Do not open the test until told to begin!

CMSC 435/634: Computer Graphics

Midterm Exam

Due 4:00 PM March 26, 2021

Instructions: You have 48 hours to complete this exam, including time to submit at the end. You can do it on paper to scan or photograph, complete it electronically directly in the pdf on a tablet, or as clearly numbered answers in a text, latex, or word document. Be sure to **add**, **commit**, and ****push**** your answers to your git repository by 4:00 PM on March 26th.

This exam is open book and notes: you may use any notes you have taken, the class texts, any of the class lecture slides, or any of the recorded lectures. You can use a calculator to assist you in coming up with numeric answers where one makes sense, and a computer for numeric computation or drawing. You cannot post new public messages on webex, discord or other forum, do searches of the wider web, or consult with anyone besides the TA and instructor about the exam. If anything is not clear, post a private message to the TA and instructor on webex or by email. If any bugs come up in the exam, I will post corrections or clarifications on webex to the class as a whole.

Good luck

Name: Nick Conway (JJ02259 / conwan1@umbc.edu) — Answers

Do not open the test until told to begin!

1 Odds and ends (30 points total)

1.1 What is the most amusing bug you've written in this class? **(5 points)**

Any answer

1.2 What is the maximum number of rays (both photon rays and shadow rays) for a 128*128 pixel ray traced image with 5 lights and a 4-bounce limit for reflection **and** refraction? **(5 points)**

width*height * (lights + 1) * ($2^{bounces+1}$ – 1)
128*128*(5+1)*(2^{4+1} – 1) = 3,047,424

1.3 What CSG operation between a red and yellow sphere does this image show? **(5 points)**

Difference

1.4 Is $x^2 + 3y^2 - 1 = 0$ a parametric or implicit equation for an ellipse? **(5 points)**

implicit

1.5 The ray-triangle intersection using barycentric coordinates between ray $\vec{p} = \vec{e} + t\ \vec{d}$ and triangle $\vec{p} = \alpha\vec{p}_0 + \beta\vec{p}_1 + \gamma\vec{p}_2$ is the solution of four equations in four unknowns. What are the unknowns? What are the four equations? **(5 points)**

Unknowns: t, α, β, γ
Equations: some form of

$$e_x + td_x = \alpha x_0 + \beta x_1 + \gamma x_2$$
$$e_y + td_y = \alpha y_0 + \beta y_1 + \gamma y_2$$
$$e_z + td_z = \alpha z_0 + \beta z_1 + \gamma z_2$$
$$\alpha + \beta + \gamma = 1$$

1.6 What is total internal reflection? Give an equation for when it happens. **(5 points)**

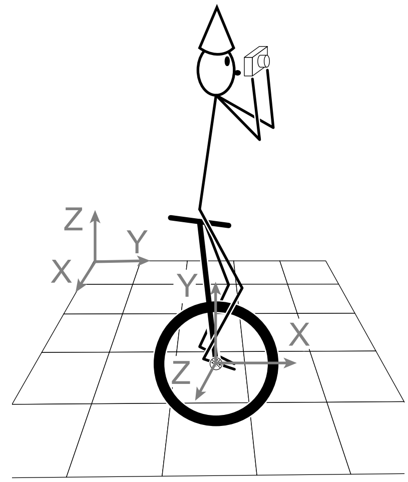
Imaginary refraction direction, so all light is reflected
Forms the equation might take:

$1 - (1 - (\hat{n} \cdot \hat{v})^2)\, n_v^2/n_t^2 < 0$
$(1 - \cos^2\theta_v)\, n_v^2/n_t^2 > 1$
$\sin^2\theta_v\, n_v^2/n_t^2 > 1$
$\sin\theta_v\, n_v/n_t > 1$
$\sin\theta_v > n_t/n_v$

2 Transformations & Viewing

(40 points total)

You need to model the view from a character on a unicycle holding a camera. The world coordinates are defined so X and Y span the map horizontally and Z points up. The unicycle coordinates are centered at the center of the axle, with Z pointing right, X pointing forward, and Y pointing up.



The standard math library function atan2(y, x) may be useful. This function computes the arctangent of y/x over the full circle, without division by 0 when x=0, and using the signs of both arguments to correctly determine the quadrant of the resulting angle.

