5th Jan 2022

Game Project Report

Space Fighter 3D

Yousaf Khan

OVERVIEW

Space Fighter 3D is an arcade style space shooter game inspired by the 1979 Asteroids and the 1978 Space Invaders. It is a side-scrolling game where the player is a space fighter and has to destroy asteroids and enemy spaceships to gain points.

The Asteroids come in different shapes and sizes with different health and speed adding a challenge for the player.

On top of this, there is a wave level that increases as the wave of asteroids is destroyed. Higher wave levels offer more asteroids making the game more difficult. After every 10 waves, an enemy fleet approaches that has 5 enemy spaceships shooting at the player adding another challenge.

The player has 3 lives that will decrease if they are hit by asteroids or enemy bullets.

CONTROLS

The player is controlled using the arrow keys. The player speed is constant in all directions.

The player can shoot bullets by pressing the space-bar. The bullets have a 0.2 second delay so they cannot be spammed. When the player shoots, two bullets are fired.

PLAYER

The player controls the space fighter. They have to dodge and destroy asteroids and enemy ships.

The player spaceship prefab was used from the asset store: Star Sparrow Modular Spaceship by Ebal Studios. (see figure on right).

I added a custom collider to the spaceship to allow it to contact the asteroids.

I added two scripts to the spaceship:

Boundaries.cs - to keep the spaceship in bounds.

shipController.cs - to add control, health, speed etc. to the spaceship.

boundries.cs

```
using UnityEngine;

public class boundries : MonoBehaviour
{
    private Vector3 viewPos;

    // Update is called once per frame
    void LateUpdate()
    {
        viewPos = transform.position;
        viewPos = Camera.main.WorldToViewportPoint(viewPos);
        viewPos.x = Mathf.Clamp(viewPos.x, 0.1f, 0.92f);
        viewPos.y = Mathf.Clamp(viewPos.y, 0.06f, 0.94f);
        transform.position =
Camera.main.ViewportToWorldPoint(viewPos);
    }
}
```

shipController.cs

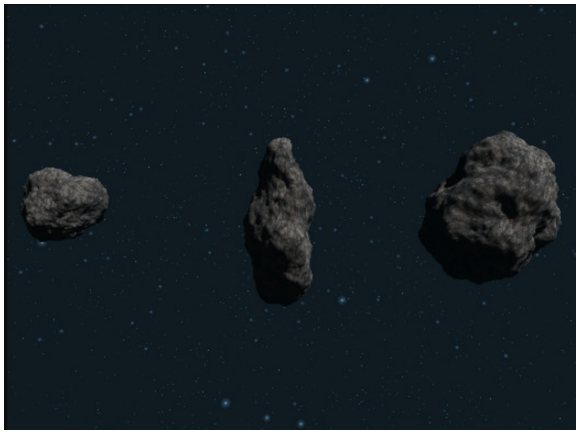
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class shipController : MonoBehaviour
{
    public float speed = 20.0f;
    public GameObject bulletReference;
    public static int playerHealth;
```

```
    public float bulletDelay = 0.2f;
    private bool hasCollide = false, canShoot = true;
    private GameObject explosion;

    void Start()
    {
        playerHealth = 3;
    }
    void Update()
    {
        if (Input.GetKey(KeyCode.UpArrow))
        {

            transform.position = transform.position + Vector3.up *
Time.deltaTime * speed;
        }
        if (Input.GetKey(KeyCode.DownArrow))
        {
            transform.position = transform.position + Vector3.down *
Time.deltaTime * speed;
        }
        if (Input.GetKey(KeyCode.LeftArrow))
        {
            transform.position = transform.position + Vector3.left *
Time.deltaTime * speed;
        }
        if (Input.GetKey(KeyCode.RightArrow))
        {
            transform.position = transform.position + Vector3.right *
Time.deltaTime * speed;
        }
        if (Input.GetKeyDown(KeyCode.Space))
        {
            if (canShoot)
            {
                shootBullet();
                canShoot = false;
                StartCoroutine(shootDelay());
            }
        }
    }

    private void LateUpdate()
    {
        hasCollide = false;
        if (playerHealth <= 0) {
            explosion =
(GameObject)GameObject.Instantiate(Resources.Load("Explosion 1"));
```

