CPSC 314, Final Exam

17 Apr 2007, 8:30am-11:00am

Closed book, one double-sided sheet of handwritten notes allowed. Nonprogrammable calculators allowed. No other electronic devices (programmable calculators, phones, PDAs, etc) allowed, cell phones and pagers must be turned off. Answer the questions in the space provided. If you run out of room for an answer, continue on the back. Place your photo ID face up on your desk. Do not open the packet until told to do so.
You may keep your notes sheet, no need to turn it in with the exam.

Rules Governing Formal Examinations
The following are the rules governing formal examinations:

- Each candidate must be prepared to produce, upon request, a Library/AMS card for identification.
- Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
- No candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination.
- Candidates suspected of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  - Having at the place of writing any books, papers or memoranda, calculators, computers, audio or video cassette players or other memory aid devices, other than those authorized by the examiners.
  - Speaking or communicating with other candidates.
  - Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

I understand and accept the above rules:

Name: Solutions

Signature:

Student Number:

Page	Points Earned	Points Possible
2		16
3		12
4		19
5		9
6		16
7		12
8		15
9		14
10		9
11		8
12		30
Total		160

1. (12 pts) True/false: mark the following statements as true or false

F a• The homogeneous points (3,4,7,6) and (3,4,8,6) map to the same Cartesian point.

F b• In a blueprint made with a cavalier projection, lines converge to two vanishing points.

T c• In a blueprint made with an isometric projection, lengths can be usefully measured directly because they are are all drawn to the same scale.

F d• High pass filtering causes blur.

T e• Perlin noise has structure at multiple scales.

T f• Taking subsurface scattering into account when rendering makes human skin look more realistic.

F g• Image-based rendering uses volumetric grids.

T h• A ray tracer for an image of 16 by 16 pixels will require 256 primary rays.

T i• The location of vertices can change as the result of vertex shader operations.

T j• A pixel shader could support Phong shading.

F k• The OpenGL select buffer contains the objects drawn in the small viewport around the cursor location, sorted from back to front in depth.

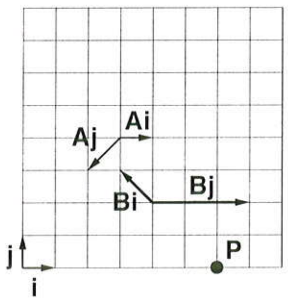
T l• Supersampling is a good antialiasing strategy to reduce the jagginess of an image.

1 pt each, all or nothing

2. (4 pts) Specify the coordinates of point P with respect to coordinate frames A and B.

2 pts each for A, B

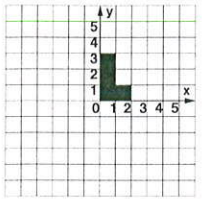
(1 pt per correct number)



(7, 4) P_A

(-2, 0) P_B

3. (12 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)`.

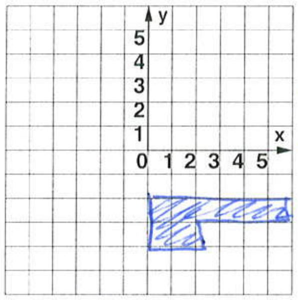


```
drawL();
```

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

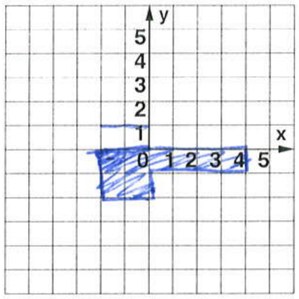
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\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

a) L' = ABC L



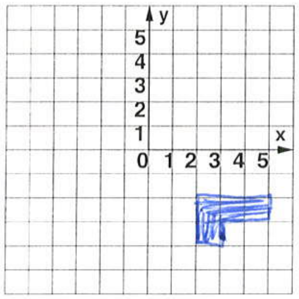
glRotate(-90,0,0,1)
glTranslate(2,0,0)
glScale(1,2,1)
draw L

