Elements of Computer Graphics CS 174, Spring 2003

Midterm, May5, 2002

Name: ______________________________________

Student ID: _________________________________

You have to answer all problems. The marks for each question are in brackets. You may use scrap paper. However, only work in the blue book will be marked. If you run out of space then indicate it clearly on your blue book and use additional paper.
The exam is closed book. No electronic equipment is allowed on your desk including cell phones, PDAs, calculators.
For all questions explain your answers.
You have to use ink.
Indicate clearly the question and sub-question you are answering.
Hint: Read all the questions and start from the ones that seem easier for you.

Only students registered in the class may take the exam.

---

1. [10] What is an affine combination of points?

   $\sum_{i=1}^{n} a_i P_i$ **Where** $\sum_{i=1}^{n} a_i = 1$

2. [10] Write the matrix that produces the reflection of a three dimensional point through the plane $z = 0$.

```
[1 0  0 0]
[0 1  0 0]
[0 0 –1 0]
[0 0  0 1]
```

3. [10] Write the matrix that produces the reflection of a two dimensional point through an arbitrary line $y = ax+b$. Give the matrix as a product of simpler matrices.

   D = $\sqrt{1+a^2}$

```
[1 0 0] [1/D -a/D 0] [1  0 0] [ 1/D a/D 0] [1 0  0]
[0 1 b] [a/D  1/D 0] [0 -1 0] [-a/D 1/D 0] [0 1 -b]
[0 0 1] [  0    0 1] [0  0 1] [   0   0 1] [0 0  1]
```

4. [10] Decompose the following 2D affine transformation into a series of elementary ones. Show the matrices and the order of their multiplication.
x'= 3x + 1
y'= 2x + 3y + 2.

```
[1 0 1] [3 0 0] [  1 0 0]
[0 1 2] [0 3 0] [2/3 1 0]
[0 0 1] [0 0 1] [  0 0 1]
OR
[3 0 0] [  1 0 0] [1 0 1/3]
[0 3 0] [2/3 1 0] [0 1 4/9]
[0 0 1] [  0 0 1] [0 0   1]
```

5. [10] Why do we use 4x4 matrices and homogeneous coordinates?

We use homogeneous coordinates to differentiate between points and vectors. We use 4x4 matrices because it allows us to treat translations with multiplication.

6. [20] Define a sequence of transformations (scale, rotate, translate) that will map line segment L ={(x0,y0,z0), (x1,y1,z1)} to the line segment Y = {(0,0,0},(1,0,0)}.
Make sure that (x0,y0,z0) maps to (0,0,0) and (x1,y1,z1) to (1,0,0).

D = $\sqrt{dy^2 + dz^2}$

L = $\sqrt{dx^2 + dy^2 + dz^2}$

dx = x1-x0, dy = y1-y0, dz=z1-z0
scale(1/L,1,1) * rotate($\tan^{-1}$(D/dx),0,1,0) * rotate($\tan^{-1}$(dy/dz),1,0,0) * translate(-x0,-y0,-z0)

7. [10] Construct a transformation that scales around a point (x0,y0,z0). That means that the transformation maps (x0, y0, z0) to itself. Construct the matrix as a product of known matrices.

```
[1 0 0 x0] [sx  0  0 0] [1 0 0 –x0]
[0 1 0 y0] [ 0 sy  0 0] [0 1 0 –y0]
[0 0 1 z0] [ 0  0 sz 0] [0 0 1 –z0]
[0 0 0  1] [ 0  0  0 1] [0 0 0   1]
```

8. [10] Show how the contents of the modelview matrix stack change as the following program segment executes.

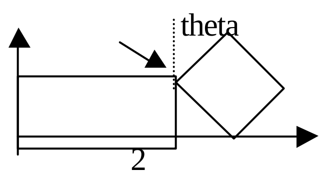
```
glLoadIdentity();
glTranslatef(1,0,5) ;
glRotatef(45, 0,0,1) ; // assume angle is given in degrees
glPushMatrix() ;
glScalef(2,1,3) ;
drawCube() ; // draws a unit cube around the origin.
glPopMatrix() ;
glTranslate(-2,1,0) ;
```

Show contents only as product of known matrices, you do not have to perform the multiplication.

```
    M1              M2               M3          M4
[1 0 0 1]  [cos 45 –sin 45 0 0]  [2 0 0 0]  [1 0 0 –2]
[0 1 0 0]  [sin 45  cos 45 0 0]  [0 1 0 0]  [0 1 0  1]
[0 0 1 5]  [     0       0 1 0]  [0 0 3 0]  [0 0 1  0]
[0 0 0 1]  [     0       0 0 1]  [0 0 0 1]  [0 0 0  1]
```

			M1*M2	M1*M2*M3	M1*M2*M3		
M(I)	M1	M1*M2	M1*M2	M1*M2	M1*M2	M1*M2	M1*M2*M4
Stack	Stack	Stack	Stack	Stack	Stack	Stack	Stack

9. [10] Assume that drawCube() draws a unit cube centered at the origin (0,0,0). Write the OpenGL part of the code that creates the following scene:



