Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №7**

**«ИССЛЕДОВАНИЕ ФРАКТАЛОВ»**

**МДК 05.02 «РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Чарушин Егор Вадимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:**

Получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

1. **Формулировка задания:**

Вариант: 23

1. Написать программу для визуализации фрактала Кривая Хартера-Хейтуэя".
2. ﻿﻿﻿Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры
3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.
4. **Описание алгоритма.**

Фрактал «Дракон Хартера-Хейтуэя» (Кривая Дракона) — это фрактал, который был придуман физиком Джоном Хейтуэем, а теорию разработали Вильям Хартер, Хейтуэй и Брюс Бенкс. Каждая итерация добавляет новые сегменты кривой, которые поворачиваются и растут по заранее определенным правилам, создавая сложные узоры, напоминающие контур дракона.

Алгоритм построения фрактала:

1. Процедура Draw - отвечает за рекурсивное построение фрактала.

* Принимает координаты двух концов отрезка (x1, y1) и (x2, y2), а также глубину depth.
* Если глубина depth достигает нуля, то рисуется отрезок между точками (x1, y1) и (x2, y2) с помощью функции Line.
* В противном случае вычисляются координаты середины отрезка и строятся две линии: одна от (x1, y1) до середины отрезка, другая от (x2, y2) до середины отрезка. При этом глубина уменьшается на единицу.

1. Процедура Draw - процедура перерисовывает окно, очищая его и вызывая процедуру Draw с новыми параметрами. Параметры масштабирования и смещения применяются к координатам, чтобы переместить и масштабировать фрактал.
2. Процедура Button - процедура отвечает за реакцию на нажатия клавиш.

* Если нажата стрелка вверх - увеличивается смещение по оси Y
* Если нажата стрелка вниз - уменьшается смещение по оси Y
* Аналогично для нажатий стрелок вправо и влево
* Если нажата клавиша "O" - происходит увеличение масштаба
* Если нажата клавиша "P" - происходит уменьшение масштаба
* Если нажата клавиша "W" - увеличивается глубина фрактала
* Если нажата клавиша "S" - уменьшается глубина фрактала (если она больше 0)

Рекурсивная триада:

1. Параметризация:

Происходит через использование переменных:

* Scale – Масштаб рисунка
* Depth – Глубина рекурсии
* offsetX и offsetY – Сдвиг по осям X и Y

1. Выделение базы:

Условие выхода их рекурсии задается в процедуре «Draw», если глубина рекурсии достигает depth = 0, то выполняется действие - рисование линии между двумя точками (x1, y1) и (x2, y2)

1. Декомпозиция:

Происходит в процедуре Draw. Данная процедура разбивает задачу на более маленькие подзадачи. Вместо того, чтобы рисовать линию между точками (x1, y1) и (x2, y2) на текущей итерации рекурсии, процедура вычисляет среднюю точку и вызывает себя для двух подзадач:

* Рисования линии от x1, y1 до средней точки
* Рисования линии от x2, y2 до средней точки

1. **Схема алгоритма.**

**Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание**

Рисунок 1 – Общая схема алгоритма.

**Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание**

Рисунок 2 – Алгоритм процедуры Draw.

**Изображение выглядит как текст, снимок экрана, диаграмма, круг

Автоматически созданное описание**

Рисунок 3 – Алгоритм процедуры Redraw.

**Изображение выглядит как снимок экрана, монохромный

Автоматически созданное описание**

Рисунок 4 – Алгоритм процедуры Button.

1. **Код программы.**

**program** DKR\_1;

**uses** GraphABC;

**var**

scale: Real = 1.0;

depth: Integer = 6;

offsetX: Real = 0.0;

offsetY: Real = 0.0;

**procedure** Draw(x1, y1, x2, y2, depth: integer); // для рисования

**begin**

**if** depth = 0 **then**

Line(x1, y1, x2, y2)

**else**

**begin**

**var** midX := (x1 + x2) **div** 2 + (y2 - y1) **div** 2;

**var** midY := (y1 + y2) **div** 2 - (x2 - x1) **div** 2;

Draw(x1, y1, midX, midY, depth - 1);

Draw(x2, y2, midX, midY, depth - 1);

**end**;

**end**;

**procedure** Redraw; // для перерисовки окна

**begin**

ClearWindow;

Draw(Round(100\*scale+offsetX), Round(400\*scale+offsetY), Round(700\*scale+offsetX), Round(400\*scale+offsetY), depth);

**end**;

**procedure** Button(Key: Integer); // нажатие кнопок

**begin**

**case** Key **of**

VK\_Down: **begin**

offsetY := offsetY - 30;

Redraw;

**end**;

VK\_Up: **begin**

offsetY := offsetY + 30;

Redraw;

**end**;

VK\_Right: **begin**

offsetX := offsetX - 30;

Redraw;

**end**;

VK\_Left: **begin**

offsetX := offsetX + 30;

Redraw;

**end**;

VK\_O: **begin**

scale := scale \* 1.5;

Redraw;

**end**;

VK\_P: **begin**

scale := scale / 1.5;

Redraw;

**end**;

VK\_W: **begin**

depth := depth + 1;

Redraw;

**end**;

VK\_S: **if** depth > 0 **then begin**

depth := depth - 1;

Redraw;

**end**;

**end**;

**end**;

**begin**

SetWindowSize(600, 600);

SetWindowCaption('Кривая Хартера-Хейтуэя');

OnKeyDown := Button;

Redraw;

**end**.

1. **Результат выполнения программы.**

**Изображение выглядит как снежинка, шаблон

Автоматически созданное описание с низким доверительным уровнем**

Рисунок 5 – Результат выполнения программы.

Изображение выглядит как снимок экрана, линия, График, диаграмма

Автоматически созданное описание

Рисунок 6 – Результат увеличения глубины.

Изображение выглядит как шаблон, вышивка, шов

Автоматически созданное описание

Рисунок 7 – Результат уменьшения глубины.

Изображение выглядит как диаграмма, текст, План, схематичный

Автоматически созданное описание

Рисунок 8 – Результат сдвига влево.

Изображение выглядит как текст, диаграмма, План, карта

Автоматически созданное описание

Рисунок 9 – Результат сдвига вправо

Изображение выглядит как текст, диаграмма, План, Прямоугольник

Автоматически созданное описание

Рисунок 10 - Результат сдвига вверх.

Изображение выглядит как снимок экрана, текст, шов, шаблон

Автоматически созданное описание

Рисунок 11 – Результат уменьшения.

1. **Вывод.**

В результате выполнения задания поставленные задачи были выполнены в полном объеме. Необходимые цели были достигнуты.

Была произведена работа с фракталом «Кривая Хартера-Хейтуэя», данный фрактал был написан и графически показан на языке программирования Pascal с помощью графического модуля GraphABC.

По ранее написанному коду программы были простроены схемы алгоритма, а также подробно описана рекурсивная триада, и каждая процедура для построения графического изображения.

Также была произведена проверка программы на работоспособность, а именно работа всех клавиш, которые управляют глубиной, увеличением, а также движением заданного фрактала.

Таким образом, все поставленные задачи были выполнены, все цели были достигнуты, закреплены и получены знания о работе с фракталами, а так же закреплены знания в работе графикой и построения схемы алгоритма программы.