Name:____________________________

ECSE-4750 Computer Graphics, Fall 1998
Handout 20*
Collection of Old Final Exam Questions

April 22, 1999

Remember that the course material changes a little each year. There are many questions on material that I didn't cover this time. Therefore, don't get too paranoid at seeing things here for the first time. I included all the questions to give you a flavor for the sorts of likely questions. On the other hand, some material that I didn't cover might be missing. It would be a mistake to study only this exam, while ignoring the class handouts and what was actually covered.

Rules

1. You may have two 2-sided 8.5 inch x11 inch note sheets, which may be mechanically printed.
2. You may have your blank paper, calculator, pens, etc.
3. You may not communicate with anyone, except Helen or me.
4. Answer all questions. Brief, concise, answers are preferred.
5. Spend time on a question proportional to its number of points.
6. Note that the last page is number **20–23**.
7. Start immediately. You have until noon.
8. Try to write legibly.
9. Write your NAME on top of this page.
10. Try to write your answers on these question sheets, tho extra paper is allowed. If an answer is on an extra sheet, say so in the normal space on this sheet.
11. Leave the small oval boxes blank; they're for our grade.

Exam

Geometry Warmup

1. *Geometry warmup:* In 3D, consider the X-axis and the point $(1, 2, 3)$.

   (a) /[2] What is the equation of the plane thru the both of them, in the form $ax+by+cz+d = 0$?

   (b) /[1] At what point does the Z-axis intersect this plane?

2. Suppose that we have a plane in 3-D thru the points A(0,0,2), B(0,2,0), and C(0,0,0).

   (a) /[1] What is its equation, in the form $ax + by + cz + d = 0$?

   (b) /[1] Consider the line L thru the points O(0,0,0) and P(1,2,3). Where does this line intersect the plane?

X

3. /[4] Contrast X/Motif and Tcl/Tk on these factors:

   (a) ease of use,

   (b) efficiency of execution,

   (c) availability of source code, and

   (d) demand by employers.

Intro

4. /[1] In what decade was the first interactive computer graphics? Select one:

   (a) 1930s
   (b) 1940s
   (c) 1950s
   (d) 1960s
   (e) 1970s

5. /[1] What is the most important technical advance that made frame buffers feasible?

6. /[4] Name 3 advantages and one disadvantage of portability and standards in graphics.

7. /[1] Portability has at least one disadvantage, other than making it easier for your competitor to copy your program. Name it.

8. About *Foley*, chap 1:

   (a) /[1] Why is raster graphics better than vector graphics?

   (b) /[1] So, why didn't people use raster graphics a lot 20 years ago?

Chapter 2, SRGP etc

9. /[2] Suppose that you are specifying a graphics system where many successive objects will share the same attributes. How might you reasonably handle attribute specification? What disadvantage would your proposal have?

10. /[1] Why do graphics packages contain polyline routines in spite of the fact that users could easily just call the line routine many times?

11. /[2] Name an advantage and a disadvantage of specifying attributes for, e.g., a line, in every call to draw the line.

12. /[3] What are bundled attributes? Give an advantage and a disadvantage of them.

13. /[2] If you're designing a simple raster graphics package, you probably want to include a few functions that would be either meaningless or difficult on a vector graphics system, but which are very useful on a raster one. Name two.

14. /[2] Give an advantage and a disadvantage of using data structures such as *Point* instead of listing the X and Y coordinates separately.

15. /[1] What's a *RasterOp*?

16. /[2] A *logical keyboard device* is a device that returns characters of text. Many different physical devices might be a logical keyboard. One such is a physical keyboard. Please name two other physical devices that might be a logical keyboard. (This requires thinking, but doesn't require having seen logical devices before.)

Algorithms

17. /[3] Suppose that you are colorizing the B&W movie *Casablanca*. Describe how your most likely polygon fill algorithm would work.

18. /[1] What should be done with people who colorize classic B&W movies?

19. /[2] Suppose you used a paint program to outline a region by setting pixels around its border. Describe how the paint program could then operate to fill the region's interior. Illustrate this by filling the interior of the letter B.

20. I described 2 fill algorithms, called *polygon fill* and *seed fill*.

    (a) /[1] Which one would *xfig* be more likely to use?

    (b) /[1] Which one would *xpaint* be more likely to use?

21. /[2] How can the seed fill method be used when colorizing movies? The problem is that a frame of a black-and-white movie doesn't have pixels of a certain value outlining each region that should be a single color.

22. /[1] You can start the Bresenham algorithm with the initial point (x_0, y_0) being a fraction by scaling everything up to integers. The pixels selected to draw will still be integers on the screen. Why might you be interested in doing a Bresenham line algorithm at a sub-pixel accuracy instead of simply rounding the line endpoints (or circle center and radius) to the nearest pixel before applying Bresenham? (This one may require some thinking.)