```
            explosion.transform.position = transform.position;
            explosion.transform.localScale = transform.localScale *
2;
            Destroy(gameObject);
            GUIScript.isGameOver = true;
        }
    }

    void shootBullet()
    {
        GameObject bullet1 = GameObject.Instantiate(bulletReference);
        GameObject bullet2 = GameObject.Instantiate(bulletReference);
        bullet1.transform.position = transform.position + new
Vector3(4,3.1f,1);
        bullet2.transform.position = transform.position + new
Vector3(4, -3.1f, 1);
    }

    IEnumerator shootDelay() {
        yield return new WaitForSeconds(bulletDelay);
        canShoot = true;

    }

    private void OnCollisionEnter(Collision collision)
    {
        if (hasCollide == false)
        {
            hasCollide = true;
            transform.position = transform.position + Vector3.left *
2f;
            playerHealth--;
        }

    }


}
```

ASTEROIDS

I used 5 different asteroid prefabs from the Asset store. Asteroids Pack by Mark Dion and Star Sparrow Spaceship by Ebal Studios.



I added colliders and rigidbodies to the asteroids.

I used an AsteroidDeployers object to deploy the asteroids randomly and I added an asteroidScript to the asteroid prefabs to add speed and health etc. to the asteroids.

asteroidDeployer.cs

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

public class asteroidDeployer : MonoBehaviour
{
    public GameObject asteroid1, asteroid2, asteroid3, asteroidBlue,
asteroidRed;
    public float respawnTime = 1.0f;
    private Vector3 viewPos;
    private static int waveLevel;
    private static int numberOfAsteroidsDestroyed;
    public static int score;

    private bool isFleetWave = false, isFleetDeployerActive = false;
    private List<GameObject> asteroidsList;
    public GameObject fleetDeployer;
    void Start()
    {
        score = 0;
        waveLevel = 1;
        numberOfAsteroidsDestroyed = 0;
        asteroidsList = new List<GameObject>();
        asteroidsList.Add(asteroid1);
        asteroidsList.Add(asteroid2);
        asteroidsList.Add(asteroid3);
        asteroidsList.Add(asteroidBlue);
        asteroidsList.Add(asteroidRed);
        StartCoroutine(asteroidWave());
    }

    private void deployAsteroid()
    {
        GameObject asteroid =
Instantiate(asteroidsList[Random.Range(0, 5)]);
        asteroid.transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.1f, Random.value,
45f));


    }
```

```
IEnumerator asteroidWave()
{
    while (true) {
        if (isFleetWave)
        {
            yield return new WaitUntil(() => !isFleetWave);
        }
        yield return new WaitForSeconds(respawnTime);
        deployAsteroid();
    }
}

public static void incrementNumberOfAsteroidsDestroyed() {
    numberOfAsteroidsDestroyed++;
}

private void Update()
{
    if (GUIScript.isGameOver)
    {
        Destroy(gameObject);
    }
    if (numberOfAsteroidsDestroyed > 10)
    {
        waveLevel++;
        if (respawnTime > 0.3f)
        {
            respawnTime -= 0.1f;
        }

        numberOfAsteroidsDestroyed = 0;
    }

    if (waveLevel % 10 == 0)
    {
        isFleetWave = true;
        if (!isFleetDeployerActive)
        {
            isFleetDeployerActive = true;
            StartCoroutine(DelayFleetDeployer());
        }

    }

    if(waveLevel % 10 == 1)
    {
        respawnTime = 0.8f;
```

```
        }

        if(waveLevel % 10 != 0)
        {
            isFleetWave = false;
            isFleetDeployerActive = false;
        }
    }

    public static int getWaveLevel() {
        return waveLevel;
    }

    public static void incrementWaveLevel()
    {
        waveLevel++;
    }

    IEnumerator DelayFleetDeployer()
    {
        yield return new WaitForSeconds(5);
        Instantiate(fleetDeployer);
    }

}
```

asteroidScript.cs

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

public class asteroidScript : MonoBehaviour
{
    private Rigidbody rb;
    public float speed = 20f;
    public float tumble = 5;
    public float asteroidHealth = 10f;
    private GameObject explosion;
    private Vector3 viewPos;
    void Start()
    {
        rb = gameObject.GetComponent<Rigidbody>();
        rb.velocity = Vector3.left * speed;
        rb.angularVelocity = Random.insideUnitSphere * tumble;
```

