/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include<windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <string>

#include <stdlib.h>

#include <iostream>

#define ESCAPE 27

#define HIGH 1

#define LOW 0.2

#define RED 0

#define YELLOW 1

#define GREEN 2

#define drawOneLine(x1,y1,x2,y2) glBegin(GL\_LINES); \

glVertex2f ((x1),(y1)); glVertex2f ((x2),(y2)); glEnd();

using namespace std;

GLint crx = -500, blx = -300;

int view=0;

bool rl1[] = {false, false, true};

bool rl2[] = {false, false, true};

void text(int x, int y, string s, int font) {

int i=0;

glColor3f(0.0,0.0,0.8);

glRasterPos2f(x,y);

for(i=0;i<s.length();i++) {

if(font==1)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24,s[i]);

else if(font==2)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,s[i]);

else if(font==3) {

glColor3f(1.0,0.0,0.0);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,s[i]);

}

}

}

void First\_win() {

glClearColor(1.0,1.0,0.6,1.0);

text(290,700,"TRAFFIC LIGHT SIMULATOR",1);

text(390,660,"Using OpenGL",1);

text(430,620,"Made By:",2);

text(300,580,"Zeya Umayya & Zishan Ahmad",1);

text(400,540,"INSTRUCTIONS::",2);

text(100,500,"For Left Traffic Light",2);

text(100,460,"Q -- Red Light",2);

text(100,420,"W -- Yellow Light",2);

text(100,380,"R -- Green Light",2);

text(560,500,"For Right Traffic Light",2);

text(600,460,"A -- Red Light",2);

text(600,420,"S -- Yellow Light",2);

text(600,380,"D -- Green Light",2);

text(200,100,"!!!!PRESS SPACEBAR TO ENTER INTO THE ARENA!!!!",3);

glutPostRedisplay();

//glutSwapBuffers();

}

void init() {

glClearColor(0, 0, 0, 0);

glPointSize(5.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0,1000,0.0,800,50.0,-50.0);

glutPostRedisplay();

}

void car\_chalao() {

if(rl1[GREEN] || crx > 0)

crx += 3;

if(!rl1[GREEN] && (crx > -100 && crx < 0)) {

crx -= 3;

}

if(rl1[RED] && crx < -100)

crx += 3;

if(rl1[YELLOW] && (crx > 0 || crx < -100))

crx += 3;

if(crx > 1200) {

crx = -300;

}

}

void bus\_chalao() {

// blx -= 2;

if(rl2[GREEN] || blx > 700) {

blx -=2;

}

if(!rl2[GREEN] && (blx < -650 && blx > -720)) {

blx += 2;

}

if((rl2[RED] || rl2[YELLOW]) && (blx > -650 || blx < -720)) {

blx -= 2;

}

if(blx < -1650) {

blx = -300;

}

}

void bus() {

glPushMatrix();

glTranslated(blx,-100,0);

glScaled(50.0,50.0,0.0);

glColor3f(0.5,0.0,0.0);

//bus out line

glBegin(GL\_POLYGON);

glVertex2f(25,8);

glVertex2f(25,9.5);

glVertex2f(26,11);

glVertex2f(32,11);

glVertex2f(32,8);

glEnd();

//window frame

glColor3f(0,0.1,1);

glBegin(GL\_POLYGON);

glVertex2f(26.1,9.5);

glVertex2f(26.1,10.5);

glVertex2f(31.8,10.5);

glVertex2f(31.8,9.5);

glEnd();

//Doors

glColor3f(0,0.8,1);

glBegin(GL\_POLYGON);

glVertex2f(26.2,9);

glVertex2f(26.2,10.4);

glVertex2f(27.7,10.4);

glVertex2f(27.7,9);

glEnd();

glColor3f(1,1,1);

glBegin(GL\_POLYGON);

glVertex2f(27,8.4);

glVertex2f(27,10.4);

glVertex2f(27.7,10.4);

glVertex2f(27.7,8.4);

glEnd();