b) L' = ACD L



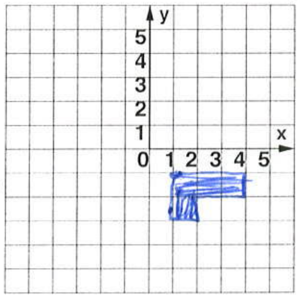
rot (-90)
scale (1,2,1)
trans(0,-1,0)
drawL

c) L' = BAB L



trans (2,0,0)
rot(-90,0,0,1)
trans(2,0,0)
draw L

d) L' = DBAD L



trans(0,-1,0)
trans(2,0,0)
rot(-90)
trans(0,-1,0)
drawL

all or nothing: 3 pts pic, 1 pt openGL × 4
(except -1 if rotate in wrong direction)

4. (1 pt) If a light of color (0.2, 1.0, 0.8) shines on a surface with ambient color (0.5, 0.5, 0.5), what is the resulting color triplet?

$(.2 \cdot .5, 1.0 \cdot .5, .8 \cdot .5) \rightarrow (.1, .5, .4)$ 1 pt all or nothing

5. (3 pts) Compute the normal for the triangle (2, 4, 1, 1), (0, 5, -2, 1), (-1, 2, 3, 1).

$\vec{a} = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ -3 \end{bmatrix}$ $a \times b = \begin{bmatrix} -4 \\ 13 \\ 7 \end{bmatrix}$ normal, unnormalized $\|a \times b\| = \sqrt{16+169+49} = \sqrt{234} = 15.3$

$\frac{a \times b}{\|a \times b\|} = (-.26, .85, .46)$

$\vec{b} = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 2 \end{bmatrix}$

2pts for normal calc, 1 pt for normalizing it

6. (15 pts) The color L_c of light L is (.9,.9,.9). Its location is (0,3,0). The eye point E is at (3,1,0). The specular coefficient k_s is (.8,.8,.1),and n^{shiny} is 5. Compute the specular color of the surface at a point P = (3,3,0) with the normal (-.707, .707, 0), showing your work. Use the Blinn-Phong model:

$\left(\hat{n} = \frac{n}{\|n\|} = \frac{(-.707, .707, 0)}{.9999} \checkmark \text{already normalized}\right)$

$$\mathbf{I_{spec}} = \mathbf{L_c}(\mathbf{k_s}(\mathbf{h} \cdot \mathbf{n})^{shiny}))$$

2pts calc, 1 pt normalize × 3 → 9 pts

$\hat{l} = \frac{L-P}{\|L-P\|} = \frac{(0,3,0)-(3,3,0)}{\|L-P\|} = \frac{(-3,0,0)}{\sqrt{3^2}} = (-1,0,0)$

$\hat{v} = \frac{E-P}{\|E-P\|} = \frac{(3,1,0)-(3,3,0)}{\|E-P\|} = \frac{(0,-2,0)}{\sqrt{2^2}} = (0,-1,0)$

$h = \frac{\hat{l}+\hat{v}}{2} = \frac{(-1,0,0)+(0,-1,0)}{2} = \frac{(-1,-1,0)}{2} = \left(\frac{-1}{2}, \frac{-1}{2}, 0\right)$

$\hat{h} = \frac{h}{\|h\|} = \frac{(-1/2, -1/2, 0)}{\sqrt{1/2^2 + 1/2^2}} = \frac{-1/2, -1/2, 0}{\sqrt{1/2}} = \left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0\right) = (-.707, -.707, 0)$

$(.9, .9, .9) \cdot (.8, .8, .1)((-.707, .707, 0) \cdot (-.707, +.707, 0))^5$

$(.72, .72, .09)\ (.499 + -.499 + 0)^5 \rightarrow (.72, .72, .09)(0) \rightarrow 0$

Ispec

1 component-wise multiply, 1 dot, 1 exp, 3 math calc → 6 pts

7. (9 pts) A cube has 6 faces and 8 vertices. The complete cost of computing Phong lighting at a point is L. The cost of barycentric interpolation is I. Assume that all vertices and faces will be rendered, and that there are P pixels in total that need shading. Each quadrilateral face is processed individually during scan conversion. The model has precomputed normals at the vertices of the model which are already normalized.

