GameManager.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class GameManager : MonoBehaviour

{

    //Integer member variable called "currentGameLevel"

    public int currentGameLevel;

    public GameObject asteroidPrefab;

    public GameObject spacefighter;

    // Start is called before the first frame update

    void Start()

    {

        //Create a new player spaceship

        CreatePlayerSpaceship();

        //Set the current game level to 0

        currentGameLevel = 0;

        /\*Camera is positioned at 0,30,0

         \* Facing towards 0,0,0 with 0,0,1 as its 'up' axis \*/

        Camera.main.transform.position = new Vector3(0, 30, 0);

        Camera.main.transform.LookAt(new Vector3(0, 0, 0), new Vector3(0, 0, 1));

        StartNextLevel();

    }

    // Update is called once per frame

    void Update()

    {

    }

    void StartNextLevel()

    {

        //Increment the current game level

        currentGameLevel++;

        //Number of asteroids depends on game level

        int numberOfAsteroids = currentGameLevel \* 5;

        //instantiate a set of asteroids towards the edges of the visible screen using a for loop

        for (int i = 0; i < numberOfAsteroids; i++)

        {

            GameObject asteroid = GameObject.Instantiate(asteroidPrefab);

            //scale the asteroid to a random size between 0.2 and 0.35

            asteroid.transform.localScale = new Vector3(Random.Range(0.1f, 0.17f), Random.Range(0.1f, 0.17f), Random.Range(0.1f, 0.17f));

        }

    }

    /\* Method which instantiates the player ship in the middle of the screen\*/

    void CreatePlayerSpaceship()

    {

        //Create a new player spaceship

        spacefighter = GameObject.Instantiate(spacefighter);

        //Set the player spaceship's position to the center of the screen

        spacefighter.transform.position = new Vector3(0, 0, 0);

        //Scale the player spaceship to a size of 0.2

        spacefighter.transform.localScale = new Vector3(0.2f, 0.2f, 0.2f);

    }

}

Asteroid.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Asteroid : MonoBehaviour

{

    public GameObject asteroidObject;

    public GameObject smallAsteroidPrefab;

    private Vector3 spawnPoint;

    private bool ignoreCollisions = true;

    // Start - called before the first frame update

    void Start()

    {

        //Set the asteroid's position at a random position near the edges of the screen

        if (Random.Range(0, 2) == 0)

        {

            //Spawn on top or bottom

            if (Random.Range(0, 2) == 0)

            {

                //Spawn on top

                spawnPoint = new Vector3(Random.Range(-30f, 30f), 0, 30);

            }

            else

            {

                //Spawn on bottom

                spawnPoint = new Vector3(Random.Range(-30f, 30f), 0, -30);

            }

        }

        else

        {

            //Spawn on left or right

            if (Random.Range(0, 2) == 0)

            {

                //Spawn on left

                spawnPoint = new Vector3(-30, 0, Random.Range(-30f, 30f));

            }

            else

            {

                //Spawn on right

                spawnPoint = new Vector3(30, 0, Random.Range(-30f, 30f));

            }

        }

        //Set the asteroid's position

        asteroidObject.transform.position = spawnPoint;

        //Move the asteroid in a random direction

        asteroidObject.GetComponent<Rigidbody>().AddForce(new Vector3(Random.Range(-700f, 700f), 0, Random.Range(-700f, 700f)));

        //Rotate the asteroid in a random direction

        asteroidObject.GetComponent<Rigidbody>().AddTorque(new Vector3(Random.Range(-500f, 500f), Random.Range(-500f, 500f), Random.Range(-500f, 500f)));

        //Wrap asteroids to other side of screen, check every 0.2 seconds. 5 times a second

        InvokeRepeating("CheckIfOffScreen", 0.2f, 0.2f);

        //This is a method that disables collisions for a tenth of a second at spawn in, in order to prevent not valid collisions

        Invoke("DisableCollisionIgnore", 0.1f);

    }

    void DisableCollisionIgnore()

    {

        //Disabling collision ignore boolean

        ignoreCollisions = false;

    }

    void CheckIfOffScreen()

    {

        //Check if the asteroid is off screen, and if so, wrap it to the other side

        Vector3 currentWorldPos = asteroidObject.transform.position;

        Vector3 viewPosition = Camera.main.WorldToViewportPoint(currentWorldPos);

        if (viewPosition.x > 1f)

