111-000

CSCI 5607, Exam #2

Noah Hendrickson

hend0500@umn.edu : ID: 5320241

Problem #1: $r=3$, $c=(2,2,10)$ orig $=(0,0,0)$ int_p $=(1,4,8)$

$\eta_i = 1.0$ $\eta_t = 1.5$

snell's law: $\frac{\sin\theta_i}{\sin\theta_t} = \frac{\eta_{t\lambda}}{\eta_{i\lambda}}$

a) find angle of incidence, θ_i: $\theta_i = \text{acos}(N \cdot I)$

N is normal at int_p, I is negative ray direction

$N = \text{int\_p} - c = (1,4,8)-(2,2,10) = (-1,2,-2) = (-\frac{1}{3}, \frac{2}{3}, -\frac{2}{3})$

$I = \text{orig} - \text{int\_p} = (0,0,0)-(1,4,8) = (-1,-4,-8) = (-\frac{1}{9}, -\frac{4}{9}, -\frac{8}{9})$

$\theta_i = \text{acos}(N \cdot I) = \text{acos}(\frac{1}{27} - \frac{8}{27} + \frac{16}{27}) = \text{acos}(\frac{9}{27}) = \text{acos}(\frac{1}{3}) = 70.53^\circ$

b) find angle of reflection, θ_r: $\theta_i = \theta_r$, thus $\theta_r = 70.53^\circ$

c) find direction of reflected ray: $\vec{R} = \vec{A} + \vec{S}$

$A = (aN)$, $a = N \cdot I = \frac{1}{3}$, $A = \frac{1}{3}N = (-\frac{1}{9}, \frac{2}{9}, -\frac{2}{9})$

$S = (aN - I) = (-\frac{1}{9}, \frac{2}{9}, -\frac{2}{9}) - (-\frac{1}{9}, -\frac{4}{9}, -\frac{8}{9}) = (0, \frac{2}{3}, \frac{2}{3})$

$R = (-\frac{1}{9}, \frac{2}{9}, -\frac{2}{9}) + (0, \frac{6}{9}, \frac{6}{9}) = (-\frac{1}{9}, \frac{8}{9}, \frac{4}{9})$

d) find angle of transmission, θ_t: $\theta_t = \text{asin}(\frac{\eta_i}{\eta_t}\sin(\theta_i))$

$\theta_t = \text{asin}(\frac{1.0}{1.5}\sin(70.53)) = \text{asin}(0.6285441) = 38.94^\circ$

e) find direction of transmitted ray: $T = \cos\theta_t(-N) + \sin\theta_t(\frac{S}{\sin\theta_i})$

$T = \cos(38.94^\circ)(\frac{1}{3}, -\frac{2}{3}, \frac{2}{3}) + \sin(38.94^\circ)\left(\frac{(0, \frac{2}{3}, \frac{2}{3})}{\sin(70.53)}\right)$

$= 0.7778(\frac{1}{3}, -\frac{2}{3}, \frac{2}{3}) + 0.6285((0, \frac{2}{3}, \frac{2}{3})/0.9428)$

$= (0.2593, -0.5185, 0.5185) + (0, 0.4444, 0.4444)$

$= (0.2593, -0.0741, 0.9629)$

Problem #2:

The sequence is as such:

→ rotate +90° around y to align with x

→ apply matrix to align with view direction

→ apply transformation matrix.

$$V_f = T R V_0$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

~~R requires a bit more work~~

normalize w and d: $c = \left(\frac{2}{3}, \frac{1}{3}, -\frac{2}{3}\right)$ $b = \left(-\frac{2}{3}, \frac{2}{3}, -\frac{1}{3}\right)$

w' and d' are orthogonal. Cross product to find v:

$a = c' \times b' = (1/3, 2/3, 2/3)$

$$R = \begin{bmatrix} a_x, & a_y, & a_z, & 0 \\ b_x, & b_y, & b_z, & 0 \\ c_x, & c_y, & c_z, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix}^T = \begin{bmatrix} -2/3 & 1/3 & 2/3 & 0 \\ 2/3 & 2/3 & 1/3 & 0 \\ -1/3 & 2/3 & -2/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This sequence will rotate the vertices into proper location

