, click and hold on the Red (Left) Axis
  - c. Using the mouse, move the new cube to the right, right next to the original cube.
- 3. Repeat Step 2 until you have 10 cubes in a row
- 4. Create an empty Object. From the Top Menu, select GameObject - - > Create Empty
- 5. Rename the empty GameObject to CubeHolder
- 6. Move the CubeHolder
  - a. In the Inspector Window, change the Transform:Position:X to 4.5, Transform:Position:Y to 0.5, and Transform:Position:Z to -1
- 7. Move all of the Cubes to the CubeHolder
  - a. In the Hierarchy Window, click on the top Cube, hold the Shift key, then click on the bottom Cube.
  - b. Drag the selected Cubes and drop them on the CubeHolder.

8. Duplicate the CubeHolder
  - a. In the Hierarchy Window, select the CubeHolder
  - b. Duplicate it
  - c. Hold the Control/Command key
  - d. In the Scene Window, click and hold on the Yellow (Up) Axis
9. Repeat Step 8 until the wall is 10 CubeHolders high

## Knock Down this Wall

1. Move the MainCamera
  - a. In the Hierarchy Window, select the MainCamera
  - b. In the Inspector Window, change the Transform:Position:X to 4, Transform:Position:Y to 3, Transform:Position:Z to -15
  - c. All Rotation values should remain at 0
2. Create a c# Script
  - a. In the Project Window, select Create - - > C# Script
  - b. Name the Script "Shooter"
  - c. In the Project Window, double-click the Shooter Script. The script default editor will open, usually the Microsoft Visual Studio.



```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Shooter : MonoBehaviour {
5
6     public Rigidbody bullet;
7     public float power = 1500f;
8     public float moveSpeed = 2.0f;
9
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void Update () {
17        float h = Input.GetAxis ("Mouse X") * Time.deltaTime * moveSpeed;
18        float v = Input.GetAxis ("Mouse Y") * Time.deltaTime * moveSpeed;
19        transform.Translate (h, v, 0);
20
21        if (Input.GetButtonUp ("Fire2"))
22        {
23            Rigidbody instance = Instantiate (bullet, transform.position, transform.rotation) as Rigidbody;
24            Vector3 fwd = transform.TransformDirection (Vector3.forward);
25            instance.AddForce (fwd * power);
26
27        }
28    }
29 }
30
```

- e. Save your code, close the Visual Studio Window
3. Attach the Shooter Script to the MainCamera
  - a. From the Project Window, click on the Shooter Script and drag it to the Hierarchy Window where you will drop it on the MainCamera object.
4. Save the Scene
5. Create the bullet
  - a. From the Hierarchy Window, select Create - - > Sphere
  - b. Create a Material for the Sphere
    - i. From the Project Window, select Create - - > Material
    - ii. Name the new Material, "BulletColor"
    - iii. In the Inspector Window, use the Color Picker to set the color to a shade of blue.
  - c. Apply the Material to the Sphere
    - i. Drag the BulletColor Material from the Project Window onto the Sphere object in the Hierarchy Window.
6. Make the bullet have physics properties
  - a. a. In the Hierarchy Window, click on the Sphere
  - b. b. From the Top Menu, select Component - - > Physics - - > Rigidbody
7. Convert the Sphere to a Prefab
  - a. In the Project Window, create a Prefabs folder
  - b. Drag the Sphere from the Hierarchy Window and drop it in the Prefabs folder you just created
  - c. In the Project Window, rename the Sphere object to Projectile
  - d. In the Hierarchy Window, delete the Sphere object (right-click - - > Delete)
8. Assign the bullet to the Projectile Prefab
  - a. In the Hierarchy Window, select the Main Camera
  - b. In the Project Window, drag the Projectile prefab to the Inspector Window and drop on the Shooter:Bullet value.
  - c. Save the Scene
9. Verify that it works
  - a. Click Play
  - b. Enjoy!
  - c. Once you've played with it for a bit, try changing around some of the parameters and experimenting.

## Add Sound Effects

1. Add script for sound as below
2. Assign sound file to shooterSFX variable in the shooter component
3. Add Audio Source and assign background sound file
4. Enjoy!!

```
Shooter.cs
Shooter ▶ Update ()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Shooter : MonoBehaviour {
6     public Rigidbody bullet;
7     public float moveSpeed = 1000.0f;
8     public float power = 100.0f;
9
10    public AudioClip shootSFX;
11
12    // Use this for initialization
13    void Start () {
14
15    }
16
17    // Update is called once per frame
18    void Update () {
19
20        float h = Input.GetAxis ("Mouse X") * Time.deltaTime * moveSpeed;
21        float v = Input.GetAxis ("Mouse Y") * Time.deltaTime * moveSpeed;
22        transform.Translate (h, v, 0);
23
24        if (Input.GetButtonDown ("Fire2"))
25        {
26            Rigidbody instant = Instantiate (bullet, transform.position, transform.rotation) as Rigidbody;
27            Vector3 fwd = transform.TransformDirection (Vector3.forward);
28            instant.AddForce (fwd * power);
29
30            if (shootSFX)
31            {
32                if(instant.GetComponent<AudioSource>())
33                {
34                    instant.GetComponent<AudioSource>().PlayOneShot(shootSFX);
35                }
36                else
37                {
38                    AudioSource.PlayClipAtPoint (shootSFX, instant.transform.position);
39                }
40            }
41        }
42    }
43 }
44
```