Midterm Info

• Time: Wednesday, October 27, 7-10pm
• Location: Cubberly Auditorium

• Open notes, open book, open computer but no internet
• This means the focus will be on concepts and applications, rather than recollection
Review of topics

• Review of topics
  – Drawing in OpenGL
  – Geometry: points, lines, vectors
  – Transforms and coordinate systems
  – Interpolation and Splines
  – Input devices and interaction
  – Fonts/typography
OpenGL
Geometry

• Why STPoint and STVector?
  – Point + Point = ? -- illegal
  – Point + Vector = ?
  – Vector – Vector = ?
  – a * Point = ?

• When might it make sense to do sum over points?
  – Averaging points (finding the centroid)
  – OK if all the weights add up to 1 (barycentric)

• How does each behave under transformation?

• What are orthonormal Vectors?

• Dot product and Cross product
Geometry

- Parametric Curves
  \[ f(t) = x(t)x + y(t)y \]

- Implicit Curves
  \[ f(x, y) = 0 \]

- Compare these two

\[ \{ x, y \mid f(x, y) = 0 \} \]
Rasterization

• Many different ways to determine whether or not a pixel is “covered” by a polygon.
  – Polygon intersects pixel
  – Center of pixel

• Desirable properties:
  – Easy to compute, should be order-independent
  – No holes between two abutting shapes
  – Boundaries between semi-transparent shapes

• Rasterization Rules for Lines

• Triangle rasterization
Drawing In OpenGL

• OpenGL primitives
  – glBegin/glEnd
• Double buffering
  – glutInitDisplayMode()
• Setup Coordinate System
• GLUT events
  – Various Callback functions
Transformations

• Many ways to represent transformations, but matrix multiplication is very convenient.
• How do we represent:
  – Scaling
  – Rotation (axis-aligned)
  – Translation
Transformations in 3D

- **Scale**
  \[
  \begin{bmatrix}
  s_x & 0 & 0 & 0 \\
  0 & s_y & 0 & 0 \\
  0 & 0 & s_z & 0 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \]

- **Translate**
  \[
  \begin{bmatrix}
  1 & 0 & 0 & t_x \\
  0 & 1 & 0 & t_y \\
  0 & 0 & 1 & t_z \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \]

- **Rotate** (by \(d\) around z-axis)
  \[
  \begin{bmatrix}
  \cos(d) & -\sin(d) & 0 & 0 \\
  \sin(d) & \cos(d) & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \]

Q: Why is 3x3 insufficient?
Inverse Transformations

• Inverse of Translation

\[ T^{-1}(t_x, t_y, t_z) = T(-t_x, -t_y, -t_z) \]

• Inverse of (axis-aligned) Rotation

\[ R^{-1}(d) = R(-d) \]

• Inverse of Scaling

\[ S^{-1}(s_x, s_y, s_z) = T(1/s_x, 1/s_y, 1/s_z) \]

\((ABC)^{-1} = ?\)
Sample Question 1

• State whether the following formulas involving inverses are True or False.

\[ R(180)^{-1} = R(180) ? \]

\[ T(1,1)^{-1} = T(-1,-1) ? \]

\[ [R(45) \ T(1,0)]^{-1} = T(1,0) \ R(45) ? \]

\[ [R(45) \ S(2,2)]^{-1} = R(-45) \ S(.5,.5) ? \]
Transformations

• Think about them in two ways
  – Applying to object
  – Applying to local coordinate axes
Order of Transformations

• Matrix multiplication is associative
• Transformations are not commutative
  – TR is not the same as RT. Example?
  – TS is not the same as ST. Example?
  – How about SR and RS (where S is a uniform scale)?
Transformation in 2D
Transformations

- Rotating around a point that is not the origin:
  - Translate to point (it is now origin)
  - Perform rotation
  - Translate back
Interesting Question

\[ T_1R_1 \]
\[ R_2T_2 \]

\[ T_1 = T(1, 1); \quad R_1 = R(45) \]
\[ T_2 = ?; \quad R_2 = ? \]
Sample Question 1

\[ T(1,0) \ S(2,2) = T(2,0) \ S(1,1) ? \]

\[ T(-1,0) \ T(0,2) = T(0,1) \ T(-1,1) ? \]

\[ R(180) = R(-180) ? \]

\[ R(45) \ S(2,1) = S(2,1) \ R(45) ? \]

\[ T(1,0) \ R(90) = T(0,1) ? \]

\[ R(-90) \ T(1,0) \ R(90) = T(0,-1) ? \]
OpenGL Matrix Stack

• How are transformations composed in OpenGL? Which order and why?
• Hierarchy of objects
Input Device

• The user presses a key. What happens?
• What is quadrature encoding?
• How does the trackball use quadrature encoding to determine the direction and the amount of movement?
• How does a mouse work?
Events and Interaction

