

# **Home Task**

CSE422: Computer Graphics Lab

Submitted To

Syada Tasmia Alvi

Senior Lecturer

Department of CSE

Daffodil International University

Submitted By

Student ID: 221-15-5277

Student Name: Arnob Das Shacha

Section: 61\_D1

Department of CSE

Daffodil International University

## **Title:**

Drawing a Line using DDA Line Drawing Algorithm.

## **Introduction:**

In this project, I have implemented the Digital Differential Analyzer (DDA) algorithm to draw straight lines using OpenGL. DDA is a simple and efficient line drawing technique used in computer graphics, where the intermediate pixel positions are calculated between two end points. The objective of this project was to draw three vertical cricket stumps only using the DDA algorithm, without using any built-in OpenGL line functions. Every stump line is generated pixel-by-pixel using DDA logic.

## **Contents:**

### **Functions Used**

1. **ddaLine()**
  - This is the main line drawing function.
  - It calculates dx, dy, steps, x increment and y increment.
  - Then repeatedly plots each pixel as part of the line.
2. **display()**
  - Clears the screen and draws three stumps by calling ddaLine() three times.
3. **init()**
  - Initializes background color and sets the coordinate system.
4. **main()**
  - Creates the window and handles display settings.

### **Shapes Used**

- ✓ Three vertical straight lines
- ✓ Each line is drawn pixel-by-pixel using DDA

## **Code:**

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

```
#include <math.h>
```

```

void ddaLine(float x1, float y1, float x2, float y2)
{
    float dx = x2 - x1;
    float dy = y2 - y1;

    float steps = fabs(dx) > fabs(dy) ? fabs(dx) : fabs(dy);

    float xInc = dx / steps;
    float yInc = dy / steps;

    float x = x1;
    float y = y1;

    glBegin(GL_POINTS);
    for (int i = 0; i <= steps; i++) {
        glVertex2i(round(x), round(y));
        x += xInc;
        y += yInc;
    }
    glEnd();
}

```

```

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    glPointSize(3);

```

```
// 3 Stumps

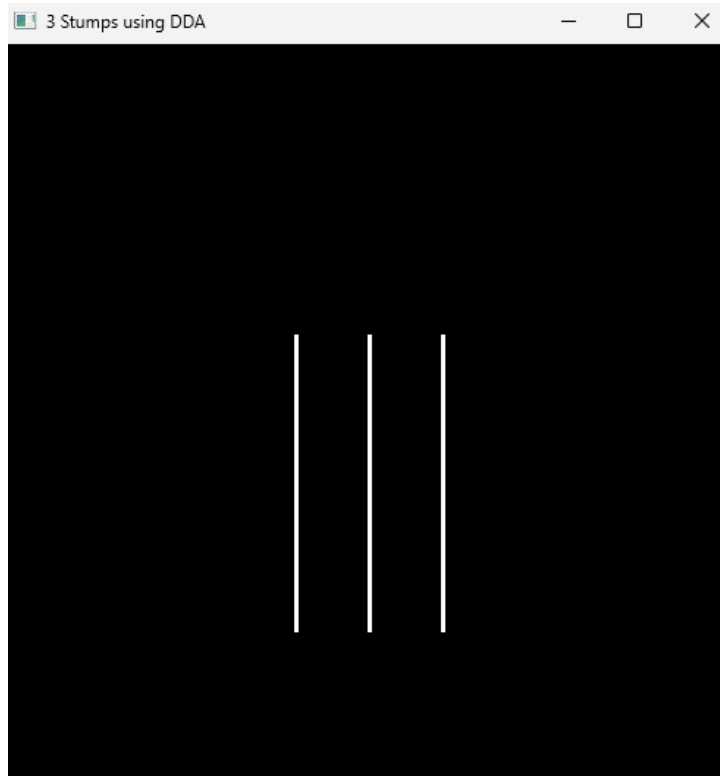
ddaLine(200, 100, 200, 300); // left stump
ddaLine(250, 100, 250, 300); // middle stump
ddaLine(300, 100, 300, 300); // right stump


glFlush();
}


void init()
{
    glClearColor(0, 0, 0, 0);
    gluOrtho2D(0, 500, 0, 500);
}


int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutCreateWindow("3 Stumps using DDA");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
}
```

## Output:



## Discussion:

The goal of the task was to apply the DDA algorithm to draw a simple cricket stump structure. The DDA algorithm works by dividing the line into very small equal steps based on the greater value of  $dx$  or  $dy$ . Then  $x$  and  $y$  values are increased gradually using calculated increments.

In this project, I calculated the increments for each stump and plotted the pixels using `glVertex2i()`.

All three stumps are created using the same DDA logic, simply by changing the starting and ending coordinates. This ensures that the output is fully drawn using the DDA algorithm only, without using any direct OpenGL line drawing commands.

## Conclusion:

This project demonstrates how the DDA algorithm can be applied to draw straight lines accurately in computer graphics. By using DDA to draw three cricket stumps, I have learned how incremental calculations generate continuous pixel-based lines on the screen.

The project successfully shows the practical implementation and visual result of the DDA line drawing method.