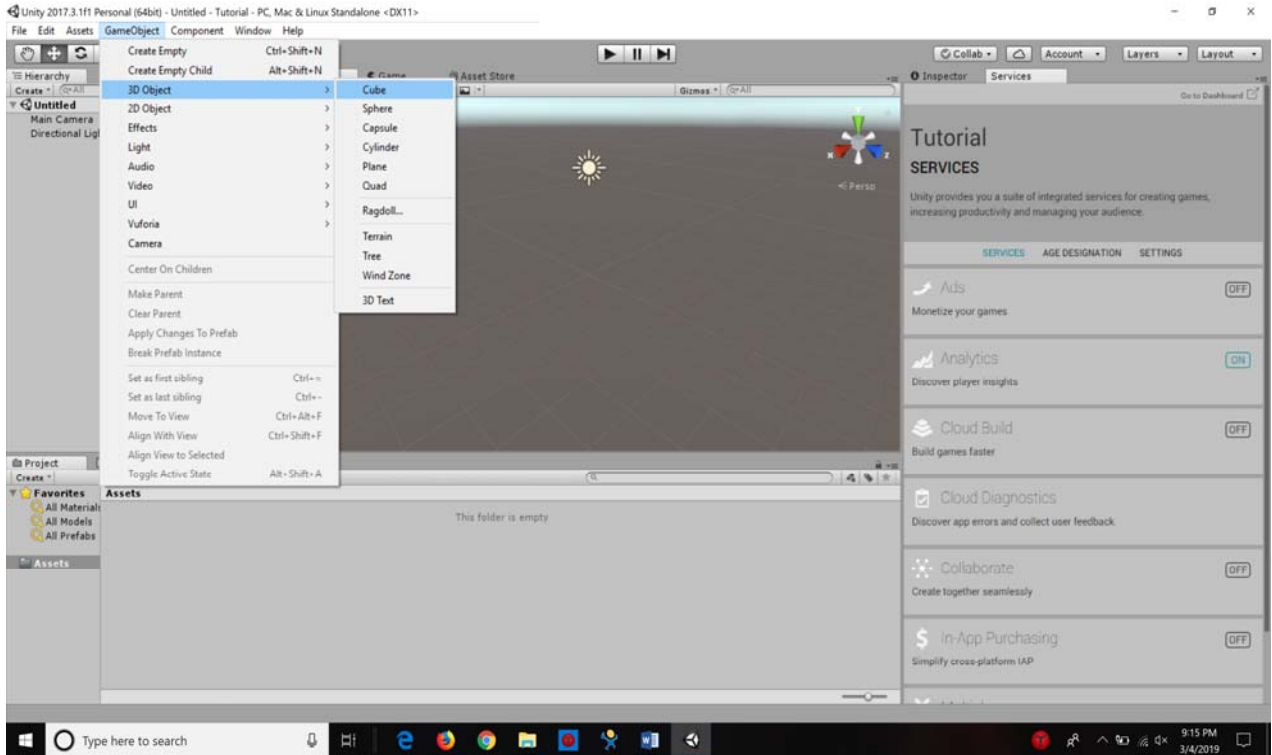
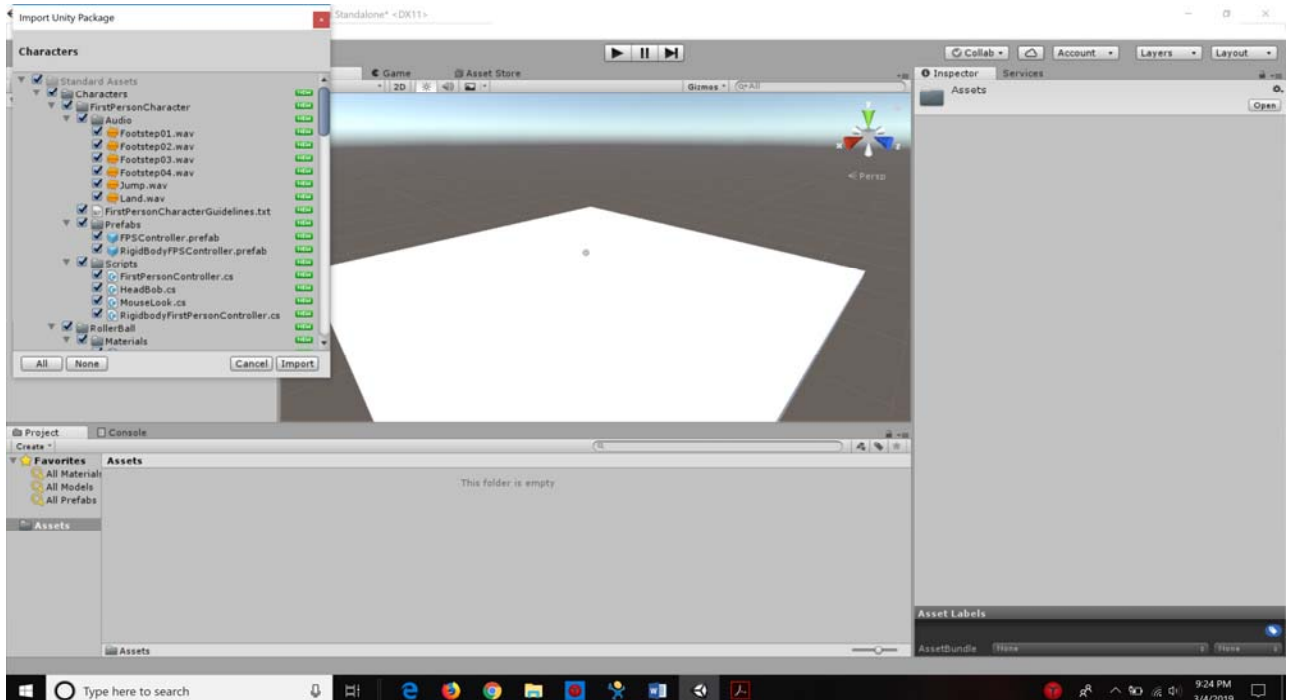


Unity Tutorial Basics - Create a Survival Game

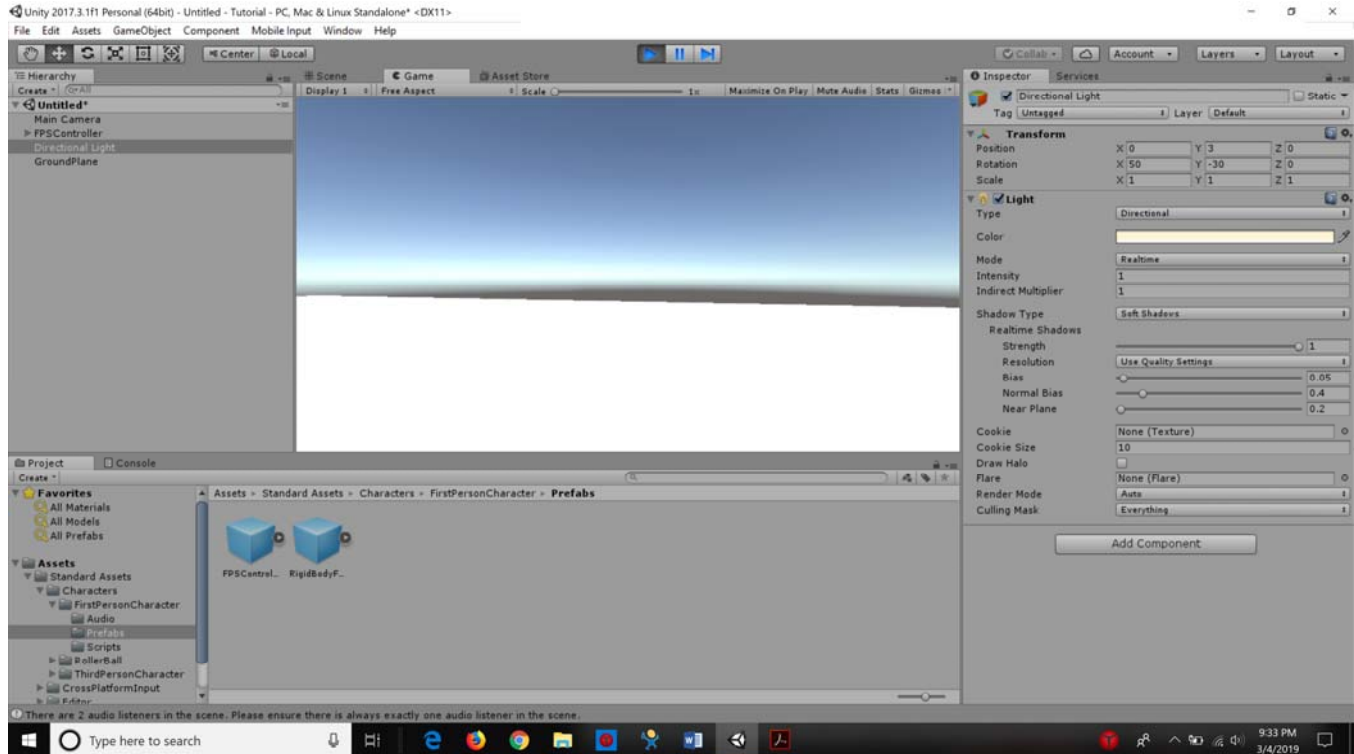
- 1) Create a new project with no imports
- 2) Create a game object-> cube



