CS 606 Computer Graphics
Term II, 2018-19

Description of Assignment 3 & 4

Note: Assignments 3 and 4 are combined, with Assignment 3 being a demonstration of Phase 1. The final submission should contain the code of Phases 1 and 2. Phase 1 demo will be scheduled between Apr 8 and 10.

Outline:
This assignment builds on the concepts and structure of Assignment 2. To the extent possible, you are encouraged to reuse the code of Assignment 2 for the features that are similar..

- Phase 1: Design a scene with multiple models and with multiple light sources. Use texture maps to render the model objects, using different schemes for generating texture parameters/coords.

- Phase 2: Define a scene graph to manage these models. Animate the objects on the lines described below, using interactive controls to vary the animation. Manage lights and camera settings to create the effects described below.

Details of Phase 1:

Create a scene comprising of a floor/ground and at least 4 objects. The floor/ground is a plane. Each of the objects is a triangulated surface model, read in as a ply file or equivalent, such as those at https://people.sc.fsu.edu/~jburkardt/data/ply/ply.html. References to ply files below are to models available at this link.

1. The objects should be of different geometrical shapes, with at least one each of the following:
    I. An object with large flat areas such as a cube (cube.ply) or a table or an icosahedron (icosahedron.ply)
    II. An object with cylinder-like symmetry and shape, such as canstick.ply, ketchup.ply etc
    III. An object which is approximately spherical: sphere.ply, apple.ply, skull.ply, etc
    IV. An object with arbitrary geometry such as streetlamp.ply, head1.ply, shark.ply, galleon.ply etc

2. Identify at least 4 texture images, covering the following kinds of patterns:
    I. Wood, fur, bricks or similar pattern that will be replicated over a surface
    II. A well-defined geometric pattern such as a checkerboard
    III. A photograph with a person's face clearly visible (a selfie would work!)

IV. Map or image of the surface of the earth, mars or other planet, with moderate level of detail

(Note: you are free to use other models or textures, so long as they broadly correspond to the descriptions above)

3. Arrange the model objects on the floor/ ground, sufficiently far apart.

4. Set up 4 point sources of light, positioned at different locations, and each shining directly on one of the objects. Lights can be turned on/off individually. Use keys/mouse-clicks (or GUI controls) to turn lights on/off. For example, hitting numbers 1-4 will toggle the state (on or off) of that light. Note that each light points to one of the model objects.

5. Each of the objects should be renderable with each of the texture images and with each of the mapping schemes listed below. Use the following schemes to generate texture coords for vertices:

- A cylindrical map
- A spherical map
- An arbitrary pre-image mapping (such as that computed by OpenGL as a default mapping)

6. Use keys/mouse-clicks or GUI controls to change the textures and texture coordinates mappings. For instance, hitting the “t” key could cycle through the set of textures and apply a different one each time. Hitting the “m” key changes the mapping used (to generate texture coords). All objects (apart from the floor) are rendered with the same texture image and mapping at each step. (Other interaction mechanisms, such as GUI controls, are welcome).

Details of Phase 2:

Add animation to the scene and introduce controls as described below.

1. Design and implement a scene graph and add the models to this, The structure of the scene graph should help the implementation of program features described below. Rendering of the scene for each frame should be implemented by rendering this scene graph. (You might find it useful to set up the scene graph as part of Phase 1, to reduce subsequent effort)
2. Add animation to the objects in the scene as follows:Assume the 4 objects are named A, B, C, D. The objects move as follows:
    - Object A is fixed with respect to the floor
    - Object B moves in a circular (or similar path) around object A, and wobbles left and right as it traverses this path

    - Object C starts some distance away from B, and tries to move towards B, constantly adjusting its direction depending on the current position of B. The speed of C is roughly half that of B
    - Object D sits on top ot C, and jumps up and down while staying on top of C.

3. The four lights used in Phase 1 now track their target objects, as they move around.
4. The speed of the objects can be increased or decreased uniformly, while maintaining the relative ratios of their speeds. For example, hitting the up and down arrows could speed up or slow down all the objects by some factor.
5. The textures of the objects can be modified as in Phase 1
6. Clicking on an object (picking it) causes it to start spinning about a vertical axis through its “center”, while it continues its movement as above. Picking it again stops the spinning.

For all the above, it is important to set up an appropriate scene graph and manage all animations and rendering through traversal of the scene graph.