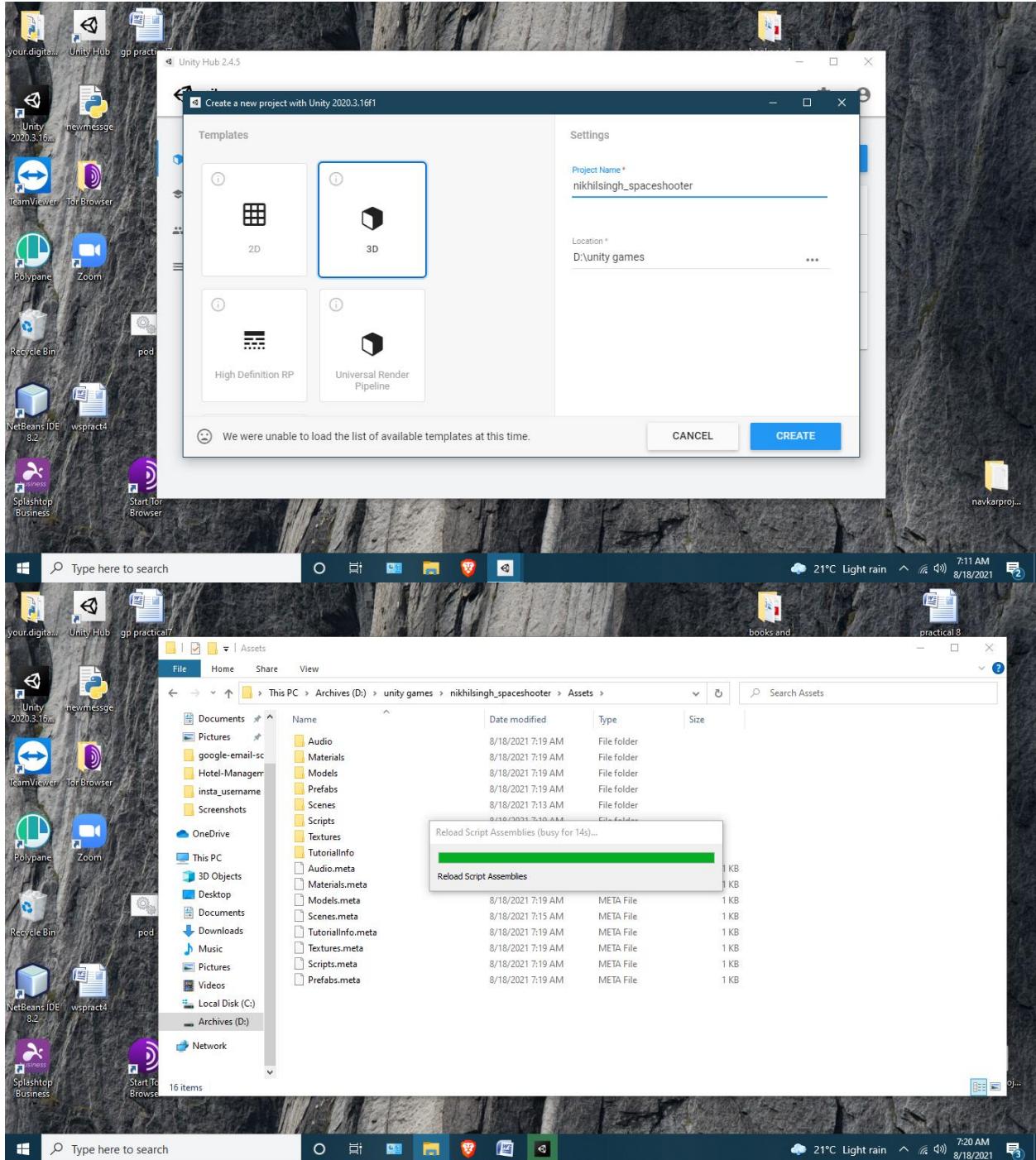


## Practical 8

### Step 1:

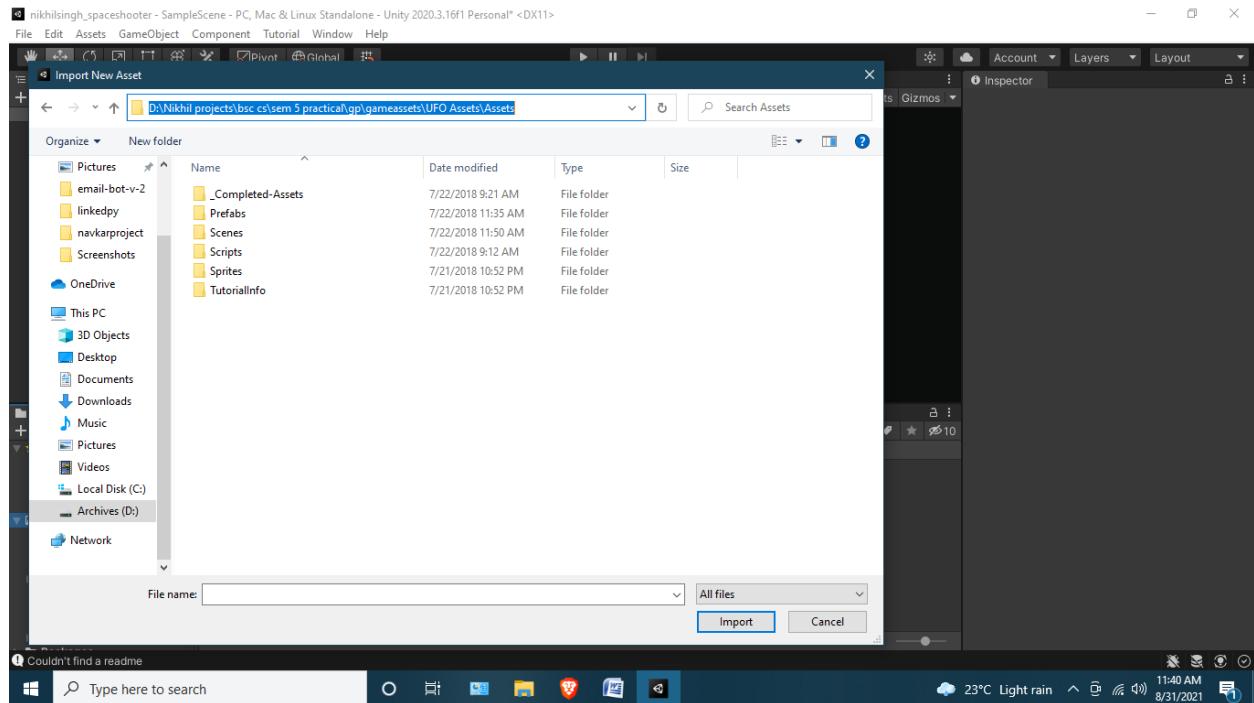
Create new 3d project name it space shooter



## Practical 8

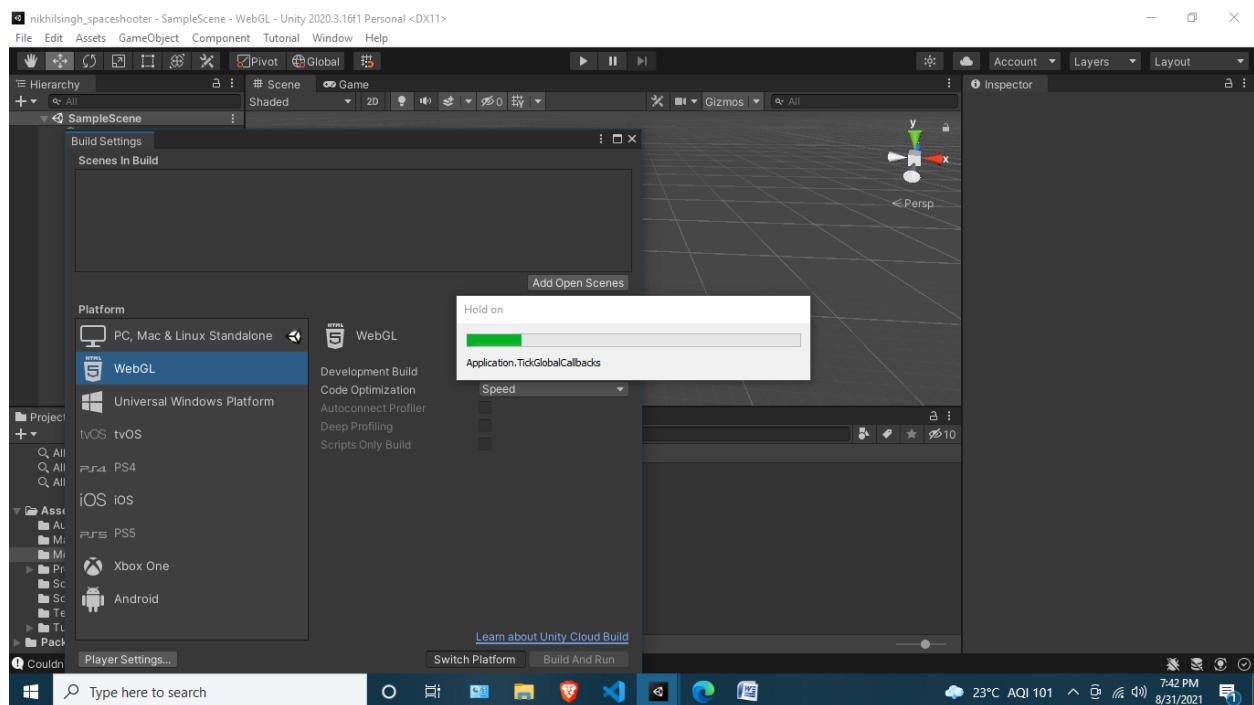
### Step 2:

After your project is created go ahead and import the assets required



### Step 3:

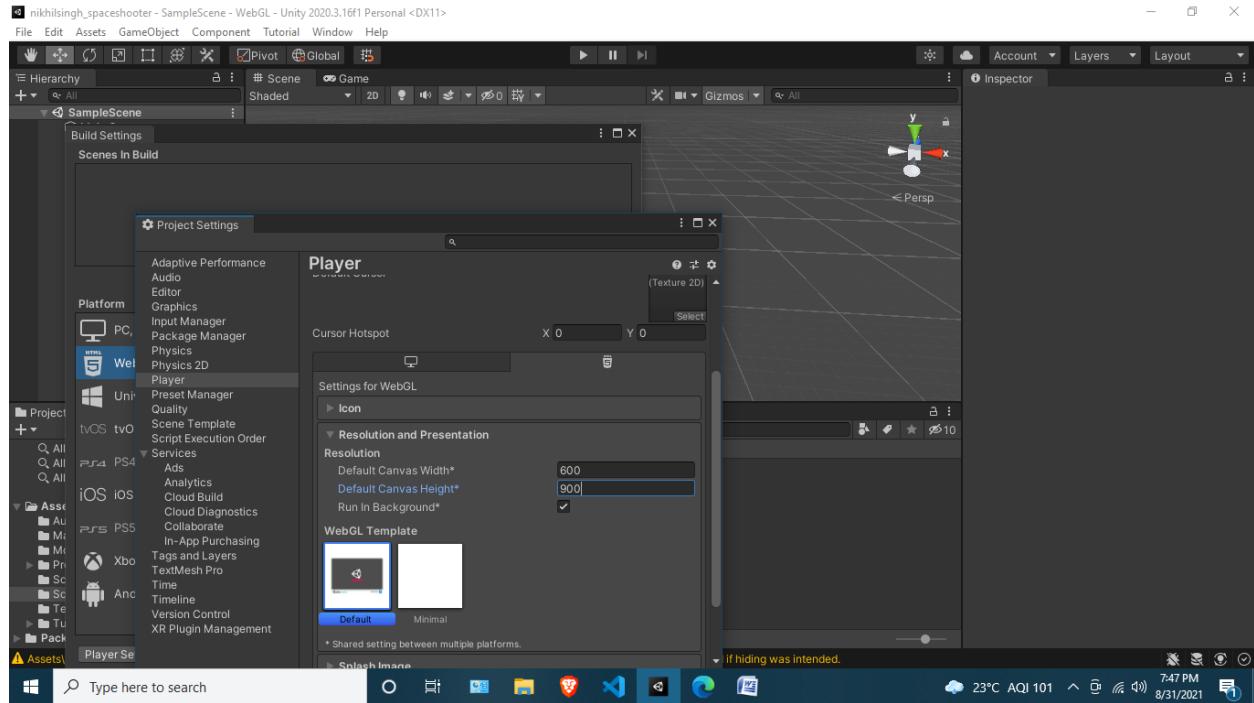
change the build setting of the project to webgl



## Practical 8

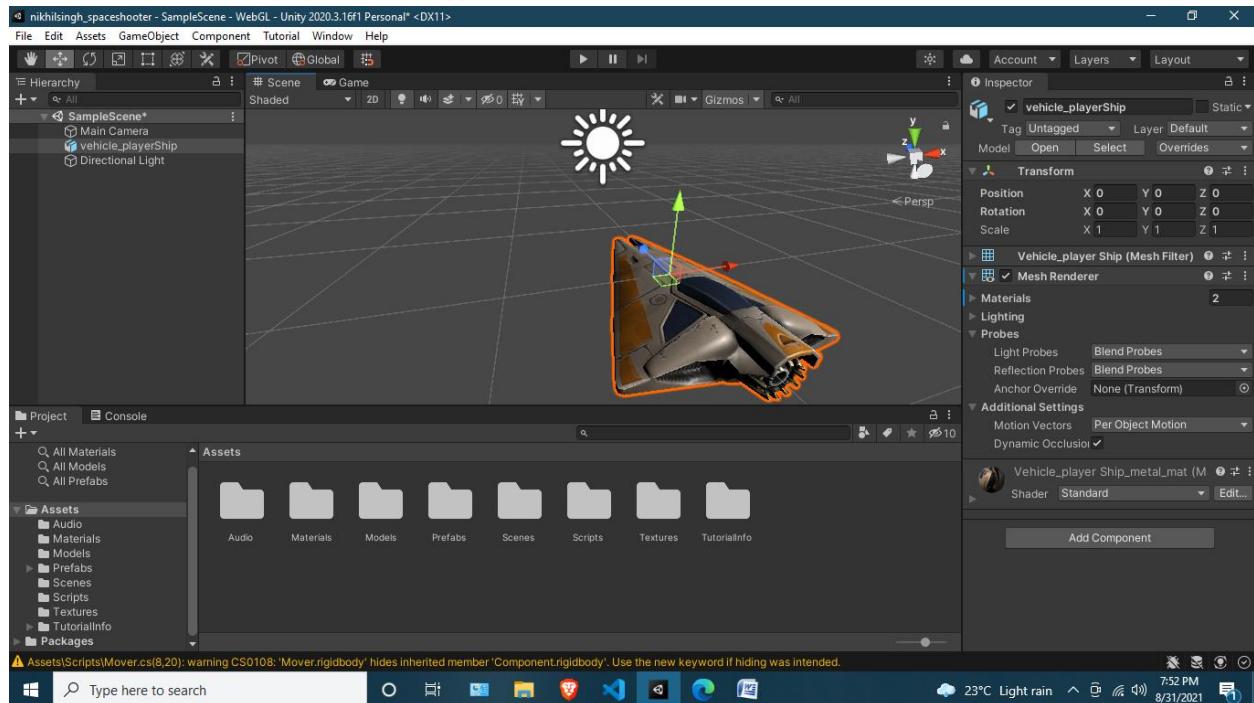
### Step 5:

change the player resolution to 600x900



### Step 6:

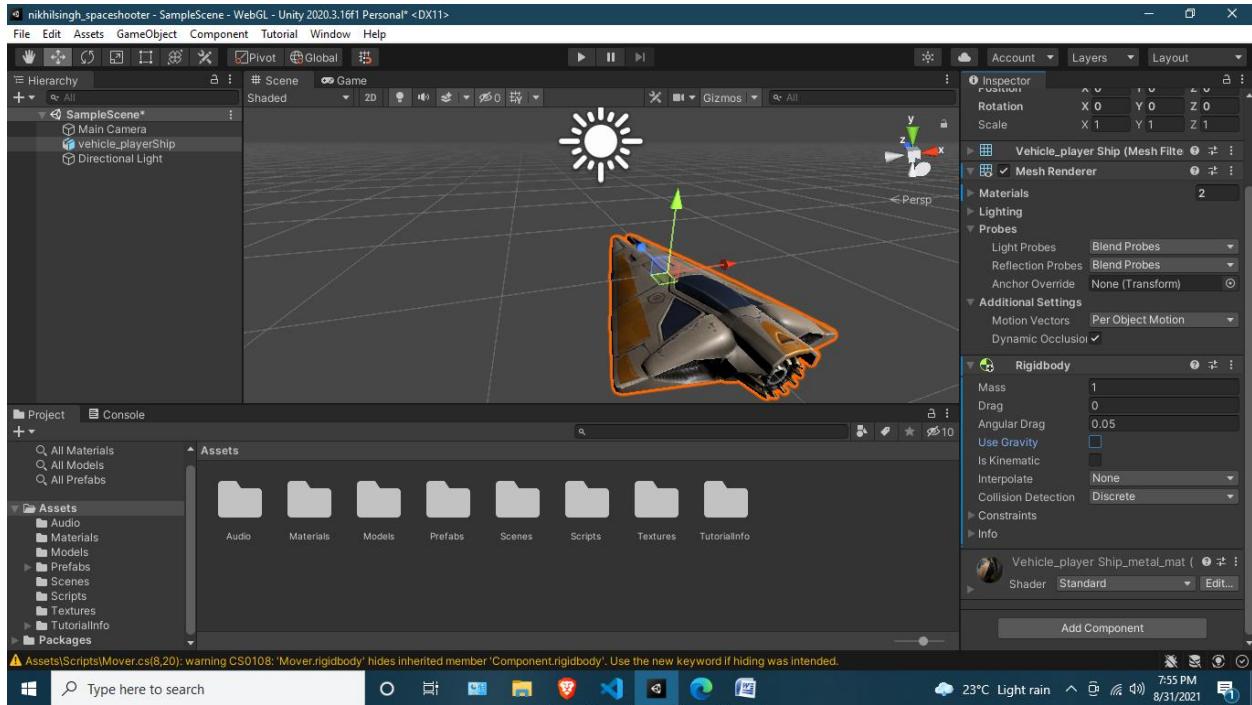
Drag and drop vehicle\_player model into hierarchy and rename it to Player and reset its transform