//small windows

glColor3f(0,1,1);

glBegin(GL\_POLYGON);

glVertex2f(27.8,9.6);

glVertex2f(27.8,10.4);

glVertex2f(29,10.4);

glVertex2f(29,9.6);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(29.1,9.6);

glVertex2f(29.1,10.4);

glVertex2f(30.2,10.4);

glVertex2f(30.2,9.6);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(30.3,9.6);

glVertex2f(30.3,10.4);

glVertex2f(31.7,10.4);

glVertex2f(31.7,9.6);

glEnd();

//driver window

glColor3f(0,0.8,1);

glBegin(GL\_POLYGON);

glVertex2f(25,9.5);

glVertex2f(26,11);

glVertex2f(26,9.5);

glEnd();

glPopMatrix();

//tyre

glPushMatrix();//front tyre

glTranslated(blx+1220,250,0.0);

glScaled(20.0,20.0,0.0);

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);

glVertex2f(3.0,2.5);

glVertex2f(3.0,2.6);

glVertex2f(3.15,3.1);

glVertex2f(3.2,3.2);

glVertex2f(3.3,3.35);

glVertex2f(3.4,3.4);

glVertex2f(3.5,3.45);

glVertex2f(3.6,3.55);

glVertex2f(3.7,3.6);

glVertex2f(3.8,3.63);

glVertex2f(4.0,3.65);

glVertex2f(4.2,3.7);

glVertex2f(4.4,3.7);

glVertex2f(4.6,3.65);

glVertex2f(4.8,3.55);

glVertex2f(5.0,3.45);

glVertex2f(5.1,3.4);

glVertex2f(5.2,3.25);

glVertex2f(5.3,3.2);

glVertex2f(5.4,3.0);

glVertex2f(5.5,2.5);

glVertex2f(5.45,2.15);

glVertex2f(5.4,1.9);

glVertex2f(5.35,1.8);

glVertex2f(5.2,1.6);

glVertex2f(5.0,1.5);

glVertex2f(4.9,1.4);

glVertex2f(4.7,1.3);

glVertex2f(4.6,1.27);

glVertex2f(4.4,1.25);

glVertex2f(4.0,1.25);

glVertex2f(3.9,1.3);

glVertex2f(3.75,1.35);

glVertex2f(3.6,1.4);

glVertex2f(3.45,1.55);

glVertex2f(3.3,1.7);

glVertex2f(3.2,1.8);

glVertex2f(3.1,2.2);

glEnd();

glPopMatrix();

glPushMatrix();//back tyre

glTranslated(blx+1460,250,0.0);

glScaled(20.0,20.0,0.0);

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);

glVertex2f(3.0,2.5);

glVertex2f(3.0,2.6);

glVertex2f(3.15,3.1);

glVertex2f(3.2,3.2);

glVertex2f(3.3,3.35);

glVertex2f(3.4,3.4);

glVertex2f(3.5,3.45);

glVertex2f(3.6,3.55);

glVertex2f(3.7,3.6);

glVertex2f(3.8,3.63);

glVertex2f(4.0,3.65);

glVertex2f(4.2,3.7);

glVertex2f(4.4,3.7);

glVertex2f(4.6,3.65);

glVertex2f(4.8,3.55);

glVertex2f(5.0,3.45);

glVertex2f(5.1,3.4);

glVertex2f(5.2,3.25);

glVertex2f(5.3,3.2);

glVertex2f(5.4,3.0);

glVertex2f(5.5,2.5);

glVertex2f(5.45,2.15);

glVertex2f(5.4,1.9);

glVertex2f(5.35,1.8);

glVertex2f(5.2,1.6);

glVertex2f(5.0,1.5);

glVertex2f(4.9,1.4);

glVertex2f(4.7,1.3);

glVertex2f(4.6,1.27);

glVertex2f(4.4,1.25);

glVertex2f(4.0,1.25);

glVertex2f(3.9,1.3);

glVertex2f(3.75,1.35);

glVertex2f(3.6,1.4);

glVertex2f(3.45,1.55);

glVertex2f(3.3,1.7);

glVertex2f(3.2,1.8);

glVertex2f(3.1,2.2);

glEnd();

glPopMatrix();

