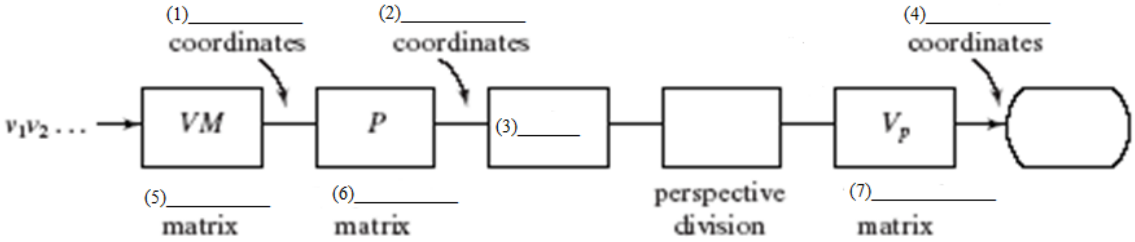
CS4550/5550 Foundations of Computer Graphics
Homework3 (7% + EC 3%) – Review Questions

- 10 points of HW3 total score.
- You will be assigned to solve a particular question.
- Submit your solution for 1~9 via CSNS by Monday, 11/20/2017, and present it on Tuesday, 11/21/2017.
- Submit your solution for 10 by email by Monday, 11/27/2017, and present it on Tuesday, 11/28/2017.

1. Modeling
    a. What are three lists defining a polygonal mesh?
    b. What is a prism?
    c. Why do you assign a normal to each vertex (instead of a normal to each face) for a polygonal mesh?
    d. Consider a polygon with vertices (0,0,0), (1,0,0), (0, 1,0), (1,1,1). Find the normal to this polygon using the Newell's method. Is the polygon planar?
    e. Define a surface of revolution.
    f. Given a parametric curve *C(t),* define *qualitatively* the Frenet frame at *t(i)* at a point *P (Px, Py, Pz).*

2. The OpenGL graphics pipeline:
    a. Each vertex goes through the fixed stages of the pipeline before being drawn on the screen. Fill (1) ~ (7) for corresponding stages.



    b. Which matrix transforms a point from the world space to the camera space?
    c. What happens after the projection matrix (i.e. what is the input and output of the projection matrix)?

3. Suppose that your program calls `gluLookAt(4,4,4,0,0,0,0,1,0)`
    a. What is the View Plane Normal (VPN)?
    b. What is the 4x4 View matrix V generate by OpenGL?
    c. Where in the camera coordinate systems is the point P (1,1,1) of the world coordinates?
    d. Roll the camera by 45 degree. What is the new matrix V?
    e. After rolling the camera, where in the camera coordinate systems is the point P (1,1,1) of the world coordinates?

4. Under perspective projection, do two parallel lines appear as parallel lines on the screen? If they do, why? Under parallel projection, do two parallel lines appear as parallel lines on the screen? If they do, why?

5. Parallel lines and vanishing point
    a. What is the vanishing point?
    b. Let's consider two line segments AB and CD in the camera coordinate systems given A=(1,1,10), B=(3,4,10), C=(-2,2,17), and D=(0,5, 17). Are they parallel? Are projections of these lines parallel? If not, what is the location of the vanishing point?
    c. Let's consider two line segments AB and CD in the camera coordinate systems given A=(1,1,12), B=(3,4,10), C=(-2,2,17), and D=(0,5, 15). Are they parallel? Are projections of these lines parallel? If not, what is the location of the vanishing point?

6. Perspective transformation and Canonical View Volume (CVV)

a. Given the function `gluPerspective(60, 640.0/480.0,1, 300)`, what are the top, bottom, left, and right of the view volume?
b. What is the pseudo depth? How does viewport transformation map the pseudo-depth information?
c. OpenGL Perspective transformation transforms the view volume into the canonical view volume (CVV). What is the CVV? What is(are) the advantage(s) of using CVV?

7. Canonical View Volume (CVV) and Clipping
    a. In OpenGL graphics pipeline, are the input and output to/from the clipping algorithm against the CVV in the Homogeneous coordinate system? If so, why?
    b. Assume that you want to clip a line AB segment of two points A=(0,0,0) and B=(2,2,2) against CVV using the modified Liang-Barsky algorithm. What are the six boundary codes computed for each end point of the edge? (p.358). What are t(in) and t(out) values computed by the algorithm assuming it is an entering edge?