23. /[2] One step along the way towards the Bresenham line algorithm has the program looking like this:

```
m=y1/x1;
d=0;
pixel(0,0);
y=0;
for(x=1;x<=x1;x++)
{  d+= m;
   if (d>= 1/2)
   {  d -= 1;
      y++;  }
   pixel(x,y);
}
```

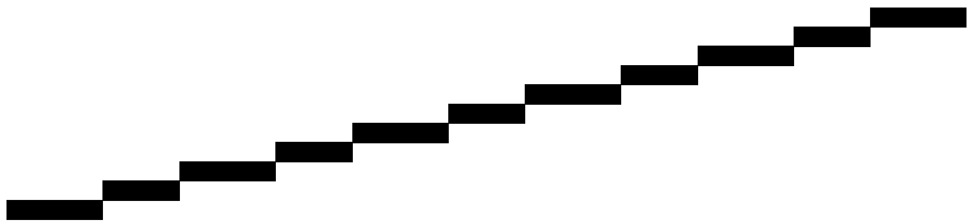
    Then I "Scale up m and d by 2x" to create the next version of the code. Why?

24. /[2] In the Bresenham algorithm for straight lines, why is the parameter d (notation used in class) shifted to vary between 0 and $-2x1$, rather than $-x1$ to $x1$?

(a) /[1] What property of many modern CPUs makes *if* statements undesirable in tight loops?

(b) /[1] How was the Bresenham line algorithm modified because of that?

25. /[2] One problem with the simple way of drawing a line is that the result has an objectionable staircase look:



(a) /[1] What is this called?

(b) /[1] What property of the human visual system makes it expecially noticeable?

(c) /[2] Describe one solution.

26. /[2] What is the *aliasing* problem in line drawing, and what's a simple reasonable fix?

27. /[1] Why would you be interested in doing a Bresenham algorithm at a sub-pixel accuracy instead of simply using a higher resolution device?

28. /[1] What is the advantage of a Bresenham-style algorithm compared to simpler, more obvious methods?

29. /[1] Why might you still want to use a Bresenham-style algorithm even tho modern processors do floating as fast as integer arithmetic?

(a) /[1] When filling a polygon, which is defined by a list of vertices, the simple way, what problem is caused by the current scan line going through a vertex?

(b) /[2] How do you work around this problem?

30. /[1] Suppose that I insist on using a Taylor series for the circle $\sqrt{1-x^2}$, in the 2nd octant: $0 \le x \le y$, where the curve doesn't get too vertical. I might expand the series about the origin, but what might I do instead to make it converge faster, while still being a Taylor series?

31. ☐/[2] Which of the following 3 methods will draw the most accurate circle $y = \sqrt{100 - x^2}$ over the interval $0 \le x \le 71$, given the same number of terms in the approximation? Which method will be second best?

(a) a Taylor expansion about $x = 0$.

(b) a Taylor expansion about $x = 35$.

(c) a Chebyshev expansion over the interval$[0, 71]$.

32. ☐/[4] In creating the Bresenham circle algorithm, we transformed this code:

```
pixel(0,r);
for(x=1;x<r/sqrt(2);x++)
{  y=sqrt(r^2-x^2);
   pixel(x,y);
}
```

into this code:

```
y=r;
pixel(0,r);
for(x=1;x<r/sqrt(2);x++)
{  d=x^2+(y-1/2)^2-r^2;
   if (d>=0) y--;
   pixel(x,y);
}
```

Please explain why the second code fragment draws the same pixels as the first.

33. ☐/[2]

In this version of the Bresenham circle algorithm, look at the comment.

```
y=r;
d= -r;
pixel(0,r);
for(x=1;x<r/sqrt(2);x++)
{  d+= 2x-1;
   if (d>=0)
   {  y--;
      d -= 2y; /* Must do this AFTER y-- */
   }
   pixel(x,y);
}
```

Would we still get the right answer if we ignored the comment, and swapped the two statements thus:

```
...
   if (d>=0)
   {  d -= 2y; /* Must do this AFTER y-- */
      y--;

   }
...
}
```

Why (not)?

34. ☐/`[4]` Calculate the pixels on a circle of radius 17 using the Bresenham method. Show your work.

35. ☐/`[1]` Suppose that I draw a circle of radius 25 thus on a 1024×1024 display:

$$
\begin{aligned}
x &= 25\cos\theta \\
y &= 25\sin\theta
\end{aligned}
$$

What should $\Delta\theta$ be, in radians?

36. Consider a spoked stagecoach wheel in a Western movie. The wheel may appear to rotate backwards even tho the coach is moving forewards.

    (a) ☐/`[1]` Suppose the wheel had twice as many spokes. Would this artifact be more likely or less likely to appear?

