CS 559: Computer Graphics

Homework 4 - Optional

This homework will be graded to give you feedback before the midterm, but the grades will not be recorded. If you hand it in to Prof Chenney's mailbox on the 5th floor of the Computer Science building by 5pm on Monday March 19, then Raj will have it ready to give back in office hours on Wednesday March 21, the day before the Midterm. Solutions will be posted Monday March 19 after 5pm.

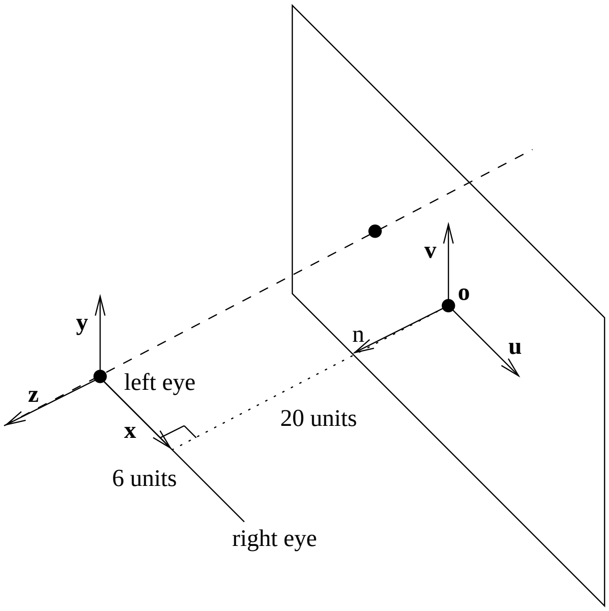
Question 1:

This question is about producing images for stereo viewing. Assume that the viewer's eyes are 6 units apart, and that the line joining the midpoint of their eyes to the origin of the image plane is normal to the image plane. Also assume that the image plane is 20 units from the viewer's eyes, and that the line joining their eyes is parallel to the horizontal axis on the image plane.

Call the normal to the image plane, in the direction of the viewer, $\mathbf{n}$. Call the horizontal axis in the image plane $\mathbf{u}$ and the vertical axis $mathbfv$. These vectors are all to be specified in the world coordinate system. Finally, define the origin of the image plane to be at the point $\mathbf{o}$, also in world space. Together, $\mathbf{n}, \mathbf{u}, \mathbf{v}$ and $\mathbf{o}$ define the view coordinate system. Let the image be 8 units wide in the $\mathbf{u}$ direction, and 6 units high in the $\mathbf{v}$ direction.

For stereo viewing, we wish to produce an image with the center of projection at the left eye, and then another with the center of projection at the right eye. Assume that things have been set up with OpenGL so that it's origin is at the left eye, and it is looking in a direction parallel to $\mathbf{n}$. In other words, the *axis of projection*, is parallel to $\mathbf{n}$ and passing through the left eye. It is shown dashed in the figure above. Use the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ to represent the axes of OpenGL's coordinate system. The $\mathbf{x}$ axis is parallel to $\mathbf{u}$, they $\mathbf{y}$ axis is parallel to $\mathbf{v}$, and the $\mathbf{z}$ axis is parallel to $\mathbf{n}$.

The whole situation is pictured below.



1. A user specifies $\mathbf{n}$ and $\mathbf{o}$ in world coordinates, and they specify a direction in world coordinates, $\mathbf{up}$, that they wish to appear vertical in the image. How can $\mathbf{v}$ be determined from the user specified information?

2. Having computed $\mathbf{v}$, how is $\mathbf{u}$ computed?

3. What is the location of the left eye in $\mathbf{u}, \mathbf{v}, \mathbf{n}$ coordinates?

4. What is the location of the left eye in world coordinates, written as an equation involving $\mathbf{o}, \mathbf{n}$ and $\mathbf{u}$?

5. What rotation and translation matrices are required to transform points from world space to OpenGL's $\mathbf{x}, \mathbf{y}, \mathbf{z}$ space, and in what order? The matrices will involve $\mathbf{n}, \mathbf{u}, \mathbf{v}$ and $\mathbf{o}$. (Hard. Hint: The translation will involve the left-eye position computed above. The left eye point in world space must be transformed to (0,0,0) in OpenGL space.)