Problem #3:

a) rotate around z axis: $\theta = 23.5°$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \boxed{\begin{bmatrix} 0.917 & -0.399 & 0 & 0 \\ 0.399 & 0.917 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

b) given an angle θ, the series is:

$M' = R_2 R_1 M$ where

$$R_1 = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } R_2 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c) extra credit: orbiting

the series of transformations would be

$M' = R_y T R_z$ where $R_y = R_2$ from part b, $R_z = R_1$ from part b, and

$$T = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3, 2, 4
4, 1, 3

5, 0, 2

6, -1, 1

-1 1 1 : -1 1 1
1 0 1 : 1 0 1

Problem #4: Camera/View

eye $= (0,0,0)$ $-n = (0,0,-1)$ up $= (0,1,0)$

a) when eye $= (2,3,5)$, dir $= (1,-1,-1)$, and up $= (0,1,0)$

$n' = (-1,1,1)$ $u' = up \times n' = (1,0,1)$ $v' = n' \times u' = (1,2,-1)$

$n = \left(-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right)$ $u = \left(\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}\right)$ $v = \left(\frac{\sqrt{6}}{6}, \frac{2\sqrt{6}}{6}, -\frac{\sqrt{6}}{6}\right)$

$d = (\text{dot}(eye, u), \text{dot}(eye, v), \text{dot}(eye, n))$

$= \left(\frac{7\sqrt{2}}{2}, \frac{\sqrt{6}}{2}, 2\sqrt{3}\right)$

$$V = \begin{bmatrix} u_x & u_y & u_z & d_x \\ v_x & v_y & v_z & d_y \\ n_x & n_y & n_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 & \frac{7\sqrt{2}}{2} \\ \sqrt{6}/6 & 2\sqrt{6}/6 & -\sqrt{6}/6 & \sqrt{6}/2 \\ -\sqrt{3}/3 & \sqrt{3}/3 & \sqrt{3}/3 & 2\sqrt{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b) the vectors u, v, and n will stay the same and so will their corresponding components in V. The vector d and its components in V will change. d_x will stay the same, d_y will also stay the same. d_z will gradually decrease.

c) n will change to match the negative of the view dir, thus, u will also change because its dependent on both up and n. v ~~will change because it is dependent on u and n~~. d will also change because it is dependent on all three. n will spin clockwise with the view dir, so will u as it is always 90° to n. v won't change as the plane spanned by u and n doesn't change (still should though to reorthogonalize)

d) n and v will change and thus so will d. u will stay same because plane spanned by n and v does not change. n and v both rotate the same way as view dir with n 180° of it and v 90° more

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix} \vdots \begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix}$$

$(0, 0, -1)$

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & -1 \end{matrix} \quad \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & -1 \end{matrix}$$

$(0, 1, 0)$

Problem #5:

a) blue :)

b) need to rotate −90 degrees (x to z) around y

$$M = \begin{bmatrix} \cos(-90) & 0 & \sin(-90) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-90) & 0 & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

c) $eye = (5, 0, -5)$ $view = (-1, 0, 0)$ $up = (0, 1, 0)$

$n = (1, 0, 0)$ $u = up \times n = (0, 0, -1)$ $v = n \times u = (0, 1, 0)$

$d = (dot(eye, u), dot(eye, v), dot(eye, n)) = (5, 0, 5)$

$$V = \begin{bmatrix} u_x & u_y & u_z & d_x \\ v_x & v_y & v_z & d_y \\ n_x & n_y & n_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & 0 & -1 & 5 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

??? 8 vertices??? not 6????

