By: Daniel Carvalho Marques

Programming learning journal

02/04/2019

Component 1 old money system. Made an old money system calculator which focuses on the old money system of £sd (https://en.wikipedia.org/wiki/£sd). £sd being pounds, shillings & pence.

The black bar divides the 2 Main functions of this calculator, the top half & the bottom half.

The +1 buttons next to the pounds, shillings & pence raises the corresponding pounds, shillings & pence.

For the top half of the screen the + button adds the right total money to the left total money. Whereas the – button takes away the right total money from the left total money.

For the bottom half of the screen the = button checks if the left total money is the same as the right total money. The > button checks if the left total money is greater than the right total money. The < button checks if the left total money is less than the right total money.

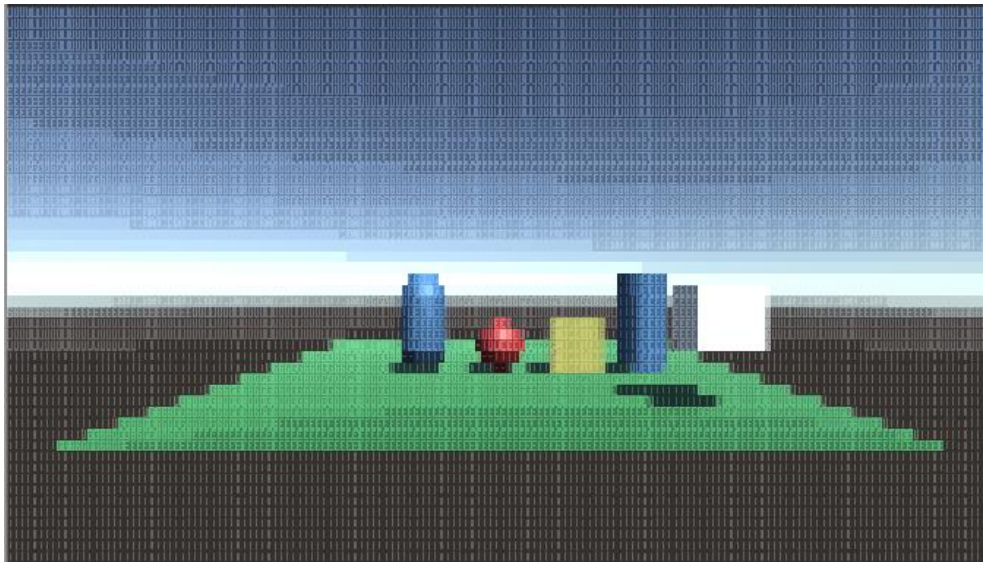
I learnt about the OR function which helped compact the script I made down as before I was making the calculator entirely with IF & ELSE statements.

Additionally, I started to use visual studio code rather than monodevelop as the code editor for unity debugging which meant that I had to learn a new software with different ways of doing things compared to what I had been doing previously e.g. monodevelop allows you to press the tab button twice after writing an if statement to general if statement which you can fill in, but in visual studio code you have to look down a dropdown list to do that.

By: Daniel Carvalho Marques

09/04/2019

Component 2 ASCII shader. Made an ASCII shader that divides the screen up into small squares and then fills the squares with ASCII art. I learnt how to create and work on shaders using unity's post processing package.



16/04/2019

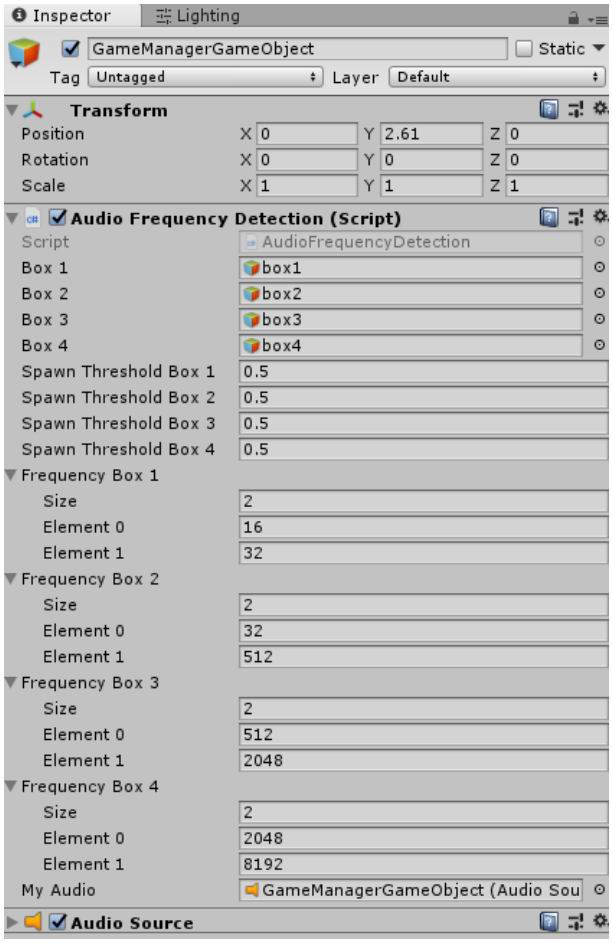
Component 3 audio frequency detector. Created an audio frequency detector that shows/hides four boxes depending on the frequency detected from the audio source playing in the scene. Each of the box's frequency ranges to detect & possible frequency thresholds are configurable in the inspector once the script is attached to an empty game object.

