Midterm Review

Answer Sheet

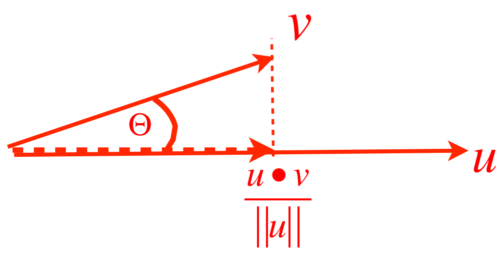
1. Provide an algebraic definition of the dot product and a geometric (picture) that shows the projection of one vector onto another using the dot product

Algebraic:

$$u \bullet v = u_x v_x + u_y v_y + u_z v_z$$

$$= \|u\| \|v\| \cos(\Theta)$$

Geometric:



Midterm Review

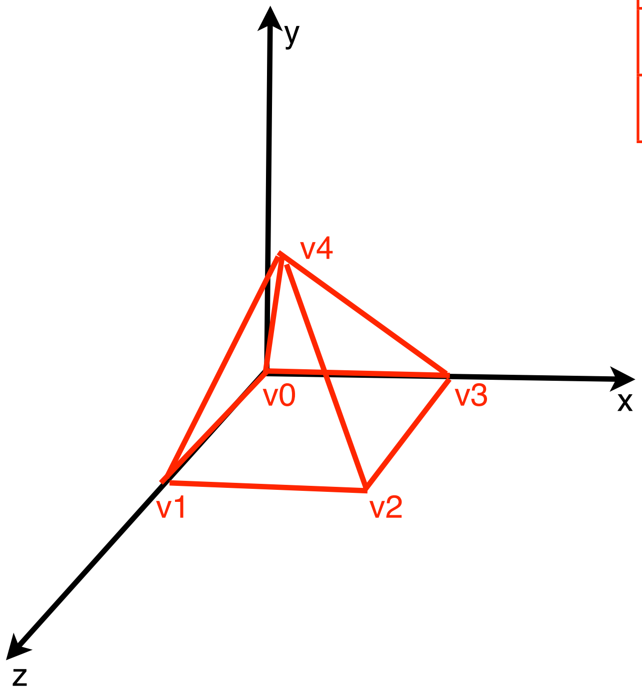
2. Set up geometric data tables as in 4-16 in the book (edge, vertex, and face) for a square pyramid (a square base with 4 triangular sides that meet at a pinnacle). Size of base and height should both be 1.0 with base at y=0 in positive x-z quadrant.

Note faces are given in counterclockwise order.

Vertices		
0.0	0.0	0.0
0.0	0.0	1.0
1.0	0.0	1.0
1.0	0.0	0.0
0.5	1.0	0.5

Faces			
e0	e5	e4	
e1	e6	e5	
e2	e7	e6	
e3	e4	e7	
e3	e2	e1	e0

Edges	
v0	v1
v1	v2
v2	v3
v3	v0
v0	v4
v1	v4
v2	v4
v3	v4



Midterm Review

3. Given a polygon with 4 vertices, write a function (pseudocode) that determines if the 4 points of a polygon lie in the same plane.

Polygon consists of points $p_1..p_4$ in counter clockwise order.

```
bool isPlaner(Point p1, Point p2, Point p3, Point p4)
{
$n = (p_1 - p_2) \times (p_3 - p_2)$
$d = -n \bullet p_2$
$h = n \bullet p_4 + d$
if (abs(h) >  0.0 (or a small tolerance)) return false;
else return true;
}
```

Midterm Review

4. Using pseudocode, write a function that will take a list of n vertices for a polygon, and determine whether the polygon is concave or convex. Assume the vertices are already planer.

```
bool isConvex(vector<Point> p, int count)
{
        if (count <4) return true; // triangles always convex

        // get cross product of edges connected to first point
        Vec3 n1 = (p[count-1]-p[0]) x (p[1]-p[0]);

        // get each succeeding cross products for all points. If they all point
        // in the same direction as the first cross product, poly is convex
        for (int i=1; i<count; ++i)
        {
                int next_idx = i+1;
                if (next_idx == count) next_idx = 0;

                // dot product with first cross product should be > 0:
                Vec3 n2 = (p[i-1]-p[i]) x (p[next_idx]-p[i]);
                if ( (n1 • n2) <0.0) return false;
        }
        return true;
}
```