    (b) ☐/`[1]` What is this problem called in graphics?

    (c) ☐/`[2]` What is one possible solution?

37. ☐/`[2]` If a color TV camera is aimed at someone wearing a vertically striped shirt, then aliasing may occur. Assume that the person is one foot wide and ten feet away from the camera, which has 512 pixels across a line. What is the maximum number of shirt stripes per inch before aliasing is a problem? If you have to make other assumptions to solve this, then state them.

38. ☐/`[2]` Name one advantage and one disadvantage of using outline character fonts instead of bitmapped chars.

39. ☐/`[1]` Are *bitmapped* or *outline* fonts more suitable to a low-res terminal?

40. ☐/`[1]` Are *bitmapped* or *outline* fonts more suitable to a 1200 DPI printer?

41. Concerning *Unicode*:

    (a) ☐/`[3]` What is it?

    (b) ☐/`[1]` How many bits per char does it use?

42. ☐/`[1]` Name the basic difference in representation between PostScript fonts and X fonts (at least up X11R4).

43. /[2] Suppose that I want to design some bitmapped fonts for my workstation, which has 75 pixels per inch. I might design the fonts to 75 dpi, so that one pixel of the font is one pixel on the screen. Alternatively, I might take a 300 dpi font, and average it down. 16 bits of the font will become one grey pixel on the screen.

    Which method is better and why?

44. /[2] How much space would a family of bitmap fonts take if each font has 128 chars, 1/6 inch hi and wide, printed on a Laserwriter at 600 dpi? Assume that the family contains four fonts: Roman, Italic, Bold, and BoldItalic, each stored separately.

45. /[2] Name two ways to make a type face harder to electronically eavesdrop via the CRT's radiation. Use at most 5 words for each way.

46. /[1] How many bits is an ISO-8859 character?

47. /[1] Give an example where seed-filling a raster polygon may fail.

48. Characters are normally typeset by assuming a rectangle around each char, and placing the rectangles adjacent to each other. That works well in most cases, such as this: **ABCDEFGHIK**. However, sometimes it causes apparent gaps, like between A and V here:
    AVXAVXAVXAVX.

    (a) /[1] What's the name of the fix in PostScript?

    (b) /[1] How does it work?

49. /[2] What is kerning? Give an example.

50. /[2] What is the difference between the sample and event input modes?

51. Cohen-Sutherland algorithm:

    (a) /[2] Summarize it.

    (b) /[2] Draw a type of input scene and clip region for which it works quite fast. Draw a case where it works slowly.

52. This is about the Sutherland-Hodgman paper and patent.

    (a) /[2] The paper says that, "The final division process in the perspective projection destroys one bit of sign information, and this is precisely the bit that $\cdots$."

    What important info gets lost here, and what problem does this cause?

(b) /[1] What's the solution?

(c) /[1] Why do you want to clip a line against a front (aka hither) clipping plane before projecting?

(d) /[1] In what form did the patent cast the algorithm to make it patentable?

(e) /[1] Tell how the patent represents vertices.

(f) /[2] Draw an example of a polygon and a clip window, where the clipped polygon will have twice as many vertices as the original one.

(g) /[2] Draw an example where the original polygon is connected, but the clipped result isn't.

(h) /[2] Suppose that you want to cut a scene into 2 parts, a left and a right part. Can the Sutherland-Hodgman algorithm be easily modified to do this? How?

Hardware

53. /[3] Name 3 different hardware ways to implement a logical locator device.

54. /[4] Name 4 different hardware ways to implement a logical keyboard.

55. /[1] What is the relevant property of liquid crystals that makes them useful in displays?

56. /[1] How does an *active matrix* flat-panel display differ from a *passive matrix* one?

57. /[1] How does an *active matrix* flat-panel display differ in design from a *passive matrix* one? I.e., what makes it more expensive and better looking?

58. /[2] Suppose that you place a lens of a pair of polarized sunglasses directly in front of a liquid crystal display, then rotate the lens and look at it. Will the appearance of the LCD change? Explain why, with reference to the construction of an LCD. (This requires actual thinking.)

59. /[1] Which hardcopy device might remind you of the movie *Animal House*?

60. Computer monitors tend to use an RGB color space, but color NTSC TVs do not.

    (a) /[1] What space do they use?

(b) /[1] How does the choice of first axis make color NTSC more compatible with something or other?

(c) /[1] How does the choice of the last 2 axes compat with something about human color perception?

61. /[1] What limits the practical resolution of color CRT monitors?

62. /[1] Why can B&W CRTs have a higher resolution than color CRTs?

63. /[1] Why are active-matrix LCD displays so expensive?

64. /[2] Name two assumptions that the designers of NTSC made about the nature of the scenes that would be televised.

