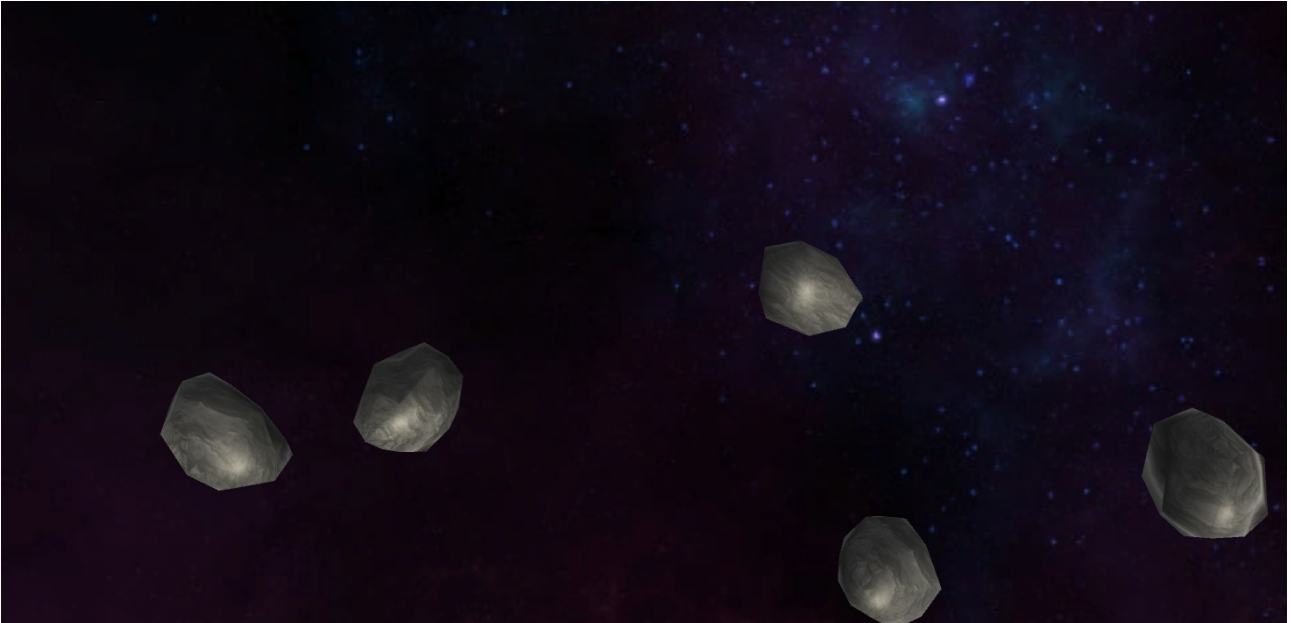


CT3536 Games Programming (Unity3D) Lab 4 Sample Solution (code)



Note that I used a MeshCollider for the asteroids, since that scales correctly with non-uniform x-y-z scaling of the Transform, whereas a SphereCollider always remains spherical.

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

public class GameManager : MonoBehaviour {

    // inspector settings
    public GameObject asteroidPrefab;
    //

    // 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;

        // find 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(0f,0f,30f));
        screenTopRight = Camera.main.ViewportToWorldPoint (new Vector3(1f,1f,30f));
        screenWidth = screenTopRight.x - screenBottomLeft.x;
        screenHeight = screenTopRight.z - screenBottomLeft.z;

        StartNextLevel ();
    }

    public static void StartNextLevel() {
        currentGameLevel++;
    }
}
```

```

// create 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);
}
}
}

```

```

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(0.06f,0.09f), Random.Range(0.06f,0.09f), Random.Range(0.06f,0.09f));
        rigidBody.mass = transform.localScale.x * transform.localScale.y * transform.localScale.z;

```

```

        // randomise velocity
        rigidBody.velocity = new Vector3 (Random.Range (-20f, 20f), 0f, Random.Range (-20f, 20f));
        rigidBody.angularVelocity = new Vector3 (Random.Range (-20f, 20f), Random.Range (-20f, 20f), Random.Range (-20f, 20f));

```

```

        // start periodically checking for being off-screen
        InvokeRepeating ("CheckScreenEdges", 0.2f, 0.2f);
    }

```

```

    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) // velocity check as sanity test
            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);
    }
}

```