Computer Graphics

T-511-TGRA

FINAL EXAM

Teacher: Kári Halldórsson

Date: 15. November 2019

Time: 9:00 – 12:00

Permitted exam materials: non-programmable calculator

Name: LAUSNIR

ID.: ______________________

Answers can be given in English or Icelandic.

1. **(10%) Blending**
   Describe blending in OpenGL.
   Describe its desired effect and give an example.
   Where in the OpenGL pipeline does it occur?
   What values are used and how?
   How are these values set and changed?
   Describe the calculations or give an example of them.

Mixes colors, new with something in frame buffer

2 e.g. transparency.

2 Happens in rasterization after frag shader!

Alpha value, 4th color value, is used, and

2 the color that is already in frame buffer[x,y].

instead of

frame_buffer[x,y] = frag_color

4 we do

frame_buffer[x,y] = src_factor * frag_color + dst_factor * frame_buffer[x,y]

where the factors can be

GL_SRC_ALPHA,

GL_SRC_COLOR,

GL_ONE, GL_ONE_MINUS_SRC_ALPHA,

GL_ONE_MINUS_DST_COLOR,

etc., etc.

2. **(10%) Shaders and lighting**
Describe the difference between per-vertex lighting and per-fragment/per-pixel lighting.
In each case bear the following questions in mind:
What are the advantages and drawbacks of the method?
What calculations happen where, and what values are set to the final result?
How is the data processed between different parts of the calculations?

per-vertex faster

2 per-fragment prettier

2 - lighting calculated for each vertex in model vs. for each pixel on screen

per vertex: all lighting calculations

2 - happen in vertex shader,

varying var is just the color which is interpolated over pixels

per fragment: lighting is set up in vertex shader but varying vars

2 - are vectors that are interpolated over the fragments and used in the fragment shader where proper lighting

2 - calculations are finished, for each pixel.

3. (15%) OpenGL pipeline
Briefly describe each part of the OpenGL graphics pipeline and in which order they are run. What is the state of the vertex data at each point in the pipeline, from first input to final output, what data does each module use and what is each module's purpose?

When describing each module bear in mind ***1)*** *it's purpose,* ***2)*** *the data "before" and* ***3)*** *the data "after"*, but you don't need to describe the algorithm or calculation of each one. As for the shaders, describe how/when they are run, what their main purpose is, their "input" and "output", how this input is prepared and how the output is used afterwards.

Next page is blank for easier organization of the answer.

Feel free to use diagrams to aid your answer.

local coords: 3D continuous

Vertex shader

Calculates geometry of vertex, using matrices

in vertex: local

out vertex: clip + varying + depth

2

2

clipping

clip coords / 2D continuous

1

clips what is outside window

2

W-2-V

re scales from window to screen

2

Rasterization

interpolates values over pixels and runs fragment shader

2

FRAG shader
in: varying
out: frag color

2

puts color in frame buffer.

2

pixel colors

4. **(25%) Matrices and transformations**

a) (10%)
A camera is set up to be positioned in (14, 23, 17)
looking at the point (-1,-1,-1).
It has an up vector (0,1,0).
Set up the values in a matrix that represents this position and orientation of a camera.
Which matrix in your shader should be set to these values?

$n = eye - center = (15, 24, 18)$

$u = up \times n = (18, 0, -15)$

$v = n \times u = (-360, 549, -432)$

Normalize!

$|u| = \sqrt{549}$ $\quad |v| = \sqrt{617625}$ $\quad |n| = \sqrt{1125} = 15\sqrt{5}$

$-eye \circ u = \frac{3}{\sqrt{549}} = \frac{1}{\sqrt{61}} = 0{,}128$

$-eye \circ v = \frac{-243}{\sqrt{617625}} = -0{,}309$

$-eye \circ n = \frac{-1068}{\sqrt{1125}} = -31{,}8$

OK if this is big. The position affects it as well as the normalize vectors.

$$\text{View matrix} = \begin{bmatrix} 0{,}768 & 0 & -0{,}64 & 0{,}128 \\ -0{,}458 & 0{,}699 & -0{,}55 & -0{,}309 \\ 0{,}447 & 0{,}716 & 0{,}537 & -31{,}8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b) (5%)
The camera should have a field of view of 75°, an aspect ratio of 4:3, a near plane at 2 and a far plane at 22. Find the exact values for a matrix that calculates this camera.
Which matrix in your shader should be set to these values?