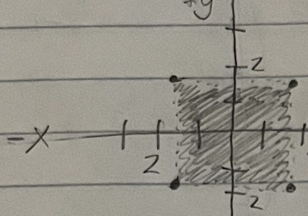
?? $V_6=(\sqrt{3},-\sqrt{3},-4\sqrt{3})$ $V_7=(-\sqrt{3},-\sqrt{3},-4\sqrt{3})$

Problem #6: $V_0=(\sqrt{3},\sqrt{3},-2\sqrt{3})$ $V_1=(-\sqrt{3},\sqrt{3},-2\sqrt{3})$ $V_2=(\sqrt{3},-\sqrt{3},-2\sqrt{3})$
$V_3=(-\sqrt{3},-\sqrt{3},-2\sqrt{3})$ $V_4=(\sqrt{3},\sqrt{3},-4\sqrt{3})$ $V_5=(-\sqrt{3},\sqrt{3},-4\sqrt{3})$

a)

i) $V_0=(\sqrt{3},\sqrt{3},0)$ $V_1=(-\sqrt{3},\sqrt{3},0)$ $V_2=(\sqrt{3},-\sqrt{3},0)$ $V_3=(-\sqrt{3},-\sqrt{3},0)$
$V_4=(\sqrt{3},\sqrt{3},-4\sqrt{3})$ $V_5=(-\sqrt{3},\sqrt{3},0)$ $V_6(\sqrt{3},-\sqrt{3},0)$ $V_7=(-\sqrt{3},-\sqrt{3},0)$

ii)



dark gray face because that was the face with the largest z coordinate

iii) only matrix needed for this is

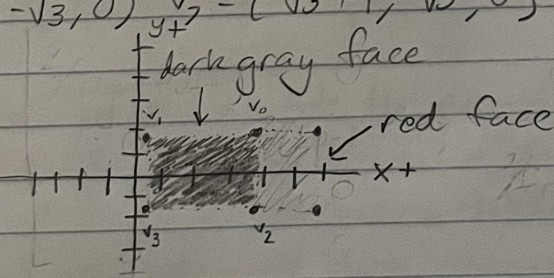
$$M=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}$$

b) $d=(1,0,\sqrt{3})$

i) $x_n = x + (-z)\frac{1}{\sqrt{3}}$ $y_n = y + (-z)\frac{0}{\sqrt{3}} = y$ $z_n = 0$

$V_0=(\sqrt{3}+2,\sqrt{3},0)$ $V_1=(-\sqrt{3}+2,\sqrt{3},0)$ $V_2=(\sqrt{3}+2,-\sqrt{3},0)$
$V_3=(-\sqrt{3}+2,-\sqrt{3},0)$ $V_4=(\sqrt{3}+4,\sqrt{3},0)$ $V_5=(-\sqrt{3}+4,\sqrt{3},0)$
$V_6=(\sqrt{3}+4,-\sqrt{3},0)$ $V_7=(-\sqrt{3}+4,-\sqrt{3},0)$

ii)



iii) two matrices are needed:

Obl = MS where

$$S=\begin{bmatrix}1&0&1/\sqrt{3}&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \text{ and } M=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}$$

S shears in the direction d and M is basic orthographic projection.

cube$_1$ vertices: $c_0 = (0.5, 0.5, -4)$ $c_1 = (-0.5, 0.5, -4)$
$c_2 = (0.5, -0.5, -4)$ $c_3 = (-0.5, -0.5, -4)$

cube$_2$ vertices: $b_0 = (1, 1.5, -7)$ $b_1 = (-1, 1.5, -7)$
$b_2 = (1, -0.5, -7)$ $b_3 = (-1, -0.5, -7)$

Problem #7:

a) $\theta_v = 90°$, near $= 1$ far $= 10$ aspect $= 1$

$$ProjM = \begin{bmatrix} \cot(\frac{\theta_v}{2}) & 0 & 0 & 0 \\ 0 & \cot(\frac{\theta_v}{2}) & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & \frac{-2\cdot far\cdot near}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -11/9 & -20/9 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Camera matrix is identity

