CS 148 Midterm Review
Midterm Information

• Time: Monday, October 26, 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
OpenGL
Geometry

• Why STPoint and STVector?
  – Point + Point = ?
  – 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?
Transformations

• Many ways to represent transformations, but matrix multiplication is very convenient.

• 3x3 matrices are not sufficient. Almost all graphics systems use 4x4 matrices.

• How do we represent:
  – Scaling
  – Rotation (axis-aligned)
  – Translation
Transformations

- 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)

\[
\begin{bmatrix}
cos(d) & -sin(d) & 0 & 0 \\
sin(d) & cos(d) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Transformations

forward 50  right 90  forward 50  right 90

forward 50  right 90  forward 50  right 90

© 2000 Logo Foundation
Transformations

- Transformations are not commutative
  - TS is not the same as ST. Example?
  - TR is not the same as RT. Example?
  - How about SR and RS (where S is a uniform scale)?

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

• How are transformations composed in OpenGL? Which order and why?
• PushMatrix saves current matrix transform. PopMatrix restores it.
• These are used to draw object hierarchies
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
  - No holes between two abutting shapes
Hit Testing

• Plane equation
  – All points on one side of a plane will have the same sign
  – Useful for convex shapes

• Other easy geometric objects:
  – Boxes (used this for MicroUI)
  – Circles

• Determining if you are inside or outside (TrueType)
  – Fire a ray outward and count how many times you intersect the surface
  – This is easy to implement when scan-converting
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
Events and Interaction

• Can you translate between interrupt and polling?
• Picking schemes: how to determine what the user clicked on?
  – MicroUI: Hit Test (what order to test?)
  – In a 3D application: color each primitive differently
Interpolation

- Lerp: Linearly interpolate between two values

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

• **Cubic-Hermite curve**
  – Specify endpoints and tangents
  – Represents cubic curves
  – What are interesting properties its basis functions?

• **How to find $H_i(t)$?**
Curves

• Cubic-Hermite Basis
  - \( H_0(t) = 2t^3 - 3t^2 + 1 \)
  - \( H_1(t) = -2t^3 + 3t^2 \)
  - \( H_2(t) = t^3 - 2t^2 + 1 \)
  - \( H_3(t) = t^3 - t^2 \)

• Can this ever be quadratic?
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
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 let’s you easily define sharp corners
- Curve contained in convex hull of control points
- Benefits of cubic vs. quadratic (TrueType)
Curves

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

• When is a quadratic bezier curve degenerate? How about a cubic?
Typography

• 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

• Wikipedia pages on font and typeface
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
Typography

• What type of font am I using?
  – Stress?
  – Serifs?
  – Thick/thin?

• Properties of this font?

PIXAR
ANIMATION STUDIOS
Typography

- Kerning
- Ligatures
- Leading
- Box / glue model

Glyph positioning on the baseline, with visible glyph origins and advance widths
Typography

- Glyph metrics

<table>
<thead>
<tr>
<th>Ascend</th>
<th>Ascender Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cap</td>
<td>Cap Height</td>
</tr>
<tr>
<td>Mean line</td>
<td>Median</td>
</tr>
<tr>
<td>Baseline</td>
<td>Descender Height</td>
</tr>
</tbody>
</table>

![Glyph Metrics Diagram](image)