## Practical 8

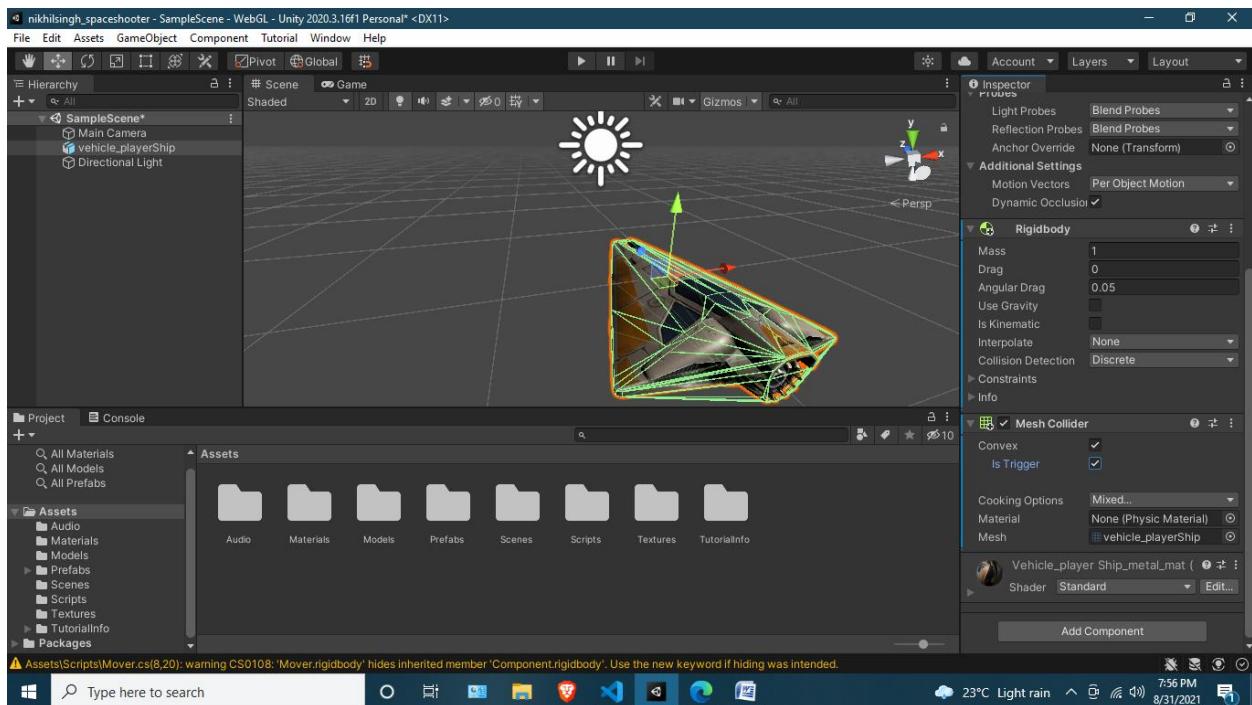
### Step 7:

Add rigidbody component to player gameobject deselect USE GRAVITY option



### Step 8:

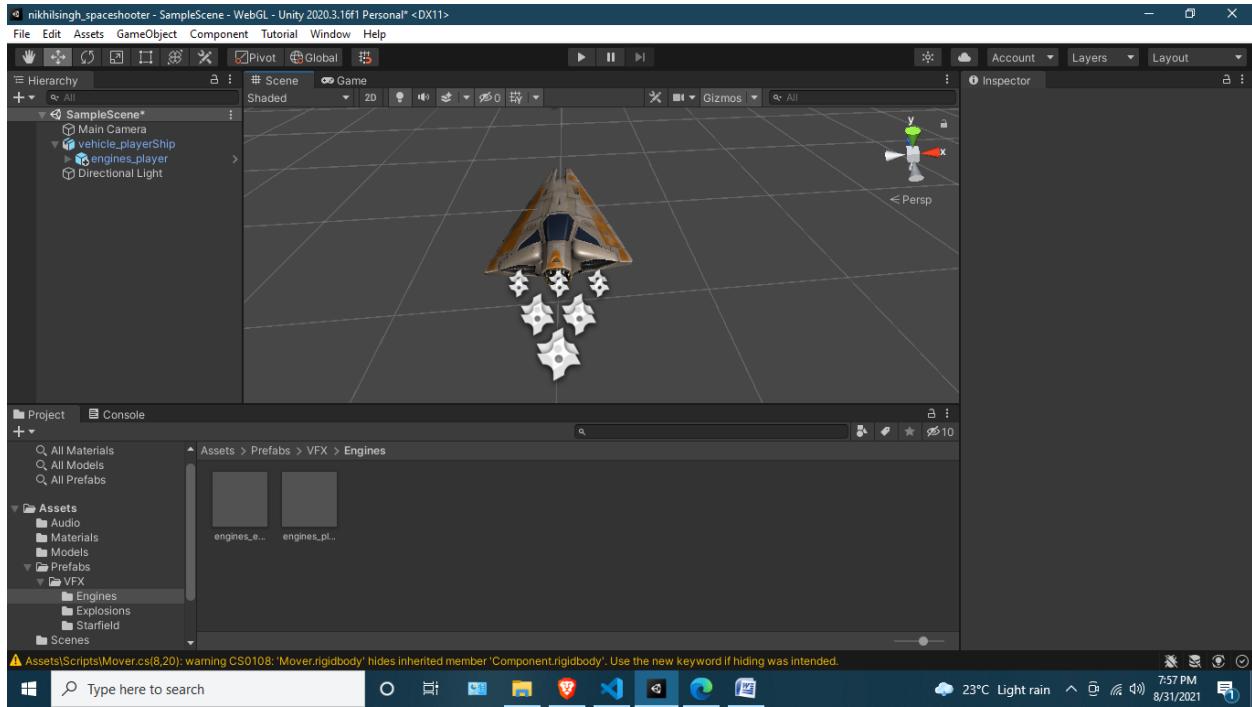
Add mesh controller to player game object and select convex and istrigger options.



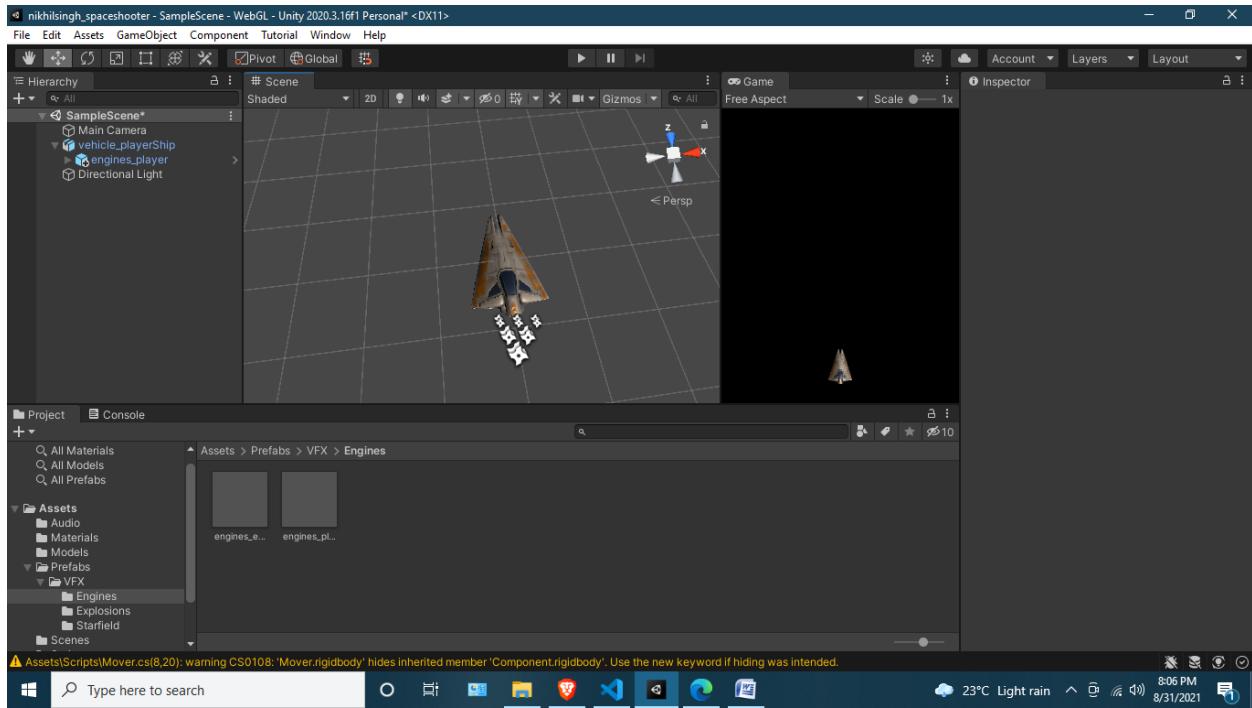
## Practical 8

### Step 9:

select enginer\_player from and drag and drop it as a child of Player. Adjust its size accordingly



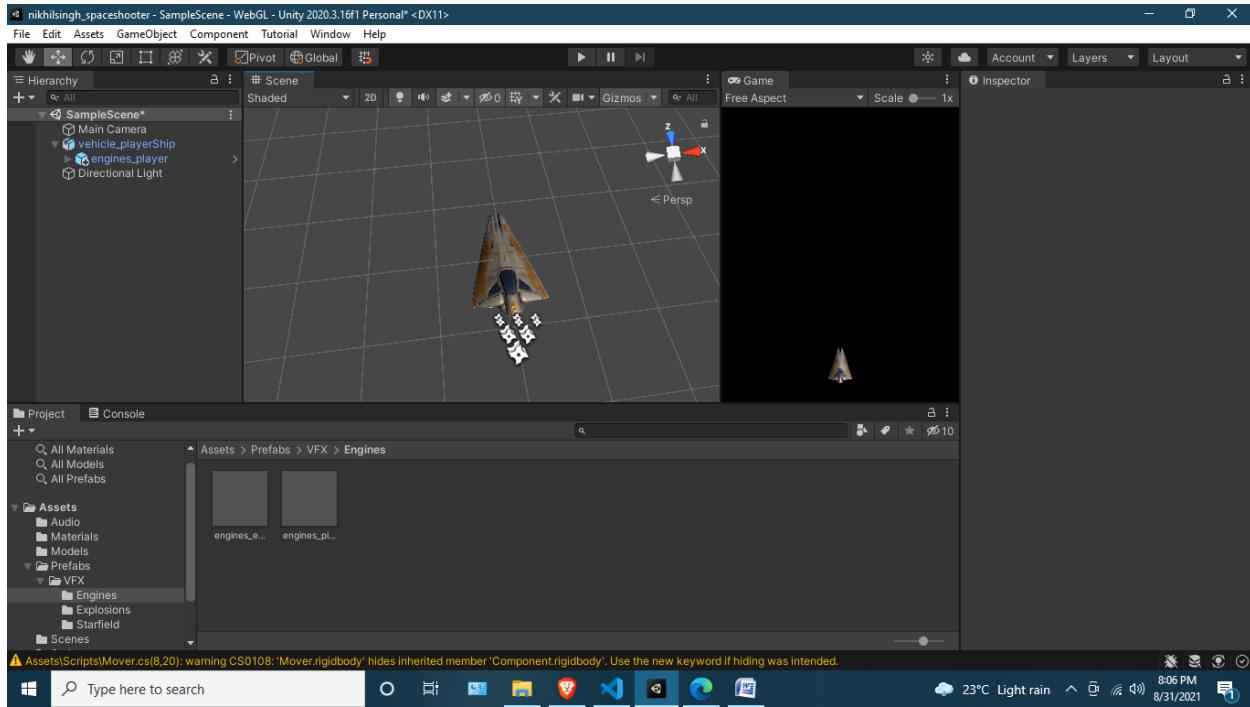
Step 10: In the hierarchy select Main Camera and reset its Transforms. Set Rotation X=90 and Position Y=10, Z=8.25. Set the Projection as Orthographic and size=10.



