CPSC 314, Midterm Exam 1

9 Feb 2007

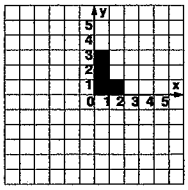
Closed book, no calculators or other electronic devices. Cell phones must be turned off. Place your photo ID face up on your desk. One single-sided sheet of handwritten notes is allowed. At the end of the exam, turn in the note sheet with the exam.
Do not open the exam until told to do so. Answer the questions in the space provided. If you run out of room for an answer, continue on the back. There are 100 points, you have 50 minutes.
Good luck!

Name: Solutions

Student Number: _____

Question	Points Earned	Points Possible
1		24
2		8
3		10
4		12
5		16
6		14
7		16
Total		100

1. (24 pts) For each equation below, sketch the new location L' of the L shape on the grid and provide the OpenGL sequence needed to carry out those operations. Use the function `drawL()`, which draws an L shape with the lower left corner at the current origin as shown below. You may assume the matrix mode is `GL_MODELVIEW` and that the stack has been initialized with `glLoadIdentity()`. For reference, the OpenGL command syntax is `glRotatef(angle,x,y,z)`, `glTranslatef(x,y,z)`, `glScalef(x,y,z)`.



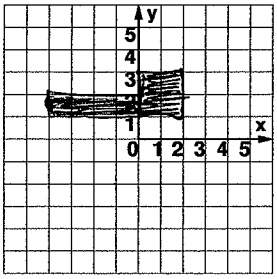

```
drawL();
```

glScale(2,1,1) glTranslate(1,1,0) glRotate(90,0,0,1) glScale(-1,1,1)

$$\mathbf{A}=\begin{bmatrix}2&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix},\mathbf{B}=\begin{bmatrix}1&0&0&1\\0&1&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix},\mathbf{C}=\begin{bmatrix}0&-1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix},\mathbf{D}=\begin{bmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

OR glRotate(180,0,0,1)

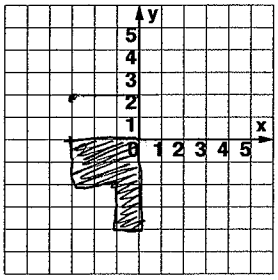
a) L' = ABC L



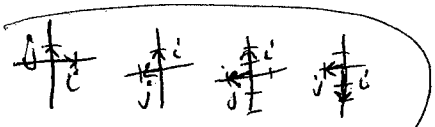

```
glScale(2,1,1);
glTranslate(1,1,0);
glRotate(90,0,0,1);
```

careful: rotate after scale!
use global, not frames, to figure out

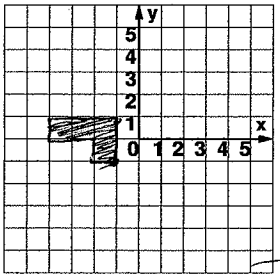
b) L' = CAD L




```
glRotate(90,0,0,1)
glScale(2,1,1)
glScale(-1,1,1)
```

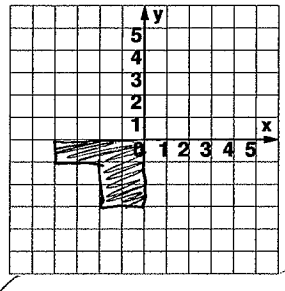


c) L' = CBD L

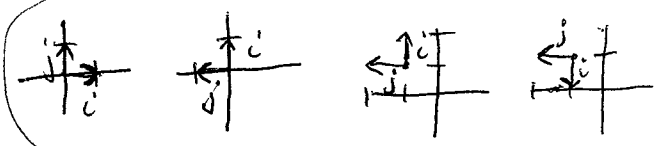


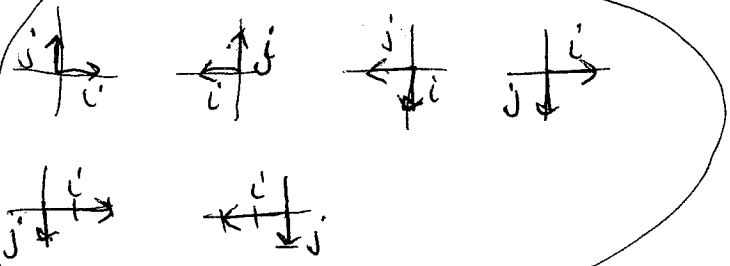

```
glRotate(90,0,0,1)
glTranslate(1,1,0)
glScale(-1,1,1)
```

