**Name : vivek lakhlani**

**Roll no : 047**

**Division : A**

**Semester : 5th**

**Subject : Computer graphics (503)**

Common header file : shapes.h

**#ifndef** SHAPES\_H\_

**#define** SHAPES\_H\_

**#include**<GL/glut.h>

**#include**<math.h>

**int** a1, b1, a2, b2, k=0, h, w,xc,yc,r;

**float** da, db, steps, ai = 0, bi, a, b;

**void** **line**(**int** a1, **int** b1, **int** a2, **int** b2)

{

**float** r = 0;

da = a2 - a1;

db = b2 - b1;

**if**(**abs**(da) >= **abs**(db))

{

steps = **abs**(da);

}

**else**

{

steps = **abs**(db);

}

ai = da / steps;

bi = db / steps;

a = a1;

b = b1;

**glPointSize**(3.0);

**glBegin**(GL\_POINTS);

k = 1;

**while**(k <= steps)

{

// r = r + 1/steps;

// glColor3f(0.0, 1.0, r);

**glVertex2f**(**round**(a),**round**(b));

a = a + ai;

b = b + bi;

// printf("a = %f and b = %f\n",round(a),round(b));

k++;

}

}

**void** **square**(**int** a1, **int** b1, **int** h)

{

line(a1,b1,a1,b1+h);

line(a1,b1+h,a1+h,b1+h);

line(a1+h,b1+h,a1+h,b1);

line(a1+h,b1,a1,b1);

**glEnd**();

**glFlush**();

}

**void** **rectangle**(**int** a1, **int** b1, **int** h, **int** w)

{

line(a1,b1,a1,b1+h);

line(a1,b1+h,a1+w,b1+h);

line(a1+w,b1+h,a1+w,b1);

line(a1+w,b1,a1,b1);

**glEnd**();

**glFlush**();

}

**void** **triangle**(**int** a1, **int** b1, **int** a2, **int** b2, **int** h, **int** w)

{

line(a1,b1,a2,b2);

line(a2,b2,h,w);

line(h,w,a1,b1);

**glEnd**();

**glFlush**();

}

**void** **putPixel**(**int** x,**int** y,**int** xc,**int** yc)

{

**glVertex2i**(x+xc, y+yc);

**glVertex2i**(y+xc, x+yc);

**glVertex2i**(x+xc, -y+yc);

**glVertex2i**(-y+xc, x+yc);

**glVertex2i**(-x+xc, -y+yc);

**glVertex2i**(-y+xc, -x+yc);

**glVertex2i**(-x+xc, y+yc);

**glVertex2i**(y+xc, -x+yc);

}

**void** **circle**(xc,yc,r)

{

**int** x,y,d1;

x=0;

y=r;

**glBegin**(GL\_POINTS);

putPixel(x,y,xc,yc);

d1=3-2-r;

**while**(x<=y)

{

**if**(d1<0)

{

x++;

d1=d1+4\*x+6;

putPixel(x,y,xc,yc);

}

**else**

{

x++;

y--;

d1=d1+4\*(x-y)+10;

putPixel(x,y,xc,yc);

}

}

}

**#endif** /\* SHAPES\_H\_ \*/

1 . Write a c program using OpenGL library to draw a clock as given below using a line drawing & Circle drawing algorithms.

Code :

**#include**<stdio.h>

**#include**<GL/glut.h>

**#include** "shapes.h"

**void** **clock**(){

**glColor3f**(1, 0, 0);

circle(0,0,150);

line(0,0,110,0);

line(0,0,0,130);

line(0,140,0,150); // 12 0'clock

line(140,0,150,0); // 03 0'clock

line(-140,0,-150,0); // 06 0'clock

line(0,-140,0,-150); // 09 0'clock

**glEnd**();

**glFlush**();

}

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

{

glutInit(&argc, argv);

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

**glutInitWindowSize**(600, 600);

**glutInitWindowPosition**(60, 60);

glutCreateWindow("Clock");