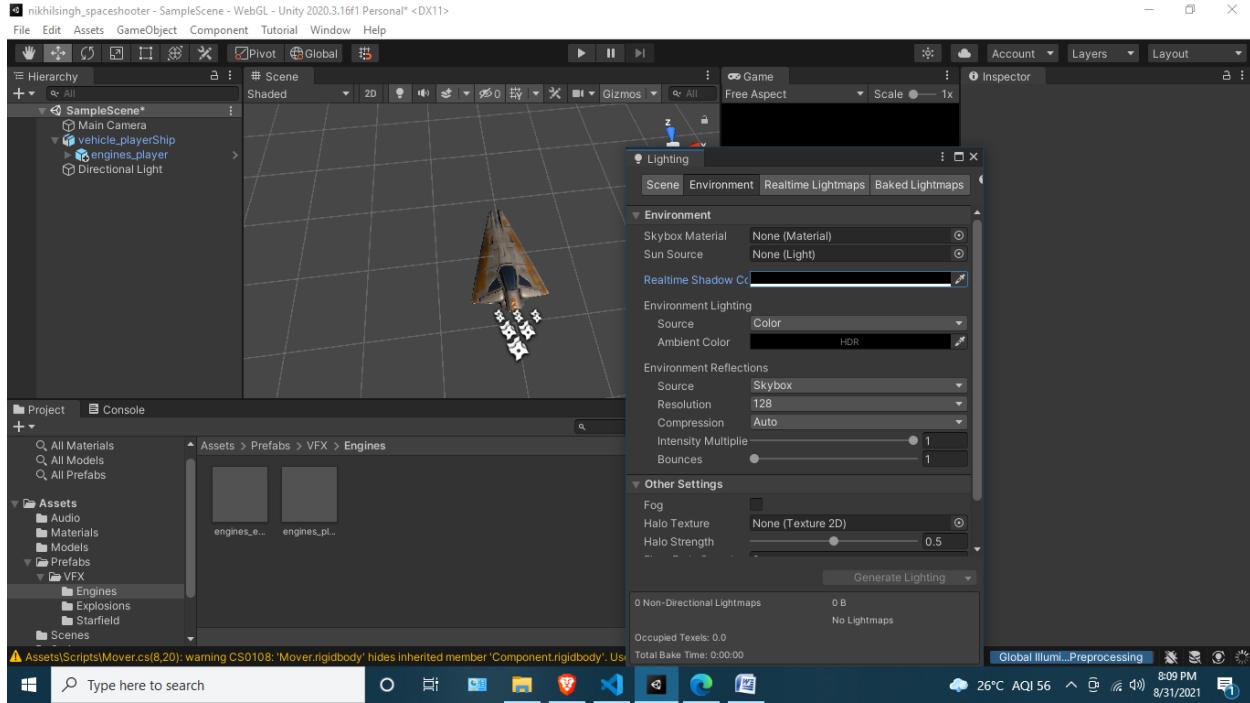
## Practical 8

### Step 11:

Set the background to black

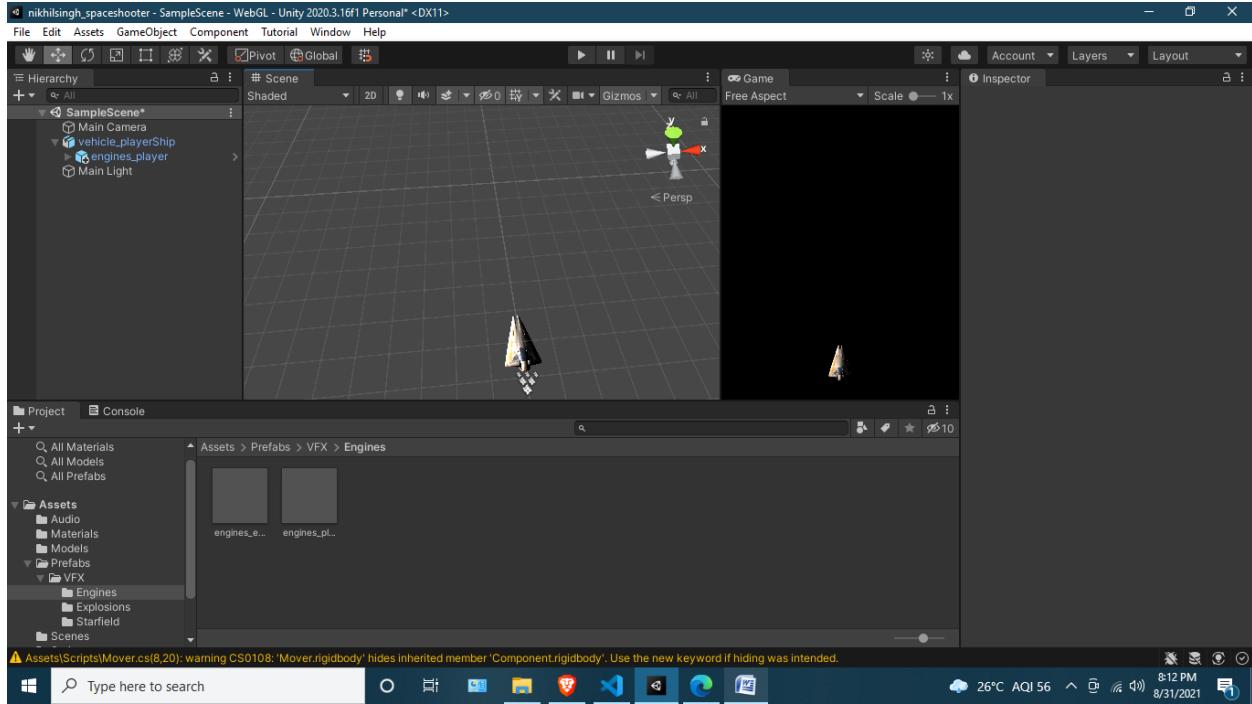


STEP 12: Go to Window – Rendering – Lighting. Select Environment, select Environment Lighting as Colour, Ambient colour as black, real-time Shadow colour as black:



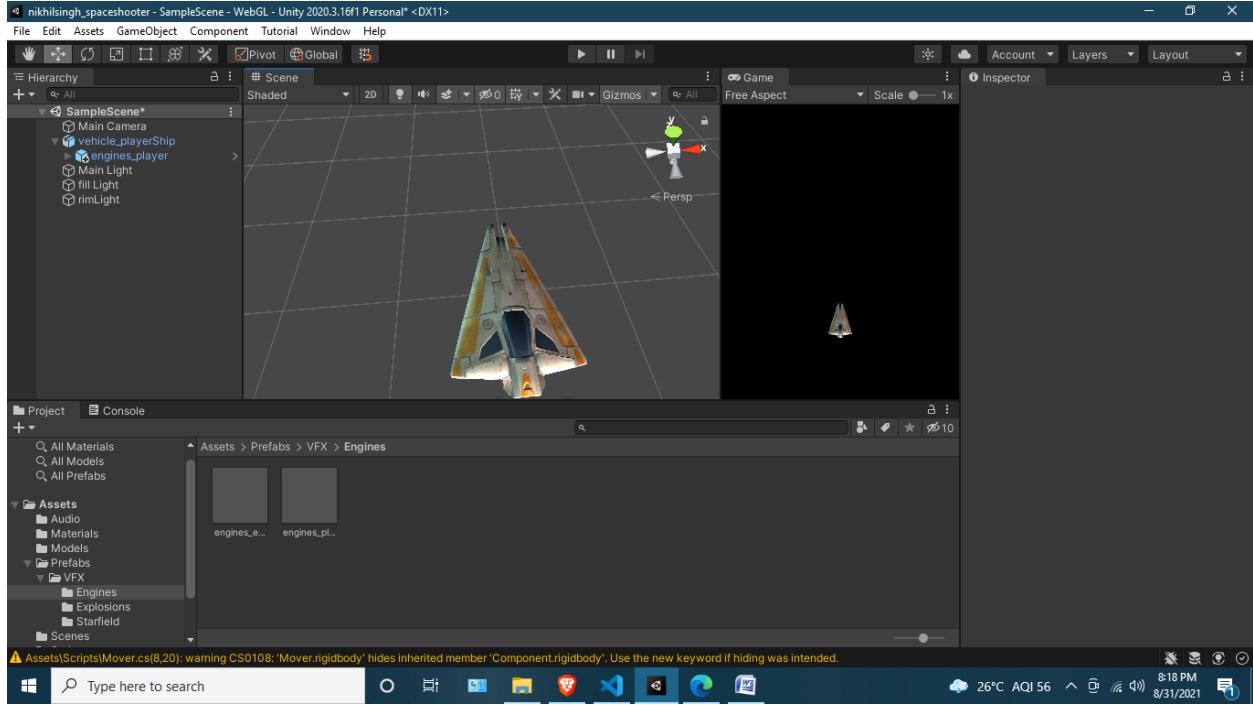
## Practical 8

Step 13: Delete Directional Lighting in hierarchy and create a new directional light by clicking on + - Light – Directional Light. Name it as Main Light. Reset its transforms. Set Rotation X=20; Y=-120, Intensity = 2

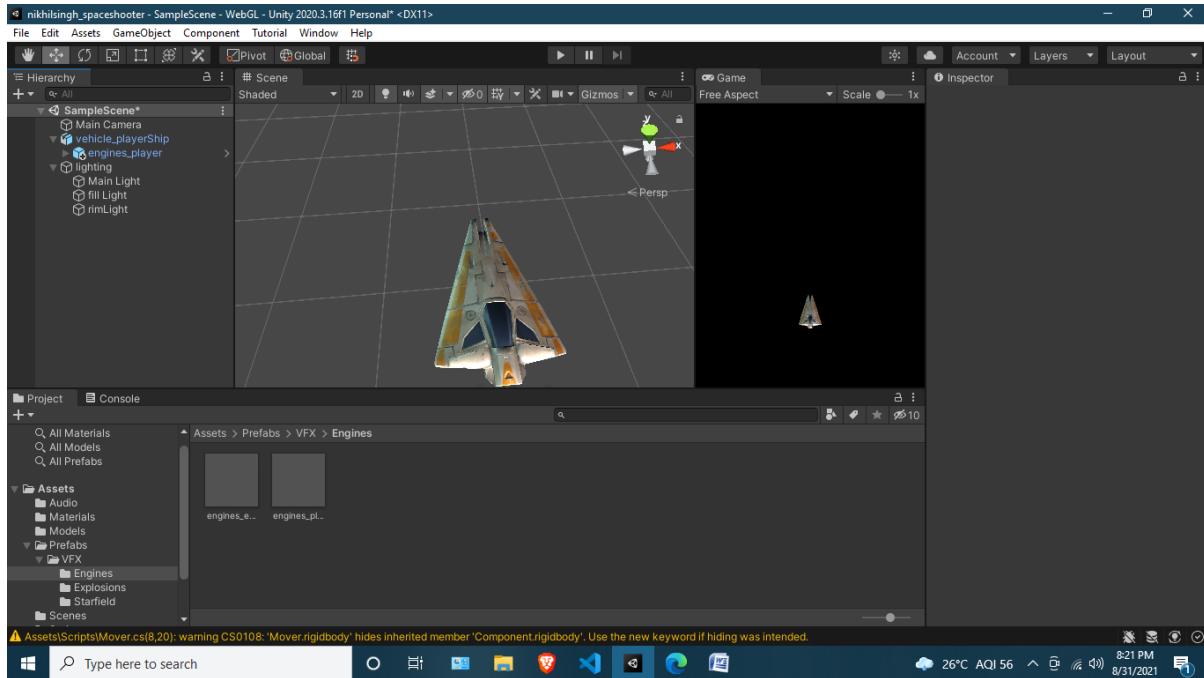


## Practical 8

Step 14: Duplicate the main light twice and name these lights as rim light and fill light respectively .Reset their transforms. For fill light Set Rotation X=5; Y=125 ,Intensity=1 and RGB of colour to R=128,G=192,B=192 A=255. For Rim Light set Rotation X=-15; Y=65, Intensity = 0.5 Change color to White

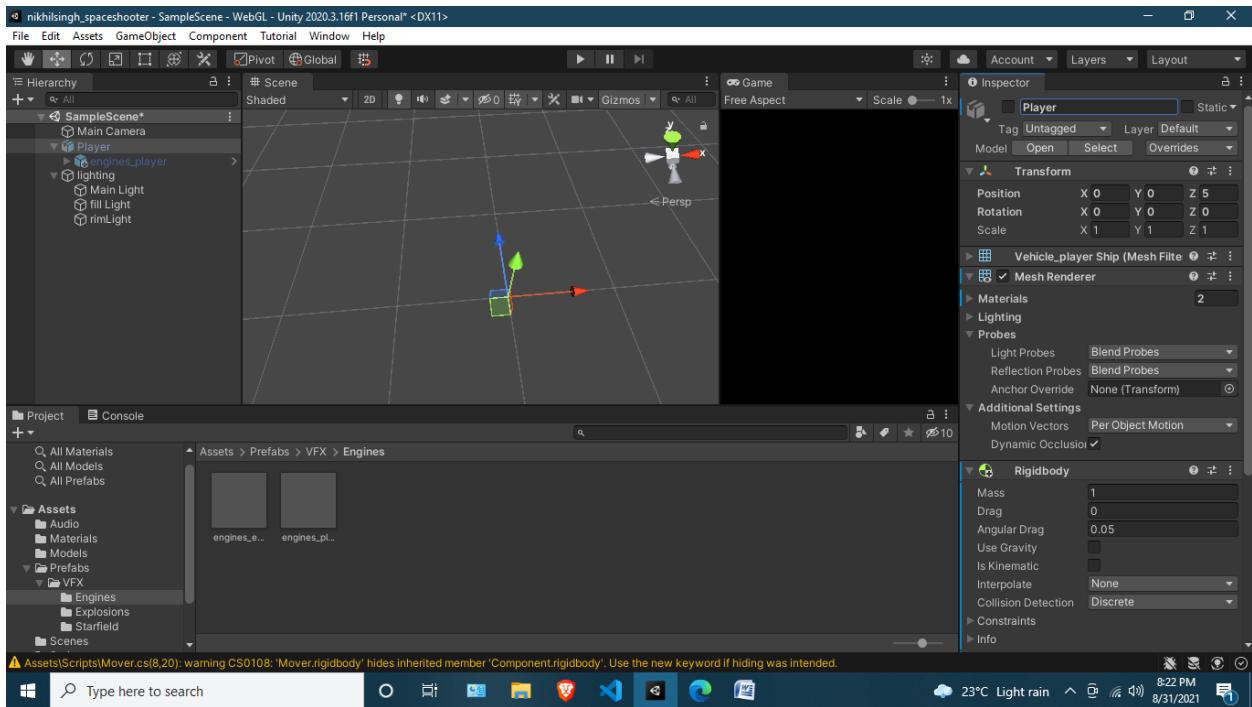


Step 15: Create a new Empty Game object name it lightning , reset its transform and set its position to Y=105 and drag and drop all lights in this empty game object .

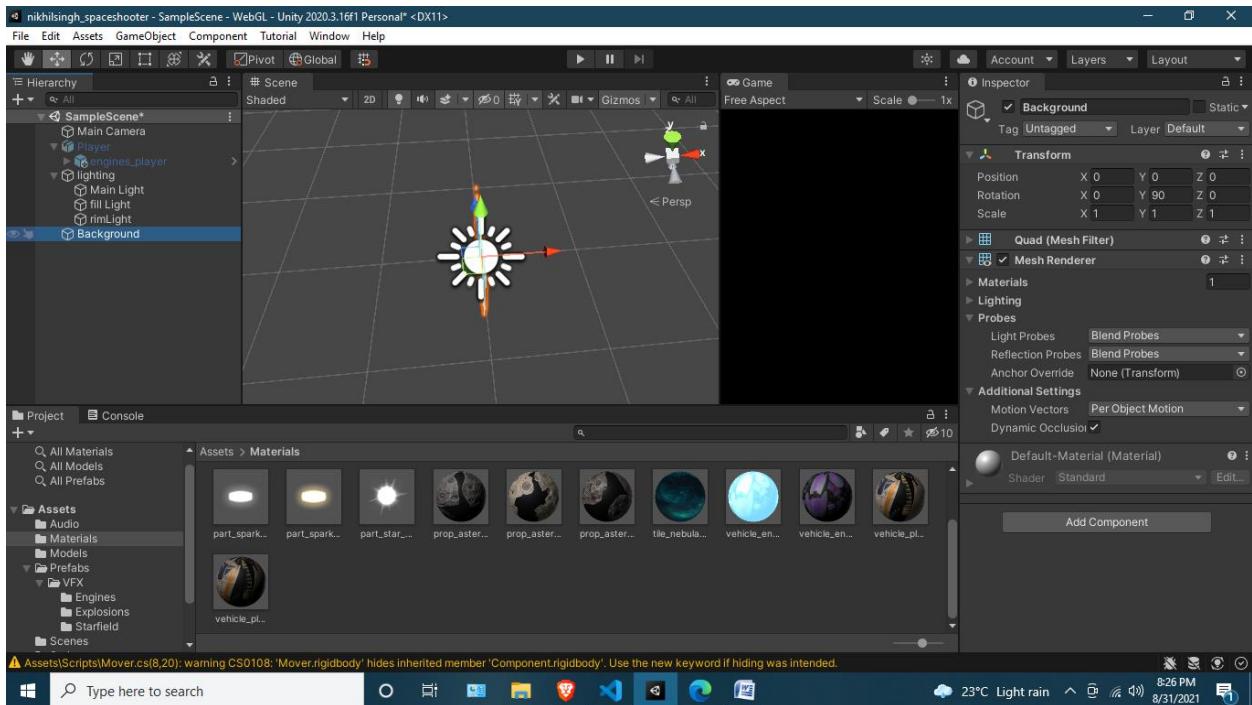


## Practical 8

Step 16: Deactivate player game object.

