

# CMSC 435/634: Introduction to Graphics

## Midterm Exam

October 9, 2002

**Instructions:** Clearly write your name on this sheet. Answer each problem in the space provided. If you need extra space, write “on extra sheet” in the answer space on the exam paper, clearly label the problem number on an extra sheet of paper, and staple the extra sheet to the exam when you hand it to me.

You may a calculator to assist you in simple arithmetic. You may also use any notes you have written on one side of a single 8.5 x 11 inch sheet of paper. No other notes or books are allowed.

The exam consists of four parts and several sub-parts. Each is marked with a number of points. There are 75 points total. You have 75 minutes.

**Strategy:** Read all parts of each problem before you start answering it. The work for some problems is split across several questions. You may find that later questions will guide the method you use to solve the problem. You may even find that later questions will jog your memory.

If something isn't clear, ask.

Good luck

**Name:** \_\_\_\_\_

**1 Odds and ends**

**(11 points total)**

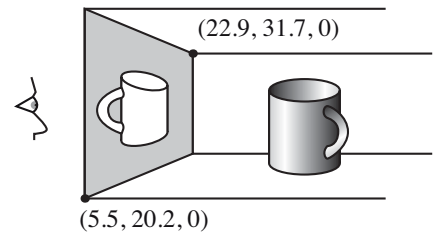
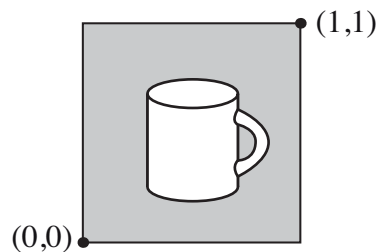
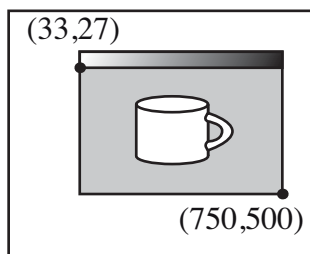
1.1 If there were a country song about your experience with assignment 2, what would the title be? **(1 point)**

1.2 Name two things Ivan Sutherland did. **(2 points)**

1.3 How many control points for a quartic ( $t^4$ ) Bezier curve? **(1 point)**

1.4 What is a scan line? **(2 points)**

1.5 What are these three coordinate systems called (as indicated by the gray boxes)? **(3 points)**



1.6 What is gamma correction and why do we need it? **(2 points)**

## 2 Transformations

(14 points total)

2.1 The OpenGL command for scaling is `glScalef(x,y,z)`

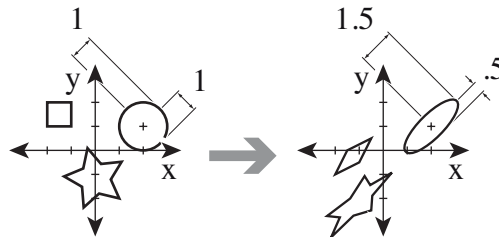
(6 points)

2.1.1 What is the 4x4 matrix form for `glScalef(2,2,2)`? Its inverse?

2.1.2 Do objects scaled by this get bigger or smaller?

2.2 I want perform the illustrated non-uniform scaling oriented at  $45^\circ$  to the coordinate axes and centered around the point (2,1):

(8 points)



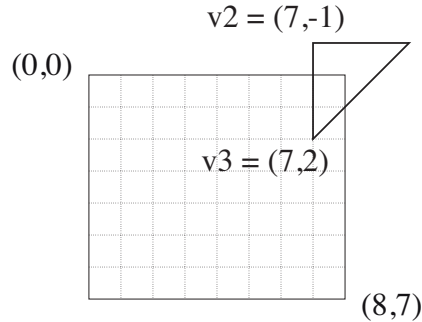
2.2.1 The basic OpenGL transformation functions are `glScalef(x,y,z)`, `glTranslatef(x,y,z)` and `glRotatef(angle,x,y,z)`. The rotation angle is given in degrees, and follows the right hand rule (so if the vector  $(x,y,z)$  points toward the viewer, the rotation will be counterclockwise). How would you decompose this transformation into a sequence of OpenGL calls? Since this is just a 2D transformation, your transformation should leave  $z$  alone. Please fill in the calls with real numbers.