- Interrupts
  - Signal when value changes
  - Only hear about important events
  - May not be good if events fire rapidly
- Polling
  - Poll for state that you are interested
  - Useful for things that change often
- glut mouse events are like interrupts.
- If you were designing a system, how would you handle input from:
  - Keyboard
  - Mouse
  - Graphics Card
Linear Interpolation

- **Lerp**: Linearly interpolate between two values.

\[ y(t) = (1 - t)y_1 + ty_2 \]

Diagram: Two points \( y_1 \) and \( y_2 \) connected by a line segment, with a point \( y(t) \) on the line between them.
Sample Question 3

3A [5 points]. We are given a point $P=(px,py,pz)$ and a vector $V=(vx,vy,vz)$. We now translate by $(tx,ty,tz)$. What is the new position of the point? What are the new coordinates of the vector?

\[ P' = (px+tx, py+ty, pz+tz), \quad V' = (vx, vy, vz). \]

3B [5 points]. Suppose we rotate the coordinate system by 90 degrees about $z$. What is the new position of the point? And the vector? (In this question, ignore the translation in 3A).

\[ P' = (py, -px, pz), \quad V' = (vy, -vx, vz) \]

OR \[ P' = (-py, px, pz), \quad V' = (-vy, vx, vz) \]
3C [5 points]. We wrote a program that computes a new point from two points using the expression \( p = a \cdot p_1 + b \cdot p_2 \), where \( p, p_1 \) and \( p_2 \) are points, and \( a \) and \( b \) are floats. Suppose the points \( p_1 \) and \( p_2 \) are translated by \((tx, ty, tz)\). That is, \( p_1' = T(tx, ty, tz) \cdot p_1 \) and \( p_2' = T(tx, ty, tz) \cdot p_2 \). We would expect the point \( p' = T(tx, ty, tz) \cdot p \). Prove whether this is true or false.

\[
p' = a \cdot p_1' + b \cdot p_2' \quad ?=\quad T(tx, ty, tz) \cdot p
\]

**Proof:**
\[
a \cdot p_1' + b \cdot p_2' = (ax_1 + at_x + bx_2 + bt_x, ay_1 + at_y + by_2 + bt_y, az_1 + at_z + bz_2 + bt_z)
\]
\[
T(tx, ty, tz) \cdot p = (ax_1 + bx_2 + tx, ay_1 + by_2 + ty, az_1 + bz_2 + tz)
\]
Sample Question 3

3D [5 points]. Suppose we compute \( p = (1-a) \cdot p1 + a \cdot p2 \). Now we translate \( p1 \) and \( p2 \) as we did in 3C. Again, we would expect \( p' = T(tx,ty,tz) \cdot p \). Is this true or false?
Interpolation

- **Bilinear interpolation**
  - 3 lerps.

- **Barycentric Interpolation**
  - Weights sum to 1
  - Use area of opposite triangle
Interpolation

• Given a set of points, find a curve that goes through these points.

\[ p(t) = \sum_{i=0}^{n} c_i B_i(t) \]

Control point: \( c_i \)

Basis function: \( B_i(t) \)
Interpolation

- Different choices for basis functions
  - Triangle: piece-wise linear
    - Why does this work?
  - Square: nearest neighbor
    - Why does this work?
Morphing

• Warping with one pair of feature Line segments

• Warping with multiple pairs of feature Line segments

\[ p' = \sum_{i} w_i p_i \]

• Bilinear interpolation in colors
Polynomial Interpolation

- Constraints vs Degrees
- Piecewise polynomial interpolation
Splines

- Cubic-Hermite interpolation
  - Specify endpoints and tangents
  - Represents cubic curves
  - What are interesting properties its basis functions?

- How to find $H_i(t)$?
Curves

- **Catmull-Rom**
  - Given a set of points, how to define a smooth curve that **interpolates** them?
  - No tangents given. Define tangents using the next and previous control point
  - This can now be reduced to a Cubic-Hermite spline
Beziers Curves

- Cubic Bezier curves as Cubic-Hermite
  - $P_0 = P_0$
  - $P_1 = P_3$
  - $T_0 = 3(P_1 - P_0)$
  - $T_1 = 3(P_3 - P_2)$

- Smooth, but lets you easily define sharp corners
- Curve contained in convex hull of control points
Bezier Curves

• Evaluating bezier curves:
  – Direct evaluation. How?
  – Chaiken’s Algorithm. How?
  – Subdivision Algorithm. How?

Linear combination of control points:

\[ P(t) = \sum_{i}^{n} P_i B_{i}^{n}(t) \]

Bernstein polynomials as weighting functions:

\[ B_{i}^{n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \]
Sample Question 4 - Bezier Curve

What should the positions of the 3 control points of the quadratic Bezier curve be?