$c'_0 = (0.5, 0.5, 2.6666, 4) = (0.125, 0.125, 0.6666, 1)$

$c'_1 = (-0.5, 0.5, 2.6666, 4) = (-0.125, 0.125, 0.666, 1)$

$c'_2 = (0.5, -0.5, 2.666, 4) = (0.125, -0.125, 0.666, 1)$

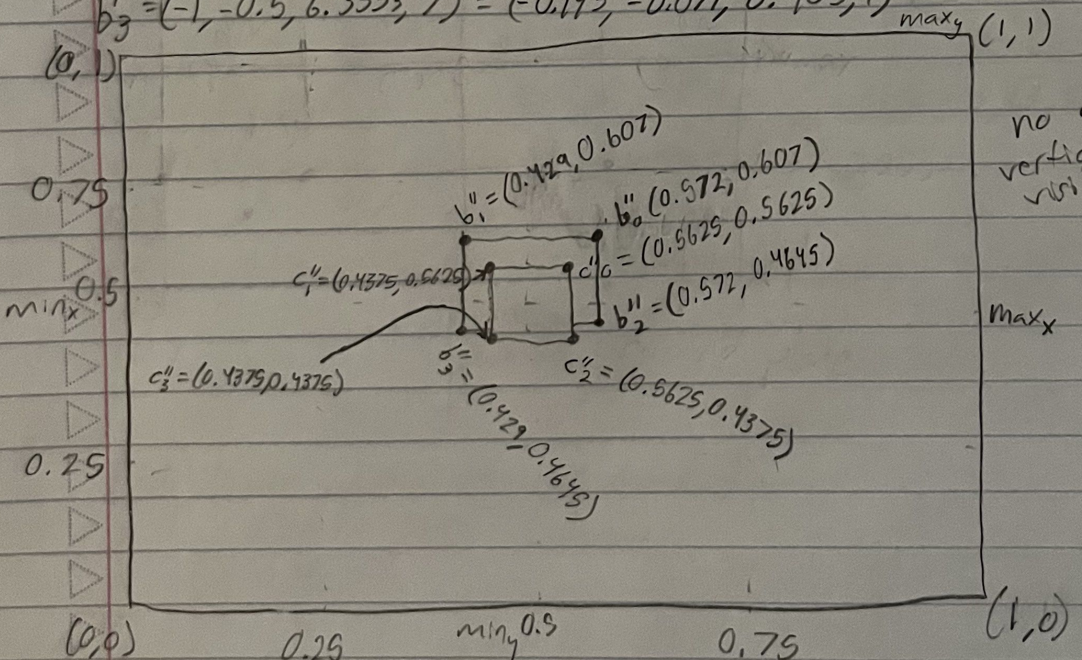
$c'_3 = (-0.5, -0.5, 2.6666, 4) = (-0.125, -0.125, 0.666, 1)$

$b'_0 = (1, 1.5, 6.3333, 7) = (0.143, 0.214, 0.905, 1)$

$b'_1 = (-1, 1.5, 6.3333, 7) = (-0.143, 0.214, 0.905, 1)$

$b'_2 = (1, -0.5, 6.3333, 7) = (0.143, -0.071, 0.905, 1)$

$b'_3 = (-1, -0.5, 6.3333, 7) = (-0.143, -0.071, 0.905, 1)$



no other vertices are visible

camera eye: $(0, 0, -2)$

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ProjM = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -11/9 & -20/9 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

b) Camera matrix no longer just negates.

ProjM stays the same

Order is $v_0 = ProjM \cdot V \cdot v_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -11/9 & -14/3 \\ 0 & 0 & -1 & -2 \end{bmatrix} v_0$

$c_0' = (0.5, 0.5, 0.2222, 2) = (0.25, 0.25, 0.111111, 1) = (0.625, 0.625)$

$c_1' = (-0.5, 0.5, 0.2222, 2) = (-0.25, 0.25, 0.111111, 1) = (0.375, 0.625)$

$c_2' = (0.5, -0.5, 0.222, 2) = (0.25, -0.25, 0.11111, 1) = (0.625, 0.375)$

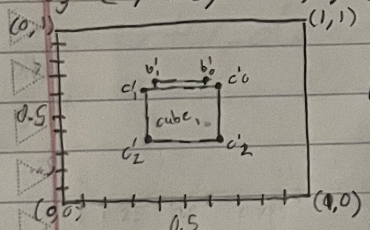
$c_3' = (-0.5, -0.5, 0.222, 2) = (-0.25, -0.25, 0.11111, 1) = (0.375, 0.375)$