65. /[3] Give three reasons why color printers have black ink in addition to magenta, cyan, and yellow, since those three should mix to make black?

66. /[3] In order to make color NTSC upward and downward compatible with B&W, two facts about human color perception and (at least) one fact about typical scenes were used. Name them.

67. /[2] Why is it that a black and white TV can receive the same signal as a color TV, yet the B&W TV was designed before the color signal was designed?

68. /[1] Why is a TV a bad monitor to use for a video game; specifically, how does the signal generated by the game often violate NTSC assumptions and rules?

69. /[3] How does a bit pad (also known as a data tablet) work? There are several methods; just mention one. What is the major difference in the user interface between a bit pad and a mouse with mouse pad? What logical input device do they both represent?

70. /[2] How does a data glove work? Mention the 2 major components.

71. /[2] What are one advantage and one disadvantage of having the frame buffer a part of the user memory instead of being a separate peripheral on its own bus? (This might require some thinking.)

72. /[2] Suppose that we have a color display with 16 bit planes. The A/D converters for each color take 8 input bits each. The display is 1024x1024. How many different colors can be displayed at one time? (I'm ignoring fancy techniques such as some Amigas use, which were not mentioned in class.)

73. /[1] Suppose that we want to display a gray scale image on a high resolution B&W device that does not have gray scale. How can we do it?

74. /[1] In color NTSC, moving objects with sharp edges may have little beads of color run along the edges. What's going on here?

75. /[1] In color NTSC, moving objects with sharp edges may exhibit *dot crawl*, which is little beads of color running along the edges. What's going on here?

76. /[1] If a color TV scene has differently colored moving objects, then little beads, or small areas of another color may move up and down along the edges of the large object as it moves. What limitation in the NTSC standard causes this?

77. Long persistence phosphors decrease the fusion frequency.

    (a) /[1] What is the fusion frequency?

    (b) /[1] Why don't we always use long persistence phosphors?

78. /[2] What is the *video controller* in a raster graphics system?

79. Assume that you are building a frame buffer for a 1024x1024x24 bits deep display.

    (a) /[2] You can use 4Mbit 60 nsec DRAMs. How many do you need? What is that maximum refresh rate possible before you are reading data from the DRAMs as fast as possible?

    (b) /[2] Now you are allowed to use 512Kx8 VRAMS (4Mbit total in each). How many do you need? What is the max possible refresh rate now?

80. /[2] Consider the design of a 1024x1024x8 frame buffer with 4Mbit DRAMS. The frame buffer would use 8 chips. If I refresh at 60Hz, then I have to read 60 million bits per second from each chip, or one bit every 1.7 nsecs. How is this compatible with the chip having a speed of 60 nsecs?

81. /[2] Give 2 reasons that the Intel i860 chip does graphics so fast.

82. /[2] If you connect a video game to a household TV, which uses NTSC, then the video signal from the game probably violates some assumptions built into NTSC.

    (a) Name one such violation.

    (b) Describe how this violation distorts the TV image.

83. /[4] There are two main types of mice. Name them, describe how they work, and give an advantage that each has relative to the other.

84. /[3] What is a VRAM? What's its reason for existing? If it's sometimes better than a DRAM, which you might guess from this question, then why are DRAMs ever used?

85. /[2] Suppose that your video game generates a narrow horizontal white line on your TV. The line may appear to flicker at 30Hz even though other parts of the image, such as large rectangles, look stable. Why? How do high quality computer graphics displays prevent this problem? (Not playing video games, and not displaying narrow lines, are not acceptable answers.)

86. Interlacing:

    (a) /[1] What is it?

    (b) /[2] Do NTSC TVs do it? Why (not)?

    (c) /[2] Do good color computer monitors do it? Why (not)?

87. /[1] It's clear why 8-bit frame buffers use color maps. However, some 24-bit buffers also do. (It's not one 2^{24} entry cmap, but 3 256-entry cmaps, one per primary color.) Why might they want to do this since it does make things more complicated?

88. /[2] Name two hardware pieces that a more expensive, higher performance, graphics system might have that the cheaper one would not.

Transformations

89. /[2] Suppose that the matrix $\mathcal{A}$ represents a rotation operation, and $\mathcal{B}$ represents a parallel projection. Remember that in this course we represent points as vertical vectors, and multiply them on the left by matrices. What is the matrix for the projection followed by the rotation?

90. /[1] Why can a rotation matrix never have an eigenvalue of 2?

91. Our vector formula for rotating a point p about an axis a is this:

$$p' = a \cdot p\, a + (p - a \cdot p\, a)\cos\theta + a \times p \sin\theta$$

    (a) /[1] Does p need to be normalized?

    (b) /[1] Does a need to be normalized?