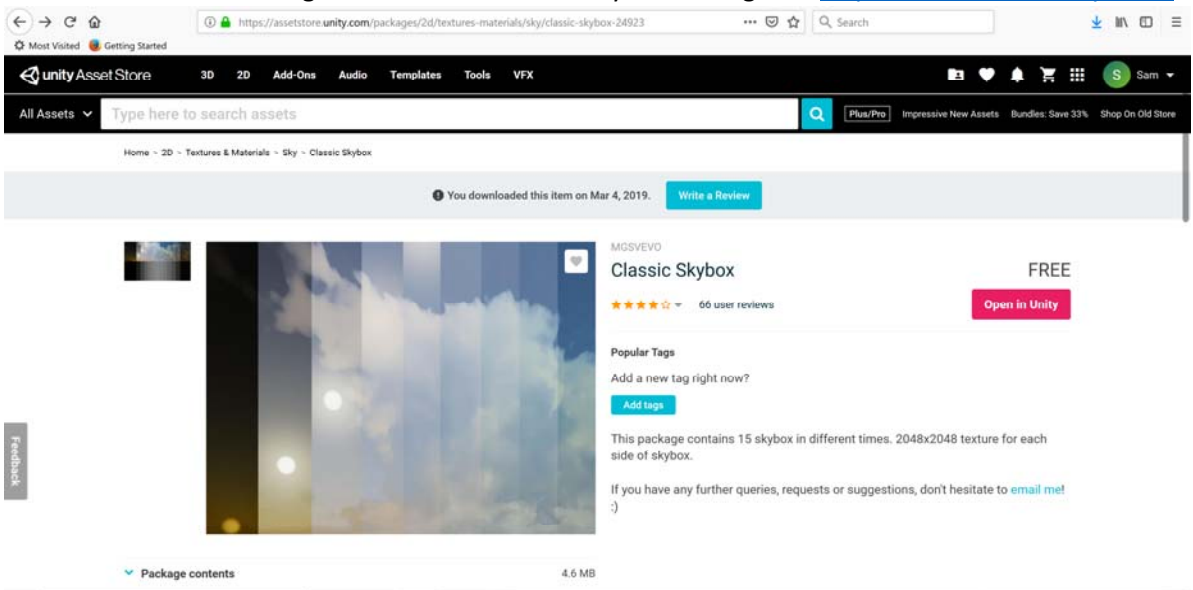
- 3) Scale x=50, y=0, z=50
- 4) Rename Cube to groundplane [NOTE: Unity is case sensitive]
- 5) Import a character: Right click on assets window and import "characters"



- 6) Go to standardassets->characters->FirstPersonController->Prefabs. Then drag first person controller on to the plane area. Hit “f” to zoom in on highlighted section (cylinder). Drag the first person controller UP.
- 7) Create a Point light from game object. Drag it up so that you can see the first person controller. Increase the Range to 50
- 8) Press Control +P or “play” Button to go to game view. You can walk around, move and jump.

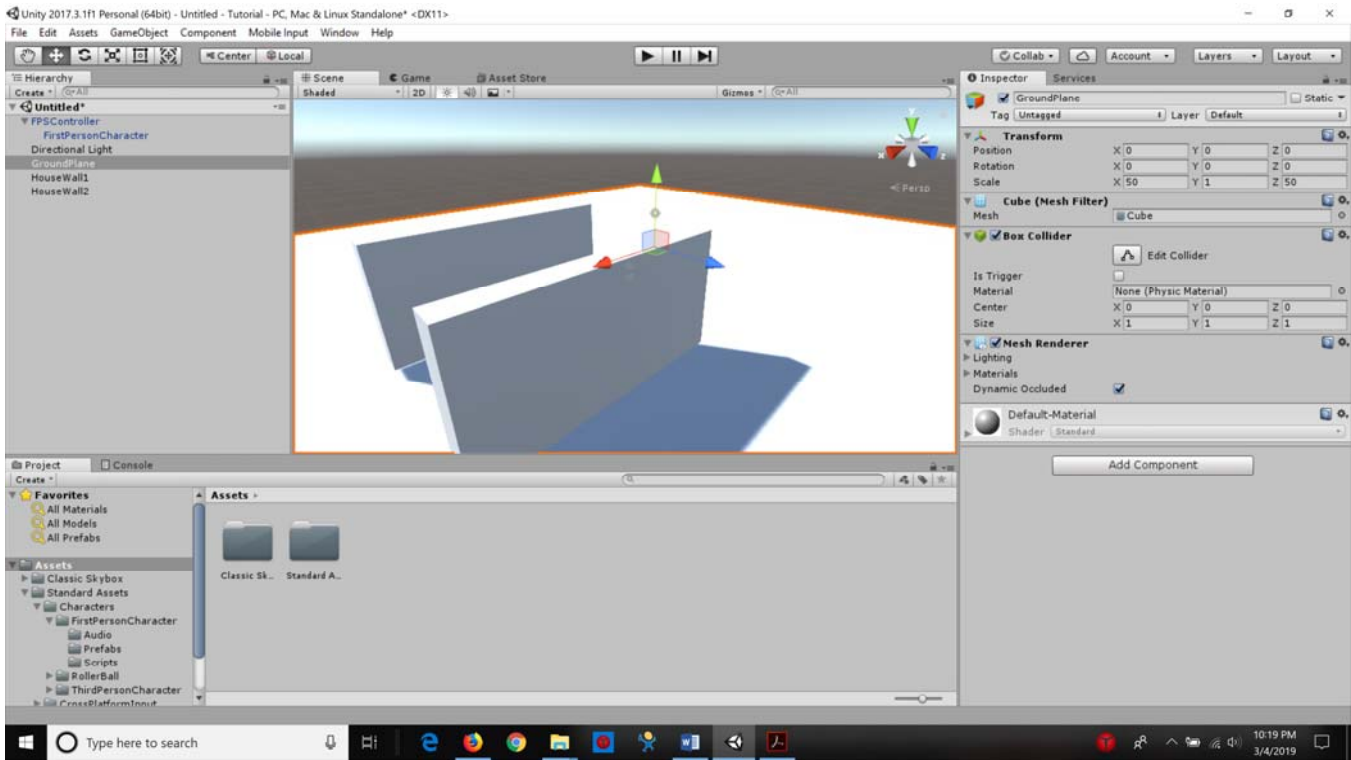


- 9) Delete Main camera. The first person controller comes with a camera
- 10) Create a SKYBOX: Click on Main camera inside the first person controller. You will be able to see inspector at right frame. Click on add component -> rendering->skybox [Last button], By default there is no texture, so let’s import some textures.
- 11) Go to assets folder and right click to add assets->skyboxes OR got to <https://assetstore.unity.com/>