**glMatrixMode**(GL\_PROJECTION);

**gluOrtho2D**(-200, 200, -200, 200);

**glutDisplayFunc**(clock);

**glClearColor**(1, 1, 1, 1);

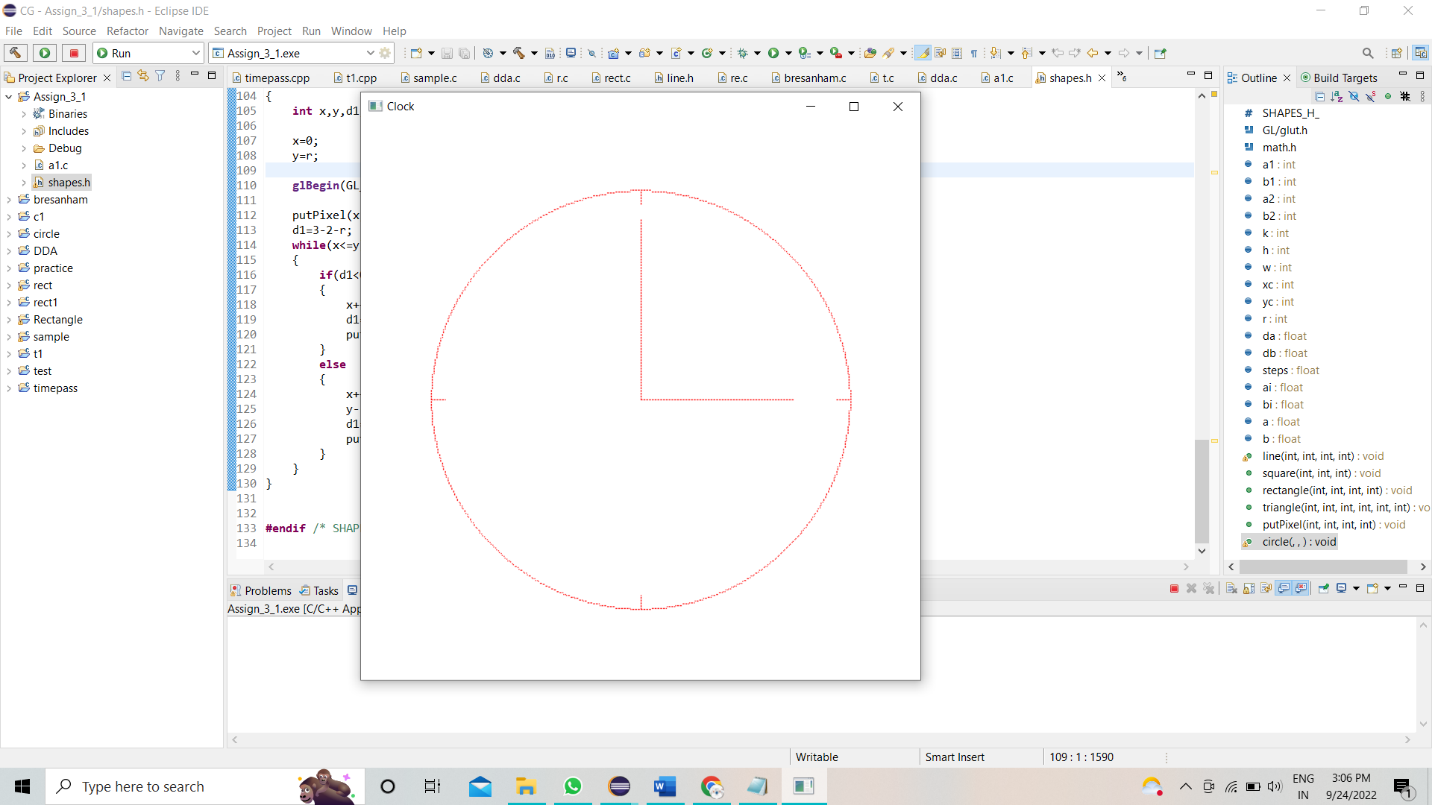
**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glutMainLoop**();

**return** 0;

}

Output :



2. Write a c program using OpenGL library to draw logo of Olympics games as given below using a line drawing & Circle drawing algorithms.

