University of London
CO3355 Advanced graphics and animation
Coursework assignment 1 2018–19

Important notes
For every step of the coursework, provide screenshots from multiple viewpoints. Describe your modelling approach and expose the problems you faced and the design decisions you made. Also, include an assessment of how well the techniques you used apply to what you are trying to do, identifying advantages and disadvantages.

What to submit

- a single .**pdf** file. This should include listings of the software you have developed, with your own contributions highlighted and an attribution for the remaining code (such as code taken from the subject guide or external sources).
- a single .**zip** file which contains all source code files that you have developed for this coursework, with instructions (as comments in the source files, or as a separate readme file) on how to run them. The files should be categorised in folders that correspond to coursework parts.

Your **.pdf** and **.zip** file must be named using the below file-naming conventions, and include your full name, SRN, course code and assignment number:

YourName_SRN_COxxxxcw#.pdf (*e.g*. MarkZuckerberg_920000000_CO3355cw1.pdf)

- **YourName** is your full name as it appears on your student record (check your student portal)
- **SRN** is your Student Reference Number, for example 920000000
- **COXXXX** is the course number, for example CO3355, and
- **cw#** is either cw1 (coursework 1) or cw2 (coursework 2)

Part A

Raster images in the form of 2D arrays are used to model terrain information in Computer Graphics. Each pixel value, represented by an element in the array, is interpreted as the amount of displacement from the floor of a surface. These raster images are also known as heightmaps [1].

1. The Processing code included in folder `partA` contains terrain information stored in an array. Add code to draw it as a Quad mesh. Make sure you incorporate peasycam [2] and gain some familiarity with its API, in order to enable camera navigation.

   [15 marks]

2. Now generate your own terrain (represented as an array) using random values. Make it include an adjustable number of "peaks", *i.e*. points with considerably greater height than the rest. Allow the user to increase or decrease their number by using corresponding keys.

   [15 marks]

3. Choose a greyscale image, load it in Processing and use its pixel values to generate a heightmap and draw it as a Quad mesh. Comment on the results.

   [10 marks]

Part B

The folder `partB` contains a Processing method `get_torus` that generates a torus-shaped object. Read through the code to understand how the method is called. Make sure you are familiar with the basic concepts of GLSL programming, using your subject guide and other resources. You can find a comprehensive walk-through tutorial for shaders in Processing in [3], while the Book of Shaders [4] is a very good resource for understanding and experimenting with GLSL fragment shaders.

1. Using [5] for inspiration, implement GLSL shaders and the appropriate Processing code in order to render tile patterns on a torus. Experiment with at least two patterns inspired by the referenced site and comment on the results.

   [10 marks]

2. Extend your code so that the implemented patterns change dynamically over time. To that end, add a "time" variable to your fragment shader(s) and use it to animate the pattern. After that, add a variable that allows you to modify the pattern, based on mouse position.

   [10 marks]

3. Improvise and experiment with more sophisticated patterns and see what effects you can produce. You may use different types of random noise as well as trigonometric or other types of mathematical functions. Make sure that your patterns vary over time and that their effects are adjustable by the user.

   [40 marks]

[Total: 100 marks]

References

[1] "Heightmap" in Wikipedia, The Free Encyclopedia, retrieved November 2018 from https://en.wikipedia.org/wiki/Heightmap

[2] Jonathan Feinberg, "Peasycam", retrieved November 2018 from http://mrfeinberg.com/peasycam/

[3] Andres Colubri, "Shaders", retrieved November 2018 from https://processing.org/tutorials/pshader/

[4] Patricio Gonzalez Vivo and Jen Lowe, "The book of shaders", retrieved November 2018 from https://thebookofshaders.com/

[5] Patricio Gonzalez Vivo and Jen Lowe, "Patterns", retrieved November 2018 from https://thebookofshaders.com/09/

[END OF COURSEWORK ASSIGNMENT 1]