# Документация по Drception

Актуально для: build 833 - build last

**https://sourceforge.net/projects/drception/**
**https://drception.sourceforge.io/**
**E-mail: programmingspecial@gmail.com**

# Оглавление

# Обозначения

Обозначения далее:

- $AppPath$ - путь к программе Drception.
- [ … ] - то, что в квадратных скобках – необязательная часть
- func() - обозначение функции с название 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 | Perlin Noise | ok | normal |
| 8 | Minkowski Curve | ok | normal |
| 9 | Hilbert Curve | ok | normal |
| 10 | Gosper Curve | ok | normal |
| 11 | Peano Curve | ok | normal |
| 12 | Circular Fractal | ok | normal |
| 13 | Apollonian Gasket | ok | normal |
| 14 | Lyapunov Fractal | ok | normal |
| 15 | Newton Fractal | ok | normal |
| 16 | Sierpinski Triangle | ok | normal |
| 17 | Heighway Dragon | ok | normal |
| 18 | Cantor Set | ok | normal |
| 19 | T-fractal | ok | normal |
| 20 | H-fractal | ok | normal |
| 21 | Durer Star | ok | normal |
| 22 | Quasiclover Fractal | ok | normal |
| 23 | Fibonacci Word Fractal | ok | normal |
| 24 | Nested Spiral Squares | ok | normal |
| 25 | Wisekk Fractal | ok | normal |
| 26 | Pythagoras Tree Fractal | ok | normal |
| 27 | Center of mass of a triangle Fractal | ok | normal |

| | | | |
|---|---|---|---|
| 28 | Curlicue Fractal | ok | normal |
| 29 | Levy C Curve | ok | normal |
| 30 | Menger Sponge (3D Fractal) | none | normal |
| 31 | Sierpinski Carpet 3D by MazyCrazy (3D Fractal) | none | normal |

Т.е. на данном этапе реализовано 29/31 (первые 29) 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 Drception 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 Drception из скрипта для отображения на экран .

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

void WriteLn(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_LeftTop() - 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_CmykaF() - 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) // rbg → hsl
HslF_RgbF(h,s,l) //hsl → rgb
```

»

# Дополнительно

Файлы в $AppPath$:

- Drception — программа на Qt.
- Drception.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 Drception.
- main_log.txt — главный лог Drception (на английском).
- build_version.txt — файл содержит одно число (номер билда текущего экземпляра программы).
- Scripts/ - папка со всеми сриптами.
- SavedFractals/ - папка с сохраненными фракталами (изображения и xml-конфигурациии).
- 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 $AppPath$/Scirpts/_NotYetImplemented.js.

   О том, как должен быть этот файл читайте в следующем пункте. \/\/\/
2) Добавьте ваш фрактал в файл $AppPath$/main_config.ini.

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

   *fractal\i\Title=$FractalName$*

   *fractal\i\Desc="<h1>$FractalName$</h1><h3>Programmer: $Programmer$</h3><h3>Discoverer: $Discoverer$</h3> Links: $Links$ Type: $AboutType$ Implementation: $AboutImplementation$ Build Arguments: $AboutBuildArguments$. $OtherDesc$ "*

   *fractal\i\DimensionsType=$DimensionsType$*

   *fractal\i\JsScriptFile=$ScriptFileSubpath$*

   *fractal\i\ArgumentsStr=*

   Здесь I – 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 Drception 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,y2) 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 *BuildractalDefRect(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 - todo.
  - Открыть Список Фракталов (<no>) - todo.
  - Печать... (<no>) - todo.
  - ----------
  - Отчистить Список Фракталов (<no>) - todo.
  - Отчистить Всё (<no>) - todo.
  - ----------
  - Недавние Файлы - todo.
    - <1> (<no>) - todo.
    - <2> (<no>) - todo.
    - <3> (<no>) - todo.
  - ----------
  - Выход (<no>) - todo.
- Инструменты - todo.
  - Проигрывать Анимацию (<no>) - todo.
  - Пауза Анимации (<no>) - todo.
  - Остановить Анимацию (<no>) - todo.
  - ----------
  - След Кадр (F12) - todo.
  - Пред Кадр (F11) - todo.
  - ----------
  - Перестроить (F5) - todo.
  - Полный экран (Esc) - todo.
  - Отображение GUI (F3) - todo.
  - Отображение Фрактала (F4) - todo.
  - Отображение Сетки (<no>) - todo.
  - Настройки Сетки (<no>) - todo.

    - Настройки Нового Размера (Ctrl+R) - todo.
    - Поворот (F6) - todo.
    - ----------
    - Сохранить Изображение (F2) - todo.
    - Сохранить Избражения Всех Кадров (<no>) - todo.
    - ----------
    - Опции (<no>) - todo.
- Задачи - todo.
    - Добавить Задачу (F9) - todo.
    - Удалить Задачу (F8) - todo.
    - ----------
    - Переместить Вниз (Shift+Up) - todo.
    - Переместить Вверх (Shift+Down) - todo.
    - ----------
    - Отчистить Список Задач (Shift+C) - todo.
    - ----------
    - Начать Процессинг (Shift+F9) - todo.
    - Пауза Процессинг (Shift+F10) - todo.
    - Остановить Процессинг (Shift+F11) - todo.
- Скрипт - todo.
    - Сохранить Файл (Ctrl+S) - todo.
    - Перезагрузить Файл (Ctrl+O) - todo.
    - Отчистить Файл (<no>) - todo.
    - ----------
    - Назад (Ctrl+Z) - todo.
    - Вперед (Ctrl+Y) - todo.
    - ----------
    - Выделить Всё (Ctrl+A) - todo.
    - Вырезать (Ctrl+X) - todo.
    - Копировать (Ctrl+C) - todo.
    - Вставить (Ctrl+V) - todo.
    - Удалить (Del) - todo.
    - ----------
    - Поиск (Ctrl+F) - todo.
    - Замена (Ctrl+R) - todo.

- Помощь - todo.
  - Помощь (<no>) - todo.
  - О Qt (<no>) - todo.
  - О Drception (<no>) - todo.

# 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 находяться только изображения фрактала в том виде, в котором их сохранил Drception, то достаточно выполнить команду в этой папке:

convert -delay 100 -loop 0 *_{0..$frame_last_i$}_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.

**Замечание.**

Если возникают следующие ошибки:

```
convert -delay 100x1000 -loop 0 $AppPath$/SavedFractals/16-02-42-739_24-02-2018_{0..315}_i.png $AppPath$/SavedFractals/16-02-42-739_24-02-2018_a_i.gif
convert-im6.q16: DistributedPixelCache '127.0.0.1' @ error/distribute-cache.c/ConnectPixelCacheServer/244.
convert-im6.q16: cache resources exhausted `$AppPath$/SavedFractals/16-02-42-739_24-02-2018_181_i.png' @ error/cache.c/OpenPixelCache/3982.
convert-im6.q16: DistributedPixelCache '127.0.0.1' @ error/distribute-cache.c/ConnectPixelCacheServer/244.
convert-im6.q16: cache resources exhausted `$AppPath$/SavedFractals/16-02-42-739_24-02-2018_182_i.png' @ error/cache.c/OpenPixelCache/3982.
...
convert-im6.q16: too many exceptions (exception processing suspended).
```

То необходимо изменить в файле /etc/ImageMagick-6/policy.xml ограничения на размер памяти в ImageMagick, для этого в строчке `<policy domain="resource" name="memory" value="256MiB"/>` замените 256MiB на другой размер, например на 2GiB. Вы так же можете изменить другие ограничения в ImageMagick в этом же файле.

Для проверки установленных ограничений в ImageMagick запустите команду:

```
identify -list resource
```

Результат будет примерно следующим:

```
Resource limits:
  Width: 16KP
  Height: 16KP
  Area: 128MP
  Memory: 2GiB
  Map: 512MiB
  Disk: 1GiB
```

```
File: 768
Thread: 4
Throttle: 0
Time: unlimited
```

# II. ffmpeg

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

С помощью ffmpeg можно получить не только GIF анимацию, но и анимацию почти в любом формате видео, при этом можно наложить дополнительную информацию на кард, наложить звук, изменить качетво 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_ должа 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 Drception 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 -y перед путем выходного файла.
- По умолчанию ffmpeg создаёт пустую звуковую дорожку, если создавать видео из изобрадений. 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 всместо anim.flv указывать anim.mp4, anim.mkv, anim.wmv и т. д.