2.2.2 Draw a series of sketches to show how each stage of the decomposition affects the three objects. Your sketches don't have to be exact, but please do show the coordinate axes in each.

### 3 Sutherland-Hodgeman Clipping

(25 points total)

This problem uses the following scenario of (small) 8x7 window and single triangle. For this problem, you will be working through the steps to clip the polygon against the screen. For ease of exam-time computation, I've set it up so the numbers will come out well. You must **show your work** including any equations you use to find the clip intersections for full credit.



The Sutherland-Hodgeman algorithm clips against each edge of the window in turn. Here is pseudo-code for clipping against one edge:

```
for each vertex input to this stage
  if (condition1)
    output vertex to next stage
  if (condition2)
    compute intersection with window edge
    output intersection vertex to next stage
```

#### 3.1 Complete the algorithm

(10 points)

- 3.1.1 What 'if' condition would you use to decide whether to output each vertex to the next stage (condition1)? Use either code or equations and describe what it means.
  
- 3.1.2 What 'if' condition would you use to decide whether to output a new intersection vertex to the next stage (condition2)? Use either code or equations and also describe what it means.
  
- 3.1.3 If you need to compute an intersection point, how would you do it? Use either code or equation form in terms of vertex and window variables.

3.2 Top Edge

**(6 points)**

3.2.1 A) Is v1 output to the next stage (condition1)? B) Is an intersection point between v1 and v2 output (condition 2)? C) If so, where (show the equation!)

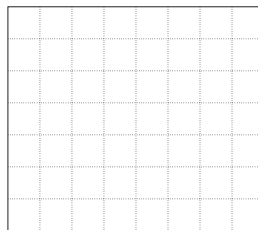
3.2.2 Repeat A), B) and C) of 3.2.1 for v2

3.2.3 Repeat A), B) and C) of 3.2.1 for v3

3.2.4 List the coordinates of all points output from the top edge clipping stage, in order.

3.2.5 Draw what the triangle looks like at this point.

(0,0)



(8,7)

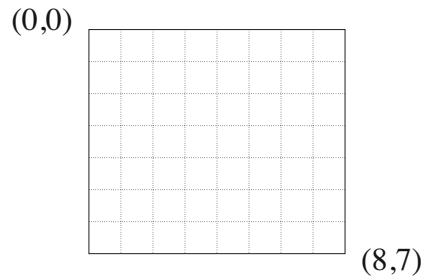
3.3 Bottom Edge

**(3 points)**

3.3.1 Repeat 3.2.1 for each point listed in 3.2.4.

3.3.2 List the coordinates of all points output from the bottom edge clipping stage, in order.

3.3.3 Draw what the triangle looks like at this point.



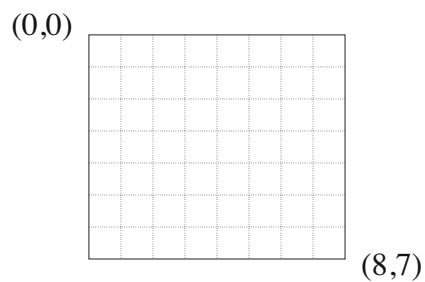
3.4 Left Edge

**(3 points)**

3.4.1 Repeat 3.2.1 for each point listed in 3.3.2.

3.4.2 List the coordinates of all points output from the left edge clipping stage, in order.

3.4.3 Draw what the triangle looks like at this point.



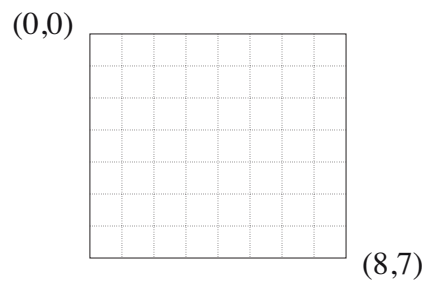
3.5 Right Edge

(3 points)

3.5.1 Repeat 3.2.1 for each point listed in 3.4.2.

3.5.2 List the coordinates of all points output from the right edge clipping stage, in order.

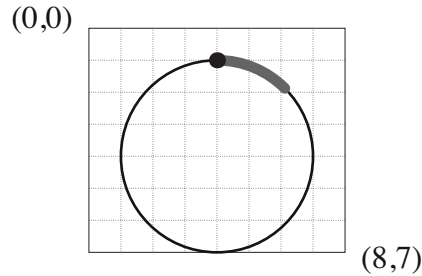
3.5.3 Draw what the triangle looks like at this point.