Midterm Review

5. Write a procedure to determine whether an input coordinate position is in front of a polygon surface or behind it, given the plane parameters A, B, C and D.

```
// Returns true when point pos is in front of a polygon.
bool InFront(float A, float B, float C, float D, Vector pos) {
    float dist = A*pos.x + B*pos.y + C*pos.z + D
    return dist > 0.0;
}
```

Midterm Review

6. Write a routine which, given a set of input vertices representing a polygon, will determine whether a point is inside the polygon using the odd-even rule. Assume you have a function to determine if two edges intersect.

```
bool inside(std::vector<Vertex> vertices, Vertex point) {
        // Find a point (any point) outside the polygon
        float maxx, maxy = -1e10f;  // large negative #
        for (int i=0; i<vertices.size(); ++i) {
                maxx = max(vertices[i].x, maxx);
                maxy = max(vertices[i].y, maxy);
        }
        Vertex outside_point(maxx*2.0, maxy*2.0);

        // compute # of intersections between polygon edges and
        // a vector from the point to the outside point.
        int edge_intersections = 0;
        for (i=0; i<vertices.size(); ++i) {
                Vertex edge_vertex1= vertices[i];
                Vertex edge_vertex2=vertices[(i+1)%vertices.size()];
                if (computeIntersection(point, outside_point,
                                                edge_vertex1, edge_vertex2))
                        ++edge_intersections;
                }
        }

        // return true (inside) if number of edge intersections is odd
        return ((edge_intersections % 2) == 1)
}
```

Midterm Review

7. Give 2 characteristics of an orthonormal matrix.

An orthonormal matrix has rows and columns that are length 1.

The rows are perpendicular to each other and the columns are also perpendicular to each other (the dot product between each pair is 0).

The transpose of an orthonormal matrix is also its inverse.

Midterm Review

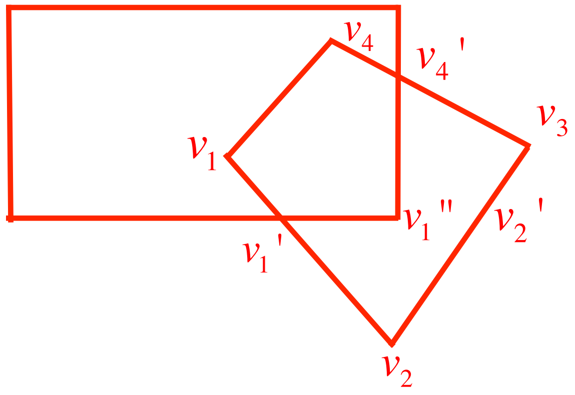
8. Explain how we can check the region codes (also called out codes) to determine if a line can be trivially rejected using the Cohen-Sutherland line clipping algorithm.

If any of the bits in the region codes for the two endpoints of a line are both 1, then the entire line segment is outside the viewport on the left/right/bottom/top boundary according to which bits match.

```
bool trivialReject(int c1, int c2)
{
        return (c1&c2);
}
```

Midterm Review

9. Draw a quad overlapping a view window with at least one full edge outside the window. Using the Sutherland-Hodgman polygon clipping algorithm, show incrementally how the clipped vertices are added to create the final clipped polygon.



Clip against bottom edge:

input	output
v4-v1	v1
v1-v2	v1’
v2-v3	v2’ v3
v3-v4	v4

Clip against right edge:

v4-v1	v1
v1-v1’	v1’
v1’-v2’	v1” (the lower right corner of the screen)
v2’-v3	(both outside right edge: no output)
v3-v4	v4’ v4

Final (v1, v1’, v1”, v4’, v4) // skip top and right - no vertices outside)

Midterm Review

10. Give another homogeneous uniform scaling matrix that will scale points relative to the origin by the same amount as this one.

$$S_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling the scale factors and homogeneous coordinate by the same amount produce the same result.

$$S_1 = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Midterm Review

11. Prove that uniform scaling and rotation form a commutative pair of operations but that, in general, scaling and rotation are not commutative operations

Uniform :

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -a\sin(\theta) & 0 \\ a\sin(\theta) & a\cos(\theta) & 0 \\ 0 & 0 & a \end{bmatrix}$$

$$\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix}\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -a\sin(\theta) & 0 \\ a\sin(\theta) & a\cos(\theta) & 0 \\ 0 & 0 & a \end{bmatrix}$$

Non – uniform :

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -b\sin(\theta) & 0 \\ a\sin(\theta) & b\cos(\theta) & 0 \\ 0 & 0 & c \end{bmatrix}$$