Code :

**#include**<stdio.h>

**#include**<GL/glut.h>

**#include** "shapes.h"

**void** **olympic**(){

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(0,1,1,0.5);

**glPointSize**(7);

**glColor3d**(1, 0, 0);

circle(-120,25,50); // 1

**glColor3d**(0, 0, 0);

circle(0,25,50); // 2

**glColor3d**(0, 0, 1);

circle(120,25,50); // 3

**glColor3d**(1, 1, 0);

circle(-60,-25,50); // 4

**glColor3d**(0, 1, 0);

circle(60,-25,50); // 5

**glEnd**();

**glFlush**();

}

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

{

glutInit(&argc, argv);

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

**glutInitWindowSize**(600, 600);

**glutInitWindowPosition**(60, 60);

glutCreateWindow("Olympics");

**glMatrixMode**(GL\_PROJECTION);

**gluOrtho2D**(-200, 200, -200, 200);

**glutDisplayFunc**(olympic);

**glClearColor**(1, 1, 1, 1);

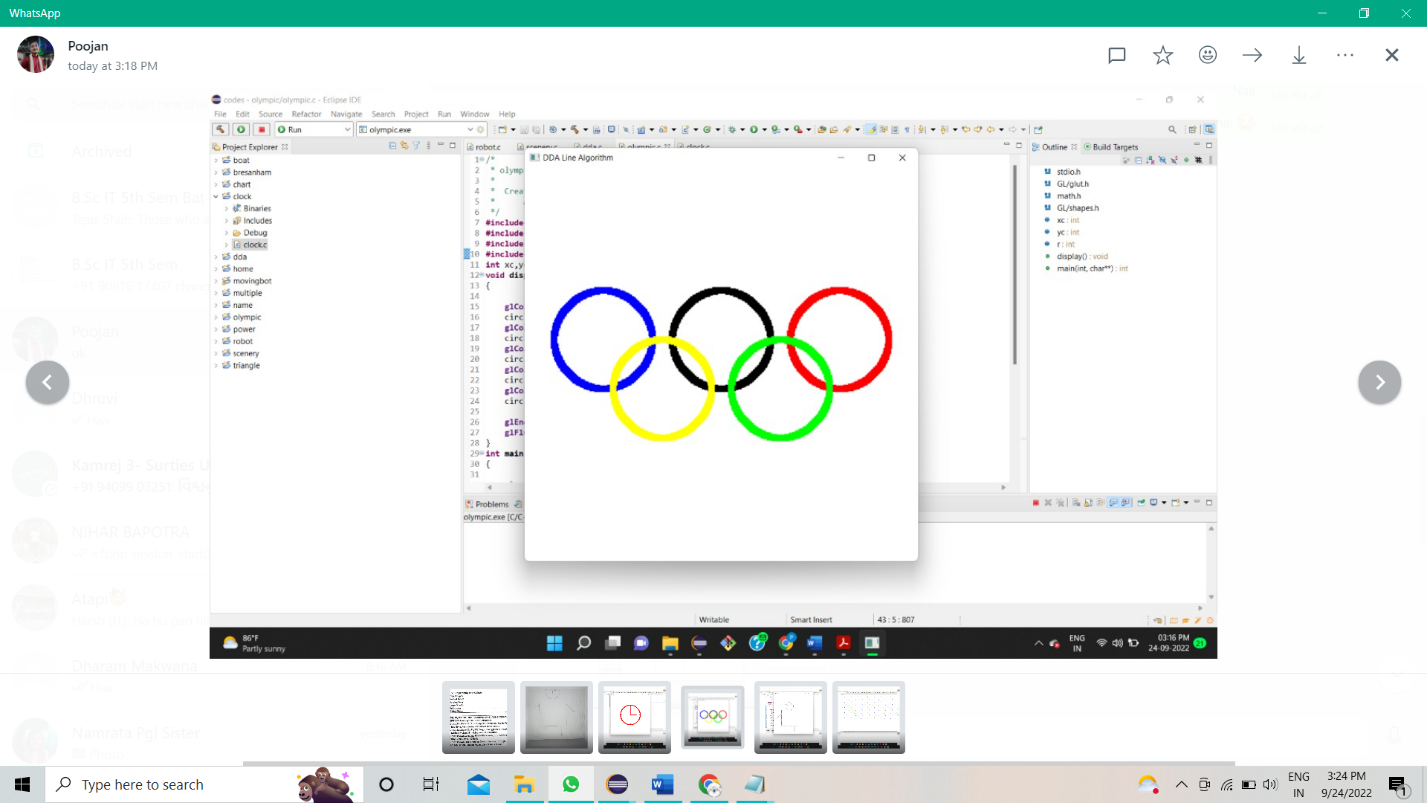
**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glutMainLoop**();