#### 4 Midpoint Circle Algorithm

(25 points total)

This final problem uses a circle of radius  $r = 3$ , centered at  $(x_c, y_c) = (4, 4)$  in the same small  $8 \times 7$  window:



4.1 What is the implicit function for this circle,  $F(x,y)=0$ ? (1 point)

4.2 We will step using the midpoint algorithm to step along marked octant, starting at  $(x_c, y_c - r) = (4, 1)$ , deciding at each step from  $x$  to  $x+1$  whether to plot pixel at  $(x+1, y)$  or  $(x+1, y+1)$ . What condition in the form  $d > 0$  would you use to decide to plot  $(x+1, y+1)$  instead of  $(x+1, y)$ ? (3 points)

4.3 Since we want this to be fast, we'll use DDA to incrementally update the decision variable,  $d$ . (12 points)

4.3.1 What is the increment,  $\Delta_x d$ , to find  $d$  at  $(x+1, y)$  from  $d$  at  $(x, y)$ ? What is the increment,  $\Delta_{xy} d$ , to find  $d$  at  $(x+1, y+1)$  from  $d$  at  $(x, y)$ ?

4.3.2  $\Delta_x d$  and  $\Delta_{xy} d$  are not constant. What are the increments,  $\Delta_x \Delta_x d$  and  $\Delta_{xy} \Delta_x d$  to find  $\Delta_x d$  at  $(x+1, y)$  and  $(x+1, y+1)$  from  $\Delta_x d$  at  $(x, y)$ ? What are the increments,  $\Delta_x \Delta_{xy} d$  and  $\Delta_{xy} \Delta_{xy} d$  to find  $\Delta_{xy} d$  at  $(x+1, y)$  and  $(x+1, y+1)$  from  $\Delta_{xy} d$  at  $(x, y)$ ?



4.4 Evaluating DDA

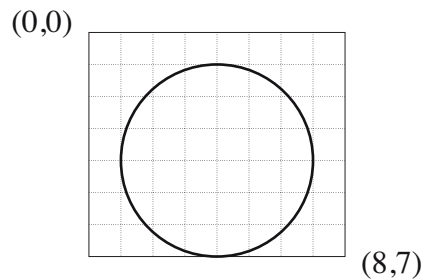
(10 points)

4.4.1 What are the values of  $d$ ,  $\Delta_x d$  and  $\Delta_{xy} d$  at  $x=4$  and  $y=0.5$ ?

4.4.2 What DDA additions would you perform to step to  $x=5$ ? What are the values of  $d$ ,  $\Delta_x d$  and  $\Delta_{xy} d$  at  $x=5$ ? What point to you plot at  $x=5$ ?

4.4.3 What DDA additions would you perform to step to  $x=6$ ? What are the values of  $d$ ,  $\Delta_x d$  and  $\Delta_{xy} d$  at  $x=6$ ? What point to you plot at  $x=5$ ?

4.4.4 Mark each of these three points with an X, and their reflections to the other eight octants with an O:



**Do not open the test until told to begin!**

## **CMSC 435/634: Introduction to Graphics**

### **Midterm Exam**

**March 17, 2004**

**Instructions:** Clearly write your name on this sheet. Answer each problem in the space provided. If you need extra space, write “on extra sheet” in the answer space on the exam paper, clearly label the problem number on an extra sheet of paper, and staple the extra sheet to the exam when you hand it to me.

You may a calculator to assist you in simple arithmetic. You may also use any notes you have written on one side of a single 8.5 x 11 inch sheet of paper. No other notes or books are allowed.

The exam consists of four parts and several sub-parts. Each is marked with a number of points. There are 75 points total. You have 75 minutes.

**Strategy:** Read all parts of each problem before you start answering it. The work for some problems is split across several questions. You may find that later questions will guide the method you use to solve the problem. You may even find that later questions will jog your memory.

