

Документация по Drception

Актуально для: build 970

SourceForge: <https://sourceforge.net/projects/drception/>

Website (SourceForge): <https://drception.sourceforge.io/>

E-mail: programmingspecial@gmail.com

Оглавление

Обозначения.....	3
Реализованные фракталы.....	4
Первые шаги.....	6
Построение фрактала из списка.....	6
Функции и модули доступные из JavaScript.....	7
Global Space.....	7
AutoInclude.js.....	9
Дополнительно.....	12
Файлы в \$AppPath\$.....	12
Устройство ini-списка фракталов.....	14
Меню и Горячие клавиши.....	15
Преобразование сгенерированного фрактала в разные форматы.....	17
I. ImageMagick.....	17
II. ffmpeg.....	18
Заключение.....	20

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, это многострочный аналог „todo“
- Например: — означает, что далее следует пример того, что было описано ранее.

Реализованные фракталы

C Drcption распространяется следующие списки фракталов (располагаются в \$AppPath\$):

- def_fractals.ini — содержит все основные фракталы.
- tricks.ini — содержит примеры, показывающие, что Drcption — не только программа для работы с фракталами, но и полноценная среда разработки JavaScript, позволяющая писать произвольные программы.

Таблица def_fractals.ini:

№	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	Vicsek 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	Weierstrass Function	ok	normal
31	Blancmange Curve	ok	normal
32	Menger Sponge (3D Fractal)	none	normal
33	Sierpinski Carpet 3D by MazyCrazy (3D Fractal)	none	normal

Т.е. на данном этапе реализовано 31/33 (первые 31) фракталов из списка.

Таблица tricks.ini:

№	Title	Icon	Уровень реализации
1	Factorial	ok	normal
2	z-function	ok	normal

Первые шаги

Построение фрактала из списка

В открытом в левой панели («Список Фракталов») нажмите по необходимому фракталу дважды ЛКМ. В результате этот пункт в списке активируется и будет обозначен флажком.

Теперь:

- Главное окно Drcption сменил название так, что оно будет содержать заголовок выбранного фрактала.
- В нижней панели («Опис Фрактала») откроется описание выбранного фрактала.
- В правой панели («Редактирование Скрипта») появится список доступных реализаций и откроется скрипт фрактала первой из них.
- На дополнительной центральной панели («Аргументы фрактала») появится список доступных командных строк.

Чтобы построить выбранный таким образом фрактал, необходимо добавить его в список задач и выполнить этот список задач. Всё это с легкостью можно проделать при помощи главного меню Drcption. Перед добавлением фрактала в список задач вы можете выбрать реализацию и командную строку (из списка или ввести её сами).

При выполнении списка задач на панель («Консоль скрипта») будет выводиться вся необходимая информация (информация о выполнении скрипта, вывод скрипта, сообщения о сохранении изображений и видео-анимаций фрактала). Изображений и видео-анимации фрактала сохраняются автоматически по пути `$AppPath$/SavedFractals/$UniqueName$`, где `$UniqueName$` - уникальное имя файла, выбираемое на основе даты и времени, а так же на основе дополнительной информации о файле. О том, как сохраняются видео-анимации см. в разделе «Преобразование сгенерированного фрактала в разные форматы».

Функции и модули доступные из JavaScript

Модули:

- Global Space — все обязательные методы и поля среды, в которой исполняется скрипт построения фракталов, предоставляемые ему программой Drcption.
- AutoInclude.js — JavaScript-файл, автоматически подключаемый к среде исполнения скриптов построения фракталов.
- Todo

Global Space

Эти методы и поля реализованы на C++, так что здесь будут приведены их объявления на псевдо-C++ для лучше понимания какие значения JavaScript вне зависимости от типов можно им передавать.

Объяснение типов:

- int — целое знаковое 32-разрядное число
- double — вещественное знаковое 32-разрядное число
- void — ничего (null)
- String — любая строка
- Vector — вектор (массив)
- {...} — структура (запись), например: {int x, int y} — структура с двумя полями, int x и int y, где int — тип поля, а затем следует его название.

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(int x, int y, int w, int h)`

— рисует прямоугольник с левым верхним углом в точке (x,y) и размером (w,h).

`void DrawPolygon(Vector<{int x, int y}> sv)`

— рисует полигон с токами в (x,y) из массива sv.