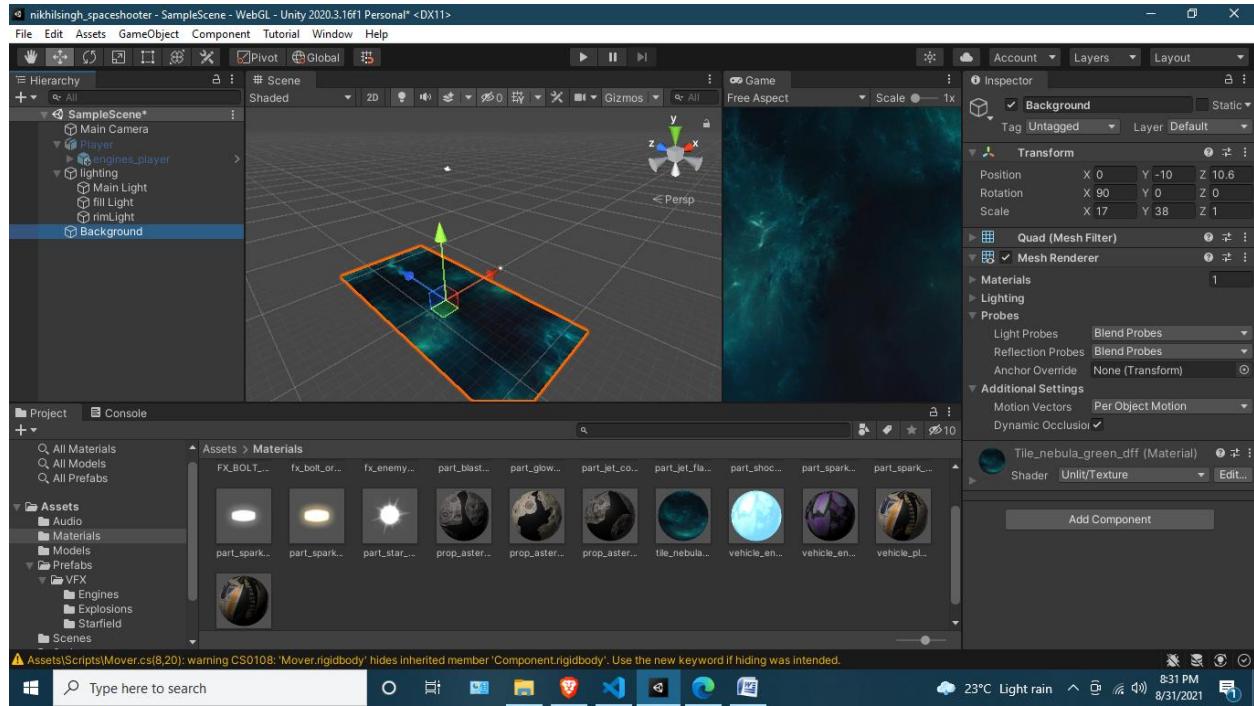


Step 17: Add a Quad , Name it Background, reset its transform and set its rotation x=90.Remove the mesh collider.

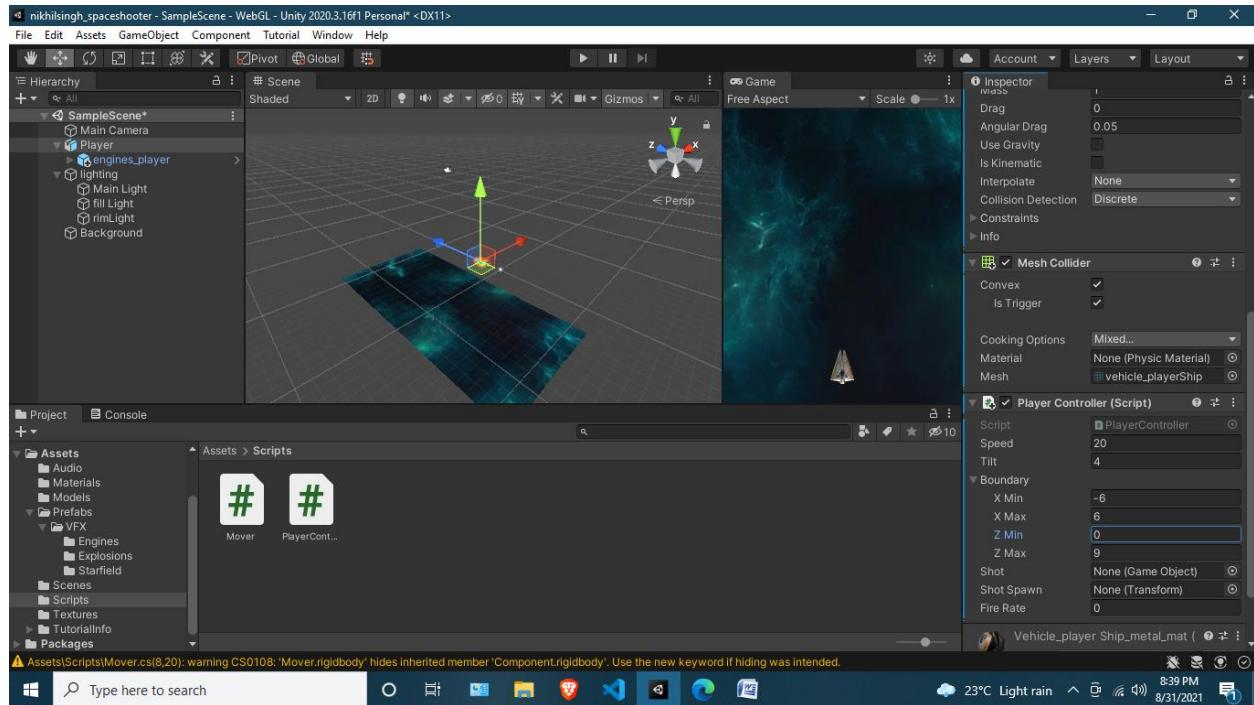


## Practical 8

STEP 18: Drag and drop tile\_nebula\_green material to Background Quad. Scale the Transform X=17 and Y=38 and Position Y= -10. Reactivate the player



Step 19: Click on player game object and add new Script Name PlayerController to it. And assign the values in Editor. Set the speed=20, tilt=4,zmin=-6,xmax=6,zmin=0 and zmax=9

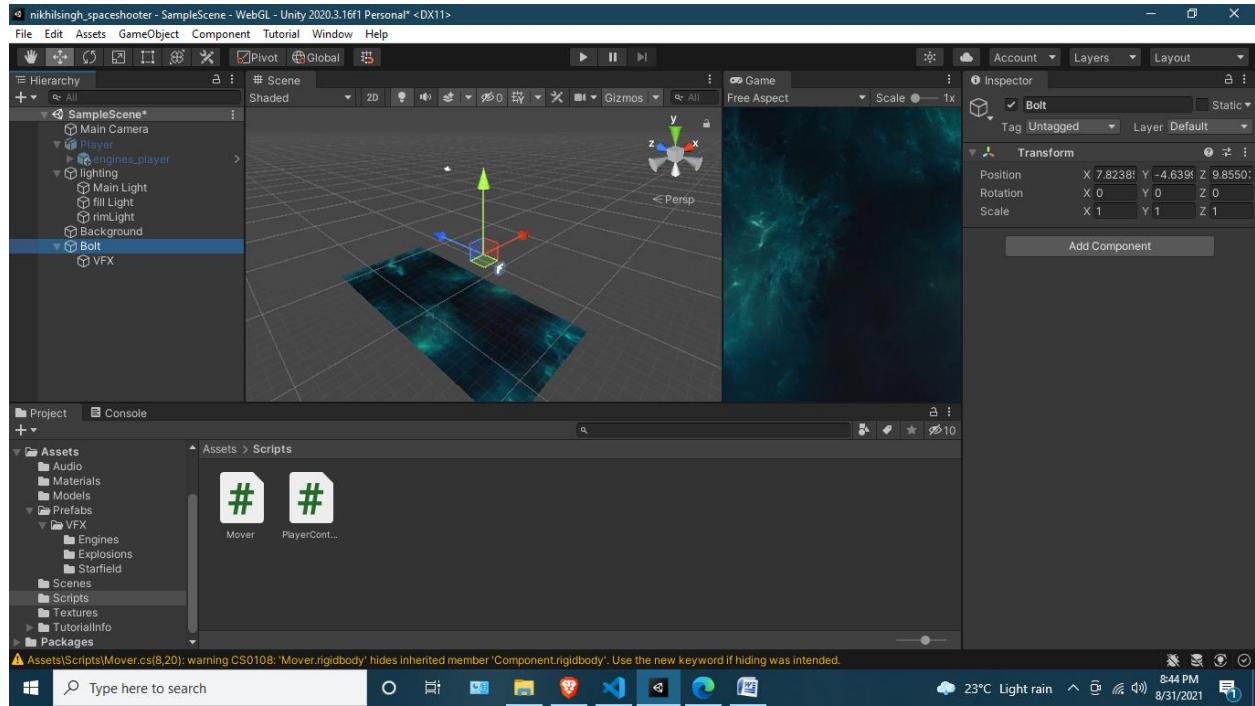


## Practical 8

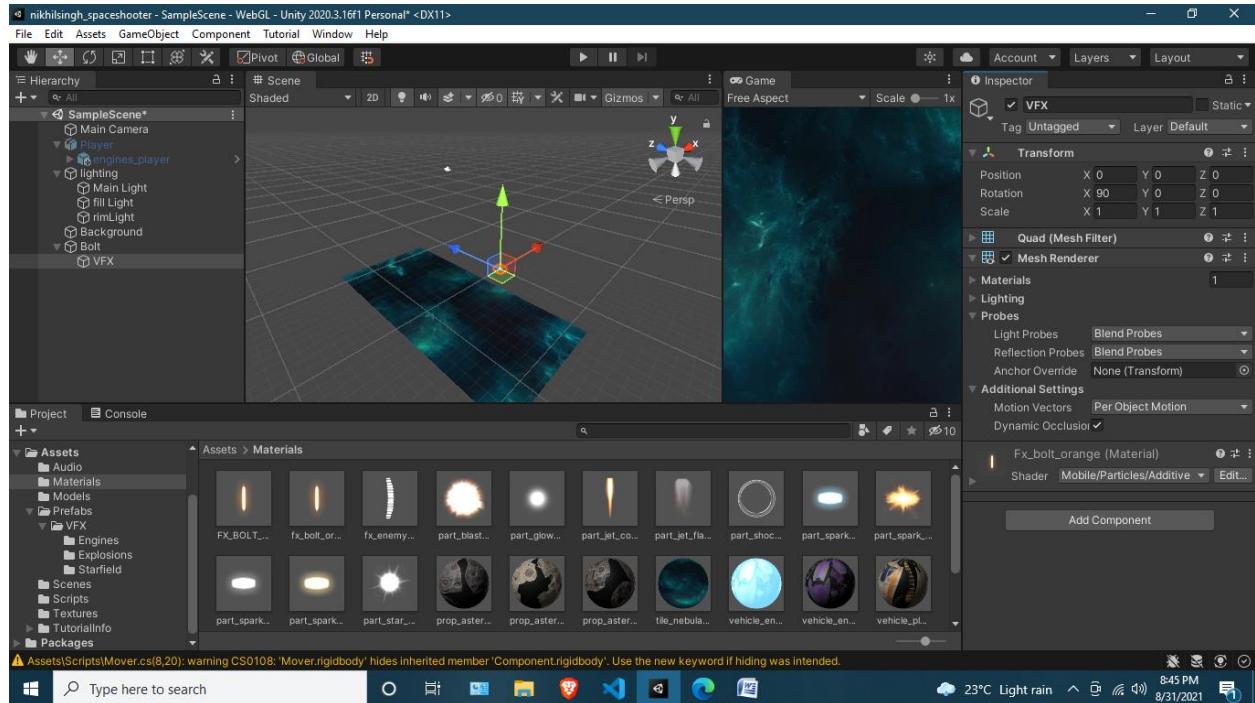
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
//unity c# script to move player object
[System.Serializable]
public class Boundary{
    public float xMin,xMax,zMax,zMin;
}
void FixedUpdate(){
    float moveHorizontal=Input.GetAxis("Horizontal");
    float moveVertical=Input.GetAxis("Vertical");
    Vector3 movement = new Vector3(moveHorizontal,0.0f,moveVertical);
    GetComponent<Rigidbody>().velocity=movement*speed;
    GetComponent<Rigidbody>().position=new Vector3(
        Mathf.Clamp(GetComponent<Rigidbody>().position.x,boundary.xMin,boundary.x
        Max),
        0.0f,
        Mathf.Clamp(GetComponent<Rigidbody>().position.z,boundary.zMin,boundary
        .zMax));
}
GetComponent<Rigidbody>().rotation=Quaternion.Euler(0.0f,0.0f,GetComponent
<Rigidbody>().velocity.x*-tilt);
}
```

## Practical 8

Step 20: Deactivate Player . Create a new Empty game object ,reset its transform and name it Bolt.Create A quad and name it VFX ,reset its transform and make it chid of Bolt.



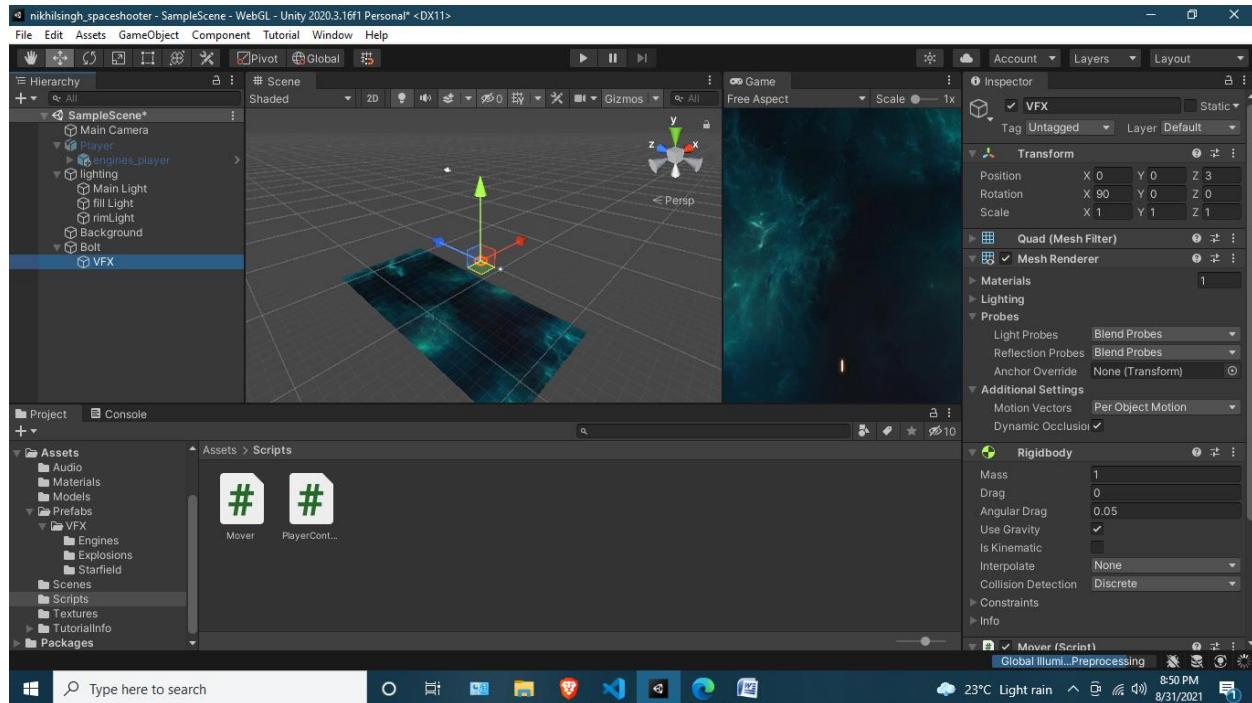
Step 21:Select VFX set rotation X=90, Drag and drop Fx\_bolt\_orange material and remove mesh collider



## Practical 8

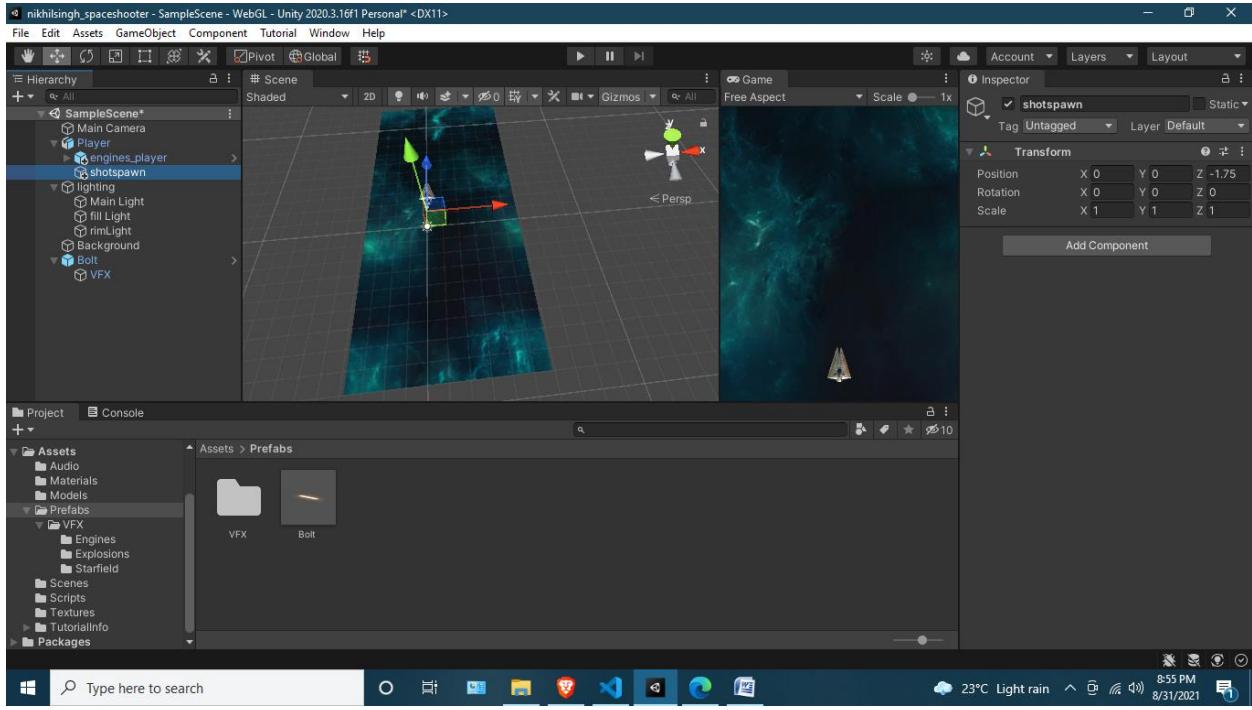
Step 22: Add rigidbody component to and add a script named Mover to Bolt game object. Add speed value 20 and test if bolts are moving

