

Отчет по проекту
ФРАКТАЛЫ ЛЯПУНОВА

Лирисман Карина, Б01-001

15 мая 2022 г.

Содержание

1	Установка	3
1.1	Библиотеки	3
1.2	Скачивание	3
1.3	Компиляция и запуск	3
2	Презентация программы	4
2.1	Меню	4
2.2	Навигация	4
3	Выполнение	4
3.1	Управление SDL	4
3.2	Генерация и отображение	5
3.3	Оптимизация и многопоточность	5
3.4	Управление цветом	6
3.5	Взаимодействие с пользователем	6
3.5.1	Увеличение, зум	7
3.5.2	Перемещение	7
3.6	Меню	7
4	Выбор	8
4.1	Язык	8
4.2	Графическая библиотека	8
5	Аналитическая механика	8
6	Скриншоты	8

1 Установка

1.1 Библиотеки

Необходимо установить две библиотеки:

- Debian-based (Debian, Ubuntu, ...)

```
$ sudo apt-get install libSDL2-dev
$ sudo apt-get install libgtkmm-3.0-dev
```

- RedHat-like (Fedora, CentOS, ...)

```
$ sudo yum install libSDL2-dev
$ sudo yum install libgtkmm-3.0-dev
```

1.2 Скачивание

Перейти на GitHub и скачать папку тут, следуя указаниям в разделе *Code*.

1.3 Компиляция и запуск

Заходим в папку ляпунов

```
$ cd AAL/
```

Запустите команды для сборки:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
```

Запустите команду для запуска исполняемого файла:

```
$ ./AAL
```

2 Презентация программы

2.1 Меню

При запуске программы отображается меню для выбора цветов, представляющих значения показателей Ляпунова желаемой точности, то есть количества итераций для вычисления показателя, и последовательности, которая будет использоваться. По нажатию "Подтвердить" запускается генерация и отображаются фракталы. Если пользователь не вводит последовательность при первом открытии меню, то последовательность генерируется случайным образом, чтобы позволить пользователю открывать новые фракталы. Пользователь может получить случайную последовательность через консоль. Меню можно снова открыть, используя клавишу Esc для изменения цветов или изменения точности/последовательности. Если последовательность остается пустой при повторном открытии меню, то последняя последовательность будет учитываться без необходимости повторного ввода пользователем последовательности.

2.2 Навигация

Навигация осуществляется стрелками направления (вверх, вниз, влево, вправо), что позволяет перемещаться на полэкрана. Масштабирование выполняется левым щелчком мыши, а область масштабирования обозначается белым квадратом вокруг указателя мыши. Уменьшение масштаба выполняется правой кнопкой мыши, чтобы вернуться к области перед масштабированием. Чтобы изменить степень масштабирования, используйте колесо мыши.

3 Выполнение

3.1 Управление SDL

Чтобы отобразить фрактал, использовалась SDL, которая представляет собой библиотеку, написанную на C, и поэтому адаптировать ее к C++ с помощью оболочки, которой является класс WindowManager.

Цель этого класса, таким образом, состоит в том, чтобы иметь возможность работать с SDL, избегая указателей, и разрешить разделение между SDL и программой. Этот класс использует текстуру для изменения пикселей, которые нужно отобразить, и визуализацию (которая является элементом SDL) для отображения текстуры на экране. Таким образом, можно легко отображать фрактал, не беспокоясь об операции на уровне SDL. Программа работает с циклом обработки событий, и для каждого интересующего нас события можно обработать его с помощью функции обработки событий вне класса, работающего с SDL.

3.2 Генерация и отображение

Фракталы Ляпунова — это некоторое представление хаотической последовательности, вычисляемое из показателей Ляпунова. Алгоритм генерации двумерных фракталов Ляпунова на плоскости $[0, 4] \times [0, 4]$ делится на несколько этапов. Сначала выбираем последовательность A и B. Затем строится последовательность S_n достаточно длинная, состоящая из повторения последовательности A и B, определенной ранее. Выбираем точку с координатами x, y в начальной плоскости. Затем определяем функцию $R_n = a$ если $S_n = A$ и $R_n = b$ в противном случае. Затем строим последовательность x_n считая $x_{n+1} = r_n x_n (1 - x_n)$ принимая за данность $x_0 = 0.5$. Затем вычисляем отмеченный показатель Ляпунова λ при помощи формулы:

$$\lambda = \lim_{N \rightarrow +\infty} \frac{1}{N} \sum_{n=1}^N \log |r_n(1 - 2x_n)|$$

Затем для отображения фракталов на экране каждому показателю присваиваем цвет, чтобы иметь возможность менять цвета на лету, которые представлены в RGB.

3.3 Оптимизация и многопоточность

Самой затратной операцией при вычислении показателя степени является логарифмирование, но так как вычисляются суммы логарифмов, можно использовать свойство $\log(a) + \log(b) = \log(a \times b)$.

Когда достигается определенный предел (здесь 1.10^{100} : тип `double` позволяет достичь этих достаточно высоких значений), логарифмируем произведение и прибавляем его к конечному результату. Таким образом, вычисляется меньше логарифмов и экономится время вычислений. Для повышения скорости расчета и отображения фракталов Ляпунова также используется распараллеливание. Получаем количество потоков процессора, а затем разрезаем генерацию из блока на несколько блоков, каждый из которых соответствует области фрактала. Различные области фрактала отделены от количества «линий» на графике. Для этого используется нативную библиотеку `<thread>`, которую довольно легко использовать. Действительно, разделив вычисления показателей на разные потоки, можно сэкономить время при расчете фрактала.