```
    }

    // Update is called once per frame
    void LateUpdate()
    {
        viewPos = transform.position;
        viewPos = Camera.main.WorldToViewportPoint(viewPos);
        if (viewPos.x < -0.1) {
            Destroy(gameObject);
        }
        if (asteroidHealth <= 0) {
            explosion =
(GameObject)GameObject.Instantiate(Resources.Load("Explosion 1"));
            explosion.transform.position = transform.position;
            explosion.transform.localScale = transform.localScale *
1.3f;
            Destroy(gameObject);
            asteroidDeployer.incrementNumberOfAsteroidsDestroyed();
            switch (gameObject.name) {
                case "Asteroid 1(Clone)":
                    asteroidDeployer.score += 10;
                    break;
                case "Asteroid 2(Clone)":
                    asteroidDeployer.score += 30;
                    break;
                case "Asteroid 3(Clone)":
                    asteroidDeployer.score += 50;
                    break;
                case "Asteroid Lava Red(Clone)":
                    asteroidDeployer.score += 150;
                    break;
                case "Asteroid Lava Blue(Clone)":
                    asteroidDeployer.score += 100;
                    break;
            }
        }
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.name == "Bullet(Clone)") {
            asteroidHealth = asteroidHealth - 10f;
        }
        if (collision.gameObject.name == "MainFighter(Clone)")
        {
            Destroy(gameObject);
        }
```

```
    }
}
```

BULLETS

For the bullets i created them using a cylinder game object and adding a yellow glowing material to it. For the enemy bullet i added a blue material. I added a collider and rigidbody to the bullet. I then added scripts to the bullets to give them velocity and direction.

bulletScript.cs

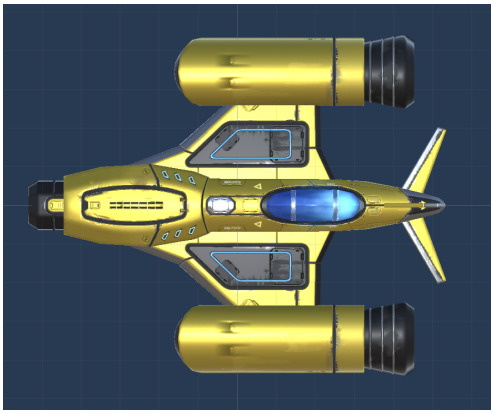
```
public class bulletScript : MonoBehaviour
{
    Rigidbody rb;
    public float bulletSpeed = 40;
    Vector3 velocity;
    void Start()
    {
        rb = gameObject.GetComponent<Rigidbody>();
        velocity = new Vector3(bulletSpeed,0,0);
        rb.velocity = velocity;
    }

    private void OnCollisionEnter(Collision collision)
    {
        Destroy(gameObject);
    }

    private void OnBecameInvisible()
    {
        Destroy(gameObject);
    }
}
```

fleetBulletScript.cs

```
using UnityEngine;

public class fleetBulletScript : MonoBehaviour
{
```

```
    Rigidbody rb;
    public float bulletSpeed = 40;
    Vector3 velocity;
    // Start is called before the first frame update
    void Start()
    {
        rb = gameObject.GetComponent<Rigidbody>();
        velocity = new Vector3(-bulletSpeed, 0, 0);
        rb.velocity = velocity;
    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnCollisionEnter(Collision collision)
    {
        Destroy(gameObject);
    }

    private void OnBecameInvisible()
    {
        Destroy(gameObject);
    }
}
```

ENEMY SPACESHIP



For the Enemy Spaceship I used a prefab from the asset store. Star Sparrow Spaceship by Ebal Studios. The enemy Spaceship comes in a fleet of 5, when the wave level is a multiple of 10. The fleet shoots bullets randomly at the player. I used a Fleet deployer object to deploy the fleet and added a script to the spaceship prefab to add speed and randomise the fire of the bullets.

FleetDeployer.cs

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