$b_0' = (1, 1.5, 3.8888, 5) = (0.2, 0.3, 0.7777, 1) = (0.6, 0.65)$

$b_1' = (-1, 1.5, 3.8888, 5) = (-0.2, 0.3, 0.7777, 1) = (0.4, 0.65)$

$b_2' = (1, -0.5, 3.8888, 5) = (0.2, -0.1, 0.7777, 1) = (0.6, 0.45)$

$b_3' = (-1, -0.5, 3.8888, 5) = (-0.2, -0.1, 0.7777, 1) = (0.4, 0.45)$



no other vertices are visible

c) projection matrix changes in cot values because θ_v is changing. Will bring cubes closer

$$P = \begin{bmatrix} 2.41 & 0 & 0 & 0 \\ 0 & 2.41 & 0 & 0 \\ 0 & 0 & -11/9 & -20/9 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$c_0' = (1.205, 1.205, 0.2222, 2) = (0.6025, 0.6025, 0.1111, 1) = (0.80, 0.80)$

$c_1' = (-1.205, 1.205, 0.2222, 2) = (-0.6025, 0.6025, 0.1111, 1) = (0.2, 0.8)$

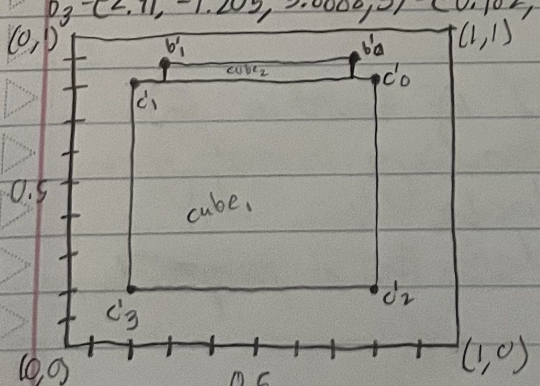
$c_2' = (0.8, 0.2)$, $c_3' = (0.2, 0.2)$

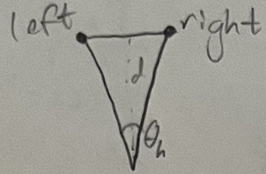
$b_0' = (2.41, 3.615, 3.8888, 5) = (0.482, 0.723, 0.7777, 1) = (0.741, 0.862)$

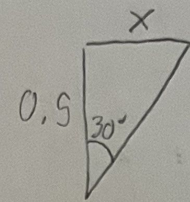
$b_1' = (-2.41, 3.615, 3.8888, 5) = (-0.482, 0.723, 0.7777, 1) = (0.259, 0.862)$

$b_2' = (2.41, -1.205, 3.8888, 5) = (0.482, -0.241, 0.7777, 1) = (0.741, 0.3795)$

$b_3' = (-2.41, -1.205, 3.8888, 5) = (-0.482, -0.241, 0.7777, 1) = (0.259, 0.3795)$







$x = 0.2887$

$\tan(30) = \frac{x}{0.5}$

$0.5\tan(30) = x$

Problem #8:

a) $d_{near} = 0.5$ hfov $= 60°$ vfov $= 60°$

left = bottom = -0.2887 right = top = 0.2887

$$P = \begin{bmatrix} 1.73 & 0 & 0 & 0 \\ 0 & 1.73 & 0 & 0 \\ 0 & 0 & -1.05 & -1.03 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

b) top = 2·top = 0.5774 bottom = -0.5774

$$P = \begin{bmatrix} 1.73 & 0 & 0 & 0 \\ 0 & 0.866 & 0 & 0 \\ 0 & 0 & -1.05 & -1.03 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

c) $\theta_h = 60°$ (didnt change)