2.1 Given a function to create a 4x4 translation matrix, Translate(t), where $t=(t_x, t_y, t_z)$, and functions to generate 4x4 rotation matrices around the coordinate axes, RotateX(θ), RotateY(θ), and RotateZ(θ), construct the transform **from** unicycle space **to** world space for a unicycle at a world-space location of $u=(u_x, u_y, u_z)$, pointing in the normalized world-space direction $d=(d_x, d_y, 0)$ **(10 points)**

Translate(t) * RotateZ(atan(dy,dx)) * RotateX(-90º)

2.2 Construct this transform in matrix form directly by columns. Give your reasoning for each column. **(10 points)**

$$\left(\begin{array}{c|c|c|c} d_x & 0 & d_y & u_x \\ d_y & 0 & -d_x & u_y \\ 0 & 1 & 0 & u_z \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Last column = u to move unicycle origin to u
Y column = (0,0,1) to transform unicycle up to world up
X column = d to transform unicycle forward to world direction
Z column = X × Y to make be perpendicular

2.3 If the camera the character is holding is at c=(c_x, c_y, c_z) in unicycle coordinates with horizontal pan angle of θ and vertical tilt angle of ϕ, what sequence of Rotate and Translate calls will create a transform **from** world space **to** view space? In view space, the camera should be at the origin, looking down the –Z axis, with X right and Y up. **(10 points)**

WorldFromView = Translate(t) * RotateZ(atan(dy,dx)) * RotateX(-90°) * Translate(c) * RotateZ(ϕ) * RotateY(θ)
so …
ViewFromWorld = RotateY(-θ) * RotateZ(-ϕ) * Translate(-c) * RotateX(90°)* RotateZ(-atan(dy,dx)) * Translate(-t)

2.4 Derive values for a, b, c, and d in this perspective matrix for a 1920×1080 screen with 60° horizontal field of view, that transforms the left frustum plane to x = –1, right frustum plane to x = 1, bottom frustum plane to y = –1, top frustum plane to y = 1, near frustum plane at z = –0.1 in view space to z = 0 in clip space, and far frustum plane at z = –∞ in view space to z = 1 in clip space. Hint: This differs from the one in class in three ways: it has a non-square aspect ratio, the near plane is transformed to z=0 in clip space instead of –1, and the far plane is at infinity in view space. **(10 points)**

$$\begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$a = \cot(\theta/2) = 1.73$
$b = \cot(\theta/2) * 1080/1920 = 0.97$

For c and d, start by solving for n and f mapping to 0 to 1 output

$$(0,0,-n,1) \rightarrow (0,0,-cn+d,n);\ \ (0,0,-f,1) \rightarrow (0,0,-cf+d,f)$$

$$\frac{-cn+d}{n} = 0;\ \ \frac{-cf+d}{f} = 1$$

$$c = \frac{f}{n-f};\ \ f = \frac{fn}{n-f}$$

Limit as f goes to ∞

$$c = -1;\ \ f = -n$$

Plug in actual values:

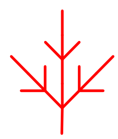
$c = -1$, $d =$ -0.1

3 L-Systems **(30 points total)**

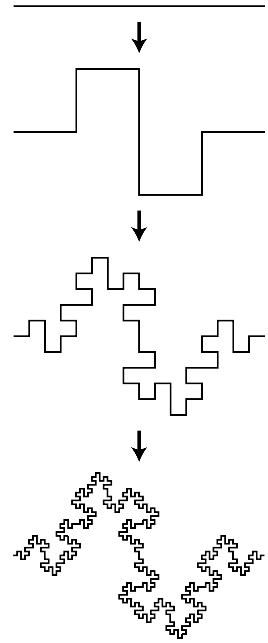
3.1 Consider a 2D L-system with five symbols: '[' and ']' push and pop the current transformation, '+' and '-' rotate counter-clockwise and clockwise by 45º, and 'T' and 'B' draw a unit-length straight line segments and translate to the end of that line segment. This L-system has just one rule: B → T[-B][+B]TB. Starting with a start symbol of B (generation zero), write the first and second generation L-system strings and sketch what they would look like. **(10 points)**

1: T[-B][+B]TB

2: T[-T[-B][+B]TB][+T[-B][+B]TB]TT[-B][+B]TB



3.2 Consider a second L-system with the following symbols: '+' and '-' rotate counter-clockwise and clockwise by 90º, 'S' scales everything that follows by $^1/_4$, and *all other letters* draw a unit-length line segment and move the current position to the end of the segment. The initiator (level 0) for *Minkowski Sausage* fractal is a single line segment "X". What replacement rule(s) would generate this fractal? Hint: pay particular attention to how many S's you generate, there should be just one per level. You may need to use additional symbols to accomplish this. **(10 points)**



X → SX-Y+Y+YY-Y-Y+Y
Y → Y-Y+Y+YY-Y-Y+Y

3.3 Write out the first and generation L-system string for this fractal and first 20 symbols of the second and third generation strings. **(10 points)**

1: SX-Y+Y+YY-Y-Y+Y
2: S SX-Y+Y+YY-Y-Y+Y Y-Y+ …
3: S S SX-Y+Y+YY-Y-Y+Y Y-Y …