```
using System.Collections;  
  
using System.Collections.Generic;  
  
using UnityEngine;  
  
public class Mover : MonoBehaviour {  
  
    public float speed;  
  
    private Rigidbody rigidbody;  
  
    void Start ()  
  
    {  
  
        rigidbody=GetComponent<Rigidbody>();  
  
        rigidbody.velocity = transform.forward * speed;  
  
    }  
  
}
```

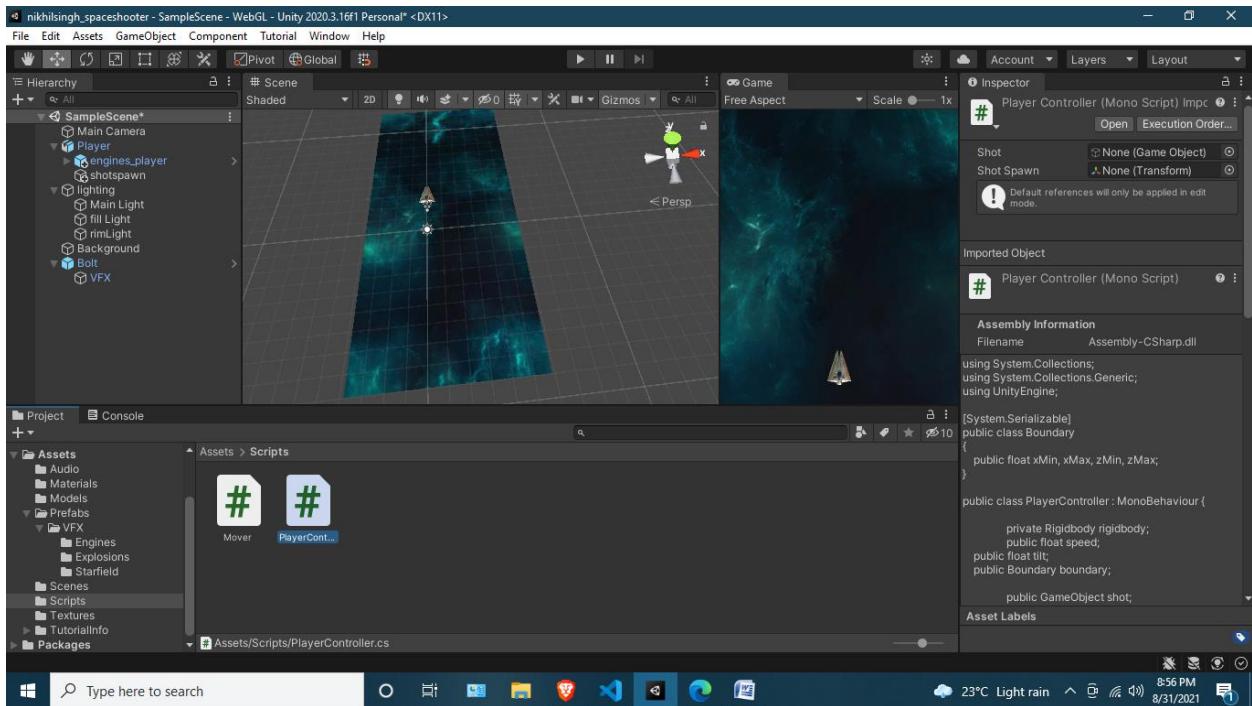


## Practical 8

Step 23: Activate Player and drag and drop Bolt game object to prefabs folder to create its prefab. Create an Empty game object name it shot spawn and make it child of Player. Set its Z position to 1.25

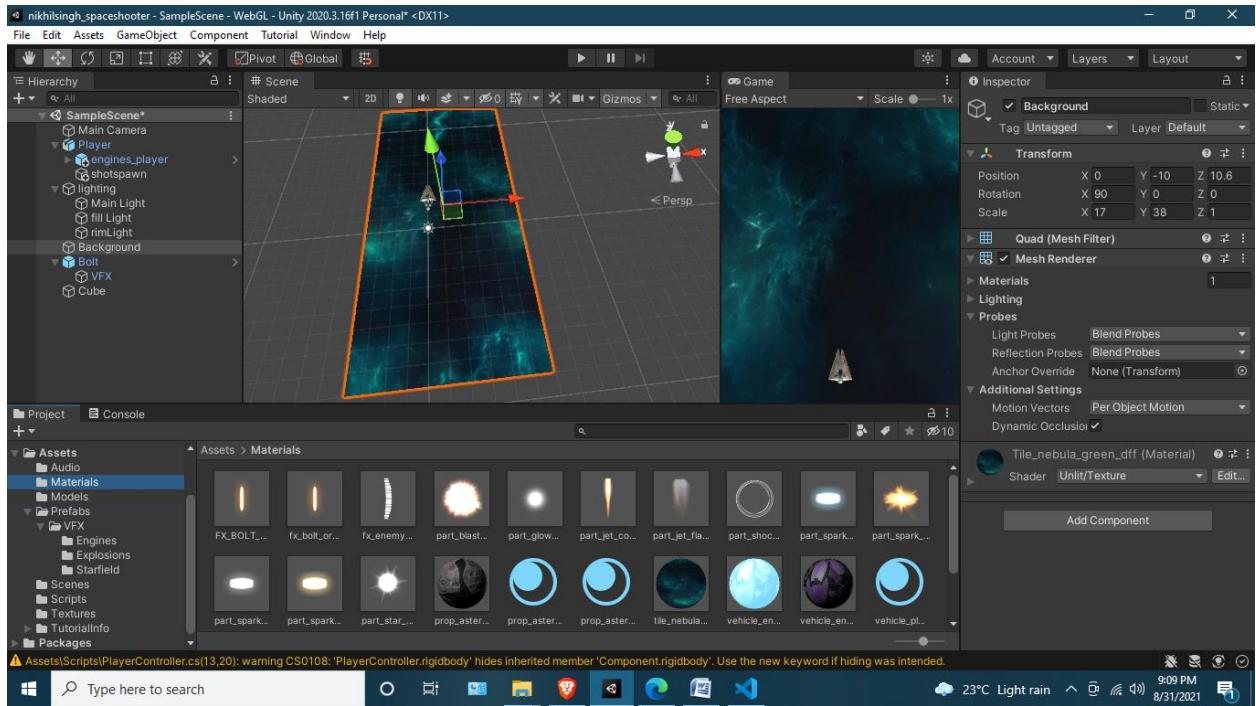


Step 24: Edit the PlayerController script add the code to fire round of shots from the playership upwards. Add Bolt prefab to shot and shotspawn object to shot spawn variables. Set FireRate=0.25



## Practical 8

Step 25: Add a cube and name it boundary , reset its transform. Set Position Z=5, Scale X=15 and Z=20. Check isTrigger component of box collider. Remove the Mesh Renderer and Mesh Filter components.

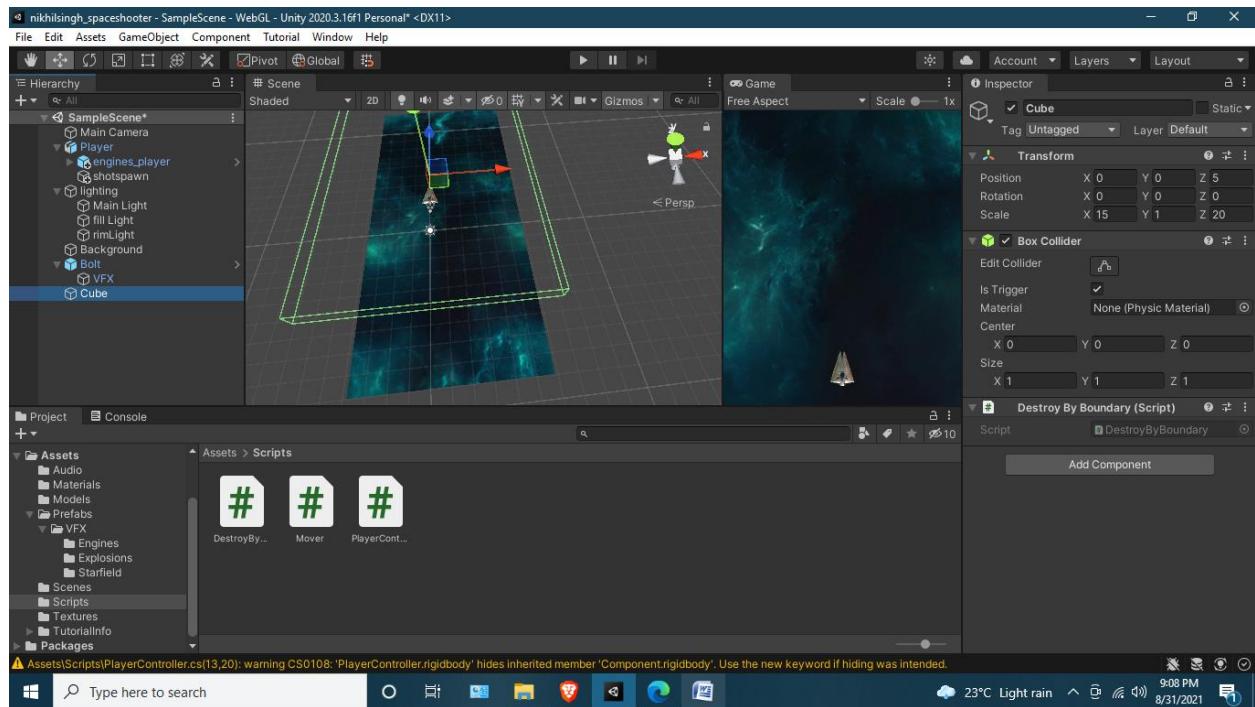


## Practical 8

Step 26 : Add a script to Boundary name it DestroyByBoundary

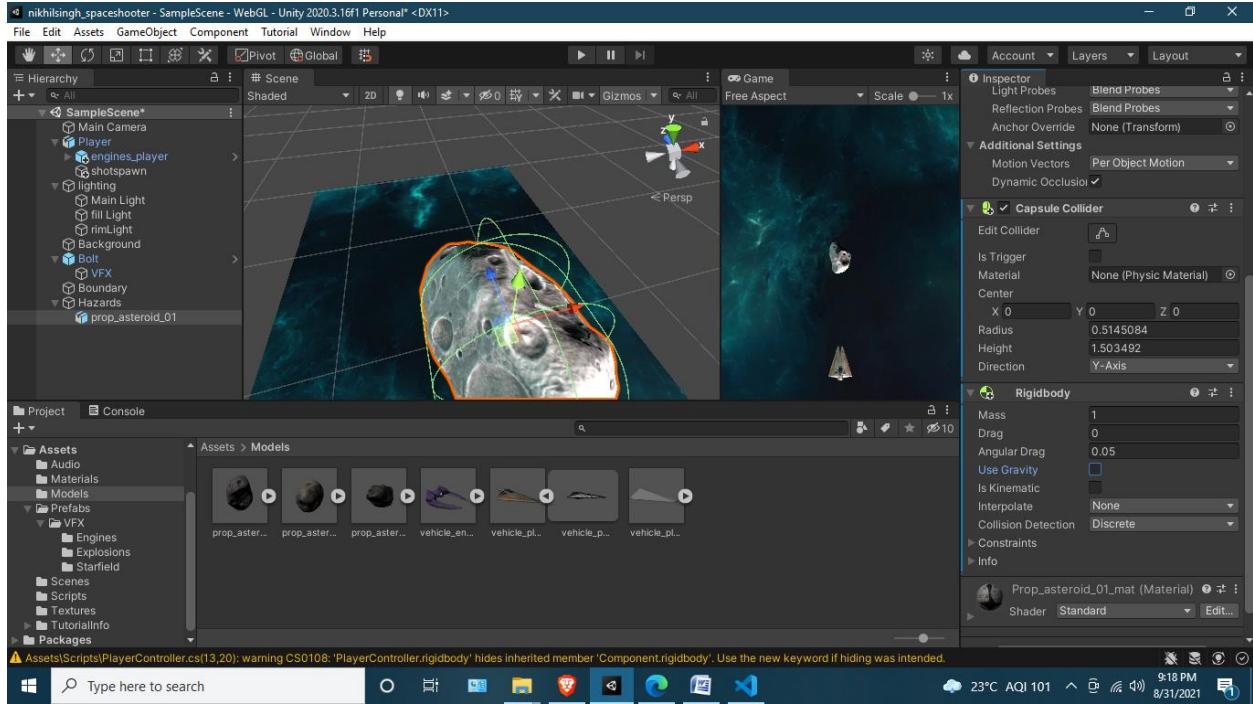
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