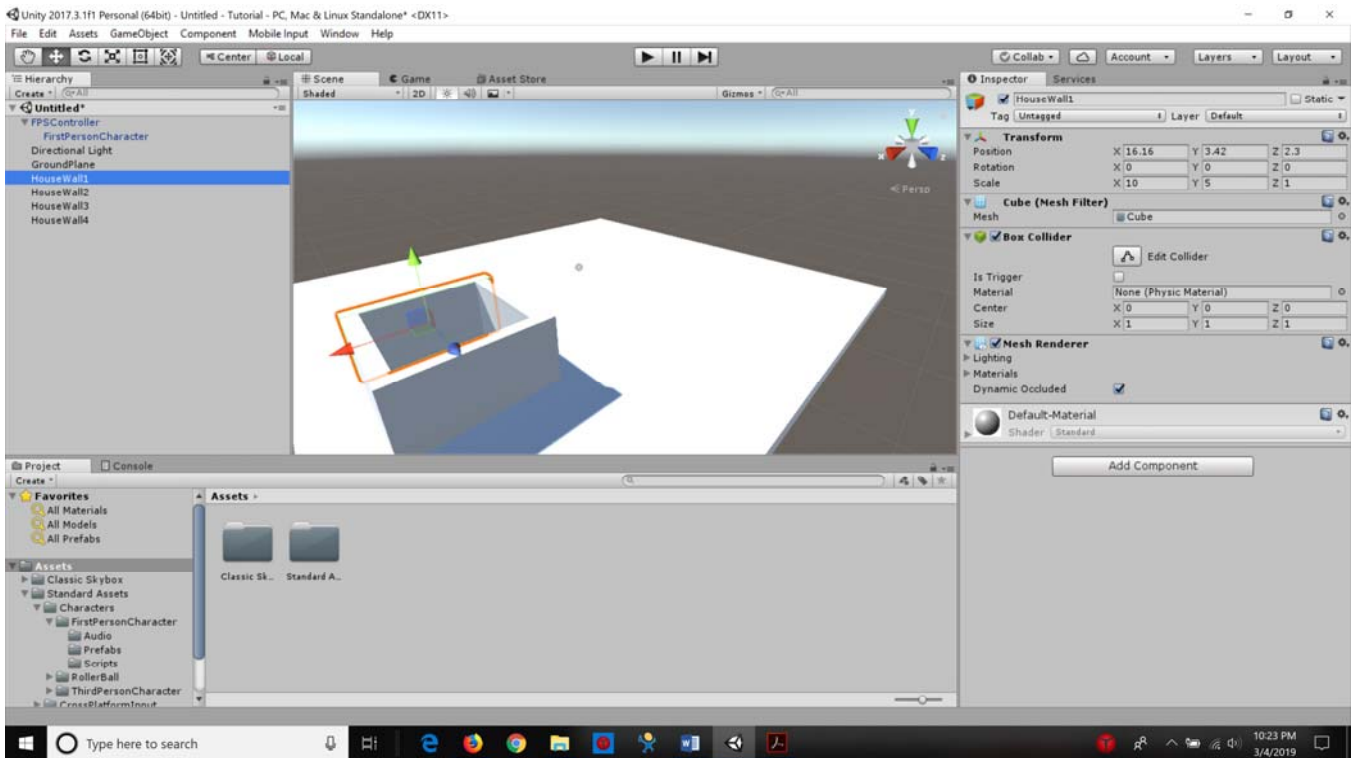


- 12) Click again on Main camera [left frame] and drag and drop the texture in skybox [right frame]

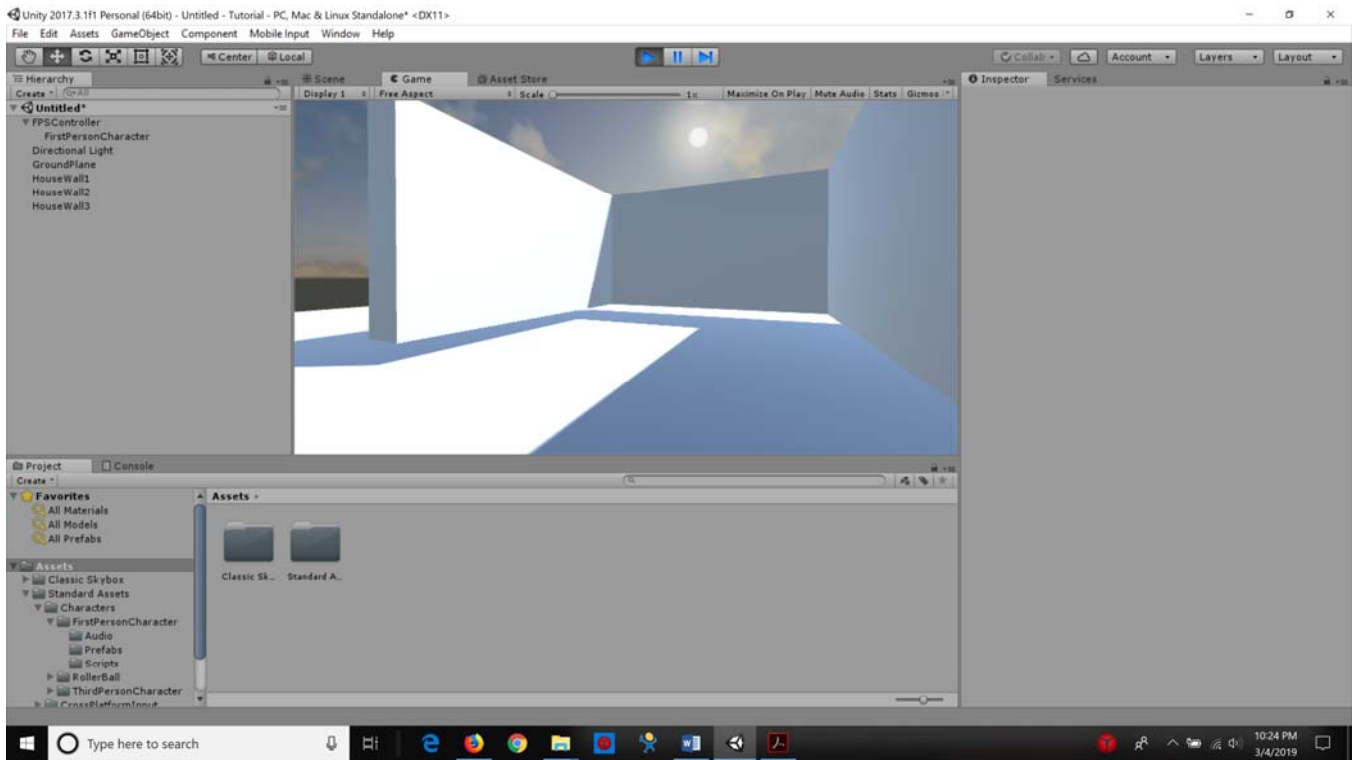
- 13) **Create a House:** remember Unity is a gaming engine and not for modeling objects.
- 14) Create a cube with scale value as x=10, y=5. Rename the cube to house wall1
- 15) Press Control +d to duplicate it. Now move the wall.



