CS241 Project 2

Synopsis

Implement the Sutherland-Hodgeman Polygon Clipping algorithm.

Instructions

This project builds upon the framework of Project 1:

- I suggest you create a copy of Project 1 before making modifictions for project 2.
- Replace the framework's **scene.cpp** with the one provided for this project. This new file provides a scene appropriate for testing the clipper.
- Find the code in **render.cpp** which extracts the polygons from the scene, transforms the vertices and draws them. Modify that code to clip the polygon after the transformation step and before the homogeneous division and drawing steps. **Note:** The clipping must take place in homogeneous coordinates – **after** all transformations, and **before** the homogeneous division.
- Since it is impossible to tell the difference between a line clipped by your code and a line clipped by the edge of your window, make the following adjustment. Instead of the usual clipping boundaries derived from

  $$-1 \le \frac{x}{w} \le 1, \quad -1 \le \frac{y}{w} \le 1, \quad 0 \le \frac{z}{w} \le 1$$

  use instead

  $$-0.9 \le \frac{x}{w} \le 0.9, \ -0.9 \le \frac{y}{w} \le 0.9, \quad 0 \le \frac{z}{w} \le 1 \ .$$

  That way, it will be clear when your code clips the polygons. I leave it as an exercise for you to turn those six inequalities into the six plane equations:

  $$0.9\,w + x \ , \ \ldots$$

You have complete freedom as to how you design your code. (Be sure to include details in your project report.) Keep in mind that the sequence of operations performed on a polygon is now enhanced with an extra step:

- Vertices are extracted from the **Scene** data structure.
- Vertices are transformed by the viewing and perspective transformations.
- **The new step:** Transformed vertices are run through the clipper, resulting in a new (longer, shorter or possibly empty) sequence of transformed vertices.
- The resulting vertices are homogeneously divided and passes to OpenGL for drawing.

Project Report

As always, submit a **succinct** project report with your project, describing anything I need in order to understand your implementation. Place your project report in a text file named **report.txt** (or **report.doc** or **report.odt** if you (unnecessarily) insist on producing a formated document).

What and how to submit

Create a zip file containing only your report, and your source code ***.cpp, *.h** and ***.vcproj** files ***and nothing else.*** Submit the **zip** file via the Moodle class web page.