```
public class DestroyByBoundary : MonoBehaviour  
{  
    void OnTriggerExit(Collider other){  
        Destroy(other.gameObject);  
    }  
}
```



## Practical 8

Step 27: Make a empty game object name it Hazards and drag and drop asteroid model as a child of it .Add empty object position z=8,add rigidbody,deselect gravity,add capsule collider.

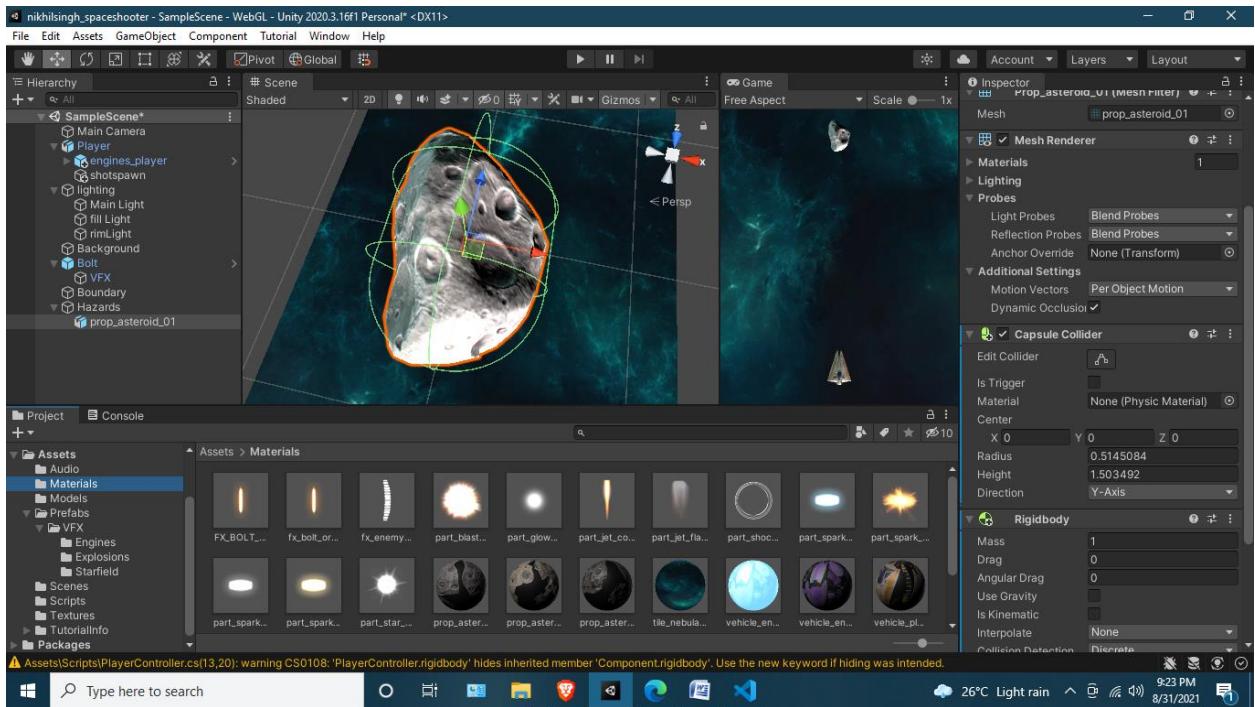


Step 28 : Add a script to asteroid game object name it RandomRotator.

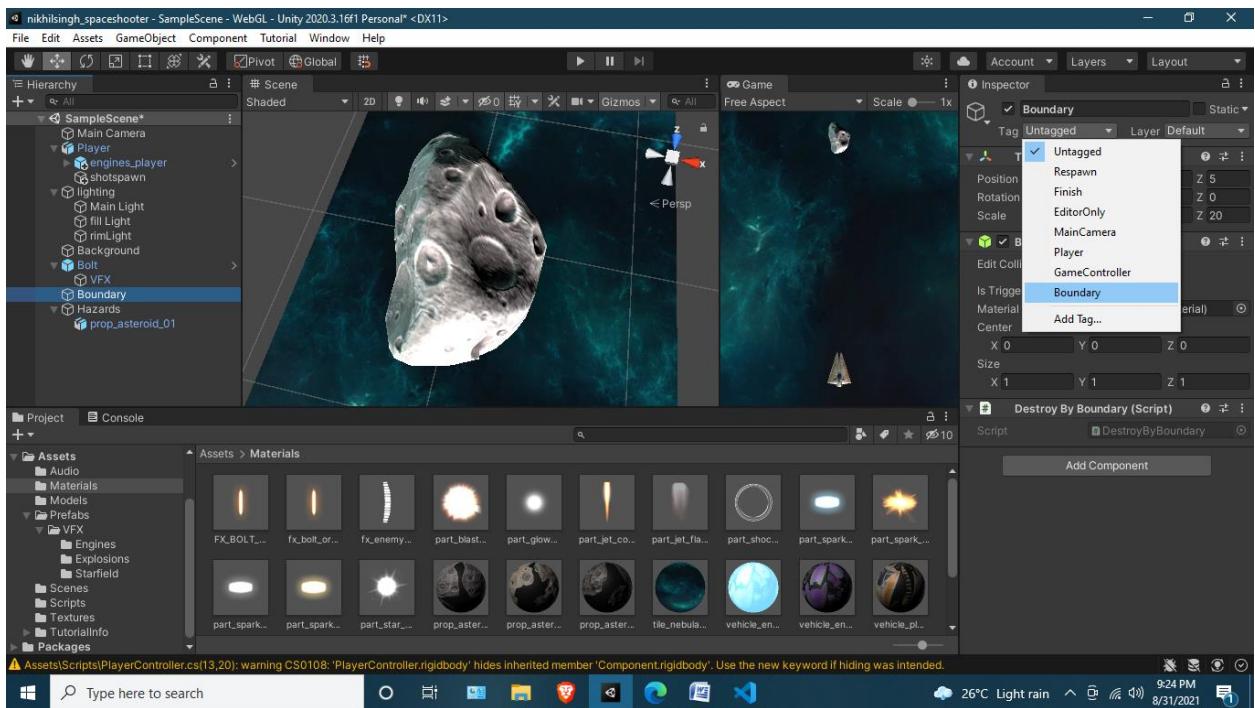
```
using System.Collections;  
  
using System.Collections.Generic;  
  
using UnityEngine;  
  
public class RandomRotator : MonoBehaviour  
{  
  
    public float tumble;  
  
    void Start(){  
  
        GetComponent<Rigidbody>().angularVelocity=Random.insideUnitSphere*tumble;  
    }  
  
}
```

## Practical 8

Step 29: Set Angular drag in rigidbody to 0 and tumble to 5. And play to test it.



Step 30: Select Boundary and create a new tag and name it as Boundary. Set the tag as Boundary.



## Practical 8

Step 31: Add A script to Asteroid name it Destroy BY Contact

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DestroyByContact : MonoBehaviour
{
    public GameObject explosion;
    public GameObject playerExplosion;
    public int ScoreValue;
    private GameController gameController;

    void Start(){
        GameObject gameControllerObject=GameObject.FindGameObjectWithTag("GameController");
        if(gameControllerObject!=null){
            gameController=gameControllerObject.GetComponent<GameController>();
        }
        if(gameControllerObject==null){
            Debug.Log("Cannot fint gamecontroller script");
        }
    }

    void OnTriggerEnter(Collider other){
        if(other.tag == "Boundary"){
            return;
        }
    }
}
```

## Practical 8

```
if(other.tag=="Player"){

    Instantiate(playerExplosion,other.transform.position,other.transform.rotation);

    gameController.GameOver();

}

gameController.AddScore(ScoreValue);

Destroy(other.gameObject);

Destroy(gameObject);

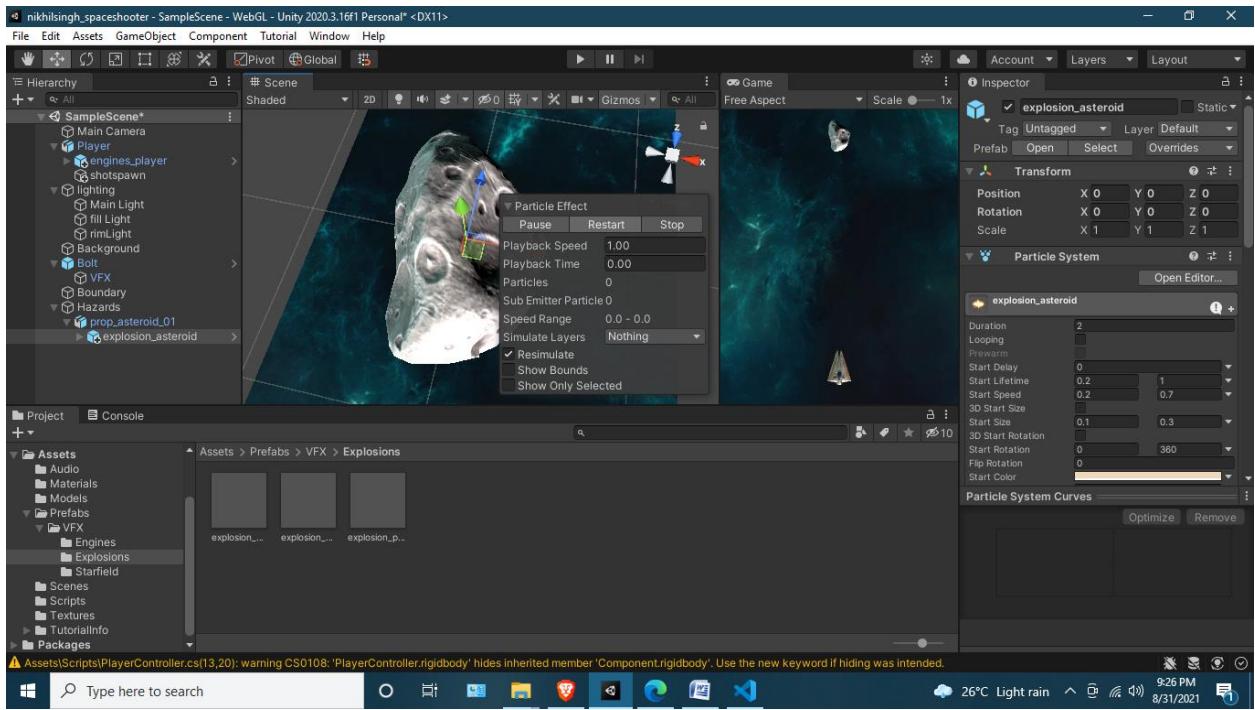
Instantiate(explosion,transform.position,transform.rotation);

}

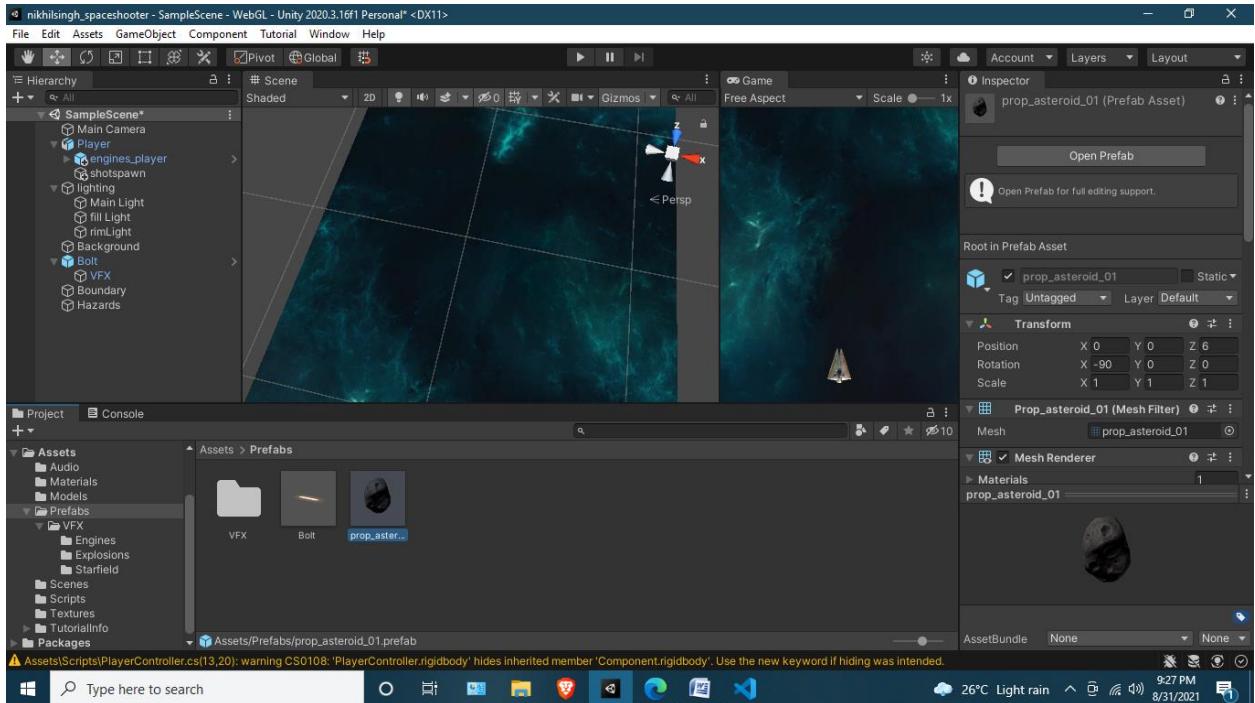
}
```

## Practical 8

### Step 32: Drag and drop Explosion VFX in unity Editor

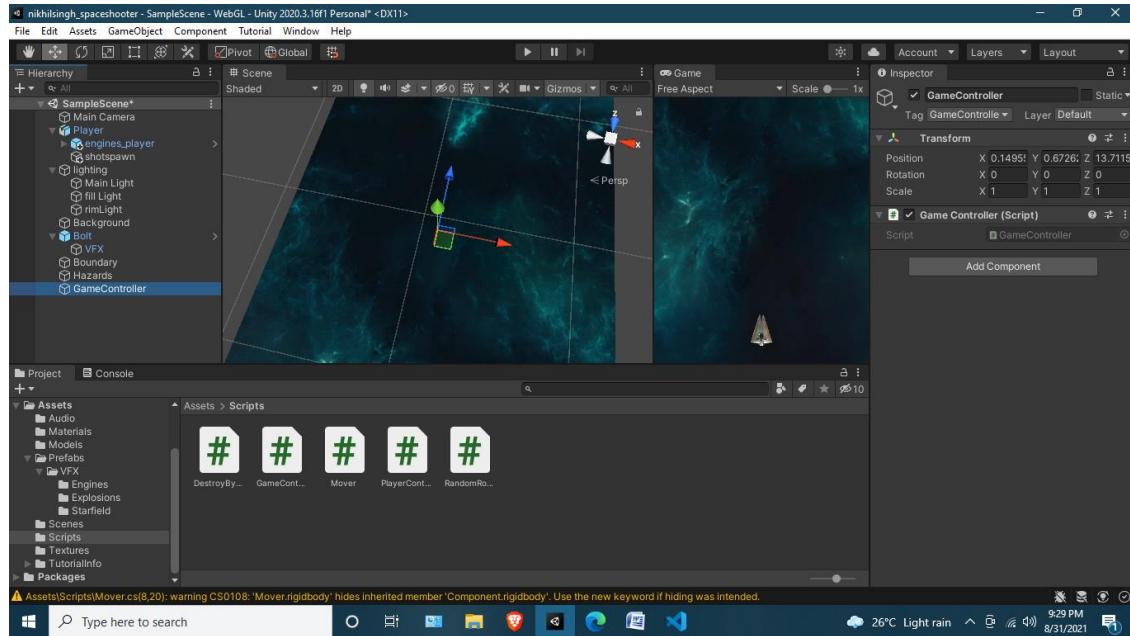


Step 32: Make prefab of asteroid by dragging the game object to prefabs folder and delete the instance from hierarchy .