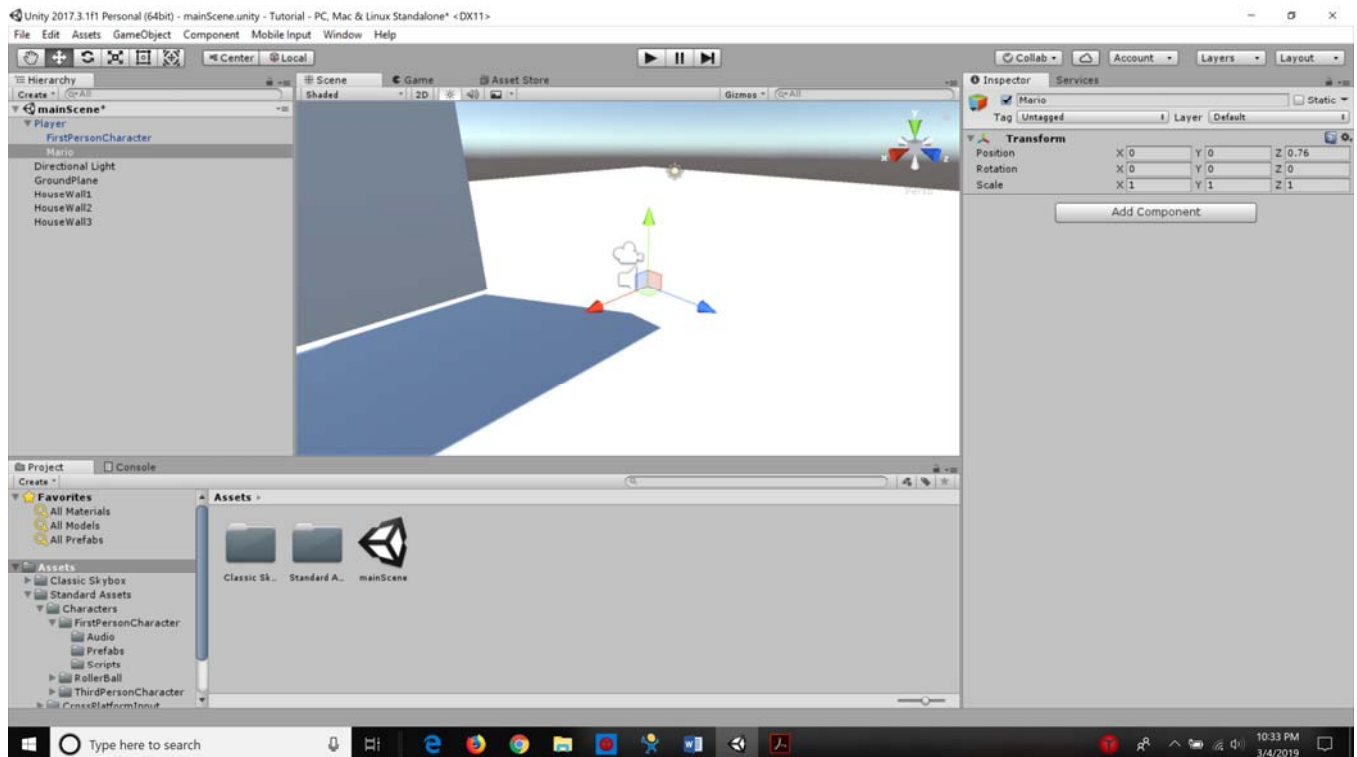
- 16) Duplicate both wall1 and wall2. To create wall 3 and wall4. Rotate y axis to 90



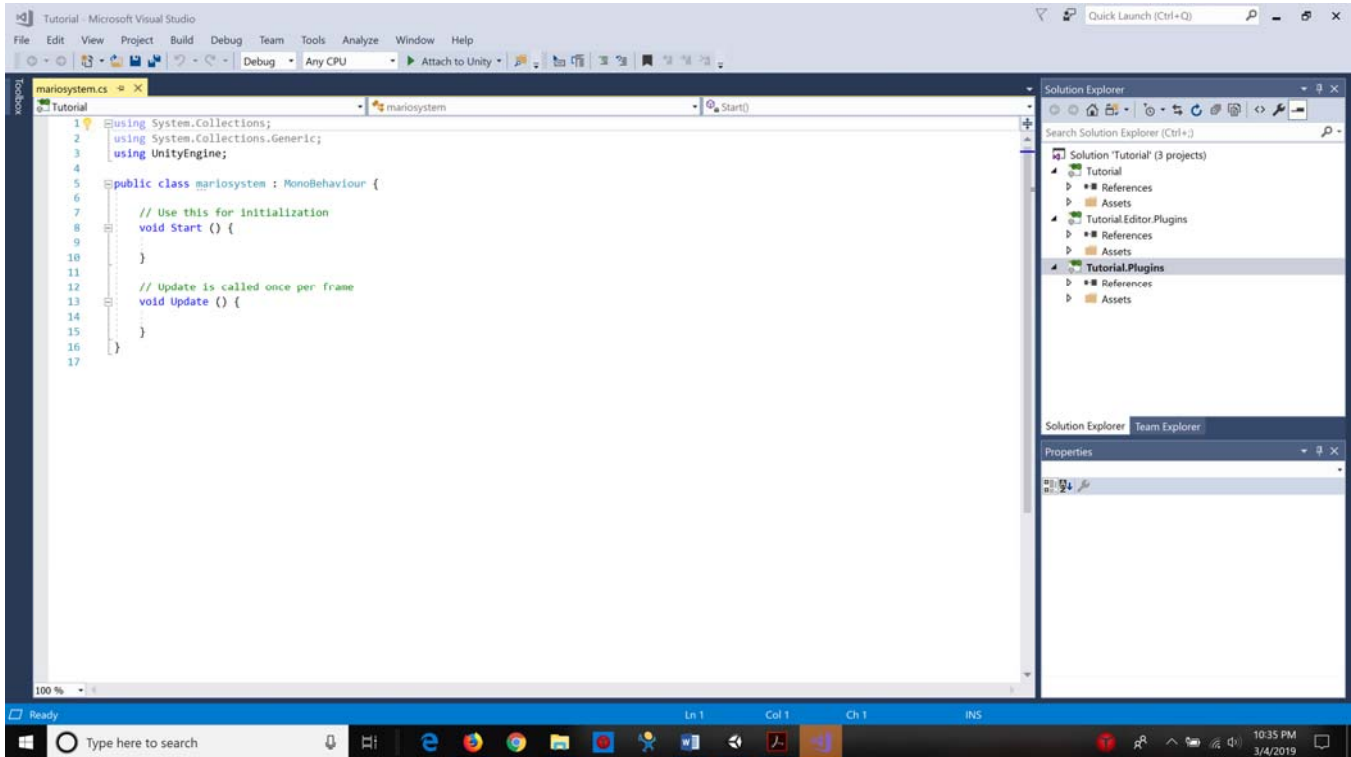
- 17) Delete one wall and make the character move around in the space.



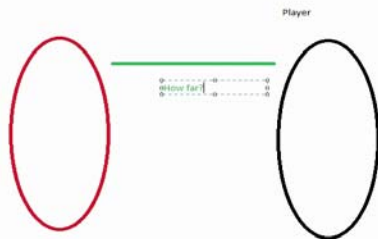
- 18) File->Save scene. Call it "mainScene"
- 19) Rename first "person game controller" to "player" in left frame
- 20) Create an empty game object and call it "Mario". Drag it on the player object. Drag it again on the main camera. Put transform x,y and z to 0 [it will be in middle of camera]. Then drag the Mario object slightly away from player. [NOTE: this is called as parenting an object]



21) Now select “Mario” on left frame and add a component [in right frame]. Add component-> new script-> mariosystem. Now double click the script in mono develop (now in visual studio).



22) Ray cast function: In mono develop delete the function start and update.