$$top = 2 \cdot \tan\left(\frac{75}{2}\right) = 1,53$$

$$bottom = -top = -1,53$$

$$right = ratio \cdot top = \frac{4}{3} \cdot 1,53 = 2,04$$

$$left = -right = -2,04$$

$$\begin{bmatrix} \frac{2 \cdot 2}{4,08} & 0 & 0 & 0 \\ 0 & \frac{2 \cdot 2}{3,06} & 0 & 0 \\ 0 & 0 & \frac{-(22+2)}{22-2} & \frac{-2 \cdot 22 \cdot 2}{22-2} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Projection matrix

$$= \begin{bmatrix} 0,98 & 0 & 0 & 0 \\ 0 & 1,307 & 0 & 0 \\ 0 & 0 & -1,2 & -4,4 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

c) (5%)
Vertex data should be drawn into a coordinate frame that has been translated by (6, 6, 1) and then rotated by 60° about the z-axis. Represent this coordinate frame in a matrix. Which matrix would this commonly be?

$$\begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0,5 & -0,866 & 0 & 0 \\ 0,866 & 0,5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0,5 & -0,866 & 0 & 6 \\ 0,866 & 0,5 & 0 & 6 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Model matrix

d) (5%)
A vertex is run through the vertex shader.
It has the position values (3, 2, 5).
Given the matrix values calculated in parts a, b & c, what values will the vertex shader set to gl_Position?
Will this vertex be within the viewing volume and thus (other tests notwithstanding) be rendered as part of the final image? Explain.

$$\begin{bmatrix} 0,5 & -0,866 & 0 & 6 \\ 0,866 & 0,5 & 0 & 6 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} 0,768 & 0 & -0,64 & 0,128 \\ -0,458 & 0,699 & -0,55 & -0,309 \\ 0,442 & 0,716 & 0,537 & -31,8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} 0,98 & 0 & 0 & 0 \\ 0 & 1,307 & 0 & 0 \\ 0 & 6 & -1,2 & -4,4 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} \begin{array}{l} \leftarrow x \\ \leftarrow y \\ \leftarrow z \\ \leftarrow w \end{array}$$

is all of these >-w and <w

then yes, otherwise, no

answer

5. **(10%) Lighting Calculations**

A single light is in the light model in an OpenGL program. It has the ambient values (0.3, 0.4, 0.6), diffuse values (0.4, 0.5, 0.7), specular values (0.6, 0.6, 0.6) and position (3.0, 7.0, -2.0). A camera is positioned in (3.0, 5.0, 1.0) and looks towards P.
P has the color values: ambient (0.3, 0.3, 0.3), diffuse (0.7, 0.7, 0.7) and specular (0.8, 0.8, 0.8). It has a shininess value of 13. It has the position (3.0, 4.0, -1.0) and a normal (0.0, 3.0, 1.0).
What will be the red color value for P on the screen?

$s = (0, 3, -1) \quad v = (0, 1, 2) \quad h = (0, 4, 1)$

$|s| = \sqrt{10} \quad |v| = \sqrt{5} \quad |h| = \sqrt{17}$

$n = (0, 3, 1)$

$|n| = \sqrt{10}$

$\text{lambert} = \frac{3 \cdot 3 + 1 \cdot (-1)}{\sqrt{10} \cdot \sqrt{10}} = \frac{8}{10} = 0{,}8$

$\text{phong} = \frac{3 \cdot 4 + 1 \cdot 1}{\sqrt{10} \cdot \sqrt{17}} = \frac{13}{\sqrt{170}} = 0{,}997$ $(0{,}99)$ $(0{,}877)$

$\text{phong}^f = 0{,}997^{13} = 0{,}96$

$I_r = 0{,}3 \cdot 0{,}3 + 0{,}4 \cdot 0{,}7 \cdot 0{,}8 + 0{,}6 \cdot 0{,}8 \cdot 0{,}96$

$= 0{,}7748 \simeq 0{,}775 \simeq 0{,}77$

$+ (0{,}6 \cdot 0{,}8 \cdot 0{,}877)$

$= (0{,}735)$

6. (10%) Vector intersections and reflections

A line has end points (-4, 0) and (2, -2).
A particle starts at (2, 5) and travels along in the direction (-2, -3).

a) (7%) In which point does the path of the particle cross the line?