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a\cos(\theta) & -a\sin(\theta) & 0 \\ b\sin(\theta) & b\cos(\theta) & 0 \\ 0 & 0 & c \end{bmatrix}$$

Midterm Review

12. Show that the transformation matrix below (7-55 in the book),

0 1 0
1 0 0
0 0 1

for a reflection about the line y=x is equivalent to counterclockwise rotation of 90 degrees followed by a reflection across the y axis

$$\cos(90) = 0;$$

$$\sin(90) = 1;$$

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(90) & -\sin(90) & 0 \\ \sin(90) & \cos(90) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Midterm Review

13. Given a clipping window with 2D coordinates of (xwmin, ywin)..(xwmax, ywmax) give matrices that when applied in the order you specify, will transfer (xwmin, ywmin) to (-1.0, -1.0) and (xwmax, ywmax) to (1.0, 1.0).

First move the window center to 0,0 then scale it to size (2,2).

$$ST = \begin{bmatrix} \frac{2}{xw_{\max} - xw_{\min}} & 0 & 0 \\ 0 & \frac{2}{yw_{\max} - yw_{\min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{xw_{\max} + xw_{\min}}{2} \\ 0 & 1 & -\frac{yw_{\max} + yw_{\min}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Midterm Review

14. Given a triangle with vertices at a (0,0), b(4, 1), and c(3,3), provide the matrix transformation or transformations that will double the size of the triangle while keeping vertex c fixed to it's current position.

Translate (c) to the origin, scale by a factor of 2, then translate c back to (3,3)

$$\begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

Midterm Review

15. Given a camera at position p1, an object you want to view at position p2, and an up vector "y", describe how to convert a point w1 in world coordinates into a point in camera coordinates. Give the final answer in terms of a homogeneous 4x4 matrix (or more than one matrix multiplied together).

1) Translate the camera to the origin (-p1) (the translation matrix)
2) Define the (u,v,n) axes of the view matrix
   a. The normalized vector (p1-p2)/lp1-p2l defines the 'n' axis
   b. The normalized crossproduct of the up vector 'y' and the 'n' axis define the u-axis
   c. The normalized cross product of the n-axis and the u-axis define the v-axis

$$n = \frac{p_1 - p_2}{|p_1 - p_2|}; u = \frac{y \times n}{|y \times n|}; v = n \times u;$$

3) The rotation matrix is constructed by making rows equal to the u,v, and n. Rotation and translation concatenated together transform a point w1 from world coordinates to view coordinates.

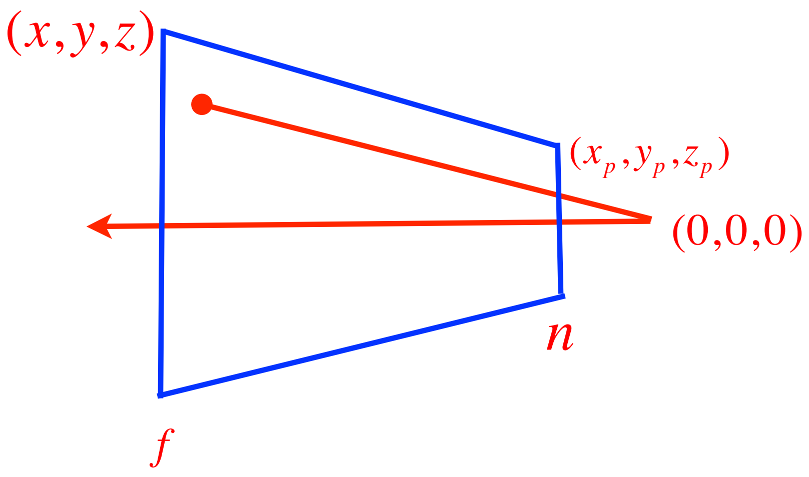
$$M = RT = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Midterm Review

16. Describe in words the difference between an orthogonal and a perspective projection. Draw a diagram of a perspective projection with a viewing point at (0,0,0), a near clipping plane that corresponds to the view plane, and a far clipping plane. Diagram the projection of a visible vertex onto the viewing plane. Give a matrix that will perform that operation.

An othogonal projection maps points in the view volume to the view plane along lines that are parallel to the view plane normal vector ‘n’.

A perspective projection maps points in the view volume along paths that intersect the view plane and converge to a single point behind the view plane called the projection reference point. This creates a foreshortening effect in which objects that are further away appear smaller.



$$M_{pers} = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & s_z & t_z \\ 0 & 0 & -1 & 0 \end{bmatrix}$$