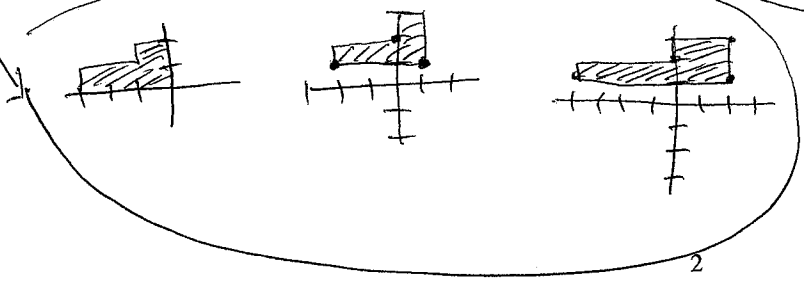
d) L' = DCCAD L




```
glScale(-1,1,1)
glRotate(90,0,0,1)
glRotate(90,0,0,1)
glScale(2,1,1)
glScale(-1,1,1)
```

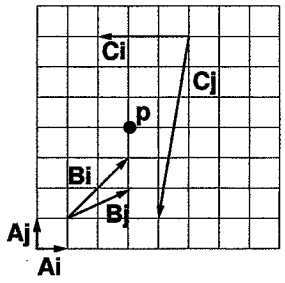






marking scheme:
5 pts for picture, 1 pt for OpenGL
3/5 if frame correct but L drawn wrong within it

2. (8 pts) The point **p** can be specified as $\mathbf{p}_A = (3,4)^T$ in the coordinate frame A, with orthonormal basis vectors **i** and **j**. Specify the coordinates of point **p** in frames B and C.



$B = (2, -1)$

$C = (.5, .5)$

marking scheme: 4 pts each

3. (10 pts) Find the 3x3 homogeneous transformation which transforms a point from Frame C into the Frame A coordinate system. That is, give M where $\mathbf{p}_A = M\mathbf{p}_C$. Verify your solution using one of your answers to question 2.

${}_A C_i = (-3, 0)$

${}_A C_j = (-1, -6)$

${}_A C_o = (5, 7)$

$$M = \begin{bmatrix} -3 & -1 & 5 \\ 0 & -6 & 7 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} .5 \\ .5 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.5 - .5 + 5 \\ -6 \cdot 5 + 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix}$$

construct M as change of basis

marking: 3 pts verify, 7 pts matrix (4 pts rotation, 3 pts translation)

transposed rotation 2/4

attempt to verify w/o matrix 1/3

4. (12 pts) True/false

- F • Specifying the eye point, lookat point, and up vector completely determines a projective camera transformation. params for lookat
- T • If a surface is transformed by a rotation, transforming its normal vector by the same rotation will leave it perpendicular to the surface. only nonuniform scales are a problem
- F • The homogeneous points (2,5,2,2) and (2,5,0,4) map to the same Cartesian point after homogenization. $(1, 5/2, 1, 1) \neq (1/2, 5/4, 0, 1)$
- F • Including an affine transformation in a display list will cause the object to shear with respect to the image plane.
- F • After undergoing an orthographic projection, a unit cube will appear to have either 1 or 2 vanishing points, but cannot have 3. orthographic means no vanishing points at all
- T • A nonuniform scaling transformation leaves the w coordinate of a homogeneous point unchanged.
- F • Both oblique and orthographic projections have projectors perpendicular to the projection plane. oblique: projectors oblique to plane
- F • The transformation from an orthographic view volume to the normalized device coordinate system depends on the size of the viewport. viewport affects NDC to display
- T • After transforming from a perspective view volume to the normalized device coordinate system, the x coordinate of a visible point has a range of 2. yes, -1 to 1
- T • Affine transformations can change the origin of the local coordinate frame. yes, includes translations
- T • $BAp = ABp$ when