The cost of a component-wise multiplication is C, the cost of a dot product is D, and the cost of exponentiation is E. The cost of normalizing a vector is N. The cost of adding or subtracting vectors, or multiplying a vector by a scalar, is A. The cost of a cross product is X. The cost of scalar-scalar multiplication is S.

a) Give the cost of flat shading the entire model in terms of the quantities above.

6 faces * 1 L each = 6L   only one lighting calculation per face

2 pts: 1 pt L, 1 pt coefficient

b) Give the cost of Gouraud shading the entire model in terms of the quantities above.

note: no storage of lighting calculation between faces - each rendered separately! so must recompute L for each vertex in each face

$6 \text{ faces} * 4 \text{ verts each} * L + \underbrace{PI}_{\text{interpolate color at each pixel}}$

24L + PI

3 pts: 1 pt L, 1 pt P, 1 pt I

c) Give the cost of Phong shading the entire model in terms of the quantities above.

P(I + N + L)   at each pixel, do lighting for interpolated normal

4 pts: 1 pt each P, I, N, L

8. (16 pts) Clip the line going through the points A=(-1,-2) and B=(5,4) to the viewport with corners (0,0) and (5,3) using the Cohen-Sutherland clipping algorithm. The order in which you should encode the outcodes is top, bottom, left, right:

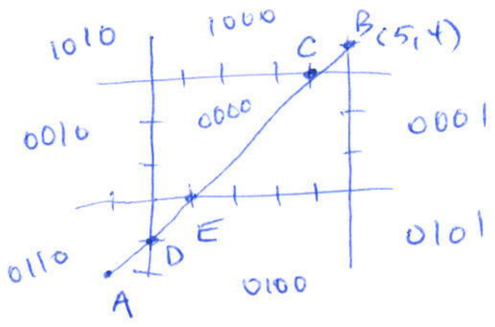
$$\begin{pmatrix} 1010 & 1000 & 1001 \\ 0010 & 0000 & 0001 \\ 0110 & 0100 & 0101 \end{pmatrix}$$

a) Show the full configuration after clipping against the top of the box with a sketch, including any new points you may need to create. If you need to create new points, give them names in alphabetical order (C, D, E, ...). Compute the exact coordinates of any new points that result from intersecting lines. (Provide the final values as decimal numbers if you have a calculator.) Show your intermediate work, including outcodes for vertices.

b) Same as above, after clipping against the left edge.

c) Same as above, after clipping against the right edge.

d) Same as above, after clipping against the bottom edge of the box.



(full sketch FYI)

trivial accept/reject:

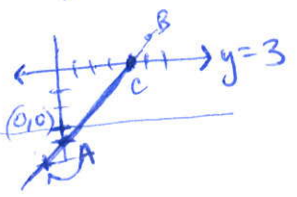
accept: oc(p1)==0 && oc(p2)==0

reject: (oc(p1) & oc(p2)) != 0

$y = mx+b \quad m = 6/6 = 1$

$4 = 6/6(5) + b \rightarrow b = -1$

a) $A \rightarrow 0110$, $B \rightarrow 1000$ } no trivial accept/reject vs $y=3$

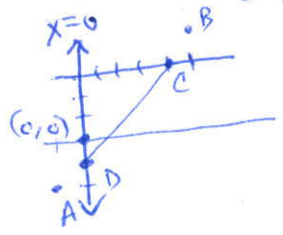


intersection

$$\boxed{\begin{aligned} x &= x_A + (y_{top} - y_A)m \\ &= -1 + (3+2)/1 \\ &= -1+5 = 4 \end{aligned}}$$