If something isn't clear, ask.

Good luck

**Name:** \_\_\_\_\_

**Do not open the test until told to begin!**

**1 Odds and ends**

**(10 points total)**

1.1 If there were a comic book written about your experience with assignment 2, what would the title be? **(1 point)**

1.2 Why do we use RGB color in graphics? What do R, G and B stand for? **(4 points)**

1.3 What degree polynomial do you need to pass through six points with a single interpolating curve? **(2 points)**




1.3.1 What is a pixel? **(2 points)**

1.4 Name one display device that is **not** raster-based. **(1 point)**

## 2 Modeling

(15 points total)

You are given an L-system grammar with the following symbols:

Symbol	Meaning	Example
S,T	Unit-length segments	S
+	Bend 45 degrees to the left	S+S 
-	Bend 45 degrees to the right	S-S 
[X]	String X branches from main string	S[-S]S 

2.1 Draw the next three stages of evolution of these rules, starting with a single T (5 points)

$$T = S[+T][-T]ST$$

$$S=SS$$

2.2 Draw the next three stages of evolution of these rules, starting with a single T (5 points)

$$T = S[+T]S[-T]ST$$

$$S=SS$$

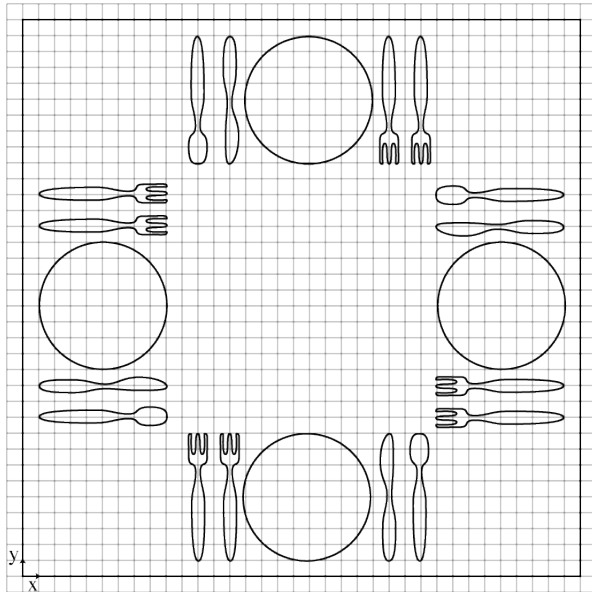
2.3 Draw the next three stages of evolution of these rules, starting with a single T (5 points)

$$T = S[-S]+S[-T]+S$$

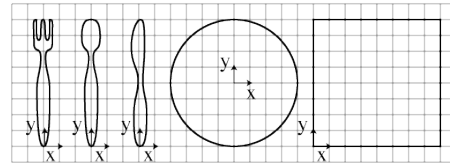
### 3 Transformations

(20 points total)

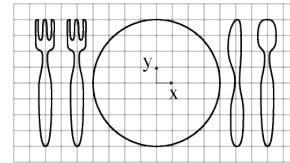
For this problem, you will be drawing a table set for four as shown in (a) below. You have already written the drawing functions `fork()`, `knife()`, `spoon()`, `plate()` and `table()`, which draw these objects with the origin, orientation and scale shown in (b).



(a)



(b)



(c)

- 3.1 Write the code for a `placeSetting()` function that produces a single place setting as shown in (c). You may use the drawing functions mentioned above as well as the graphics functions `Scale(x,y,z)`, `RotateXY( $\theta$ )`, `Translate(x,y,z)`, `PushMatrix()`, and `PopMatrix()`. `RotateXY` rotates around the Z axis: equivalent to the axis/angle rotation `Rotate( $\theta$ , 0,0,1)`

(7 points)

3.2 Now write the code to lay out all four place settings as shown in (a), using any of the functions mentioned above as well as the new `placeSetting()` function. **(7 points)**

3.3 Write out each of the following transformation matrices **(6 points)**

3.3.1 `Scale(x,y,z)`

3.3.2 `RotateXY( $\theta$ )`

3.3.3 `Translate(x,y,z)`

## 4 Rasterization

**(30 points total)**

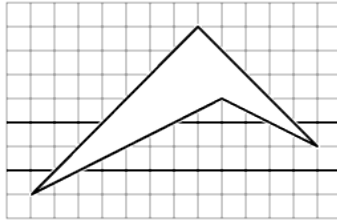
The first part of this problem deals with the sorted-edge-table scan-line polygon rasterization algorithm covered in the notes and in class.