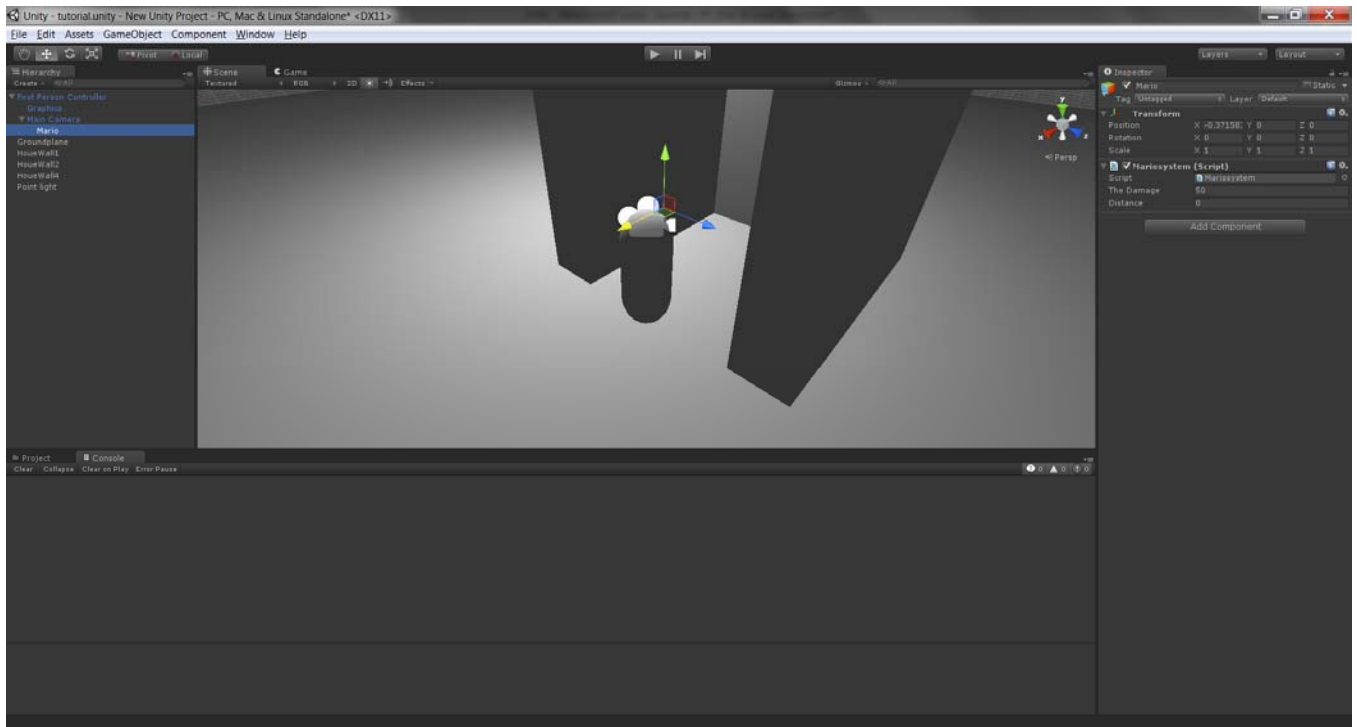


23) Cast a line from an empty object. When it hits an object given some information.

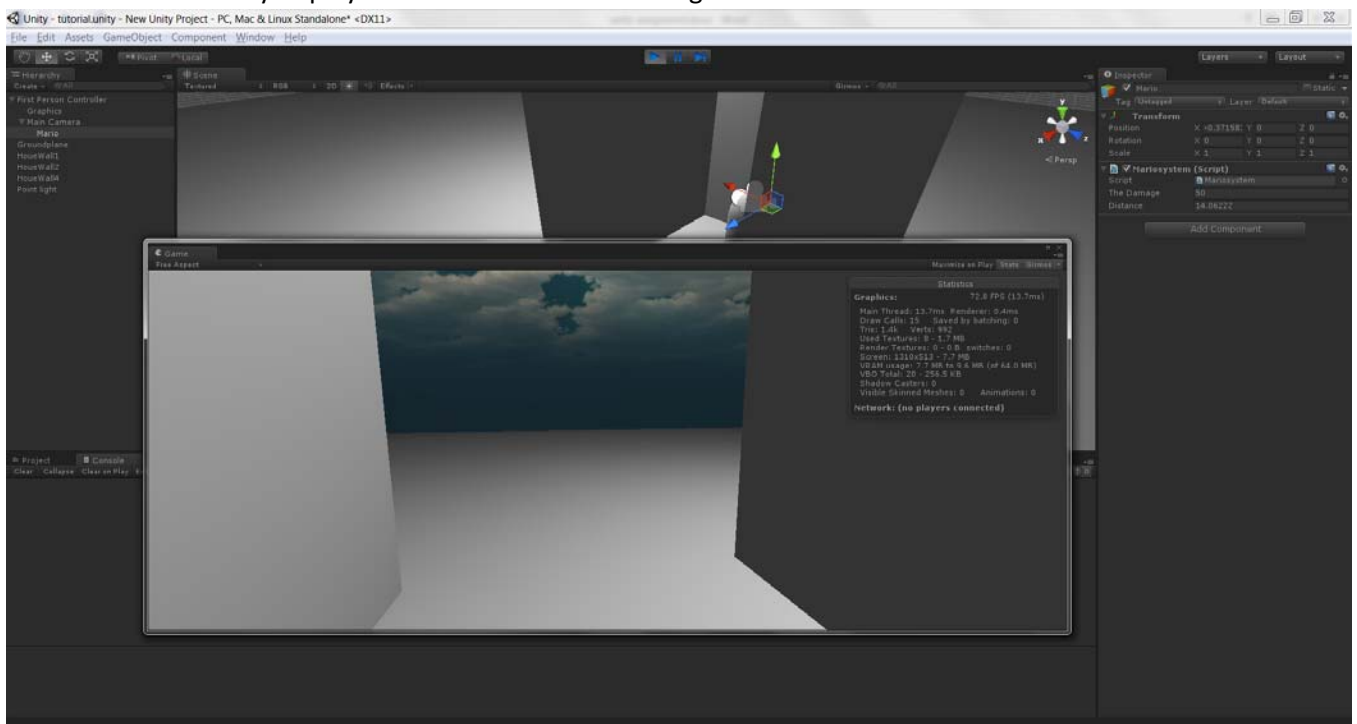
24) Add the following script

```

#pragma strict
var TheDamage : int = 50;
var Distance : float;
function Update()
{
    if (Input.GetButtonDown("Fire1"))
    {
        var hit : RaycastHit;
        if (Physics.Raycast (transform.position, transform.TransformDirection(Vector3.forward), hit))
        {
            Distance = hit.distance;
            hit.transform.SendMessage("ApplyDamage",TheDamage, SendMessageOptions.DontRequireReceiver);
        }
    }
}
    
```



25) You would see when you play. The distance variable changes when mouse is clicked.



26) Update the script [Fire1 is default button for shooting]

```
#pragma strict
```

```
var TheDamage : int = 50;
```

```
var Distance : float;
```

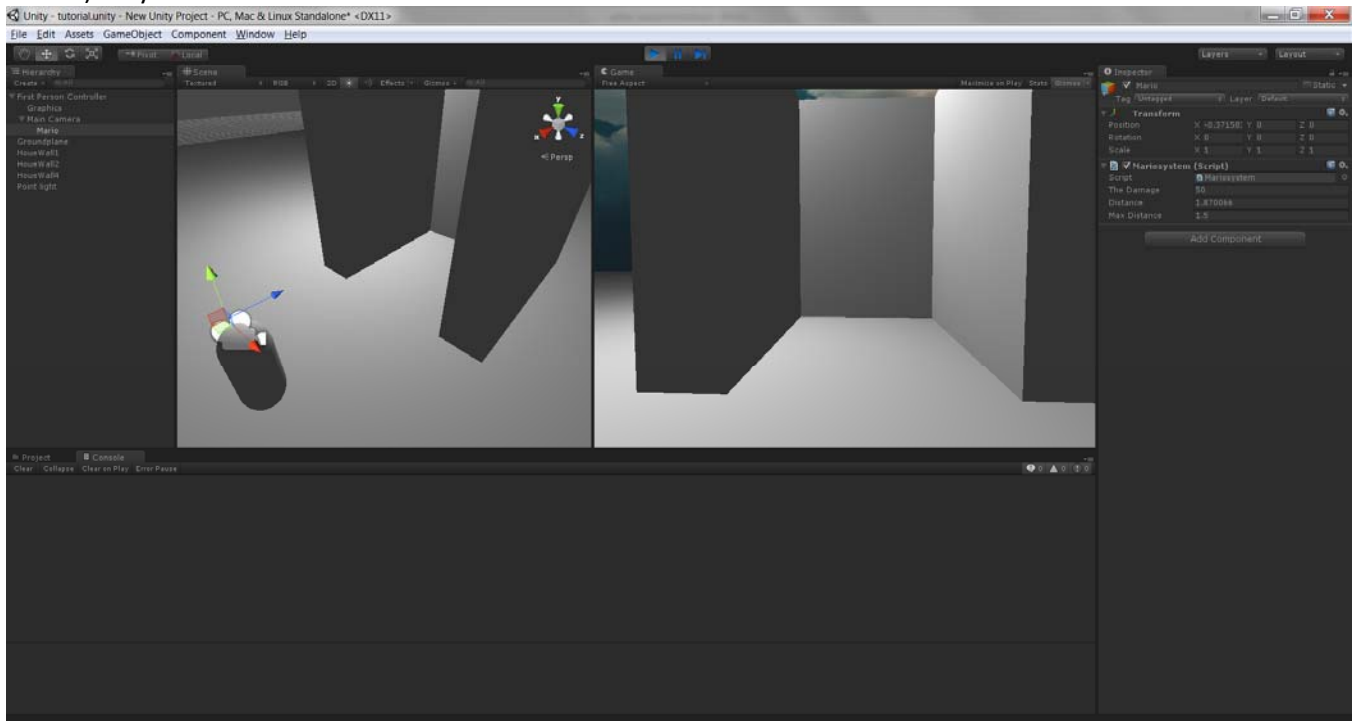
```
var MaxDistance : float = 1.5;
```

```

function Update()
{
    if (Input.GetButtonDown("Fire1"))
    {
        var hit : RaycastHit;
        if (Physics.Raycast (transform.position, transform.TransformDirection(Vector3.forward), hit))
        {
            Distance = hit.distance;
            if (Distance < MaxDistance)
            {
                hit.transform.SendMessage("ApplyDamage",TheDamage,
SendMessageOptions.DontRequireReceiver);
            }
        }
    }
}