$C = [4,3] \rightarrow 0000$

b) $A \rightarrow 0110$, $C \rightarrow 0000$ } no trivial acc/rej vs $x=0$



intersection

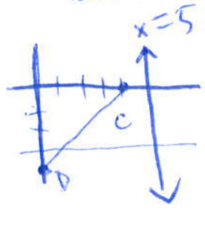
$D = [0,-1] \rightarrow 0100$

$$\boxed{\begin{aligned} y &= y_A + (x_{left} - x_A)m \\ &= -2 + 1(0+1) \\ &= -2+1 \\ &= -1 \end{aligned}}$$

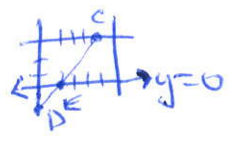
c) $D \rightarrow 0100$, $C \rightarrow 0000$ } no trivial acc/rej vs $x=5$

no change to line

~~[illegible]~~



d) $D \rightarrow 0100$, $C \rightarrow 0000$ } no trivial acc/rej vs $y=0$



intersection

$E = [1,0]$

$$\boxed{\begin{aligned} x &= x_D + (y_{bot} - y_D)/m \\ &= 0 + (0+1)/1 \\ &= 1 \end{aligned}}$$

1pt opcodes

1pt sketch

2 pts intersections

x 4

9. (12 pts) Clip the triangle with points A = (-1.5, .1), B = (.5, -.5), C = (0,1.5) against the box (-1,-1), (1,-1), (1,1), (-1,1). Use the Sutherland-Hodgeman algorithm to clip the polygon.

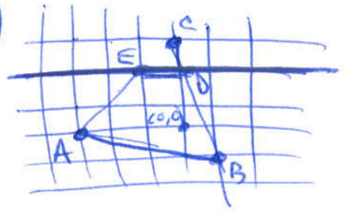
a) Give the vertex list after clipping against the top edge, and draw a sketch showing the configuration including the location of any new points you may need to create. If you need to create new points, give them names in alphabetical order (D, E, F, ...). Use the same labels for the new points in your sketch and in the vertex list. Show your intermediate work, in terms of inside/outside checks for each vertex pair. You do **not** need to compute exact values of the the intersection points.

b) Same as above, after clipping against the bottom edge.

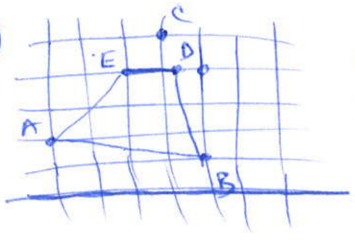
c) Same as above, after clipping against the right edge.

d) Same as above, after clipping against the left and final edge of the box.

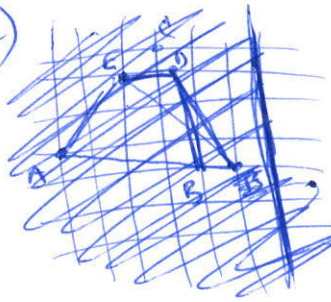
initial list : ABC

a) 

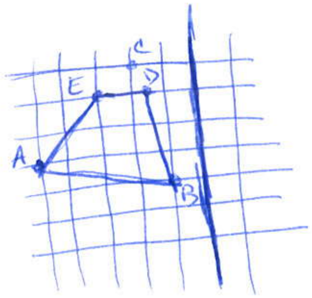
	List
AB in → add B	B
BC in/out → add isect D	BD
CA out/in → add isect E and A	BDEA

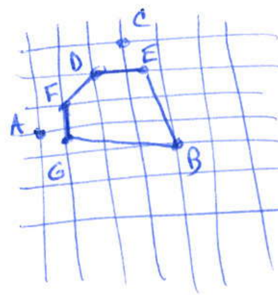
b) 

	List
BD in → add D	D
DE in → add E	DE
EA in → add A	DEA
AB in → add B	DEAB

c) 

