-lopengl32  
-lfreeglut  
-lglu32

#include <GL/gl.h>

#include <GL/glut.h>

void display(void){

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (1.0, 1.0, 1.0);

// glBegin();

//

// glEnd();

glFlush ();

}

void init (void) {

glClearColor (0.0, 0.0, 0.0, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-200.0, 200.0, -200.0, 200.0, -1.0, 1.0);

}

int main(int argc, char\*\* argv){

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow ("demo");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

-lopengl32  
-lfreeglut  
-lglu32

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

void Draw(){

glPushMatrix();

glutWireTeapot (1.0);

glPopMatrix();

}

void display(void){

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (1.0, 1.0, 1.0);

glLoadIdentity ();

gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

Draw();

glFlush ();

}

void reshape (int w, int h){

glViewport (0, 0, (GLsizei) w, (GLsizei) h);

glMatrixMode (GL\_PROJECTION);

glLoadIdentity ();

glFrustum (-1.5, 1.5, -1.5, 1.5, 1.5, 20.0);

glMatrixMode (GL\_MODELVIEW);

}

void init(void){

glClearColor (0.0, 0.0, 0.0, 0.0);

glShadeModel (GL\_FLAT);

}

int main(int argc, char\*\* argv){

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow (argv[0]);

init ();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutMainLoop();

return 0;

}

-lopengl32

-lfreeglut

-lglu32

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

void Draw(){

glPushMatrix();

glutSolidSphere (1.0, 20, 16);

glPopMatrix();

}

void display(void){

glClear (GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity ();

gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

Draw();

glFlush ();

}

void reshape (int w, int h){

glViewport (0, 0, (GLsizei) w, (GLsizei) h);

glMatrixMode (GL\_PROJECTION);

glLoadIdentity ();

glFrustum (-1.5, 1.5, -1.5, 1.5, 1.5, 20.0);

glMatrixMode (GL\_MODELVIEW);

}

void init(void){

GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat mat\_shininess[] = { 50.0 };

GLfloat light\_position[] = { 1.0, 1.0, 1.0, 0.0 };

glClearColor (0.0, 0.0, 0.0, 0.0);

glShadeModel (GL\_SMOOTH);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mat\_shininess);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

}

int main(int argc, char\*\* argv){

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow (argv[0]);

init ();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutMainLoop();

return 0;

}