Assignment 4 – Asteroids Game

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

public class GameManager : MonoBehaviour
{
    public int currentGameLevel = 1; // Set the starting game level
    public GameObject asteroidPrefab; // Reference to the asteroid prefab

    void Start()
    {
        // Set the camera's position
        Camera.main.transform.position = new Vector3(0f, 30f, 0f);
        Camera.main.transform.LookAt(new Vector3(0f, 0f, 0f), Vector3.up);

        // Call the method to start a new level
        StartNewLevel();
    }

    // Method to start a new game level
    void StartNewLevel()
    {
        // Calculate the number of asteroids based on the current game level
        int numAsteroids = currentGameLevel * 3;

        for (int i = 0; i < numAsteroids; i++)
        {
            // Generate a random spawn position within the screen boundaries
            Vector3 spawnPosition = new Vector3(Random.Range(-15f, 15f), 0f,
Random.Range(-15f, 15f));

            // Ensure the Y position is at ground level (0)
            spawnPosition.y = 0f;

            // Add a buffer to the Z position to prevent immediate wrap-around
            spawnPosition.z += 2f;

            // Instantiate asteroid
            Instantiate(asteroidPrefab, spawnPosition, Quaternion.identity);
        }
    }
}
```

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

public class Asteroid : MonoBehaviour
{
    Rigidbody rb;
    public float moveSpeed = 5f;

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

        // Calculate a random direction vector on the XZ plane
        Vector3 randomDirection = Random.onUnitSphere;
        randomDirection.y = 0f;
        // Set the initial velocity based on your moveSpeed
        rb.velocity = randomDirection * moveSpeed;

        // Generate random torque (angular velocity)
        Vector3 randomTorque = new Vector3(
            Random.Range(5f, 15f),
            Random.Range(5f, 15f),
            Random.Range(5f, 15f)
        );
        InvokeRepeating("CheckScreenEdges", 0f, 0.2f);
    }

    void CheckScreenEdges()
    {
        Debug.Log("Current position: " + transform.position);
        // Check if the asteroid has left the screen
        if (Mathf.Abs(transform.position.x) > 25f ||
Mathf.Abs(transform.position.z) > 20f)
        {
            // Wrap around to the opposite side
            transform.position = new Vector3(-transform.position.x, 0, -
transform.position.z);
        }
    }
}
```