CS 543 – FALL 2013 – Midterm Exam

Name:

Instructions: Read questions carefully before answering. Do not hesitate to ask for clarifications. Show all work. Partial credit is given, so do not leave anything blank! Ask for extra paper if you need it! Good luck!

Bonus Question (1 point) What is the name of WPI’s provost?

Question 1: (15 points) Short Questions

Briefly answer the following questions

a. What performance issue does immediate mode graphics create? How does retained mode graphics solve it?

b. What is a vertex attribute?

c. Why is it preferable to pre-multiply matrices together before applying the resulting matrix to each vertex of a mesh?

d. What is a convex hull?

e. What is self-similarity in fractals?
**Question 2: (9 points) OpenGL/GLUT**

a. What command (with the right argument(s)) will you issue to draw a closed loop of polylines?

b. When setting up shaders, what data type does the command `glShaderSource` expect as an argument?

c. In the vertex shader, variables declared as "in" are received from where?

**Question 3: (6 points) View Transformation**

You are looking at the origin of your world from the position (x, y, z) with up vector (ux, uy, uz) and have computed your viewing transformation matrix M. You now want to look at the other sides of what you could originally see by simply moving to a new viewing location (e.g., (−x, −y, −z)) but with the same up vector (ux, uy, uz). Can you reuse some elements of the matrix M? Or would you have to recalculate the matrix M from scratch? If you can reuse M, write out the matrix formed to achieve view transformation and circle which elements would CHANGE in the above scenario.
**Question 4: (6 points) Window-to-Viewport Mapping**

A given mouse callback function maps the polyline file `dino.dat` to 6x6 tiles on a 640x480 (WxH) viewport. Calculate the (x,y) values each of the 4 corners of the tile in the 5th column, 4th row.

**Question 5 (12 points): Projection**

(15 points) For orthographic projection (gluOrtho), OpenGL projects the original view volume (cuboid) onto the Canonical View Volume (CVV). DirectX, another graphics API, does a similar orthographic projection except that its version of the CVV is shortened along the z-axis, and ranges from 0 to 1 instead of –1 to 1 for the CVV in OpenGL. The x and y ranges are the same for both DirectX and OpenGL

(1) Write out the 4x4 matrix for the **OpenGL** orthographic projection matrix and

(2) Derive the new orthographic projection matrix for DirectX
Question 6: (12 points) Transformations

a. Give the 4x4 matrix that can be used to reflect a mesh about the Z-axis in 3D

b. Matrix S is the scale matrix with scaling factors $S_x = 6, S_y = 12, S_z = 8$. Write out its inverse matrix.

c. How does instancing improve performance?

d. Using Euler's rule, write out in matrix form, the sequence of X, Y and Z rolls that can be used to achieve a rotation of 30 degrees about an arbitrary axis that has azimuth = 35 degrees and latitude = 46 degrees. You don't have to multiply these matrices together. Just write out the matrices.
Question 7 (12 points): Lighting and Materials

a. Why is it difficult to model real light sources such as a light bulb when rendering a scene?

b. Distinguish between Phong lighting and Phong shading?

c. What is a physically-based reflection model?

d. What OpenGL command and parameters would you use to specify a light source at infinity?
Question 8: (8 points) Vectors in Graphics

Give a formula for finding a reflection \( r \) the reflection of a vector \( a \) about a surface with normal \( n \). Using this formula, what is the value of the reflection vector \( r \) if \( a = (4, -2) \) and \( n = (0, 3) \).

Question 9: Coordinate Systems (6 points)

We have discussed several different coordinate spaces, including:

(i) Screen coordinate space
(ii) World coordinate space
(iii) Camera coordinate space

Finish each of the statements below with either (i), (ii), or (iii), and give a reason for your answer (3 points each).

a. Object transformations are most easily specified in...

b. A viewport is specified in...
**Question 10 (14 points): Meshes**

Consider the following declaration of a data structure to store meshes.

```cpp
//############################### VertexID ###############################
Class VertexID{
    public:
        int vertIndex;  // index of this vertex in vertex list
        int normIndex;  // index of this vertex’s normal
};
//############################### Face ###################################
Class Face{
    public:
        int nVerts;    // number of vertices in this face
        VertexID *vert;  // the list of vertex and normal indices
    Face( ){nVerts = 0; vert = NULL;} // constructor
    ~Face( ){delete[] vert; nVerts = 0;} // destructor
};
//############################### Mesh ####################################
Class Mesh{
    private:
        int numVerts;  // number of vertices in the mesh
        Point3* pt;    // array of 3D vertices
        int numNormals;  // number of normal vectors for the mesh
        Vector3 *norm;  // array of normal
        int numFaces;  // number of faces in the mesh
        Face* face;    // array of face data
    // ... others to be added later
    public:
        Mesh();   // constructor
        ~mesh( ); // destructor
        int readFile(char *filename);   // read in a filed mesh
    // ... others...
};
```
Figure 1 shows an instance of that mesh.

<table>
<thead>
<tr>
<th>NumVerts</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>pt</td>
<td>6</td>
</tr>
<tr>
<td>NumNorms</td>
<td>?</td>
</tr>
<tr>
<td>Norm</td>
<td>4</td>
</tr>
<tr>
<td>NumFaces</td>
<td>6</td>
</tr>
<tr>
<td>Face</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>-0.75</td>
<td>-0.75</td>
</tr>
</tbody>
</table>

Figure 1

Use figures 1 and 2, to answer the following questions:

a. (2 points) What is the value of numNorms? Why?

b. (2 points) Give the (x, y, z) coordinates of all the vertices of face 3 of this mesh. Note: All indexing starts from 1 and not 0.
c. (10 points) Using the Newell, calculate the normal to face 4. Note: All indexing starts from 1 and not 0. Show all your work!