`void DrawPolyline(Vector<{int x, int y}> sv)`

— рисует полилайн с токами в (x,y) из массива sv.

`Void DrawArc(int x, int y, int w, int h, int starta, int spana)`

— рисует дугу заданную прямоугольником с левым верхним углом в точке (x,y) и размером (w,h), а так же с начальным углом starta и конечным углом spana; starta и spana должны быть указаны в 1/16 градуса..

`void DrawChord(int x, int y, int w, int h, int starta, int spana)`

— рисует хорду заданную прямоугольником с левым верхним углом в точке (x,y) и размером (w,h), а так же с начальным углом starta и конечным углом spana; starta и spana должны быть указаны в 1/16 градуса.

`void DrawEllipse(int x, int y, int w, int h)`

— рисует эллипс вписанный в прямоугольник с левым верхним углом в точке (x,y) и размером (w,h).

`void DrawCircle(int x, int y, int r)`

— рисует круг с центром в точке (x,y) и радиусом r.

`void DrawSquare(int xc, int yc, int xa, int ya)`

— рисует квадрат с двумя противоположными углами в точках (xc,yc) и (xa,ya) (левый верхний и правый нижний углы соответственно).

`void DrawText_Center(int x1, int y1, int x2, int y2, String str)`

— выводит строку str в центре прямоугольной области (x1,y1,x2,y2).

`void DrawText_LeftTop(int x1, int y1, int x2, int y2, String str)`

— выводит строку str в левом верхнем углу прямоугольной области (x1,y1,x2,y2).

`void FontSet(String fontFamily, double fontSize)`

— устанавливает шрифт семьи `fontFamily` и размера `fontSize`.

`void PenSetColor_RgbaF(double r, double g, double b, double a)`

— устанавливает цвет пера в палитре RGBA со значениями каналов соответственно `r,g,b,a`, где все значения — это вещественные числа от 0 до 1.

`void BrushSetColor_RgbaF(double r, double g, double b, double a)`

— устанавливает цвет кисти в палитре RGBA со значениями каналов соответственно `r,g,b,a`, где все значения — это вещественные числа от 0 до 1.

`void PenSetColor_HslaF(double h, double s, double l, double a)`

— устанавливает цвет пера в палитре HSLA со значениями каналов соответственно `h,s,l,a`, где все значения — это вещественные числа от 0 до 1.

`void BrushSetColor_HslaF(double h, double s, double l, double a)`

— устанавливает цвет кисти в палитре HSLA со значениями каналов соответственно `h,s,l,a`, где все значения — это вещественные числа от 0 до 1.

`void PenSetColor_HsvaF(double h, double s, double v, double a)`

— устанавливает цвет пера в палитре HSVa со значениями каналов соответственно `h,s,v,a`, где все значения — это вещественные числа от 0 до 1.

`void BrushSetColor_HsvaF(double h, double s, double v, double a)`

— устанавливает цвет кисти в палитре HSVa со значениями каналов соответственно `h,s,v,a`, где все значения — это вещественные числа от 0 до 1.

`void PenSetColor_CmykaF(double c, double m, double y, double k, double a)`

— устанавливает цвет пера в палитре CMYKA со значениями каналов соответственно `c,m,y,k,a`, где все значения — это вещественные числа от 0 до 1.

`void BrushSetColor_CmykaF(double c, double m, double y, double k, double a)`

— устанавливает цвет кисти в палитре CMYKA со значениями каналов соответственно `c,m,y,k,a`, где все значения — это вещественные числа от 0 до 1.

AutoInclude.js

Код модуля разбит на условные группы.

Определения далее используют синтаксис JavaScript.

Группа «Additional»

`DrawNotYetImplemented(x1, y1, x2, y2)`

— выводит в центре прямоугольной области `(x1, y1, x2, y2)` строку "Not yet implemented".

Группа «My Math»

`sqr(a)`

— возвращает квадрат значения `a`.

`pow(a)`

— возвращает результат возведения числа `a` в натуральную степень `n`.

`sh(x)`

— возвращает гиперболический синус числа x .

`ch(x)`

— возвращает гиперболический косинус числа x .

`th(x)`

— возвращает гиперболический тангенс числа x .

`cth(x)`

— возвращает гиперболический котангенс числа x .

`sech(x)`

— возвращает гиперболический секанс числа x .