translations commute

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

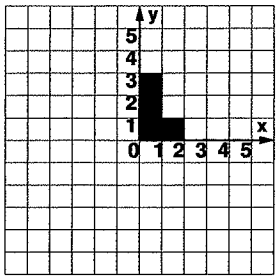
marking: 1 pt each except last is 2 pts

5. (16 pts) Draw shapes 2, 3, 4, and 5 transformed by the appropriate OpenGL commands in the left column below. The `drawShape()` code is shown in the middle column, and the result of the first call is shown in the right column.

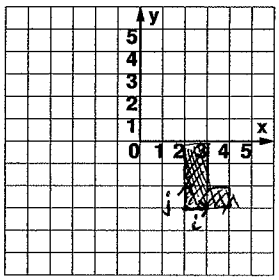
```
glIdentity();
drawShape(); // shape 1
glTranslate(2,-3,0);
drawShape(); // shape 2
glRotate(90, 0, 0, 1);
drawShape(); // shape 3
glPushMatrix();
glTranslate(1, 0, 0);
drawShape(); // shape 4
glTranslate(0, 2, 0);
glScale(2,1,1);
glRotate(90, 0, 0, 1);
glTranslate(-1, 0, 0);
glPopMatrix();
glScale(1, .5, 1);
glTranslate(0, -2, 0);
drawShape(); // shape 5
```

ignore

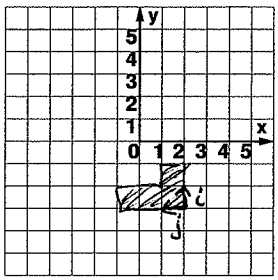
```
drawShape() {
  glBegin(GL_POLYGON);
  glVertex(0,0,0,1);
  glVertex(2,0,0,1);
  glVertex(2,1,0,1);
  glVertex(1,1,0,1);
  glVertex(1,3,0,1);
  glVertex(0,3,0,1);
  glEnd(GL_POLYGON);
}
```



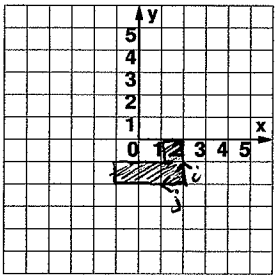
a) shape 2



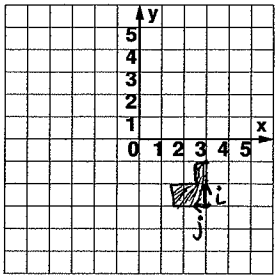
b) shape 3

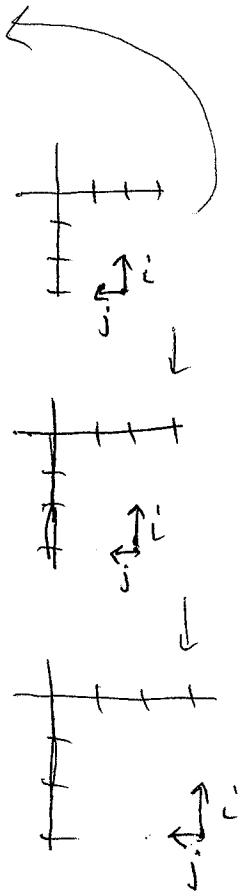


c) shape 4



d) shape 5



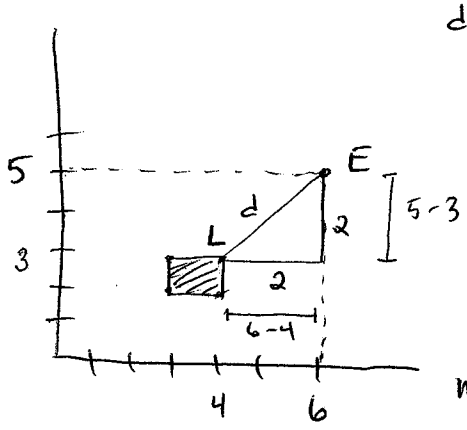


marking: 4 pts each

6. (14 pts) A square is drawn in world coordinates, with vertices $(3,2,0,1)^T$, $(4,2,0,1)^T$, $(4,3,0,1)^T$, $(3,3,0,1)^T$. (That is, vertices are column vectors.) The camera has an eye point of $(6,5,0,1)^T$, a lookat point of $(4,3,0,1)^T$, and an up vector of $(0,0,1,1)^T$. The view frustum has a near plane of 4 and a far plane of 6, with an aspect ratio of 1:1 and field of view of 90°.