## Practical 8

Step 33: Create an Empty game object named Game Controller add a script named GameController to the game object . Select Game controller tag

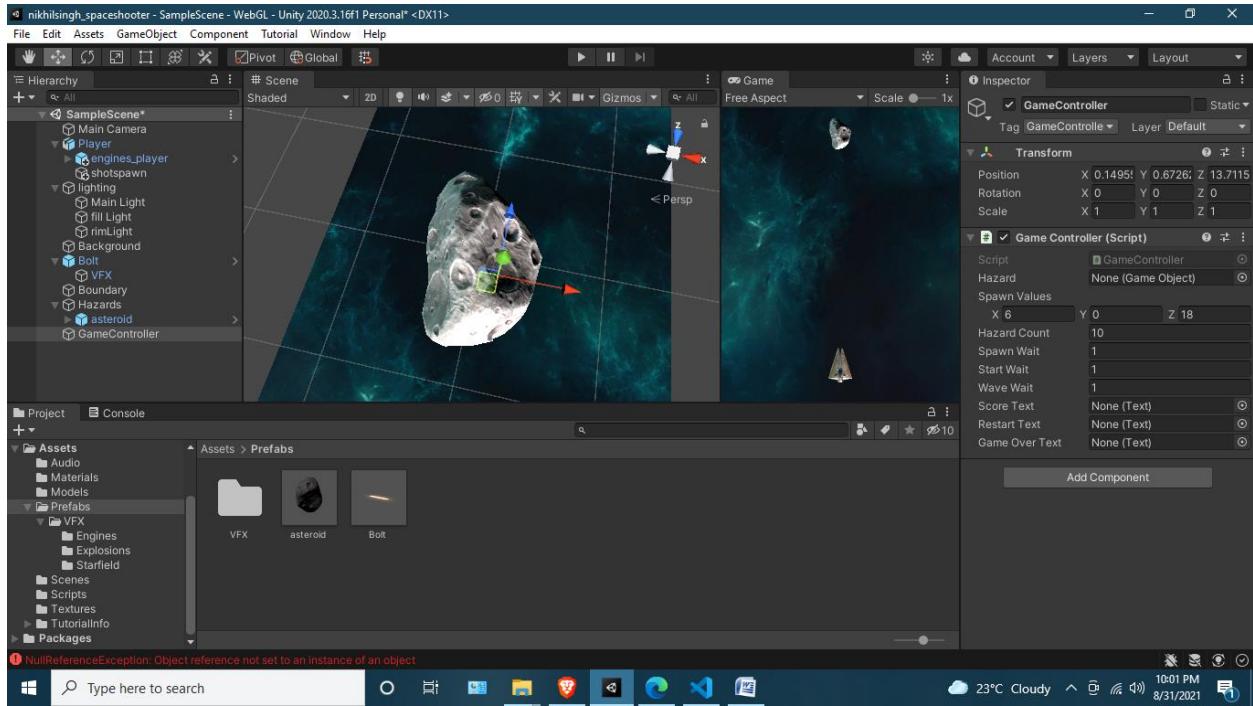


Step 34 : Add the following code to the Game Controller script.

```
GameController.cs
Assets > Scripts > GameController.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GameController : MonoBehaviour
6  {
7
8      public GameObject hazard;
9      public Vector3 spawnValues;
10     public int hazardCount;
11     public float spawnWait;
12     public float startWait;
13     public float waveWait;
14
15     void Start()
16     {
17         StartCoroutine(SpawnWaves());
18     }
19     IEnumerator SpawnWaves()
20     {
21         yield return new WaitForSeconds(startWait);
22         while(true)
23         {
24             for (int i = 0; i < hazardCount; i++)
25             {
26                 Vector3 spawnPosition=new Vector3(Random.Range(spawnValues.x,-spawnValues.x),spawnValues.y,spawnValues.z);
27                 Quaternion spawnRotation= Quaternion.identity;
28                 Instantiate(hazard, spawnPosition, spawnRotation);
29                 yield return new WaitForSeconds(spawnWait);
30             }
31             yield return new WaitForSeconds(startWait);
32         }
33     }
34 }
```

## Practical 8

Give Spawn Values x=6&z=18 ,Hazard Count=10,Spawn Wait,startWait,WaveWait=1 in Unity Editor and play to test



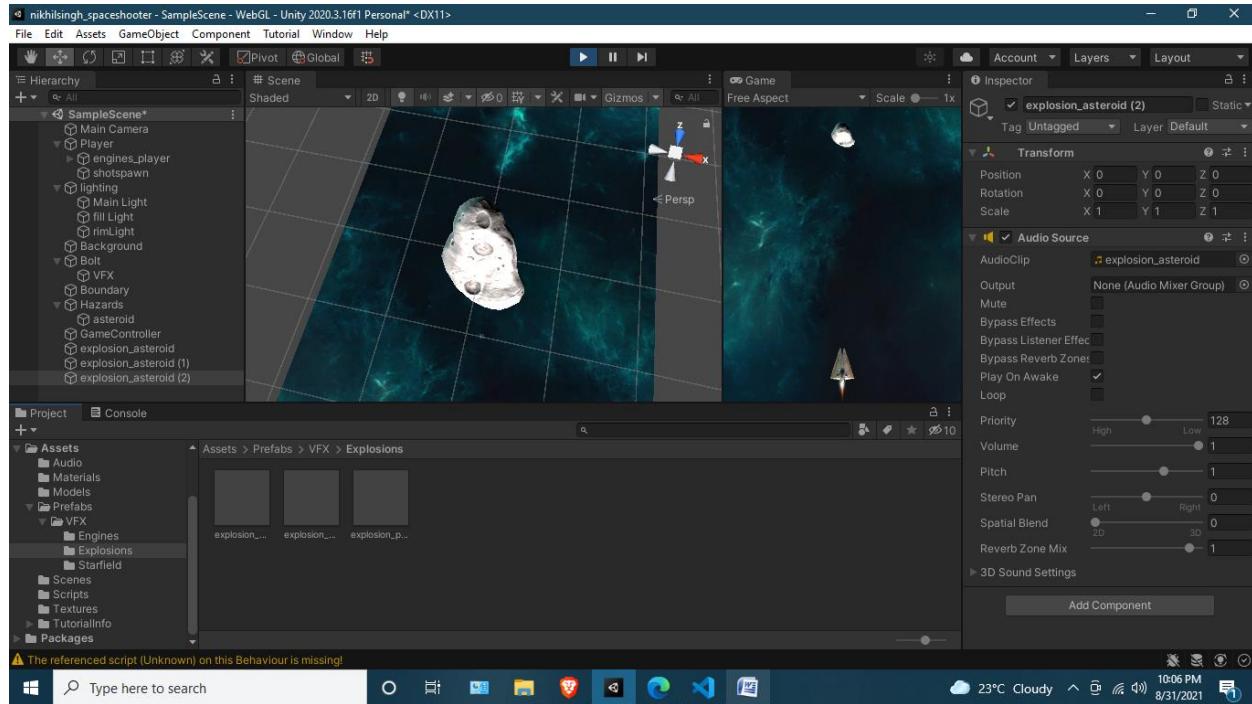
Step 35: Add a script named DestroyByTime to All explosion prefabs. Set lifetime=2

```
using System.Collections;  
  
using System.Collections.Generic;  
  
using UnityEngine;
```

```
public class DestroyByTime : MonoBehaviour  
{  
  
    public float lifeTime;  
  
    void Start(){  
  
        Destroy(gameObject,lifeTime);  
  
    }  
  
}
```

## Practical 8

Step 36: Drag and drop explosion\_asteroid audio into explosion\_asteroid VFX prefab and weapon\_player audio to Player game object. Uncheck play on awake for weapon\_player

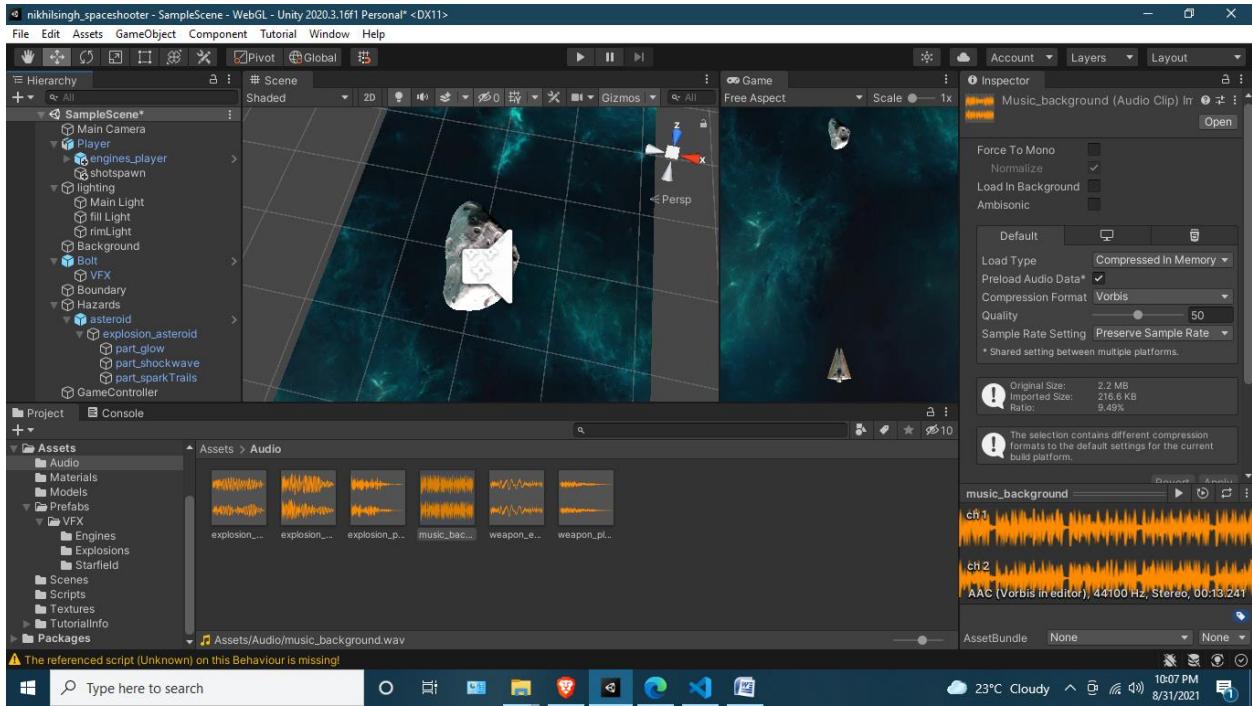


Step 37 :Edit the PlayerController script to play audio when the shots are being fired. Play and test

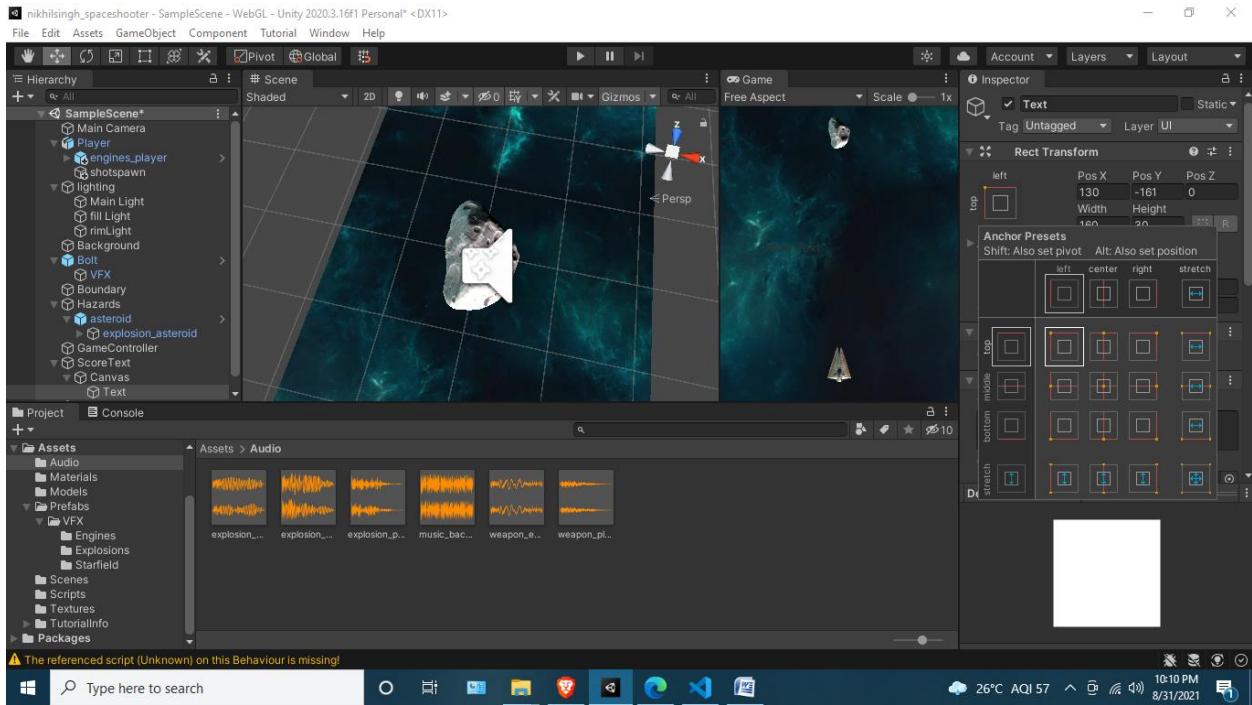
```
C# PlayerController.cs
Assets > Scripts > C# PlayerController.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  //unity c# script to move player object
5  [System.Serializable]
6  public class Boundary{
7      public float xMin,xMax,zMax,zMin;
8  }
10
11
12
13  public class PlayerController : MonoBehaviour
14  {
15      public Boundary boundary;
16      public float tilt;
17      public float speed;
18      public GameObject shot;
19      public Transform shotSpawn;
20      public float nextFire=0.5f;
21      public float fireRate;
22      void Update(){
23
24          if(Input.GetButton("Fire1")&&Time.time>nextFire){
25              nextFire=Time.time+fireRate;
26              Instantiate(shot,shotSpawn.position,shotSpawn.rotation);
27              GetComponent< AudioSource >().Play();
28          }
29      }
}
```

## Practical 8

Step 38: Drag and drop music\_background audio to gamecontroller object .Check loop and set volume to 0.



Step 39: Create an Empty gameobject name it ScoreText. Add a UI-text component to this object , on rect transform hold alt and select. Option .change the text to Score Text and colour to white



## Practical 8

Step 40: Edit game controller script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.UIElements;

public class GameController : MonoBehaviour
{
    public GameObject hazard;
    public Vector3 spawnValues;
    public int hazardCount;
    public float spawnWait;
    public float startWait;
    public float waveWait;

    public Text scoreText;
    public Text restartText;
    public Text gameOverText;
    private bool gameOver;
    private bool restart;

    private int score;

    void Start(){
        gameOver = false;
    }
```

## Practical 8

```
restart = false;

restartText.text = "";
gameOverText.text = "";
score=0;
UpdateScore();
StartCoroutine(SpawnWaves());
}

void Update(){
if(restart){
    if(Input.GetKeyDown(KeyCode.R)){
        Application.LoadLevel(Application.loadedLevel);
    }
}
IEnumerator SpawnWaves()
{
    yield return new WaitForSeconds(startWait);
    while(true){
        for (int i = 0; i < hazardCount; i++)
        {
            Vector3 spawnPosition=new Vector3(Random.Range(spawnValues.x,-spawnValues.x),spawnValues.y,spawnValues.z);
            Quaternion spawnRotation= Quaternion.identity;
            Instantiate(hazard, spawnPosition, spawnRotation);
            yield return new WaitForSeconds(spawnWait);
        }
    }
}
```

## Practical 8

```
}

yield return new WaitForSeconds(startWait);

if(gameOver){

    restartText.text = "Press 'R' for Restart";

    restart=true;

    break;

}

}

}

public void AddScore(int newScoreValue){

    score=score+newScoreValue;

    UpdateScore();

}

void UpdateScore()

{

    scoreText.text="Score:"+score.ToString();

}

public void GameOver(){

    gameOverText.text="Game Over";

    gameOver=true;

}

}
```

## Practical 8

Step 41:Edit Destroy By Contact Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DestroyByContact : MonoBehaviour
{
    public GameObject explosion;
    public GameObject playerExplosion;
    public int ScoreValue;
    private GameController gameController;

    void Start(){
        GameObject gameControllerObject=GameObject.FindGameObjectWithTag("GameController");
        if(gameControllerObject!=null){
            gameController=gameControllerObject.GetComponent<GameController>();
        }
        if(gameControllerObject==null){
            Debug.Log("Cannot fint gamecontroller script");
        }
    }

    void OnTriggerEnter(Collider other){
        if(other.tag == "Boundary"){
            return;
        }
    }
}
```

