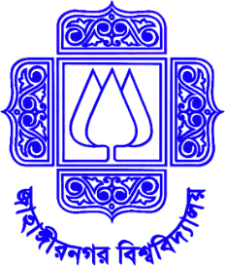
**Lab Report 03**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester 2022*

**Date of Submission**: 11/06/2023



###### **Submitted to-**

###### Dr. Mohammad Shorif Uddin

###### Professor

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

*And*

*Dr. Morium Akter*

###### Associate Professor

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 349 | 202161 | Subarna Saha |

**Experiment No.: 1**

**Experiment Name:** Scan conversion of a line object from (0,0) to (100,50).

1. Rotate it by 30 degree.

2. Scale it to 50%

3. Translate it on x-axis by 75 pixels

**Source code in C:**

#include <stdio.h>

#include<math.h>

#include <graphics.h>

void drawLineBresenham(int x1, int y1, int x2, int y2)

{

int dx = abs(x2 - x1);

int dy = abs(y2 - y1);

int x, y;

int p;

int xStep, yStep;

if(x1<x2)

{

xStep=1;

}

else

{

xStep=-1;

}

if(y1<y2)

{

yStep=1;

}

else

{

yStep=-1;

}

if(dx >= dy)

{

p = 2\*dy-dx;

y = y1;

for (x=x1; x!=x2; x+=xStep)

{

putpixel(x, y, WHITE);

if (p >= 0)

{

y += yStep;

p -= 2 \* dx;

}

p+=2\*dy;

}

}

else

{

p = 2\*dx-dy;

x = x1;

for (y=y1; y!=y2; y+=yStep)

{

putpixel(x, y, WHITE);

if (p >= 0)

{

x+=xStep;

p-=2 \* dy;

}

p+=2\*dx;

}

}

}

void rotateLine(int x1, int y1, int x2, int y2, double angle)

{

double pi=acos(-1);

double radians = angle\*(pi / 180.0);

int newX, newY;

newX = x2\*cos(radians) - y2\*sin(radians);

newY = x2\*sin(radians) + y2\*cos(radians);

drawLineBresenham(x1, y1, newX, newY);

return;

}

void scaleLine(int x1, int y1, int x2, int y2, double percentage)

{

double per=percentage/100.0;

int newX, newY;

newX = (x2 - x1) \* per + x1;

newY = (y2 - y1) \* per + y1;

drawLineBresenham(x1, y1, newX, newY);

return;

}

void translateLine(int x1, int y1, int x2, int y2, int translate\_amount)

{

int newX1, newX2;

newX1 = x1+translate\_amount;

newX2 = x2+translate\_amount;

drawLineBresenham(newX1, y1, newX2, y2);

return;

}

int main()

{

int x1, y1, x2, y2;

x1=0;y1=0;

x2=100; y2=50;

int gd = DETECT, gm;

initgraph(&gd, &gm, (char\*)"");

//Initial line

drawLineBresenham(x1, y1, x2, y2);

//rotate it by 30 degree

double angle=30;

rotateLine(x1, y1, x2, y2, angle);

//Scale it to 50%

double percentage=50;

scaleLine(x1, y1, x2, y2, percentage);

//Translate it on x-axis by 75 pixels

int translate\_amount=75;

translateLine(x1, y1, x2, y2, translate\_amount);

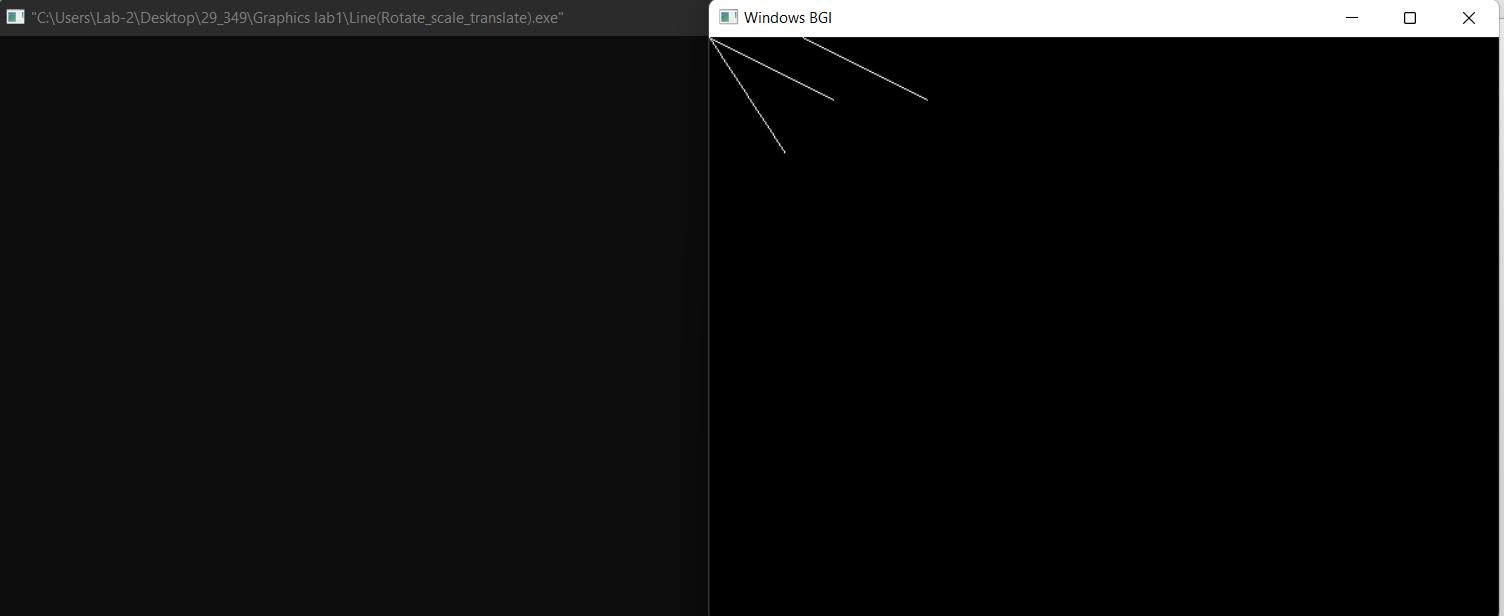
delay(5000);

