аМинистерство 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»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: Евенок Мария

ИСПк-203-52-00

Колледж ВятГУ

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель работы: получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

2. Скриншот задания с вариантом:

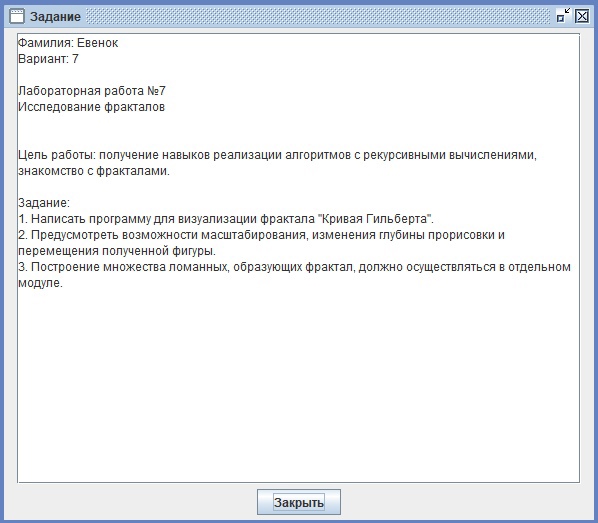


Рисунок 1. Скриншот задания

3. Описание алгоритма:

Использование кривой Гильберта в пространственном поиске сталкивается с рядом трудностей. Например, само вычисление ключа из 3 координат занимает (у автора) около 1000 тактов, примерно в 20 раз больше, чем то же для zcurve. Возможно, существует какой-то более эффективный метод, кроме того, есть мнение, что уменьшение числа прочитанных страниц окупает любое усложнение работы. Т.е. диск — более дорогой ресурс чем процессор. Поэтому мы будем в первую очередь ориентироваться именно на чтения буферов, а времена просто примем к сведению.

4. Код программы:

**Uses** GraphWPF;

**var** (u,p, wx,wy) := (32, 4, 64,64);

**Procedure** a(i: Integer); **forward**;

**Procedure** b(i: Integer); **forward**;

**Procedure** c(i: Integer); **forward**;

**Procedure** d(i: Integer); **forward**;

**Procedure** a(i: Integer) :=

**If** i > 0 **Then begin**

d(i - 1); LineBy(+u, 0);

a(i - 1); LineBy(0, u);

a(i - 1); LineBy(-u, 0);

c(i - 1)

**end**;

**Procedure** b(i: integer) :=

**If** i > 0 **Then begin**

c(i - 1); LineBy(-u, 0);

b(i - 1); LineBy(0, -u);

b(i - 1); LineBy(u, 0);

d(i - 1)

**end**;

**Procedure** c(i: integer) :=

**If** i > 0 **Then begin**

b(i - 1); LineBy(0, -u);

c(i - 1); LineBy(-u, 0);

c(i - 1); LineBy(0, u);

a(i - 1)

**end**;

**Procedure** d(i: integer) :=

**If** i > 0 **Then begin**

a(i - 1); LineBy(0, u);

d(i - 1); LineBy(u, 0);

d(i - 1); LineBy(0, -u);

b(i - 1)

**end**;

**Procedure** KeyDown(k: Key);

**begin**

**case** K **of**

Key.Down: wy -= 10;

Key.Up: wy += 10;

Key.Left: wx += 10;

Key.Right: wx -= 10;

Key.Escape: halt(1);

Key.add: **if** p\*u < 1000 **then** u := p\*u>50 ? (u \* 1.1).trunc : u+5;

Key.subtract: **if** (p\*u > 50) **and** (u > 9) **then** u := (u / 1.1).trunc;

Key.w: **if** (p\*u > 50) **and** (p < 5) **then** (u,p) :=(u **div** 2, p+1);

Key.s: **if** p > 1 **then** (u,p) :=(u\*2, p-1);

**end**;

MoveTo(wx,wy);

