CS460/560 Computer Graphics

Prof. Lijun Yin

TA: Huiyuan Yang
Draw rectangle
Draw polygon
Draw lines
Draw letter
Draw circle
Draw eclipse
Draw UP
Draw face

Revert equations back to LaTeX code
Detect and typeset equations (LaTeXiT)
Detect and typeset equations (LaTeXiT)
Add to Evernote
OpenGL

- 2D/3D graphics API
- Most widely used/supported by industry
- Runs on Unix, Linux, Window, etc.
- C, C++, Java, etc.
- Hardware independent
- Easy to render objects
GLUT – OpenGL Utility Toolkit

• GLUT makes it considerably easier to learn about and explore OpenGL programming.
• GLUT provides a portable API so your code can run cross platforms.
• GLUT is designed for constructing small to medium sized OpenGL programs.
• GLUT is well-suited to learning OpenGL and developing simple OpenGL applications, GLUT is not a full-featured toolkit so large applications requiring sophisticated user interfaces are better off using native window system toolkits.

• In summary, GLUT is simple, easy, and small.

https://www.opengl.org/resources/libraries/glut/
GLUT – OpenGL Utility Toolkit

FreeGLUT, an alternative of GLUT, as the original GLUT has been unsupported for 20 years.

int main(int argc, char** argv) {
    glutInit(&argc, argv);            // Initialize GLUT
    glutCreateWindow("Hello CS460/560"); // Create a window with the given title
    glutInitWindowSize(500, 500);    // Set the window's initial width & height
    glutInitWindowPosition(200, 200); // Position the window's initial top-left corner
    glutDisplayFunc(display);        // Register display callback handler for window re-paint
    glutMainLoop();                 // Enter the event-processing loop
    return 0;
}
void display() {
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque
    glClear(GL_COLOR_BUFFER_BIT); // Clear the color buffer (background)

    // Define shapes enclosed within a pair of glBegin and glEnd
    glBegin(GL_TRIANGLES); // Each set of 3 vertices form a triangle
    glColor3f(0.0f, 0.0f, 1.0f); // Blue
    glVertex2f(0.1f, -0.6f);
    glVertex2f(0.7f, -0.6f);
    glVertex2f(0.4f, -0.1f);
    glEnd();

    glFlush(); // Render now
}
Header file

• Linux:
  • #include <GL/glut.h> //Unix, GLUT, include glu.h and gl.h

• Mac osx:
  • #include <GLUT/glut.h> //Max OSX
How to Compile

• Linux:
  • `g++ test.cpp -o a.out -lGL -lglut -lGLU`

• Mac osx:
  • `g++ test.cpp -o a.out -framework OpenGL -framework GLUT`
Vertex, Primitive and Color: example-1
OpenGL operates as a *state machine*, and maintain a set of *state variables* (such as the foreground color, background color, and many more). In a state machine, once the value of a state variable is set, the value persists until a new value is given.

```
// Red
glColor3f(1.0f, 0.0f, 0.0f);
glVertex2f(0.3f, -0.4f);
// Green
glColor3f(0.0f, 1.0f, 0.0f);
glVertex2f(0.9f, -0.4f);
// Blue
glColor3f(0.0f, 0.0f, 1.0f);
glVertex2f(0.6f, -0.9f);
```
Naming Convention

begins with lowercase
  gl (for core OpenGL),
  glu (for OpenGL Utility)
  glut (for OpenGL Utility Toolkit).

eg.
  glVertex2f
  glColor3f

followed by specifications
  purpose of the function
  for the parameters
Coordinate System

OpenGL 2D Coordinates

Viewport Coordinates

Clipping-Area
(default of 2x2 square centered at origin)

Viewport
(width-by-height in pixels)
Menu: Example