

HOMEWORK #2

THE ART CRITIC

This assignment may be done in teams of two. **Due Feb 21, 2012 by 11:59PM** via D2L. .



Figure 1. Scene concept: perusing a wall of art.

For this assignment, you will simulate the view of an art critic as she takes a virtual tour along the wall of an art gallery. Imagine that beautiful works of art line this wall. Call them $pic_1, pic_2 \dots pic_5$, and include them with your submission. Your program should simulate the art critic's view as she walks slowly along the wall, examining each picture in turn as shown in Fig.1. In particular, your scene construction must consider the following:

- The critic's walk is along a straight line parallel to the wall. The critic is looking both at the wall and forward, such that her view axis forms a 45° angle with the wall's surface.
- The pictures should be large enough that, as the critic walks along the wall, one picture fills much of her view as she gets close to it. As the critic moves along the wall, there will also be times when two pictures are indeed seen, as pic_i moves out of view and pic_{i+1} comes into view.

- The wall should be infinitely long, so that when the critic reaches the last picture pic_n , the series restarts with pic_1 , then pic_2 , etc. Although the wall has both top and bottom edges, you can ignore the existence of the ceiling and floor.
- Use perspective projection to render the critic's view. As the view axis forms a 45° angle with the wall, perspective foreshortening should be observed in your scene for both the wall and pictures.
- The construction of your scene should *resemble* Fig.1. Nonetheless you are free to make individual decisions about the critic's field of view and walking speed, as well as the size of the pictures, their placement on the wall, etc. Excluding the walking motion, you may assume that all other parameters of the scene remain constant.

Your program should render the scene using the following draw modes:

- (1) **Basic Mode:** If the 'b' key is pressed, only pixel-level GL operations are allowed. You should only plot colors pixel-by-pixel using `glVertex` and `glColor`, as in HW1. To implement "Basic" mode, you should consider only what is seen by each pixel in the window space. Imagine a line (*i.e.* ray) extending from the center of projection through a pixel on the projection plane into the view volume (Fig.2). Determine the intersection of this line with the wall full of pictures. The color of the pixel is the color of the wall, or a picture on it, at the point of intersection. Do not worry too much about the efficiency of your solution here.

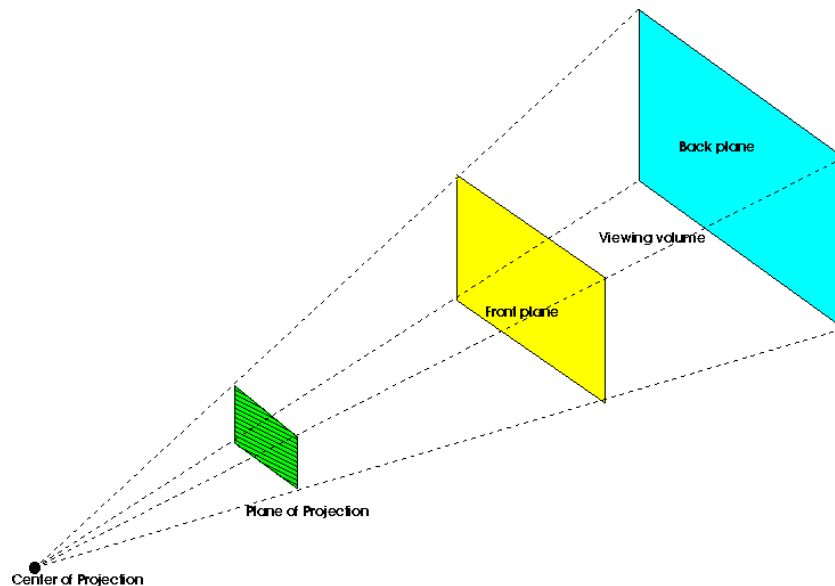


Figure 2. Geometric construction of view volume and projection plane.

- (2) **GL Mode:** If the 'g' key is pressed, your program should draw the scene using the hardware-accelerated GL pipeline. Your scene construction should be similar to Basic mode, but it does not have to be identical. For this mode you are free to use any GL or GLU functions that you find convenient to draw the scene. For instance, you may exploit GL to maintain the view and perspective transformations and to perform polygon filling. You may use textured polygons to represent the gallery pictures. Note that your GL mode should run at interactive frame rates! For specifics on initializing your program for texturing and polygon filling, please consult the GL tutorials and documentation. You may also find the following functions useful:

gluLookAt: defines the eye (camera) pose in world coordinates

glFrustum: defines the sides of a perspective view volume

gluPerspective: builds a view volume based on FOV, aspect ratio

glBindTexture: activates an image ID for texture mapping

glTexCoord2{f,d}: defines texture coordinates for vertices