92. I'm trying to rotate a point many times by a small angle, $\theta = 0.0001^o$, by applying the usual 2D rotation matrix. $\cos\theta = 1$, and $\sin\theta = 0.000001745$, both to within 10^{-10}. Therefore,

$$M = \begin{bmatrix} 1 & 0.000001745 \\ -0.000001745 & 1 \end{bmatrix} \tag{1}$$

Now I rotate $p = (1, 2)$ for 36000000 times thus: $p' = M \cdot p$, which should be ten complete circles. Unfortunately, I get, not $(1, 2)$, but $(0.9763, 2.0118)$.

(a) ☐/[1] What happened? Obviously it's some sort of roundoff error, but could you be more specific about something in my method that was bad?

(b) ☐/[1] How might I fix it, w/o using more digits of precision?

93. ☐/[2] I'm trying to rotate the square with opposite corners at (0,0) and (1,1) many times by a small angle, $\theta = 0.1^o$, by applying the usual 2D rotation matrix. $\cos\theta = 0.99984770$, and $\sin\theta = 0.00174524$, both to within 10^{-8}. Therefore,

$$M = \begin{bmatrix} 0.99984770 & 0.00174524 \\ -0.00174524 & 0.99984770 \end{bmatrix} \tag{2}$$

The problem is that after I use Matlab to apply this matrix 3600 times, which is one complete rotation, the area of the rotated square is only 0.3377, down from 1. What test on the matrix could have told me that this would happen, including giving the actual resulting area, without my having actually to rotate the points?

94. ☐/[1] Show that the following matrix cannot be a 3-D rotation. You should be able to do it by looking at it, w/o doing any messy math or using a calculator.

$$\begin{bmatrix} 0.7222222223 & -0.4662391580 & 0.5108973566 \\ 0.6884613802 & 0.5555555557 & -4.662391580 \\ -0.0664529124 & 0.6884613802 & 0.7222222223 \end{bmatrix} \tag{3}$$

95. Consider the following matrix, which is a 3-D rotation:

$$\begin{bmatrix} 0.7222222223 & -0.4662391580 & 0.5108973566 \\ 0.6884613802 & 0.5555555557 & -0.4662391580 \\ -0.0664529124 & 0.6884613802 & 0.7222222223 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} 0.9023689270 & -0.1910440617 & 0.3863062075 \\ 0.3863062075 & 0.7559223176 & -0.5285347327 \\ -0.1910440617 & 0.6261658057 & 0.7559223176 \end{bmatrix} \tag{5}$$

(a) ☐/[1] What is the cosine of the angle of rotation?

(b) ☐/[1] What is the axis of rotation? You don't need to normalize it.

96. Look at the following list of operations.

(a) Rotation about the axis through (0,0,0) and ($\sqrt{2}$,2,$\sqrt{2}$).

(b) Translation by a displacement of (-5,1,-1).

(c) Orthographic projection onto the plane passing through (0,1,0), (1,1,1) and (1,0,1).

(d) Perspective projection onto the plane through (1,0,0), (0,1,0), and (0,0,1) with origin as the view-point.

(e) Rotation about the straight line through (0,1,0) and (1,1,1).

(f) Reflection about the xz plane.

(a) /[3] Which of them cannot be represented by a 3x3 matrix operator?

(b) /[3] Which ones can be represented by a 4x4 homogeneous coordinates matrix?

97. /[2] Give 2 different homogeneous coordinate matrices for scaling any given homogeneous vector (x, y, z, w) to half its cartesian length.

98. /[1] Give a homogeneous coordinate matrix for scaling any given homogeneous vector (x, y, z, w) to half its cartesian length.

99. /[1] Quickly prove by simple inspection that the following matrix cannot be a rotation matrix.

$$\begin{bmatrix} 1.8047378540 & -0.3106172175 & 0.5058793633 \\ 0.5058793633 & 0.8047378540 & -0.3106172175 \\ -0.3106172175 & 0.5058793633 & 0.8047378540 \end{bmatrix}$$

100. /[2] Rotate the 3D point (1,0,0) about the axis (0.577, 0.577, 0.577) by 180 degrees.

101. /[2] Rotate the 3D point (0,1,0) about the axis (0.577, 0.577, 0.577) by 180 degrees.

102. /[2] Consider the function $F(p) = (a \cdot p)a$, where a and p are 3-D Cartesian vectors, and $a = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$. Find a 3x3 matrix M, such that $F(p) = Mp$.

103. /[2] Consider the function $F(p) = (a \times p)$, where a and p are 3-D Cartesian vectors, and $a = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$. Find a 3x3 matrix M, such that $F(p) = Mp$.

104. /[2] Consider the function $F(p) = (a \times p)$, where a and p are 3-D Cartesian vectors, and $a = \begin{pmatrix} 7 \\ 6 \\ 5 \end{pmatrix}$. Find a 3x3 matrix M, such that $F(p) = Mp$.

