# Workshop 5 Lab

In this activity, we are going to implement the classic Tower Defence Game in 3D using 3D. This is a single player game. The aim of the game is to protect the Castle that is attacked by the Monster.

Monster will be placed in the game scene and will move towards the Castle to attack it. Each Monster hit on the Castle will reduce the Health level by 1.

The player can click on the Tower Defence Cube to build a Tower. The Tower will launch bullet to shoot the Monster when a Monster moves near a Tower.

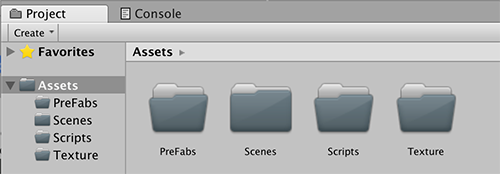
**Procedure:**

Create a new Unity Project “Tower Defence Game” with 3D template.

Change the name of SampleScene to GameScene.

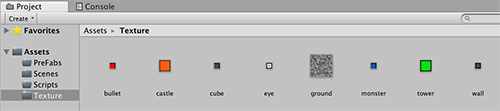
Create project folders.

Create the following project folders in Assets.



Prepare project images and texture

Drag and drop all the images in the resources folder into Texture folder.



Main camera setting

Select the Main Camera and set the background colour to black and the Position and Rotation as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 18 | -23 |
| Rotation | 45 | 0 | 0 |

Directional light setting Select Directional light.

Set the Directional light to the following to ensure that the light shines onto the game scene.

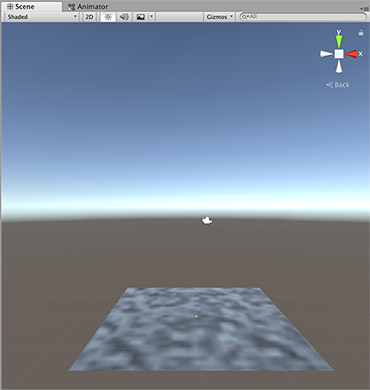
|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 0 | 0 |
| Rotation | 45 | 25 | 12 |

|  |  |  |
| --- | --- | --- |
| Intensity | 0.6 |  |
| Shadow Type | Hard shadows |

Add game scene Ground

Add a new GameObject->3D Object->Plane. Change the name to Ground. Change the following setting:

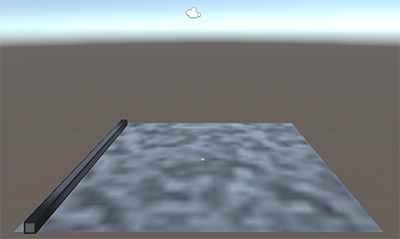
|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 0 | 0.75 |
| Scale | 3.1 | 1 | 3.1 |

From the Texture folder, drag and drop ground image on the Plane Ground. A Materials folder will be created in the Texture folder.

The four walls

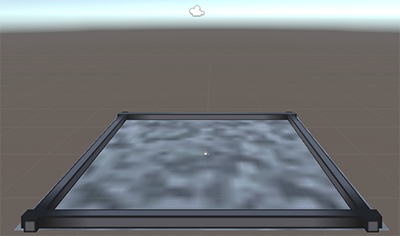
Select GameObject->Empty Object and name it Walls. Make sure the Transform Position of X, Y, Z are all 0. Under Walls create 3D Object->Cube and name it LeftWall. Change the following setting:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | -14 | 0.5 | 0 |
| Scale | 1 | 1 | -30 |



From the Texture folder, drag and drop Wall image on to the LeftWall.

Add another three walls (RightWall, FrontWall and BackWall) around the Ground.

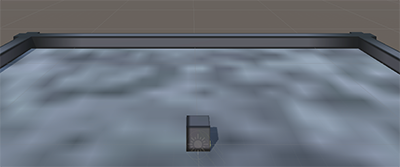


The Tower Defence Cube

Select GameObject->Empty Object and name it TowerDefenceCubes.

Under TowerDefenceCubes create 3D Object->Cube and name it TDCube. Change the following setting:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 0.5 | 0 |

From the Texture folder, drag and drop the cube image onto the TDCube to create a texture.

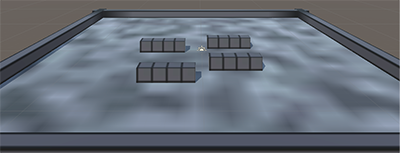
Next create a Script to build a Tower (which can defence the castle later) on top of the TDCube when the player clicks on the TDCube. In the Scripts folder, create a BuildTower C# script.

Type in the following codes:



Attached this Script to the TDCube. (Note: We will attach Game Object towerPrefab later.)

Create more tower defence cubes

Duplicate the TDCube in the game scene and change the Position (X and Z) to create something similar. You may create more tower defence cubes.