$\theta_v = 2\mathrm{atan}(0.5774/0.5) = 2 \cdot 49.1 = 98.2°$

d) $$\begin{bmatrix} 1.73 & 0 & 0 & 0 \\ 0 & 1.73 & 0 & 0 \\ 0 & 0 & -1.05 & -1.03 \\ 0 & 0 & -1 & 0 \end{bmatrix} = P_{alt}$$

e) $$\begin{bmatrix} 0.866 & 0 & 0 & 0 \\ 0 & 1.73 & 0 & 0 \\ 0 & 0 & -1.05 & -1.03 \\ 0 & 0 & -1 & 0 \end{bmatrix} = P_{alt}$$ aspect ratio is $\frac{1}{2}$

f) $\theta_v = 60°$, stays same. $\theta_h = 98.2°$ as found in c)

The contents change by expanding what you can see horizontally but it looks like its zoomed out? Idk how to describe it properly, like its distorted.

g) $P_{alt} = \begin{bmatrix} 2.46 & 0 & 0 & 0 \\ 0 & 1.73 & 0 & 0 \\ 0 & 0 & -1.05 & -1.03 \\ 0 & 0 & -1 & 0 \end{bmatrix}$ aspect ratio is 2

h) θ_v stays at 60° content in the screen zooms in because of the decreas in horizontal fov
to find θ_h:
$height_{1/2} = \tan(\frac{\theta_v}{2}) \cdot 0.5 = 0.2886$ and because
aspect ratio $= \frac{h}{w} = \frac{2}{1} \rightarrow width = h/2 = 0.1443$
$\theta_h = 2\operatorname{atan}(\frac{1.443}{0.5}) = 32.1°$

i) you would need to adjust the field of view along with the aspect ratio. i.e. if the window is made taller, increase θ_v and if its made shorter, decrease θ_v. or alternatively just dont use P_{alt} :)

Problem #9: $p_0=(3,3)$ $p_1=(9,5)$ $p_2=(11,11)$

a) $e_1=p_1-p_0$ $e_2=p_2-p_1$ $e_3=p_0-p_2$

$n_1=(-(p_{1y}-p_{0y}),(p_{1x}-p_{0x}))$

$n_2=(-(p_{2y}-p_{1y}),(p_{2x}-p_{1x}))$

$n_3=(-(p_{0y}-p_{2y}),(p_{0x}-p_{2x}))$

$e_{(p_n-p_m)}(x,y)=(-(p_{ny}-p_{my}),(p_{nx}-p_{mx}))\cdot((x,y)-(p_{mx},p_{my}))$

$=-(p_{ny}-p_{my})\cdot x+(p_{nx}-p_{mx})y+[(p_{ny}-p_{my})\cdot p_{mx}-(p_{nx}-p_{mx})\cdot p_{my}]$

a point (x,y) is in triangle defined by p_0,p_1,p_2 if, $\forall$ e defined by those vertices, $e(x,y)\geq 0$:

$e_{p_1-p_0}(x,y)=-2x+6y+(6-18)=-2x+6y-12\geq 0$ and

$e_{p_2-p_1}(x,y)=-6x+2y+(54-10)=-6x+2y+44\geq 0$ and

$e_{p_0-p_2}(x,y)=8x-8y+(-88+88)=8x-8y\geq 0$

b) $e_{p_1-p_0}(6,4)=-2(6)+6(4)-12=0\geq 0$ True

$e_{p_2-p_0}(6,4)=-6(6)+2(4)+44=16\geq 0$ True

$e_{p_0-p_2}(6,4)=8(6)-8(4)=16\geq 0$ True

point $(6,4)$ lies inside

$e_{p_1-p_0}(7,7)=-2(7)+6(7)-12=16\geq 0$ True

$e_{p_2-p_1}(7,7)=-6(7)+2(7)+44=16\geq 0$ True

$e_{p_0-p_2}(7,7)=8(7)-8(7)=0\geq 0$ True

point $(7,7)$ lies inside

$e_{p_1-p_0}(10,8)=-2(10)+6(8)-12=16\geq 0$ True

$e_{p_2-p_1}(10,8)=-6(10)+2(8)+44=0\geq 0$ True

$e_{p_0-p_2}(10,8)=8(10)-8(8)=16\geq 0$ True

point $(10,8)$ lies inside

Problem #10:

