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;

    // Start is called before the first frame update

    void Start()

    {

        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(0.2f, 0.35f), Random.Range(0.2f, 0.35f), Random.Range(0.2f, 0.35f));

        }

    }

}

Asteroid.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Asteroid : MonoBehaviour

{

    public GameObject asteroidObject;

    private Vector3 spawnPoint;

    // 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);

    }

    void CheckIfOffScreen()

    {

        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()

    {

    }

}