- 4.1 Write pseudo-code for this algorithm. Clearly label the setup, per-scanline, and per-pixel phases of your pseudo-code. **(10 points)**

- 4.2 What data-structures do you require? **(5 points)**

4.3 Given the following polygon

**(10 points)**



4.3.1 What are the values in your data structures after setup?

4.3.2 What are the values at the beginning of the second scanline (the lower bold line in the figure)?

4.3.3 What are the values at the beginning of the fourth scanline (the upper bold line in the figure)?

4.4 If you have a scan-line renderer using this algorithm and must add visibility, which method would you use? Why? What are the potential drawbacks to your choice?

**(5 points)**



**Do not open the test until told to begin!**

## **CMSC 435/634: Introduction to Graphics**

### **Midterm Exam**

**October 21, 2004**

**Instructions:** Clearly write your name on this sheet. Answer each problem in the space provided. If you need extra space, write “on extra sheet” in the answer space on the exam paper, clearly label the problem number on an extra sheet of paper, and staple the extra sheet to the exam when you hand it to me.

You may a calculator to assist you in simple arithmetic. You may also use any notes you have written on one side of a single 8.5 x 11 inch sheet of paper. No other notes or books are allowed.

The exam consists of four parts and several sub-parts. Each is marked with a number of points. There are 75 points total. You have 75 minutes.

**Strategy:** Read all parts of each problem before you start answering it. The work for some problems is split across several questions. You may find that later questions will guide the method you use to solve the problem. You may even find that later questions will jog your memory.

If something isn't clear, ask.

Good luck

**Name:** \_\_\_\_\_

**Do not open the test until told to begin!**

**1 Odds and ends**

**(10 points total)**

1.1 If your experience with assignment 3 were made into a Broadway musical, what would the show be called? **(1 point)**

1.2 What is Non-Photorealistic Rendering? **(2 points)**

1.3 Interactive graphics requires approximately what range of frame rates? **(2 points)**

1.4 Why would you ever want to use a rendering technique that is not interactive? **(3 points)**

1.5 How long is the cross product of two vectors,  $\vec{a}$  and  $\vec{b}$ . **(2 point)**

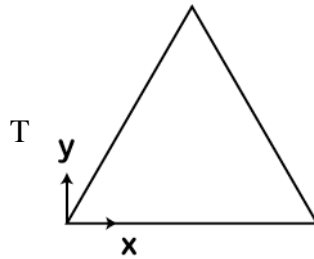
## 2 Modeling

(30 points total)

Consider an L-system with terminal symbols interpreted to perform the following transformations:

[     PushTransform()  
]     PopTransform()  
 $r(\theta)$  Rotate( $\theta$ , 0,0,1)  
 $s(\alpha)$  Scale( $\alpha$ , $\alpha$ , $\alpha$ )  
 $t(x)$  Translate( $x$ ,0,0)

and one non-terminal, interpreted to draw a triangle with the origin shown (z pointing out of the page):



2.1 Given this rule:

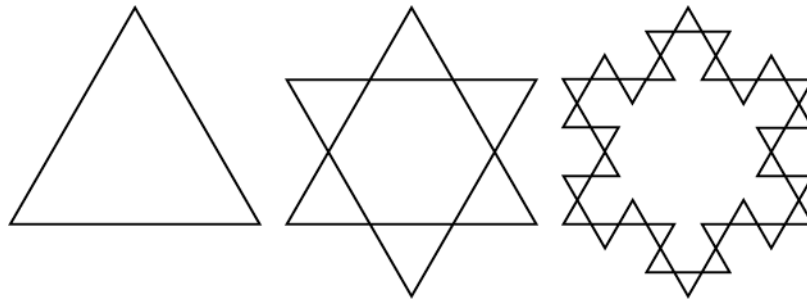
(10 points)

