

**Assignment Report of Computer Graphics Lab**

Course title: Computer Graphics Lab

Course code: CSE422

Submitted by,

Name: Rashidul Hasan Hridoy

ID: 171-15-8596

Section: O3

Department of Computer Science & Engineering,

Daffodil International University

Submitted to,

[S](https://sites.google.com/view/syedatanjilaatik)yeda Tanjila Atik

Lecturer

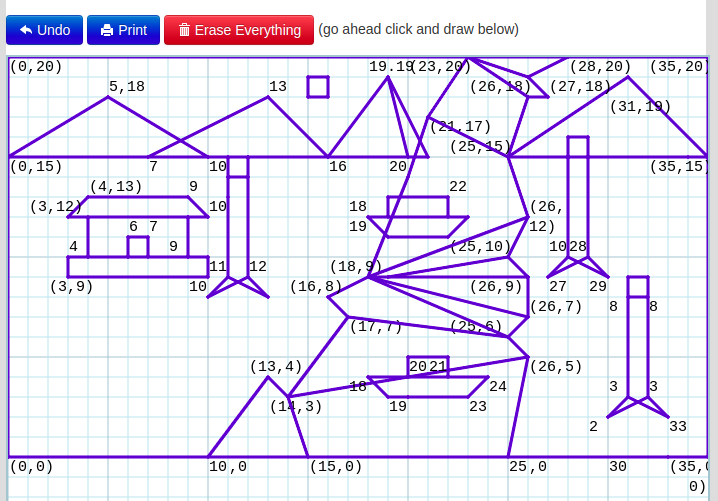
Department of Computer Science and Engineering,

Daffodil International University

Date of submission: 05-08-20

**Answer no. - 1**

**Graph**

****

**Code**

#include <GL/glut.h>

#include <math.h>

void init(void)

{

// Set display window colour to white

glClearColor(0.0, 0.0, 0.0, 0.0);

// Set projection parameters

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0, 35.0, 0.0, 20.0);

}

void rashidul(void)

{

// Clear display window

glClear(GL\_COLOR\_BUFFER\_BIT);

float theta;

int i;

//main\_box

glColor3f(0.9224, 0.93, 0.4743);

glBegin(GL\_POLYGON);

glVertex2i(0, 20);

glVertex2i(35, 20);

glVertex2i(0, 20);

glVertex2i(0, 0);

glVertex2i(0, 0);

glVertex2i(35, 0);

glVertex2i(35, 0);

glVertex2i(35, 20);

glEnd();

//second\_box

glColor3f(0.7235, 0.93, 0.2418);

glBegin(GL\_QUADS);

glVertex2i(0, 15);

glVertex2i(35, 15);

glVertex2i(35, 00);

glVertex2i(0, 0);

glEnd();

//mountain

glColor3f(0.61, 0.4209, 0.0427);

glBegin(GL\_TRIANGLES);

glVertex2i(0, 15);

glVertex2i(5, 18);

glVertex2i(10, 15);

glEnd();

glColor3f(0.61, 0.4209, 0.0427);

glBegin(GL\_TRIANGLES);

glVertex2i(7, 15);

glVertex2i(13, 18);

glVertex2i(16, 15);

glEnd();

glColor3f(0.61, 0.4209, 0.0427);

glBegin(GL\_TRIANGLES);

glVertex2i(16, 15);

glVertex2i(19, 19);

glVertex2i(21, 15);

glEnd();

glColor3f(0.61, 0.4209, 0.0427);

glBegin(GL\_TRIANGLES);

glVertex2i(25, 15);

glVertex2i(31, 19);

glVertex2i(35, 15);

glEnd();

//river\_start

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(10, 0);

glVertex2i(13, 4);

glVertex2i(14, 3);

glVertex2i(15, 0);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(15, 0);

glVertex2i(14, 3);

glVertex2i(26, 5);

glVertex2i(25, 0);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(14, 3);

glVertex2i(17, 7);

glVertex2i(25, 6);

glVertex2i(26, 5);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(17, 7);

glVertex2i(16, 8);

glVertex2i(18, 9);

glVertex2i(25, 6);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_TRIANGLES);

glVertex2i(18,9);

glVertex2i(25,6);

glVertex2i(26,7);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_TRIANGLES);

glVertex2i(18,9);

glVertex2i(26,7);

glVertex2i(26,9);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_TRIANGLES);

glVertex2i(18,9);

glVertex2i(26,9);

glVertex2i(25,16);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_TRIANGLES);

glVertex2i(18,9);

glVertex2i(25,10);

glVertex2i(26,12);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(18, 9);

glVertex2i(21, 17);

glVertex2i(25, 15);

glVertex2i(26, 12);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(21, 17);

glVertex2i(25, 15);

glVertex2i(26, 18);

glVertex2i(23, 20);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_QUADS);

glVertex2i(23, 20);

glVertex2i(26, 18);

glVertex2i(27, 18);

glVertex2i(26, 19);

glEnd();

glColor3f(0.7872, 0.96, 0.9542);

glBegin(GL\_TRIANGLES);

glVertex2i(23, 20);

glVertex2i(27, 18);

glVertex2i(28, 20);

glEnd();