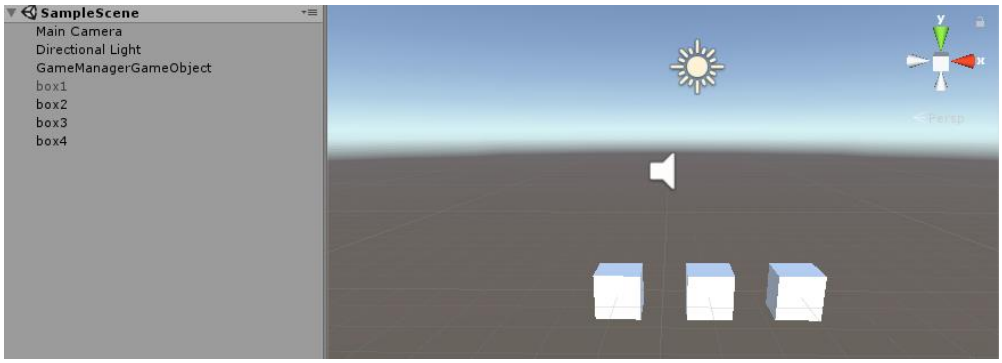
The component contains an audio frequency detection script & a free to use music file from https://soundcloud.com/aka-dj-quads.

The frequency ranges that are set for the four boxes are different sound ranges perceivable to human hearing https://en.wikipedia.org/wiki/Audio_frequency ranging from the lowest frequency range a human can hear, to almost the highest.

I learnt how to work with array loops and instantiating objects based on the array loop I created that used GetSpectrumData to find the audio frequency data of the song that was playing in the scene.

By: Daniel Carvalho Marques





23/04/2019

Component 4 pause menu. Created a pause menu that pauses both the scene and any running music (useful for the audio frequency detector). I learned to use Time.timeScale (to slow down the time in the scene to a halt for the pausing) & pausing audio sources at the same time (to pause the music).

By: Daniel Carvalho Marques

PauseMenu.cs

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

public class PauseMenu : MonoBehaviour {

    public GameObject Canvas;
    public GameObject Camera;
    bool Paused = false;
    public AudioSource GameMusic;

    // Use this for initialization
    void Start () {
        Canvas.gameObject.SetActive (false);
    }

    // Update is called once per frame
    void Update () {
        if (Input.GetKeyDown (KeyCode.Escape)) {
            if (Paused == true) {
                //Debug.Log ("Unpause game!", gameObject);
                Time.timeScale = 1.0f;
                Canvas.gameObject.SetActive (false);
                Cursor.visible = false;
                Cursor.lockState = CursorLockMode.Locked;
                GameMusic.UnPause();
                Paused = false;

            } else {
                //Debug.Log ("Pause game!", gameObject);
                Time.timeScale = 0.0f;
                Canvas.gameObject.SetActive (true);
                //Debug.LogFormat ("Pause Menu is {0}", Canvas.gameObject.activeSelf);
                Cursor.visible = true;
                Cursor.lockState = CursorLockMode.None;
                GameMusic.Pause ();
                Paused = true;

            }
        }
    }

    public void Resume () {
        Time.timeScale = 1.0f;
        Canvas.gameObject.SetActive (false);
        Cursor.visible = false;
        Cursor.lockState = CursorLockMode.Locked;
        GameMusic.UnPause();
        Paused = false;
    }
```

By: Daniel Carvalho Marques

30/04/2019

Combining all the components into a small game I learnt how to integrate various components together so they could work and function cohesively such as the different canvas', components & scripts which all needed to be reassigned so that they could function cohesively.