Name: *Key*
CS Login:

CS 559 Computer Graphics

Midterm Exam

March 22, 2010

2:30- 3:45 pm

- This exam is closed book and closed notes.
- Please write your name and CS login on every page! (we may unstaple the exams for grading) Please do this first (and check to make sure you have all of the pages)
- You will have the entire period (until 3:45pm) to complete the exam, however it was designed to take less time.
- Write numerical answers in fractional form or use radicals (square root symbols) – we would prefer to see $\sqrt{3}/2$ than .866. You should not need a calculator for this exam.
- If you need extra space, use the back of a page, but clearly mark what everything is. We may look at your work to determine partial credit.

Question #	Ave Score	Max Score
1	18.00	20
2	18.57	20
3	15.40	20
4	14.70	20
5	15.47	20
Total	82.10	100

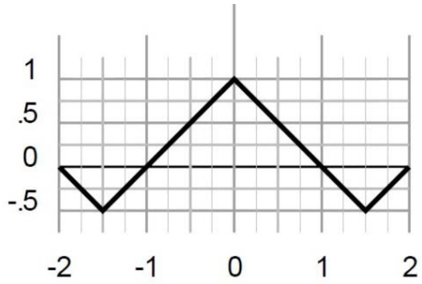
*At least one student got the max score on each question

Name: *Key*
CS Login:

1. Filtering and Convolution (20 pts)

Consider the following discrete signal f(t) = [0 8 4 2 2 8 4] defined from t=0 to t=6

a. [Discrete – Continuous Convolution]
Use the filter kernel given below for this part. The Kernel is defined over the range -2… 2. The Grid units are ¼. Assume that the sample outside the 0-6 range is 0 for this part.



What is the value of the convolved signal at

t= 1.5 *-.5*0 + .5*8 + .5*4 + -.5*2 = 5*

t= 2 *4 (interpolating)*

t= 3.5 *-.5*4 + .5*2 + .5*2 + -.5*8 = -4*

t= 4 *2 (interpolating)*

b. [Discrete – Discrete Convolution]
Use the filter kernel [-1 1 1] defined over the range -1… 1. Assume that the sample outside the 0-6 range is 0 for this part. What is the value of the convolved signal at
You need to flip the Kernel

t= 1 *1*0 + 1*8 + -1*4 = 4*

t= 3 *1*4+ 1*2 + -1*2 = 4*

t= 5 *1*2 + 1*8 + -1*4 = 6*

if kernel wasn't flipped (-1 point each) 12, 0, 10

c. [Discrete – Discrete Convolution at boundries]
Use the filter g(t) =[¼ ½ ¼] defined over the range -1… 1. Use kernel re-normalization at the boundaries. What is the value of the convolved signal at

t= 0 *⅔*0 + ⅓*8 = 8/3*

t= 6 *⅓*8 + ⅔*4 = 16/3*

Name: *Key*
CS Login:

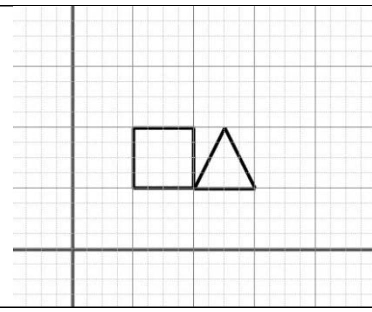
2. Transformations (20 pts)

a. Consider a simple graphics toolkit that works like OpenGL (that is, it has a matrix stack, and the transformation commands post-multiply themselves onto it):

The toolkit has the following commands

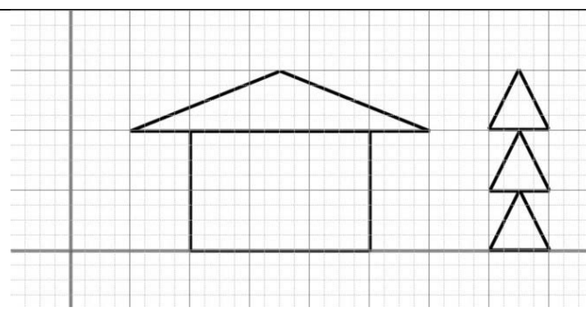
translate(x,y) – post-multiplies a translation matrix onto the top of the matrix stack
rotate(a) – rotates (counter clockwise around the origin) by a degrees
scale(x,y) – scales by x and y. BOTH X and Y MUST BE POSITIVE
push() – pushes a copy of the top element on the matrix stack
pop() – removes the top element from the matrix stack
draw(triangle) – draws a triangle with unit base and unit height
draw(square) – draws a unit square

<u>Sample</u>
translate(1,1)
draw(square)
translate(1,0)
draw(triangle)



Write down the sequence of commands to make the following drawing in a minimum number of steps