`csch(x)`

— возвращает гиперболический косеканс числа x .

Группа «Complex numbers»

`Complex(x, y)`

— конструктор объекта- экземпляра класса `Complex` (комплексное число), где x - действительная часть, а y - мнимая часть; если x - это не число, а другой объект-экземпляр класса `Complex`, то конструктор создаст копию x .

`z.copу(a)`

— делает объект-экземпляр класса `Complex` z копией объекта-экземпляра класса `Complex` a .

Далее функции начинающиеся с z являются операциями с комплексными числами (объектами-экземплярами класса `Complex`).

`zadd(a, b)`

— возвращает сумму a и b .

`zsub(a, b)`

— возвращает разность a и b .

`zmul(a, b)`

— возвращает произведение a и b .

`zdiv(a, b)`

— возвращает частное a и b .

`zabs(a)`

— возвращает модуль a .

`zabs2(a)`

— возвращает модуль a возведенный в квадрат.

`zpow(a)`

— возвращает результат возведения a в натуральную степень n .

`zsin(a)`

— возвращает синус a .

`zcos(a)`

— возвращает косинус a .

`ztg(a)`

— возвращает тангенс a .

`zctg(a)`

— возвращает котангенс a .

Вы можете свободно добавлять в этот модуль то, что вам будет часто необходимо. Оно будет автоматически доступно из всех скриптов фракталов.

В настоящий момент реализован аналогичный класс `Complex` на C++, одна из-за его платформы, вместо ожидаемой оптимизации, он работает значительно медленнее, чем используемая сейчас реализация на JavaScript. В силу этого он был отключён и полностью заменен на реализацию на JavaScript. Разработчики постараются всё-таки решить эту проблему в будущем.

Вы так же вместо использования уже реализованного разработчиками класса `Complex` можете использовать свой собственный, или вообще в явном виде вычислять в основном коде (без использования функций) действительную и мнимую часть всех комплексных чисел, храня их как обычные объекты JavaScript. В последнем случае будет достигнута максимальная производительность, которая будет сильно заметна уже при обработке 1е6 операций с комплексными числами. В большинстве скриптов, в которых есть функции использующие класс `Complex` от разработчиков `Drcption`, содержат так же версии этих функций без этого класса (у этих копий в отличии от оригинала отсутствует суффикс «`Complex`» в названии). Все такие скрипты используют по умолчанию версии функций с классом `Complex` от разработчиков `Drcption`, но вы можете ускорить работу скрипта, заменив названия используемых функций.

Дополнительно

Файлы в `$AppPath$`

Список

- `Logs/` - папка с логами `Drcception`.
 - `clear_logs.sh` — отчищает эту папку (папку `Log`); запускать необходимо из каталога `$AppPath$` (то есть этот каталог должен быть рабочим).
 - `[main_log.log]` — главный лог `Drcception` (на английском).
- `SavedFractals/` — папка с сохраненными фракталами (изображения и `xml`-конфигурации).
 - `clear_saved_fractals.sh` — отчищает эту папку (папку `SavedFractals`); запускать необходимо из каталога `$AppPath$` (то есть этот каталог должен быть рабочим).
 - `[...]`
- `Scripts/` — папка со всеми скриптами.
 - `Utility/` — дополнительные файлы-утилиты для скриптов.
 - `AutoInclude.js` — модуль для скриптов построения фрактала, см. подробнее «Функции и модули доступные из JavaScript».
 - `[...]` — множество скриптов фракталов и описаний фракталов.
- `Translations/` — папка с переводами программы на разные языки.
 - `drcception_en.qm` — английский перевод.
 - `drcception_ja.qm` — японский перевод.
 - `drcception_ru.qm` — русский перевод.
- `build_version.txt` — файл содержит одно число (номер билда текущего экземпляра программы).
- `clear_personal.sh` — скрипт чистки персональных данных сохраненных программой (чистит папку `SavedFractals` и папку `Logs`); запускать необходимо из каталога `$AppPath$` (то есть этот каталог должен быть рабочим).
- `completer_sl.txt` — дерево подсказок для поля ввода скрипта построения фрактала.
- `def_fractals.ini` — `ini`-список фракталов по умолчанию, описывает все реализованные на данный момент фракталы, поставляемые с программой.
- `Drcception` — программа на `Qt`.
- `Drcception.sh` — скрипт запуска программы.
- `files_list.txt` — список всех файлов `Drcception`.
- `main_config.ini` — конфигурация программы.
- `README.txt` — `readme`-файл `Drcception`.
- `tricks.ini` — `ini`-список фракталов, который описывает дополнительные штуки, которые можно делать с помощью `Drcception`.