```

27) Play and shift the frames



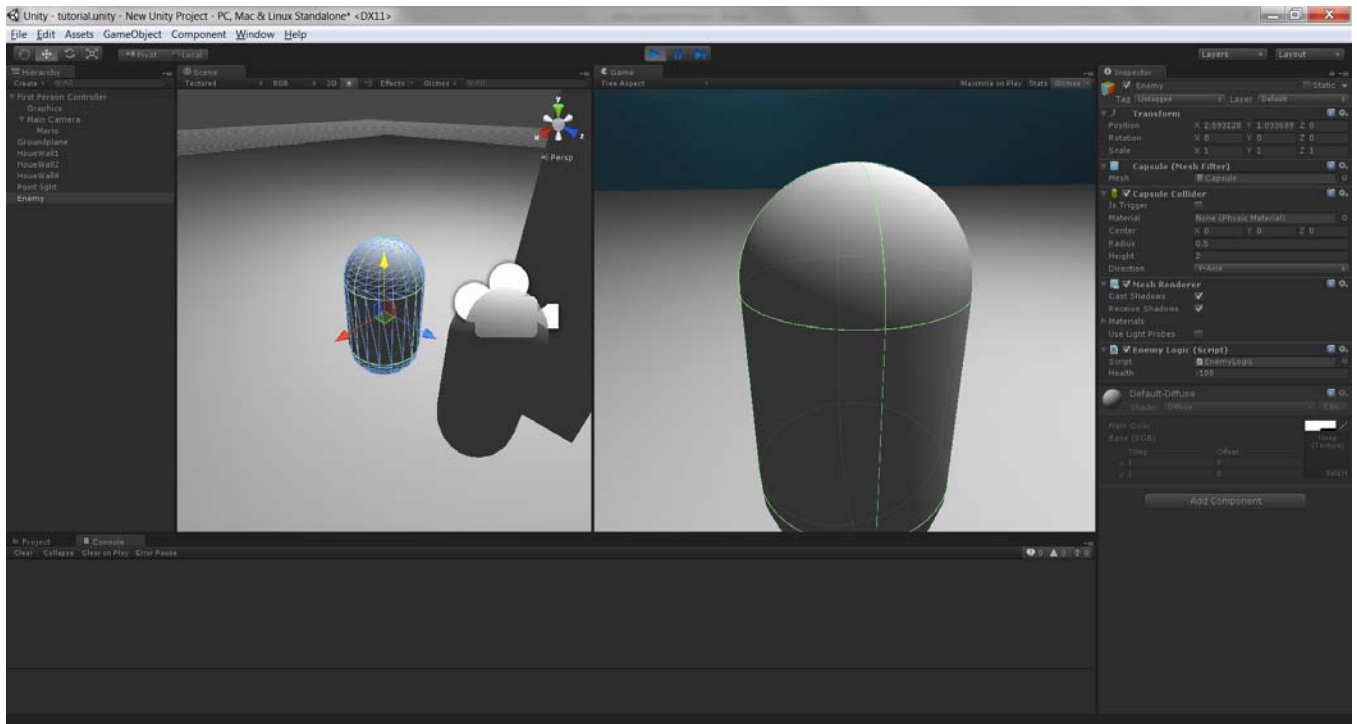
28) Now let's create an opponent. Go to create object-> create capsule. Put transform x,y,z to zero(0). Rename the capsule to enemy. Drag it above the plane.

29) Addcomponnt -> createnewscript->enemy logic

```

#pragma strict
var Health = 100;
function ApplyDamage(TheDamage : int)
{
    Health = Health - TheDamage;
}

```



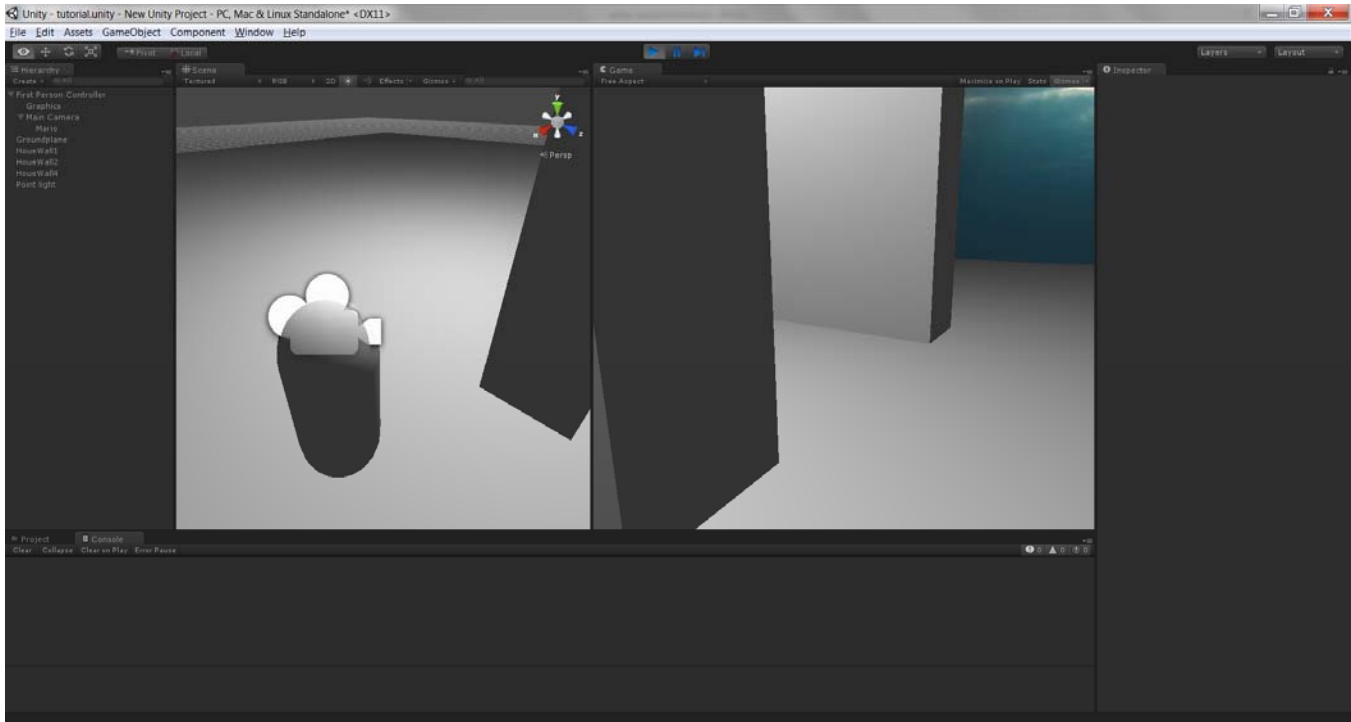
30) When the player is brought closer to enemy and the mouse button is pressed Health variable decreases by 50 [every click]

