Lab 9 Vertex Arrays and Procedural Generation

# Vertex arrays

1. Using the code provided on Blackboard (shape.cpp/h) and in the lecture, add code into the function render1() inside shape.cpp. This should render a quad using the vertex array data provided and using dereferencing method one discussed in the lecture; glArrayElement().
2. Complete the code for function render2(). This should use method two discussed in the lecture; glDrawArrays().
3. Complete the code for function render3(). This should use method three discussed in the lecture and requires the use of the index array; glDrawElements().
4. Extend the vertex array data so it draws a cube instead of a quad.

# Procedural Generation

1. Create an application that procedurally generates and renders a disc. Similar to that discussed in the lecture.
   1. Add normals and texture coordinates to the generation and render of the disc.
2. Add to your application a procedurally generated sphere, including normals and texture coordinates.
3. Generate a cylinder, including normals and texture coordinates
4. Remember your generation should take in values so we can control how many segments make up a disc, or sphere, or whatever the shape is. This will allow us to control how detailed the object is. If it does not already, update your generation code to generate shapes based on a given resolution.

# For fun

1. Research and procedurally generate a torus (donut). Look here for the maths: <http://en.wikipedia.org/wiki/Torus>