OpenGL

- State machine: implicit global variables
  - The window, where it is, size, etc.
  - The current color for drawing.
  - Buffers
  - Type of projection
- Interactive programming: callbacks
  - Don’t think about executing a program. Think about setting up functions that the user’s actions will call.
Callbacks

• glutDisplayFunc – gets called when screen display needed.
• glutMouseFunc – called when mouse action occurs.
• glutKeyboardFunc

Getting Started

• Conventions
  – OpenGL functions begin gl, each word in caps: eg., glBegin, glPolygonMode
  – Constants: GL_2D, GL_RGB, …
  – Data types: GLbyte, GLfloat, …
GLUT

• OpenGL machine independent
• GLUT machine dependent
  – Display
  – Input devices
  – GLUT functions: glutInitWindowSize, glutIdleFunc, …
  – GLUT constants: GLUT_RIGHT_BUTTON, …

Color

• Displays with three colors
• RGB representation of color
  – Red (1,0,0); Green (0,1,0); blue (0,0,1)
  – White (1,1,1); black (0,0,0)
  – Pink? Purple?
Initialization

• `#include <GL/glut.h>`
  – Also includes windows stuff and OpenGL

• `glutInit` (int * argcp, char **argv)
  – Initialize GLUT library, parse and use command-line options:

• `glutInitWindowSize` (int width, int height)
• `glutInitWindowPosition` (int x, int y)
• `glutInitDisplayMode` (unsigned int mode)
  – GLUT_RGBA | GLUT_DEPTH | GLUT_DOUBLE, etc…
  – Single argument with OR of constants
  – Type of buffering, we’ll use single at first.

Initialization

• `glutCreateWindow` (char *window_name)

• `glClearColor (1.0, 1.0, 1.0, 0.0)`
  – Background properties
  – First three give RGB values
  – Fourth gives blending for transparent objects. We won’t use this for a while.
Projection

- `glMatrixMode (GL_PROJECTION);`
  - The current matrix relates to projection. We won’t use others right now.
- `gluOrtho2D (0.0, winWth, 0.0,winHght);`
  - Sets up orthographic projection from 3D scene to image. More on this later.
  - This form sets up most trivial projection.

GLUT Callback Registration

- `glutDisplayFunc (void (*func) (void))`
GLUT Main Event Loop

- **glutMainLoop** (void)
  - Starts the GLUT event processing loop
  - Never returns
  - Calls registered function callbacks (user-defined event handlers) as appropriate
  - Should be called at most once

Specifying Vertices

- **glVertex2s** (200, -150);
  - 2D point in short coordinates

- **glVertex3i** (200, -150, 40);
  - 3D point in integer coordinates

- **GLdouble** *dpoint*[3] = {200.0, -150.5, 40.0};
  - 3D point in double coordinates
  - **glVertex3dv** (*dpoint*);
Points, Lines, Polygons

- `glBegin(mode)` and `glEnd()` delimit an object

- `mode` can be one of the following:
  - `GL_POINTS`
  - `GL_LINES`
  - `GL_POLYGON`
  - `GL_LINE_STRIP`
  - `GL_TRIANGLE_STRIP`
  - `GL_TRIANGLES`
  - `GL_QUADS`
  - `GL_LINE_LOOP`
  - `GL_QUAD_STRIP`
  - `GL_TRIANGLE_FAN`

```c
glBegin(GL_POINTS);
glVertex2i(0, 0);
glVertex2i(0, 1);
glVertex2i(1, 0);
glVertex2i(1, 1);
glEnd();
```
Line Loop (Polyline)

```c
glBegin(GL_LINE_LOOP);
glVertex2i( 0, 0 );
glVertex2i( 0, 1 );
glVertex2i( 1, 1 );
glVertex2i( 1, 0 );
glEnd();
```

---

Polygon

```c
glBegin(GL_POLYGON);
glVertex2i( 0, 0 );
glVertex2i( 0, 1 );
glVertex2i( 1, 1 );
glVertex2i( 1, 0 );
glEnd();
```
**Triangles**

```c
glBegin(GL_TRIANGLES);
glVertex2i(0, 0); // a
glVertex2i(0, 1); // b
glVertex2i(1, 0); // c
glVertex2i(0, 1); // b
glVertex2i(1, 0); // c
glVertex2i(1, 1); // d
glVertex2i(0, 1); // b
glVertex2i(1, 1); // d
glVertex2i(2, 0); // e
glEnd();
```

**Triangle Strip**

```c
glBegin(GL_TRIANGLE_STRIP);
glVertex2i(0, 0); // a
glVertex2i(0, 1); // b
glVertex2i(1, 0); // c
glVertex2i(1, 1); // d
glVertex2i(2, 0); // e
glEnd();
```
Attributes

- **Point**
  - Point size: \texttt{glPointSize}(2.0);
  - Point color: \texttt{glColor3f}(0.0, 0.0, 1.0);

- **Line**
  - Line width: \texttt{glLineWidth}(2.0);
  - Line color: \texttt{glColor3f}(0.0, 0.0, 1.0);

- **Face**
  - Front and/or back: \texttt{GL_FRONT, GL_BACK, GL_FRONT_AND_BACK}
  - Face color: \texttt{glColor3f}(0.0, 0.0, 1.0);

GLUT Callback Registration

- \texttt{glutDisplayFunc} (void (*func) (void))

- \texttt{glutReshapeFunc} (void (*func) (int width, int height))

- \texttt{glutKeyboardFunc} (void (*func) (unsigned char key, int x, int y))
  - Mouse position (x, y) when key was pressed

- \texttt{glutMouseFunc} (void (*func) (int button, int state, int x, int y))
  - Button: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON, GLUT_RIGHT_BUTTON
  - State: GLUT_UP, GLUT_DOWN
  - Position (x, y): window relative coordinates
GLUT Callback Registration

- `glutMotionFunc` (void (*func) (int x, int y))
  - Mouse motion while pressed
- `glutPassiveMotionFunc` (void (*func) (int width, int height))
  - Mouse motion without button press
- `glutIdleFunc` (void (*func) (void))
  - Called whenever no other events are on the event queue
  - Passing NULL disables this
- `glutTimerFunc` (unsigned int msecs, void (*func) (int value), value))
  - Callback every msecs milliseconds (or more): Best effort
  - Function func called with the specified value parameter
  - Can register multiple timer functions