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

CT3536

2023-10-04

Assignment 03 – Starting a Proper Asteroids Game

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

public class GameManager : MonoBehaviour
{
    public int currentGameLevel = 1;
    public GameObject asteroid;

    // Start is called before the first frame update
    void Start()
    {
        Camera.main.transform.position = new Vector3(0, 30, 0);
        Camera.main.transform.LookAt(new Vector3(0f, 0f, 0f), new Vector3(0f, 0f, 1f));

        StartNextLevel();
    }


    // Update is called once per frame
    void Update()
    {

    }

    void StartNextLevel() {
        // assuming we want to spawn 3 asteroids plus 5 more per level
        int numberAsteroids = 3 + (5 * currentGameLevel);

        // get dimensions of screen
        float halfWidth = Camera.main.orthographicSize * Camera.main.aspect;
        float halfHeight = Camera.main.orthographicSize;

        for (int i = 0; i < numberAsteroids; i++)
        {
            float randomX;
            float randomZ;
            do {
                randomX = Random.Range(-halfWidth, halfWidth);
                randomZ = Random.Range(-halfHeight, halfHeight);
            } while ((randomX < 5 || randomX > -5) && (randomZ < 5 || randomZ > -5));

            Vector3 spawnPosition = new Vector3(randomX, 0, randomZ);
            Instantiate(asteroid, spawnPosition, Quaternion.identity);
        }
    }
}
```

Listing 1: GameManager.cs

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

public class AsteroidScript : MonoBehaviour
{
    public GameObject asteroid;
    public float maxSpeed =  300;
    public float minSpeed = -300;

    // Start is called before the first frame update
    void Start()
    {
        // adding force to the asteroid
        Vector3 forceVector = new Vector3(Random.Range(minSpeed, maxSpeed),Random.Range(minSpeed,
        ↪  maxSpeed), Random.Range(minSpeed, maxSpeed));
        asteroid.GetComponent<Rigidbody>().AddForce(forceVector);

        InvokeRepeating("CheckOffscreen", 0f, 0.2f);
    }

    // Update is called once per frame
    void Update()
    {

    }

    // method to check if asteroid has gone offscreen
    private void CheckOffscreen()
    {
        float halfWidth = Camera.main.orthographicSize * Camera.main.aspect;
        float halfHeight = Camera.main.orthographicSize;

        int padding = 5; // padding variable to account for size of asteroid

        Vector3 minBounds = new Vector3(-halfWidth, 0, -halfHeight);
        Vector3 maxBounds = new Vector3(halfWidth, 0, halfHeight);

        // if asteroid goes offscreen, wrapping around
        if (transform.position.x < minBounds.x - padding)
        {
            transform.position = new Vector3(maxBounds.x, transform.position.y,
            ↪  transform.position.z);
        }
        else if (transform.position.x > maxBounds.x + padding)
        {
            transform.position = new Vector3(minBounds.x, transform.position.y,
            ↪  transform.position.z);
        }

        if (transform.position.z < minBounds.z - padding)
```

```
        {
            transform.position = new Vector3(transform.position.x, transform.position.y,
            ↪  maxBounds.z);
        }
        else if (transform.position.z > maxBounds.z + padding)
        {
            transform.position = new Vector3(transform.position.x, transform.position.y,
            ↪  minBounds.z);
        }
    }
}
```

Listing 2: `Asteroid.cs`