105. /[1] Give me a set of linear equations whose solution is the axis of rotation.

106. /[1] Is the following matrix a 2-D rotation? Why (not)?

$$\begin{bmatrix} 4/5 & 4/5 \\ -4/5 & 9/20 \end{bmatrix} \tag{6}$$

107. /[1] Explain the rationale for the columns, each considered as a vector, being perpendicular to each other, when testing whether M is a rotation matrix.

108. /[1] Explain the rationale for $|M| = 1$ when testing whether M is a rotation matrix.

109. /[1] If the determinant of a transformation matrix is not one, then when it is applied to an object, some property of the object is changed. What?

110. /[1] Do 2D rotations commute, that is, if $\mathcal{A}$ and $\mathcal{B}$ are two 2D rotation matrices, is $\mathcal{AB} = \mathcal{BA}$ always, usually, occasionally, or never?

111. /[2] Answer the above question for 3D rotations. If the answer is "sometimes", then give an example when they do.

112. /[2] Suppose that you wish to rotate about an axis that does not go thru the origin, but goes thru the point C instead. How could you do this?

113. /[2] The *trace* of a matrix is defined as the sum of the elements along the main diagonal, that is, $a_{11} + a_{22} + a_{33}$ for the 3×3 matrix A. It is invariant if the matrix's coordinate system is rotated, which implies that the trace always equals the sum of the eigenvalues.

Describe how you could use this to find $\cos(\theta)$ of a rotation matrix really easily.

114. /[1] Consider the following matrix, which is a 3-D rotation:

$$\begin{bmatrix} 0.7222222223 & -0.4662391580 & 0.5108973566 \\ 0.6884613802 & 0.5555555557 & -0.4662391580 \\ -0.0664529124 & 0.6884613802 & 0.7222222223 \end{bmatrix} \tag{7}$$

What is the cosine of the angle of rotation?

Projection

115. /[3] Name 3 advantages of homogeneous coords.

   (a) /[2] One advantage of homogeneous coordinates is that points at ∞ can be represented. What is the homogeneous coordinate representation for the point at ∞ on the line $y = x + 2$? (Might require thinking.)

   (b) /[2] One advantage of homogeneous coordinates is that points at ∞ can be represented. What is the homogeneous coordinate representation for the point at ∞ on the line $y = x - 3$?

   (c) /[2] Name 2 other advantages of homogeneous coords.

116. /[2] Calculate the 4x4 homogeneous matrix for a perspective projection from the origin to the projection plane at $x = d$.

117. /[2] Find the 4x4 homogeneous matrix for a perspective projection from the origin to the projection plane with cartesian equation $x + 3y + 2z = 1$.

118. Consider a perspective projection with the center of projection at $(0, 0, -1)$ and the projection plane at $z = 0$. For example, the point $(0, 0, 0)$ would project to itself since it's on the plane. The point $(2, 2, 1)$ would project to $(1, 1, 0)$ since a line from $(2, 2, 1)$ to $(0, 0, -1)$ would intersect $z = 0$ at $(1, 1, 0)$.

   (a) /[3] Where would these Cartesian points project to?

      i. $(1, 0, 3)$
      ii. $(1, 1, 4)$
      iii. $(0, 1, 5)$

   (b) /[3] Find the 4×4 homogeneous matrix for this projection transformation, scaled so that all the entries are integers. *(This requires understanding of how I derived the matrix in class, and then thinking.)*

119. /[3] With that projection, what Cartesian points would these Cartesian points project onto:

   (a) (1,0,0)
   (b) (1,1,0)
   (c) (1,1,1)

120. /[1] What point would the infinite homogeneous point (1,2,3,0) project onto?

121. /[3] What are the 3 steps of the conceptual model of the 3D viewing process?

122. Consider these terms: VRP, VUP, VPN, and PRP.

(a) /[1] Which is/are used to define the view plane?

(b) /[1] Which is/are used to define the center of a perspective projection?

(c) /[1] Which one has nothing to do with the local uv coordinate system on the view plane?

123. /[2] (From my rotation paper) What are Euler angles?

124. /[2] Rewrite the following matrix as a 4x4 homogeneous rotation matrix, using only integers.

$$\begin{bmatrix} -2/3 & 1/3 & 1/3 \\ 2/3 & 1/6 & 2/3 \\ 1/3 & 2/3 & -2/3 \end{bmatrix}$$

125. This question requires some original thinking about homogeneous coordinates. To keep it simple, I'll use 2-D points, which have 3 homogeneous components: (x, y, w).

