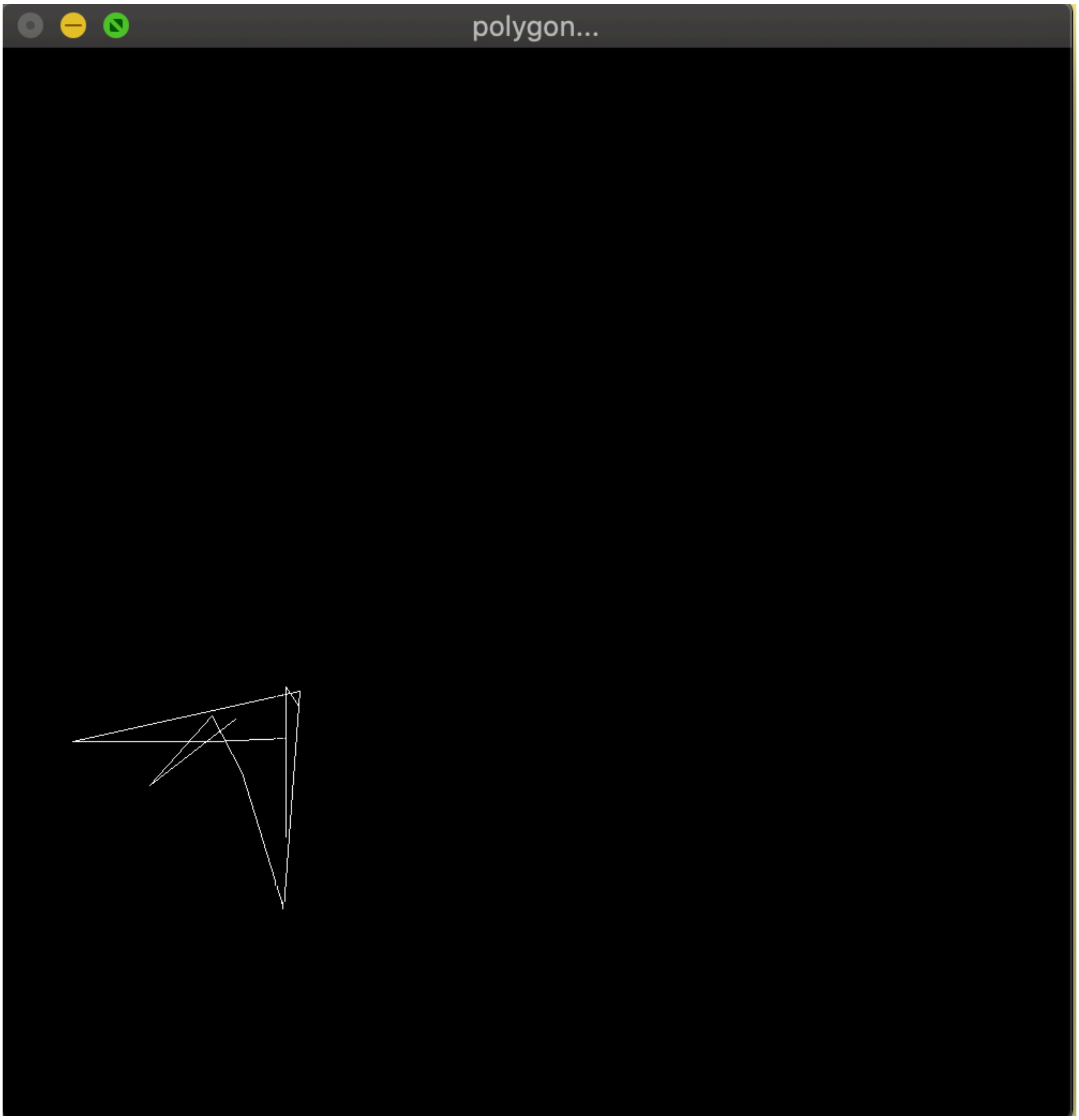
```
/
*==========================================================================
========
Assignment No: 3
Batch       : H10
Roll No     : 23257
Date        :
Aim         : To implement mouse handling in OpenGL .