Where the shorter edges are of length 1, the longer one 2, and the cubes are centered with respect to z = 0.
You should construct your scene in such a way that you can easily rotate the cube around the corner that touches the parallepiped. That is 'theta' appears only once in your code. You may need the following functions: glTranslatef(x,y,z), glRotatef(deg,x,y,z), glScalef(x,y,z), glPushMatrix() and glPopMatrix().

glPushMatrix();
glScalef(2, 1, 1);
glTranslatef(0.5, 0.5, 0);
drawCube();
glPopMatrix();
glTranslatef(2, 1, 0);
glRotatef(-theta, 0, 0, 1);
glTranslatef(0.5, 0.5, 0);
drawCube();

10. [10] A 2D line is defined by the following points (x1,y1) (x2,y2). What is the slope of the line?

slope = (y2-y1)/(x2-x1)

11. [10] Show that the first order difference of the polynomial $y(x) = \sum_{k=0}^{n} a_k x^k$ is a polynomial of degree (n-1) assuming that a_k is not zero. Hint: compute y(x+d) as a function of y(x).

$$y(x) = \sum_{k=0}^{n} a_k x^k$$

$= a_0x^0 + a_1x^1 + ... + a_nx^n$
$= a_0 + a_1x^1 + ... + a_nx^n$

$$y(x+d) = \sum_{k=0}^{n} a_k (x+d)^k$$

$= a_0(x+d)^0 + a_1(x+d)^1 + ... + a_n(x+d)^n$
$= a_0 + a_1x + a_1d + ... + a_n(x^n + (n*d)x^{n-1} + ... + d^n)$
$= y(x) + a_1d + ... + a_n(n*d)x^{n-1} + ... + a_nd^n$
Where x^{n-1} is the dominating term

12. [10] Describe a method of filling in a simple 2D polygon.

Using scan conversion, we can start intersecting each scan line with every edge. We then sort the intersections by their x values. We then fill between each pair of intersections. Another method is selecting a pixel within the polygon and fill every pixel adjacent to the selected pixel, and then every pixel adjacent to those pixels and so on, stopping only when we hit an edge.

13. [10] Why are triangles a popular graphics primitive?

They are always simple, convex, and planar

14. [10] Assume that you have a grayscale (8bits/per pixel) image with gray ranging between 0 and 255. Explain how you can use the error diffusion algorithm to produce a bilevel image of the same dimensions.

See Section 10.9.2 of the text book.

15. [5] How do we represent a vector in homogeneous coordinates?

```
[Vx]
[Vy]
[Vz]
[ 0]
```

16. [5] How do we represent a point in homogeneous coordinates?

```
[Px]
[Py]
[Pz]
[ 1]
```

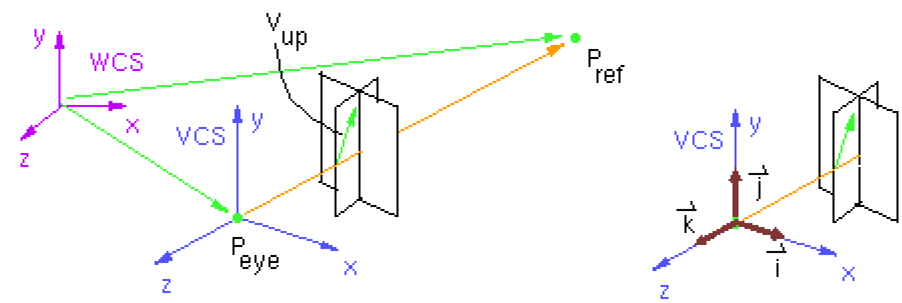
17. [5] Show an example of a linear combination of points that does not result in a valid point.

```
[Px1]    [Px2]    [Px1 - Px2]
[Py1] - [Py2] = [Py1 - Py2]
[Pz1]    [Pz2]    [Pz1 - Pz2]
[  1]    [  1]    [        0]
```

18. [5] Given two vectors a = (2,1) and b = (1,2), compute the cosine of the angle between a and b.

a . b = |a| * |b| * cos q
cos q **= (a . b) / (|a| * |b|) =**
((2,1) . (1,2)) / (|(2,1)| * |(1,2)|)
cos q **= (2 + 2) / (** $\sqrt{2^2+1^2}$ ***** $\sqrt{2^2+1^2}$ **) = 4 / 5**

19. [20] Derive the viewing transformation.



Assume that Peye, Pref and Vup are given with respect to the WCS.
Hint: $P_{VCS} = M_{WCS\text{->}VCS}\, P_{WCS} = M^{-1}{}_{VCS\text{->}WCS}\, P_{WCS}$. Find the inverse of the required transformation using a change of basis approach.

```
k = (Peye – Pref) / |Peye – Pref|
i = (Vup / |Vup|) x k
j = k x i
```

M_{cam} =

```
[1 0 0 Peye,x] [ix jx kx 0]
[0 1 0 Peye,y] [iy jy ky 0]
[0 0 1 Peye,z] [iz jz kz 0]
[0 0 0      1] [ 0  0  0 1]
```

$M_{cam\text{-}1}$ =

```
[ix iy iz 0] [1 0 0 –Peye,x]
[jx jy jz 0] [0 1 0 –Peye,y]
[kx ky kz 0] [0 0 1 –Peye,z]
[ 0  0  0 1] [0 0 0       1]
```