**ЛАБОРАТОРНА РОБОТА №4**

З дисципліни «Комп’ютерна графіка»

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

**Завдання**

* Реалізувати побудову 3 різних фракталів:
* Крива Коха;
* Трикутник Серпінського;
* Хмара Мальденброта;

**Код С#**

Алгоритм побудови **хмари Мальденброта**

private void cloudyWorld(int zoomX, int zoomY, int DEPTH)

{

double zi, zr, ci, cr, tmp;

int k, xMax = 530, yMax = 310;

bitmap = new Bitmap(xSize, ySize);

for (int x = -xMax; x < xMax; x++)

{

ci = Convert.ToDouble(x) / zoomX;

for (int y = -yMax; y < yMax; y++)

{

cr = ((float)y) / zoomY;

zi = zr = 0.0; // Очищуємо z

for (k = 0; k < DEPTH; k++) // Обчислюємо множину Мандельброта

{

tmp = zr \* zr - zi \* zi; /\* tmp = zn^2 \*/

zi = 2 \* zr \* zi + ci; /\* zi = zi + ci \*/

zr = tmp + cr; /\* zr = zr + cr \*/

if (zr \* zr + zi \* zi > 1.0E16) // якщо |zn| > R, то

break; // виходимо з циклу

}

if (k > DEPTH)

bitmap.SetPixel(x + xMax, y + yMax, Color.LightGreen);

}

}

pBox.Image = bitmap;

}

Алгоритм побудови **трикутника Серпінського:**

void triangleWorld(int x1, int y1, int x2, int y2, int x3, int y3, int N)

{

int x12, y12, x23, y23, x31, y31;

if(N == 0) return;

x12 = (x1+x2) / 2;

y12 = (y1+y2) / 2;

x23 = (x2+x3) / 2;

y23 = (y2+y3) / 2;

x31 = (x3+x1) / 2;

y31 = (y3+y1) / 2;

Graphics g = Graphics.FromImage(bitmap);

g.DrawLine(pen, x12, y12, x31, y31);

g.DrawLine(pen, x23, y23, x31, y31);

g.DrawLine(pen, x12, y12, x23, y23);

triangleWorld(x1, y1, x12, y12, x31, y31, N - 1);

triangleWorld(x2, y2, x12, y12, x23, y23, N - 1);

triangleWorld(x3, y3, x31, y31, x23, y23, N - 1);

}

Алгоритм побудови **кривої Коха:**

private void curvyWorld(int order, int X1, int Y1, int X2, int Y2)

{

if (order == 0)

{

Graphics g = Graphics.FromImage(bitmap);

g.DrawLine(pen, X1, Y1, X2, Y2);

}

else

{

double alpha = Math.Atan2(Y2 - Y1, X2 - X1);

double R = Math.Sqrt((X2 - X1) \* (X2 - X1) + (Y2 - Y1) \* (Y2 - Y1));

double Xa = X1 + R \* Math.Cos(alpha) / 3,

Ya = Y1 + R \* Math.Sin(alpha) / 3;

double Xc = Xa + R \* Math.Cos(alpha - Math.PI / 3) / 3,

Yc = Ya + R \* Math.Sin(alpha - Math.PI / 3) / 3;

double Xb = X1 + 2 \* R \* Math.Cos(alpha) / 3,

Yb = Y1 + 2 \* R \* Math.Sin(alpha) / 3;

curvyWorld(order - 1, X1, Y1, (int)Xa, (int)Ya);

curvyWorld(order - 1, (int)Xa, (int)Ya, (int)Xc, (int)Yc);

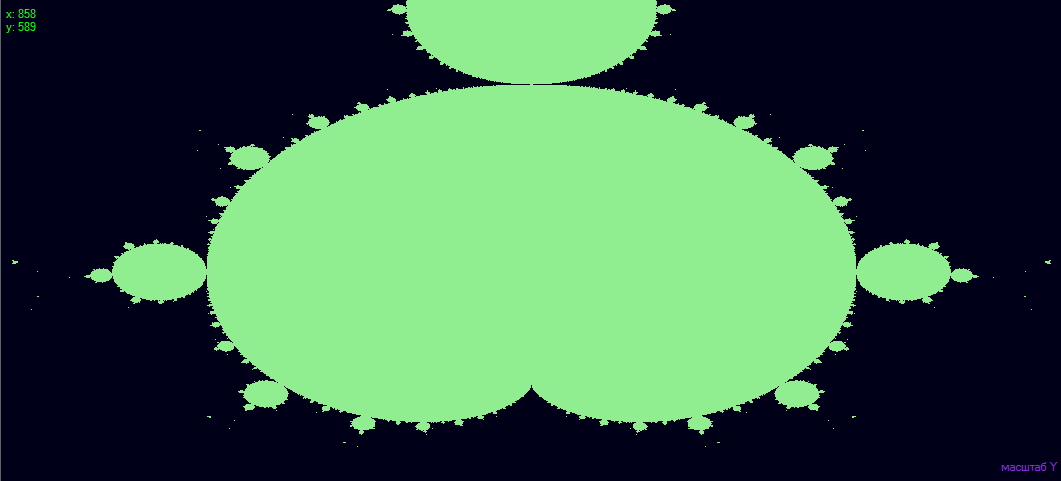
curvyWorld(order - 1, (int)Xc, (int)Yc, (int)Xb, (int)Yb);

curvyWorld(order - 1, (int)Xb, (int)Yb, X2, Y2);

}

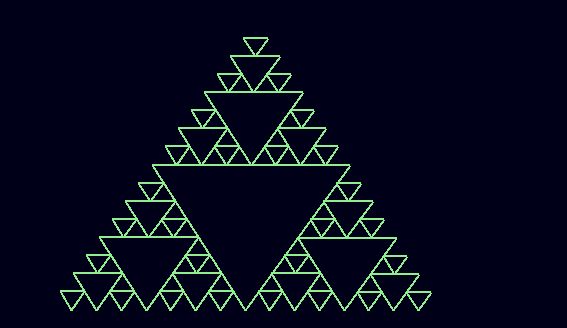
}

**Скриншоти заливки**

**Хмара Мальденброта**

Простий рекурсивний алгоритм 8 точок

**Трикутник Серпінського**



**Крива Коха**