**return** 0;

}

Output :



3. Write a c program using OpenGL library to draw cartoon character games as given below using a line drawing & Circle drawing algorithms.

Code :

**#include**<stdio.h>

**#include**<GL/glut.h>

**#include** "shapes.h"

**void** **robot**(){

**glColor3f**(1, 1, 0);

rectangle(190,150,200,130); // body

circle(255,405,55); // head

line(255,390,255,410); // nose

circle(235,420,10); // eye-1

circle(275,420,10); // eye-2

line(235,370,275,370); // mouth

line(190,300,150,250); // hand-1

line(320,300,370,250); // hand-2

line(210,150,210,70); // leg-1

line(300,150,300,70); // leg-2

line(210,70,230,70); // shoes-1

line(300,70,320,70); // shoes-2

**glEnd**();

**glFlush**();

}

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

{

glutInit(&argc, argv);

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

**glutInitWindowSize**(600, 600);

**glutInitWindowPosition**(500, 50);

glutCreateWindow("ROBOT");

**glMatrixMode**(GL\_PROJECTION);

**gluOrtho2D**(0, 500, 0, 500);

**glutDisplayFunc**(robot);

**glClearColor**(1, 1, 1, 1);

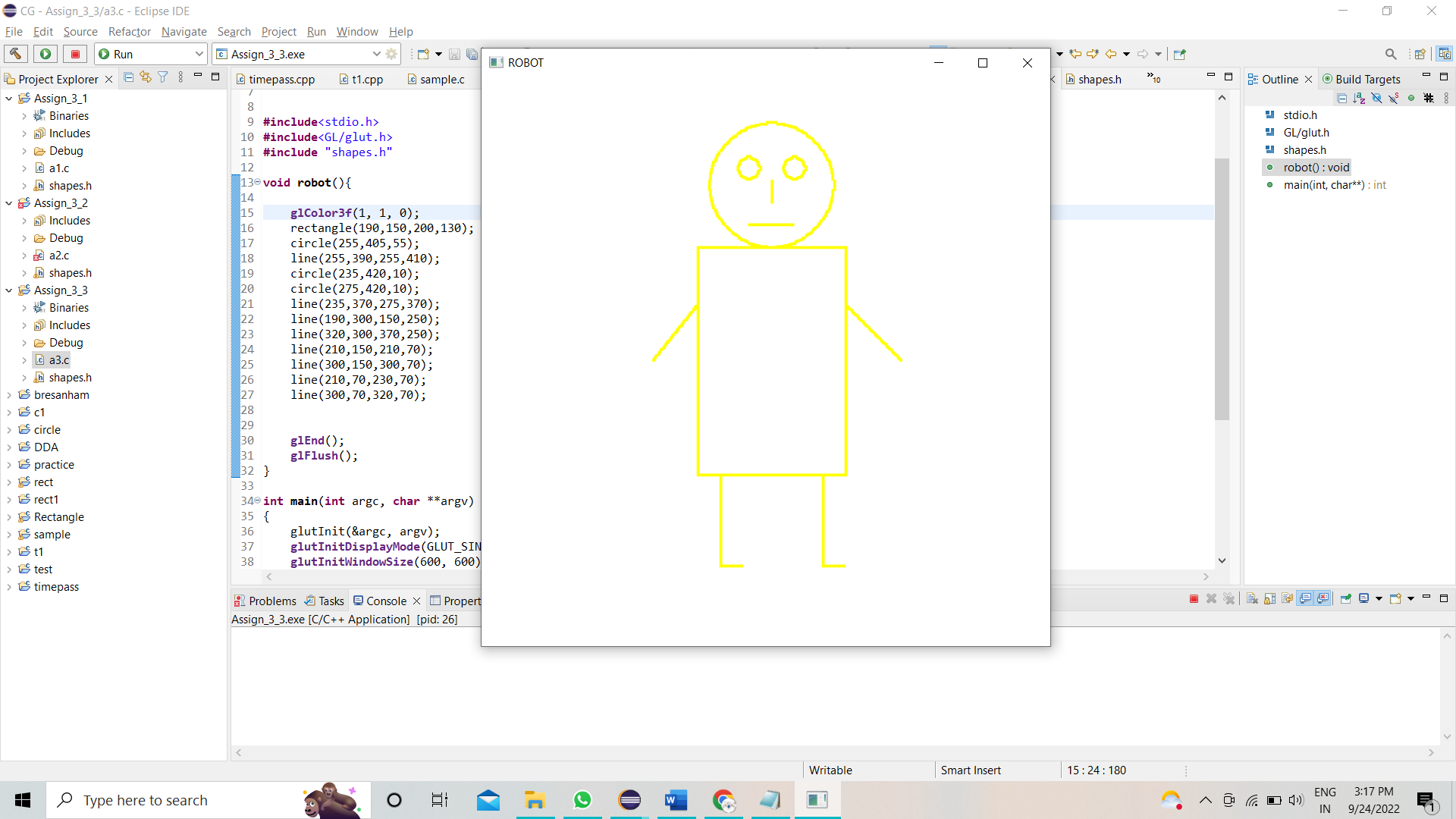
**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glutMainLoop**();

**return** 0;

}

Output :



