The University ofMelbourne
School of Computing and Information Systems
COMP30019 Graphics and Interaction

Project 1, 2017

Set: 7th August

Electronic submission: 4pm, 4th September

Marks: This project counts towards 10% of the marks for this subject

This project should be completed in groups of two or three. If you have any strong objections regarding doing group work, please contact the responsible tutor or coordinator and give your reasoning behind wanting to complete the project individually.

Aim

The purpose of this project is to get you acquainted with basic Unity development.

Task

In this project, you will use fractals to automatically generate a 3D landscape, and enable a user to navigate around the landscape in a 'flight simulator' style.

Diamond-square algorithm

You must use the ***Diamond-Square algorithm***, which is a de-facto standard in fractal landscape generation. Your fractal will generate a ***heightmap*** which can then be used to construct the 3D landscape geometry.

There is information on the diamond-square algorithm on Lecture 5 and on LMS, and you will receive support during the tutorials.

Specifications and marking criteria

A project that satisfies all of the criteria listed below will receive 10 marks.

- Modelling of fractal landscape (3 marks):
  - You must automatically generate a randomly seeded fractal landscape at each invocation of the program, via a correct implementation of the diamond-square algorithm.
  - You must use Unity's architecture appropriately to generate and render the landscape.
  - There must be no notable problems or artifacts with the polygonal representation.

- Camera motion (3 marks):
  - Your camera controls should be implemented in a ‘flight simulator’ style, with the following specifications:
    - Moving the mouse should control the relative pitch and yaw of the camera
    - The ’w’ and ’s’ keys should cause the camera to move forwards and backwards respectively, relative the camera’s current orientation
    - The ’a’ and ’d’ keys should cause the camera to move left and right respectively, relative to the camera’s current orientation
    - The ’q’ and ’e’ keys should control the roll of the camera
  - You must allow the user to move anywhere in the world (including up into the sky), and prohibit the user from moving “underground” or outside the bounds of the landscape.
  - The camera must not become ‘stuck’ upon nearing or impacting the terrain, i.e. reversing and continuing to move must always be possible.
  - You must utilise perspective projection, and choose a suitable default perspective, so that the landscape is clearly visible from the start.
- Surface properties (4 marks):
  - The colour of the terrain must correspond in a sensible way with the height of the terrain at any particular point (for example rocky outcrops or snow on top of mountains and grass or soil in valleys).
  - Suitable lighting must be present based on the Phong illumination model (diffuse, specular and ambient components). You should use a custom Cg/HLSL shader for this.
  - The direction of the lighting must change with time, to simulate the effect of a sun rising and setting.
  - The sun itself must also be drawn, in order to help verify the correctness of your lighting implementation. You may use any simple geometric shape such as a pyramid, cube or sphere to represent the sun.
  - A constant and reasonable frame refresh rate must be maintained during program execution (i.e., 30 frames per second or more)

Electronic submission

Your submission *must* open and run in the environment installed in the tutorial rooms. Because of this, it is probably safest not to upgrade or modify the Unity installation provided, just in case there is some sort of resultant incompatibility.

You will need to submit your project via the LMS by the due date. Your submission must include:

- a readme.txt file that (briefly) describes your implementation. Be sure to include a brief description of how you generate the terrain using Unity. Several paragraphs of text are sufficient and concise descriptions are preferred over long, verbose descriptions;
- a group.txt that lists each of the user names of the project participants, so we can associate you with your project. Note: **only one member of your group needs to make a submission to the LMS**.

Important: If your code contains code from other sources, in particular from other web sites, you have to clearly indicate this in *readme.txt*, which classes or methods are your own and which are from a different source. Remember that copying code from the Internet or from your colleagues will be considered cheating. We will be checking for similarity between submissions and with code available over the Internet.

Peer Assessment of Contribution

If you complete the project in a group, you will be required to provide an assessment of your contribution and your colleagues' contribution in LMS. This is mandatory and aimed at minimizing potential conflicts between members of a group, and providing a fairer grading scheme.

Delays

Make sure you deliver your work on time using LMS. Overdue delivery will result in a reduction of 10% of the marks for each day of delay.