SMITH COLLEGE

Spring 2021 FINAL EXAM

Course Number and Section: CSC 240

Course Title: Introduction to Computer Graphics

Instructor: Nicholas Howe

Exam Format: Take-home, Open book

STUDENT NAME:
ID NUMBER:
CLASS YEAR:

ACADEMIC HONOR CODE

Students and faculty at Smith are part of an academic community defined by its commitment to scholarship, which depends on scrupulous and attentive acknowledgement of all sources of information and honest and respectful use of college resources.

Smith College expects all students to be honest and committed to the principles of academic and intellectual integrity in their preparation and submission of course work and examinations. All submitted work of any kind must be the original work of the student who must cite all the sources used in its preparation.

Student Signature: _____

EXAM INSTRUCTIONS

ALL ANSWERS SHOULD BE SCANNED AND SUBMITTED ON MOODLE. PLEASE INCLUDE THE SIGNED AND SCANNED HONOR CODE STATEMENT FROM THE FRONT PAGE OF THIS EXAM.

THIS IS AN OPEN-BOOK ASSESSMENT. YOU MAY USE THE TEXTBOOK, CLASS NOTES, AND ANY RESOURCES LINKED FROM THE COURSE WEB PAGE. YOU MAY NOT DISCUSS ANY PART OF THIS EXAM WITH ANY PERSON OTHER THAN THE INSTRUCTOR, ELECTRONICALLY OR OTHERWISE, EXCEPT AS BELOW.

Option A. *Take the exam on your own, without any assistance from others. Under this option, the exam is due on May 21 at 4:00 PM. (All deadlines for this assignment are EDT.)*

Option B. *Participate in a group exam process. Under this option, you must turn in your individual effort on the exam by Wednesday, May 19 at 4:00 PM. I will then assign you to a group, and groups will have until Thursday, May 20 at 4:00 PM to submit a single revised solution. Finally, the entire set of individuals who have chosen Option B will collaborate to produce their best combined solution by Friday, May 21 at 4:00 PM.*

Line Drawing *(12 points)*

Consider the following three algorithms.

```
function line1(p1,p2) {
    m = (p2.y-p1.y)/(p2.x-p1.x);
    b = (p2.x*p1.y-p1.x*p2.y)/(p2.x-p1.x);
    for (x = p1.x; x <= p2.x; x++) {
        y = m*x+b;
        colorPixel(x,y);
    }
}
```

```
function line2(p1,p2) {
    y = Math.round(p1.y);
    x = Math.round(p1.x);
    while (x <= p2.x) {
        colorPixel(x,y);
        F = (p1.y-p2.y)*x+(p2.x
          -p1.x)*y+p1.x*p2.y-p2.x*p1.y;
        if (F<0) {
          y++;
        }
        x++;
    }
}
```

```
function line3(p1,p2) {
    y = Math.round(p1.y);
    x = Math.round(p1.x)
    dx = p1.y-p2.y;
    dy = p2.x-p1.x;
    dd = dx+dy;
    d = dx*p1.x+dy*y+p1.x*p2.y
      -p2.x*p1.y+dx/2+dd;
    while (x <= Math.round(p2.x)) {
        colorPixel(x,y);
        if (d<0) {
          y++;
          d += dd;
        } else {
          d += dx;
        }
        x++;
    }
}
```

a.) Will these algorithms color the same pixels for a line between p_1 and p_2, assuming the slope is 0 < m < 1? If there are difference, describe them. Which is closest to the true line?

Line1 doesn't account for fractions in the point coordinates, and also doesn't compute the line at the column midpoints. The result will be shifted away from the true line. Line2 accounts for fractions by rounding, but doesn't compute the value F at the correct location so it will also be shifted. Line3 correctly colors the pixels closest to the true line.

b.) Comment on any advantages of each algorithm relative to the others (aside from the pixels colored).

Line2 does the most calculations per iteration and will run the slowest.

Transformations *(16 points)*

Match each of the following matrices with all the descriptions from the list that correspond. If none of the descriptions applies then select "No match." Assume the Y axis is oriented upwards, and we are applying each of the steps in the description to the graphics transform in the order shown. An example has been done for you.