```
public class FleetDeployer : MonoBehaviour
{
    public GameObject fleetShip;
    List<GameObject> ships;
    private static bool allShipsDestroyed = false;
    void Start()
    {
        ships = new List<GameObject>();
        for (int i = 0; i < 5; i++)
        {
            ships.Add(Instantiate(fleetShip));
        }
        ships[0].transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.1f, 0.5f, 45f));
        ships[1].transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.15f, 0.3f, 45f));
        ships[2].transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.15f, 0.7f, 45f));
        ships[3].transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.2f, 0.9f, 45f));
        ships[4].transform.position =
Camera.main.ViewportToWorldPoint(new Vector3(1.2f, 0.1f, 45f));

    }

    // Update is called once per frame
    void Update()
    {
        if (allShipsDestroyed)
        {
            asteroidDeployer.incrementWaveLevel();
            Destroy(gameObject);
        }
    }

    public static void setAllShipsDestroyed(bool val)
    {
        allShipsDestroyed = val;
    }
}
```

FleetShipScript.cs

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

public class fleetShipScript : MonoBehaviour
```

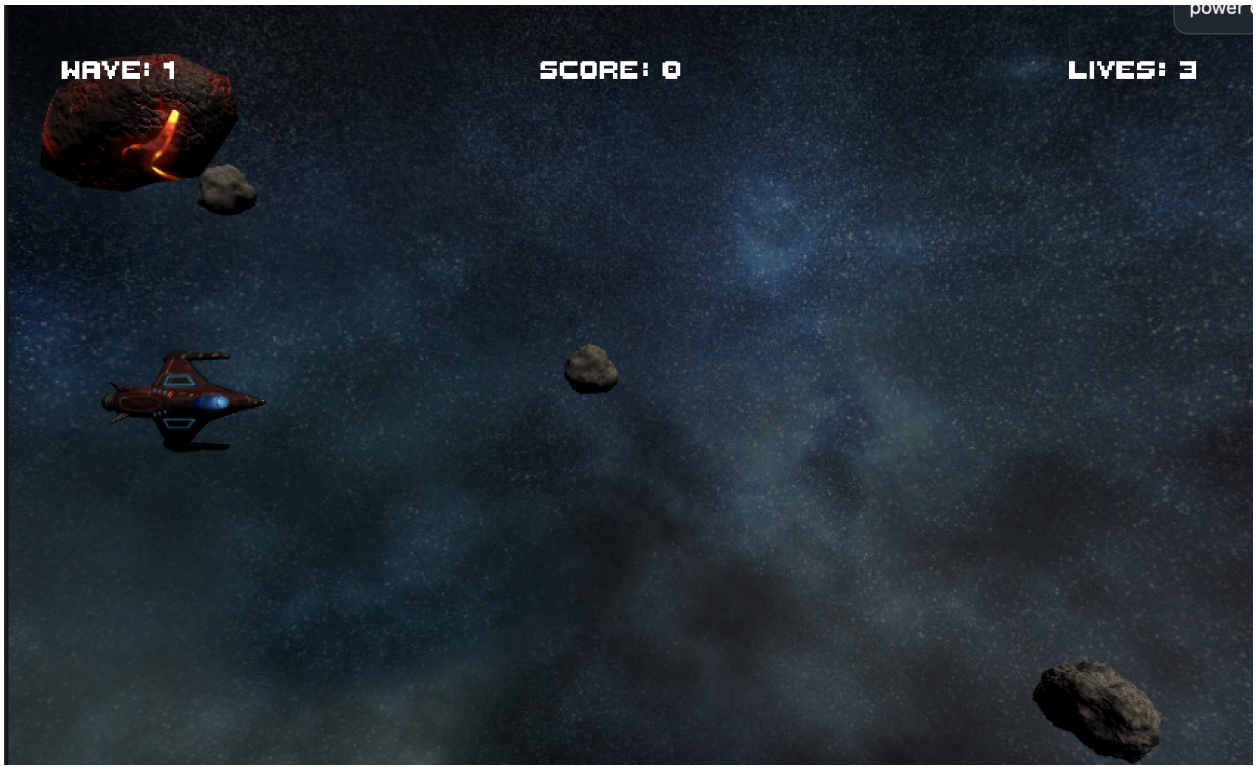
```
{
    public GameObject bulletReference;
    private Rigidbody rb;
    private bool canShoot;
    private float health;
    private static int numberOfShips;
    void Start()
    {
        canShoot = true;
        health = 150;
        numberOfShips = 5;
        rb = gameObject.GetComponent<Rigidbody>();
    }

    // Update is called once per frame
    void Update()
    {
        transform.position = transform.position + Vector3.left *
0.05f;
        if (canShoot)
        {
            canShoot = false;
            shootBullet();
            StartCoroutine(bulletDelay());
        }

        if (Camera.main.WorldToViewportPoint(transform.position).x <
-0.1)
        {
            numberOfShips--;
            if (numberOfShips == 0)
            {
                FleetDeployer.setAllShipsDestroyed(true);

            }
            Destroy(gameObject);
        }

    }

    private void shootBullet()
    {
        GameObject bullet = GameObject.Instantiate(bulletReference);
        bullet.transform.position = transform.position +
Vector3.left*7.8f;
    }

    IEnumerator bulletDelay() {
```

```
        yield return new WaitForSeconds(Random.Range(0.2f, 3));
        canShoot = true;
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.name == "Bullet(Clone)")
        {
            health -= 10f;
            if(health <= 0)
            {
                asteroidDeployer.score += 300;
                numberOfShips--;
                if(numberOfShips == 0)
                {
                    FleetDeployer.setAllShipsDestroyed(true);

                }
                Destroy(gameObject);
            }
        }

    }
}
```

