CPSC 414
Assignment 2

Due Tuesday March 4, 2003, 11:00am

Answer the questions in the spaces provided on the question sheets. If you run out of space for an answer, use separate pages and staple them to your assignment.

Name: 

Student Number: 

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>8</td>
</tr>
<tr>
<td>Question 2</td>
<td>8</td>
</tr>
<tr>
<td>Question 3</td>
<td>6</td>
</tr>
<tr>
<td>Question 4</td>
<td>30</td>
</tr>
<tr>
<td>TOTAL</td>
<td>52</td>
</tr>
</tbody>
</table>
1. Scan Conversion of Arbitrary Polygons

(a) (4 points) Determine an appropriate edge-table to use for scan-converting the above non-simple polygon. Use the space to the right of the figure above.

(b) (2 points) What does the sorted active edge-list look like for scan-line 6 (i.e., \( y = 6 \))?
   Assign each edge in your active edge list a label, \( e_n \).

(c) (2 points) List what happens in order to update the active edge list in preparation for the drawing of scan-line 7. Give the sorted active edge list for scan-line 7.
2. Interpolation

(a) (2 points) Interpolate the value of $z$ for the point $P$ using bilinear interpolation.

(b) (2 points) Interpolate the value of $z$ for the point $P$ by writing a plane equation in $x, y,$ and $z$, and solving this for the given point $P$.

(c) (4 points) Determine the barycentric coordinates of point $P$. Interpolate the values of $z$ and the $r, g, b$ colour.
3. (6 points) Clipping

Clip the triangle that has VCS coordinates of $P_1(-3, -2, -8), P_2(0, -8, -9), P_3(0, -3, -13)$ to the perspective viewing frustum given by $\text{bot}=-1, \text{top}=1, \text{left}=-1, \text{right}=1, \text{near}=2, \text{far}=10$. Show your intermediate results, and use the following clipping order: top, bot, left, right, near, far.
4. Coding Question (30 marks total)

In this coding questions, you will be implementing a surface of revolution using a basic triangle mesh data structure, computing surface normals, and further learning how to use glut to build an interactive 3D application.

A surface of revolution consists of sweeping a two-dimensional outline about an axis in order to obtain a circularly-symmetric 3D surface. This can be a simple way to build the geometry of objects such as a vase, wine glass, or hat, for example.

(a) (0 points) Download and compile the template code. See the CPSC 414 web page for this. This code represents a starting point. You may change it in any way, as you see fit. Add new files to your code as appropriate.

(b) (6 points) First implement the code that creates a curve by adding a line segment for each mouse click. If you like, simply use a fixed-size array to store the curve points. Update the display of the curve after each mouse click. Now update your code to be able to sketch the outline curve continuously with one mouse down ‘stroke’.

(c) (10 points) Implement the mesh generating code and call it when the ‘m’ key is pressed. Use 32 slices to build your mesh. Store the mesh using a vertex list and a triangle list. Each triangle should then consist of three integer indexes that refer to the vertex list. Display the mesh in wire-frame, i.e., using something like GL_LINE_LOOP primitives. You will need to compute the mesh vertex locations yourself; glRotate() is not the way to approach this.

(d) (4 points) Implement a means to use the mouse to interactively rotate the mesh.

(e) (4 points) Use a cross-product to compute and store surface normals for each triangle. Use the ‘n’ key to toggle the display of the surface normals. The surface normals should be pointing outwards, i.e., away from the axis of revolution.

(f) (2 points) Use the surface normals to obtain a lit, flat-shaded model. This consists of calling glEnable(GL_LIGHTING), and then calling glNormal3f(nx,ny,nz) immediately before the first glVertex() call for every triangle. Use the ‘s’ key to toggle the display between solid shading and wireframe rendering.

(g) (4 points) Implement two other functions of interest. Ideas include ways of changing the scale of the object, using a colour ramp to smoothly change the colour over the length of the object, changing the lighting used, and allowing a combined display of the wireframe mesh + solid shading.

(h) (0 points) For up to 6 bonus marks, consider implementing any or all of the following. Your assignment mark will not be capped at 100%. (a) an automatic way of filling virtual vases or wine-glasses with some representation of water (b) a way of selecting a point on the surface of the surface of revolution by computing the 3D line equation for the selected 2D point and intersecting this with all triangles in the scene. (c) produce a smoothly-shaded representation by computing separate normals for each vertex (d) allow for interactive control over the number of slices used to produce the surface of revolution.
Hand-in Instructions

Hand in a printed copy of your README file.

Create a folder called 'assn2' under your cs414 directory and put all the source files, your makefile, and your README file there. Do not use further sub-directories.

In your README file, include: your name and login ID, a description of how to run your program, what features you have implemented, as well as any kind of information you would like to give us for getting credit for a partial implementation.

The assignment should be handed in with the exact command:

```
handin cs414 assn2
```