closegraph();

return 0;

}

**Input and Output:**

****

**Experiment No.: 2**

**Experiment Name:** Scan conversion of a kite using Bresenham’s algorithm.

**Source code in C:**

#include <stdio.h>

#include<graphics.h>

#include <math.h>

void drawLineBresenham(int x1, int y1, int x2, int y2)

{

int dx = abs(x2 - x1);

int dy = abs(y2 - y1);

int x, y;

int p;

int xStep, yStep;

if(x1<x2)

{

xStep=1;

}

else

{

xStep=-1;

}

if(y1<y2)

{

yStep=1;

}

else

{

yStep=-1;

}

if(dx >= dy)

{

p = 2\*dy-dx;

y = y1;

for (x=x1; x!=x2; x+=xStep)

{

putpixel(x, y, WHITE);

if (p >= 0)

{

y += yStep;

p -= 2 \* dx;

}

p+=2\*dy;

}

}

else

{

p = 2\*dx-dy;

x = x1;

for (y=y1; y!=y2; y+=yStep)

{

putpixel(x, y, WHITE);

if (p >= 0)

{

x+=xStep;

p-=2 \* dy;

}

p+=2\*dx;

}

}

}

int main()

{

int x[4] = {50, 100, 50, 0};

int y[4] = {0, 40, 200, 40};

int gd = DETECT, gm;

initgraph(&gd, &gm, (char\*)"");

for (int i = 0; i < 4; i++)

{

int x0 = x[i];

int y0 = y[i];

int x1 = x[(i + 1) % 4];

int y1 = y[(i + 1) % 4];

drawLineBresenham(x0, y0, x1, y1);

}

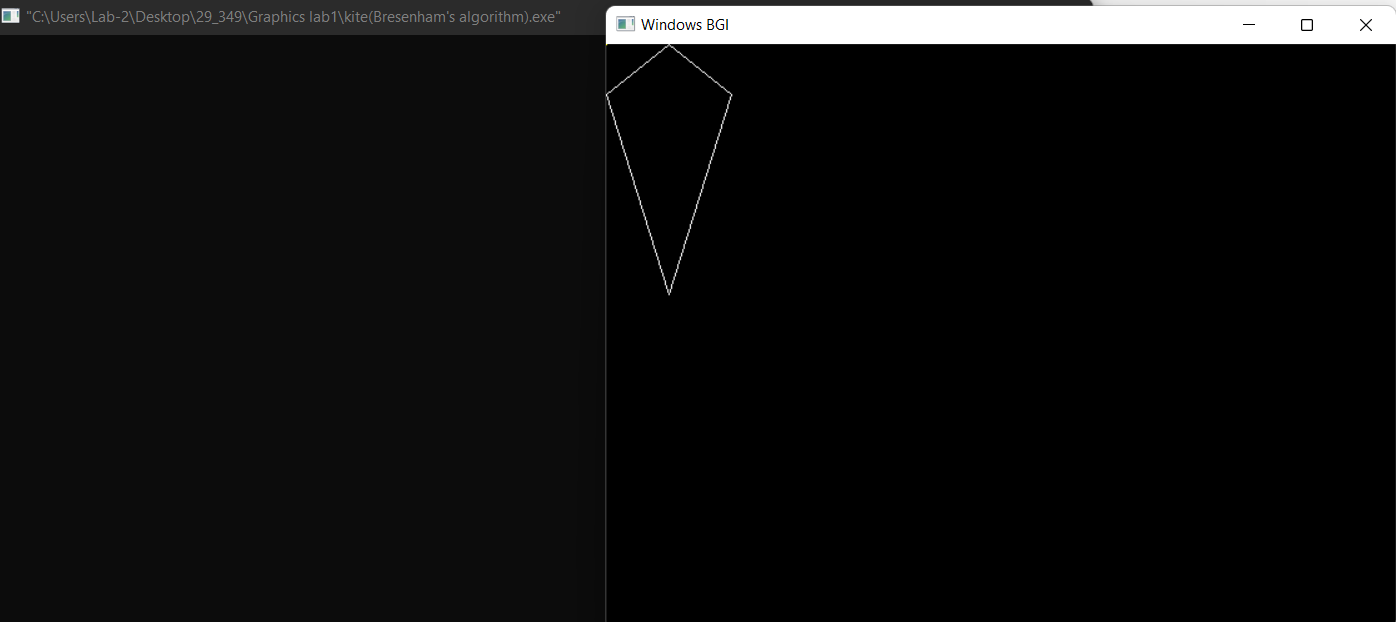
delay(5000);

closegraph();

return 0;

}

**Input and Output:**

****