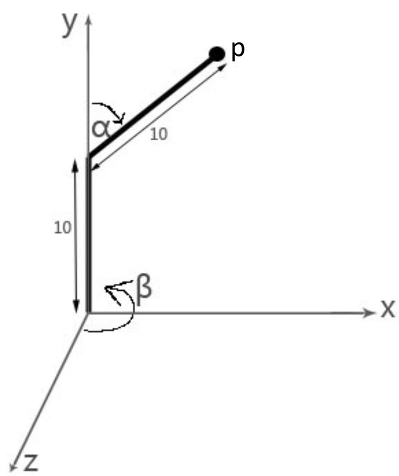
1. *translate(2,0)*
2. *scale(3,2)*
3. *draw(square)*
4. *scale(⅓,½)*
5. *translate(5,0)*
6. *draw(triangle)*
7. *translate(0,1)*
8. *draw(triangle)*
9. *translate(0,1)*
10. *draw(triangle)*
11. *translate(-6,0)*
12. *scale(5,1)*
13. *draw(triangle)*



-1 points for each step after 14 steps

b. Consider the robotic arm given in the diagram. It has two parts, both 10 units in length. The bottom part can rotate by β^o around the y axis (counter clockwise) and the top part can move α^o away from the y axis. We are interested in the final position of the arm, marked by •

The robotic arm takes the following commands
reset () - brings the arm initial position where $\alpha = 0$, $\beta=0$ and p = (0,20,0)
move(α, β) - moves the arm down from y-axis α^o, and rotates around y-axis β^o



Give the position of p after the given set of commands

Example:
reset()
move(90, 0) p = *(10, 10, 0)*

i reset()
move(90, 90) p = *(0,10,-10)*

ii reset()
move(90,45) p = *(5√2, 0, -5√2)*

iii reset()
move(60,30)
move(-15,60) p = *(0, 10+5√2, -5√2)*

Name: *Key*
CS Login:

3. Projection (20 pts)

a. A vertex (point) is drawn at the origin. It is viewed through a camera that is positioned with the viewing matrix:

Step 2 $$\begin{bmatrix}2 & -1 & 0 & -2\\ 1 & 2 & 0 & -2\\ 0 & 0 & 1 & 3\\ 0 & 0 & 0 & 1\end{bmatrix} * \begin{bmatrix}5\\3\\2\\1\end{bmatrix} = \begin{bmatrix}5\\9\\5\\1\end{bmatrix}$$

The object that the vertex is drawn with transformation matrix:

Step 1 $$\begin{bmatrix}0 & -1 & 0 & 5\\ 1 & 0 & 0 & 3\\ 0 & 0 & 1 & 2\\ 0 & 0 & 0 & 1\end{bmatrix} * \begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{bmatrix}5\\3\\2\\1\end{bmatrix}$$

This simple projective transform matrix is used:

Step 3 $$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -1 & 0\end{bmatrix} * \begin{bmatrix}5\\9\\5\\1\end{bmatrix} = \begin{bmatrix}5\\9\\1\\-5\end{bmatrix}$$

Where does the point appear in screen coordinates? (give the x,y position)
(-1, -9/5)

b. For each of the properties, mark
P if it is true of a perspective projection,
O if it is true of an orthographic projection,
B if it is true of both orthographic and projective, and
N if it is true of neither. (use P or O only if the property is only true of one type of transformation).

_*B*_ Straight lines are mapped to straight lines

_*N*_ Distances are preserved

_*N*_ Angles are preserved

_*O*_ Far away objects appear the same size as closer ones

Name: *Key*
CS Login:

4. Curves (20 Pts)

a. Consider a Cubic Bezier Curve defined by the following 4 control points (0,0), (0,4), (4,4), (4,0). Compute the location of the curve at parameter u = 0.5 using the Decastijau's algorithm

(0,0) (0,4) (4,4) (4,0)
(0,2) (2,4) (4,2)
(1,3) (3,3)
(2,3)

b. Consider a Catmull-Rom spline defined by the following 4 control points (1,2), (4,8), (5,8), (6,1). Convert this spline in to a Hermite Cubic segment, List the controls of the Hermite segment. You only need to list the controls, it is not necessary to give cubic eqation

Catmull-Rom control points

P_0: (1,2)

P_1: (4,8)

P_2: (5,8)

P_3: (6,1)

Hermite Cubic segment controls

$f(0) = P_1 = (4,8)$

$f'(0) = \frac{1}{2}(P_2 - P_0) = (2,3)$

$f(1) = P_2 = (5,8)$

$f'(1) = \frac{1}{2}(P_3 - P_1) = (1,-3\frac{1}{2})$

Name: *Key*
CS Login:

5. Shading (20 pts)

For this question observe the 4 rows displayed on the screen. In each row, there is one parameter in the phong lighting model changing. determine which parameter is the most likely one that is changing.

The Phong lighting model equations is

$$I = \max\{L.n, 0\}\, L_d \,.\, K_d + \max\{s.n, 0\}^{shininess}\, L_s \,.\, K_s$$

where:
K_d is the diffuse reflection coefficient
L_d is the intensity of the light source (remains fixed for the 4 rows)
n is the normal to the surface (unit vector)
L is the direction to the light source (unit vector)
K_s is the specular reflection coefficient
L_s is the intensity of the point light source (remains fixed for the 4 rows)

row 1: _______________*Shininess*__________________________ parameter is changing

row 2: ____________________ K_d ___________________________ parameter is changing

row 3: ____________________ K_s ___________________________ parameter is changing

row 4: ____________________L____________________________ parameter is changing