SCIT

School of Computing & Information Technology

CSCI336 – Interactive Computer Graphics
SIM Session 3 2017

Assignment 2

Task

Write an OpenGL program to create a simple planetary system using colour cubes to represent the "planets". The image in Figure 1 shows an example of this.
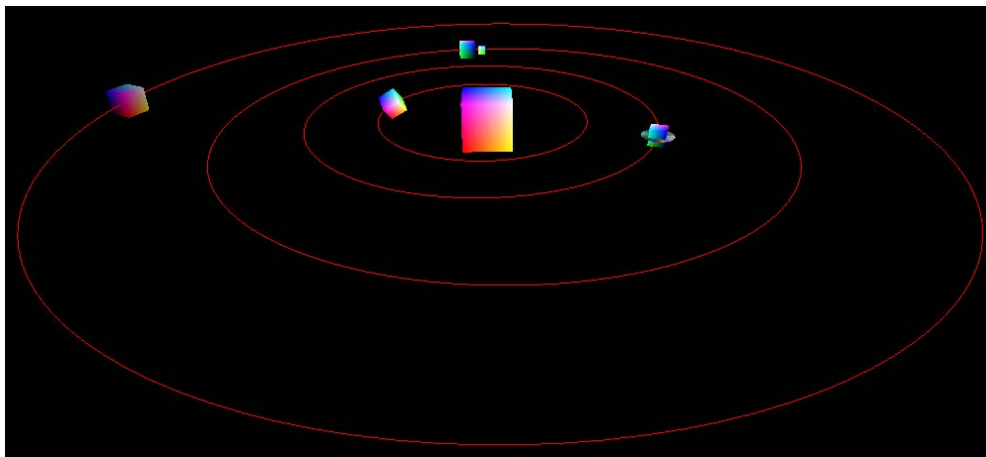

Figure 1: Simple planetary system.

- Construct the simple planetary system scene.

    o The scene should consist of a large "sun" in the centre and four other "planets" that orbit the sun. Note that your program should contain the vertex specification of a single colour cube. The sun and the planets are to use the same vertex specification, i.e. use transformations to alter the position, size and rotation.

    (1 mark)

    o Implement basic animation.

    Each planet is to orbit the sun. In addition, each planet also rotates about its own centre. The animation should be updated based on frame time, so that your program updates consistently regardless of frame rate. The planetary orbits and rotation only needs to be in the horizontal plane, i.e. rotation about the y-axis.

    Allow the user to toggle animation by pressing "p" on the keyboard.

    (1 mark)

o Randomise the size, orbit speed and rotation speed of the planets. When the user presses "r", re-randomise these values. Make sure the planets are smaller than the sun and the planet do not collide on their orbit paths.

(1 mark)

o Display the orbit paths, i.e. the red circles in Figure 1.

(1 mark)

- Create a perspective camera.

    o When the user presses "1", the camera should be set to view the scene from a camera position like the one shown in Figure 1.

    o When the user presses "2", the camera should view the scene from a top-down view, as depicted in Figure 2.

    o The user should also be able to control the camera using first person camera controls: WASD keys and mouse movement. Camera movement and rotation should be consistent regardless of frame rate.
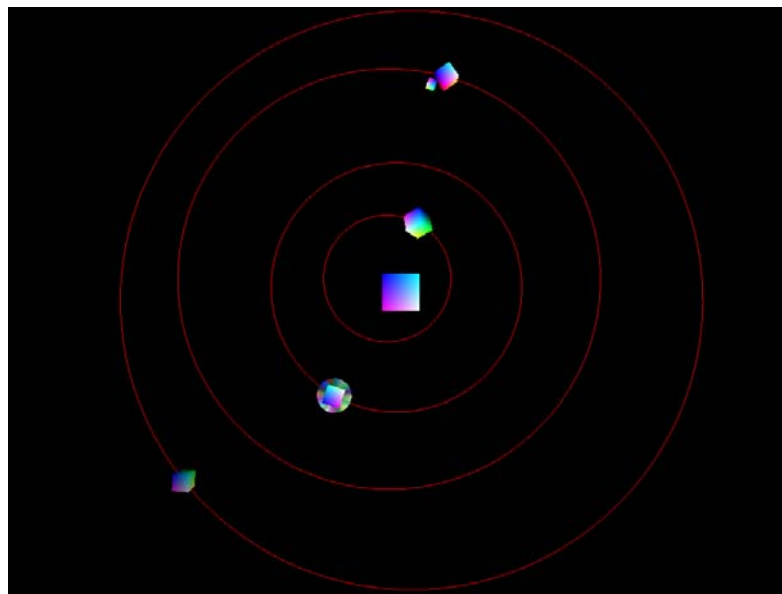
(1 mark)

Figure 2: Top-down view.

- Each planet should have the following unique characteristics:

    o First planet: is orientated like a diamond shape (use transformations. DO NOT redefine the colour cube by declaring different vertex coordinates)

    o Second planet: has a ring around the planet (the ring has random colours)

    o Third planet: has a "moon" orbiting the planet. The moon also rotates about its own centre

    o Fourth planet: gradually fades in and out between full colour and black colour[1]

(4 marks)

---

[1] Hint: use a uniform variable that periodically changes between 0.0 and 1.0

- Offset each of the planetary orbits by a random amount. This can be seen in Figure 2, where the orbit paths are not necessarily centred on the centre of the sun. These random orbits should also be re-randomised when the user presses "r".

(1 mark)

Instructions and Assessment

Submit an electronic copy of your work to your tutor at the start of the lab in which this assessment task is due. Do not try to fix your code during this lab, otherwise late penalties may apply. Your program must work on the computers in the lab or you must demonstrate it in the lab using your own laptop. The assignment must be your own work. If asked, you must be able to explain what you did and how you did it. Marks will be deducted if you cannot correctly explain your code.

The marking allocations shown above are merely a guide. Marks will be awarded based on the overall quality of your work. Marks may be deducted for other reasons, e.g., if your code is too messy or inefficient, is not well commented, if you cannot correctly explain your code, etc. For code that does not compile, does not work or for programs that crash, the most you can get is half the assessment marks or less.