        {

            asteroidObject.transform.position = new Vector3(-currentWorldPos.x + 1, 0, currentWorldPos.z);

        }

        if (viewPosition.y < 0f)

        {

            asteroidObject.transform.position = new Vector3(currentWorldPos.x, 0, -currentWorldPos.z - 1);

        }

        if (viewPosition.x < 0f)

        {

            asteroidObject.transform.position = new Vector3(-currentWorldPos.x - 1, 0, currentWorldPos.z);

        }

        if (viewPosition.y > 1f)

        {

            asteroidObject.transform.position = new Vector3(currentWorldPos.x, 0, -currentWorldPos.z + 1);

        }

    }

    // Update is called once per frame

    void Update()

    {

    }

    /\*Each time an asteroid collides with something, spawn a few of the tiny asteroid prefabs at the point of

impact. They should be destroyed shortly afterwards. \*/

    void SpawnCollisionDebris(Vector3 *collisionPoint*)

    {

        //Spawn 3 small asteroids at the point of collision

        for (int i = 0; i < 3; i++)

        {

            GameObject smallAsteroid = GameObject.Instantiate(smallAsteroidPrefab);

            //Setting position to the collision point and scaling it down

            smallAsteroid.transform.position = *collisionPoint*;

            smallAsteroid.transform.localScale = new Vector3(0.01f, 0.01f, 0.01f);

            //Adding a random force and torque to the small asteroids

            smallAsteroid.GetComponent<Rigidbody>().AddForce(new Vector3(Random.Range(-100f, 100f), 0, Random.Range(-100f, 100f)));

            smallAsteroid.GetComponent<Rigidbody>().AddTorque(new Vector3(Random.Range(-100f, 100f), Random.Range(-100f, 100f), Random.Range(-100f, 100f)));

            //Destroying the small asteroids after 1.5 seconds

            Destroy(smallAsteroid, 1.5f);

        }

    }

    /\*Method for calling SpawnCollisionDebris on collisions \*/

    void OnCollisionEnter(Collision *collision*)

    {

        //Checking if it is on spawn in, and if so, ignore collisions

        if (ignoreCollisions)

        {

            return;

        }

        //Calling SpawnCollisionDebris with the point of collision

        SpawnCollisionDebris(*collision*.contacts[0].point);

    }

}

SpaceShip.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Spaceship : MonoBehaviour

{

    public GameObject spaceship;

    // Start is called before the first frame update

    void Start()

    {

        //Wrap spaceship to other side of screen, check every 0.2 seconds. 5 times a second

        InvokeRepeating("CheckIfOffScreen", 0.2f, 0.2f);

    }

    // Update is called once per frame

    void Update()

    {

        /\*apply a physics force to accelerate the spaceship forward if the Up arrow is held, or

        rotate it left/right if the Left/Right arrows are held.\*/

        //Checking if the Up arrow is held, if so check if within velocity limit, if so add force

        if (Input.GetKey(KeyCode.UpArrow) && GetComponent<Rigidbody>().velocity.magnitude < 14)

        {

            GetComponent<Rigidbody>().AddForce(transform.up \* 7);

        }

        if (Input.GetKey(KeyCode.LeftArrow))

        {

            GetComponent<Rigidbody>().AddTorque(transform.forward \* -4);

        }

        if (Input.GetKey(KeyCode.RightArrow))

        {

            GetComponent<Rigidbody>().AddTorque(transform.forward \* 4);

        }

    }

    // Having the player spaceship respond to moving off-screen, in the same way that asteroids already do

    void CheckIfOffScreen()

    {

        Vector3 currentWorldPos = spaceship.transform.position;

        Vector3 viewPosition = Camera.main.WorldToViewportPoint(currentWorldPos);

        if (viewPosition.x > 1f)

        {

            spaceship.transform.position = new Vector3(-currentWorldPos.x + 1, 0, currentWorldPos.z);

        }

        if (viewPosition.y < 0f)

        {

            spaceship.transform.position = new Vector3(currentWorldPos.x, 0, -currentWorldPos.z - 1);

        }

        if (viewPosition.x < 0f)

        {

            spaceship.transform.position = new Vector3(-currentWorldPos.x - 1, 0, currentWorldPos.z);

        }

        if (viewPosition.y > 1f)

        {

            spaceship.transform.position = new Vector3(currentWorldPos.x, 0, -currentWorldPos.z + 1);

        }

    }

}