Pairs of cartesian points have equations of lines thru them, such as (1,0) and (0,1) are on the line $x+y-1=0$. Similarly, a pair of homogeneous points has a line thru them, and it has an equation. (1,0,1) and (0,1,1) are on the line $x+y-w=0$. Note the similarity. Note that (2,0,2) and (10,5,15) work also, as they should. All homogeneous line equations can be written in the form $\alpha x + \beta y + \gamma w = 0$ for some α, β, γ.

(a) /[2] What would be the equation of the homogeneous line thru the homogeneous points (0,1,0) and (0,0,1)?

(b) /[2] What would be the point that is on that line and also on the line $x - y - w = 0$?

126. /[1] Consider these 2 operations: 1) uniform scale about the origin, and 2) projection with the viewplane at $x - y + z = 1$ and center of projection at the origin. Do they commute?

127. /[2] What is the 4x4 homogeneous matrix for a parallel projection with viewplane at $x = d$, view reference point at $(d, 0, 0)$ and projection reference point at $(0, 0, 0)$?

128. /[1] Would it be ok to have the view-up vector the same as the view plane normal? Why?

129. /[2] Name 2 operations on objects that can be represented as homogeneous matrix multiplications, that cannot be represented as Cartesian matrix multiplications.

130. /[1] Why should you clip before projecting an object?

131. /[2] What is the 4x4 homogeneous matrix for a perspective projection from the origin to the projection plane at x=d?

132. /[3] Consider a unit cube with corners at (0,0,0), (1,0,0), (1,0,1), (1,1,0), (1,1,1), (0,0,1), (0,1,0), and (0,1,1) in the world coordinate system. Project it with the center of projection (0,0,3) onto the projection plane z=0. What are the projected corners, in the world coordinate system?

133. Consider the following fragment of SPHIGS code.

```
SPH_evaluateViewOrientationMatrix
   (vrp1,
    vpn1,
    vupv1,
    vo_matrix);

SPH_evaluateViewMappingMatrix
   (umin1, umax1,   vmin1, vmax1, persptype1,
    prp1,
    fplane1, bplane1,
    viewportxmin,viewportxmax, viewportymin,viewportymax, 0.0,1.0,
    vm_matrix);
```

(a) /[3] What is going on in `SPH_evaluateViewOrientationMatrix`? What are the first 3 arguments? You might sketch an illustration.

(b) /[3] What is going on in `SPH_evaluateViewMappingMatrix`? What are the first 12 arguments? You should be able to tell based on your knowledge of the function of this routine, and the names of the arguments.

134. /[2] The UVN coordinate system occurs in projections. What is it (≤ 50 words)?

135. /[1] What does the acronym PHIGS stand for?

136. /[4] Name 4 possible types of structure elements in PHIGS.

137. /[2] In PHIGS, usually the display is updated immediately when you edit a structure. Sometimes you might not want this. When?

138. Consider the following fragment of SRGP code:

```
SRGP_setColor (COLOR_WHITE);
SRGP_setFillStyle (SOLID);
```

(a) /[1] How long will this color and fill style remain set for?

(b) /[2] What sort of thing is SOLID probably? A variable name, macro, procedure with no arguments, or what? Why is this method of indicating fill style used instead of just giving a number, like graPHIGS does?

139. ☐/[1] Suppose you wish to use Phigs to model a car with 4 wheels that are identical except that they are all different colors. You want to use one wheel structure for the 4 copies. How might you make them different colors then?

Color

140. ☐/[2] In art class in primary school (in many countries), teachers say that the three primary colors are red, YELLOW, and blue. So why are the three primary colors on a CRT red, GREEN, and blue?

141. ☐/[1] One problem with color printing is that the inks are very unsaturated; they reflect quite a range of wavelengths. If our inks were closer to pure colors then we could represent more colors. However, even if such inks existed, printers would not use them, even if there were no cost, environmental, or other such reasons. Why?

142. ☐/[1] Why do printers use black ink in addition to magenta, cyan, and yellow?

143. ☐/[1] Color CRT phosphor choices are quite limited, but it's not that hard to invent new color inks. However in spite of this freedom, all the color inks used in practice (except black) are quite unsaturated, and reflect a wide range of wavelengths? Why?

144. ☐/[1] Why are color slides produced with a black&white CRT and color wheel more saturated than slides made by photographing a color CRT directly?

145. ☐/[2] How is it possible that 2 objects might look the same color under incandescent light, but look different from each other under fluorescent light?

146. ☐/[1] Suppose that you wish to calculate the apparent color resulting from mixing certain proportions of some given colors. What coordinate space should you describe the colors in, to make this easy?

147. ☐/[1] There are several other color spaces used besides the correct answer to the above question. Why doesn't everyone just use that one?

148. ☐/[2] What's the point of the odd shape of the CIE chromaticity diagram, i.e., why not simply lay out the primary colors in a circle or else a straight line?

149. ☐/[2] Specifying colors in an (R,G,B) space causes standardization problems. Why?

