

Implementation of Bresenham's Line Algorithm



```

#include <GL/glut.h>
#include <windows.h>
#include <stdio.h>
GLint x0 = 120, y0 = 150, xEnd = 360, yEnd = 420;
void init()
{
    glClearColor(1, 1, 1, 0);
    glColor3ub(255, 0, 0);
    glPointSize(1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 600, 0, 600);
}
void setPixel(GLint xcoordinate, GLint ycoordinate)
{
    glBegin(GL_POINTS);
    glVertex2i(xcoordinate, ycoordinate);
    glEnd();
    glFlush();
}
void lineBA(GLint x0, GLint y0, GLint xEnd, GLint yEnd)
{
    GLint dx = xEnd - x0;
    GLint dy = yEnd - y0;
    GLint steps, k;
    steps = dx;
    GLint x, y, p0 = (2 * dy) - dx;
    setPixel(x0, y0);
    x = x0;
    y = y0;
    for (k = 0; k < steps; k++)
    {
        if (p0 < 0)
        {
            p0 = p0 + (2 * dy);
            x += 1;
        }
        else
        {
            p0 = p0 + (2 * dy) - (2 * dx);
            x += 1;
            y += 1;
        }
        setPixel(x, y);
    }
}

```

```
void Display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    lineBA(x0, y0, xEnd, yEnd);
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("201-15-13706");
    glutDisplayFunc(Display);
    init();
    glutMainLoop();
    return 0;
}
```