for help with this, bounding box bottom left is (0.5,0.5)
and top right is (7.5,7.5) (each box is 1 unit)
$V_0=(1.5,8.5)$ $V_1=(1.5,1.5)$ $V_2=(3.5,6.5)$ $V_3=(3.5,-0.5)$
$V_4=(6.5,7.5)$ and $V_5=(6.5,2.5)$
because V_0 and V_3 lie outside clip such that
V_0 is split into $V_{00}=(1.5,7.5)$ and $V_{01}=(2.5,7.5)$
V_3 is split into $V_{30}=(2.5,0.5)$ and $V_{31}=(4.5,0.5)$
In the new strip, 6 triangles will be encoded:
$T_1=(V_{00},V_1,V_{01})$ $T_2=(V_{01},V_1,V_{30})$ $T_3=(V_{01},V_{30},V_2)$
$T_4=(V_2,V_{30},V_{31})$ $T_5=(V_2,V_{31},V_4)$ $T_6=(V_4,V_{31},V_5)$
image shown on next page of pdf ☺

1 $V_{00} = (1.5, 7.5)$

Label

2 $V_{01} = (2.5, 7.5)$

Label

3 $V_1 = (1.5, 1.5)$

Label

4 $V_2 = (3.5, 6.5)$

Label

5 $V_{30} = (2.5, 0.5)$

Label

6 $V_{31} = (4.5, 0.5)$

Label

7 $V_4 = (6.5, 7.5)$

Label

8 $V_5 = (6.5, 2.5)$

Label

9 $B_0 = (0.5, 0.5)$

Label

10 $B_1 = (0.5, 7.5)$

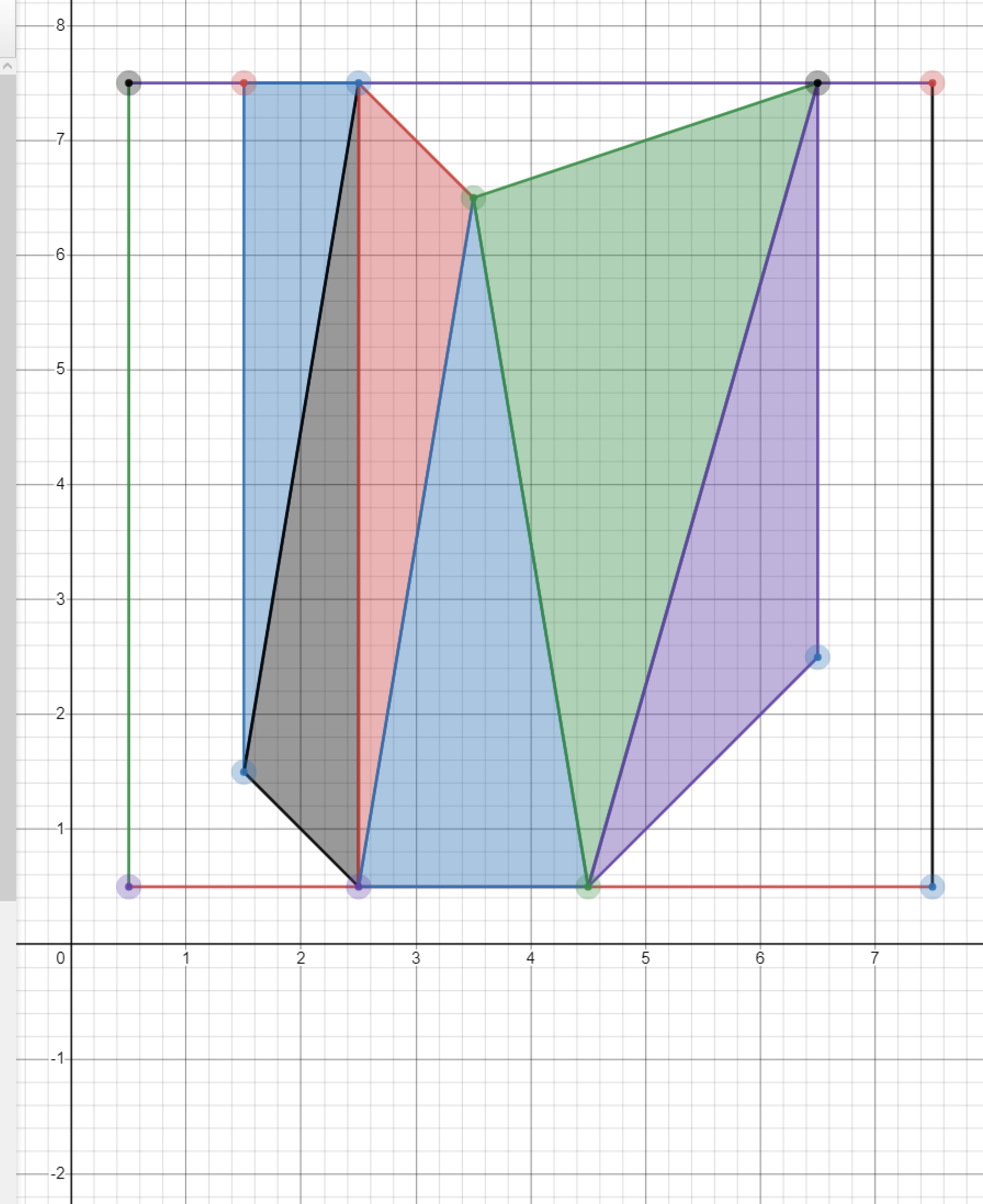
Label

11 $B_2 = (7.5, 7.5)$

Label

12 $B_3 = (7.5, 0.5)$

Label



actual first is
vertex shader application

Problem #11: Scan Conversion Steps

→ first is vertex operations:

Apply the model transformation matrix, then the camera matrix, then perspective matrix to get the vertices into normalized clipping coordinates

→ Once all vertices have been transformed, apply clipping algorithm and create any new necessary vertices to make the clipped polygons.

Vertices then undergo the viewport transformation which maps (x,y) values to frame buffer coordinates and z values to depth buffer values

→ after is Rasterization

Rasterization rasterizes triangles into fragments such that each fragment maps to exactly one of the pixels enclosed by triangle. This is done by determing which pixels lie inside the triangle with the edge equations.

Anti aliasing can then be done by splitting pixels into sub pixels and coloring based on how much of the pixel is in/out of the triangle.