31) Update the script enemy logic to

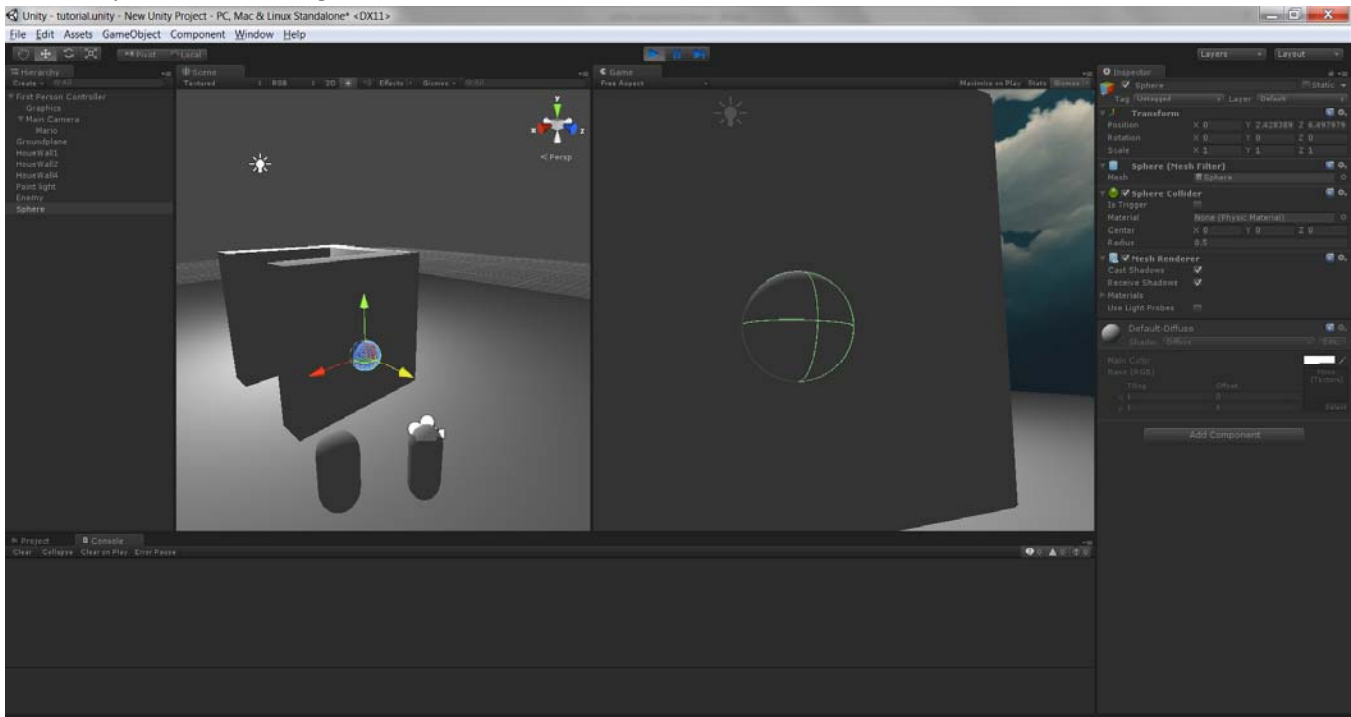
```
#pragma strict
var Health = 100;
function Update ()
{
    if (Health <=0)
    {
        Dead();
    }
}
function ApplyDamage(TheDamage :int)
{
    Health = Health - TheDamage;
}

function Dead ()
{
    Destroy (gameObject);
}
```

32) Game object (enemy) disappears when mouse is clicked close to it.

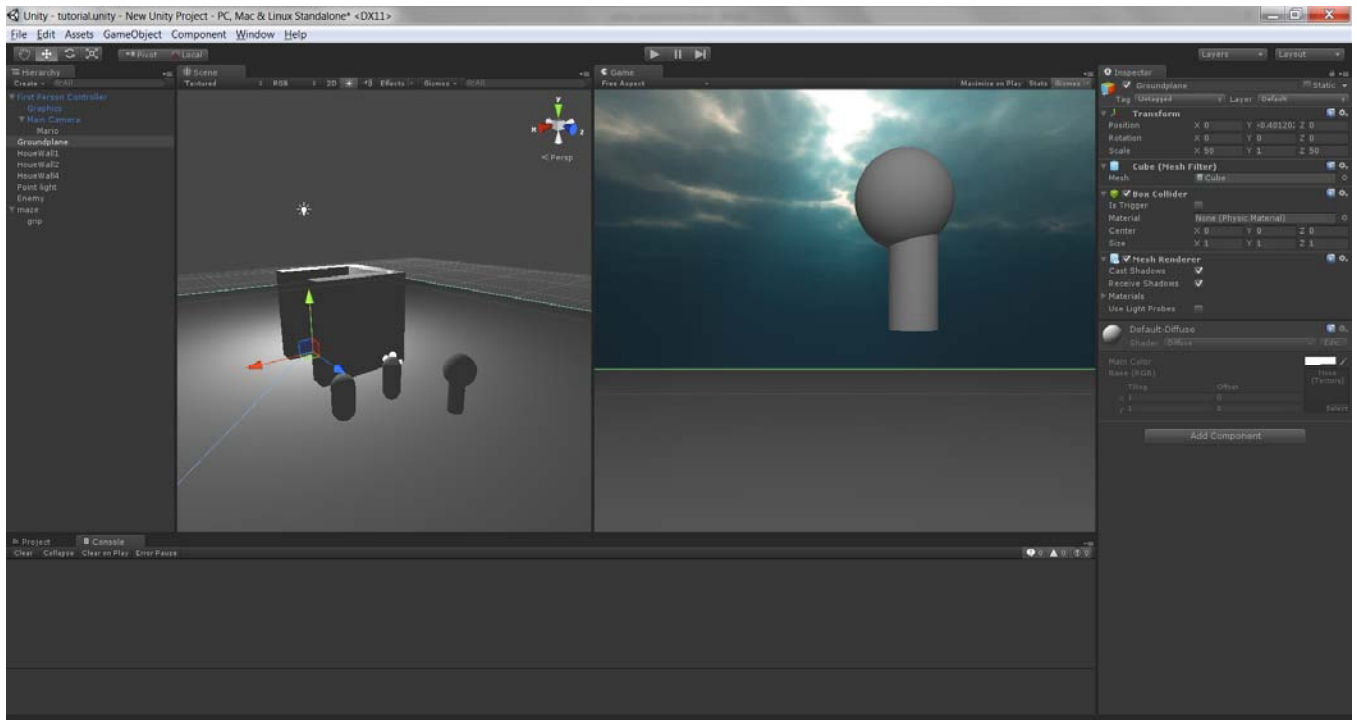


33) Animation and Modeling: Add gameobject->Sphere. Put zero values in transform and then drag the sphere above the ground.

