

Using OpenGL

CSCI 4229/5229

Computer Graphics

Fall 2026

Event Driven Programming

- Don't call us, we'll call you
 - register callbacks corresponding to events
 - similar to interrupt driven programs
- DO NOT explicitly call `display()`
 - request redisplay using `glutPostRedisplay()`
- NEVER call `sleep()`
 - use global/static variables and wall time for timing
 - use `glutTimerFunc()` for regular events
- Return control as soon as possible

Types of Objects

- glBegin(type)
 - GL_POINTS points
 - GL_LINES lines between pairs of points
 - GL_LINE_STRIP series of line segments
 - GL_LINE_LOOP closed GL_LINE_STRIP
 - GL_POLYGON simple polygon
 - GL_TRIANGLES triangles between triples of points
 - GL_TRIANGLE_STRIP series of triangles
 - GL_TRIANGLE_FAN fan of triangles
 - GL_QUADS quadrangles between quads of points
 - GL_QUAD_STRIP series of quadrangles
- Set coordinates with glVertex
- glEnd()

Qualifiers

- `glPointSize(float size)`
 - POINT size in pixels (default 1)
- `glLineWidth(float width)`
 - LINE width in pixels (default 1)
- `glLineStipple(int factor,unsigned short pattern)`
 - LINE type
 - Requires `glEnable(GL_LINE_STIPPLE)`

Color

- Default is RGB color
 - X11 TrueColor
 - R,G,B 0-1 or integer range
 - glColor3f(1.0 , 0.0 ,0.0)
 - glColor3b(127 , 0 , 0);
 - glColor3ub(255 , 0 , 0);
 - glColor3fv(rgbarray);
- Color can also contain transparency (alpha)
 - glColor4f(1.0 , 0.0 , 0.0 , 0.5);
 - Default alpha=1 (opaque)
- Stays in effect until you change color

Indexed Color

- X11 Direct Color
 - Based on a colormap
- Set color using `glIndexi(27)`
- Need to load colors into color map using `glutSetColor()`
- Use RGB color unless hardware constrain
- Deprecated in OpenGL 3 since it really is obsolete

Displaying a scene

- Register using `glutDisplayFunc()`
- `glClear()`
- *Draw Something*
- `glFlush()`
- `glutSwapBuffers()`
- Schedule using `glutPostRedisplay()`

Transformations

- Transformation apply to everything that follows
- Transformations are cumulative
 - Call `glLoadIdentity()` in `display()`
- Primitive operations
 - `glLoadIdentity();`
 - `glTranslate[fd](dx , dy , dz)`
 - `glScale[fd](Sx , Sy , Sz)`
 - `glRotate[fd](angle , Ux , Uy , Uz)`
- Compatibility profile in OpenGL4 still useful

glTranslate[fd](dx , dy , dz);

- Move an object in three dimensions
- Allows you to easily produce multiple copies of an object
- Always takes 3D coordinates (float or double)

glScale[fd](Sx , Sy , Sz)

- Change the scale along the axes
- Multiplicative factors
 - $|S| < 1$ shrink
 - $|S| > 1$ expand
 - Negative values creates mirror image
- Allows you to easily create multiple copies of the same type at different sizes

glRotate[fd](angle , Ux , Uy , Uz)

- Rotates around the origin and axis (Ux,Uy,Uz)
- Angle is measured in degrees
- The axis often a primary axis, but may be any axis
 - Undefined behavior if $Ux=Uy=Uz=0$
- Allows you to create multiple copies of the same object viewed from different sides, or to view the scene from different positions

Temporary Transformations

- `glPushMatrix()`
 - Saves the current transformation
- `glPopMatrix()`
 - Resets the transformation to what it was when you did the push
- Allows you to build complex transformations and then get them back

Compound Transformations

- Rotate angle around the point (X,Y,Z) and axis (Ux,Uy,Uz)
 - `glTranslated(-X,-Y,-Z)`
 - `glRotated(angle,Ux,Uy,Uz)`
 - `glTranslated(X,Y,Z)`
- OpenGL does this intelligently

Projections

- Orthographic
 - `glOrtho(left,right,bottom,top,near,far)`
 - Same size regardless of distance
 - Easiest to use
- Perspective
 - `glFrustum(left,right,bottom,top,near,far)`
 - Closer objects are bigger
 - GLU convenience functions
 - `gluPerspective(fov,aspect,Znear,Zfar)`
 - `gluLookAt(Ex,Ey,Ez , Cx,Cy,Cz , Ux,Uy,Uz)`

Text

- OpenGL provides only hooks for fonts
- Stroked fonts
 - Lines and fills write the characters
- Bitmap (raster) fonts
 - Characters are raster images
- Orientation, size, etc. treated just like any other drawing elements

Text using GLUT

- `glutBitmapCharacter(GLUT_FONTTYPE,ch)`
 - Single character
 - Limited font selection
- `glRasterPos3d(x,y,z)`
 - Sets position to write text in (x,y,z) coordinates
- `glWindowPos2i(x,y)`
 - Sets position to write text in pixels coordinates

Registering Callbacks

- Display
 - `glutDisplayFunc()` Draw the scene
 - `glutReshapeFunc()` Window resized
 - `glutIdleFunc()` Nothing more scheduled
- User input
 - `glutKeyboardFunc()` Key pressed
 - `glutSpecialFunc()` Special key pressed
 - `glutMouseFunc()` Mouse button
 - `glutMotionFunc()` Mouse motion
- Many more

Keyboard Input

- `special(int key,int x,int y)`
 - Cursor keys `GLUT_KEY_LEFT`, `GLUT_KEY_UP`,...
 - Function keys `GLUT_KEY_Fx`
 - Basically anything not an ASCII key
- `keyboard(char ch,int x,int y)`
 - Regular ASCII keystrokes
- `(x,y)` is the mouse position in pixels

Setting Modes

- `glutInitDisplayMode`
 - Interfaces with the window manager to get the right kind of window (BE CAREFUL ABOUT DEFAULTS)
- `glEnable()` & `glDisable()`
 - Switches OpenGL into various modes
 - `GL_DEPTH_TEST`
 - `GL_ALPHA_TEST`
 - `GL_CULL_FACE`
 - `GL_LIGHTING`
 - Different modes for different objects

Checking for Errors

- OpenGL fails silently
- Functions do not return an error code
- `glGetError()` must be called explicitly to check for errors
- A black screen is a sure signal of an error