

G. Farin

Name: ASU ID: **Midterm**

1. Let a 4×4 matrix be given by

$$\begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

When interpreted as an affine map, what is the action of this matrix?

2. Let 3D (oriented) triangle be given by the vertices

$$\mathbf{p}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} 0 \\ 6 \\ 0 \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

Find the normal to this triangle

3. What is the aspect ratio of a rectangle? Will it change when an affine map is applied to the rectangle? Explain.

4. What is a viewing frustum? Draw a sketch.

5. Can an affine map take a rectangle to a parallelogram? Explain.

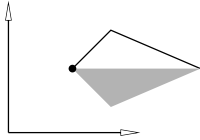
6. Suppose we have a viewport defined by the lower-left point $\mathbf{l} = [10, 10]^T$, width 20 and height 10. Given the point $[0, 0]^T$ in 2D NDC coordinates, what is the corresponding point in the viewport? Recall that NDC coordinates live in the range $[-1, 1]$ in each coordinate.

7. Explain the concept of a z -buffer.

8. Sketch an arbitrary 2D triangle and mark its vertices as $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$. Any point in the triangle has barycentric coordinates (u, v, w) . Mark all points with $u = w$.

9. Sketch the elements of the OpenGL camera model (i.e., the input parameters to `gluLookAt`).

10. 8. Circle the set of transformations which maps the light triangle to the dark triangle. The marked point is assumed to have coordinates $[50, 50]$.



- A) Translate(-50,-50), Reflect about x, Translate(50,50)
- B) Translate(-50,-50), Rotate 180, Translate(50,50)
- C) Translate(-50,-50), Reflect about y, Translate(50,50)
- D) Translate(50,50), Shear -20 in y, Translate(-50,-50)
- E) None of the above

11. Let R be a rotation matrix and let \mathbf{p} be a point. For a point \mathbf{x} , give the equation which rotates \mathbf{x} around \mathbf{r} , using the matrix R .

12. 7. Suppose `Square()` draws a unit square in the $z = 0$ plane, centered at the origin. Sketch what will be displayed as the result of the following OpenGL pseudo-code.

```
glTranslatef(2, 0, 0);  
glRotatef(45, 0, 0, 1);  
Square();
```