COSC422 Advanced Computer Graphics

Assignment 1

Due: 11pm, Friday, 19 August 2016

Maximum Marks: 25

I. Task Description:

The tessellation and geometry shader stages of the OpenGL-4 pipeline have found several applications in mesh processing and rendering. In this assignment, you will select ***any one*** of the tasks **(1)..(5)** listed below, and demonstrate the use of a tessellation shader or a geometry shader or both in altering the characteristics and rendering a three dimensional object. Each section contains a list of subtasks that should be completed.

(1) Bezier surfaces: A set of mesh models defined using 4x4 Bezier patches is provided in the file "BezPatches.zip". The models include the popular Utah Teapot, and two other not so popular objects, the teaspoon and the teacup. You may select any other model for which patch based definitions are available.

1.(a) Use the tessellation stage to generate bi-cubic Bezier surfaces for the model.

1.(b) Apply lighting calculations to render the generated model under a light source.

1.(c) Use a view-dependent tessellation to change the tessellation level factors based on the distance of the patch/object from the camera. As the patch/object moves closer to the camera, the tessellation levels should increase. Your program should be able to demonstrate this using a wireframe display mode for the model.

1.(d) Apply a texture or a displacement map[1] on the object.

(2) Terrain rendering: A terrain model can be generated using a height map to define the *y*-values at the vertices of a two-dimensional polygonal grid. A sample height map is provided (HeightMap.tga, HeighMap.raw, HeightMap.jpg). You may use any other height map or a digital elevation map (DEM) for this assignment.

2.(a) Use the tessellation stage to implement a mesh subdivision method where the tessellation level for a patch depends on its distance from the camera as well as some other geometrical characteristic such as curvature, slope etc. Your program should be able to demonstrate this by changing the camera position in a wireframe display mode.

2.(b) Apply lighting calculations to render the terrain under a light source.

2.(c) Use a set of textures for terrain features (eg. water, rock, grass, snow), and select a texture for mapping based on the height value at a vertex.

1. Displacement map: Wikipedia http://en.wikipedia.org/wiki/Displacement_mapping.

(3) Surface approximation: The application of tessellation shaders presented in Ex-03 may be extended to generate a mesh subdivision for approximating shapes that are more complex than a sphere (parametric surfaces, implicit surfaces etc.).

3.(a). Use the tessellation stage for subdividing a coarse mesh model, and update the vertex positions to get a closer approximation of the surface.

3(b). Use a view-dependent tessellation to change the tessellation level factors based on the distance of the patch/surface from the camera. As the patch/surface moves closer to the camera, the tessellation levels should increase. Your program should be able to demonstrate this using a wireframe display mode for the model.

3(c). Apply lighting calculations to render the generated model under a light source.

(4) Mesh explosion: A geometry shader has access to complete primitives that are generated by the application. Primitives can also be generated inside a geometry shader. A primitive's characteristics (shape, position, colour) can be modified inside the geometric shader. Use models with at least 50 triangles for this task.

4.(a) Use a geometry shader to generate the animation of a mesh explosion, where the primitives move radially outwards from the centre of the object.

4.(b) Create additional effects such as trajectories for the primitives, or changes in colour and size of primitives as they move away from each other.

4. (c) Add lighting calculations to the scene.

(5) Particle System: Use point sprites (GL_POINT_SPRITES) or a geometry shader to generate a sprite based animation of a particle system representing a natural phenomenon (fire, smoke, water etc).

5.(a) Some characteristics of the sprite (eg. colour, texture) must change depending on the particle system parameters (eg. poistion, life)

5.(b) The particle system must be based on some physics based models.

II. Report (Max. 4 pages; Max. marks: 5):

Please prepare a report describing your work, and include the following sections:

- A brief outline of the implemented methods. You may also describe problems/challenges faced and how you attempted to solve them.

- All relevant equations describing how tessellation levels and any other animation parameters are updated, how vertex positions and other primitive characteristics are modified.

- A few screenshots showing the outputs of your program.

- The complete list of keyboard/mouse functions defined for user interaction.

- List of references, acknowledgements.

III. Program Development:

You may use math library functions, mesh models, and images that are available on the Internet or obtained from other sources such as books. Please acknowledge the source in your report. You may also use programs and other supplementary materials provided in this course. If your implementation is based on a method described in a paper, book etc., please give full details of the source in the list of references.

Demo programs found on the Internet and other OpenGL resources should not be submitted as part of the assignment. Please do not use OpenGL Extensions (ARB, EXT etc) or third-party mesh processing libraries in your program.

IV. Assignment Submission

Submit your files using the assignment link on Learn (learn.canterbury.ac.nz) before 11pm on 19 August 2016. Your submission must contain:

1. The source code(s) and all supplementary files (textures, mesh files) needed to run your program. Please do not include freeglut, opengl, glew or glm library files.

2. Your report in Word or PDF format.

Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications.
2. You may submit up to one week late for a 15% penalty.

3. This is not a group project. Your assignment must represent your own individual work. In particular, students are not permitted to share program source code in any way. However, you may discuss ideas, implementation issues etc using the class forum on Learn.

4. Standard departmental regulations regarding dishonest practices and late submissions apply.