Descriptions:

1. Rotate by 45 degrees, then scale by a factor of two in both X and Y
2. Rotate by -90 degrees, then translate by two units in the positive Y direction
3. Translate by two units in the negative X direction, the rotate by 90 degrees
4. Reflect across the line Y = X, then rotate by 45 degrees clockwise
5. Rotate by -315 degrees, then reflect across the X axis
6. Scale by a factor of 2 in X, then rotate 90 degrees
7. Rotate 90 degrees, then scale by a factor of 2 in X
8. Shear by 1 in the X direction, then rotate 90 degrees
9. Shear by 1 in the X direction, then rotate 45 degrees, then shear by 1 in the X direction, then rotate 45 degrees
10. Scale by 2 in the X dimension and 1/2 in the Y, then shear by 3 in the Y direction.
11. Rotate by -45 degrees and scale by 2 in both dimensions.
12. Reflect across the Y axis, then shear by 1 in the X direction
13. Reflect across the X axis, then shear by 1 in the Y direction
14. Translate 3 units in the positive X direction, rotate by 180 degrees, then translate 3 units in the negative X direction

a.) $\begin{bmatrix} 1.4 & -1.4 & 0 \\ 1.4 & 1.4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ← *This is description #1*

b.) $\begin{bmatrix} 2 & 0 & 0 \\ 1.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *#9, #10*

c.) $\begin{bmatrix} 1 & 0 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *#13*

d.) $\begin{bmatrix} 0 & -2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *#6*

e.) $\begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *No match*

f.) $\begin{bmatrix} 0.7 & 0.7 & 0 \\ 0.7 & -0.7 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *#4, #5*

g.) $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *No match*

h.) $\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *No match*

i.) $\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ *#3*

Line Clipping *(16 points)*

Suppose that we have a 100x100 viewport, and want to draw a line segment from (-50,-80) to (150,120). If the line is clipped using the algorithm we studied in class, indicate for each time the recursive function is called (i) which case is triggered, and (ii) new clipped endpoints computed, if any.

Step 1. Endpoint codes 1001 and 0110. Case 3a. New endpoint p1 = (30,0).

Step 2. Endpoint codes 0000 and 0110. Case 3f. New endpoint p2 = (130,100).

Step 3. Endpoint codes 0000 and 0010. Case 3g. New endpoint p2 = (100,70).

Step 4. Endpoint codes 0000 and 0000. Case 1. Draw the line!

Rendering Pipeline *(12 points)*

Suppose we create a camera using the following Three.js commands:

```
camera = new THREE.PerspectiveCamera(120, 2, 10, 100);
camera.position.set(0,0,5);
```

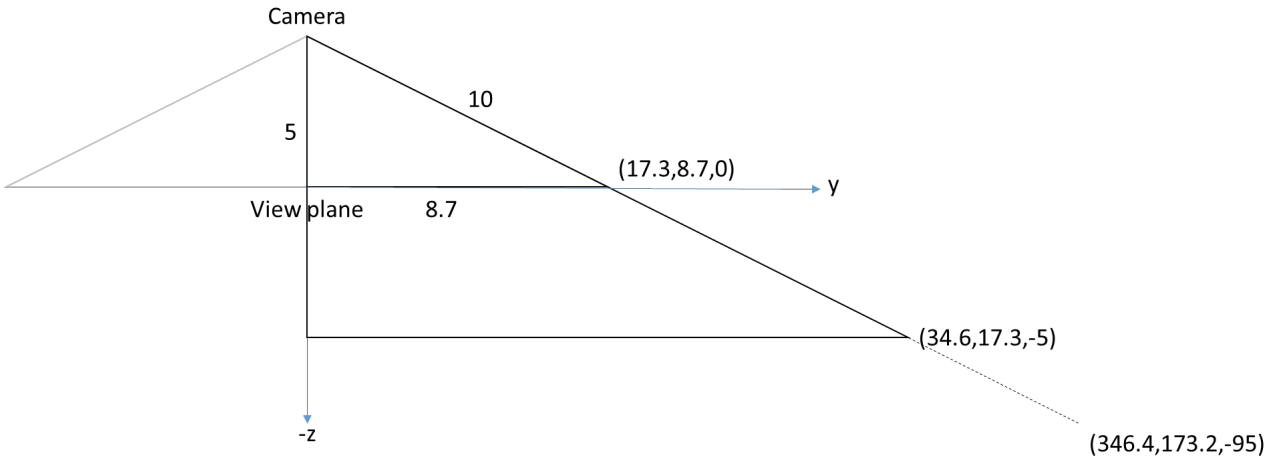
a.) The camera's focal length is 5. What is the perspective projection matrix? $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \end{bmatrix}$

