/\*

\* 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.

\*/

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

#include <math.h>

#include <time.h>

#include <sys/timeb.h>

//#include<bits/stdc++.h>

using namespace std;

void circle(float cx=0.0, float cy=0.0, float r=5.0)

{

int num=50;

float pi=2\*3.1416;

glLineWidth(1.0);

glBegin(GL\_TRIANGLE\_FAN);

for(double i=0;i<=num;i+=(0.2))

{

float theta = pi\*float(i)/float(num);

float x= r\*cosf(theta);

float y= r\*sinf(theta);

glVertex2d(x+cx,y+cy);

}

glEnd();

}

void circle\_half(float cx=0.0, float cy=0.0, float r=2.0)

{

int num=50;

float pi=3.1416;

glLineWidth(10.0);

glBegin(GL\_TRIANGLE\_FAN);

for(double i=0;i<=num;i+=(0.2))

{

float theta = pi\*float(i)/float(num);

float x= r\*cosf(theta);

float y= r\*sinf(theta);

glColor3f(0.0,1.0,0.0);

glVertex2d(x+cx,y+cy);

}

glEnd();

}

void draw(int dis)

{ dis%=20;

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3d(0,0,0);

glBegin(GL\_LINES);

glVertex2d(0,85);

glVertex2d(640,85);

glVertex2d(0,40);

glVertex2d(640,40);

glEnd();

glColor3d(0,0,0);

glBegin(GL\_POINTS);

circle\_half(200,200,80);

glEnd();

glBegin(GL\_LINES);

glVertex2d(120,200);

glVertex2d(280,200);

glEnd();

glBegin(GL\_POINTS);

circle(150,150,20);

glEnd();

glBegin(GL\_LINES);

glVertex2d(150,130);

glVertex2d(150,78);

glEnd();

glBegin(GL\_LINES);

glVertex2d(150,78);

glVertex2d(150-dis,50);

glEnd();

glBegin(GL\_LINES);

glVertex2d(150,78);

glVertex2d(150+dis,50);

glEnd();

glBegin(GL\_LINES);

glVertex2d(200,200);

glVertex2d(200,130);

glEnd();

glBegin(GL\_LINES);

glVertex2d(150,120);

glVertex2d(170,110);

glEnd();glBegin(GL\_LINES);

glVertex2d(150,115);

glVertex2d(170,105);

glEnd();

glBegin(GL\_LINES);

glVertex2d(200,130);

glVertex2d(170,110);

glEnd();

glBegin(GL\_LINES);

glVertex2d(200,130);

glVertex2d(170,105);

glEnd();

for(int i=1;i<=1000;i++)

{

int x=rand(),y=rand();

x%=640; y%=480;

if(x>=120&&x<=280&&y<=280)continue;

glBegin(GL\_LINES);

glColor3b(1,1,1);

glVertex2d(x,y);

glVertex2d(x+5,y+5);

glEnd();

}

}

void plot(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3d(0,1,0);

glBegin(GL\_POINTS);

for(int i=1;i<=100;i++)

{

int x=rand(),y=rand();

x%=640; y%=480;

if(x>=120&&x<=280&&y<=280)continue;

glBegin(GL\_LINES);

glColor3b(1,1,1);

glVertex2d(x,y);

glVertex2d(x+5,y+5);

glEnd();

}

for(int i=1;i<=10000;i+=5)

draw(i);

glFlush();

}

void Init()

{

/\* Set clear color to white \*/

glClearColor(1.0,1.0,1.0,0);

/\* Set fill color to black \*/

glColor3f(0.0,0.0,0.0);

/\* glViewport(0 , 0 , 640 , 480); \*/

/\* glMatrixMode(GL\_PROJECTION); \*/

/\* glLoadIdentity(); \*/

gluOrtho2D(0 , 640 , 0 , 480);

}

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

{

/\* Initialise GLUT library \*/

glutInit(&argc,argv);

/\* Set the initial display mode \*/

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

/\* Set the initial window position and size \*/

glutInitWindowPosition(0,0);

glutInitWindowSize(640,480);

/\* Create the window with title "DDA\_Line" \*/

glutCreateWindow("Rain");

/\* Initialize drawing colors \*/

Init();

/\* Call the displaying function \*/

glutDisplayFunc(plot);

/\* Keep displaying untill the program is closed \*/

glutMainLoop();

return 0;

}