4. Write a c program using OpenGL library to draw natural scenery with rain as given below using a line drawing & Circle drawing algorithms.

Code :

**#include**<stdio.h>

**#include**<GL/glut.h>

**#include** "shapes.h"

**void** **painting**(){

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(0,0,0,1);

// glPointSize(7);

**glColor3d**(0, 0, 1);

rectangle(0,0,150,1200);// lake

// rain

**for**(**int** i=30;i<1200;i+=50)

{

line(i,180,i,200);

line(i,260,i,280);

line(i,340,i,360);

line(i,420,i,440);

line(i,500,i,520);

}

**for**(**int** i=50;i<1200;i+=50)

{

line(i,220,i,240);

line(i,300,i,320);

line(i,380,i,400);

line(i,460,i,480);

}

**glColor3d**(1, 1, 0);

circle(350,350,55); // Sun

**glColor3d**(0, 1, 0);

line(0,200,200,400); // hill-1

line(200,400,400,200); // hill-1

line(350,250,500,400); // hill-2

line(500,400,700,200); // hill-2

line(650,250,800,400); // hill-3

line(800,400,1000,200); // hill-3

line(950,250,1100,400); // hill-4

line(1100,400,1300,200);// hill-4

**glEnd**();

**glFlush**();

}

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

{

glutInit(&argc, argv);

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

**glutInitWindowSize**(1200, 520);

**glutInitWindowPosition**(100, 50);

glutCreateWindow("Painting");