Requirements
1. so that the Bezier curve goes through (0,0) and (1,1)
2. the Bezier curve is tangent to the parabola at these two points
Sample Question 4 - Bezier Curve

What should the positions of the 3 control points of the quadratic Bezier curve be? \( P_0, P_1, P_2 = ? \)

Tangent of the curve: \( \frac{dy}{dx} = 2x \)

slope at \((0, 0)\) = 0
slope at \((1, 1)\) = 2

\( P_0 = (0, 0) \)
\( P_1 = (a, b) \)
\( P_2 = (1, 1) \)

The tangents in \( P_0 \) and \( P_2 \) both pass through \( P_1 \).

tangent at \((0, 0)\) = 0 = \( b/a \) \( \Rightarrow \) \( b = 0 \)
tangent at \((1, 1)\) = 2 = \( 1/(1-a) \) \( \Rightarrow \) \( a = 0.5 \)
Sample Question 4 - Bezier Curve

What should the positions of the 4 control points of the cubic Bezier curve be?

Requirements:
1. The Bezier curve goes through (0,0) and (1,1)
2. The Bezier curve is tangent to the parabola at these points.
3. the point (1/2,1/4) on the parabola be on the Bezier curve, that is P(1/2) = (1/2,1/4)

P₀, P₁, P₂, P₃ = ?
Sample Question 4 - Bezier Curve

What should the positions of the 4 control points of the cubic Bezier curve be?

Requirements:
1. The Bezier curve goes through (0,0) and (1,1)
2. The Bezier curve is tangent to the parabola at these points.
3. The point (1/2,1/4) on the parabola be on the Bezier curve, that is $P(t = 1/2) = (1/2,1/4)$

$P_0, P_1, P_2, P_3 = ?$

$P_0 = (0, 0); P_1 = (x_1, y_1);$
$P_2 = (x_2, y_2); P_3 = (1, 1);$
Sample Question 4 - Bezier Curve

What should the positions of the 4 control points of the cubic Bezier curve be?

Requirements:
1. The Bezier curve goes through (0,0) and (1,1)
2. The Bezier curve is tangent to the parabola at these points.
3. The point (1/2, 1/4) on the parabola be on the Bezier curve, that is \( P(t=1/2) = (1/2, 1/4) \)

\[
\begin{align*}
slope \text{ at } (0, 0) &= 0 \quad \Rightarrow y_1 = 0 \\
slope \text{ at } (1, 1) &= 2 \quad \Rightarrow 1-y_2/1-x_2 = 2
\end{align*}
\]
Sample Question 4 - Bezier Curve

What should the positions of the 4 control points of the cubic Bezier curve be?

Requirements:
1. The Bezier curve goes through (0,0) and (1,1)
2. The Bezier curve is tangent to the parabola at these points.
3. The point (1/2, 1/4) on the parabola be on the Bezier curve, that is P(t = 1/2) = (1/2, 1/4)

Parametric expression of P(t):

\[ P(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3 \]

\[ P(1/2) = 1/8P_0 + 3/8P_1 + 3/8P_2 + 1/8P_3 \]
Sample Question 4 - Bezier Curve

What should the positions of the 4 control points of the cubic Bezier curve be? $P_0, P_1, P_2, P_3 = ?$

$P_0 = (0, 0); P_1 = (x_1, y_1); P_2 = (x_2, y_2); P_3 = (1, 1);$

slope at $(0, 0) = 0$  --> $y_1 = 0$
slope at $(1, 1) = 2$  --> $1-y_2/1-x_2 = 2$

$P(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3$

Require that $P(1/2) = (1/2, 1/4)$

$P(1/2) = 1/8P_0 + 3/8P_1 + 3/8P_2 + 1/8P_3$

$P_x(1/2) = 1/8*0 + 3/8*x_1 + 3/8*x_2 + 1/8*1 = 1/2$

$P_y(1/2) = 1/8*0 + 3/8*y_1 + 3/8*y_2 + 1/8*1 = 1/4$

$P_y(1/2) = 3/8*y_2 + 1/8*1 = 1/4$  --> $y_2 = 1/3$

$1-y_2/1-x_2 = 2$  --> $x_2 = 2/3$

$P_x(1/2) = 3/8*x_1 + 3/8*2/3 + 1/8*1 = 1/2$  --> $x_1 = 1/3$
Typography

- Encoding vs Fonts
  - Unicode
  - Glyph index
  - Font metrics

- Different properties of fonts
  - Serif
  - Stress
  - Thick/thin transitions + ratio

- Also, there are variants:
  - Style (italic/oblique)
  - Weight
  - Stretch
  - Font sizes: pt, pc, em, en
Typography

Old style

- Diagonal stress
- Slanted lowercase serifs
- Moderate thick/thin
Typography

Modern

- Vertical stress
- Serifs are thin and perpendicular
- Large thick/thin
Typography

Slab Serif

- Vertical Stress
- Flat serifs
- Very little thick/thin
Layout

• Kerning
• Ligatures
• Leading
• Box / glue model

• Can you point out where is the baseline?
• Where is the cap height?
• Were are serifs applied?
• Can you identify Ligatures?

Affiliation
Typography

• Glyph metrics