

Introduction to Computer Graphics
Midterm Examination

Name: _____

Honor Pledge: This is a closed-book, closed-notes, independent exam. Please sign the honor pledge: On my honor as a student, I have neither given nor received information on this exam. Signed, _____

Display Technology

- 1) What is a scanline?

- 2) Frame buffers are described as having a certain depth. To what does the depth refer?

- 3) What is stored in the frame buffer when a color map lookup table is used?

Mathematical Foundations

- 5) Given two points, (a, b) and (c, d), provide the following equations of a line:

Slope-intercept:

Parametric:

- 6) Provide a geometric (use a picture) and an algebraic definition of the dot product of two vectors, $[\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z]$ and $[\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z]$

- 7) What is an orthonormal matrix?

- 8) Compute the inverse of the following orthonormal matrix:

$$\begin{bmatrix} 0.71 & 0.0 & 0.71 \\ 0.0 & 1.0 & 0.0 \\ -0.71 & 0.0 & 0.71 \end{bmatrix}^{-1} =$$

OpenGL

- 9) Use `glPushMatrix()`, `glPopMatrix()`, `glRotatef()`, `glTranslatef()`, and `glutWireCube()` to draw the hour marks on the face of a clock. The hour marks should look be unit cubes of size 1.0 and their centers should be 3.0 units from the center.

- 10) Given a variable, `v`:
- ```
Float v[3];
v[0] = 1.0; v[1] = 0.0; v[2] = 0.0;
```

Complete the following two GL functions that will use `v` as a parameter:

```
glVector3f(_____);
glVector3fv(_____);
```

## Rasterization

11) Given two integer endpoints for a line, and the following rasterization algorithm:

```
public void lineSimple(int x0, int y0,
 int x1, int y1, Color color)
{
 int pix = color.getRGB();
 int dx = x1 - x0;
 int dy = y1 - y0;

 raster.setPixel(pix, x0, y0);
 if (dx != 0) {
 float m = (float) dy / (float) dx;
 float b = y0 - m*x0;
 dx = (x1 > x0) ? 1 : -1;
 while (x0 != x1) {
 x0 += dx;
 y0 = Math.round(m*x0 + b);
 raster.setPixel(pix, x0, y0);
 }
 }
}
```

- a) Describe a case where the algorithm will poorly render a line between the two points.

- 12) One triangle rasterization technique is called *Edge Equations*. List three optimizations implemented in the following two code segments:

```
findBoundingBox(&xmin, &xmax, &ymin, &ymax);
setupEdges (&a0,&b0,&c0,&a1,&b1,&c1,&a2,&b2,&c2);

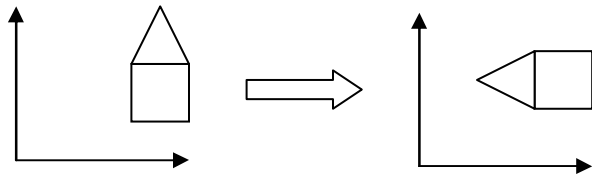
for (int y = yMin; y <= yMax; y++) {
 for (int x = xMin; x <= xMax; x++) {
 float e0 = a0*x + b0*y + c0;
 float e1 = a1*x + b1*y + c1;
 float e2 = a2*x + b2*y + c2;
 if (e0 > 0 && e1 > 0 && e2 > 0)
 setPixel(x,y);
 }
}
```

```
findBoundingBox(&xmin, &xmax, &ymin, &ymax);
setupEdges (&a0,&b0,&c0,&a1,&b1,&c1,&a2,&b2,&c2);
float e0 = a0*x + b0*y + c0;
float e1 = a1*x + b1*y + c1;
float e2 = a2*x + b2*y + c2;
int xflag = 0;
for (int x = xMin; x <= xMax; x++) {
 if (e0|e1|e2 > 0) {
 setPixel(x,y);
 xflag++;
 } else if (xflag != 0) break;
 e0 += a0; e1 += a1; e2 += a2;
}
```

## Geometric Transformations

- 16) Why do we use a 4x4 homogenous coordinate transformation matrix when describing translations, rotations, and scales, etc.?

- 18) Provide a 3x3 matrix that will compute the new vertices of a planar house centered at  $[10, 5]$  after a rotation of 90 degrees about its center:



- 18) (Worth double points) Calculate a 4x4 matrix that rotates points about the vector  $A = [1, 1, 1]$  by 90 degrees. Show your work.

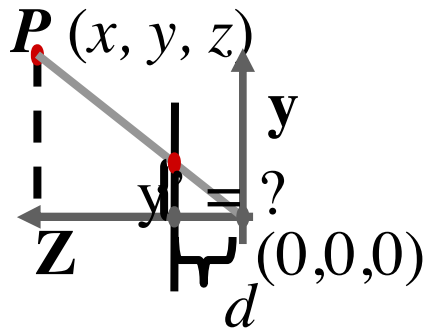
## Viewing in 3-D

19) What is the viewing frustum?

20) (Worth double points) Given a camera position  $\mathbf{P}$ , a vector normal to the image plane  $\mathbf{N}$ , and an up vector  $\mathbf{V}_{up}$ , describe how to convert a point  $\mathbf{W}$  in world coordinates to a point in camera coordinates. Provide your final answer in the form of one (or a product of many) transformation matrix. Hint, the origin in camera coordinates is located at  $\mathbf{P}$  and the world coordinate axes must be rotated to align with the camera's coordinate axes.



- 21) Calculate  $y'$  in the following figure



**View  
plane**

- 22) The 4x4 perspective projection matrix,  $\mathbf{M}_{\text{per}}$ , converts points from the camera coordinate system to the view plane coordinate system. It performs the calculation from the previous question. What is the 4x4 matrix,  $\mathbf{M}_{\text{per}}$ ?

### **Color and Light**

- 23) What is the name of the region of the retina where our vision is the sharpest?
- 24) What color are we the least sensitive to? Red, Green, or Blue
- 25) Why do monitors need to perform gamma correction?
- 26) If you were an artist, trying to carefully color polygons by hand, which color space would you prefer to work in, RGB or HSV? Why?

27) Given:

an incoming light intensity,  $I$

a surface normal,  $\mathbf{N}$

a vector from the surface to the light source,  $\mathbf{L}$

a vector from the surface to the viewer's eye,  $\mathbf{V}$

specular constant,  $k_s$

define the intensity of the light reaching the viewer as a result of the specular effects of the Phong lighting equation. Your definition may only use these variables.



28) Both Gouraud and Phong Shading interpolate along polygon edges to compute intensities. But the two shading models interpolate different things.

a) What does Gouraud Shading interpolate along edges?

b) What does Phong Shading interpolate along edges?