



OpenGL programming

- for a programmer the OpenGL API is `gl.h`
- the other libraries are `glu.h` and `glut.h` typically located in `include/GL/`
- The OpenGL drivers under Windows are in `windows/system32/opengl32.dll` as well as `glu32.dll` and `glut32.dll`



OpenGL program example

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv); // init OpenGL
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
    glutCreateWindow("Rotating teapot"); //make a window
    glutDisplayFunc(Display); //main callback
    glutReshapeFunc(Reshape); //reshape callback
    glutMainLoop(); //main event loop
    return 0;
}
```



OpenGL program example (contd.)

```
#include <GL/glut.h>
#include <stdlib.h>
//no need to include more - it is already included in glut.h

void Reshape(int w, int h)
{
    glViewport(0, 0, w, h); //set some strange transformations
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1, 1, -1, 1, -10, 10);
}
```



OpenGL program example (contd.)

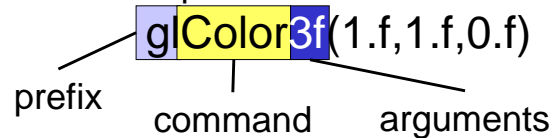
```
void Display(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //clear the window
    glutWireTeapot(0.6); //display teapot
    glFlush(); //make sure it is on the screen
}
```



Command syntax

<i>type</i>	<i>prefix</i>	<i>example</i>
procedure call	gl	glFogf()
constant	GL_	GL_FOG_COLOR
data type	GL	GLfloat

another example:



Suffix	Data type	typical C	OpenGL definition
b	8bit integer	signed char	GLbyte
s	16bit integer	short	GLshort
i	32bit integer	int	GLint, GLsizei
ub	8bit uns.int	unsigned char	GLubyte, GLboolean
us	16bit uns.int	unsigned short	GLushort
ui	32bit uns.int	unsigned long	GLuint, GLenum, GLbitfield
f	32bit float	float	GLfloat, GLclampf
d	64bit float	double	GLdouble, GLclampd



The “v” syntax

- some commands end with “v”
indicating the argument is a pointer to vector

GLfloat r=1.f, g=1.f, b=1.f; glColor3f(r,g,b);	GLfloat x[]={1,1,1}; glColor3fv(x);
---	--

Both have the same effect

What is the difference?



The wildcard agreement

a set of command with the same effect and different arguments, for example

```

glVertex3i(x,y,z)
glVertex4f(a,b,c,d)
glVertex2ub(ub1,ub2)
glVertex3dv(dv)
etc.
  
```

Will be called [glVertex*\(\)](#)

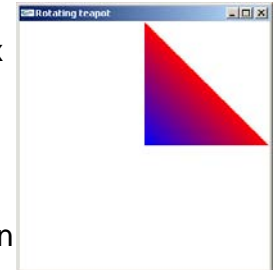


- GLUT opening windows and a basic interaction
- GLU set of useful things (fonts, NURBS, etc.)
- GL the OpenGL itself
- each program makes two things
 - sets some states (glOrtho() sets transforms)
 - feeds OpenGL with data (glColor, glVertex)
- OpenGL displays the frame buffer in the window



OpenGL program example

```
void Display(void)
{
    glShadeModel(GL_SMOOTH); //color interpolation
    glClearColor(1,1,1,1); //color of the background
    glClear(GL_COLOR_BUFFER_BIT); //clear the window
    glBegin(GL_TRIANGLES); //changing the PVPA state
        glColor3ub(0,0,255); //blue color
        glVertex3f(0,0,0); //blue vertex
        glColor3ub(255,0,0); //red color
        glVertex3f(1,0,0); //red vertex
        glVertex3f(0,1,0); //red vertex
    glEnd();
    glFlush(); //make sure it is on the screen
}
```



- Jackie Neider, Tom Davis, Mason Woo
OpenGL Programming Guide,
Addison-Wesley Publication Company
ON LINE at <http://www.opengl.org.ru/docs/>
- www.opengl.org/developers/code/tutorials.html
- SIGGRAPH 2001
An Interactive Introduction To OpenGL Programming
www.opengl.org/developers/code/s2001/index.html
- SIGGRAPH '99
Lighting and Shading Techniques for Interactive Applications
www.opengl.org/developers/code/sig99/index.html