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

public class GameManager : MonoBehaviour
{
    public GameObject camera;
    private int currentGameLevel = 1;
    public GameObject asteroidPrefab;
    public GameObject PlayerSpaceship;
    // Start is called before the first frame update
    void Start()
    {
        asteroidPrefab.transform.localScale = Vector3.one / 3;


        Vector3 cameraPosition = new Vector3(0f, 200f, 0f);
        camera.transform.position = cameraPosition;

        // Set the camera's rotation to look at the target with the specified up axis
        Vector3 lookAtTarget = new Vector3(0f, 0f, 0f);
        Vector3 upAxis = new Vector3(0f, 0f, 1f);

        camera.transform.LookAt(lookAtTarget, upAxis);
        StartNextLevel();

        CreatePlayerSpaceship();
    }

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

    }

    public void StartNextLevel()
    {
        currentGameLevel++;
        int numberOfAsteroids = currentGameLevel * 5;
        Vector3 randomSpawnPosition = new Vector3(0f,0f,0f);

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

            if (Random.Range(0,1) > 0.51)
            {
                randomSpawnPosition = new Vector3(0f, 0f, 190f);
            }else
            randomSpawnPosition = new Vector3(190f, 0f, 0f);





          GameObject asteroid = GameObject.Instantiate(asteroidPrefab,
randomSpawnPosition, Quaternion.identity);

          asteroid.transform.localScale = new Vector3(Random.Range(0.2f, 0.35f),
            Random.Range(0.2f, 0.35f), Random.Range(0.2f, 0.35f));
      }
   }


   private void CreatePlayerSpaceship()
   {
      PlayerSpaceship.transform.position = Vector3.zero;
      PlayerSpaceship.transform.Rotate(90f, 0f, 0f, Space.Self);
   }







   private void CreateAsteroidField()
   {
      Vector3 randomSpawnPosition = new Vector3(Random.Range(-50, 51), 0,
Random.Range(-50,51));
      GameObject newAsteroid = Instantiate(asteroidPrefab, randomSpawnPosition,
Quaternion.identity);

   }




}

using UnityEngine;
using System.Collections;

public class Asteroid : MonoBehaviour
{
   public float speed = 10f;
   private Rigidbody rb;
   private Camera mainCamera;
```

```
    public GameObject smallAsteroidPrefab;
    public int smallAsteroidsToSpawn = 3;
    public float destructionDelay = 2f;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        mainCamera = Camera.main;

        Vector3 randomDirection = new Vector3(Random.Range(-1f, 1f), 0f,
Random.Range(-1f, 1f)).normalized;
        rb.velocity = randomDirection * speed;
        rb.useGravity = false;
        rb.constraints = RigidbodyConstraints.FreezePositionY |
RigidbodyConstraints.FreezeRotation;

        StartCoroutine(CheckEdgesCoroutine());
    }
    void OnCollisionEnter(Collision collision)
    {
        SpawnSmallAsteroids(collision.contacts[0].point);
    }
    void SpawnSmallAsteroids(Vector3 spawnPosition)
    {
        for (int i = 0; i < smallAsteroidsToSpawn; i++)
        {
            GameObject smallAsteroid = Instantiate(smallAsteroidPrefab, spawnPosition,
Quaternion.identity);
            Destroy(smallAsteroid, destructionDelay);  // Destroy the small asteroid after the
specified delay
        }
    }

    IEnumerator CheckEdgesCoroutine()
    {
        while (true)
        {
            yield return new WaitForSeconds(0.2f);  // Check 5 times per second
            CheckEdges();
        }
    }

    void CheckEdges()
    {
        Camera cam = Camera.main;
        Vector3 viewportPosition = cam.WorldToViewportPoint(transform.position);
        Vector3 newPosition = transform.position;
        float distanceFromCamera = cam.transform.position.y - transform.position.y;

        if (viewportPosition.x < 0 || viewportPosition.x > 1)
        {
            newPosition = cam.ViewportToWorldPoint(new Vector3(1 - viewportPosition.x,
viewportPosition.y, distanceFromCamera));
        }

        if (viewportPosition.y < 0 || viewportPosition.y > 1)
        {
            newPosition = cam.ViewportToWorldPoint(new Vector3(viewportPosition.x, 1 -
viewportPosition.y, distanceFromCamera));
        }

        newPosition.y = 0;
        transform.position = newPosition;
    }



}

using System.Collections;
using UnityEngine;

[RequireComponent(typeof(Rigidbody))]
public class Spaceship : MonoBehaviour
{
    public float thrust = 10f;  // Adjust the thrust force
    public float rotationSpeed = 10f;  // Adjust the rotation speed
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();

        StartCoroutine(CheckEdgesCoroutine());



        CheckEdges();
    }

    void Update()
    {
        HandleMovement();
    }
    void HandleMovement()
    {
        if (Input.GetKey(KeyCode.UpArrow))
        {
            // Apply force to accelerate the spaceship forward
            rb.AddForce(transform.up * thrust, ForceMode.Force);
        }
        if (Input.GetKey(KeyCode.DownArrow))
        {
            // Apply force to accelerate the spaceship forward
            rb.AddForce(transform.up * -thrust, ForceMode.Force);
        }

        if (Input.GetKey(KeyCode.LeftArrow))
        {
            // Rotate the spaceship to the left
            rb.AddTorque(0f, -rotationSpeed, 0f, ForceMode.Force);
        }

        if (Input.GetKey(KeyCode.RightArrow))
        {
            // Rotate the spaceship to the right
            rb.AddTorque(0f, rotationSpeed, 0f, ForceMode.Force);
        }
    }

    void CheckEdges()
    {
        Camera cam = Camera.main;
        Vector3 viewportPosition = cam.WorldToViewportPoint(transform.position);
        Vector3 newPosition = transform.position;

        float distanceFromCamera = cam.transform.position.y - transform.position.y;

        if (viewportPosition.x < 0 || viewportPosition.x > 1)
        {
            newPosition = cam.ViewportToWorldPoint(new Vector3(1 - viewportPosition.x,
viewportPosition.y, distanceFromCamera));
        }

        if (viewportPosition.y < 0 || viewportPosition.y > 1)
        {
            newPosition = cam.ViewportToWorldPoint(new Vector3(viewportPosition.x, 1 -
viewportPosition.y, distanceFromCamera));
        }

        newPosition.y = 0;
        transform.position = newPosition;
    }

    IEnumerator CheckEdgesCoroutine()
    {
        while (true)
        {
            yield return new WaitForSeconds(0.2f);  // Check 5 times per second
            CheckEdges();
        }
    }
}
```