}

void car() {

glPushMatrix(); //making color for outer line

glLineWidth(1.0);

glTranslated(crx,400.0,0.0);

glScaled(20.0,20.0,0.0);

glColor3f(0.2,0.3,1.0);

glBegin(GL\_POLYGON);

glVertex2f(2.5,2.5);

glVertex2f(3.0,3.5);

glVertex2f(3.5,3.75);

glVertex2f(4.0,4.0);

glVertex2f(4.5,4.0);

glVertex2f(5.0,3.75);

glVertex2f(5.5,3.5);

glVertex2f(5.75,3.0);

glVertex2f(6.0,2.5);

glVertex2f(16.5,2.5);

glVertex2f(16.75,3.0);

glVertex2f(17.0,3.5);

glVertex2f(17.5,3.75);

glVertex2f(18.0,4.0);

glVertex2f(18.5,4.0);

glVertex2f(19.0,3.75);

glVertex2f(19.5,3.5);

glVertex2f(19.75,3.0);

glVertex2f(20.0,2.5);

glVertex2f(21.0,2.5);

glVertex2f(21.0,4.0);

glVertex2f(21.5,4.0);

glVertex2f(21.0,4.5);

glVertex2f(20.0,5.0);

glVertex2f(15.0,5.0);

glVertex2f(14.0,5.5);

glVertex2f(13.0,6.0);

glVertex2f(12.0,6.5);

glVertex2f(11.0,7.0);

glVertex2f(6.0,7.0);

glVertex2f(5.0,6.5);

glVertex2f(4.5,6.25);

glVertex2f(4.25,6.0);

glVertex2f(4.0,5.75);

glVertex2f(3.5,5.5);

glVertex2f(3.0,5.5);

glVertex2f(1.9,5.45);

glVertex2f(1.8,5.4);

glVertex2f(1.7,5.35);

glVertex2f(1.6,5.3);

glVertex2f(1.5,5.25);

glVertex2f(1.4,5.15);

glVertex2f(1.3,5.0);

glVertex2f(1.2,4.85);

glVertex2f(1.1,4.7);

glVertex2f(1.0,4.3);

glVertex2f(1.0,3.2);

glVertex2f(1.1,3.05);

glVertex2f(1.2,2.9);

glVertex2f(1.3,2.9);

glVertex2f(1.4,2.75);

glVertex2f(1.5,2.65);

glVertex2f(1.6,2.6);

glVertex2f(1.7,2.55);

glVertex2f(1.8,2.5);

glVertex2f(1.9,2.45);

glVertex2f(2.0,2.5);

glEnd();

glColor3f(1.0,1.0,1.0); //color for outer window

glBegin(GL\_POLYGON);

glVertex2f(5.0,5.0);

glVertex2f(14.0,5.0);

glVertex2f(11.5,6.5);

glVertex2f(10.5,6.75);

glVertex2f(7.0,6.75);

glEnd();

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON); //drawing a back tyre

glVertex2f(3.0,2.5);

glVertex2f(3.0,2.6);

glVertex2f(3.15,3.1);

glVertex2f(3.2,3.2);

glVertex2f(3.3,3.35);

glVertex2f(3.4,3.4);

glVertex2f(3.5,3.45);

glVertex2f(3.6,3.55);

glVertex2f(3.7,3.6);

glVertex2f(3.8,3.63);

glVertex2f(4.0,3.65);

glVertex2f(4.2,3.7);

glVertex2f(4.4,3.7);

glVertex2f(4.6,3.65);