Problem Statement:

Draw the polygons by using the mouse. Choose colors by clicking on the designed
colorpane.Use window port to draw.(Use DDA algorithm for line drawing)
===========================================================================
=========*/

#include<GLUT/glut.h>
#include<stdio.h>
#include<math.h>
int n;

int Abs(int x) {

        if(x<0)
                return (x*(-1));
        else
                return x;
}

void init()
{
  glClearColor(0.0,0.0,0.0,0.0);
  glMatrixMode(GL_PROJECTION);
  //glLoadIdentity();
  gluOrtho2D(0,500,0,500);
}

int round1(double number)
{
  return (number>=0) ? (int)(number+0.5):(int)(number-0.5);
}

void setPixel(GLint x,GLint y)
{
  //glColor3f(0,1,1);
  glBegin(GL_POINTS);
  glVertex2f(x,y);
  glEnd();
}

void LineWithDDA(int x0,int y0,int x1,int y1)
{
  int dy=y1-y0;
  int dx=x1-x0;
  int steps,i;
  float xinc,yinc,x=x0,y=y0;
  if(Abs(dx)>Abs(dy))
  {
    steps=Abs(dx);
  }
```

```
   else
   {
     steps=Abs(dy);
   }
   xinc=(float)dx/(float)steps;
   yinc=(float)dy/(float)steps;
   setPixel(round1(x),round1(y));
   for(i=0;i<steps;i++)
   {
     x+=xinc;
     y+=yinc;
     setPixel(round1(x),round1(y));
   }
   glutSwapBuffers();
}

int k=1;
int X[10],Y[10];

void mouseClick(int button, int state, int x, int y)
{
   switch(button)
   {
      case GLUT_LEFT_BUTTON: printf(" LEFT ");
      if (state == GLUT_DOWN)
      {
            printf("DOWN\n");
            printf("(%d, %d)\n", x, y);
            X[k]=x;
            Y[k]=500-y; //i did this to match the glut coordinate to window coordinate :)
            //LineWithDDA(0,0,x,y);
            if(k%2==0)

              LineWithDDA(X[k-1],Y[k-1],X[k],Y[k]);

            else if(k>1 && k%2!=0)

              LineWithDDA(X[k-1],Y[k-1],X[k],Y[k]);


            glutSwapBuffers();
            k++;
      }
      else if (state == GLUT_UP)
      {
             printf("UP\n");
      }


     break;
     case GLUT_RIGHT_BUTTON: printf(" RIGHT \n Clearing Screen");
     // Repaint/Clear the screen as same code of function Display()
                          glClearColor(0,0,0,0);
                          glClear(GL_COLOR_BUFFER_BIT);
                          //LineWithDDA(0,0,50,50);
                          glFlush();
    // Reset Value of K to 1
       k=1;
     break;

     default:
     break;
   }
   fflush(stdout);
}

void Display()
{
  glClearColor(0,0,0,0);
  glClear(GL_COLOR_BUFFER_BIT);
  //LineWithDDA(0,0,50,50);
  glFlush();
}

int main(int argc,char** argv)
{
  glutInit(&argc,argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowPosition(50,50);
  glutInitWindowSize(500,500);
  glutCreateWindow("polygon...");
  init();
  glutDisplayFunc(Display);
  glutMouseFunc(mouseClick);
  glutMainLoop();
  return 0;
}

/*

Output:

Someshwars-MacBook-Pro:Assignment3 someshwargaikwad$ ./Assignment3a
 LEFT DOWN
(219, 126)
 LEFT UP
 LEFT DOWN
(139, 188)
 LEFT UP
 LEFT DOWN
(197, 123)
 LEFT UP
 LEFT DOWN
(225, 176)
 LEFT UP
 LEFT DOWN
(264, 303)
 LEFT UP
 LEFT DOWN
(279, 100)
 LEFT UP
 LEFT DOWN
(67, 147)
 LEFT UP
 LEFT DOWN
(197, 147)
 LEFT UP
 LEFT DOWN
```

```
(266, 144)
 LEFT UP
 LEFT DOWN
(266, 236)
 LEFT UP
 LEFT DOWN
(266, 96)
 LEFT UP
 LEFT DOWN
(278, 114)
 LEFT UP
```



```
*/
```