Information about the triangle is then interpolated across it, such as color, normals, etc...

Other operations such as hidden surface removal and depth testing are also done.

Additionally, if there is a fragment shader attached, that is applied after the fragments are determined.

→ To the Screen

Once all is applied, the frame buffer is shown to the screen

Hidden surface removal used the z buffer to get rid of blocked triangles.

Problem #12:

Ray Tracing and Scan conversion are different in that raytracing is built on rays and scan conversion is built on vertices. To achieve perspective, ray tracing shoots rays in different directions whereas scan conversion uses a perspective transformation matrix and a view matrix to transform vertex positions. Scan conversion requires triangles or quads whereas raytracing can render anything natively as long as it had a parametric equation. Scan conversion is much faster than raytracing. When it comes to what each can do, raytracing is far ahead of scan conversion, being able to render incredibly realistic lighting, reflections, and refractions. This all comes at a cost that scan conversion doesn't have to deal with though.

Ray tracing can do reflections because you can follow the rays through the scene easily and see what they interact with. With scan conversion, you only have the depth information. The same reasoning for refraction and lighting. However, scan conversion has some nice things too like shaders that can be easily applied and clipping of vertices. In order to do something like clipping when ray tracing, you need to make use of hierarchical spatial data structures. When it comes to the shader thing, I dont think you could do that as efficiently? For example, you wouldn't be able ~~to~~ to do something like a particle simulation nearly as fast when ray tracing as you can with scan conversion. Both have their upsides and downsides, but personally I think raytracing is far, far cooler :)