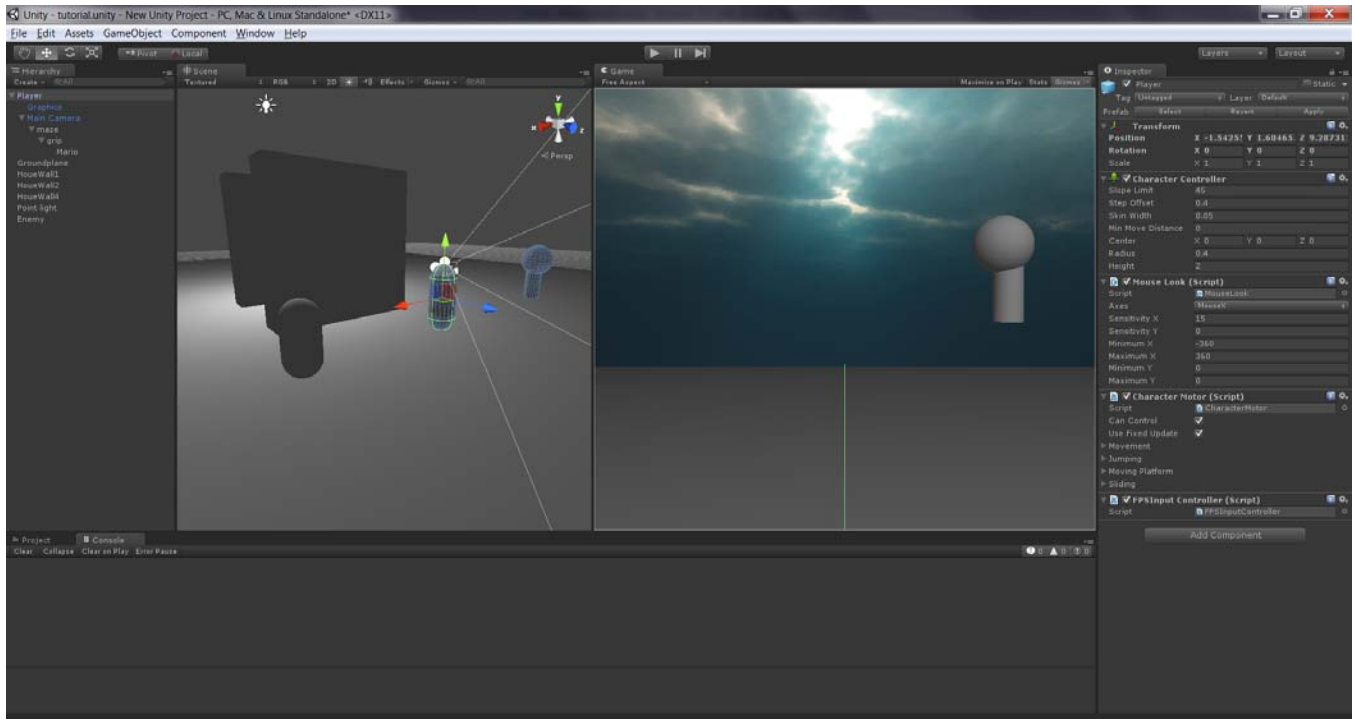


34) Create a cylinder and make it a child of sphere. Zero out the transform values of x,y,z. This will result in sphere and cylinder in center. Rename cylinder to grip and sphere to maze.

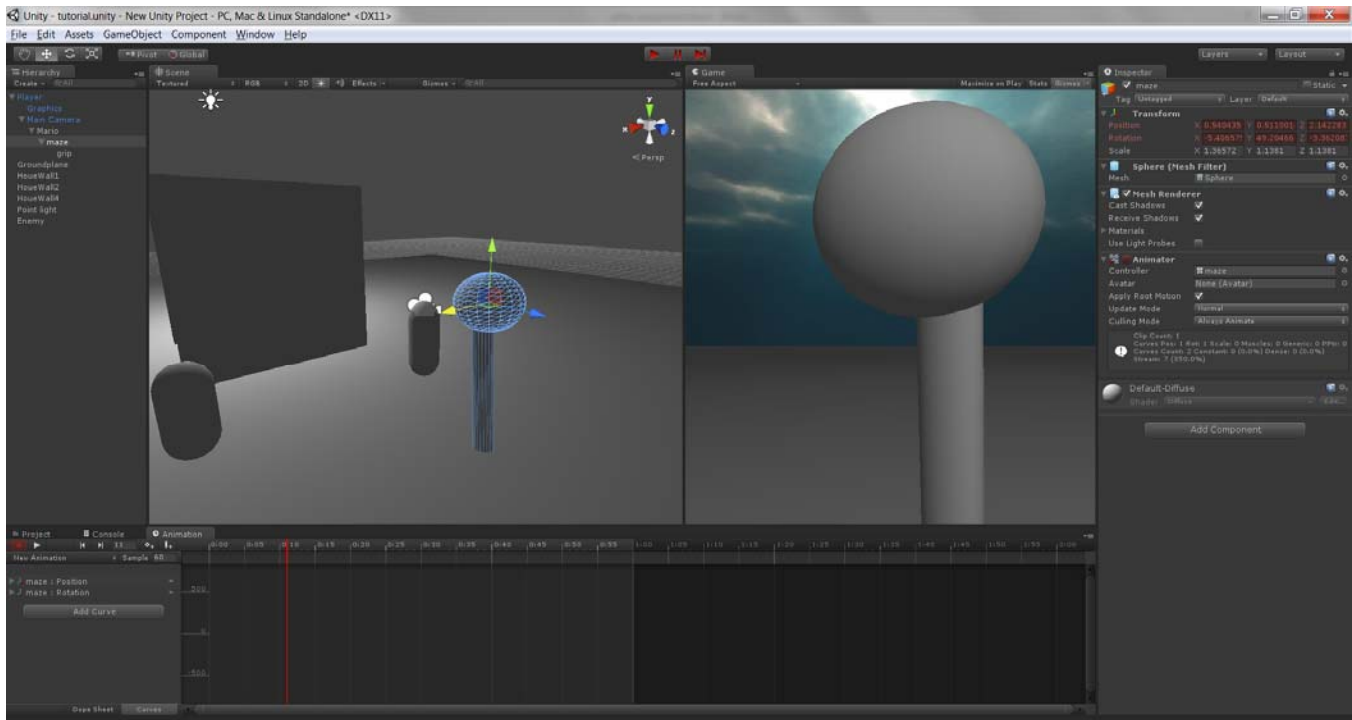
35) Move it in front of the camera



- 36) Remove sphere collider in maze object. Right click sphere collider and remove component. Also remove capsule collider. Now the maze can pass through objects.
- 37) Now drop the maze on to the player. Then on to the camera. This will make the maze follow the player.



- 38) Go to window->animation
- 39) Click on add curve and click key on maze translation X and rotation X. Create a "New Animation" file when prompted.
- 40) Now move the red vertical ribbon in animation to 0.10. Then move and rotate the maze in front of the camera.

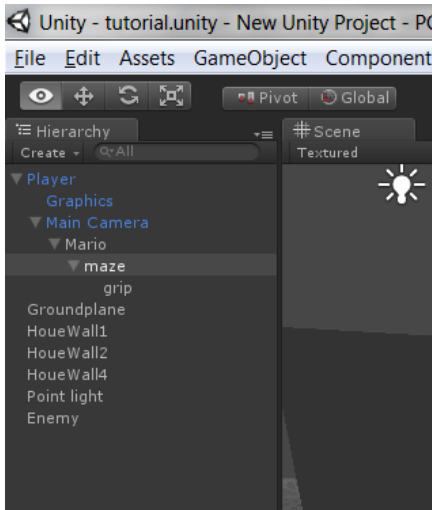


- 41) You can view the animation by moving the vertical line. Click the red circle button in animation to stop the animation.
- 42) Lets go to project. Now maze (one left frame) has animation component applied to it (on right frame).
- 43) Now we need to play this animation when we attack. Lets go back to the MarioSystem script and update it.

```
#pragma strict
var TheDamage :int = 50;
var Distance :float;
var MaxDistance : float = 1.5;
var TheMaze : Transform;

function Update()
{
    if (Input.GetButtonDown("Fire1"))
    {
        TheMaze.animation.Play("New Animation");
        var hit : RaycastHit;
        if (Physics.Raycast (transform.position, transform.TransformDirection(Vector3.forward), hit))
        {
            Distance = hit.distance;
            if (Distance < MaxDistance)
            {
                hit.transform.SendMessage("ApplyDamage",TheDamage,
                SendMessageOptions.DontRequireReceiver);
            }
        }
    }
}
```

- 44) Now drag Maze into Mario object



45) We have got a New variable “TheMaze”. It says NONE Transform [on right frame]. Drag and drop maze object from left window to TheMaze variable in right window. {NOTE: We are saying that this object should play the animation}



46) Now play to go to game mode. You will see the animation playing when the player is moving.

References

Video by Brackeys: <https://www.youtube.com/watch?v=FKT7jIW6ORU>

Video by Brackeys: <https://www.youtube.com/watch?v=HzTceINFowY>

Video by Brackeys: https://www.youtube.com/watch?v=fRED_-LvJKQ

<https://www.youtube.com/watch?v=aTnBAzin9vE>

<https://www.youtube.com/user/Brackeys/videos>

<https://www.youtube.com/watch?v=ICPMbWVlq3c>