COSC422 Advanced Computer Graphics

Assignment 1

Due: 11:55pm, Thursday, 11 August 2022

Maximum Marks: 20

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 these shader stages to generate a terrain model.

I. Terrain Modelling and Rendering:

A programming exercise ("4. Terrain Rendering") 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:

I(a): Basic Terrain Model (Max. 12 Marks)

- 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.

- Lighting (ambient and diffuse terms) calculations must be performed 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 two terrain models (using two different height maps)

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

I(b): Extra features (Max. 5 Marks)

Some of the additional features that you could implement to gain extra marks up to a maximum of 5 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]. Please use keyboard functions 'q', 'a' to increase and decrease water level.
- Adjustable snow level [1 mark]. Please use keyboard functions 'w', 's' to increase and decrease snow level.
- Fog [1 mark]. Please use keyboard function 'f' to toggle between fog-on and fog-off.
- Water features (ripples, colour variation with depth etc.) [0.5 - 2 marks]
- Smooth shading [1 mark]
- Procedural height map [1-2 marks]

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

II. Report (3-4 pages; Max. marks: 3):

Please prepare a brief report containing the following:

- A brief outline of the terrain model and a screen shot of the rendered (texture mapped) terrain. You may also describe problems/challenges faced and how you attempted to solve them.
- A brief outline including any relevant equations, of each extra feature implemented in the program. Please include a screen shot of a part of the terrain clearly showing the implemented feature.
- 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 program.

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

Please submit your files using the assignment link on Learn (learn.canterbury.ac.nz) before the due date. Your submission must contain:

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