**Lab Report: 03**

**Title: Scan conversion**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

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

****

**Submitted to-**

*Dr. Mohammad Shorif Uddin*

*Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*&*

*Dr. Morium Akter*

*Associate Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

| Class Roll | Exam Roll | Name |
| --- | --- | --- |
| 356 | 20200650726 | Syeda Faria Sithi |

**Experiment no: 01**

# Experiment Name: Scan conversion of a line object from (0, 0) to (100, 50) and rotted it by 30 degrees

Source Code:

#include <graphics.h>

#include <math.h>

void rotatePoint(int x, int y, int& x\_rot, int& y\_rot, int angle) {

float angle\_rad = angle \* 3.14159 / 180;

x\_rot = x \* cos(angle\_rad) - y \* sin(angle\_rad);

y\_rot = x \* sin(angle\_rad) + y \* cos(angle\_rad);

}

void scanConvertLine(int x1, int y1, int x2, int y2, int angle) {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x1\_rot, y1\_rot, x2\_rot, y2\_rot;

rotatePoint(x1, y1, x1\_rot, y1\_rot, angle);

rotatePoint(x2, y2, x2\_rot, y2\_rot, angle);

float m\_rot = static\_cast<float>(y2\_rot - y1\_rot) / (x2\_rot - x1\_rot);

for (int x = x1\_rot; x <= x2\_rot; ++x) {

int y = static\_cast<int>(m\_rot \* (x - x1\_rot) + y1\_rot);

putpixel(x, y, WHITE);

}

delay(500);

getch();

closegraph();

}

int main() {

int x1 = 0;

int y1 = 0;

int x2 = 100;

int y2 = 50;

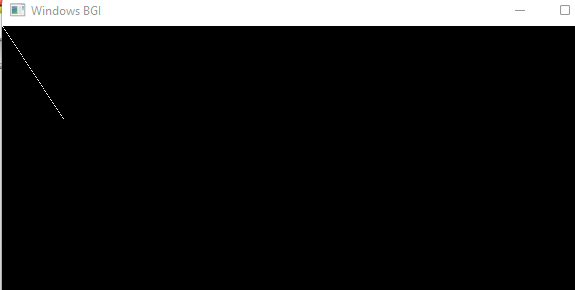
int rotation\_angle = 30;

scanConvertLine(x1, y1, x2, y2, rotation\_angle);

return 0;

}

**Output:**



**Experiment no: 02**

# Experiment Name: Scan conversion of a line object from (0, 0) to (100, 50) and scale it to 50%.

Source Code:

#include <graphics.h>

void scanConvertLine(int x1, int y1, int x2, int y2, float scale) {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x1\_scaled = static\_cast<int>(x1 \* scale);

int y1\_scaled = static\_cast<int>(y1 \* scale);

int x2\_scaled = static\_cast<int>(x2 \* scale);

int y2\_scaled = static\_cast<int>(y2 \* scale);

line(x1\_scaled, y1\_scaled, x2\_scaled, y2\_scaled);

delay(5);

getch();

closegraph();

}

int main() {

int x1 = 0;

int y1 = 0;

int x2 = 100;

int y2 = 50;

float scale = 0.5;

scanConvertLine(x1, y1, x2, y2, scale);

return 0;

}

**Output:**

****

**Experiment no: 03**

# Experiment Name: Scan conversion of a line object from (0, 0) to (100, 50) and translate it on x axis by 75 pixels.

Source Code:

#include <graphics.h>

void scanConvertLine(int x1, int y1, int x2, int y2, int translateX) {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x1\_translated = x1 + translateX;

int x2\_translated = x2 + translateX;

line(x1\_translated, y1, x2\_translated, y2);

delay(5);

getch();

closegraph();

}

int main() {

int x1 = 0;

int y1 = 0;

int x2 = 100;

int y2 = 50;

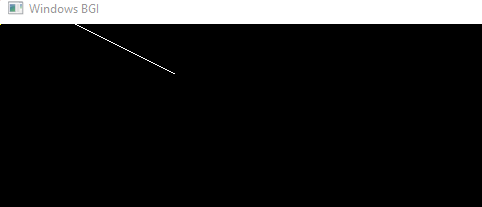
int translateX = 75;

scanConvertLine(x1, y1, x2, y2, translateX);

return 0;

}

**Output:**

****

**Experiment no: 04**

**Experiment Name: Drawing a kite using Bresenham's line Algorithm**

Source Code:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int main()

{

int gd,gm;

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

line(200,200,300,100);

line(300,100,400,200);

line(400,200,300,300);

line(300,100,300,300);

line(300,300,200,200);

arc(300,300,45,135,140);

setfillstyle(SOLID\_FILL,YELLOW);

floodfill(301,105,WHITE);

setfillstyle(SOLID\_FILL,GREEN);

floodfill(299,105,WHITE);

setfillstyle(SOLID\_FILL,LIGHTCYAN);

floodfill(299,275,WHITE);

setfillstyle(SOLID\_FILL,LIGHTBLUE);

floodfill(301,275,WHITE);

line(300,300,250,350);

line(250,350,350,350);

line(300,300,350,350);

setfillstyle(SOLID\_FILL,CYAN);

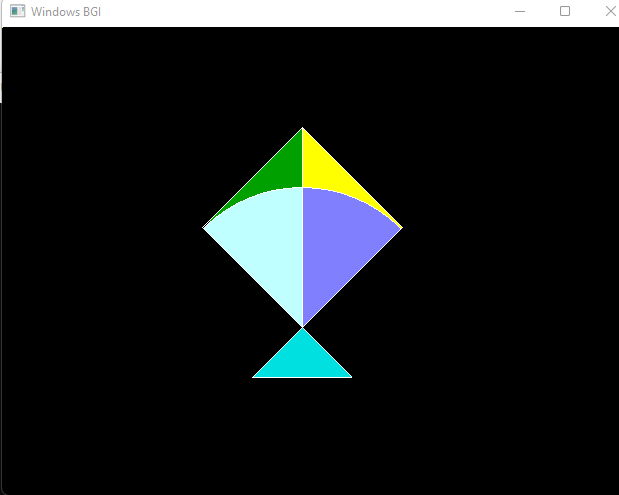
floodfill(300,310,WHITE);

getch();

closegraph();

}

**Output:**

****