//river\_end

//home\_1

glColor3f(0.98, 0.6532, 0.1274);

glBegin(GL\_QUADS);

glVertex2i(3, 12);

glVertex2i(4, 13);

glVertex2i(9, 13);

glVertex2i(10, 12);

glEnd();

glColor3f(0.4261, 0.4261, 0.0106);

glBegin(GL\_QUADS);

glVertex2i(4, 12);

glVertex2i(9, 12);

glVertex2i(9, 10);

glVertex2i(4, 10);

glEnd();

glColor3f(0.98, 0.6532, 0.1274);

glBegin(GL\_QUADS);

glVertex2i(3, 10);

glVertex2i(10, 10);

glVertex2i(10, 9);

glVertex2i(3, 9);

glEnd();

glColor3f(1, 1, 0.1274);

glBegin(GL\_QUADS);

glVertex2i(6, 10);

glVertex2i(6, 11);

glVertex2i(7, 11);

glVertex2i(7, 10);

glEnd();

//tree\_1

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_QUADS);

glVertex2i(11, 9);

glVertex2i(11, 14);

glVertex2i(12, 14);

glVertex2i(12, 9);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(10, 8);

glVertex2i(11, 9);

glVertex2i(12, 9);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(11, 9);

glVertex2i(12, 9);

glVertex2i(13, 8);

glEnd();

//round

glColor3f(0.1003, 0.43, 0);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/10;

glVertex2f(11.5+2.6\*cos(theta),14+2.6\*sin(theta));

}

glEnd();

//tree\_2

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_QUADS);

glVertex2i(31, 3);

glVertex2i(31, 8);

glVertex2i(32, 8);

glVertex2i(32, 3);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(30, 2);

glVertex2i(31, 3);

glVertex2i(32, 3);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(31, 3);

glVertex2i(32, 3);

glVertex2i(33, 2);

glEnd();

//tree\_3

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_QUADS);

glVertex2i(28, 10);

glVertex2i(28, 15);

glVertex2i(29, 15);

glVertex2i(29, 10);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(27, 9);

glVertex2i(28, 10);

glVertex2i(29, 10);

glEnd();

glColor3f(0.89, 0.3382, 0.6325);

glBegin(GL\_TRIANGLES);

glVertex2i(28, 10);

glVertex2i(29, 10);

glVertex2i(30, 9);

glEnd();

//round

glColor3f(0.1003, 0.43, 0);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/10;

glVertex2f(31.5+2.6\*cos(theta),8+2.6\*sin(theta));

}

glEnd();

//round

glColor3f(0.1003, 0.43, 0);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/10;

glVertex2f(28.5+2.6\*cos(theta),15+2.6\*sin(theta));

}

glEnd();

//boat\_1

glColor3f(0.98, 0.6532, 0.1274);

glBegin(GL\_QUADS);

glVertex2i(19, 3);

glVertex2i(18, 4);

glVertex2i(24, 4);

glVertex2i(23, 3);

glEnd();

glColor3f(0.4261, 0.4261, 0.0106);

glBegin(GL\_QUADS);

glVertex2i(19, 4);

glVertex2i(19, 5);

glVertex2i(23, 5);

glVertex2i(23, 4);

glEnd();

//boat\_2

glColor3f(0.98, 0.6532, 0.1274);

glBegin(GL\_QUADS);

glVertex2i(19, 11);

glVertex2i(18, 12);

glVertex2i(24, 12);

glVertex2i(23, 11);

glEnd();

glColor3f(0.4261, 0.4261, 0.0106);

glBegin(GL\_QUADS);

glVertex2i(19, 12);

glVertex2i(19, 13);

glVertex2i(23, 13);

glVertex2i(23, 12);

glEnd();

glFlush();

// Process all OpenGL routines

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(100, 100);

glutInitWindowSize(700, 600);

glutCreateWindow("171-15-8596");

init();

glutDisplayFunc(rashidul);

glutMainLoop();

return 0;

