

Additional OpenGL Tidbits

- Function naming conventions
 - ◆ Prefixes: *gl**, *glu**, *glut**, *GL**, *GLU**, *GLUT** — you know about those already
 - ◆ Postfixes: OpenGL functions have explicit variants for different number types, such as integers (*i*), floats (*f*), and doubles (*d*)
 - ◆ Many functions also take arrays, a.k.a. vectors (*v*)
 - ◆ A final variant: some functions have versions with different numbers of arguments (2, 3, 4)

- So, *function_name* ← *base* [*arg_count*] [*arg_type*] [*v*]
 - ◆ Thus, for the *glColor()* base function, we can have...
 - glColor3s*(*GLshort red, GLshort green, GLshort blue*)
 - glColor3ub*(*GLubyte red, GLubyte green, GLubyte blue*)
 - glColor4b*(*GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha*)
 - GLColor4dv*(*const GLdouble *v*) — where *v* is expected to point to a 4-element array of doubles representing RGBA
 - ◆ Ditto with *glVertex()*...
 - glVertex2d*(*GLdouble x, GLdouble y*) — 2D is a special case of 3D where *z = 0*
 - glVertex2iv*(*const GLint *v*) — where *v[0]* is *x* and *v[1]* is *y*
 - glVertex3fv*(*const GLfloat *v*)
 - glVertex4i*(*GLint x, GLint y, GLint z, GLint w*)
 - ◆ *glTranslate()* has two variants, *glTranslatef*(*GLfloat x, GLfloat y, GLfloat z*) and *glTranslated()* with *GLdoubles*
- See Table I-1 in the red book for the full list

GLUT Prefab Shapes

- GLUT has a number of functions for ready-made shapes — useful for experimentation and prototyping
- All shapes paint at the origin; translate, rotate, and scale as needed to customize them
- Since the shapes are “turnkey” functions, you can’t interject new settings while drawing them, such as changes to color, material, or texture mapping
- Thus, you will probably outgrow them eventually

glutWireSphere(GLdouble radius, GLint slices, GLint stacks)

glutSolidSphere(GLdouble radius, GLint slices, GLint stacks)

glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks)

glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks)

glutWireCube(GLdouble size)

glutSolidCube(GLdouble size)

glutWireTorus(GLdouble innerRadius, GLdouble outerRadius, GLint sides, GLint rings)

glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint sides, GLint rings)

glutWireDodecahedron(void)

glutSolidDodecahedron(void)

glutWireTeapot(GLdouble size)

glutSolidTeapot(GLdouble size)

glutWireOctahedron(void)

glutSolidOctahedron(void)

glutWireTetrahedron(void)

glutSolidTetrahedron(void)

glutWireIcosahedron(void)

glutSolidIcosahedron(void)

OpenGL Settings and Switches

- OpenGL has tons of switches and settings — true to the state machine metaphor, these values start with reasonable defaults and can be changed at any time
- Some settings have their own functions: *glColor*, *glTexImage2D*, *glMatrixMode*, *glViewport*, *glNormal*
- Some settings are simple on/off toggles; touching these involve two functions (*glEnable*, *glDisable*) plus a constant indicating what “switch” should be toggled (*GL_CULL_FACE*, *GL_LINE_SMOOTH*, *GL_LIGHT0*, and many, many more)

Optimizations

- Some functions in OpenGL are devoted to speed and optimization — we won’t cover these in detail in class, since they’re fairly straightforward
- The red book contains all of the details and sample code that you’ll need
- To name a couple, to be found in Chapters 2 and 7 of the red book, respectively:
 - ◊ *Vertex arrays* enable setting vertices “in bulk”
 - ◊ *Display lists* can “cache” drawing operations