## Practical 8

```
if(other.tag=="Player"){

    Instantiate(playerExplosion,other.transform.position,other.transform.rotation);

    gameController.GameOver();

}

gameController.AddScore(ScoreValue);

Destroy(other.gameObject);

Destroy(gameObject);

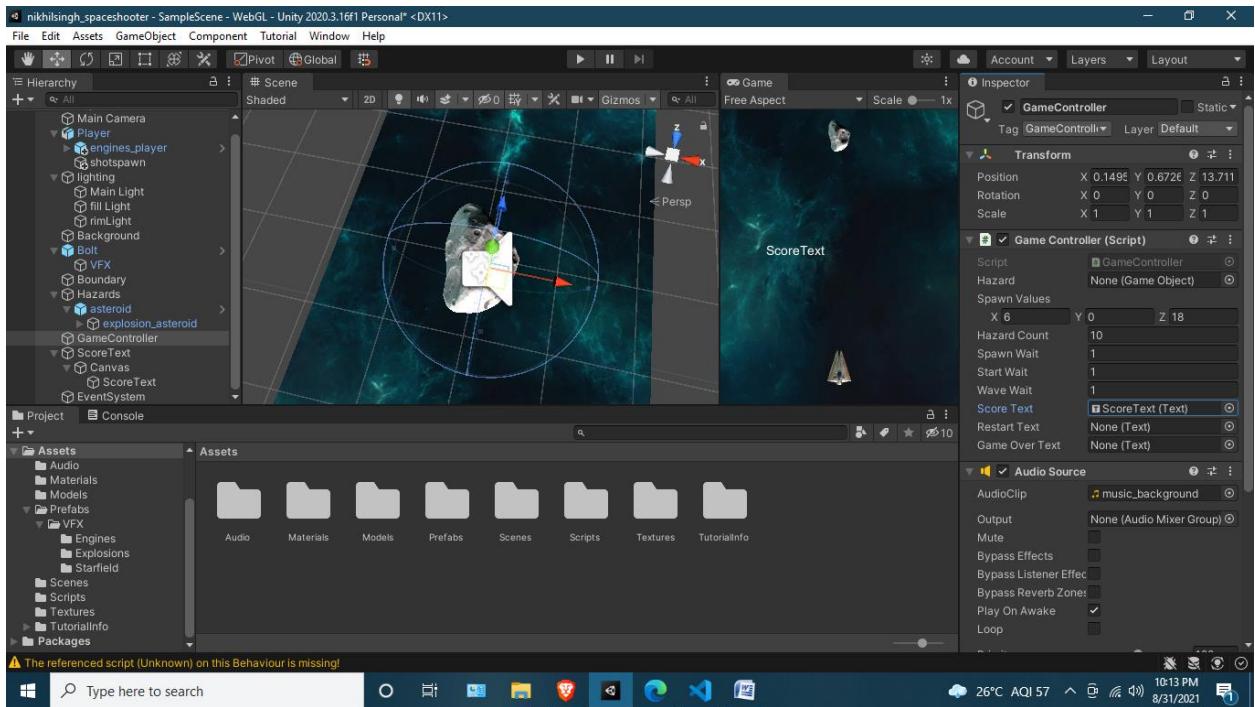
Instantiate(explosion,transform.position,transform.rotation);

}

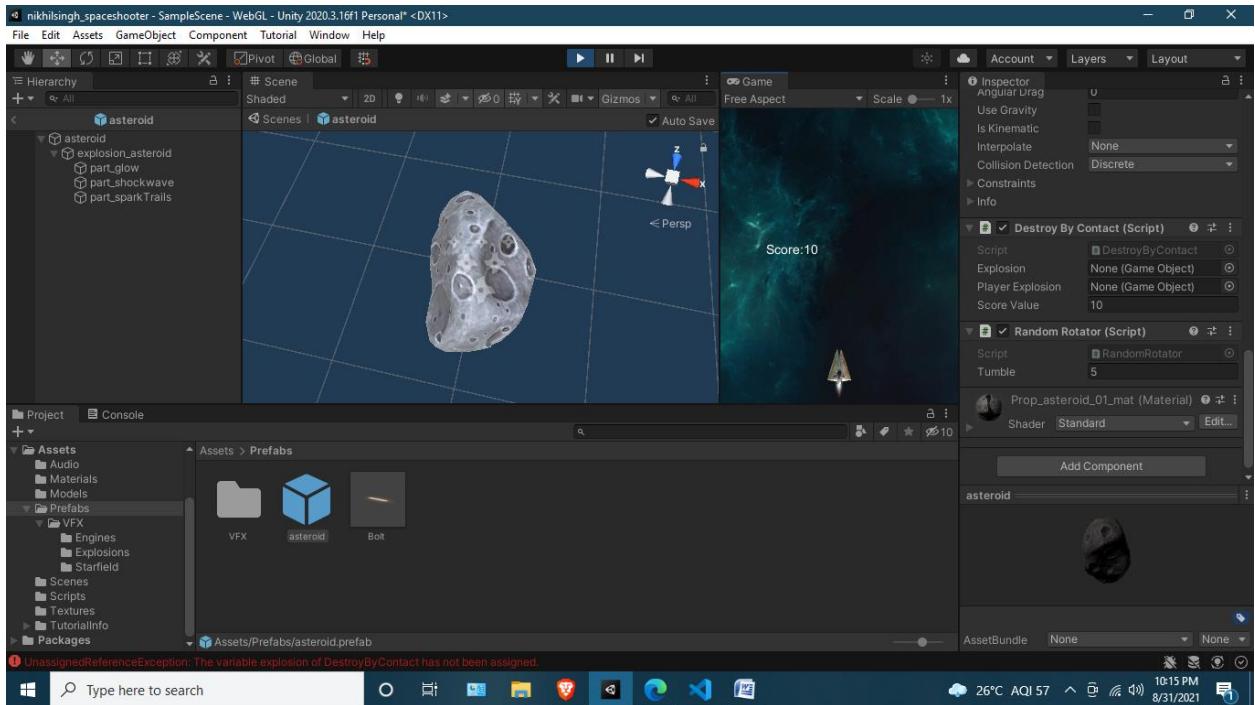
}
```

## Practical 8

Step 42: Add score text component to game controller game object.

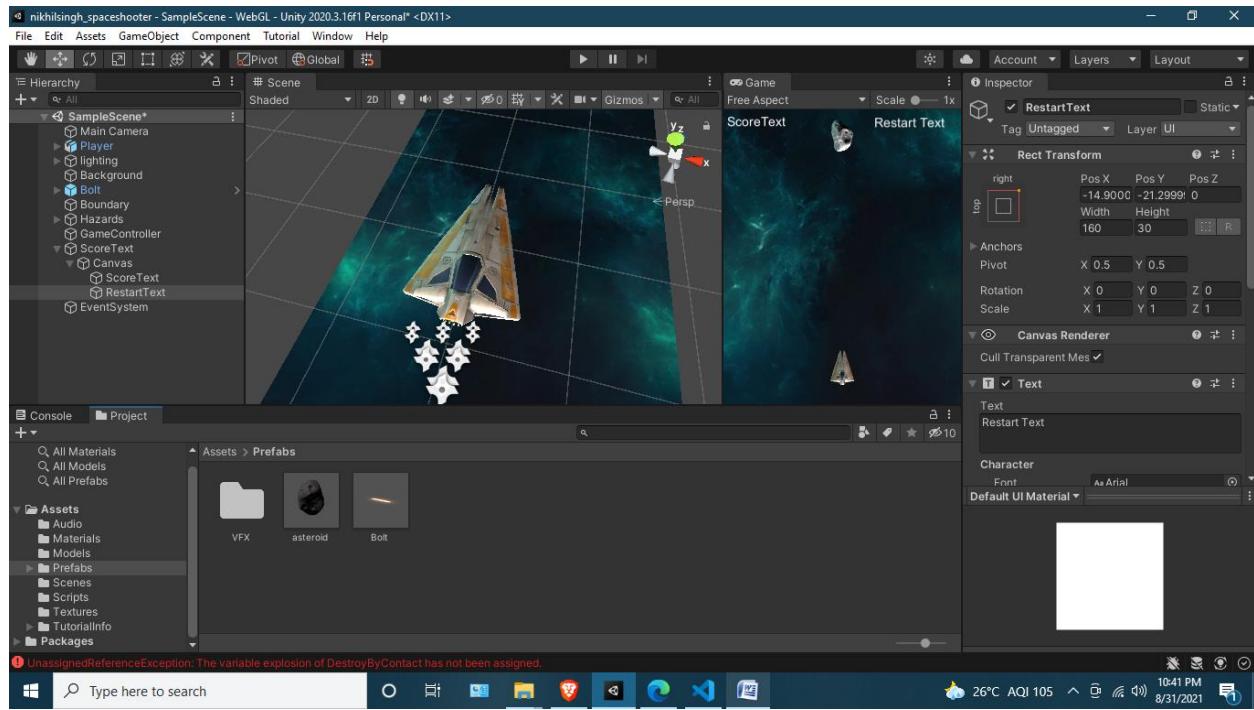


Step 43: Set the score value=10 in asteroid prefab and check if the score update

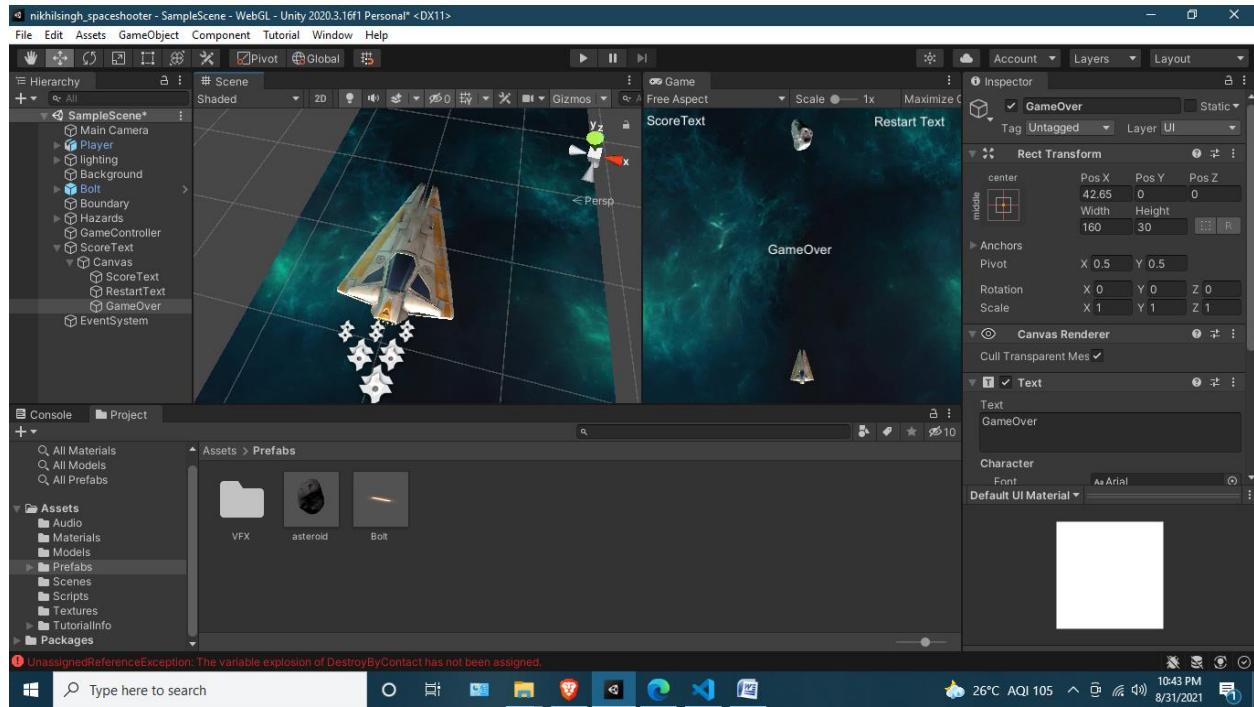


## Practical 8

Step 44: Add a new UI Text and name it Restart Text position it to the top right of the screen



Step 45: Add a new UI Text Component and name it as Game Over Text reset its text property .



## Practical 8

Step 46:Edit GameController Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.UIElements;

public class GameController : MonoBehaviour
{
    public GameObject hazard;
    public Vector3 spawnValues;
    public int hazardCount;
    public float spawnWait;
    public float startWait;
    public float waveWait;

    public Text scoreText;
    public Text restartText;
    public Text gameOverText;
    private bool gameOver;
    private bool restart;

    private int score;

    void Start()
    {
        gameOver = false;
    }
```

## Practical 8

```
restart = false;

restartText.text = "";

gameOverText.text = "";

score=0;

UpdateScore();

StartCoroutine(SpawnWaves());

}

void Update(){

if(restart){

    if(Input.GetKeyDown(KeyCode.R)){

        Application.LoadLevel(Application.loadedLevel);

    }

}

IEnumerator SpawnWaves()

{

    yield return new WaitForSeconds(startWait);

    while(true){

        for (int i = 0; i < hazardCount; i++){

            {

                Vector3 spawnPosition=new Vector3(Random.Range(spawnValues.x,-spawnValues.x),spawnValues.y,spawnValues.z);

                Quaternion spawnRotation= Quaternion.identity;

                Instantiate(hazard, spawnPosition, spawnRotation);

                yield return new WaitForSeconds(spawnWait);

            }

        }

    }

}
```

## Practical 8

```
}

yield return new WaitForSeconds(startWait);

if(gameOver){

    restartText.text = "Press 'R' for Restart";

    restart=true;

    break;

}

}

}

public void AddScore(int newScoreValue){

    score=score+newScoreValue;

    UpdateScore();

}

void UpdateScore()

{

    scoreText.text="Score:"+score.ToString();

}

public void GameOver(){

    gameOverText.text="Game Over";

    gameOver=true;

}

}
```

## Practical 8

Step 47:Edit DestroyBy Contact script to add reference to gameOver

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DestroyByContact : MonoBehaviour
{
    public GameObject explosion;
    public GameObject playerExplosion;
    public int ScoreValue;
    private GameController gameController;

    void Start(){
        GameObject gameControllerObject=GameObject.FindGameObjectWithTag("GameController");
        if(gameControllerObject!=null){
            gameController=gameControllerObject.GetComponent<GameController>();
        }
        if(gameControllerObject==null){
            Debug.Log("Cannot fint gamecontroller script");
        }
    }

    void OnTriggerEnter(Collider other){
        if(other.tag == "Boundary"){
            return;
        }
    }
}
```

## Practical 8

```
if(other.tag=="Player"){

    Instantiate(playerExplosion,other.transform.position,other.transform.rotation);

    gameController.GameOver();

}

gameController.AddScore(ScoreValue);

Destroy(other.gameObject);

Destroy(gameObject);

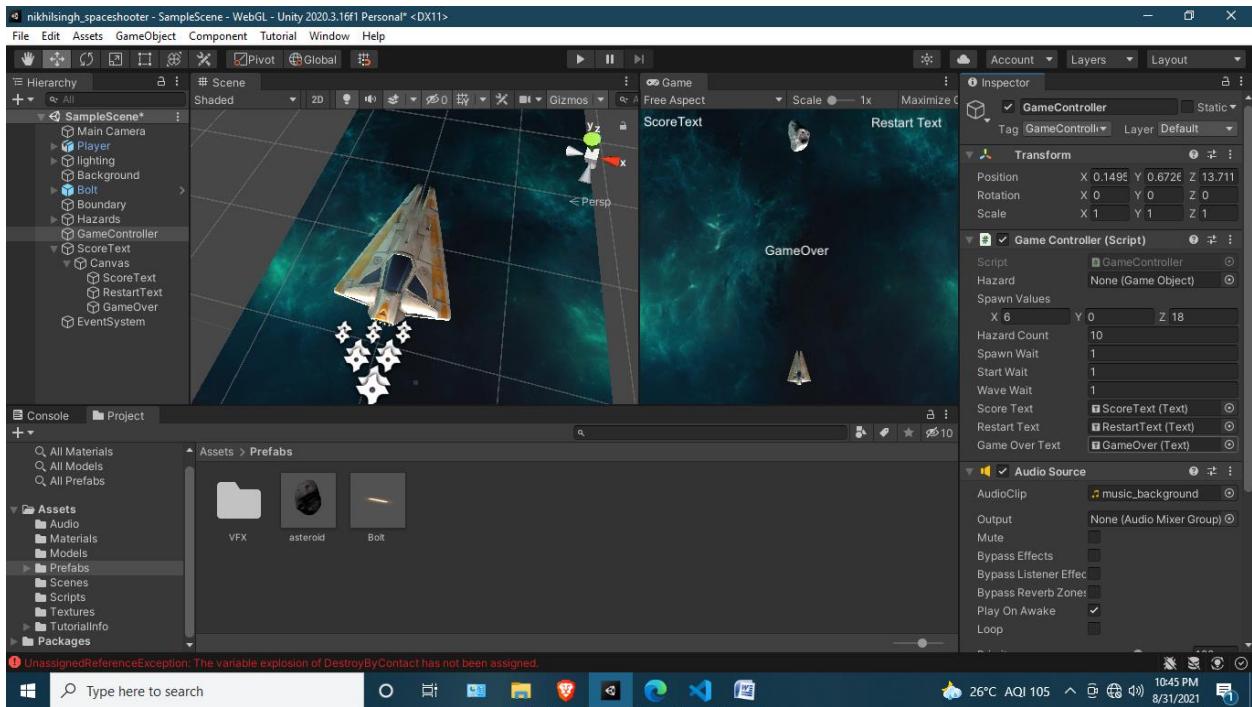
Instantiate(explosion,transform.position,transform.rotation);

}

}
```

## Practical 8

Step 48: Drag and drop gameOverText and restartText in their reference in the gamecontroller script



Step 49: Go to File – Build Settings. Drag and drop your scene into the scene slot. Click build and save it at desired location.

