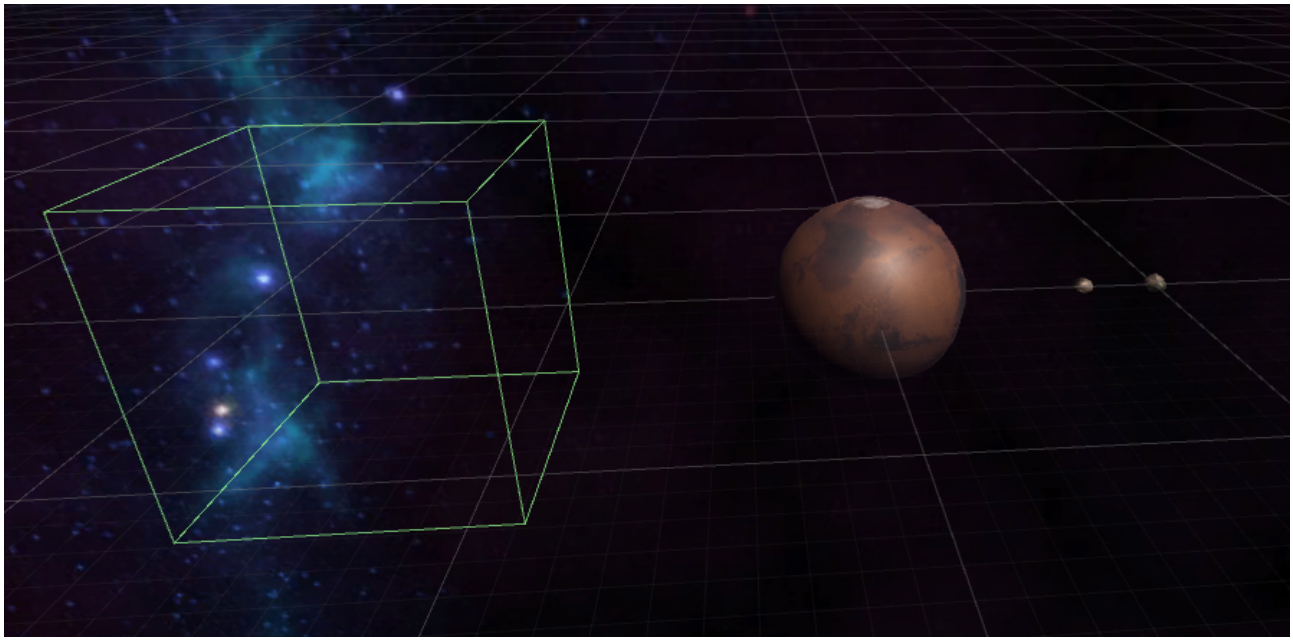
CT3536 Unity3D Lab 3
Sample Solution (code)



Note that the SphereCollider components which were attached to Mars, Phobos, Deimos, and the Asteroid prefab are all set to be Triggers rather than physical colliders

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

public class GameManager : MonoBehaviour {
    // inspector values
    public GameObject mars;
    public GameObject phobos, deimos;
    public GameObject asteroidPrefab;
    public BoxCollider asteroidSpawnRegion;
    //

    public static GameManager instance;

    // Use this for initialization
    void Start () {
        Camera.main.transform.position = new Vector3 (0f, 0f, -200f);
        Camera.main.transform.LookAt(mars.transform);
        mars.GetComponent<Rigidbody> ().AddTorque (new Vector3 (0f, 20f, 0f));
        instance = this;
    }

    // Update is called once per frame
    void Update () {
        phobos.transform.RotateAround (Vector3.zero, Vector3.up, 3f * Time.deltaTime);
        deimos.transform.RotateAround (Vector3.zero, Vector3.up, 2f * Time.deltaTime);

        // NB we are using the camera's own coordinate system (rather than the global co
ordinate system) to specify the axis of rotation
        if (Input.GetKey (KeyCode.LeftArrow))
            Camera.main.transform.RotateAround (Vector3.zero, Camera.main.transform.up,
50f * Time.deltaTime);
        else if (Input.GetKey (KeyCode.RightArrow))
            Camera.main.transform.RotateAround (Vector3.zero, Camera.main.transform.up,
-50f * Time.deltaTime);
        if (Input.GetKey (KeyCode.UpArrow))
            Camera.main.transform.RotateAround (Vector3.zero, Camera.main.transform.righ
```

```
t, 50f * Time.deltaTime);
        else if (Input.GetKey (KeyCode.DownArrow))
            Camera.main.transform.RotateAround (Vector3.zero, Camera.main.transform.righ
t, -50f * Time.deltaTime);

        // maybe spawn a new asteroid (one every three seconds, on average)
        if (Random.Range (0f, 1f) < Time.deltaTime / 3f) {
            GameObject go = GameObject.Instantiate (asteroidPrefab);
            go.transform.position = ChooseAsteroidSpawnPoint();
        }
    }

    private Vector3 ChooseAsteroidSpawnPoint() {
        return new Vector3 (Random.Range (asteroidSpawnRegion.bounds.min.x, asteroidSpaw
nRegion.bounds.max.x),
            Random.Range (asteroidSpawnRegion.bounds.min.y, asteroidSpawnRegion.bounds.m
ax.y),
            Random.Range (asteroidSpawnRegion.bounds.min.z, asteroidSpawnRegion.bounds.m
ax.z));
    }
}
```

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

public class AsteroidScript : MonoBehaviour {

    private float destroyMinX;

    // Use this for initialization
    void Start () {
        Rigidbody r = GetComponent<Rigidbody>();
        r.velocity = new Vector3 (50f+Random.Range(5f,15f), 0f, 0f);
        r.AddTorque (new Vector3 (Random.Range(5f,15f),Random.Range(5f,15f),Random.Range
(5f,15f)));
        InvokeRepeating ("CheckForOffscreen", 3f, 1f);
        destroyMinX = Mathf.Abs(transform.position.x); // transform.position.x will be a
 negative value
        Debug.Log ("Asteroid spawned: " + GetInstanceID() + " with destroyMinX="+destroy
MinX);
    }

    void OnTriggerEnter(Collider other) {
        if (other.name != "SpawnRegion") {
            Debug.Log ("Asteroid destroyed by collision: " + GetInstanceID ());
            GameObject.Destroy (this.gameObject);
        }
    }

    private void CheckForOffscreen() {
        if (transform.position.x>destroyMinX) {
            Vector3 viewPos = Camera.main.WorldToViewportPoint (transform.position);
            if (viewPos.x>1f || viewPos.x<0f || viewPos.y>1f || viewPos.y<0f || viewPos.
z<0f) {
                Debug.Log ("Asteroid destroyed by going offscreen: " + GetInstanceID());
                GameObject.Destroy (this.gameObject);
            }
        }
    }
}
```