SCIT

School of Computing & Information Technology

CSCI336 – Interactive Computer Graphics
Spring 2023

Assignment 2

Due on Friday, 6th Oct 2023 at 17:00

Task

Write an OpenGL program to create a simple 3D scene as shown in Figure 1 (a working program will be demonstrated in one of the lectures).

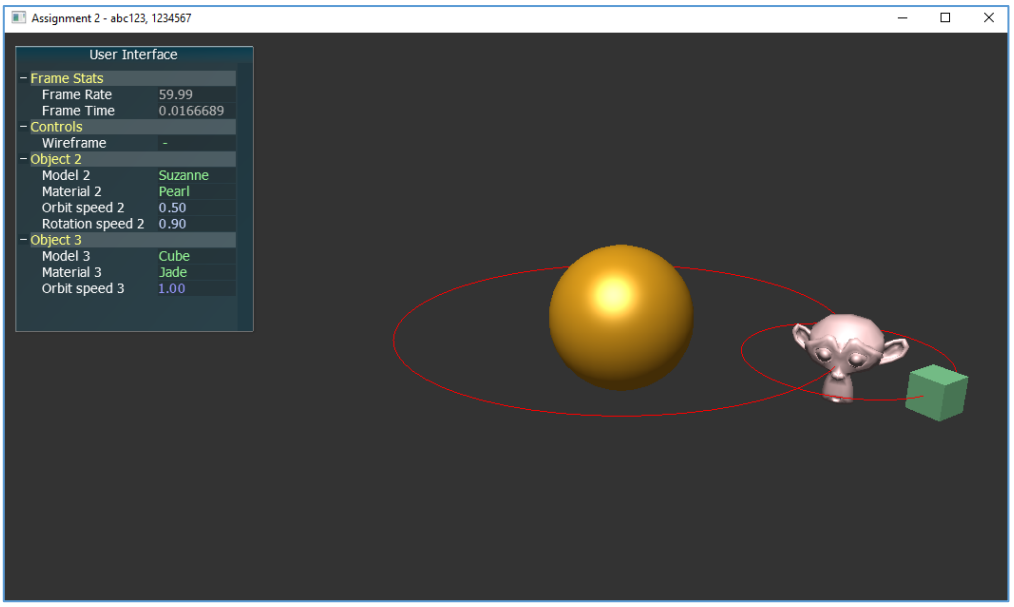


Figure 1: The 3D scene.

- 3D scene and animation (4 marks)

  Construct a 3D scene like the one shown in Figure 1. The scene should consist of the following objects:

  - A large sphere in the centre
  - A second object that rotates around its own centre and orbits the first object (i.e. the sphere in the centre)
  - A third object that orbits the second object
  - Two orbit paths

Figure 2 below illustrates the animation. The second object rotates around its own centre and orbits the object in the centre. The third object orbits the second object. The red circles show the orbit paths.

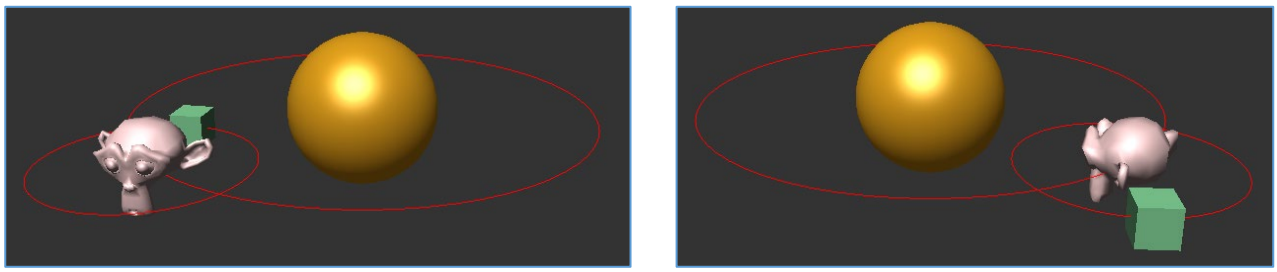
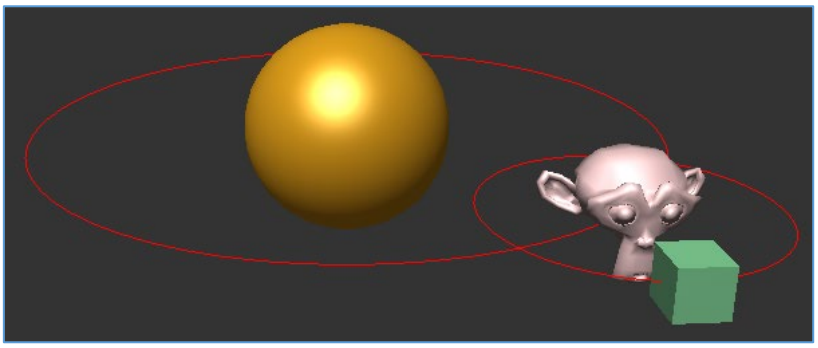


Figure 2: Images illustrating the animation.

