Research Title: Comparison of CPU and GPGPU performance as applied to procedurally generating complex cave systems

Student: Tony Oakden (U4750194 )

Date: 21/08/2015

Abstract
The student would gain a good understanding of GPGPU software development. With a focus on looking at performance and scale relating to computer graphics and procedural generation. Also it is expected that the student would gain general skills relating to: writing a report, and giving a seminar.

Research Plan
The investigation will be carried out in three phases

Phase 1 - Background:
Starts: week1, ongoing throughout project

• Literature survey of the use of GPGPU for procedural generation of 3D environments. This will include researching implementation details for OpenCL and Tessellation shaders. Initial research will look at techniques for generating geometry procedurally. Later research will look for specific implementation details

Phase 2 – Implementation:

• Replacement of the existing CPU geometry processing stage with GPGPU kernel(s) and processing operation. The fundamental nature of the algorithm will not change but the computationally intensive stages will be carried out on the GPU instead of the CPU data and read back to main memory once complete.

  3. Rewrite existing C++ code to be closer in structure to the OpenGL code
  4. Rewrite code mesh smoothing routine to run on the GPU
  5. Collect initial results and debug code
  6. Write first part of report

• The existing processing will be changed so that some of the processing of geometry can be performed as part of the graphics pipeline. Research will need to be carried out into how best to accomplish this but it will probably involve the tessellation of patch data on the GPU instead of pre-calculating all the primitives for the geometry using the CPU.

  7. Create basic tessellation shaders and modify engine to load, link and use them
  8. Modify code to pass required data in tessellation shaders and produce displacement shader
  9. Test code and debug
  10. Performance tests and result collection
  11. Write second part of report

• Combination of the above two processes so that the GPGPU is used to pre-calculate patch data and then the tessellation shader is used to compute the final geometry during the render stage.

  12. Research interoperability between OpenCL and OpenGL and decide on a solution which allows for part OpenCL and part OpenCL
  13. Implement, and test
  14. Gather results
  15. Write third part of report
Phase 3 – Analysis and evaluation:

- Timing comparisons will be made between CPU and GPGPU based approaches. This performance would be discussed in comparison to current procedural generation approaches as reviewed in the literature survey. Also the balance between pre-calculation and the calculation during rendering will be evaluated.
  - 16 – 18 Write remaining portion of report
  - Produce final presentation and present
  - Package up artefact and submit