Министерство образования Республики Беларусь

Министерство образования и науки Российской Федерации

МУВПО “Белорусско-Российский университет”

Кафедра “ПОИТ”

Отчет по

Лабораторной работе №6

# Алгоритмы заполнения многоугольников методом заливки

Выполнил: ст. гр. АСОИ-181

Остапенко А. К.

Преподаватель

Шилов А. В.

Могилёв 2020

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace KGLAB6

{

public partial class Form1 : Form

{

public Form1()

{

int n = 1;

InitializeComponent();

DrawCDA();

}

private static void PutPixel(Bitmap bmp, Color col, int x, int y, int alpha)

{

bmp.SetPixel(x, y, col);

}

public void DrawCDA()

{

Bitmap bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

Graphics graph = Graphics.FromImage(bmp);

Pen pen = new Pen(Color.Black);

int x1 = 200, y1 = 100, x2 = 100, y2 = 200;

int longs;

double x = x1;

double y = y1;

if (Math.Abs(x2 - x1) <= Math.Abs(y2 - y1))

{

longs = Math.Abs(x2 - x1);

}

else

{

longs = Math.Abs(y2 - y1);

}

double dx = (x2 - x1) / longs;

double dy = (y2 - y1) / longs;

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

{

bmp.SetPixel((int)x, (int)y, Color.Black);

x += dx;

y += dy;

}

x1 = 100;

y1 = 200;

x2 = 180;

y2 = 200;

graph.DrawLine(pen, x1, y1, x2, y2);

x1 = 180;

y1 = 200;

x2 = 180;

y2 = 300;

graph.DrawLine(pen, x1, y1, x2, y2);

x1 = 180;

y1 = 300;

x2 = 220;

y2 = 300;

graph.DrawLine(pen, x1, y1, x2, y2);

x1 = 220;

y1 = 300;

x2 = 220;

y2 = 200;

graph.DrawLine(pen, x1, y1, x2, y2);

x1 = 220;

y1 = 200;

x2 = 300;

y2 = 200;

graph.DrawLine(pen, x1, y1, x2, y2);

x1 = 300;

y1 = 200;

x2 = 200;

y2 = 100;

x = x1;

y = y1;

if (Math.Abs(x2 - x1) <= Math.Abs(y2 - y1))

{

longs = Math.Abs(x2 - x1);

}

else

{

longs = Math.Abs(y2 - y1);

}

dx = (x2 - x1) / longs;

dy = (y2 - y1) / longs;

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

{

bmp.SetPixel((int)x, (int)y, Color.Black);

x += dx;

y += dy;

}

Point pt = new Point(201, 102);

FloodFill(bmp, pt, Color.Black, Color.Blue);

pictureBox1.Image = bmp;

}

private void FloodFill(Bitmap bmp, Point pt, Color targetColor, Color replacementColor)

{

targetColor = bmp.GetPixel(pt.X, pt.Y);

if (targetColor.ToArgb().Equals(replacementColor.ToArgb()))

{

return;

}

Stack<Point> pixels = new Stack<Point>();

pixels.Push(pt);

while (pixels.Count != 0)

{

Point temp = pixels.Pop();

int y1 = temp.Y;

while (y1 >= 0 && bmp.GetPixel(temp.X, y1) == targetColor)

{

y1--;

}

y1++;

bool spanLeft = false;

bool spanRight = false;

while (y1 < bmp.Height && bmp.GetPixel(temp.X, y1) == targetColor)

{

bmp.SetPixel(temp.X, y1, replacementColor);

if (!spanLeft && temp.X > 0 && bmp.GetPixel(temp.X - 1, y1) == targetColor)

{

pixels.Push(new Point(temp.X - 1, y1));

spanLeft = true;

}

else if (spanLeft && temp.X - 1 == 0 && bmp.GetPixel(temp.X - 1, y1) != targetColor)

{

spanLeft = false;

}

if (!spanRight && temp.X < bmp.Width - 1 && bmp.GetPixel(temp.X + 1, y1) == targetColor)

{

pixels.Push(new Point(temp.X + 1, y1));

spanRight = true;

}

else if (spanRight && temp.X < bmp.Width - 1 && bmp.GetPixel(temp.X + 1, y1) != targetColor)

{

spanRight = false;

}

y1++;

}

}

pictureBox1.Refresh();

}

}

}