**glMatrixMode**(GL\_PROJECTION);

**gluOrtho2D**(0, 1200, 0, 520);

**glutDisplayFunc**(painting);

**glClearColor**(1, 1, 1, 1);

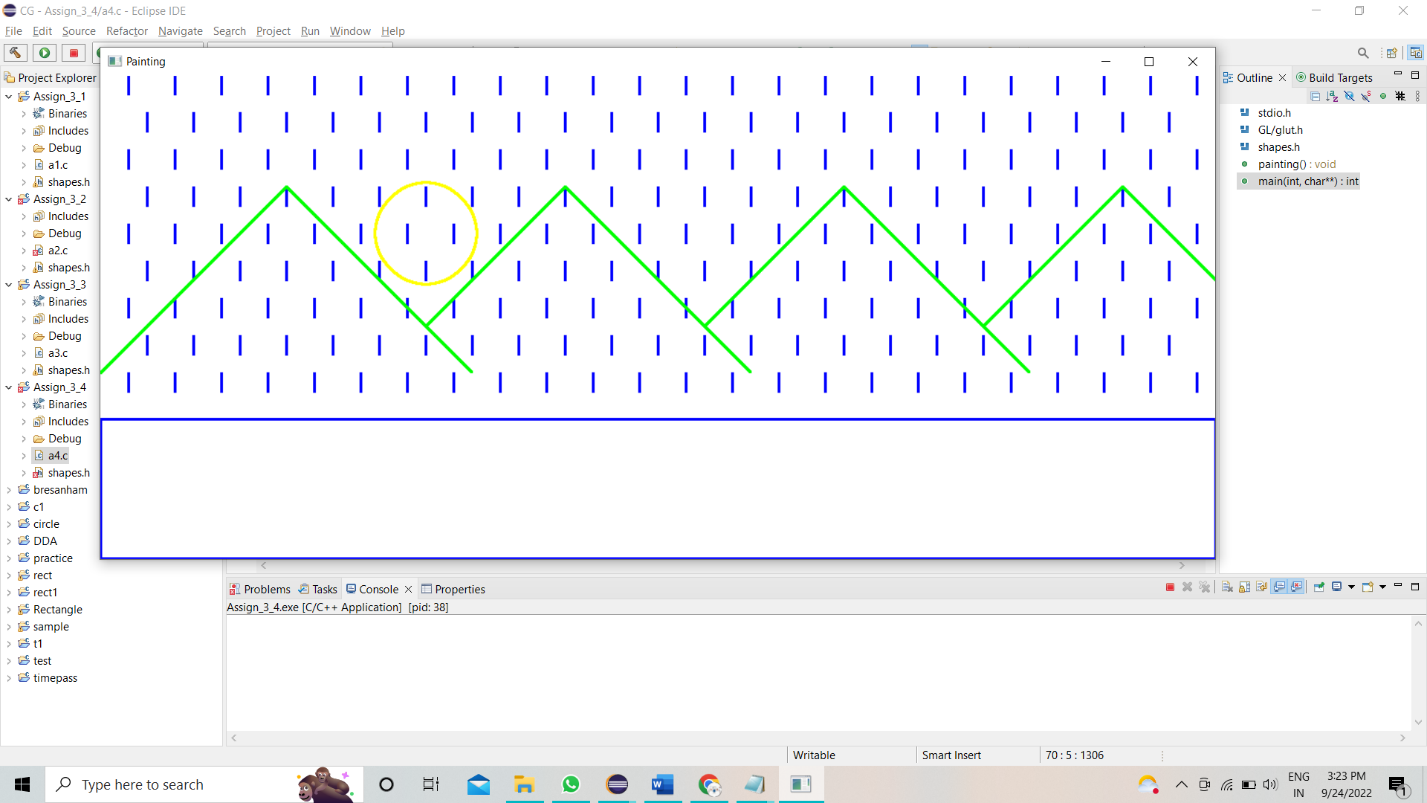
**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glutMainLoop**();

**return** 0;

}

Output :