8. Shading
    a. What is shading?
    b. What is the diffuse reflection? How do you compute it? What is the additional information you need to compute specular reflection (compared to diffuse reflection)?
    c. In OpenGL graphics pipeline, where does the shading take place?
    d. What is the difference specifying the light position as a vector from specifying it as a point in OpenGL programming for shading?
    e. What are differences among flat, Gouraud and Phong shading?
    f. Assume that your program rasterizes a triangle (given by three vertices) on the screen using the Gouraud shading. How does it compute the colors of pixels inside the triangle? Describe it qualitatively.
    g. Recall that the transformation of a normal N given an arbitrary transformation matrix M applied to the points requires us to take the inverse transpose of M before applying it to N. N' = $(M^{-1})^T$ N (equation 1). If M were to be only composed of a rotation R, then we can simply apply *R* directly to N. N' = *R* N (equation 2). Explain why equation 2 is correct by using equation 1.

9. Visibility
    a. What is backface culling? Write the equation that checks if a face is back-face or not using n (the normal of the face) and the VPN.
    b. Why do we need Z-buffer? If we change the camera location or orientation, do we need to calculate z-values for the z-buffer? Explain your answer.
    c. For object visibility test, do we need z-buffer algorithm even though we cull back-faces? Explain your answer.
    d. Write at least two disadvantages of using the z-buffer algorithm.

10. This is for graduate students only.

- Consider the following OpenGL program that draws two lines and answer all 10 questions below.

```
glutInitWindowSize(500,500);
...
glViewport(0, 0, 500, 500);
...

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(50.0f, 1.0f, 1.0f, 200.0f);

...

glMatrixMode(GL_MODELVIEW);
gluLookAt(0,0,4,0,0,0,0,1,0);

...

glClear(GL_COLOR_BUFFER_BIT);

// first line
glBegin(GL_LINES);
glVertex3f(0,0,0);     // p1
glVertex3f(0,1,0);     // p2
glEnd();

// second line
glTranslatef(1,1,1);
glBegin(GL_LINES);
glVertex3f(0,0,0);     // p3
glVertex3f(0,1,0);     // p4
glEnd();
glFlush();

...
```

<u>Hints:</u> Refer to the following hints if necessary.

$$V = \begin{pmatrix} u_x & u_y & u_z & d_x \\ v_x & v_y & v_z & d_y \\ n_x & n_y & n_z & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
n = eye - look
u = up x n
v = n x u
dx = -eye ·u, dy= -eye ·v, dz= -eye ·n
```

$$top = N \tan(\frac{\pi}{180} viewAngle / 2)$$

$$bott = -top$$

$$right = top * aspect$$

$$left = -right.$$

$$P = \begin{pmatrix} \frac{2N}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2N}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & \frac{-(F+N)}{F-N} & \frac{-2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

1) What are the end points of two lines in the world coordinate system?

In the world coordinate system

Line 1: P1 = (, , ,)
Line 1: P2 = (, , ,)
Line 2: P3 = (, , ,)
Line 2: P4 = (, , ,)

2) What is the 4x4 view matrix **V** generate by the OpenGL program? (Do not forget to normalize u, v, and n.)

3) Where in the camera coordinate system are these end points of two lines?

In the camera/eye coordinate system

Line 1: P1 = (, , ,)
Line 1: P2 = (, , ,)
Line 2: P3 = (, , ,)
Line 2: P4 = (, , ,)

4) What are the top, bottom, left, and right of the view volume defined by the program?

5) Where in CVV are the end points of two lines?

In CVV

Line 1: P1 = (, , ,)
Line 1: P2 = (, , ,)
Line 2: P3 = (, , ,)
Line 2: P4 = (, , ,)

6) Show if these two lines are parallel or not.

7) Show if the projections of two lines are parallel? If they are not parallel, where is the vanishing point?

8) If necessary, perform Liang-Barsky clipping for each line and write new endpoints.

In CVV after clipping

Line 1: P1 = (, , ,)
Line 1: P2 = (, , ,)
Line 2: P3 = (, , ,)
Line 2: P4 = (, , ,)

9) After projection, where on the image view plane do the end points of two lines (or clipped ones) appear?

On the image view plane of CVV

Line 1: P1 = (,)
Line 1: P2 = (,)
Line 2: P3 = (,)
Line 2: P4 = (,)

10) Where on the viewport do the end points of two lines (or clipped ones) appear?

On the viewport

Line 1: P1 = (,)
Line 1: P2 = (,)
Line 2: P3 = (,)
Line 2: P4 = (,)