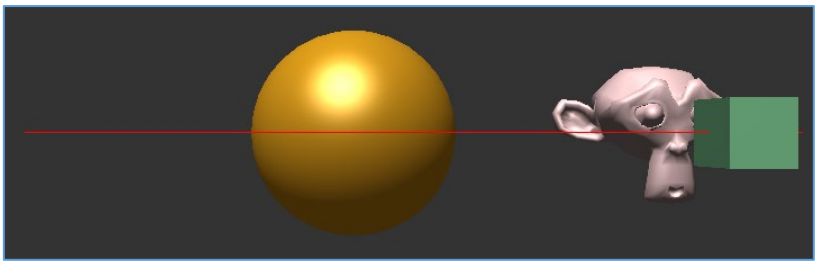
- Camera and viewport (2 marks)

  The scene should be rendered using perspective projection with the virtual camera placed at an appropriate position to view all objects in the scene.

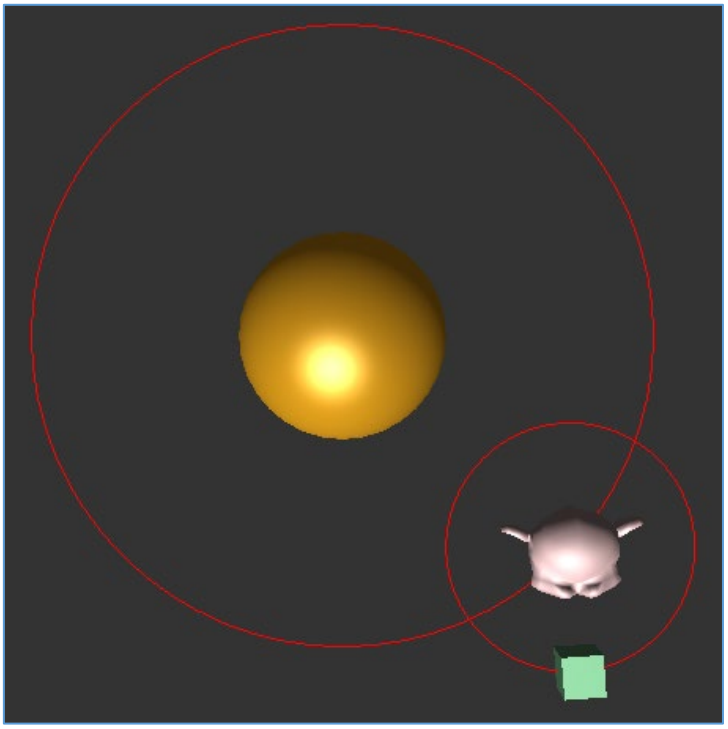
  - Viewport – The scene is to be rendered within a viewport that does not overlap with the user interface (see Figure 1).
  - Camera views – By pressing the '1', '2', or '3' keys, change the camera view to the following:
    - View from '1':



    - View from '2' (front view):



- View from ‘3’ (top-down view):



- Lighting, material and models (4 marks)
  - The scene is to be lit by a directional light source.
  - Each object is to be rendered with a different model and a different material.
  - The orbit paths should be rendered with a single colour.
    - Hint: use different shaders to render the orbit paths (you may use the shaders included with these specifications)
- User interface (2 marks)
  - Create a user interface to allow the user to change object 2 and 3’s:
    - Model.
    - Material.
    - Orbit speed.
    - Rotation speed (only for object 2).

  The images in Figure 3 demonstrate the user interface and some of the scene properties that can be controlled.