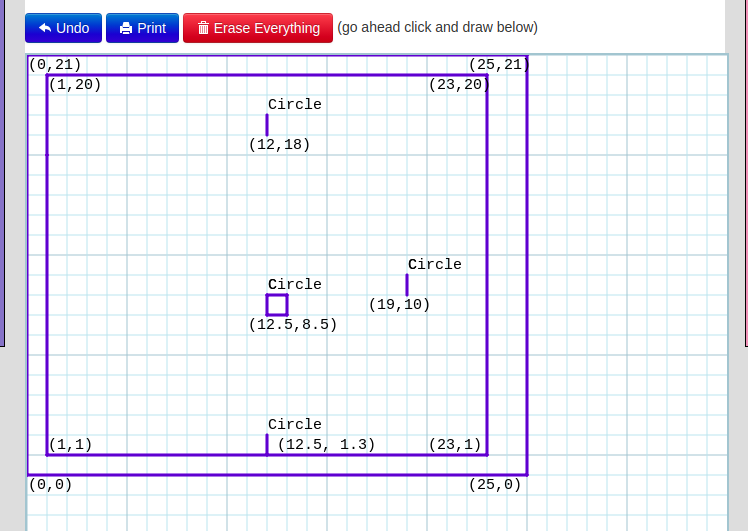
}

**Output**

****

**Answer no. - 2**

**Graph**

****

**Code**

#include <GL/glut.h>

#include <math.h>

void init(void)

{

// Set display window colour to white

glClearColor(0.0, 0.0, 0.0, 0.0);

// Set projection parameters

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0, 25.0, 0.0, 21.0);

}

void rashidul(void)

{

// Clear display window

glClear(GL\_COLOR\_BUFFER\_BIT);

float theta;

int i;

//main\_box

glColor3f(0, 0, 0);

glBegin(GL\_QUADS);

glVertex2i(0, 0);

glVertex2i(0, 21);

glVertex2i(25, 21);

glVertex2i(25, 0);

glEnd();

//second\_box

glColor3f(1, 1, 1);

glBegin(GL\_QUADS);

glVertex2i(1, 1);

glVertex2i(1, 20);

glVertex2i(23, 20);

glVertex2i(23, 1);

glEnd();

//apple

//main

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/50;

glVertex2f(12.5+6.25\*cos(theta),8.5+7.25\*sin(theta));

}

glEnd();

//1

glColor3f(1,1,1);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/180;

glVertex2f(19.0+3.5\*cos(theta),10+3.5\*sin(theta));

}

glEnd();

//2

glColor3f(1,1,1);

glBegin(GL\_POLYGON);

for(i=0;i<360;i++)

{

theta=i\*3.142/180;

glVertex2f(12.5+2\*cos(theta),16.0+1\*sin(theta));

}

glEnd();

//3

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

for(i=45;i<360;i++)

{

theta=i\*3.142/180;

glVertex2f(12+0.55\*cos(theta),18.0+1.7\*sin(theta));

}

glEnd();

//4

glColor3f(1,1,1);

glBegin(GL\_POLYGON);

for(i=45;i<360;i++)

{

theta=i\*3.142/180;

glVertex2f(12.5+3\*cos(theta),1.3+0.9\*sin(theta));

}

glEnd();

//down

glColor3f(0, 0, 0);

glBegin(GL\_QUADS);

glVertex2i(0, 0);

glVertex2i(0, 1);

glVertex2i(25, 1);

glVertex2i(25, 0);

glEnd();

glFlush();

// Process all OpenGL routines

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(100, 100);

glutInitWindowSize(600, 600);

glutCreateWindow("171-15-8596");

init();

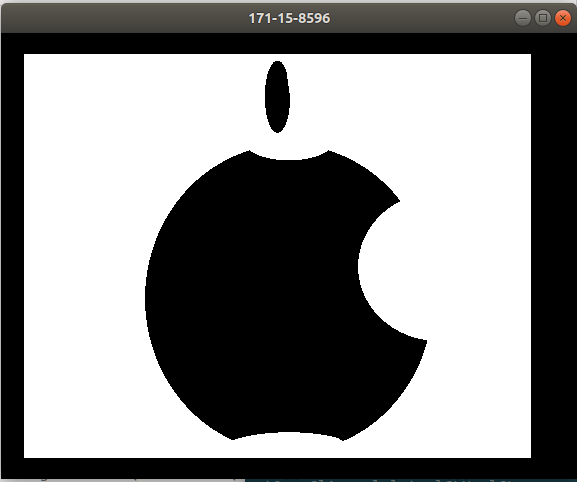
glutDisplayFunc(rashidul);

glutMainLoop();

return 0;

}

**Output**

****