- файлы Qt:
 - qt.conf — файл-конфиг Qt.
 - qtfiles/ — папка с остальными файлами Qt.
- [файлы созданные оффлайн-инсталлятором Qt IFW]:
 - Licenses/
 - license.txt — лицензия Drcption.
 - components.xml
 - InstallationLog.txt
 - maintenancetool — программа «инструменты для обслуживания», она позволяет добавлять, обновлять или удалять компоненты Drcption или Drcption полностью.
 - maintenancetool.dat
 - maintenancetool.ini
 - network.xml

Устройство ini-списка фракталов

Этот подраздел позволит понять, как Drcption хранит информацию о зарегистрированных фракталах. Прочитав этот раздел вы так же сможете добавить свой фрактал.

Для понимания этого раздела также необходимо понимать, как устроен ini-файл.

Структура ini-списка фракталов:

- [FractalsList]
 - count — целое число, обозначающее количество фракталов в списке.
 - fractal — список фракталов
 - i — где i - это целое число от 1 до count, обозначающее индекс фрактала в списке. В результате такой объект и является фракталом.
 - Desc — описание фрактала как одна строка.
 - DescFile — описание фрактала как файл браузера (точнее относительный путь к такому файлу), подразумевается, что это html-файл.
 - DimensionsType — не используется.
 - Title — заголовок (название фрактала).
 - Icon — иконка фрактала (разработчиком используется две иконки: попе и ok, которые обозначают нереализованный и реализованный фрактал соответственно).
 - ScriptFile — скрипт построения фрактала как файл (точнее относительный путь такому файлу). Это поле можно оставить пустым, либо оно вообще может отсутствовать, тогда будут использованы следующие ниже поля, в противном случае в них нет необходимости.
 - ImplementationCnt — количество реализаций (количество различных скриптов для данного фрактала).
 - Implementation — список реализаций
 - j — где j - это целое число от 1 до ImplementationCnt, обозначающее индекс реализации в списке.
 - ScriptFile — скрипт построения фрактала как файл (точнее относительный путь такому файлу).
 - Title — заголовок (название) реализации.
 - CommandlineCnt — количество командных строк, которые вызывают скрипт и которые являются пресетами, а следовательно доступны для быстрого вызова из списка прямо в Drcption.
 - Commandline — список командных строк.
 - k — где k - целое число от 1 до CommandlineCnt, обозначающее индекс командной строки в списка.
 - Str — командная строка как строка.

Меню и Горячие клавиши

Больше сочетаний клавиш смотри в выпадающем меню программы.

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 Изображения 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

Drcption автоматически пытается создать:

- gif-анимацию с помощью ImageMagick с заданным frameDuration.
- gif-анимацию с помощью ffmpeg с выходной частотой кадров равной 30 и частотой кадров источника равной 1000/frameDuration.
- flv-анимацию с помощью ffmpeg с выходной частотой кадров равной 30 и частотой кадров источника равной 1000/frameDuration.
- mp4-анимацию с помощью ffmpeg с выходной частотой кадров равной 30 и частотой кадров источника равной 1000/frameDuration.

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..$frame_last_i$}_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.

Например:

```
todo
```

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 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'

Например:

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

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

Например:

```
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 -r $frame_rate$ -i *_%d_i.png -i sound.ogg -r $frame_rate$ anim.flv
```

Например:

```
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 наложением изображения (с водяным знаком)

Пример с добавлением 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 $frame_rate$ -i "*_%d_i.png" -i $logo -filter_complex 'overlay=main_w-overlay_w-1:1' -r $frame_rate$ anim.flv
```

Например:

```
ffmpeg -f image2 -r 10 -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 10 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 добавте `-y` 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` перед 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.

Заключение

Недостаточно информации, остались вопросы?

Проверь:

- SourceForge: <https://sourceforge.net/projects/drception/>
- Website (SourceForge): <https://drception.sourceforge.io/>
- E-mail: programmingspecial@gmail.com