3.4 Управление цветом

Цвета берутся из файла конфигурации. Затем сохраняются в таблице и распределяются по красной, зеленой и синей компонентам разницу между максимальным и минимальным цветом по знаку экспоненты, затем цвет, которому эта разница приписана. Сохраняются максимальное и минимальное значения показателя степени вычисляемой области, чтобы создать шкалу цветов, следующих за знаком. Для каждого компонента цвета цветовая разница умножается путем деления экспоненты на соответствующую границу и, наконец, добавляется к сохраненному цвету.

3.5 Взаимодействие с пользователем

Для того, чтобы взаимодействовать с фракталом, нужно уметь преобразовывать экранные координаты в координаты Ляпунова. При этом используем область, которая определяет начальную и конечную координаты по обеим осям. Для преобразования делим для каждой координаты разницу между новой и старой точкой на ширину текстуры, затем умножаем на текущую ширину и добавляем начало координат.

3.5.1 Увеличение, зум

При масштабировании извлекаем область путем преобразования координат экрана и вычисляем область для отображения с этой областью. Чтобы иметь возможность уменьшать масштаб, сохраняем каждую область перед масштабированием, которую помещаем в стопку. С каждым щелчком правой кнопкой восстанавливаем эту область и пересчитываем, чтобы заново отобразить старый фрактал.

3.5.2 Перемещение

Зона, в которую мы хотим переместиться, рассчитывается по текущему региону путем добавления половины экрана. При попытке выйти за границу, регион прилипнет к границе.

3.6 Меню

Меню было сделано с помощью библиотеки <GTKmm>, C++ версии GTK+, в частности графической библиотеки GIMP. Она имеет множество классов, позволяющих пользователю очень легко создавать графические интерфейсы и навигационные меню. Поэтому была выбрана эта библиотека для реализации меню: есть много виджетов, - селекторы цветов или даже области ввода текста, позволяющие пользователю выбирать последовательность и цвета между другими. Меню в основном использует несколько небольших функций, которые упрощают заполнение файла конфигурации. Решено записать различные варианты выбора пользователя в файл конфигурации, чтобы иметь возможность легко и быстро извлекать значения между различными классами, например, без необходимости использовать аксессуар. Меню может записывать различные цвета в форме 255 255 255, введенную последовательность, а также точность, желаемую пользователем в файле конфигурации.

4 Выбор

4.1 Язык

Решено использовать язык C++, потому что он быстрый, что важно для расчетов фракталов, а также для развития собственных навыков программирования.

4.2 Графическая библиотека

Использована SDL, потому что это неизвестная мне библиотека, таким образом освоен новый инструмент для работы. Дополнительно, другие библиотеки показались менее подходящими для попиксельного отображения. Еще использована GTKmm для создания меню из-за простоты использования для создания графического интерфейса.

5 Аналитическая механика

Причастность к курсу МФТИ: в первой половине учебного семестра активно изучались темы по теории устойчивости Ляпунова. Введены понятия фазового пространства, понятие о бифуркации.

Цель: темы бифуркации прямо не были отражены в предложенных задачах, хотя показались мне интересными и важными. Таким образом, для углубления своих знаний по курсу, было решено самостоятельно изучить эту тему.

Выбор задачи: в Интернете я нашла статьи по фракталам Ляпунова, меня привлекли красивые картинки, и я захотела посмотреть, какими они могут быть еще, и в итоге было решено написать программу для фракталов Ляпунова.

6 Скриншоты

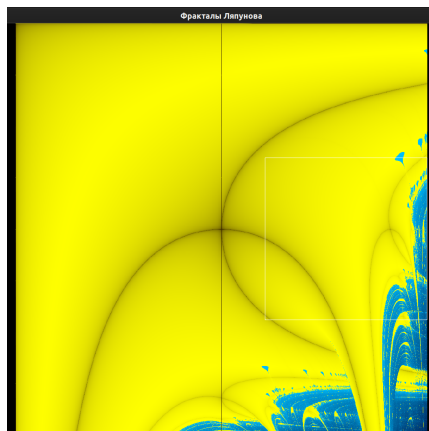


Рис. 6.1: Последовательность АВ

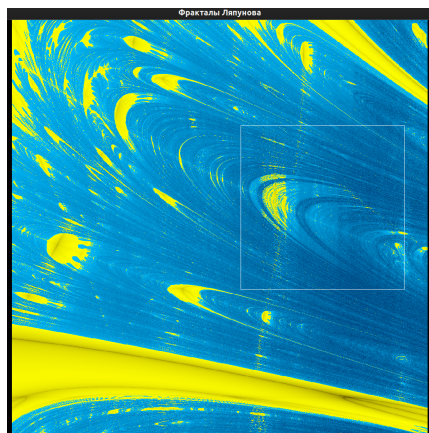


Рис. 6.2: Последовательность BBBBBBAAAAAAAAA

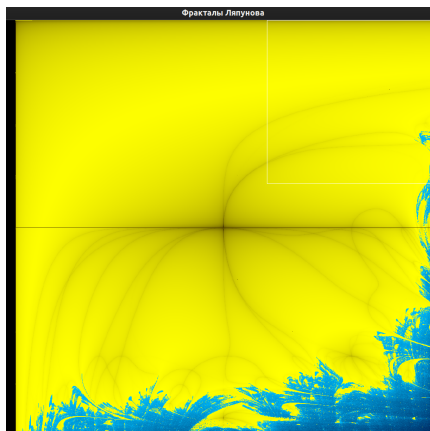


Рис. 6.3: Последовательность BABBBAABV

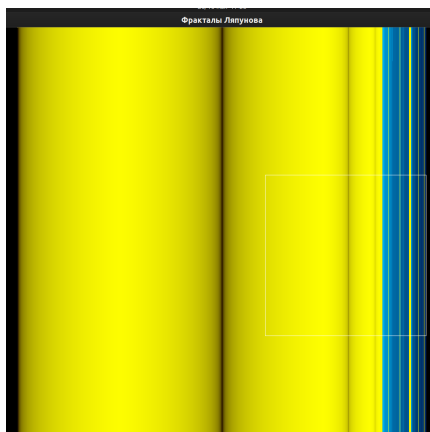


Рис. 6.4: Последовательность AAAAA