Name _____

CS445 Final Exam
Fall 2010

1.	(max = 12)	5.	(max = 9)
2.	(max = 15)	6.	(max = 6)
3.	(max = 12)	7.	(max = 22)
4.	(max = 8)		
Final Score _____(max=84)			

1) (4 pts each, 12 pts total) **2D Transforms**: What is the 3x3 matrix transform (or sequence of transforms) in homogeneous coordinates for the following. <u>Also give the inverse.</u>
   a) A translation by 3 along x and -4 along y.

   The transform: The inverse:

   b) A uniform scale by 2 about the fixed point (a,b):

   The transform: The inverse:

   c) A projection onto the y-axis.

   The transform: The inverse:

2) (5 pts each, 15 pts total) Suppose you want to apply the following texture to the square surface on the right.

If the code below is applied, the square will be textured as shown below on the right.

```
gl.glTexParameteri(GL.GL_TEXTURE_2D,
    GL.GL_TEXTURE_WRAP_S.GL.GL_ CLAMP);
gl.glTexParameteri(GL.GL_TEXTURE_2D,
    GL.GL_TEXTURE_WRAP_T.GL.GL_CLAMP);
gl.glBegin(GL.GL_POLYGON);
  gl.glTexCoord3f(0,0);   gl.glVertex3f(v0x,v0y,0)
  gl.glTexCoord3f(1,0);   gl.glVertex3f(v1x,v1y,0);
  gl.glTexCoord3f(1,1);   gl.glVertex3f(v2x,v2y,0);
  gl.glTexCoord3f(0,1);   gl.glVertex3f(v3x,v3y,0);
gl.glEnd();
```
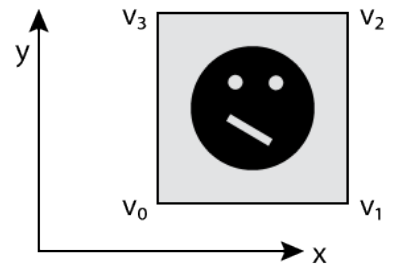

In the rectangle on the right, sketch how the texture would appear if the following texture coordinates are used instead?

a. gl.glTexCoord3f(-0.5,0);   gl.glVertex3f(v0x,v0y,0);
    gl.glTexCoord3f(0.5,0);   gl.glVertex3f(v1x,v1y,0);
    gl.glTexCoord3f(0.5,1);   gl.glVertex3f(v2x,v2y,0);
    gl.glTexCoord3f(-0.5,1);   gl.glVertex3f(v3x,v3y,0);
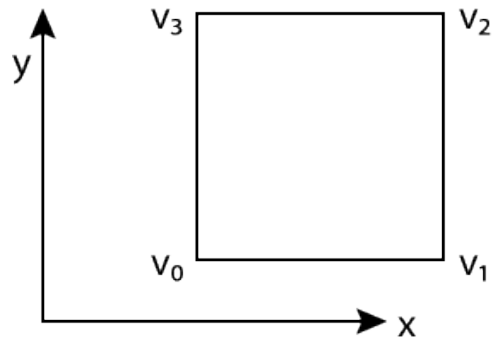
b. gl.glTexCoord3f(1,0);   gl.glVertex3f(v0x,v0y,0);
    gl.glTexCoord3f(1,1);   gl.glVertex3f(v1x,v1y,0);
    gl.glTexCoord3f(0,1);   gl.glVertex3f(v2x,v2y,0);
   gl.glTexCoord3f(0,0);   gl.glVertex3f(v3x,v3y,0);
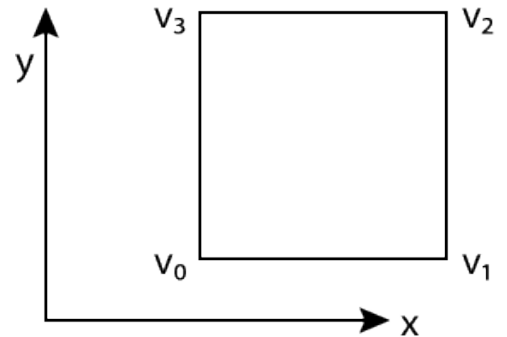
c. gl.glTexCoord3f(0,0);   gl.glVertex3f(v0x,v0y,0);
    gl.glTexCoord3f(1,0);   gl.glVertex3f(v1x,v1y,0);
    gl.glTexCoord3f(1,2);   gl.glVertex3f(v2x,v2y,0);
    gl.glTexCoord3f(0,2);   gl.glVertex3f(v3x,v3y,0);

