What is OpenGL?

• Cross-language, cross-platform application programming interface (API) for rendering 2D/3D graphics

• Originally released by Silicon Graphics Inc. (SGI) in 1992

• Now managed by non-profit technology consortium Khronos Group

• Closely outlines what GPUs are meant to do

Things You Can Do with OpenGL

Source: UNiGiNE
Things You Can Do with OpenGL

Disclaimer

• This tutorial does NOT cover the “modern” OpenGL (version 3.x and higher, latest is 4.5) which uses lower level API’s to give you more flexibility.

• Instead, we focus on the “older” OpenGL (version 2.1) to get your feet wet with high-level API’s.
Drawing Primitive Shapes
Drawing a Triangle

- Starts the draw triangles state
  - `glBegin(GL_TRIANGLES);`
  - `glVertex2f(-2, -1);`
  - `glVertex2f(1, -1);`
  - `glVertex2f(0, 2);`
  - `glEnd();`

- *Default color is actually white

![Diagram of a triangle with vertices at (-2, -1), (1, -1), and (0, 2).]
OpenGL API Convention

- `glBegin(GL_TRIANGLES);`
- `glVertex2f(-2, -1);`
- `glVertex2f(1, -1);`
- `glVertex2f(0, 2);`
- `glEnd();`
Drawing Multiple Triangles

\[
\begin{align*}
&\text{• glEnd();} \\
&\text{• glVertex2f(-2, -1);} \\
&\text{• glVertex2f(1, -1);} \\
&\text{• glVertex2f(0, 2);} \\
&\text{• glVertex2f(0, -4);} \\
&\text{• glVertex2f(3, -4);} \\
&\text{• glVertex2f(3, -2);} \\
&\text{• glEnd();}
\end{align*}
\]

• What happens if number of vertices are not 3n?
Drawing Other Shapes

• `glBegin(GL_POINTS);`
• `glVertex2f(-3, 2);`
• `glVertex2f(2, 1);`
• `glEnd();`

• `glBegin(GL_LINES);`
• `glVertex2f(-2, -2);`
• `glVertex2f(2, -1);`
• `glEnd();`
Some Things Cannot Be Done Inside glBegin/glEnd

- `glLineWidth(2.0);`
- `glBegin(GL_LINES);`
- `glVertex2f(-2, -2);`
- `glVertex2f(2, -1);`
- `glEnd();`

- `glLineWidth(1.0);`
- `glBegin(GL_LINES);`
- `glVertex2f(-2, 3);`
- `glVertex2f(1, 1);`
- `glEnd();`
Color in OpenGL
Setting Color

- `glColor3f(1, 0, 0);`
- `glBegin(GL_TRIANGLES);`
- `glVertex2f(-2, -1);`
- `glVertex2f(1, -1);`
- `glVertex2f(0, 2);`
- `glEnd();`

- OpenGL is a state machine.

Takes in RGB
Color Per Vertex

- `glBegin(GL_TRIANGLES);`
- `glColor3f(1, 0, 0);`
- `glVertex2f(-2, -1);`
- `glColor3f(0, 1, 0);`
- `glVertex2f(1, -1);`
- `glColor3f(0, 0, 1);`
- `glVertex2f(0, 2);`
- `glEnd();`

- Why does the triangle color look like this?
Transparency and Blending

- `glEnable(GL_BLEND);`
- `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);`
- `glColor4f(1, 0.5, 0.5, 0.5);` Takes in RGBA
- `glBegin(GL_QUADS);`
- `glVertex2f(-2, -2);`
- `glVertex2f(2, -2);`
- `glVertex2f(2, 2);`
- `glVertex2f(-2, 2);`
- `glEnd();`

Many possible blend modes here!
Transparency and Blending: Drawing Order

- glEnable(GL_BLEND);
- glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- drawRedSquare();
- drawGreenSquare();

![Diagram showing drawing order of squares]
Transparency and Blending: Drawing Order

- glEnable(GL_BLEND);
- glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- drawGreenSquare();
- drawRedSquare();
Transformations
Translation

Notice it comes before the triangle

• `glTranslatef(-2, -2, 0);`
• `glBegin(GL_TRIANGLES);`
• `glVertex2f(-2, -1);`
• `glVertex2f(1, -1);`
• `glVertex2f(0, 2);`
• `glEnd();`
Rotation

Notice it comes before the triangle

- `glRotatef(90, 0, 0, 1);`
- `glBegin(GL_TRIANGLES);`
- `glVertex2f(-2, -1);`
- `glVertex2f(1, -1);`
- `glVertex2f(0, 2);`
- `glEnd();`
Transformations Are NOT Commutative

• `glRotatef(90, 0, 0, 1);`
• `glTranslatef(-2, -2, 0);`

• vs.

• `glTranslatef(-2, -2, 0);`
• `glRotatef(90, 0, 0, 1);`

• Transformations are stacked (LIFO)
Going to 3D
Depth Test

• Makes sure objects in front cover objects in back
• See glEnable(GL_DEPTH_TEST)
Lighting

• Colors primitives based on light and surface normal
• See glEnable(GL_LIGHTING) and glNormal
Projection

• Controls how 3D coordinates get mapped to 2D
• See glMatrixMode(GL_PROJECTION)

Source: http://www.real3dtutorials.com/tut00002.php
Demo and Beyond
Animation

- Live demo
More Resources

• Official OpenGL Documentation
  • https://www.opengl.org/wiki/OpenGL_Reference
  • Or “man glVertex” on Linux/Mac

• Legacy OpenGL Tutorials
  • NeHe (http://nehe.gamedev.net/tutorial/lessons_01_05/22004/)

• Modern OpenGL Tutorials
  • OpenGL-Tutorial (http://www.opengl-tutorial.org/)
  • OpenGL-Introduction (https://open.gl/)