$T \rightarrow s(.5)T[r(60)t(1)r(-60)T]t(1)T$

2.1.1 What is the symbolic form for the next two generations after the start symbol T

2.1.2 Draw the graphical representation of the same two generations.

- 2.2 Propose a rule or rules to generate this snowflake from start symbol T (T and the following two generations are shown) **(10 points)**



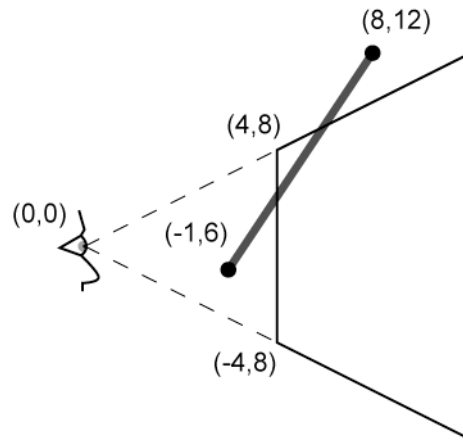
- 2.3 Show that a symbol for  $\text{Translate}(x,y,0)$  is not necessary by constructing  $\text{Translate}(x,y,0)$  using any of  $r(\theta)$ ,  $s(\alpha)$ ,  $t(x)$ , and any standard math library functions. **(5 points)**

- 2.4 Show that the parameter in  $t(x)$  is not necessary by constructing  $t(x)$  using any of  $r(\theta)$ ,  $s(\alpha)$ ,  $t(1)$  and any standard math library functions. **(5 points)**

### 3 Cohen-Sutherland Clipping

(35 points total)

For this problem, you will be clipping the line shown against the top, bottom and near planes of the view frustum:



#### 3.1 Edge testing

(8 points)

3.1.1 What is the general formula for determining if a point is inside or outside one of the clip planes?

3.1.2 Compute this test for each of the endpoints and each of the clip planes (you should have six answers).

3.1.3 Show the three-bit outcode for each endpoint (in near/top/bottom order)

3.2 Trivial accept and reject

**(4 points)**

3.2.1 Show the pseudo-code and result for the trivial reject test. Is this line trivially rejected?

3.2.2 Show the pseudo-code and result for the trivial accept test. Is this line trivially accepted?

3.3 Intersection computation

**(8 points)**

3.3.1 What is the general formula for computing the intersection point of the line with a clip plane?

3.3.2 Compute the intersection point with the near plane.

3.3.3 Compute the intersection point with the top plane.

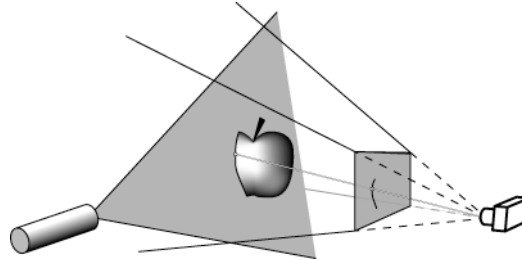
3.4 Walk through the Cohen-Sutherland clipping algorithm for this line. You may use text, equations, pseudo-code, computed values and/or pictures to illustrate the algorithm's progression, but please be complete and descriptive enough to show you know how the algorithm works. For results computed earlier in this part, you may refer back to the previous computation (e.g. "find intersection with top plane, *blah*, from 3.3.3")

**(15 points)**

#### 4 Extra Credit

(15 points total)

In a laser scanner, a laser is used to create a plane of light. The plane of light shines on the object to be scanned in a line. This line of laser light is seen by a camera. If you know the configuration of the laser and camera, you can triangulate the position of points on the object.



Consider three spaces:

1. laser: laser at origin; light projected along the  $y/z$  plane
2. camera: camera at origin; projected point on image at  $P_{IM}$
3. world: 3D space containing model

And two transforms

1.  $M_L$ : transforms points in laser space to world space
2.  $M_C$ : transforms points in camera space to world space

Write formula(s) to compute  $P$  in world space in terms of  $P_{IM}$ ,  $M_L$ , and  $M_C$ .

(Hint: try actually solving in laser space)