2
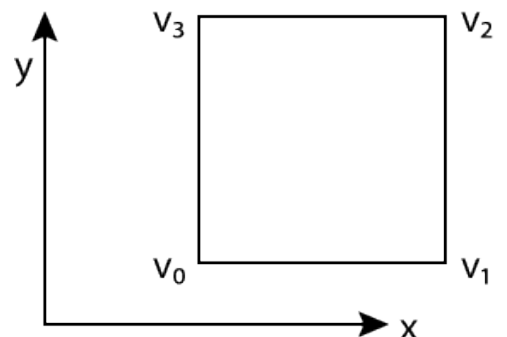
3) (4pts each, 12 pts total) Blending in OpenGL: Suppose that you draw two rectangles S_1 and S_2, using OpenGL where
- S_1 is closer to the camera than S_2.
- S_1 is drawn first followed by S_2.
- The background color is set to white=(1,1,1).
- the glColor is set to green=(0,1,0, α_1) when S_1 is drawn.
- the glColor is set to red=(1,0,0,α_2) when S_2 is drawn.
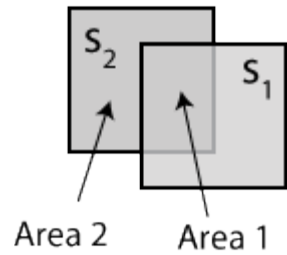
What is the _resulting (r,g,b) color in the frame buffer_ corresponding to each of the following. *Show how you calculated your answer* (you will not get full credit for just an answer written down)

a. **Area** 2 if α_1=0.5, α_2=0.6 when z-buffering is turned on.

(r,g,b) = _____

b. **Area** 1 if α_1=0.4, α_2=1.0 when z-buffering is turned on.

(r,g,b) = _____

c. **Area** 1 if α_1=0.25, α_2=0.9 when z-buffering is turned off.
(r,g,b) = _____

Area 2   Area 1

3

4) (4 pts each, 8 pts total) **Fly thoughs**: Given the viewer location *Loc*, and the direction/orientation vectors *Forward*, *Right*, and *Up*, how do you update *Loc, Forward, Right, and Up* in order to do the following. Indicate not only what changes but also what doesn't change.

a) Move the viewer a small distance (e.g. α) forward.

b) Turn the viewer a small amount to the left.

5) (3 pts each, 9 pts total) The position of a light can be set in any number of places in your code, e.g. before or after gluLookAt, at some point in scene hierarchy, etc. Where should the position be set if

a) You want the light to move with the camera?

b) You want the light to stay fixed in the world coordinate system?

c) You want the light to be attached to an object (e.g. if you had a headlight on a car).

6) (6 pts) Cross Products: If v1 = (1,0,0) and v2 = (0, 0, 1), what is

a) v1 x v2 = _____

b) v1 x v1 = _____

7) (22 pts total) Recall that *OpenGL uses a right handed coordinate system where the default camera looks down the negative z world axis, the y axis is up, and the x axis is to the right.* In this default position, the world coordinates of a vertex p_w are the same as its camera coordinates p_e.

a) (3 pts) Suppose the camera starts in the default position and is translated one unit back (along positive z). If a vertex has world coordinates $p_w = (0,0,0)$, what would be its new camera coordinates, p_e? (no math required – just reason it through)

b) (3 pts) Suppose the camera starts in the default position and is rotated by 90 degrees about x so that it is looking directly up along the y world axis. If a vertex has coordinates $p_w = (0,0,-1)$, what would p_e be? (no math required – just reason it through)

(Continued on next page)

c) (16 pts) Suppose we want to reposition the camera given the parameters:
$\qquad$ `Eye` = position of the camera
$\qquad$ `Look` = what the camera is looking at
$\qquad$ `Up` = the up vector
We can use the function `gluLookAt(`*Eye*`, Look, Up)` to calculate `M`, where `M` is the 4x4 matrix used to transform vertices p_w (world coordinates) into p_e (eye coordinates):
$$p_e = M\ p_w$$

How does gluLookAt calculate `M` from `Eye`, `Look`, and `Up`? That is, what is `M` in terms of `Eye`, `Look`, and `Up`?   (Hint: first calculate camera coordinate vectors `u,v,n` and use them to compute the rotational part of `M`. Then determine the translational part of `M` and combine with the rotational part to get to `M`. Use homogeneous coordinates.)