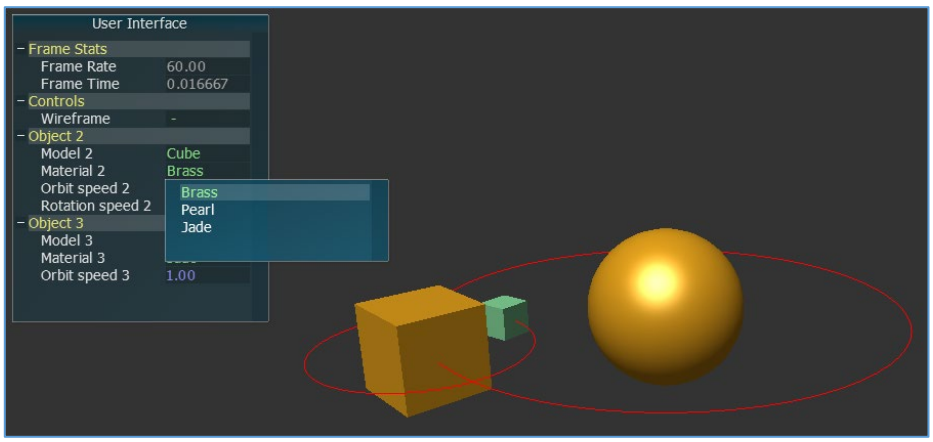


Figure 3a: Object 2 has been changed to a cube with brass material.

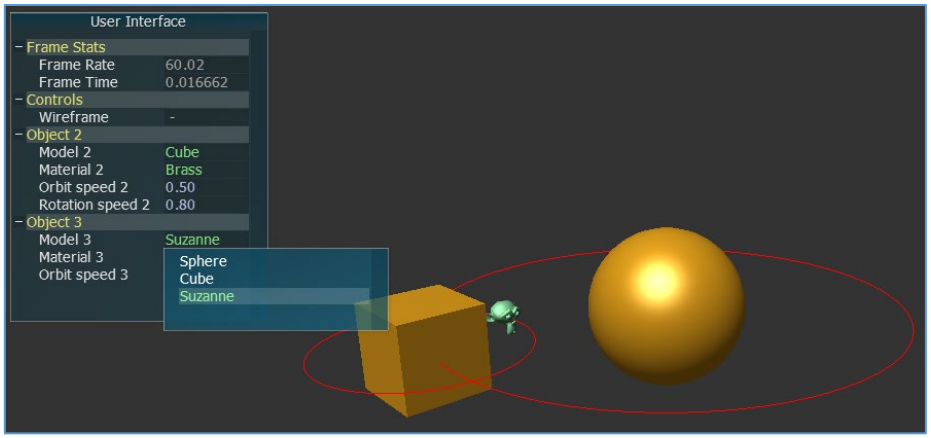


Figure 3b: Object 3 has been changed to the Suzanne model.

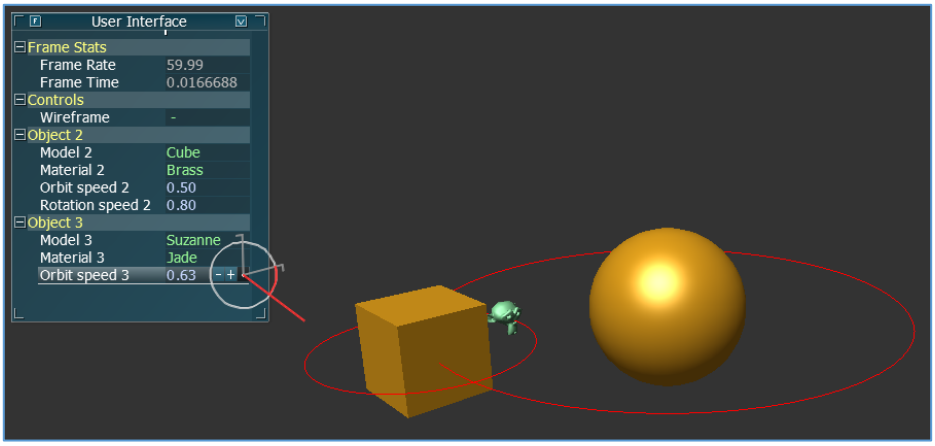


Figure 3c: Changing object 3's orbit speed.

Screenshots

In your submission, include screenshots demonstrating your working program and the features that were implemented. Save the screenshots using one of the common image formats, i.e. bmp/jpg/png.

Instructions and Assessment

Zip all source files (.cpp, .h, .vert and .frag) and **screenshots** (.bmp/.jpg/.png) into a single file. Submit this via Moodle by the due date and time (**do NOT zip your entire project file** as this can be very large and **do NOT use .rar format**). If not submitted on Moodle, the assignment will not be marked.

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.

NOTE: The mark 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, 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 marks (i.e. 6 marks or less). It is better to comment out sections of your code that do not work, and include a note for the marker.

