Comp280 Worksheet 3

Tyler Pinney - 229736

January 2, 2021

1 Introduction

In this assignment I am going to use a project that i created to test how well my new computer handles intense operations. I have made this project's output variable so i can increase it over time, for this assignment it will be set very high and spawning 24 spheres every 0.1 of a second.

https://github.com/tylerpinney88/Comp280-Optimisation

2 Baseline readings

As we can see from the graph in figure 1, the frame rate drops the longer the program run for. This is due to the amount of spheres being spawned and kept within the scene. The frame rate drops well below 15fps which as a result of any code, we do not want.

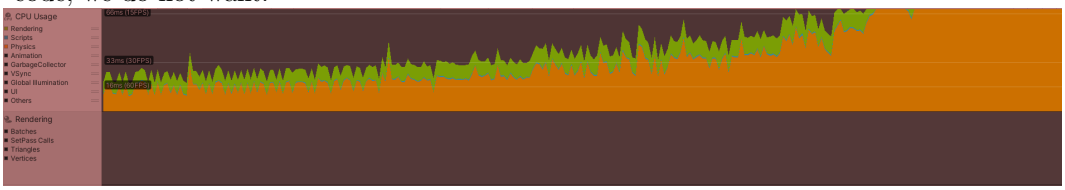

Figure 1

3 Optimisation

The first part of optimisation I will preform is to delete the spheres after a short duration, this will make it so that unity is not processing as many objects and therefore should speed up the frame rate. This is done with one simple line of code attached to the sphere that deletes it after 5 seconds as seen in figure 2.

```
void Start()
{
    Destroy(this.gameObject, 5f);
}
```

Figure 2

From this first optimisation it has drastically improved the fps of this simulation. It has gone from sitting below 15fps to staying at just above 100fps. This optimisation has improved the fps so much because of the fact that unity is not having to calculate the physics and rendering of more and more spheres over time. The results from this first iteration can be seen in figure 3.

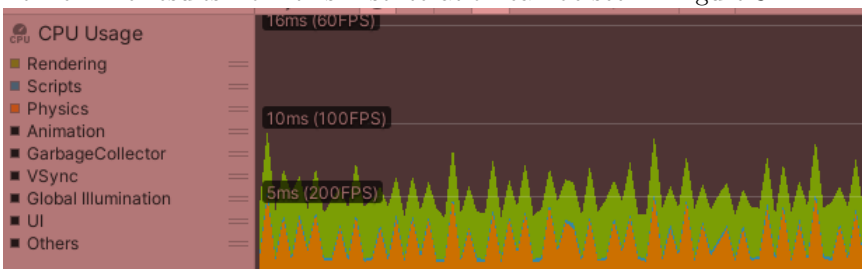

Figure 3

The next optimisation that I have done to this program is to change the code that spawns in the spheres. Originally I had a "InvokeRepeating" command in the start function that would call the "SpawnsSphere" function every 0.1 of a second. In the "SpawnsSphere" function I had 24 Instantiates that spawned the spheres. All of this can be seen in figure 4 and figure 5 below.

```
void Start()
{
    InvokeRepeating("SpawnsSphere", 0, 0.1f);
}
```

Figure 4

```
void SpawnsSphere()
{
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
    Instantiate(Sphere, transform.position, transform.rotation);
}
```

Figure 5

To optimise this code i have added a new public varible called "counter" which stats how many spheres will be spawned every 0.1 of a second. Having 24 instantiate commands is not the most optimized why to create all the spheres so I changed it so that there is only one instantiate command in the "SpawnsSphere" function. Then in the "InvokeRepeating" I have made it so that the spawn rate is divided by the counter so that the same amount of spheres are spawned in the same time frame but without using the same instantiate 24 times over. These changed are visable in figure 6.

```
public class Spawner : MonoBehaviour
{
    public GameObject spawner;
    public GameObject Sphere;
    public int counter;
    // Start is called before the first frame update
    0 references
    void Start()
    {
        InvokeRepeating("SpawnsSphere", 0, (0.1f/counter));
    }

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

    }


    0 references
    void SpawnsSphere()
    {
        Instantiate(Sphere, transform.position, transform.rotation);
    }
}
```

Figure 6

From the results in figure 7 we can see that with this newly optimised code the program is sitting around the 200fps mark which is another big leap from the previous results. The fact that I have taken out all of the instantiates from the "SpawnsSphere" script means that the script in essentially only running 2 lines of code repeatedly instead of 25 lines of code which results in the same amount of spheres spawning whilst sitting at a much higher frame rate.

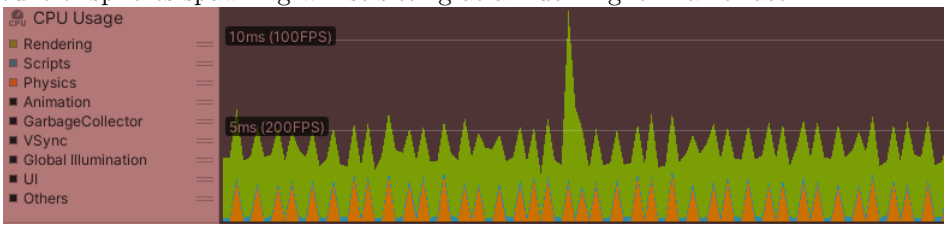

Figure 7