4.1 with flood fill

**#include**<stdio.h>

**#include**<GL/glut.h>

**#include** "shapes.h"

**int** sx,sy;

**float** intcolor[3] = {1.0,1.0,1.0};

**float** fillcolor[3] = {0.4,0.8,0.1};

**void** **getPixel**(**int** x, **int** y, **float** pixels[3])

{

**glReadPixels**(x,y,1.0,1.0,GL\_RGB,GL\_FLOAT,pixels);//glReadPixels returns pixel data from the frame buffer

}

**void** **setPixel**(**int** pointx, **int** pointy, **float** color[3])

{

**glRasterPos2i**(pointx, pointy);//Specifies the raster position for pixel operations

**glDrawPixels**(1,1,GL\_RGB,GL\_FLOAT,color);

**glFlush**();

}

**void** **floodfill4**(**int** x, **int** y, **float** oldcolor[3], **float** newcolor[3])

{

**float** color[3];

getPixel(x,y,color);

**if**(color[0] == oldcolor[0] && color[1]==oldcolor[1] && color[2]==oldcolor[2])

{

setPixel(x,y,newcolor);

floodfill4(x+1,y,oldcolor,newcolor);//right

floodfill4(x-1,y,oldcolor,newcolor);//left

floodfill4(x,y+1,oldcolor,newcolor);//top

floodfill4(x,y-1,oldcolor,newcolor);//bottom

}

}

**void** **floodfill8**(**int** x, **int** y, **float** oldcolor[3], **float** newcolor[3])

{

**float** interiorColor[3];

getPixel(x,y,interiorColor);

**if**(interiorColor[0] == oldcolor[0] && interiorColor[1]==oldcolor[1] && interiorColor[2]==oldcolor[2])

{

setPixel(x,y,newcolor);

floodfill4(x+1,y,oldcolor,newcolor); //right

floodfill4(x-1,y,oldcolor,newcolor); //left

floodfill4(x,y+1,oldcolor,newcolor); //top

floodfill4(x,y-1,oldcolor,newcolor); //bottom

//diagonal pixels

floodfill4(x-1,y-1,oldcolor,newcolor); //bottom left

floodfill4(x-1,y+1,oldcolor,newcolor); //top left

floodfill4(x+1,y-1,oldcolor,newcolor); //bottom right

floodfill4(x+1,y+1,oldcolor,newcolor); //top right

}

}

**void** **mouse**(**int** button, **int** state, **int** x, **int** y)

{

**if**(button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

**int** xi = x;

**int** yi = (480-y);

floodfill8(xi,yi,intcolor,fillcolor);

}

}

**void** **painting**(){

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glClearColor**(0,0,0,1);

// glPointSize(7);

**glColor3d**(0, 0, 1);

rectangle(0,0,150,1200);// lake

// rain

**for**(**int** i=30;i<1200;i+=50)

{

line(i,180,i,200);

line(i,260,i,280);

line(i,340,i,360);

line(i,420,i,440);

line(i,500,i,520);

}

**for**(**int** i=50;i<1200;i+=50)

{

line(i,220,i,240);

line(i,300,i,320);

line(i,380,i,400);

line(i,460,i,480);

}

**glColor3d**(1, 1, 0);

circle(350,350,55); // Sun

**glColor3d**(0, 1, 0);

line(0,200,200,400); // hill-1

line(200,400,400,200); // hill-1

line(350,250,500,400); // hill-2

line(500,400,700,200); // hill-2

line(650,250,800,400); // hill-3

line(800,400,1000,200); // hill-3

line(950,250,1100,400); // hill-4

line(1100,400,1300,200);// hill-4

floodfill4(sx,sy,intcolor,fillcolor);

**glEnd**();

**glFlush**();

}

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

{

glutInit(&argc, argv);

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

**glutInitWindowSize**(1200, 520);

**glutInitWindowPosition**(100, 50);

glutCreateWindow("Painting");

**glMatrixMode**(GL\_PROJECTION);

**gluOrtho2D**(0, 1200, 0, 520);

**glutDisplayFunc**(painting);

**glClearColor**(1, 1, 1, 1);

**glClear**(GL\_COLOR\_BUFFER\_BIT);

**glutMouseFunc**(mouse);

**glutMainLoop**();

**return** 0;

}

Output :