Redraw(() ->**begin** Window.Clear; a(p); **end**);

**end**;

**begin**

Window.SetSize(620,620);

MoveTo(wx, wy);

Redraw(() ->a(p));

OnkeyDown += KeyDown;

**end**. KeyDown(1);

OnKeyDown += KeyDown;

**end**.

5. Результат выполнения работы

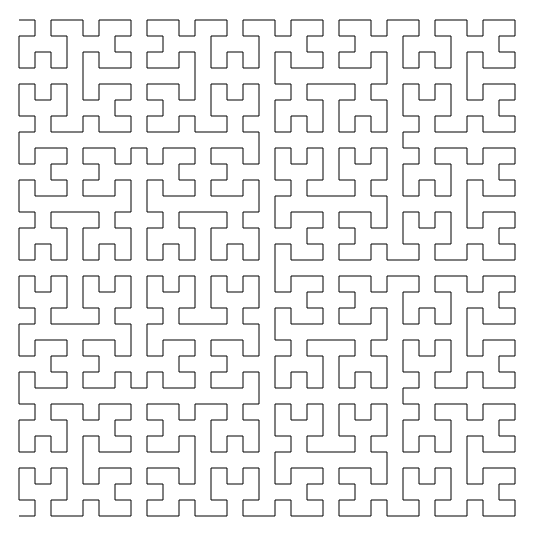


Рисунок 2. Кривая Гильберта

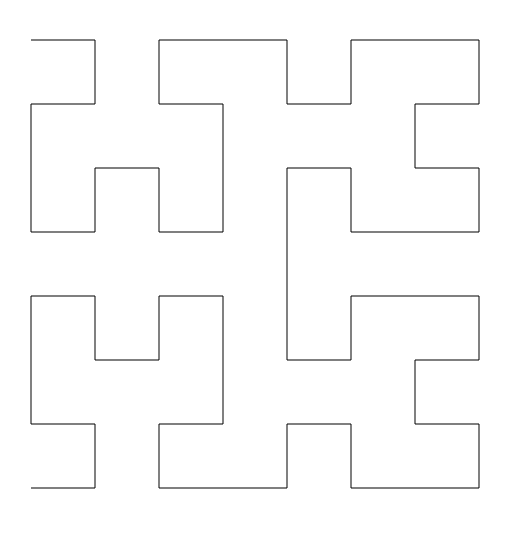


Рисунок 3. Приближение

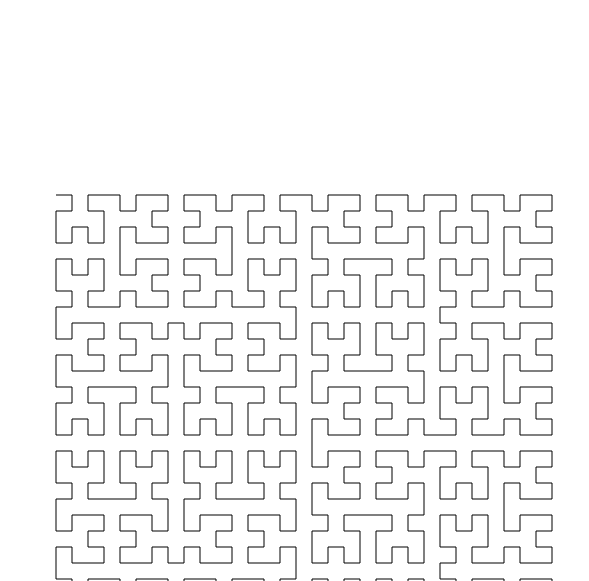


Рисунок 4. Движение

6. Вывод:

Мы получили навыки и знания при реализации алгоритмов с рекурсивными вычислениями. Поняли, как реализовать кривую под названием «Кривая Гильберта».

Узнали, что такое рекурсия и как она образуется. Рекурсия – это определение объекта через обращение к самому себе. Научились делать приближение и движение.