glVertex2f(4.8,3.55);

glVertex2f(5.0,3.45);

glVertex2f(5.1,3.4);

glVertex2f(5.2,3.25);

glVertex2f(5.3,3.2);

glVertex2f(5.4,3.0);

glVertex2f(5.5,2.5);

glVertex2f(5.45,2.15);

glVertex2f(5.4,1.9);

glVertex2f(5.35,1.8);

glVertex2f(5.2,1.6);

glVertex2f(5.0,1.5);

glVertex2f(4.9,1.4);

glVertex2f(4.7,1.3);

glVertex2f(4.6,1.27);

glVertex2f(4.4,1.25);

glVertex2f(4.0,1.25);

glVertex2f(3.9,1.3);

glVertex2f(3.75,1.35);

glVertex2f(3.6,1.4);

glVertex2f(3.45,1.55);

glVertex2f(3.3,1.7);

glVertex2f(3.2,1.8);

glVertex2f(3.1,2.2);

glEnd();

glBegin(GL\_POLYGON); //drawing front tyre

glVertex2f(17.0,2.5);

glVertex2f(17.0,2.6);

glVertex2f(17.15,3.1);

glVertex2f(17.2,3.2);

glVertex2f(17.3,3.35);

glVertex2f(17.4,3.4);

glVertex2f(17.5,3.45);

glVertex2f(17.6,3.55);

glVertex2f(17.7,3.6);

glVertex2f(17.8,3.63);

glVertex2f(18.0,3.65);

glVertex2f(18.2,3.7);

glVertex2f(18.4,3.7);

glVertex2f(18.6,3.65);

glVertex2f(18.8,3.55);

glVertex2f(19.0,3.45);

glVertex2f(19.1,3.4);

glVertex2f(19.2,3.25);

glVertex2f(19.3,3.2);

glVertex2f(19.4,3.0);

glVertex2f(19.5,2.5);

glVertex2f(19.45,2.15);

glVertex2f(19.4,1.9);

glVertex2f(19.35,1.8);

glVertex2f(19.2,1.6);

glVertex2f(19.0,1.5);

glVertex2f(18.9,1.4);

glVertex2f(18.7,1.3);

glVertex2f(18.6,1.27);

glVertex2f(18.4,1.25);

glVertex2f(18.0,1.25);

glVertex2f(17.9,1.3);

glVertex2f(17.75,1.35);

glVertex2f(17.6,1.4);

glVertex2f(17.45,1.55);

glVertex2f(17.3,1.7);

glVertex2f(17.2,1.8);

glVertex2f(17.1,2.2);

glEnd();

glPopMatrix();

}

void traffic\_light() {

//traffic signal controller

glPushMatrix();

glTranslatef(-190,20,0);

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);

// glColor3f(0.7,0.3,0.0);

glVertex2d(600,350); //1

glVertex2d(650,350); //2

glVertex2d(650,370); //3

glVertex2d(600,370); //12

glEnd();

glPopMatrix();

glPushMatrix();

glTranslatef(-190,20,0);

glBegin(GL\_POLYGON);

// glColor3f(0.6,0.2,0.0);

glVertex2d(630,370); //4

glVertex2d(630,520); //5

glVertex2d(620,520); //10

glVertex2d(620,370); //11

glEnd();

glPopMatrix();

// Left Red Light

glPushMatrix();

glTranslatef(-190,20,0);

glBegin(GL\_POLYGON);

// glColor3f(0.7,0.3,0.0);

glVertex2d(600,450); //6

glVertex2d(600,600); //7

glVertex2d(550,600); //8

glVertex2d(550,450); //9

glEnd();

glPopMatrix();

//Right Red Light

glPushMatrix();

glTranslatef(-190,20,0);

glBegin(GL\_POLYGON);

// glColor3f(0.7,0.3,0.0);

glVertex2d(700,450); //11

glVertex2d(700,600); //12

glVertex2d(650,600); //13

glVertex2d(650,450); //14

glEnd();

glPopMatrix();

