CS 130
Midterm I

Winter 2013

Name	
Student ID	
Signature	

You may not ask any questions during the test. If you believe that there is something wrong with a question, write down what you think the question is trying to ask and answer that.

Question	Points	Score
True/False	10	
Multiple Choice		
1	2	
2	2	
3	2	
4	2	
5	2	
6	2	
7	2	
Written		
1	5	
2	5	
3	10	
4	10	
5	10	
Total	64	

True/False (1 pt each)

1. (T/F) Processing vertices indpendently allows the pipeline to be highly parallel.
2. (T/F) The OpenGL graphics pipeline was designed with the goal of optimizing global illumination.
3. (T/F) If two nonparallel vectors are tangent to a surface, their cross product is the normal.
4. (T/F) The canonical view volume is a frustum in perspective transforms and a cube of length, width, and height 2 in orthographic transforms.
5. (T/F) The viewing transformation matrix is M=Mcam * Mproj *Mvp.
6. (T/F) Clipping of a triangle against a plane may result in 0, 1, or 2 triangles.
7. (T/F) If we have only ambient lighting in a scene, then 3D objects will look flat.
8. (T/F) Shiny surfaces have higher Phong (specular) exponents than dull surfaces.
9. (T/F) Phong shading fixes fixes the polygonal silhouettes seen in flat and Gouraud shading.
10. (T/F) Loading and running custom programmable shaders causes them to run before the openGL default shading behavior.

Multiple Choice (2 pts each)

1. Homogenous Coordinates:

   I. Allow for translation
   II. Allow for the nonlinear effect of perspective transformation
   III. Allow us to mathematically distinguish between points and vectors

   (a) I only
   (b) I and II only
   (c) I and III only
   (d) II and III only
   (e) I, II and III

2. Which of the following are true?

   I. Z-buffering and backface culling both work on a per-fragment basis.
   II. The Painter's algorithm can't handle cycles but can handle intersections.
   III. OpenGL supports both Z-buffering and backface culling but not the Painter's algorithm.

   (a) I only
   (b) II only
   (c) III only
   (d) I, II and III
   (e) None

3. Concerning vertex and fragment shaders:

   I. Flat shading can be implemented in either the Vertex or Fragment shader.

   II. Phong shading can be implemented in either the Vertex or Fragment shader.

   III. Users can define additional data types to be passed between shaders in addition to the defaults (normals, lighting, etc.).

   (a) I only
   (b) I and II only
   (c) I and III only
   (d) II and III only
   (e) I, II and III

4. Which of the following transformations preserve Parallel lines?

   I. linear

   II. affine

   III. perspective

   (a) I only
   (b) I and II only
   (c) I and III only
   (d) II and III only
   (e) I, II and III

5. Perspective transformations

   I. are monotone in Z within the viewing frustum

   II. Preserve Z within the viewing frustum

   III. Preserve Z beyond the viewing frustum

   (a) I only
   (b) I and II only
   (c) I and III only
   (d) II and III only
   (e) I, II and III

6. Which of the following update steps would you use in DDA for the line $y = -5x$?
   a) x++; y += abs(m)
   b) x−−; y += abs(m)
   c) y++; x += abs(1/m)
   d) y−−; x += abs(1/m)

7. A point with barycentric coordinates $(-1, 1, 1)$ is:
   a) inside the triangle
   b) outside the triangle
   c) either inside or outside the triangle but there isn't enough information to tell

1 Written Response

1. (5 pts) What is the effect of applying the following matrix to a point? Be explicit: what do a, b, c, d, e, f do to the point?

$$\begin{pmatrix} a & 0 & 0 & d \\ 0 & b & 0 & e \\ 0 & 0 & c & f \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. (5 pts) Come up with a series of matrices as well as an order of multiplication (you don't need to actually perform the multiplication) to transform the triangle (0,0), (1,0), (0,3) to (-1,0), (-3,0),(-1,-6). Sketch the triangle at every step of the transformation.

3. (10 pts) Consider a ray with endpoint $\mathbf{a}$ and a normalized direction $\mathbf{u}$,

$$\mathbf{P}(t) = \mathbf{a} + t\mathbf{u}, \quad t \geq 0,$$

and a plane with normal $\mathbf{N}$ and point $\mathbf{q}$. The implicit equation is given as follows:

$$f(\mathbf{p}) = \mathbf{N} \cdot (\mathbf{p} - \mathbf{q}) = 0$$

Write pseudocode for an algorithm to find any intersection of the ray with the plane, showing the math explicitly.

4. (10 pts) Using the functions Rotate(angle in degrees), Translate(x,y), PushMatrix(), and PopMatrix(), defined analogously to those in the transformations lab, as well as the new functions DrawSun(), DrawPlanet(), and DrawMoon(), compose a scene where a sun is at the origin, orbited by two planets with two moons each. The scene is in 2D so you just need to give an angle relative to the X-axis for Rotate(), and x and y distances to Translate(). The planets orbit at a distance of 10 from the sun, 180 degrees out of phase, and the moons orbit at a distance 2 from the planets, also 180 degrees out of phase. Use T as your timer variable like in lab. Assume coordinates start properly initialized at (0,0).

5. (10 pts) Write an algorithm for rasterizing the part of a circle that falls in the first quadrant ($x \geq 0$ and $y \geq 0$), similar to the Midpoint algorithm. The circle has radius R and is centered at the origin. You do not have to fill in the interior, just draw the circumference. Write out mathematically what your function f() is.

Midpoint algorithm for a line:

```
y=y0
for x = x0 to x1 do
    draw(x,y)
    if(f(x+1,y+1/2)<0) then
        y = y + 1
```