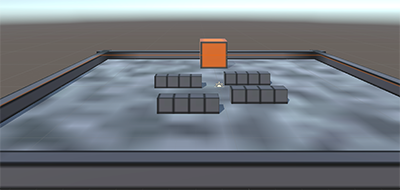
The Castle cube

Next create a cube that represents the Castle that we are supposed to protect in this game.

Select GameObject->3D Objecct->Cube, rename it to Castle. Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 1.5 | 9 |
| Scale | 3 | 3 | 3 |

From the Texture folder, drag and drop the image castle onto the Castle cube.



The Health Bar

Next we add a heath bar to the castle.

Under Castle, create an empty Game Object and rename it to HealthBar.

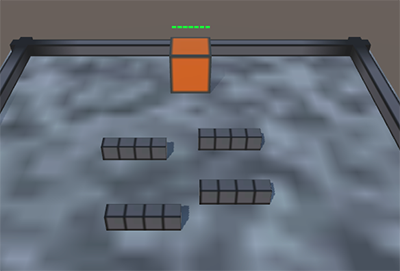
Go to the Inspector, select Add Component->Mesh->Text Mesh.

Position the HealthBar Text Mesh directly above the Castle by modifying the properties below.

Transform Position Y: 1

Text Mesh Text: -------

Character Size: 0.04 Anchor: Middle Centre Alignment: Centre

Font Size: 100 Font Style: Bold Font: Arial Colour: Green

Next create a Health script to

* Count and return the number of “-“ in the text (indication of health level).
* Remove a “-“ to indicate a decrement of health.
* Ensure the HealthBar always faces the Camera.

Type in the following codes:



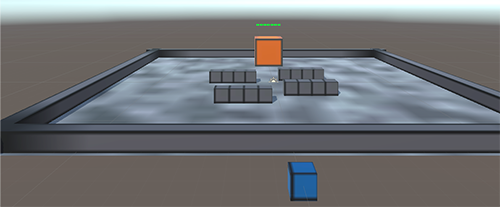
Attach this Health script to the HealthBar.

The Monster Cube

Next we create the enemy cube.

GameObject->3D Object->Cube and name it as Monster. Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 1.5 | -17 |

Note we are creating the monster outside the game scene and then we will move it to the PreFabs. We will use a Spawner to create many monsters in the game scene later. From the Texture folder, drag and drop the image monster on the Monster cube.

Under the Monster, create another two more cubes. Name them as LeftEye and RightEye.

EyeLeft: Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0.25 | 0.2 | 0.57 |
| Scale | 0.26 | 0.26 | 0.23 |

EyeRight: Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | -0.25 | 0.2 | 0.57 |
| Scale | 0.26 | 0.26 | 0.23 |

From the Texture folder, drag and drop the image eye on the two Monster eye cubes.



Monster Rigidbody

All game objects in Unity that move need to have a Rigidbody. A Rigidbody takes care of physics properties such as gravity, velocity and other forces.

Add a Rigidbody to the Monster:

Select the Monster, in the Inspector Add Component->Rigidbody. Set the Angular Drag to 0.

The Monsters should be able to run on the game scene ground, avoid the walls and tower defence cube and into the Castle to decrease the health level.

The above can be easily done by Unity.

* + Bake a Navigation Mesh (*which area is walkable*)
  + Add a NavMeshAgent to the Monster

Select Ground, Walls and TowerDefenceCubes and make it static in the Inspector. (including the children)

Select Window->AI->Navigation->Bake and set the following properties:

Agent Height: 0.5

Click on Bake button.

Select Monster, in the Inspector, Add Component->Nav Mesh Agent.

Change the following properties:

Base Offset: 0.69

Speed: 8

Acceleration: 3

Radius: 0.5

Height: 1.2

Add a Health Bar to the Monster.

The step is the same as adding Health bar to the Castle. Except the colour is red instead of green. Under Monster, create an empty Game Object and rename it to HealthBar.

Go to the Inspector, select Add Component->Text Mesh.

Position the HealthBar Text Mesh directly above the Monster by modifying the properties below.

Transform Position Y: 1

Text Mesh Text: -------

Character Size: 0.04 Anchor: Middle Centre Alignment: Centre Font Size: 100

Font Style: Bold Font: Arial Colour: Red



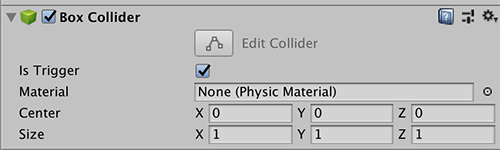
Attach the Health script to the Monster HealthBar.

Create a Monster Script such that it will move to the Castle and damage the Health level of the Castle by one.