//Connecting Rod

glPushMatrix();

glTranslatef(-190,20,0);

glBegin(GL\_POLYGON);

// glColor3f(0.7,0.3,0.0);

glVertex2d(650,520); //15

glVertex2d(650,540); //16

glVertex2d(600,540); //17

glVertex2d(600,520); //18

glEnd();

glPopMatrix();

//Left Lights

//Red 1

glPushMatrix();

glTranslatef(385, 580, 0);

if(rl1[RED])

glColor3f(1.0, 0.0, 0.0);

else

glColor3f(0.2, 0.0, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

//Yellow 1

glPushMatrix();

glTranslatef(385, 540, 0);

if(rl1[YELLOW])

glColor3f(1.0, 1.0, 0.0);

else

glColor3f(0.2, 0.2, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

//Green 1

glPushMatrix();

glTranslatef(385, 500, 0);

if(rl1[GREEN])

glColor3f(0.0, 1.0, 0.0);

else

glColor3f(0.0, 0.2, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

//Right Lights

//Red 2

glPushMatrix();

glTranslatef(485, 580, 0);

if(rl2[RED])

glColor3f(1.0, 0.0, 0.0);

else

glColor3f(0.2, 0.0, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

//Yellow 2

glPushMatrix();

glTranslatef(485, 540, 0);

if(rl2[YELLOW])

glColor3f(1.0, 1.0, 0.0);

else

glColor3f(0.2, 0.2, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

//Green 2

glPushMatrix();

glTranslatef(485, 500, 0);

if(rl2[GREEN])

glColor3f(0.0, 1.0, 0.0);

else

glColor3f(0.0, 0.2, 0.0);

glutSolidSphere(12, 80, 80);

glPopMatrix();

}

void gra(int x,int y,int p,int q,float r,float g,float b)

{

glPushMatrix();

glColor3f(r,g,b);

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2d(x,y);

glVertex2d(p,q);

glEnd();

glPopMatrix();

}

void grass()

{

float r=1.0,g=0.0,b=0.8;

for(int k=0;k<300;k+=50) ////left grass

{

if(r>0)

r-=.2;

else

r=1.0;

if(g<1)

g+=.2;

else

g=0.0;

if(b>0)

b-=.2;

else

b=1.0;

for(int i=0, j=525,c=0;c<5;i+=15,j-=5,c++)

gra(i+k,525,30+k,500,r,g,b);

}

for(int k=0;k<500;k+=50) //right grass

{

if(r>0)

r-=.2;

else

r=1.0;

if(g<1)

g+=.2;

else

g=0.0;

if(b>0)

b-=.2;

else

b=1.0;

for(int i=500,c=0;c<5;i+=15,c++)

gra(i+k,525,530+k,500,r,g,b);

}

}

void white\_strips() {

//Left

glPushMatrix();

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_LINE\_STIPPLE);

glLineWidth(12.0);

glLineStipple (4, 0x0FFF);

drawOneLine (0.0, 365.0, 250.0, 365.0);

glDisable(GL\_LINE\_STIPPLE);

glPopMatrix();

//Right

glPushMatrix();

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_LINE\_STIPPLE);

glLineWidth(12.0);

glLineStipple (4, 0x0FFF);

drawOneLine (620.0, 365.0, 1500.0, 365.0);

glDisable(GL\_LINE\_STIPPLE);

glPopMatrix();

}

void zebra() {

//Left

glPushMatrix();

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_LINE\_STIPPLE);

glLineWidth(12.0);

glLineStipple (1, 0x00FF);

for(float i=360.0, j=280.0, c=0; c<10; i-=5, j-=5, c++) {

drawOneLine (i, 250.0, j, 500.0);

}

glDisable(GL\_LINE\_STIPPLE);

glPopMatrix();

