ASSIGNMENT 2

DDA’s Algorithm

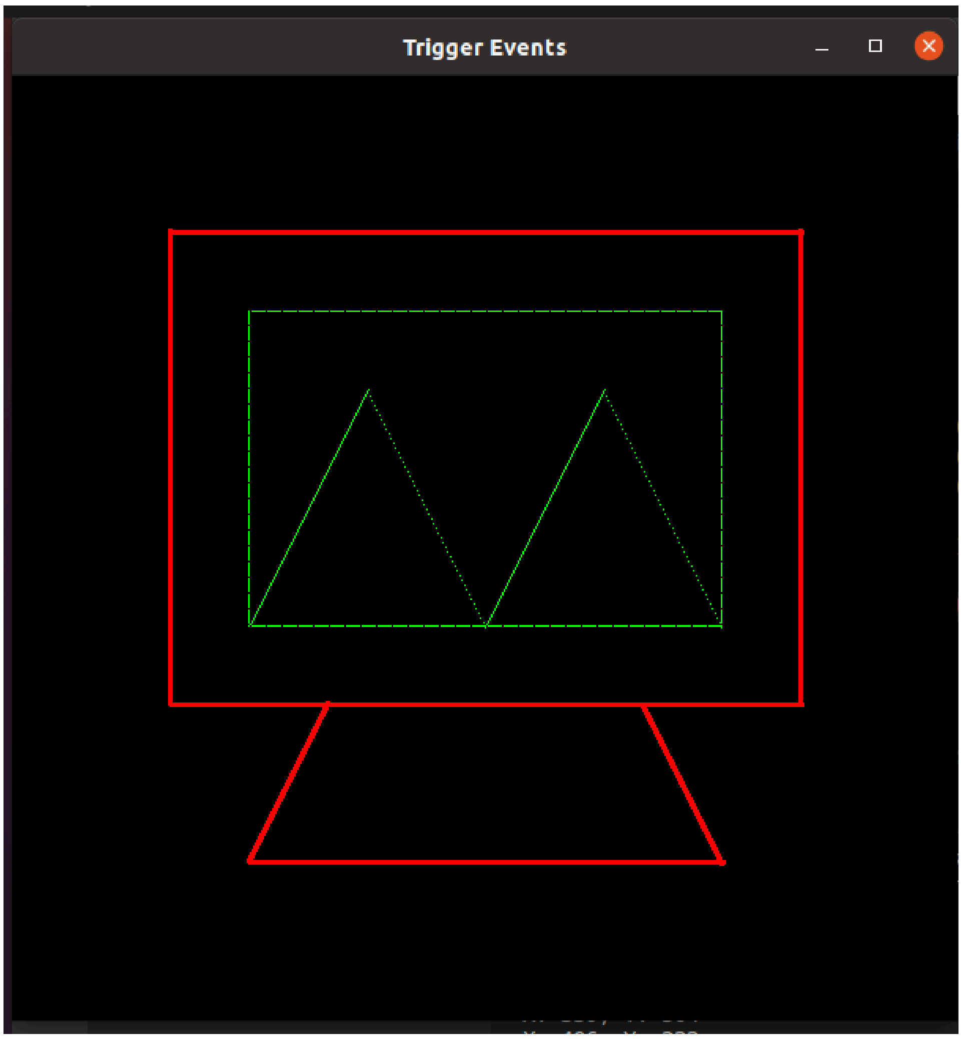
Source Code:

```
#include <iostream>
#include <GL/gl.h>
#include <GL/freeglut.h>
#include <bits/stdc++.h>
using namespace std;
void displayPoint(int x, int y){
   glColor3f(0,1,0);
   glPointSize(1);
   glBegin(GL_POINTS);
   glVertex2i(x,y);
   glEnd();
}
void displayPointBold(int x, int y){
   glColor3f(1,0,0);
   glPointSize(3);
   glBegin(GL_POINTS);
   glVertex2i(x,y);
   glEnd();
}

void simpleLine(float x1,float x2,float y1,float y2){
   float step;
   float dx=x2-x1;
   float dy=y2-y1;
   step= max(abs(dx),abs(dy));
   float xin=dx/float(step);
   float yin=dy/float(step);
   float x=x1;
   float y=y1;
   for(int i=0;i<=step;i++){
       displayPoint(x,y);
       x=x+xin;
       y=y+yin;
   }
   glFlush();

}
```

```
void dottedLine(float x1,float x2,float y1,float y2){
   float step;
   float dx=x2-x1;
   float dy=y2-y1;
   step= max(abs(dx),abs(dy));
   float xin=dx/float(step);
   float yin=dy/float(step);
   float x=x1;
   float y=y1;
   for(int i=0;i<=step;i++){
       x=x+xin;
       y=y+yin;
       if(i%3==0)  displayPoint(x,y);

   }
   glFlush();

}

void dashedLine(float x1,float x2,float y1,float y2){
   float step;
   float dx=x2-x1;
   float dy=y2-y1;
   step= max(abs(dx),abs(dy));
   float xin=dx/float(step);
   float yin=dy/float(step);
   float x=x1;
   float y=y1;
   for(int i=0;i<=step;i++){
       x=x+xin;
       y=y+yin;
       if(i%10!=0)  displayPoint(x,y);
   }
   glFlush();

}

void boldLine(float x1,float x2,float y1,float y2){
   float step;
   float dx=x2-x1;
   float dy=y2-y1;
   step= max(abs(dx),abs(dy));
   float xin=dx/float(step);
   float yin=dy/float(step);
   float x=x1;
   float y=y1;
   for(int i=0;i<=step;i++){
       x=x+xin;
       y=y+yin;
       displayPointBold(x,y);
   }
   glFlush();

}

void primitive(void){
   glClearColor(0.0,0.0,0.0,0.0);
   glClear(GL_COLOR_BUFFER_BIT);
   gluOrtho2D(0,600,0,600);
   glColor3f(1,0,0);

   boldLine(100,500,500,500);
   boldLine(100,500,200,200);
   boldLine(100,100,200,500);
   boldLine(500,500,200,500);

   dashedLine(150,450,450,450);
   dashedLine(150,450,250,250);
   dashedLine(150,150,250,450);
   dashedLine(450,450,250,450);

   simpleLine(150,225,250,400);
   dottedLine(225,300,400,250);
   simpleLine(300,375,250,400);
   dottedLine(375,450,400,250);

   boldLine(150,450,100,100);
   boldLine(150,200,100,200);
   boldLine(400,450,200,100);
}

int main(int argc , char** argv){
   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowPosition(0,0);
```

```
    glutInitWindowSize(600,600);
    glutCreateWindow("Simple Line");
    glutDisplayFunc(primitive);
    glutMainLoop();
    return 0;
}
```

OUTPUT :


Trigger Events