**ЛАБОРАТОРНА РОБОТА №3**

З дисципліни «Комп’ютерна графіка»

|  |  |
| --- | --- |
| **Виконав**  Студент групи КВ-21  Комарницький О.Б.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Перевірив**  Ст. викл. кафедри СПіСКС  Клятченко Я.М.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Завдання**

* Реалізувати 3 різних алгоритми заливки;

**Код С#**

Програма представлена класами:

**MyDrawer** – здійснює прорисовку екрану різними алгоритмами

**Mainform** – аналізує зчитані дії користувача та викликає методи із **MyDrawer**

Клас MyDrawer

class MyDrawer

{

private PictureBox pBox;

private Bitmap bitmap;

private int xSize, ySize, step = 1;

public MyDrawer(Bitmap bmp, PictureBox \_pBox)

{

bitmap = bmp;

pBox = \_pBox;

xSize = bitmap.Width;

ySize = bitmap.Height;

}

public Bitmap getBitmap()

{

return bitmap;

}

public void recurseAlgourithm\_x4(Point currentPoint)

{

bitmap.SetPixel(currentPoint.X, currentPoint.Y, Color.Red);

if (currentPoint.X + 1 < xSize && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y).G != 128)

recurseAlgourithm\_x4(new Point(currentPoint.X + 1, currentPoint.Y));

if (currentPoint.X - 1 > 0 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y).G != 128)

recurseAlgourithm\_x4(new Point(currentPoint.X - 1, currentPoint.Y));

if (currentPoint.Y + 1 < ySize && bitmap.GetPixel(currentPoint.X, currentPoint.Y + 1).R != 255 && bitmap.GetPixel(currentPoint.X, currentPoint.Y + 1).G != 128)

recurseAlgourithm\_x4(new Point(currentPoint.X, currentPoint.Y + 1));

if (currentPoint.Y - 1 > 0 && bitmap.GetPixel(currentPoint.X, currentPoint.Y - 1).R != 255 && bitmap.GetPixel(currentPoint.X, currentPoint.Y - 1).G != 128)

recurseAlgourithm\_x4(new Point(currentPoint.X, currentPoint.Y - 1));

}

public void recurseAlgourithm\_x8(Point currentPoint)

{

if (bitmap.GetPixel(currentPoint.X, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X, currentPoint.Y).G != 128)

{

bitmap.SetPixel(currentPoint.X, currentPoint.Y, Color.Red);

if (currentPoint.X + 1 < xSize && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X + 1, currentPoint.Y));

if (currentPoint.X - 1 >= 0 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X - 1, currentPoint.Y));

if (currentPoint.Y + 1 < ySize && bitmap.GetPixel(currentPoint.X, currentPoint.Y + 1).R != 255 && bitmap.GetPixel(currentPoint.X, currentPoint.Y + 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X, currentPoint.Y + 1));

if (currentPoint.Y - 1 >= 0 && bitmap.GetPixel(currentPoint.X, currentPoint.Y - 1).R != 255 && bitmap.GetPixel(currentPoint.X, currentPoint.Y - 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X, currentPoint.Y - 1));

if (currentPoint.X + 1 < xSize && currentPoint.Y + 1 < ySize && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y + 1).R != 255 && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y + 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X + 1, currentPoint.Y + 1));

if (currentPoint.X + 1 < xSize && currentPoint.Y - 1 >= 0 && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y - 1).R != 255 && bitmap.GetPixel(currentPoint.X + 1, currentPoint.Y - 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X + 1, currentPoint.Y - 1));

if (currentPoint.X - 1 >= 0 && currentPoint.Y + 1 < ySize && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y + 1).R != 255 && currentPoint.Y + 1 < ySize && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y + 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X - 1, currentPoint.Y + 1));

if (currentPoint.X - 1 >= 0 && currentPoint.Y - 1 >= 0 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y - 1).R != 255 && bitmap.GetPixel(currentPoint.X - 1, currentPoint.Y - 1).G != 128)

recurseAlgourithm\_x8(new Point(currentPoint.X - 1, currentPoint.Y - 1));

}

}

public void Algourithm\_N3(Point currentPoint)

{

int y\_tmp = currentPoint.Y;

bool spanLeft, spanRight;

while (y\_tmp >= 0 && bitmap.GetPixel(currentPoint.X, y\_tmp).R != 255 && bitmap.GetPixel(currentPoint.X, y\_tmp).G != 128)

y\_tmp -= step;

y\_tmp += step;

spanLeft = spanRight = false;

while (y\_tmp < ySize && currentPoint.X < xSize && bitmap.GetPixel(currentPoint.X, y\_tmp).R != 255 && bitmap.GetPixel(currentPoint.X, y\_tmp).G != 128)

{

bitmap.SetPixel(currentPoint.X, y\_tmp, Color.Red);

if (!spanLeft && currentPoint.X > 0 && bitmap.GetPixel(currentPoint.X - step, y\_tmp).R != 255 && bitmap.GetPixel(currentPoint.X - step, y\_tmp).G != 128)

{

Algourithm\_N3(new Point(currentPoint.X - step, y\_tmp));

spanLeft = true;