//Right

glPushMatrix();

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_LINE\_STIPPLE);

glLineWidth(12.0);

glLineStipple (1, 0x00FF);

// drawOneLine (660, 250.0, 550, 500.0);

for(float i=660.0, j=550.0, c=0; c<10; i-=5, j-=5, c++) {

drawOneLine (i, 250.0, j, 500.0);

}

glDisable(GL\_LINE\_STIPPLE);

glPopMatrix();

//Top

glPushMatrix();

glColor3f(1.0, 1.0, 1.0);

glEnable(GL\_LINE\_STIPPLE);

glLineWidth(12.0);

glLineStipple (1, 0x00FF);

// drawOneLine (300, 505.0, 500, 505.0);

for(float i=550.0, j=550.0, c=0; c<10; i-=5, j-=5, c++) {

drawOneLine (300, i, 500, j);

}

glDisable(GL\_LINE\_STIPPLE);

glPopMatrix();

}

void road\_strips() {

//first bottom green strip

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0.85,0.64,0.12);

glVertex2d(0,200);

glVertex2d(0,250);

glVertex2d(1000,250);

glVertex2d(1000,200);

glEnd();

glPopMatrix();

//second grey strip

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0.5,0.5,0.5);

glVertex2d(0,250);

glVertex2d(0,500);

glVertex2d(1000,500);

glVertex2d(1000,250);

glEnd();

glPopMatrix();

//tilted grey strip

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0.5,0.5,0.5);

glVertex2d(300,500);

glVertex2d(200,800);

glVertex2d(350,800);

glVertex2d(500,500);

glEnd();

glPopMatrix();

zebra();

//Grass

grass();

//Dotted strips

white\_strips();

car();

traffic\_light();

bus();

car\_chalao();

bus\_chalao();

}

void main\_display() {

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

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

if(view==0) {

init();

First\_win();

}

else {

//---------CONSTANT PARTS

//ROAD STRIP

road\_strips();

}

glutSwapBuffers();

}

void traffic\_start() {

}

void keyboard(unsigned char key, int x, int y) {

switch(key) {

case ESCAPE:exit(1);

case ' ':

view=1;

glClearColor(0.6, 0.8, 0.2, 0);

//glColor3f(0.85,0.64,0.12);

traffic\_start();

break;

case 'q':

cout<<"Red light 1 to RED"<<endl;

rl1[RED] = true;

rl1[YELLOW] = false;

rl1[GREEN] = false;

break;

case 'w':

cout<<"Red light 1 to YELLOW"<<endl;

rl1[RED] = false;

rl1[YELLOW] = true;

rl1[GREEN] = false;

break;

case 'e':

cout<<"Red light 1 to GREEN"<<endl;

rl1[RED] = false;

rl1[YELLOW] = false;

rl1[GREEN] = true;

break;

case 'a':

cout<<"Red light 2 to RED"<<endl;

rl2[RED] = true;

rl2[YELLOW] = false;

rl2[GREEN] = false;

break;

case 's':

cout<<"Red light 2 to YELLOW"<<endl;

rl2[RED] = false;

rl2[YELLOW] = true;

rl2[GREEN] = false;

break;

case 'd':

cout<<"Red light 2 to GREEN"<<endl;

rl2[RED] = false;

rl2[YELLOW] = false;

rl2[GREEN] = true;

break;

default:

cout<<"You pressed: "<<key;

}

}

void reshape(int w,int h) {

glViewport(0,0,w,h);

cout<<"Width="<<w<<" height= "<<h<<endl;

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_RGBA|GLUT\_DEPTH|GLUT\_DOUBLE);

glutInitWindowSize(1000,800);

glutInitWindowPosition(0,0);

glutCreateWindow("Traffic Light Simulation");

//glutFullScreen();

glutDisplayFunc(main\_display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutIdleFunc(main\_display);

//glutMouseFunc();

glutMainLoop();

return 0;

}