6. What are the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ coordinates of the bottom left corner of the image in OpenGL's coordinate frame?

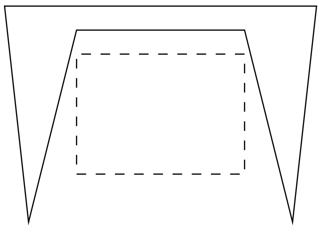
7. What are the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ coordinates of the top right corner of the image in OpenGL's coordinate frame?

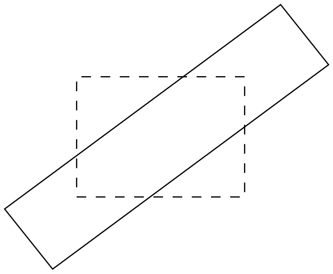
To tell OpenGL to produce an off-center image such as this, you would need to explicitly give it the rotation and transformation matrices from step 5, and the coordinates from steps 6 and 7.

To see why this would produce a stereo pair, consider what each eye would see if you looked at an object behind the image plane. The left eye and right eye images will reflect these views. Then, to display the pair, you put them side by side and defocus your eyes so that your left eye sees the left image and your right eye sees the right image. Your brain fuses the two images into a stereo pair. If you decide to try try this at home, you will probably find that it helps to exaggerate the eye separation.

Question 2:

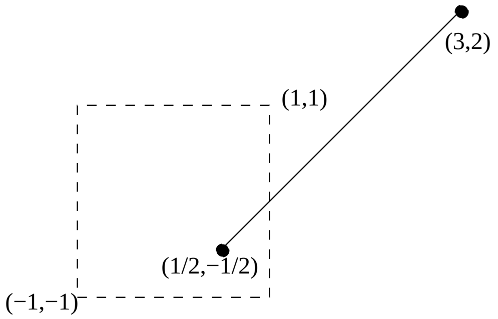
Perform Sutherland Hodgman clipping on each of the figures below to the rectangular clip region shown dashed. Show the intermediate results after clipping with the top edge, the results after clipping with the top and right edges, the results after the top, right and bottom edges, and the final results. (You should show 8 figures in all.)





Question 3:

This question explores Liang-Barsky clipping. Consider the line segment and clip region shown below.



1. What is the parametric equation of the line? Write it in the form:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + t \begin{bmatrix} c \\ d \end{bmatrix}$$

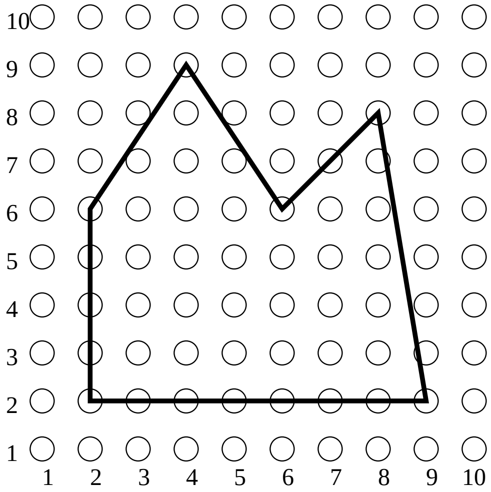
2. What are the parametric coordinates (the t values) for the intersections of the line with each clip edge? Label them as entering or leaving intersections.
3. What are the parametric coordinates of the endpoints of the visible segment?
4. What are the (x,y) coordinates of the endpoints of the visible segment?

Question 4:

Fill out a table with the per-pixel values for x_i, y_i and p_i when Bresenham's algorithm is applied to the line from (3,2) to (10,4). There will be eight sets of values.

Question 5

Consider a sweep fill of the polygon shown:



1. Write down the edge table.
2. Show the active edge list for rows 1 to 10.
3. Draw a diagram showing which pixels are filled. You might just want to cut out and draw on the figure above. (Attach it well.)