	List
DE in → add E	E
EA in → add A	EA
AB in → add B	EAB
BD in → add D	EABD

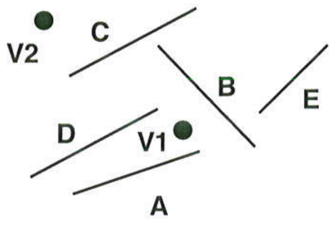


d) 

	List
EA in/out → add isect F	F
AB out/in → add isect G and B	FGB
BD in → add D	FGBD
DE in → add E	FGBDE

1 pt sketch, 2 pts vertex list (w/ intermediate in/out checks!) x 4

10. (15 pts) Build a BSP tree for the following scene using the polygons (shown as line segments). The cutting plane induced by a polygon should just extend along the line itself. The labelled side of the polygon should be the right child in the tree, and the unlabelled side should be the left child.



2pts a) Give the BSP tree with the single root node of polygon A, and sketch the entire scene with the addition of the new cutting plane.

2 pts b) Same as above, after adding polygon B.

2 pts c) Same as above, after adding polygon C.

2 pts d) Same as above, after adding polygon D.

1 pt e) Same as above, after adding polygon E.

3 pts f) Traverse your BSP tree to produce a painter's algorithm ordering from eye point V1. Show your work at each step in the traversal, starting from the root of the BSP tree.

3 pts g) Same as above, instead using eye point V2.

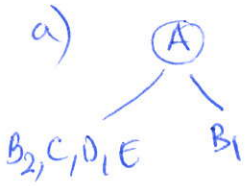
-1 if traverse backwards

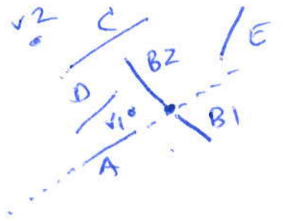
-1 if not subdivide B or C (-1 each)

-1 for wrong side of parent

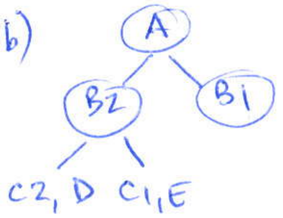
-1 overall if did left-handed

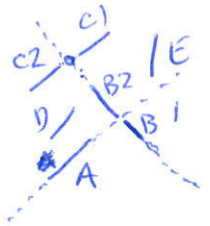
a)



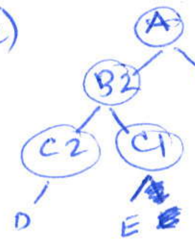


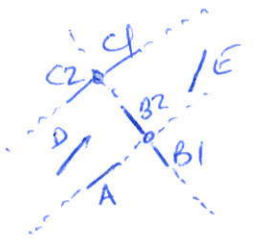
b)



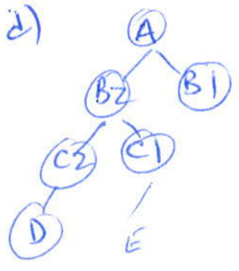


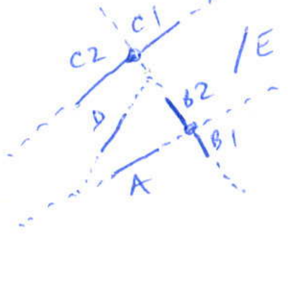
c)



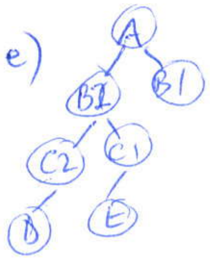


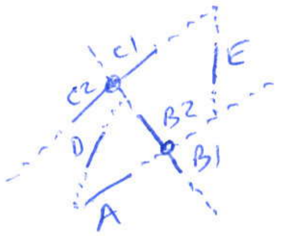
d)