b.) If the canvas is 200 pixels wide, how high is it? *100 pixels*

c.) Compute 3D coordinates for the eight corners of the view frustum. (Round to the nearest tenth.)

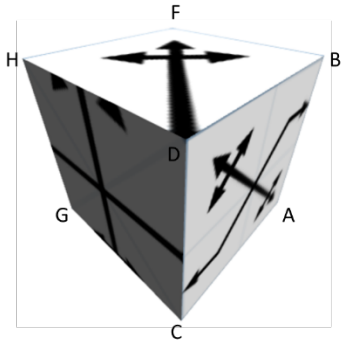
The near plane has z = -5, and the far plane has z = -95. Documentation shows that FOV measures the vertical angle (y).

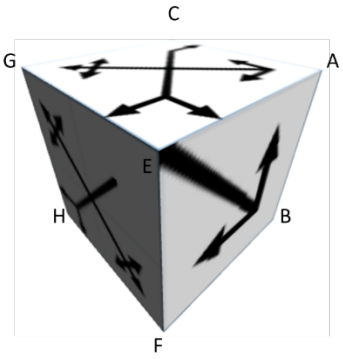
Trigonometry gives the coordinates as $(\pm 34.6, \pm 17.3, -5)$ *and* $(\pm 346.4, \pm 173.2, -95)$

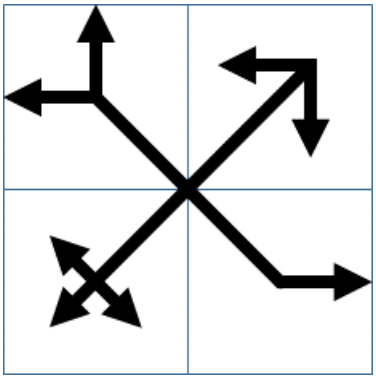


Texture Mapping *(16 points)*

Suppose that we want to map a cube as shown in the images below, using the texture image at right. What UV coordinates should we use for the vertices of the specified polygons? (Hint: all the UV coordinates are multiples of 0.5.)







a.) ABC *(0,0), (0,1), (1,0)*
b.) BDC *(1,0), (0,0), (0,1)*
c.) GEC *(0,0), (0,1), (1,0)*
d.) CAE *(0,1), (1,1), (1,0) or (1,0), (1,1), (0,1)*
e.) EAF *(0.5,0.5), (1,0.5), (0.5,1) or (0.5,0.5), (0.5,1) , (1,0.5)*
f.) FBA *(0.5,1), (1,1), (1,0.5) or (1,0.5), (1,1), (0.5,1)*
g.) GHC *(0,0.5), (0.5,1), (0.5,0)*
h.) CDH *(0.5,0), (1,0.5), (0.5,1)*

Shading *(12 points)*

A surface rotates on an axis at point F. Material A has 100% diffuse reflectance and 0% specular reflectance, while material B has 0% diffuse reflectance and 100% specular reflectance, with $\alpha = 10$. Both the materials and the light source are white. The light is a directional light slanting downwards at a 20 degree angle away from the camera, as shown. Use this information to answer the questions below.

a.) For what angle θ_A will the shading at point F show the brightest shading with material A? $20°$

