

Документация по Drception

Актуально для: build 734 - build last

<https://drception.sourceforge.io/>

Оглавление

Обозначения.....	3
Реализованные фракталы.....	4
Функции доступные из JavaScript.....	6
Подключаемые модули.....	8
Дополнительно.....	9
Добавление фрактала.....	10
Горячие клавиши.....	12
Преобразование сгенерированного фрактала в разные форматы.....	13
I. ImageMagick.....	13
II. ffmpeg.....	13

Обозначения

Обозначения далее:

- `$AppPath$` - путь к программе Drception.
- `[...]` - то, что в квадратных скобках – необязательная часть
- `func()` - обозначение функции с названиее `func`
- `//comment` – однострочный комментарий с текстом `comment` внутри вставки кода
- `/*comment*/` – многострочный комментарий с текстом `comment` внутри вставки кода
- `$VarName$` - введение нового обозначений как переменной, которое будет обговорено при её объявлении
- `<text>`- `text` пока ложен
- `Example:` - означает, что далее следует пример того, что было описано ранее.
- `int` - тип переменной: число целочисленного типа размера 32бита.
- `String` - тип переменной: строка.
- `void` - тип переменной: ничего.

Реализованные фракталы

Таблица:

№	Title	Icon	Уровень реализации
1	Koch Curve	ok	normal
2	Barnsley Fern	ok	normal
3	Sierpinski Carpet	ok	normal
4	Mandelbrot Set	ok	normal
5	Julia Set	ok	normal
6	Burning Ship	ok	normal
7	Sierpinski Carpet 3D by MazyCrazy	none	normal
8	Minkowski Curve	ok	normal
9	Hilbert Curve	ok	normal
10	Gosper Curve	ok	normal
11	Peano Curve	none	normal
12	Circular Fractal	ok	normal
13	Apollonian Gasket	none	normal
14	Lyapunov Fractal	ok	normal
15	Newton Fractal	ok	normal
16	Sierpinski Triangle	ok	normal
17	Menger Sponge	none	normal
18	Heighway Dragon	ok	normal
19	Cantor Set	ok	normal
20	T-fractal	ok	normal
21	H-fractal	ok	normal
22	Durer Star	ok	normal
23	Quasiclover Fractal	ok	normal
24	xxx	none	normal
25	Fibonacci Word Fractal	ok	normal
26	Nested Spiral Squares	ok	normal
27	Wisekk Fractal	ok	normal

28	Pythagoras Tree Fractal	ok	normal
29	Perlin Noise	none	normal
30	Center of mass of a triangle Fractal	ok	normal
31	Curlicue Fractal	ok	normal
32	Levy C Curve	ok	normal

Т.е. на данном этапе реализовано 26/32 фракталов из списка.

Функции доступные из JavaScript

`int frameDuration` — переменная равная длительности одного кадра в анимации в миллисекундах; можно записывать и читать.

«

`void setFrameDuration(int val)` - устанавливает длительность одного кадра анимации в `val` миллисекундах.

`Int getFrameDuration()` - возвращает длительность одного кадра анимации в миллисекундах.

»

`void NextFrame()` - перейти к рисованию следующего кадра анимации, при этом текущий кадр будет возвращен в `Drcption` из скрипта для отображения на экран, а затем будет создан новый холст, на котором сразу же можно продолжать рисовать новый кадр.

`void PainterEnd()` - закончить рисовать, при этом текущий кадр будет возвращен в `Drcption` из скрипта для отображения на экран .

`void SetBuildProgress(int val)` - устанавливает текущий прогресс работы скрипта в `val` процентов (%).

`void WriteLine(String value)` - выводит `value` в новой строке консоли, при этом сначала преобразовав в строку.

`void RandomF(int a)` - возвращает случайное число с плавающей точкой от 0 до 1 с точностью `a` знаков после запятой.

`void SetPixel(int x,int y)` - красит пиксель с координатами `(x,y)` в выбранный цвет пера.

`void DrawLine(int x1,int y1,int x2,int y2)` - рисует линию с координатами вершин `(x1,y1)` и `(x2,y2)` цвета пера.

`void DrawRect()` - кек.

`void DrawPolygon()` - кек.

`void DrawPolyline()` - кек.

`void DrawArc()` - кек.

`void DrawChord()` - кек.

`void DrawEllipse()` - кек.

`void DrawCircle()` - кек.

`void DrawSquare()` - кек.

`void DrawText_Center()` - кек.

`void DrawText_LeftRight()` - кек.

`void FontSet()` - кек.

`void PenSetColor_RgbaF()` - кек.

`void BrushSetColor_RgbaF()` - кек.

`void PenSetColor_HslaF()` - кек.

`void BrushSetColor_HslaF()` - кек.

`void PenSetColor_HsvaF()` - кек.

`void BrushSetColor_HsvaF()` - кек.

`void PenSetColor_СmykaF()` - кек.

void BrushSetColor_CmykaF() - кек.

Подключаемые модули

Они располагаются по пути \$AppPath\$/Scripts/Utility .

«

complex.js — всё для работы с комплексными числами.

//todo

other.js — всё, что не было распределено по другим модулям.

//todo

RgbF_HslF(r,g,b) // rgb → hsl

HslF_RgbF(h,s,l) //hsl → rgb

»

Дополнительно

Файлы в `$AppPath$`:

- `Drcception` — программа на Qt.
- `Drcception.sh` — скрипт запуска программы.
- `clear_personal.sh` — скрипт чистки персональных данных сохраненных программой (чистит папку `SavedFractals` и чистит файл `main_log.txt`); запускать необходимо из каталога `$AppPath$` (то есть этот каталог должен быть рабочим).
- `main_config.ini` — конфигурация программы.
- `def_fractals.ini` — ini-список фракталов по умолчанию, описывает все реализованные на данный момент фракталы, поставляемые с программой.
- `tricks.ini` — ini-список фракталов, который описывает дополнительные штуки, которые можно делать с помощью `Drcception`.
- `main_log.txt` — главный лог `Drcception` (на английском).
- `build_version.txt` — файл содержит одно число (номер билда текущего экземпляра программы).
- `Scripts/` - папка со всеми скриптами.
- `SavedFractals/` - папка с сохраненными фракталами (изображения и xml-конфигурации).
- `Translations/` - папка с переводами программы на разные языки.
- и файлы Qt (`qt.conf` и другие).

Добавление фрактала

Порядок действий:

- 1) Создать файл файл с JavaScript'ом рисования фрактала по пути, который будет являться любым продолжением пути `$AppPath$/Scripts/`. Пусть таким продолжение является `$ScriptFileSubpath$`.

В качестве основы этого файла может выступать любой другой файл, так будет удобно, но рекомендуется, что бы основой был файл `$AppPath$/Scripts/_NotYetImplemented.js`.

О том, как должен быть этот файл читайте в следующем пункте. `WWW`

- 2) Добавьте ваш фрактал в файл `$AppPath$/main_config.ini`.

Для этого добавляем в конец раздела [FractalsList] следующие строки:

```
fractal\l>Title=$FractalName$
fractal\lDesc="<h1>$FractalName$</h1><h3>Programmer:
$Programmer$</h3><h3>Discoverer: $Discoverer$</h3><p>Links:
$Links$</p><p>Type: $AboutType$</p><p>Implementation:
$AboutImplementation$<p>Build Arguments:
$AboutBuildArguments$.</p><p>$OtherDesc$</p>"
fractal\lDimensionsType=$DimensionsType$
fractal\lJsScriptFile=$ScriptFileSubpath$
fractal\lArgumentsStr=
```

Здесь `l` – номер этого фрактала в списке. Он он должен быть на 1 больше номера фрактала передним, таким образом все фракталы идут по порядку, а нумерация начинается с 1.

Заместо слов содержащих знак `$` по обе стороны вы должны подставить свою информацию. Описание этих слов:

`$FractalName$` - название (тайтл) фрактала

`$Programmer$` - человек(люди), написавший скрипт

`$Discoverer$` - человек(люди), изучавшие фрактал, те кто его описали и др.

`$Links$` - ссылки на материалы об этом фрактале в формате HTML

`$AboutType$` - тип фрактала, случит своеобразными метками, которые записываются через запятую

`$AboutImplementation$` - о данной реализации, этот нужно для тех случаев, когда один и тот же фрактал можно реализовать разными способами, в таком случае придётся добивать отдельные пункты для каждой реализации, но в этом месте указать, что есть у другие реализации, и чем эта отличается от тех; этот пункт связан с пунктов `$Programmer$`, так как они оба зависят от текущей реализации

`$AboutBuildArguments$` - скрипт может содержать в себе много разных функций для построения фрактала, которые будут отличаться тем, какие параметры им нужно передавать, в данном пункте и описывается использование этих функций. В данной версии программы `Drcception` не возможно использовать никакие функции, кроме `BuildFractal(<arguments>)`, где `<arguments>` - все аргументы через запятую. Если же никакие аргументы не заданы, то автоматически используется функция

`BuildFractalDefRect(x1,y1,x2,y2)` – которая задаёт, в какой области рисовать фрактал (это прямоугольник с левой нижней вершиной в $(x1,y1)$ и правой верхней вершиной в $(x2,y2)$). Так же необходима реализация функции `BuildFractalDef()`, которая в данной версии не используется (можете просто делать из нее вызов `BuildFractalDefRect(0,0,500,500)`).

`$OtherDesc$` - всё остальное описание, которое вы хотите добавить, соответственно в формате HTML.

`$DimensionsType$` - тип фрактала по измерениям: 0 – 2d, 1 – 3d.

`$ScriptFileSubpath$` - продолжение пути `$AppPath$/Scripts`, которое упоминалось раньше, которое является путём к скрипту фрактала относительно пути `$AppPath$/Scripts`.

!!Внимание!! Файл `$AppPath$/main_config.ini` можно редактировать произвольным образом, например менять местами существующие элементы, если вам необходимо, но перед тем как это делать убедитесь в том, что понимаете, что делаете, в противном случае после изменений программа может просто не запуститься.

Что бы изменения в файле `$AppPath$/main_config.ini` вступили в силу необходимо перезапустить программу.

Теперь увеличиваем значение `count` в том же разделе `[FractalsList]` на 1, так что бы новый добавленный в конец элемент в сформированный список.

- 3) Всё, теперь фрактал виден в `Drception` и с ним можно работать так же, как и с остальными фракталами.

Меню и Горячие клавиши

Больше сочетаний клавиш смотри в выпадающем меню программы.

Главное меню

Главное меню:

- **Файл**
 - Открыть Список Фракталов
 - Печать...
 - -----
 - Отчистить Список Фракталов
 - Отчистить Всё
 - -----
 - Недавние Файлы
 - <1>
 - <2>
 - <3>
 - -----
 - Выход
- **Инструменты**
 - Проигрывать Анимацию
 - Пауза Анимации
 - Остановить Анимацию
 - -----
 - След Кадр
 - Пред Кадр
 - -----
 - Перестроить
 - Полный экран
 - Отображение GUI
 - Отображение Фрактала
 - Отображение Сетки
 - Grid Settings

- New Size Settings
- Поворот
- -----
- Сохранить Изображение
- Сохранить Избражения Всех Кадров
- -----
- Опции
- Задачи
 - Добавить
 - Удалить
 - -----
 - Up
 - Down
 - -----
 - Отчистить
 - -----
 - Начать Процессинг
 - Пауза Процессинг
 - Остановить Процессинг
- Скрипт
 - Сохранить
 - Перезагрузка
 - Отчистить
 - -----
 - Назад
 - Вперед
 - -----
 - Выделить Всё
 - Вырезать
 - Копировать
 - Вставить
 - Удалить
 - -----
 - Поиск
 - Замена

- Помощь
 - Помощь
 - О Qt
 - О Drception

Преобразование сгенерированного фрактала в разные форматы

I. ImageMagick

Это команды из пакета ImageMagick. Подробнее см. Документацию ImageMagick.

Создание Gif из изображений кадров

В папке, в которой находятся только изображения фрактала в том виде, в котором их сохранил Drcption, то достаточно выполнить команду в этой папке:

```
convert -delay 100 -loop 0 *_{0..$frame_last_i$}.png anim.gif
```

, где вместо `$frame_last_i$` необходимо вставить номер последнего кадра, т. е. так как нумерация кадров начинается с нуля, то `$last_last_i$` = количество кадров - 1.

Таким образом результат будет сохранён в файл `anim.gif` в той же папке.

II. ffmpeg

Это команды из пакета ffmpeg. Подробнее см. Документацию ffmpeg.

С помощью ffmpeg можно получить не только GIF анимацию, но и анимацию почти в любом формате видео, при этом можно наложить дополнительную информацию на кадр, наложить звук, изменить качество и т.д.

Создание Gif из изображений кадров

```
ffmpeg -f image2 -r $frame_rate$ -i "$date_time$__0_%d_i.png" -r $frame_rate$ anim.gif
```

`$date_time$` - дата и время, кода был сохранен фрактал. По сути, строка `$date_time$__0_` должна просто являться перфексом всех файлов изображений кадров фрактала.

`$frame_rate$` = 1000 / 'длительность одного кадра'

Ex:

```
ffmpeg -f image2 -r 10 -i "20-15-19-147_31-12-2017__0_%d_i.png" -r 10 anim.gif
```

Создание FLV из изображений кадров

```
ffmpeg -f image2 -i *_%d_i.png -r $frame_rate$ anim.flv
```

`$frame_rate$` = 1000 / 'длительность одного кадра'

Ex:

```
ffmpeg -f image2 -r 1 -i "20-15-19-147_31-12-2017__0_%d_i.png" -r 1 anim.flv
```

Создаие FLV из изображений кадров, со звуком

```
ffmpeg -f image2 -i *_%d_i.png -i sound.ogg anim.flv
```

Ex:

```
ffmpeg -f image2 -r 1 -i "20-15-19-147_31-12-2017__0_%d_i.png" -i sound.ogg -r 1
```

anim.flv

Создание FLV из изображений кадров, с наложение изображения (с водяным знаком)

Пример с добавление логотипа Drcption на протяжении всего видео поверх него в правом верхнем углу с отступом от этого угла 1 и 1 пикселей по двум осям:

```
ffmpeg -f image2 -r 1 -i "*"_%d_i.png" -i $logo -filter_complex 'overlay=main_w-overlay_w-1:1' -r 1 anim.flv
```

Ex:

```
ffmpeg -f image2 -r 1 -i "20-15-19-147_31-12-2017__0_%d_i.png" -i logo.png -filter_complex 'overlay=main_w-overlay_w-1:1' -r 1 anim1.flv
```

Примечания:

- Для улучшения качества вывода, можно добавить `-qscale 0` перед `anim.flv` (путем выходного файла). Это отключит автоматическое масштабирование всех изображений.
- По умолчанию `ffmpeg` постоянно спрашивает о перезаписи, если файл по пути выходного файла уже существует. Чтобы автоматически перезаписывать выходной файл добавти `-u` перед путем выходного файла.
- По умолчанию `ffmpeg` создаёт пустую звуковую дорожку, если создавать видео из изображений. Чтобы звуковая дорожка вообще не создавалась в выходном файле можно добавить `-an` перде путем выходного файла.
- Формат выходного файла может быть почти любой (см. Документацию `ffmpeg` и её поддерживаемые выходные форматы). Для большинства случаев необходимо просто поменять расширение выходного файла. Например в конец команды вы можете вместо `anim.flv` указывать `anim.mp4`, `anim.mkv`, `anim.wmv` и т.д.