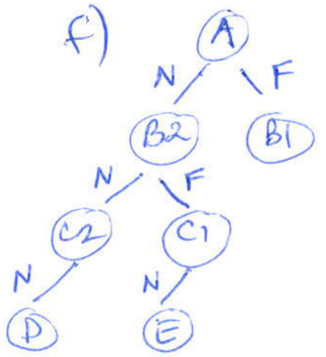


e)





f)



[far A] A [near A]

↓

B_1 A [far B_2] B_2 [near B_2]

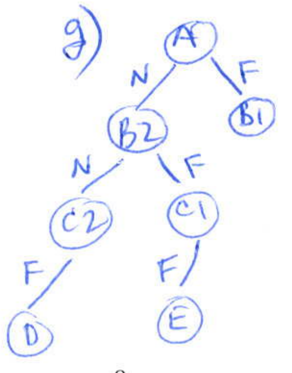
↓

B_1 A [far C_1] C_1 [near C_1] B_2 [far C2] C2 [near C2]

↓

B_1 A C_1 E B_2 C_2 D

g)



[far A] A [near A]

↓

B_1 A [far B2] B_2 [near B_2]

↓

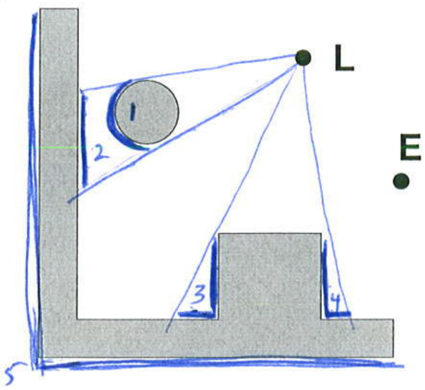
B_1 A [far C_1] C_1 [near C_1] B_2 [far C2] C2 [near C2]

↓

B_1 A E C_1 B_2 D C_2

11. (5 pts) You are raytracing a scene with only a single light source L from the eye point E. There is no ambient light. The grey objects are completely opaque. Sketch on the diagram below the regions where there is no illumination.

1 pt each region
−1 pt any wrong region
(1 pt for any concept of drawing in shadow rays)



12. (4 pts) You have a checkerboard image where each square is 5x5 pixels. How close together would your sample points be at the Nyquist Rate?

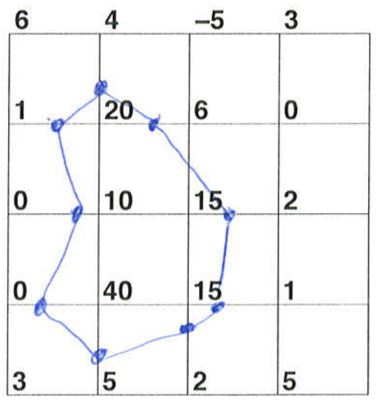
sample every $(5+5)/4$ pixels → 2.5 pixels. round <u>down</u> to 2

full credit for 2.5 or 2

13. (1 pt) Why was the Z-buffer an impractical approach to visibility in the 1960's? Answer briefly.

memory was expensive, z buffer requires a lot more of it than analytic/bsp approaches

14. (4 pts) Sketch the 1D isosurface at iso-value 9 for the 2D density function below.

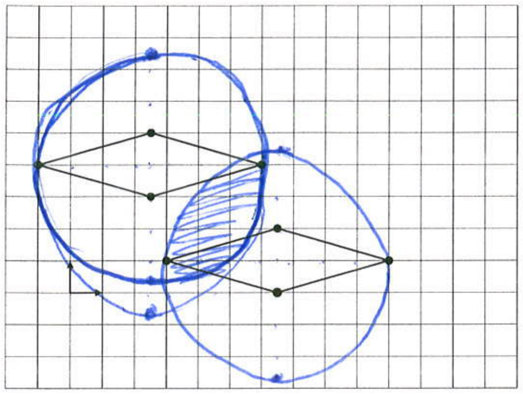


−1 pt for each wrong intersection spot

15. ~~(4~~ 2 pts) Object A has vertices (2.5,3), (6,4), (2.5,5), (-1,4). Object B has vertices (6.5,0), (10,1), (6.5,2), (3,1). Perform collision detection on these two objects using bounding spheres as the collision proxy. Show your work with a sketch, and state whether the proxies collide.

