CT3536 Unity3D Lab 5
Sample Solution (code)

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

public class GameManager : MonoBehaviour {

    // inspector settings
    public GameObject asteroidPrefab, spaceshipPrefab, fragmentPrefab;
    //

    // class-level statics
    public static GameManager instance;
    public static int currentGameLevel;
    public static Vector3 screenBottomLeft, screenTopRight;
    public static float screenWidth, screenHeight;
    //

    // Use this for initialization
    void Start () {
        instance = this;
        Camera.main.transform.position = new Vector3 (0f, 30f, 0f);
        Camera.main.transform.LookAt (Vector3.zero, new Vector3 (0f, 0f, 1f));
        currentGameLevel = 0;

        StartNextLevel ();
        CreatePlayerSpaceship ();
    }

    public static void StartNextLevel() {
        currentGameLevel++;

        // find (slightly expanded) screen corners and size, in world coordinates
        // for ViewportToWorldPoint, the z value specified is in world units from the camera
        screenBottomLeft = Camera.main.ViewportToWorldPoint(new Vector3(-0.1f,-0.1f,30f));
        screenTopRight = Camera.main.ViewportToWorldPoint (new Vector3(1.1f,1.1f,30f));
        screenWidth = screenTopRight.x - screenBottomLeft.x;
        screenHeight = screenTopRight.z - screenBottomLeft.z;
        Debug.Log ("BottomLeft: "+screenBottomLeft);
        Debug.Log ("TopRight: "+screenTopRight);
        Debug.Log ("Width: " + screenWidth);
        Debug.Log ("Height: " + screenHeight);

        // instantiate some asteroids near the edges of the screen
        for (int i = 0; i < currentGameLevel * 2 + 3; i++) {
            GameObject go = Instantiate (instance.asteroidPrefab) as GameObject;
            float x, z;
            if (Random.Range (0f, 1f) < 0.5f)
                x = screenBottomLeft.x + Random.Range (0f, 0.15f) * screenWidth; // near the left
edge
            else
                x = screenTopRight.x - Random.Range (0f, 0.15f) * screenWidth; // near the right
edge
            if (Random.Range (0f, 1f) < 0.5f)
                z = screenBottomLeft.z + Random.Range (0f, 0.15f) * screenHeight; // near the
bottom edge
            else
                z = screenTopRight.z - Random.Range (0f, 0.15f) * screenHeight; // near the top
edge
            go.transform.position = new Vector3(x, 0f, z);
        }

    }

    private static void CreatePlayerSpaceship() {
```

```
        // instantiate the player's spaceship
        GameObject go = Instantiate (instance.spaceshipPrefab);
        go.transform.position = Vector3.zero;
    }
}
```

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

public class Asteroid : MonoBehaviour {

    // inspector settings
    public Rigidbody rigidBody;
    //

    // Use this for initialization
    void Start () {
        // randomise size+mass
        transform.localScale = new Vector3(Random.Range(0.08f,0.12f), Random.Range(0.08f,0.12f),
Random.Range(0.08f,0.12f));
        rigidBody.mass = transform.localScale.x * transform.localScale.y *
transform.localScale.z;

        // randomise velocity
        rigidBody.velocity = new Vector3 (Random.Range (-10f, 10f), 0f, Random.Range (-10f,
10f));
        rigidBody.angularVelocity = new Vector3 (Random.Range (-4f, 4f), Random.Range (-4f, 4f),
Random.Range (-4f, 4f));
    }

   private void OnCollisionEnter(Collision collision) {
      // spawn fragments (the fragment prefab has no collider and has no gameplay effect)
      float colSpeed = collision.relativeVelocity.magnitude;
      int num =  Mathf.Clamp( Mathf.RoundToInt(colSpeed), 2, 8);
      for (int i=0; i<num; i++) {
         GameObject go = Instantiate(GameManager.instance.fragmentPrefab);
         go.transform.position = collision.contacts[0].point;
         Rigidbody r = go.GetComponent<Rigidbody>();
         r.velocity = new Vector3 (Random.Range (-colSpeed, colSpeed), 0f, Random.Range (-
colSpeed, colSpeed));
         r.angularVelocity = new Vector3 (Random.Range (-4f, 4f), Random.Range (-4f, 4f),
Random.Range (-4f, 4f));
      }
   }

}
```

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

public class ScreenEdgeChecker : MonoBehaviour {

    // inspector settings
    public Rigidbody rigidBody;
    //

    // Use this for initialization
    void Start () {
        // start periodically checking for being off-screen
        InvokeRepeating ("CheckScreenEdges", 0.1f, 0.1f);
    }

    private void CheckScreenEdges() {
```

```
        Vector3 pos = transform.position;
        Vector3 vel = rigidBody.velocity;
        float xTeleport = 0f, zTeleport = 0f;

        if (pos.x < GameManager.screenBottomLeft.x && vel.x <= 0f)
            xTeleport = GameManager.screenWidth;
        else if (pos.x > GameManager.screenTopRight.x && vel.x >= 0f)
            xTeleport = -GameManager.screenWidth;

        if (pos.z < GameManager.screenBottomLeft.z && vel.z <= 0f)
            zTeleport = GameManager.screenHeight;
        else if (pos.z > GameManager.screenTopRight.z && vel.z >= 0f)
            zTeleport = -GameManager.screenHeight;

        if (xTeleport != 0f || zTeleport != 0f)
            transform.position = new Vector3 (pos.x + xTeleport, 0f, pos.z + zTeleport);

    }
}
```

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

public class Spaceship : MonoBehaviour {

    private Rigidbody rigid;

    // Use this for initialization
    void Start () {
        rigid = GetComponent<Rigidbody>();
    }

    void FixedUpdate () {
        if (Input.GetKey(KeyCode.UpArrow))
            rigid.AddForce(transform.forward * (rigid.mass * Time.fixedDeltaTime * 2000f));

        // we're using an Angular Drag of 15.0 on the rigid body, so need a lot of torque here
        if (Input.GetKey(KeyCode.LeftArrow))
            rigid.AddTorque(-Vector3.up * (rigid.mass * Time.fixedDeltaTime * 4000f));
        else if (Input.GetKey(KeyCode.RightArrow))
            rigid.AddTorque(Vector3.up * (rigid.mass * Time.fixedDeltaTime * 4000f));
    }
}
```

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

public class AutoDestroy : MonoBehaviour
{
    public float minLifetime, maxLifetime;

    void Start() {
        StartCoroutine( HandleLifetime() );
    }

    private IEnumerator HandleLifetime() {
        yield return new WaitForSeconds(Random.Range(minLifetime, maxLifetime));
        Destroy(gameObject);
    }
```

```
}
```