}

else if (spanLeft && currentPoint.X > 0 && bitmap.GetPixel(currentPoint.X - step, y\_tmp).R != 255 && bitmap.GetPixel(currentPoint.X - step, y\_tmp).G != 128)

{

spanLeft = false;

}

if (!spanRight && currentPoint.X < xSize - step && bitmap.GetPixel(currentPoint.X + step, y\_tmp).R != 255 && bitmap.GetPixel(currentPoint.X + step, y\_tmp).G != 128)

{

Algourithm\_N3(new Point(currentPoint.X + step, y\_tmp));

spanRight = true;

}

else if (spanRight && currentPoint.X < xSize - step && bitmap.GetPixel(currentPoint.X + step, currentPoint.Y).R != 255 && bitmap.GetPixel(currentPoint.X + step, currentPoint.Y).G != 128)

{

spanRight = false;

}

y\_tmp += step;

}

}

Клас Mainform

// Функція, що зчитує події мишки та викликає потрібний алгоритм

public void mouseClick(object sender, MouseEventArgs e)

{

if (MODE == 2) // Ставимо точку для заливки

{

TimeTakenLbl.Text = "Only for simple show";

myDrawer = new MyDrawer(bitmap, pBox);

if (btnAlgourithm1.Checked && ShowDelayBtn.Checked)

myDrawer.recurseAlgourithm\_x4\_Delay(new Point(e.X, e.Y));

else if (btnAlgourithm1.Checked)

{

Thread th = new Thread(() => { myDrawer.recurseAlgourithm\_x4(new Point(e.X, e.Y)); }, 1024 \* 1024 \* 256);

DateTime whenStart = DateTime.Now, whenFinish;

th.Start();

th.Join();

whenFinish = DateTime.Now;

pBox.Image = myDrawer.getBitmap();

TimeTakenLbl.Text = Convert.ToString(whenFinish - whenStart);

} else if (btnAlgourithm2.Checked && ShowDelayBtn.Checked)

myDrawer.recurseAlgourithm\_x8\_Delay(new Point(e.X, e.Y));

else if (btnAlgourithm2.Checked)

{

Thread th = new Thread(() => { myDrawer.recurseAlgourithm\_x8(new Point(e.X, e.Y)); }, 1024 \* 1024 \* 1024);

DateTime whenStart = DateTime.Now, whenFinish;

th.Start();

th.Join();

whenFinish = DateTime.Now;

pBox.Image = myDrawer.getBitmap();

TimeTakenLbl.Text = Convert.ToString(whenFinish - whenStart);

} else if (btnAlgourithm3.Checked && ShowDelayBtn.Checked)

myDrawer.Algourithm\_N3\_Delay(new Point(e.X, e.Y));

else if (btnAlgourithm3.Checked)

{

Thread th = new Thread(() => { myDrawer.Algourithm\_N3(new Point(e.X, e.Y)); }, 1024 \* 1024 \* 1024);

DateTime whenStart = DateTime.Now, whenFinish;

th.Start();

th.Join();

whenFinish = DateTime.Now;

pBox.Image = myDrawer.getBitmap();

TimeTakenLbl.Text = Convert.ToString(whenFinish - whenStart);

}

MODE = 0;

BtnFill.ForeColor = Color.Gainsboro;

previousPoint = new Point();

testLabel.Text = "Click, move your mouse, click again";

return;

}

if(MODE == 0)

{

MODE = 1;

BtnFill.ForeColor = Color.Gainsboro;

testLabel.Text = "Click to finish selecting";

}else

{

MODE = 2;

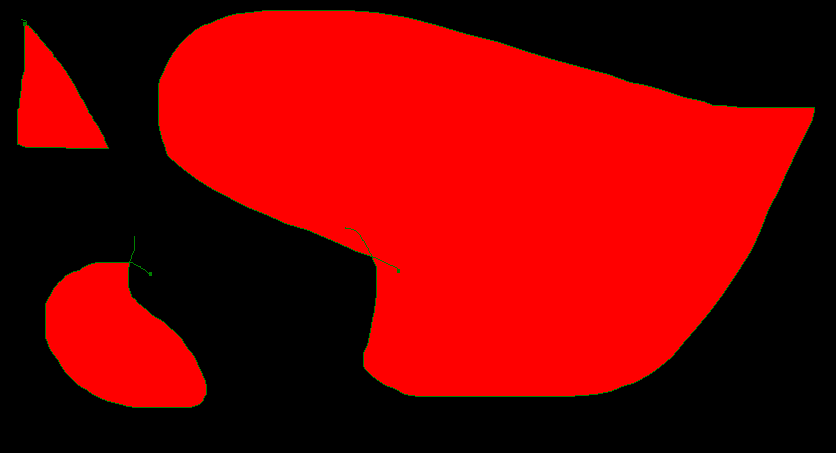
testLabel.Text = "Click to start filling";

BtnFill.ForeColor = Color.BlueViolet;

}

}

Скриншоти заливки

Простий рекурсивний алгоритм 4 точок

Простий рекурсивний алгоритм 8 точок

Простий рекурсивний алгоритм 8 точок

Порядковий алгоритм заливки