In the Scripts folder, create a Monster C#script. Type in the following codes:



Note to call OnTriggerEnter method, in the Monster Box Collider enable “Is Trigger”.



Attach this script to the Monster game object.

Move Monster to PreFabs folder and delete Monster from the game scene.

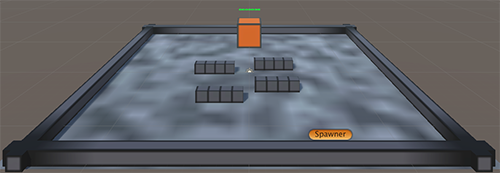
The Spawner

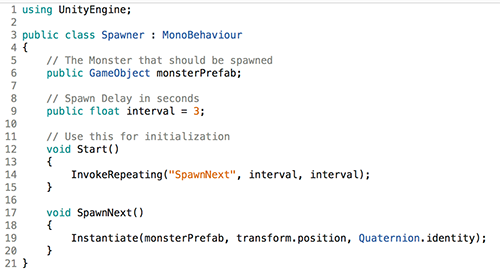
Create a Spawner to create Monsters every few seconds in the game scene. GameObject-

>Create Empty and rename it to Spawner.

In the Inspector, select a visual icon to represent the Spawner in the game scene. Change the following properties:

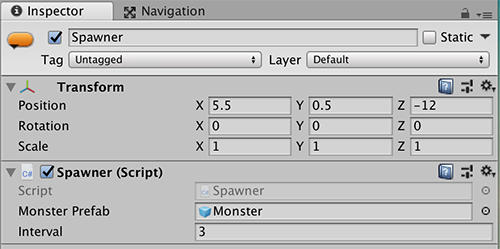
|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 5.5 | 0.5 | -12 |



In the Scripts folder, create a Spawner C# Script. Type in the following codes:

Attach this Script to the Spawner game object.

In the Inspector, attach the Monster in the PreFabs to the Monster Prefab in the Script.



Play and test out the play.

You should have monster being created every 3 seconds. The monster should be moving towards the Castle.

Each monster hit on the Castle should reduce its health level by one. When the health level is zero, the Castle will disappear.

The Bullet Cube

Next, we will create a Bullet for the Tower to shoot (note we have yet to create the Tower).

GameObject->3D Object->Cube, rename it to Bullet, enable isTrigger in the Box Collider.

Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 0 | 0 |
| Scale | 0.5 | 0.5 | 0.5 |

From the Texture folder, drag and drop the image bullet on the Bullet cube.

As the bullet needs to fly, we need to add a rigid body on the bullet cube.

Add Component->Rigidbody and disable gravity.

Create a Bullet C#script. Type in the following codes:



Attach this script to the Bullet.

Drag the Bullet to the PreFabs and delete it from the game scene.

The Tower Cube

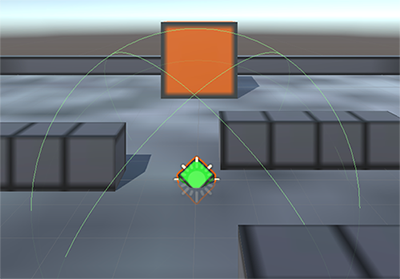
GameObject->3D Object->Cube, rename it to Tower.

Change the following properties:

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Y | Z |
| Position | 0 | 0 | 0 |
| Rotation | 45 | 0 | 45 |
| Scale | 0.5 | 0.5 | 0.5 |

From the Texture folder, drag and drop the image tower on the Tower cube. The Tower will detect if there is a monster nearby and launch the bullet.

To detect the Monster, we need to add a Sphere Collider to the Tower. First, remove the Box Collider and add a Sphere Collider.

Add Component->Sphere Collider and set Radius to 6. In the game scene, we can see the Sphere Collider outline

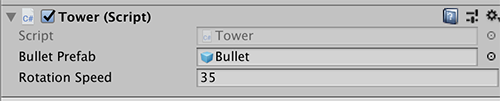
Create a Tower C# script.

Type in the following codes:



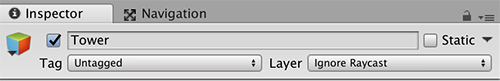
Attach the script to the Tower game object.

Attach the Bullet in the PreFabs folder to the Bullet Prefab property.

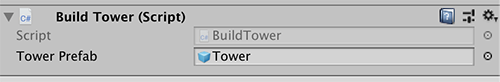


The Tower Sphere Collider overlap the Tower Defence Cube which will prevent OnMouseUpAsButton function from being called.

To overcome this, in the Inspector of Tower cube, select Ignore Raycast Layer.



Drag Tower to PreFabs folder and delete it from the game scene.

Finally, select all the TDCubes. In the Build Tower script, attach Tower to the Tower Prefab property.

Play the game. Click on each Tower Defence Cube to build the Tower.