COSC422 Advanced Computer Graphics

Assignment 1

Due: 11:55pm, Friday, 23 August 2019

Maximum Marks: 25

The tessellation and geometry shader stages of the OpenGL-4 programmable pipeline have found several applications in modelling and rendering of three dimensional objects. In this assignment, you will use tessellation and geometry shaders to generate models of Bezier surfaces and terrains. The assignment includes two tasks as outlined below, and both tasks should be completed.

I. Bezier Surface Modelling [Max. 9 Marks]:

The patch definitions of two Bezier surface models, teapot (PatchVerts_Teapot.txt) and Gumbo (PatchVerts_Gumbo.txt), are provided. Both models use 4x4 Bezier patches (16 vertices per patch). You may use any one of these two models for your assignment. The first line of the patch definitions file contains the total number of vertices (=number of patches * 16), followed by the x,y,z coordinates of all vertices (one vertex per line). The vertex buffer object defined for vertex coordinates must store these vertices in the given sequence.

1.(a) Use the tessellation stage to generate the display of a bi-cubic Bezier surface model with 4x4 Bezier patches as inputs. The scene should contain a floor plane constructed using a set of parallel lines along x and z directions.

1.(b) Apply lighting calculations to render the generated model under a light source. You will need a geometry shader for this.

1.(c) Use view-dependent tessellation to change the tessellation level factors based on the distance of the object from the camera. The tessellation levels should decrease as the object is moved away from the camera. Your program should be able to demonstrate this using a wireframe display mode for the model. Use the 'w' key to toggle between wireframe and solid fill modes, the the 'up' and 'down' arrow keys to move the camera towards and away from the object.

1.(d) Generate an animation sequence for displaying an explosion of the Bezier surface model (please refer to lecture notes). Use the space key to initiate the animation sequence.

II. Terrain Modelling and Rendering [Max. 12 Marks]:

A programming exercise (TerrainProgramming.pdf) containing a sample height map (HeightMap1.tga), a description of the method for generating the wireframe model of a terrain, and sample code are provided. You may extend this program to incorporate the additional requirements of this assignment listed below. Your implementation of the terrain generation algorithm should include the following functions/features:

- Dynamic level of detail: The tessellation level for a patch should depend on its distance from the camera. Your program should be able to demonstrate this by moving the camera over the terrain in wireframe display mode.

- The geometric shader must include lighting calculations (ambient and diffuse terms) to render the terrain under a light source.

- The implementation must use at least three textures for terrain features (eg. water, rock, grass, snow etc.), and use height based texturing with appropriate blending of textures to get a smooth transition from one texture to another. Water regions must have flat surfaces.

- The program should be able to display at least two terrain models (using two different height maps)

- The program should include the following keyboard and mouse functions for user interaction:
  'w': Toggle between wireframe and solid-fill (textured) display
  '1', '2' : Display terrain models 1, 2.
  Arrow keys: Camera motion over the terrain
  You may include additional keyboard/mouse functions as necessary.

- Extra features [Max marks 3]: Some of the additional features that you could implement to gain extra marks up to a maximum of 3 marks are listed below:

  - Cracking is a common problem in terrain rendering where the tessellation levels of two adjacent patches do not match along a common edge. A simple solution could be devised to solve the problem of cracking. Please provide screenshots in the report showing the working of your solution. [1 mark]

  - Adjustable water levels [1 mark]

  - Adjustable snow level [1 mark]

  - Fog [1 mark]

  - Water features (ripples, colour variation with depth etc.) [1-2 marks]

  - Billboarded trees or other objects [1-2 marks]

  The list given above should not be taken as the complete set of features that can be implemented.

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

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

- A brief outline of your programs including a description of the extra features implemented. 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.

- A few screenshots showing the outputs of your programs.

- The complete list of keyboard/mouse functions defined for user interaction. Please also include references to the sources of textures, models etc. used in your programs.

III. Program Development:

You may use math library functions (eg. GLM), 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 any part of 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. Please use only C/C++ as the programming language for application development. 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 23 August 2019. Your submission must contain:

1. The source code(s) and all supplementary files (textures, data 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.