GUI

The GUI of the game includes the main menu ie. Start and Quit buttons and the gameplay UI. The gameplay UI consists of the wave level, score, and player lives. To

update these values I used a gui script.



GUIScript.cs

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

public class GUIScript : MonoBehaviour
{
    public Text waveText, scoreText, livesText, fleetWave, gameOver;
    private int waveLevel, score, lives;
    private bool fleetTextFlashed;
    public static bool isGameOver, DestroyCanvas;
    public GameObject mainMenu;
    // Start is called before the first frame update
    void Start()
    {
        fleetTextFlashed = false;
        isGameOver = false;
        DestroyCanvas = false;
        fleetWave.enabled = false;
        gameOver.enabled = false;

    }
```

```
    // Update is called once per frame
    void Update()
    {
        if (isGameOver)
        {
            gameOver.enabled = true;
            StartCoroutine(mainMenuDelay());

        }
        score = asteroidDeployer.score;
        waveLevel = asteroidDeployer.getWaveLevel();
        lives = shipController.playerHealth;
        waveText.text = "wave: " + waveLevel;
        scoreText.text = "score: " + score;
        livesText.text = "lives: " + lives;
        if (waveLevel % 10 == 0 && fleetTextFlashed == false) {
            StartCoroutine(flashFleet());
        }
        if (waveLevel % 10 != 0)
        {
            fleetTextFlashed = false;
        }
    }

    public IEnumerator flashFleet()
    {
        fleetTextFlashed = true;
        for (int i = 0; i < 5; i++)
        {
            fleetWave.enabled = true;
            yield return new WaitForSeconds(.5f);
            fleetWave.enabled = false;
            yield return new WaitForSeconds(.5f);
        }

    }

    public IEnumerator mainMenuDelay()
    {
        yield return new WaitForSeconds(6);
        DestroyCanvas = true;
      SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
- 1);

    }

}
```

SCENES

The game has two scenes, the main menu and the gameplay. Each scene has its own UI and background. To make the background scroll I used a scroll script on the space skybox that I got from Star Sparrow Spaceship Asset Pack.

Scroll.cs

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

public class skyScroll : MonoBehaviour
{
    public float speed = 1.0f;

    void Update()
    {
        RenderSettings.skybox.SetFloat("_Rotation", Time.time *
speed);
    }
}
```

DIFFICULTIES

Some difficulties i encountered:

- I had difficulty getting the sound to work in the game.
- Sky scroll looks a bit off because I used a sky box instead of a wrapped sprite.

HINDSIGHT

In hindsight, in this Unity Game I would have used less scripts as I tended to use more scripts than necessary. I would also improve the instantiation of the gameObjects as this seemed to cause many problems when creating the game.