$$v = (2-(-4) \; ; \; -2-0) = (6, -2) \qquad A = (2,5)$$

$$n = v^{\perp} = (2, 6) \qquad B = (2, -2) \qquad c = (-2, -3)$$

$$t_{hit} = \frac{(2,6) \circ (0, -7)}{(2,6) \circ (-2,-3)} = \frac{-42}{-22} = \frac{42}{22} \approx 1{,}91$$

$$P_{hit} = A + c \cdot t_{hit} = (2,5) + 1{,}91 \cdot (-2, -3)$$

$$= (-1{,}82 \; ; \; -0{,}73)$$

$$\left(-\frac{20}{11} \; ; \; -\frac{8}{11}\right)$$

b) (3%) If the particle is made to bounce off the line, what will its new direction vector be?

$$a = c = (-2, -3)$$

$$n = (2, 6)$$

$$r = a - 2\,(a \circ \hat{n}) \cdot \hat{n}$$

$$= a - \frac{2 \cdot (a \circ n)}{|n|^2} n = (-2,-3) - 2 \cdot \frac{-2 \cdot 2 - 3 \cdot 6}{2 \cdot 2 + 6 \cdot 6} \cdot (2,6)$$

$$= (-2, -3) + \frac{44}{40} \cdot (2, 6)$$

$$= (0{,}2 \; ; \; 3{,}6) \qquad \left(\frac{1}{5} \; ; \; \frac{18}{5}\right)$$

7. **(10%) Bezier motion**

Scalars in bezier curves can be found by factoring Bernstein polynomials: $BL = ((1-t) + t)^L$ for a bezier curve with $L + 1$ control points.

The camera is moved along a bezier curve with 4 control points.
P1 = (-2, -2, 0), **P2** = (2, 2, 0), **P3** = (5, 1, 0), **P4** = (6, -3, 0)

The motion should start 15 seconds after the program starts and it should end 30 seconds later, 45 seconds after the program starts.
What is the camera's position 25 seconds after the program started?

$$t = \frac{current\_time - start\_time}{end\_time - start\_time}$$

$$t = \frac{25 - 15}{45 - 15} = \frac{10}{30} = \boxed{\frac{1}{3}}$$

$$P = (1-t)^3 \cdot P1 + 3 \cdot (1-t)^2 \cdot t \cdot P2 + 3 \cdot (1-t) \cdot t^2 \cdot P3 + t^3 \cdot P4$$

$$P = (1{,}63 \,;\, 0{,}41 \,;\, 0)$$

$$\left({}^{44}/_{27} \,;\, {}^{11}/_{27} \,;\, 0 \right)$$

8. (10%) Rasterization

Three vertices of a triangle have been sent through the OpenGL pipeline. They have the following pixel positions as well as values for the varying variable v_d:

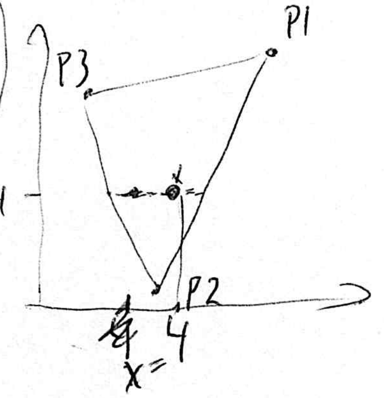
P1: position = (9,8) – v_d = 10

P2: position = (3,2) – v_d = 4

P3: position = (1,6) – v_d = 20

What will the fragment shader value of v_d be set to at pixel (4,4)?

$$t_{left} = \frac{\cancel{P3}.y - P2.y}{P3.y - P2.y} = \frac{4-2}{6-2} = \boxed{\frac{1}{2}}$$



$$t_{right} = \frac{y - P2.y}{P1.y - P2.y} = \frac{4-2}{8-2} = \boxed{\frac{1}{3}}$$

$$x_{left} = lerp(P2.x, P3.x, t_{left}) = \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 1 = 2$$

$$x_{right} = lerp(P2.x, P1.x, t_{right}) = \frac{2}{3} \cdot 3 + \frac{1}{3} \cdot 9 = 5$$

$$v\_d_{left} = lerp\left(4, 20, \frac{1}{2}\right) = \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot 20 = 12$$

$$v\_d_{right} = lerp\left(4, 10, \frac{1}{3}\right) = \frac{2}{3} \cdot 4 + \frac{1}{3} \cdot 10 = 6$$

$$v\_d_{(4,4)} = lerp\left(12, 6, \frac{4-2}{5-2}\right) = \frac{1}{3} \cdot 12 + \frac{2}{3} \cdot 6 = \boxed{8}$$