a) Provide a new value for the near plane location so that the object is entirely contained within the view frustum and within one unit of the near plane.

7 pts



$d = \sqrt{2^2+2^2} = \sqrt{8} = 2\sqrt{2} < 4$

so in front of near clip plane

$2 < 2\sqrt{2} < 3$

2.7

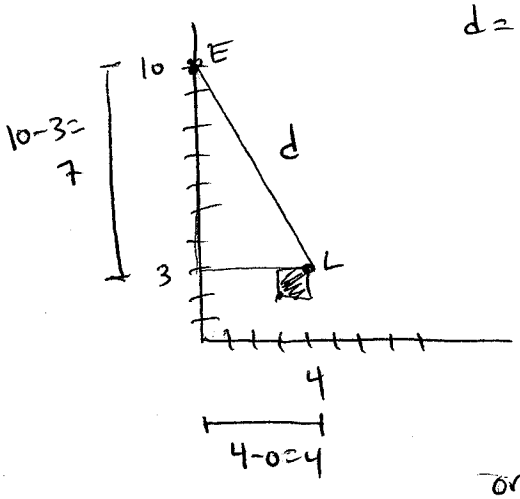
round up to 2, since round down would clip the square

marking scheme: no work shown: 5/7

drew picture or showed calculation but wrong answer: 3/7

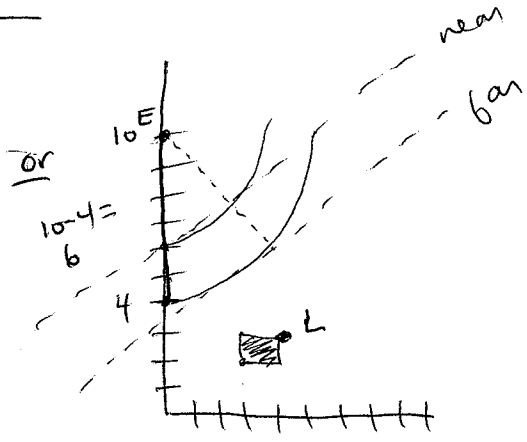
b) If the eye point is then changed to $(0,10,0,1)$, will the entire object be within the view frustum? All other parameters stay the same, including the near clipping plane value you provided above.

marking: 7 pts



$d = \sqrt{7^2+4^2} = \sqrt{49+16} = \sqrt{65} > 8$

$8 > 6$, so past far clip plane NO



7. (16 pts) Construct a matrix $M = ABC$ that transforms points from normalized device coordinates (NDCS) to display coordinates (DCS) given a viewport of width 1000 and height 1000 with the origin in the upper left. Show your intermediate work.

$M = ABC$ either trans(500, 500, .5)
scale(500, 500, .5)
reflect(1, -1, 1)

or ~~translate~~
scale(500, 500, .5)
translate(1, 1, .5)
reflect(1, -1, 1)

A

$$\begin{bmatrix} 1 & 0 & 0 & 500 \\ 0 & 1 & 0 & 500 \\ 0 & 0 & 1 & .5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B

$$\begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C

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

$$M = ABC = \begin{bmatrix} 500 & 0 & 0 & 500+.5 \\ 0 & -500 & 0 & 500-.5 \\ 0 & 0 & .5 & .5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- no penalty if z is wrong.
- pixel center translation optional.

marking scheme: correct answer w/ no work shown = 10/16

4 pts each A, B, C; 4 pts multiply to get M

1/16 if only some idea of translate, ~~s~~scale, reflect