Name: Andrew Hayes
E-mail: a.hayes18@universityofgalway.ie
ID: 21321503

CT3536

2023-10-10

Lab Assignment 05

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

public class GameManager : MonoBehaviour {

    // inspector settings
    public GameObject asteroidPrefab;
    public GameObject spaceship;

    // 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;

        CreatePlayerSpaceship();
        StartNextLevel();
    }

    public void CreatePlayerSpaceship() {
        Instantiate(spaceship);
        spaceship.transform.position = new Vector3(0, 0, 0);
    }

    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);
        }
    }
}
```

Listing 1: GameManager.cs

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

public class Asteroid : MonoBehaviour {

    // inspector settings
    public Rigidbody rigidBody;
    public GameObject miniAsteroid;

    // 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);

    }

    // method to spawn mini-asteroid fragments at the contact point(s) of a collision
    private void OnCollisionEnter(Collision collision) {
        // Arraylist to keep track of the mini asteroids created for a collision
        ArrayList fragments = new ArrayList();

        foreach (ContactPoint contact in collision.contacts) {
            // instantiating a random number of mini asteroid between 1 and 5 inclusive
            int numFragments = Random.Range(1, 5);

            for (int i = 1; i <= numFragments; i++) {
                GameObject fragment = Instantiate(miniAsteroid);
                fragment.transform.position = contact.point;
                fragments.Add(fragment);
            }
        }

        StartCoroutine(DestroyFragments(fragments));
    }

    // coroutine to destroy all the fragments from a collision
    IEnumerator DestroyFragments(ArrayList fragments) {
        yield return new WaitForSeconds(3);

        foreach (GameObject fragment in fragments) {
            Destroy(fragment);
        }
    }
}
```

Listing 2: Asteroid.cs

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

public class Spaceship : MonoBehaviour
{
    public GameObject spaceship;
    public float speed = 5.0f;
    public float rotationalSpeed = 2.0f;
    // Start is called before the first frame update
    void Start()
    {
        // start periodically checking for being off-screen
        InvokeRepeating ("CheckScreenEdges", 0.2f, 0.2f);
    }

    // Update is called once per frame
    void Update()
    {
        // move spaceship according to arrow keys
        // applying just a force to the spaceship object creates some unusual handling, but i feel that this is correct as in space there should be 0 drag, and if a force is applied in one direction, it should remain until it's cancelled out
        if (Input.GetKey(KeyCode.LeftArrow)) {
            spaceship.GetComponent<Rigidbody>().AddTorque(new Vector3(0, -rotationalSpeed, 0));
        }
        else if (Input.GetKey(KeyCode.RightArrow)) {
            spaceship.GetComponent<Rigidbody>().AddTorque(new Vector3(0, rotationalSpeed, 0));
        }
        else if (Input.GetKey(KeyCode.UpArrow)) {
            spaceship.GetComponent<Rigidbody>().AddRelativeForce(new Vector3(0, 0, speed));
        }
        else if (Input.GetKey(KeyCode.DownArrow)) {
            spaceship.GetComponent<Rigidbody>().AddRelativeForce(new Vector3(0, 0, -speed));
        }
    }

    private void CheckScreenEdges() {
        Vector3 pos = spaceship.transform.position;
        Vector3 vel = spaceship.GetComponent<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);

    }
}
```

Listing 3: `Spaceship.cs`