Lab III

Course title: Computer Graphics Lab
Course code: CSE-304

3rd Year 1st Semester Examination 2022

Date of Submission: 11-06-2023

Submitted to-

Prof. Dr. Mohammad Shorif Uddin
Professor

Dr. Morium Akther
Associate Professor

Department of Computer Science and Engineering
Jahangirnagar University

Sl	Class Roll	Exam Roll	Name
01	391	202203	Md. Sadman Sakib Sarkar

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

1. Experiment Name: Scan converted a line object from (0,0) to (100,50)

Source Code:

```
#include <iostream>
#include <graphics.h>
void drawLine(int x1, int y1, int x2, int y2) {
   int dx = abs(x2 - x1);
   int dy = abs(y2 - y1);
   int sx = (x1 < x2) ? 1 : -1;
   int sy = (y1 < y2) ? 1 : -1;
   int err = dx - dy;
   while (true) {
      putpixel(x1, y1, WHITE);

      if (x1 == x2 && y1 == y2) {
         break;
      }
      int e2 = 2 * err;
      if (e2 > -dy) {
         err -= dy;
         x1 += sx;
      }
      if (e2 < dx) {
         err += dx;
```

```
            y1 += sy;
        }
    }
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    drawLine(0, 0, 100, 50);
    getch();
    closegraph();
    return 0;
}
```

Output:

1(i). Experiment Name: Rotate it by 30 degree

Source Code:

```
#include <iostream>

#include <graphics.h>

#include <cmath>

void plotPixel(int x, int y) {

   putpixel(x, y, WHITE);

}

void drawLine(int x0, int y0, int x1, int y1) {

   int dx = abs(x1 - x0);

   int dy = abs(y1 - y0);

   int sx = (x0 < x1) ? 1 : -1;

   int sy = (y0 < y1) ? 1 : -1;

   int err = dx - dy;

   while (true) {

      plotPixel(x0, y0);

      if (x0 == x1 && y0 == y1) {

         break;

      }

      int e2 = 2 * err;

      if (e2 > -dy) {

         err -= dy;

         x0 += sx;

      }
```

```
        if (e2 < dx) {
            err += dx;
            y0 += sy;
        }
    }
}
void rotateLine(int& x0, int& y0, int& x1, int& y1, double angle) {
    double theta = angle * 3.14159 / 180.0;
    int newX0 = round(x0 * cos(theta) - y0 * sin(theta));
    int newY0 = round(x0 * sin(theta) + y0 * cos(theta));
    int newX1 = round(x1 * cos(theta) - y1 * sin(theta));
    int newY1 = round(x1 * sin(theta) + y1 * cos(theta));
    x0 = newX0;
    y0 = newY0;
    x1 = newX1;
    y1 = newY1;
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int x0 = 0, y0 = 0;
    int x1 = 100, y1 = 50;
    drawLine(x0, y0, x1, y1);
```

```
    rotateLine(x0, y0, x1, y1, 30.0);
    drawLine(x0, y0, x1, y1);
    getch();
    closegraph();
    return 0;
}
```

Output:

1(ii). Experiment Name: Scale it to 50%

Source Code:

```
#include <iostream>
#include <graphics.h>
#include <cmath>
void plotPixel(int x, int y) {
    putpixel(x, y, WHITE);
}
```

```
void drawLine(int x0, int y0, int x1, int y1) {
   int dx = abs(x1 - x0);
   int dy = abs(y1 - y0);
   int sx = (x0 < x1) ? 1 : -1;
   int sy = (y0 < y1) ? 1 : -1;
   int err = dx - dy;
   while (true) {
      plotPixel(x0, y0);
      if (x0 == x1 && y0 == y1) {
         break;
      }
      int e2 = 2 * err;
      if (e2 > -dy) {
         err -= dy;
         x0 += sx;
      }
      if (e2 < dx) {
         err += dx;
         y0 += sy;
      }
   }
}
```

```
void scaleLine(int& x0, int& y0, int& x1, int& y1, double scaleFactor) {
    int newX0 = round(x0 * scaleFactor);
    int newY0 = round(y0 * scaleFactor);
    int newX1 = round(x1 * scaleFactor);
    int newY1 = round(y1 * scaleFactor);
    x0 = newX0;
    y0 = newY0;
    x1 = newX1;
    y1 = newY1;
}
void rotateLine(int& x0, int& y0, int& x1, int& y1, double angle) {
    double theta = angle * 3.14159 / 180.0;
    int newX0 = round(x0 * cos(theta) - y0 * sin(theta));
    int newY0 = round(x0 * sin(theta) + y0 * cos(theta));
    int newX1 = round(x1 * cos(theta) - y1 * sin(theta));
    int newY1 = round(x1 * sin(theta) + y1 * cos(theta));
    x0 = newX0;
    y0 = newY0;
    x1 = newX1;
    y1 = newY1;
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
```

```
    int x0 = 0, y0 = 0;
    int x1 = 100, y1 = 50;
    drawLine(x0, y0, x1, y1);
    scaleLine(x0, y0, x1, y1, 0.5);
    rotateLine(x0, y0, x1, y1, 30.0);
    drawLine(x0, y0, x1, y1);
    getch();
    closegraph();
    return 0;
}
```

Output:

1(iii). Experiment Name: Translate it on X axis by 75 pixels

Source Code:

```
#include <iostream>

#include <graphics.h>

#include <cmath>

void plotPixel(int x, int y) {

   putpixel(x, y, WHITE);

}

void drawLine(int x0, int y0, int x1, int y1) {

   int dx = abs(x1 - x0);

   int dy = abs(y1 - y0);

   int sx = (x0 < x1) ? 1 : -1;

   int sy = (y0 < y1) ? 1 : -1;

   int err = dx - dy;

   while (true) {

      plotPixel(x0, y0);

      if (x0 == x1 && y0 == y1) {

         break;

      }

      int e2 = 2 * err;

      if (e2 > -dy) {

         err -= dy;

         x0 += sx;

      }
```

```
        if (e2 < dx) {
            err += dx;
            y0 += sy;
        }
    }
}
void translateLine(int& x0, int& y0, int& x1, int& y1, int dx) {
    int newX0 = x0 + dx;
    int newY0 = y0;
    int newX1 = x1 + dx;
    int newY1 = y1;
    x0 = newX0;
    y0 = newY0;
    x1 = newX1;
    y1 = newY1;
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int x0 = 0, y0 = 0;
    int x1 = 100, y1 = 50;
    drawLine(x0, y0, x1, y1);
    translateLine(x0, y0, x1, y1, 75);
```

```
    drawLine(x0, y0, x1, y1);

    getch();

    closegraph();

    return 0;

}
```

Output:

2. Experiment Name: Drawing a kite using Bresenham's line algorithm

Source Code:

```
#include <iostream>

#include <graphics.h>

void drawLine(int x1, int y1, int x2, int y2) {

    int dx = abs(x2 - x1);

    int dy = abs(y2 - y1);
```

```
    int sx = (x1 < x2) ? 1 : -1;
    int sy = (y1 < y2) ? 1 : -1;
    int err = dx - dy;
    while (true) {
        putpixel(x1, y1, WHITE);
        if (x1 == x2 && y1 == y2) {
            break;
        }
        int e2 = 2 * err;
        if (e2 > -dy) {
            err -= dy;
            x1 += sx;
        }
        if (e2 < dx) {
            err += dx;
            y1 += sy;
        }
    }
}
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int centerX = getmaxx() / 2;
    int centerY = getmaxy() / 2;
```

```
    int width = 200;
    int height = 200;

    int topX = centerX;
    int topY = centerY - height / 2;
    int leftX = centerX - width / 2;
    int leftY = centerY;
    int rightX = centerX + width / 2;
    int rightY = centerY;
    int bottomX = centerX;
    int bottomY = centerY + height / 2;
    drawLine(topX, topY, leftX, leftY);
    drawLine(leftX, leftY, bottomX, bottomY);
    drawLine(bottomX, bottomY, rightX, rightY);
    drawLine(rightX, rightY, topX, topY);
    getch();
    closegraph();
    return 0;
}
```

Output: