Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 10**

**по дисциплине**

**«Программирование компьютерной графики»**

**Тема: «Фракталы»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Жигалов И.Е.

Владимир, 2021

ЦЕЛЬ РАБОТЫ

Изучение методов построения изображений с использованием фрактальной геометрии, приобретение навыков использования алгоритмического аппарата фракталов при составлении графических программ.

ЗАДАНИЕ

1. Ознакомиться по методическим указаниям и литературе с теоретическим материалом.

2. Выполнить действия, приведенные в разделе 10.2. При разработке программы имя проекта, создаваемого в MS Visual Studio, должно содержать фамилию студента и группу (например, Ivanov\_Ivan\_ISG\_105\_lab\_1).

3. При выполнении работы изменить программу, применив к фракталу параметры в соответствии с вашим вариантом.

4. Дополнить программу возможностью отрисовки множества Жулиа. Параметры множества использовать в соответствии с вашим вариантом.

ХОД РАБОТЫ

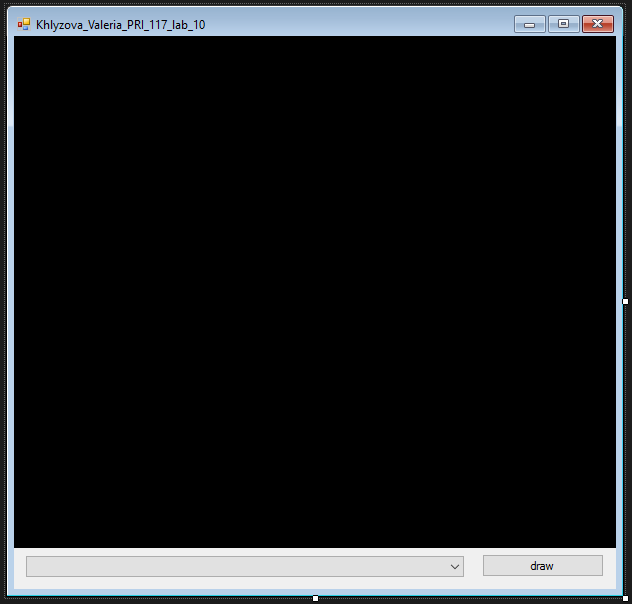


Рисунок 1 - Конструктор формы.

using System;

using System.Threading;

using System.Windows.Forms;

using Tao.FreeGlut;

using Tao.OpenGl;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_10

{

public partial class Form1 : Form

{

static private double[,] mode\_ = new double[5, 2];

ParamsForThread[] threadInputParams = new ParamsForThread[8];

delegate void RenderDLG();

static private byte[,,] PixelsArray = new byte[600, 600, 3];

Thread th\_1 = null;

Thread th\_2 = null;

Thread th\_3 = null;

Thread th\_4 = null;

Thread th\_5 = null;

Thread th\_6 = null;

Thread th\_7 = null;

Thread th\_8 = null;

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void button1\_Click(object sender, EventArgs e)

{

threadInputParams[0] = new ParamsForThread(0, 75, 600);

threadInputParams[1] = new ParamsForThread(75, 150, 600);

threadInputParams[2] = new ParamsForThread(150, 225, 600);

threadInputParams[3] = new ParamsForThread(225, 300, 600);

threadInputParams[4] = new ParamsForThread(300, 375, 600);

threadInputParams[5] = new ParamsForThread(375, 450, 600);

threadInputParams[6] = new ParamsForThread(450, 525, 600);

threadInputParams[7] = new ParamsForThread(525, 600, 600);

threadInputParams[0].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[1].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[2].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[3].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[4].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[5].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[6].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[7].\_pointerToDraw = new ParamsForThread.\_RenderDLG(Draw);

threadInputParams[0].code\_mode = comboBox1.SelectedIndex;

threadInputParams[1].code\_mode = comboBox1.SelectedIndex;

threadInputParams[2].code\_mode = comboBox1.SelectedIndex;

threadInputParams[3].code\_mode = comboBox1.SelectedIndex;

threadInputParams[4].code\_mode = comboBox1.SelectedIndex;

threadInputParams[5].code\_mode = comboBox1.SelectedIndex;

threadInputParams[6].code\_mode = comboBox1.SelectedIndex;

threadInputParams[7].code\_mode = comboBox1.SelectedIndex;

th\_1 = new Thread(CalculateImage);

th\_2 = new Thread(CalculateImage);

th\_3 = new Thread(CalculateImage);

th\_4 = new Thread(CalculateImage);

th\_5 = new Thread(CalculateImage);

th\_6 = new Thread(CalculateImage);

th\_7 = new Thread(CalculateImage);

th\_8 = new Thread(CalculateImage);

th\_1.Priority = ThreadPriority.Lowest;

th\_2.Priority = ThreadPriority.Lowest;

th\_3.Priority = ThreadPriority.Lowest;

th\_4.Priority = ThreadPriority.Lowest;

th\_5.Priority = ThreadPriority.Lowest;

th\_6.Priority = ThreadPriority.Lowest;

th\_7.Priority = ThreadPriority.Lowest;

th\_8.Priority = ThreadPriority.Lowest;

RenderTimer.Start();

th\_1.Start(threadInputParams[0]);

th\_2.Start(threadInputParams[1]);

th\_3.Start(threadInputParams[2]);

th\_4.Start(threadInputParams[3]);

th\_5.Start(threadInputParams[4]);

th\_6.Start(threadInputParams[5]);

th\_7.Start(threadInputParams[6]);

th\_8.Start(threadInputParams[7]);

}

private void Form1\_Load(object sender, EventArgs e)

{

Glut.glutInit();

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

for (int ax = 0; ax < 600; ax++)

{

for (int bx = 0; bx < 600; bx++)

{

PixelsArray[ax, bx, 0] = 0;

PixelsArray[ax, bx, 1] = 0;

PixelsArray[ax, bx, 2] = 0;

}

}

comboBox1.SelectedIndex = 0;

mode\_[1, 0] = 0.34;

mode\_[1, 1] = 0.05;

// значения выбранны в соответствии с вариантом

mode\_[2, 0] = 0.32;

mode\_[2, 1] = 0.04;

// значения подобраны для наглядного представления множество Жюлиа

mode\_[3, 0] = -0.70176;

mode\_[3, 1] = -0.3842;

}

private void Draw()

{

Gl.glRasterPos2i(-1, -1);

Gl.glDrawPixels(600, 600, Gl.GL\_RGB, Gl.GL\_UNSIGNED\_BYTE, PixelsArray);

Gl.glFlush();

AnT.Invalidate();

}

// отклик таймера

private void RenderTimer\_Tick(object sender, EventArgs e)

{

Gl.glRasterPos2i(-1, -1);

Gl.glDrawPixels(600, 600, Gl.GL\_RGB, Gl.GL\_UNSIGNED\_BYTE, PixelsArray);

Gl.glFlush();

AnT.Invalidate();

}

static void CalculateImage(object Settings)

{

ParamsForThread thisThreadSettings = (ParamsForThread)Settings;

// значения выбранны в соответствии с вариантом

double xmin = -1.5;

double ymin = -1.2;

double xmax = 1.5;

double ymax = 1.5;

int W = 600;

int H = 600;

double dx = (xmax - xmin) / (double)(W - 1);

double dy = (ymax - ymin) / (double)(H - 1);

double x, y, X, Y, Cx, Cy;

for (int ax = thisThreadSettings.\_FromImageH; ax < thisThreadSettings.\_ToImageH; ax++)

{

for (int bx = 0; bx < thisThreadSettings.\_ImageW; bx++)

{

x = xmin + ax \* dx;

y = ymin + bx \* dy;

Cx = x;

Cy = y;

X = x;

Y = y;

double ix = 0, iy = 0, n = 0;

while ((ix \* ix + iy \* iy < 4) && (n < 64))

{

switch(thisThreadSettings.code\_mode)

{

case 0:

ix = X \* X - Y \* Y + Cx;

iy = 2 \* X \* Y + Cy;

break;

case 1:

ix = X \* X - Y \* Y + mode\_[thisThreadSettings.code\_mode, 0];

iy = 2 \* X \* Y + mode\_[thisThreadSettings.code\_mode, 1];

break;

case 2:

// изменение параметров в соответствии с вариантом

ix = X \* X - Y \* Y + Math.Pow(mode\_[thisThreadSettings.code\_mode, 0], 2);

iy = 2 \* X \* Y + Math.Pow(mode\_[thisThreadSettings.code\_mode, 1], 2);

break;

case 3:

// множество Жюлиа

ix = X \* X - Y \* Y + mode\_[thisThreadSettings.code\_mode, 0];

iy = 2 \* X \* Y + mode\_[thisThreadSettings.code\_mode, 1];

break;

}

n++;

X = ix;

Y = iy;

}

PixelsArray[bx, ax, 0] = (byte)(255 - n \* 4);

PixelsArray[bx, ax, 1] = (byte)(255 - n \* 4);

PixelsArray[bx, ax, 2] = (byte)(255 - n \* 4);

thisThreadSettings.\_pointerToDraw();

}

}

}

}

}

Класс ParamsForThread будет иметь ряд полей, которые будут хранить в себе необходимые данные для работы потока:

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_10

{

class ParamsForThread

{

public ParamsForThread(int startH, int endH, int Width)

{

\_FromImageH = startH;

\_ToImageH = endH;

\_ImageW = Width;

}

public int code\_mode;

public delegate void \_RenderDLG();

public \_RenderDLG \_pointerToDraw = null;

public int \_FromImageH;

public int \_ToImageH;

public int \_ImageW;

}

}

Результат выполнения:

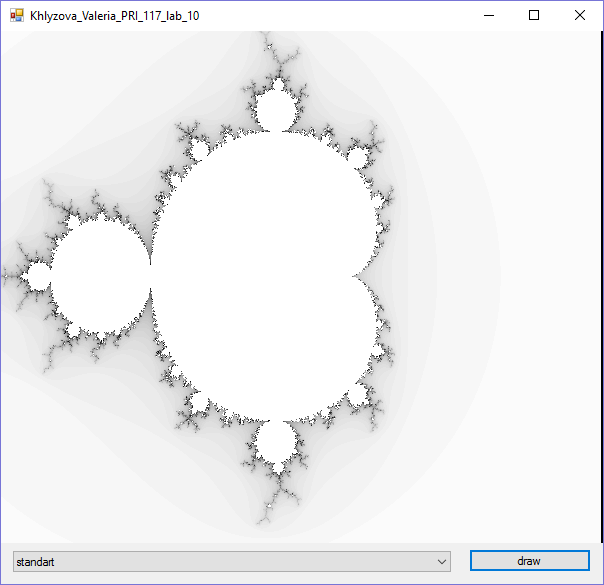


Рисунок 2 - Множество Мандельброта.

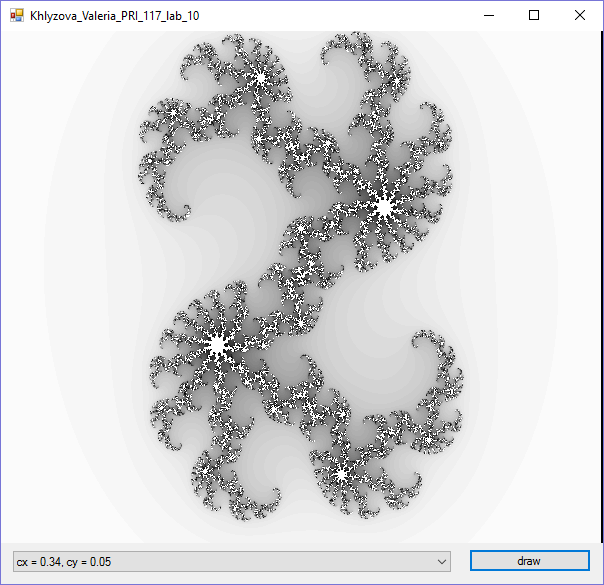


Рисунок 3 - Множество Мандельброта с заданными параметрами cx и cy.

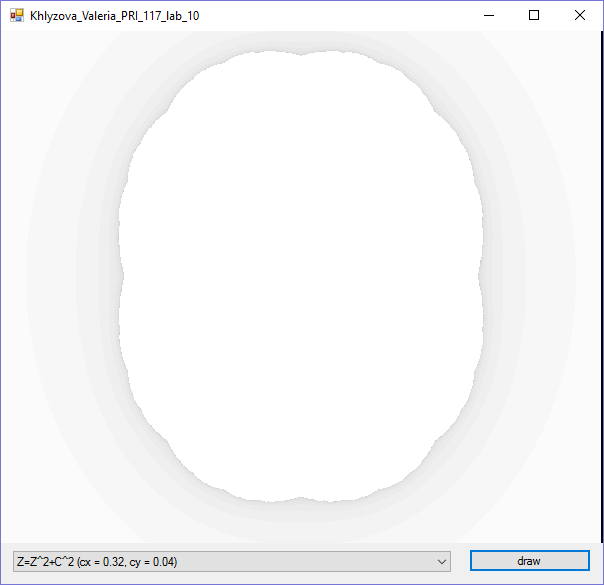


Рисунок 4 - Задание по варианту.

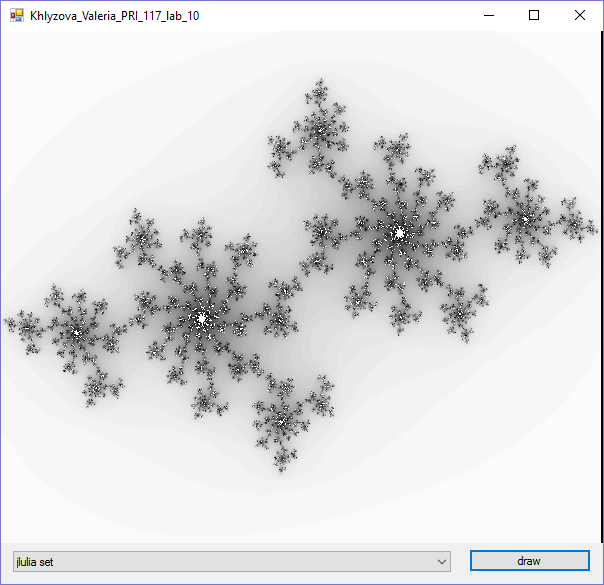


Рисунок 5 - Множество Жюлиа.

ВЫВОД

В ходе выполнения лабораторной работы были изучены методы построения изображений с использованием фрактальной геометрии, приобретены навыки использования алгоритмического аппарата фракталов при составлении графических программ.