1 pt sketch

1 pt yes/no

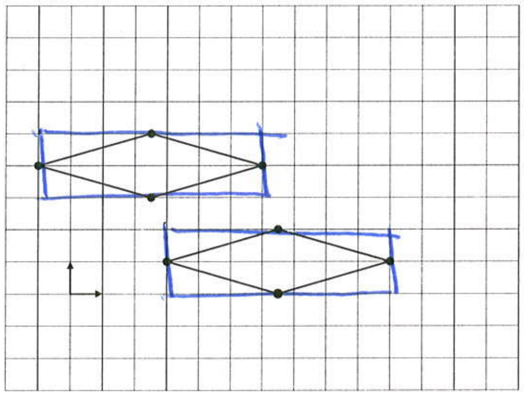


yes, collide

16. (2 pts) Perform collision detection on the same objects using axis-aligned bounding boxes for the collision proxy. Show your work with a sketch, and state whether the proxies collide.

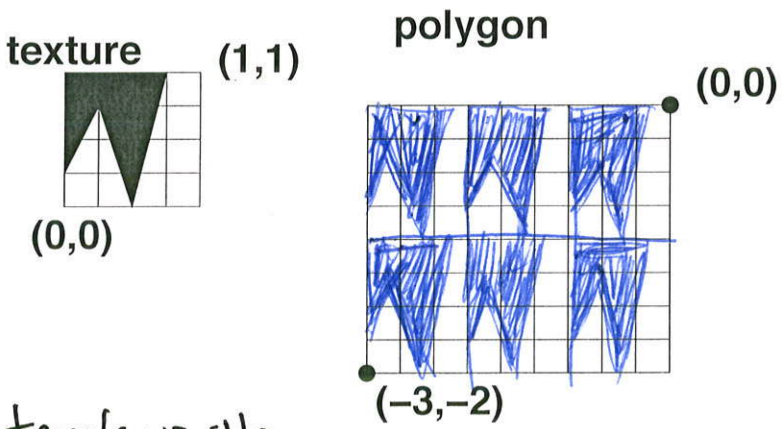
1 pt sketch

1 pt yes/no



no, donot collide

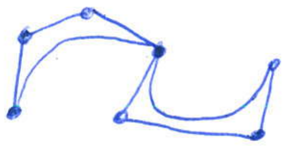
17. ~~(2~~ 4 pts) Sketch the texture on the polygon with the given (s,t) texture coordinates, if the texture mode is tile/repeat (as opposed to clamping):



2 pts: 6 texels visible

2pts: orientations correct

18. (3 pts) Draw a curve made from two piecewise Bézier curves that is C^0 continuous but not C^1 continuous.

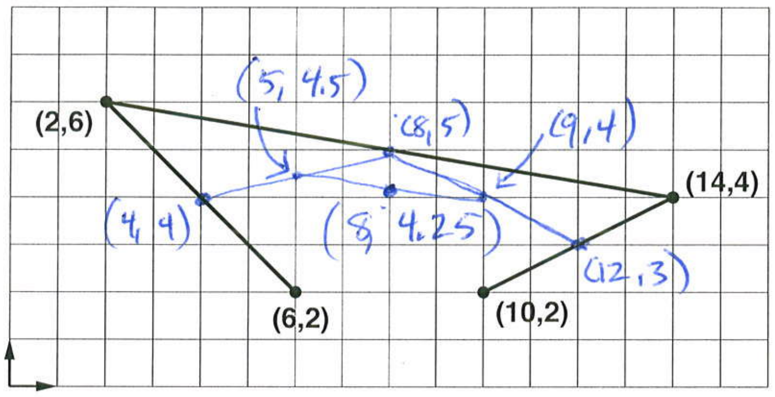


1 pt C^0 cont

1 pt not C^1 cont

1 pt both Bez curves

19. (5 pts) Compute the midpoint of the Bézier curve specified by the four control points below using the de Casteljau algorithm, showing your work. Your answer should be in the form of a sketch on the grid below, labelling all new points in your construction with (x,y) locations.