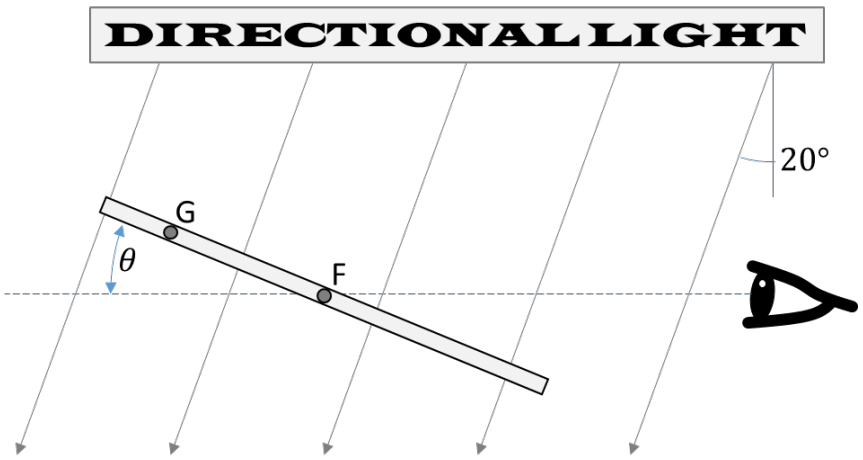
b.) For what angle θ_B will the shading at point F show the brightest shading with material B? $55°$

c.) How will the shading at point G compare to that at point F for angle θ_A and material A? (e.g., is it brighter, darker, or the same) *Same*

d.) How will the shading at point G compare to that at point F for angle θ_B and material B? *Darker*

e.) Suppose $\theta = 55°$ with material A gives shading intensity s at point F. Compute the angle θ' that would result in shading intensity $s/2$ at point F. *cos(55-20) = 0.82.* $20 + \cos^{-1} 0.91/2 = 85.8°$

f.) Suppose $\theta = 55°$ with material B gives shading intensity s at point F. Compute the angle θ' that would result in shading intensity $s/2$ at point F. *At 55 we have a specular peak. We want* $(R \cdot V)^{\alpha} = 0.5$. *V is fixed. Since* $\alpha = 10$ *we take the 10th root of both sides.* $R \cdot V = 0.933$, *which mean that the angle between R and V is 21.1 degrees. R changes twice as fast as* θ, *so* $\theta' = \theta \pm 10.5° = 44.5°$ *or* $65.5°$



Ray Tracing *(16 points)*

While rendering an image, a computed ray passes through the air and strikes a semi-transparent planar surface. The ray equation is $R = (0,0,0) + (1,0,0)t$, and the surface normal vector of the surface is$(-0.87,0.50,0)$. If the point of contact is (10,0,0), compute the following:

a.) The equation for the reflected light ray.

$$R' = (10,0,0) + (-0.5,0.87,0)t$$

b.) The equation for the light ray transmitted through the material, if its index of refraction is 1.5.

$$\sin\theta_2 = \frac{n_1}{n_2}\sin\theta_1 = \frac{1}{1.5}\sin 30° = 0.33 \Rightarrow \theta_2 = 19.5°$$

Because the surface normal is 30 degrees below the horizontal, the transmitted vector is (30-19.5) = 10.5 degrees below horizontal. The equation is thus

$$R' = (10,0,0) + (\cos(10.5), -\sin(10.5), 0)t = (10,0,0) + (0.98, -0.18, 0)t$$

c.) If there is a light source at (0,25,0), what is the diffuse illumination at the point of contact as a fraction of the maximum strength of the light?

The light is at an angle of $\tan^{-1}\frac{25}{10} = 68.2$ *above the horizontal, which is 38.2 degrees off the normal.* $\cos 38.2 = 0.79$

d.) Suppose there is a sphere of radius 2 centered at (4,10,0). Determine whether it will block the light coming from the source mentioned in part c and provide the justification for your answer.

We will solve for the intersection point.

Ray equation is $\vec{p} = \vec{R}_0 + t\vec{R}_d = (10,0,0) + t(-10,25,0)$

$$\vec{R}_0' = \vec{R}_0 - \vec{s} = (10,0,0) - (4,10,0) = (6,-10,0)$$

$$a = \vec{R}_d \cdot \vec{R}_d = 725$$

$$b = 2\vec{R}_0' \cdot \vec{R}_d = 2(-60 - 250 + 0) = -620$$

$$c = \vec{R}_0' \cdot \vec{R}_0' - r^2 = (36 + 100 + 0) - 4 = 132$$

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{620 \pm \sqrt{384400 - 382800}}{1450} = \frac{620 \pm 40}{1450}$$

Because there are real solutions, the sphere does block the light.