Visible Surface Determination

150. /[2] A problem with ray-tracing is that it takes a lot of time to test each ray against each object in the scene. Can you think how this might be speeded up so that twice as many objects doesn't take twice as long?

151. /[1] Altho the painters algorithm has the problems mentioned in class, there are times when it is useful. Name one such application.

152. /[2] Would the painters algorithm be appropriate for the horizontal floor, vertical walls, and horizontal ceiling of a maze? Why (not)?

153. /[2] Draw (another?) arrangement of polygons that the painters algorithm would not be able to handle.

154. /[2] Draw an arrangement of polygons that the painters algorithm would not be able to handle, w/o some modification, such as splitting one polygon into two parts.

155. /[1] Suppose that your frame buffer is $1024 \times 1024 \times 8$. You want to use a Z-buffer. How many bytes should the Z-buffer reasonably be? Pick one:

   (a) 128KB
   (b) 1MB
   (c) 2MB
   (d) 4MB
   (e) 8MB
   (f) 24MB

156. /[1] Suppose that you have a scene with a lot of polyhedra that you are about to process with a Z-buffer algorithm. What's the simple technique that will double the speed?

157. /[1] Suppose that you have a scene with a lot of closed polyhedra that you are about to process with a Z-buffer algorithm. What's the simple technique that will double the speed?

158. /[2] Why is the painters algorithm harder than it looks? Give a 3 polygon example to illustrate one problem.

159. /[2] What's the painter's algorithm? What's its biggest problem?

160. /[2] How can antialiasing be added to raytracing?

161. /[2] Give 2 ways that antialiasing can be added to ray tracing without increasing the time by a factor of, say, 16?

Illumination and Shading

162. /[2] Suppose you wish to make a virtual trip all around through a to-be-built building, and would like reasonable shading. Would ray-tracing or radiosity be your primary shading method? Why?

163. /[1] If you want to model a scene with lots of transparent glass balls, is ray-tracing or radiosity better?

164. /[3] What are the differences between Phong and Gouraud shading in

   (a) method,

   (b) ability to represent highlights, and

   (c) cost of implementing in hardware?

165. For Phong and Gouraud shading, which

   (a) /[1] represents highlights better?

   (b) /[1] is faster to execute, or similarly, uses less HW to implement?

166. /[2] In radiosity, if you cut each patch or facet into facets with sides 1/2 as long, how much more time will it take to calculate the intensities? I'm thinking of simple radiosity without any optimizations. Do this by thinking about how many more facets there are, how many more unknowns there are to solve for, and how much more time this will take.

167. /[1] One of ray tracing or radiosity is often combined with a Z-buffer algorithm. Which one?

168. /[3] How can a movie like Roger Rabbit combine live action and animation? Describe in under 100 words. Include a sketch.

169. /[2] What is bump mapping? Why is it useful?

170. /[3] One part of illumination equation might look like this:

$$(V \cdot R)^n \tag{8}$$

where V points to the viewer and R is the direction of a perfect reflected ray. What does n do? What sort of lighting do $n = 0$ and $n = \infty$ give?

171. /[3] Give one advantage of ray tracing relative to radiosity, and two advantages of radiosity compared to ray tracing.

172. /[2] What is morphing and how is it done?

173. /[2] Why is it easier to change your viewpoint and do a new plot when you are using radiosity than with ray-tracing?

Java

174. /[2] What is it?

175. /[2] Who is trying to use it to do what to whom?

176. /[2] Name 2 differences between Java and Javascript.

Citriniti's talk

177. /[1] What's an API?

Parsons's talk

178. /[1] What does *VRML* stand for?

179. /[1] What is one limitation of VRML?

180. /[2] Do game developers tend to use, or not tend to use, standard APIs? Why (not)?

Narayanaswami's talk

181. /[3] Name any 2 instructions that either Intel, HP, or Sun has added to their instruction set to enhance graphics.

182. /[1] Why do you want to avoid explicit test-and-branches in machine code?

183. /[2] What is saturating arithmetic? Why is it useful in graphics?

Misc

184. /[1] *Weighted area sampling* is a solution to what problem mentioned in class?

185. /[1] What is *MathML*?

186. /[2] Fill in the blanks: "__________ is the __________ language without the guns and knives."

`Total: 391 points`

End of exam

Notes

1. Grades should be ready by July. We'll email them, or you may phone Helen or me.
2. You may examine your graded exam, but not keep it (you may have a copy if you want). We will gladly correct solid, definite, unambiguous, grading errors.
3. You're welcome to ask me graphics, or other, questions, in the future (if you can find me). I might even have answers.
4. Have a good summer.
5. I might also be willing to sign off on independent reading courses on this next fall. The way that this works is that you do all the work on your own, then you write a report on what you did, and get a grade.