2 pts correct number labels

3 pts correct sketch

(1 pt vague idea)

(2 pts close but not quite)

-1 if did 1/4 not 1/2

-1 if actual midpoint not computed

20. (30 pts) Your Z buffer has 24 bits, so objects can be sorted by depth into one of $2^{24} = 16777216$ bins represented as integers. The OpenGL perspective matrix is used for projection, with the near plane set to $n = .01$ and the far plane set to $f = 100$.

The scene consists of several pairs of polygons. All polygons are parallel with the image plane. Each pair has partial overlaps in x and y, but has different depths. In eye/view/camera coordinates (VCS), the polygons have the following depths:

- Polygon A: 99.02
- Polygon B: 99.03
- Polygon C: .02
- Polygon D: .03

The relationship between device and view coordinates is

$$z_D = 1/2 + 1/2\left(\frac{f+n}{f-n} + \frac{1}{z_V} * \frac{2*f*n}{f-n}\right)$$

8 pts a) Give the depth of each polygon in device coordinates (DCS) as a floating-point number ranging from 0 to 1.

8 pts b) Give the depth of each polygon in integer screen coordinates as an integer ranging from 0 to $2^{24} - 1$.

4 pts c) Which polygon pairs could result in shimmering artifacts when drawn, because there is insufficient precision in the z-buffer to resolve their depth relationship?

10 pts d) Derive the equation above from the OpenGL perspective projection matrix and the definitions of normalized device coordinates (NDCS), and device coordinates (DCS).

$$\frac{f+n}{f-n} = \frac{100+.01}{100-.01} = \frac{100.01}{99.99} = 1.0002 \qquad \frac{2fn}{f-n} = \frac{2*100*.01}{99.99} = \frac{2}{99.99} = .02$$

a) $z_D = .5 + .5\left(1.0002 + \frac{.02}{z_V}\right) \Rightarrow .9999\ A,\ .9999\ B,\ .5001\ C,\ .6667\ D$

2 pts each × 4

b) 16,775,537 A, 16,775,537 B, 8,390,285 C, 11,185,369 D

same

$2^{24} = 16{,}777{,}216$

$2^{24} - 1 = 16{,}777{,}215$

2 pts each × 4

answers depend on precision, range accepted w/ full credit

c) A/B will fight (but C/D will not!)

d)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \cancel{x_v} \\ \cancel{y_v} \\ \cancel{z_v} \\ \cancel{w_v} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ z_V \\ 1 \end{bmatrix} = \begin{bmatrix} - \\ - \\ \frac{-(f+n)}{(f-n)} z_V - \frac{2fn}{(f-n)} \\ -z_V \end{bmatrix}$$

V2N

2 pts

$$z_N = \frac{-(f+n)}{f-n} z_V + \frac{-2fn}{f-n}$$

$$w_N = -z_V$$

2 pts

homogenize to get z_N / w_N

$$\frac{f+n}{f-n} + \frac{2fn}{f-n} \cdot \frac{1}{z_V}$$

2 pts

in DCS, we know z_{DCS} is $[0,1]$ and in NDCS, we know z_{NDCS} is $[-1, 1]$

thus scale + offset 1/2, 1/2

$$z_{DCS} = \frac{1}{2} + \frac{1}{2} z_{NDCS} = 1/2 + 1/2\left(\frac{f+n}{f-n} + \frac{2fn}{f-n}\left(\frac{1}{z_V}\right)\right)$$

4 pts