600086 Lab Book

# Week 8 – Lab 8 A simple Particle animation in CUDA

Date: 24th Mar 2022

## Exercise 1. Draw a box without front wall.

### Question1: adjust the code from lab 7 so that you have an open box drawn to the sscreen

### Solution:

I added in a set of 5 spheres that will make up the box they are very large making the curvature barely noticeable at our current scale.

Text

Description automatically generated

### Test data:

N/A

### Sample output:

A screenshot of a computer

Description automatically generated with low confidence

### Reflection:

none

### Metadata:

### Further information:

## Exercise 2. Free motion animation

### Question1: implement code to allow the particle to rotate about the centre of the box

### Solution:

Added a device wide global variable named tick



Changed the sphere to be drawn to the following



This moves the ball every time the image is rendered due to the tick variable being incremented

### Test data:

n/a

### Sample output:

Chart, funnel chart

Description automatically generated A screenshot of a computer

Description automatically generated with low confidence Chart, funnel chart

Description automatically generated

### Reflection:

This was fairly perfunctory and following the lab made sense

### Metadata:

## Exercise 3. Ball-box walls collision animation

### Questions:

1. Modify the previous code to give the sphere a velocity and bounce off of the walls of the box
2. Implement a code change that will make the ball change colour after a collision.
3. Implement code change to allow multiple balls to move at the same time;

### Solution:

1. I first defined some global variables for the sphere to track its movement

Text

Description automatically generated

I then modified the create world kernel to move the ball and check for collisions with the walls

Text

Description automatically generated

The collision is calculated by working out if the distance between the spheres position adjusted for the radius is intersecting with the wall position adjusted for the wall radius.

1. In order to achieve this:

I modified the Sphere class so that it contained a vec3 named colour to store its rgb values and modified the hit class so that hit\_record had a colour variable which would be set when the ray hits the sphere as shown below

Text

Description automatically generated

Modified the cast ray function so that when setting the colour, it applies the colour from the sphere that was hit to the shader.

A screenshot of a computer

Description automatically generated with medium confidence

The results of this can be seen in sample output section 2

1. To achieve multiple spheres at the same time I modified the static device variable to store the following values

Text

Description automatically generated

Then in the create world kernel if they are null then they are initialised and used to store the persistent sphere values the create world kernel now looks as follows

Text

Description automatically generated with low confidence

Text

Description automatically generated with medium confidence

Text

Description automatically generated

The above code will create 3 moving balls within the walls and the results can be seen in sample output ref 3.

### Test data:

N/A

### Sample output:

|  |  |
| --- | --- |
| Ref | output |
| 1 | A screenshot of a computer  Description automatically generated with medium confidenceChart, funnel chart  Description automatically generatedA screenshot of a computer  Description automatically generated with low confidence |
| 2 |  |
| 3 |  |

### Reflection:

Getting the shading to apply to the coloured balls lead to some difficulty as I had not considered applying the colour to the normal value to achieve the shading results.

### Metadata: