**Основы работы в программе Zapilyator 3, Gif Annimator 5.0, BPM2SCR, ZX Paint, Vortex Tracker, Poly 3D.**

*Инструкция А.А.А., дополняется по мере появления времени.  
В редакции 5.0 от 18.01.2026 г.*

Страница **Zapilyator**, инструкции, вспомогательные программы, примеры демонстраций: <https://zxaaa.net/AAABAND3.html>

# Вступление.

Программа **Zapilyator 3 -** это глючный, но мощный автоматический сборщик музыкальных демонстраций для компьютера **ZX Spectrum 128k**. Не требует знания языков программирования. Всё что нужно - это умение делать красивый дизайн картинки, рисовать, мыслить как художник и слегка поверхностные знания основ устройства экрана ZX Spectrum, которые я дам в этой книге. Самое главное чтобы у Вас было желание сделать свою демонстрацию. Знания, которые я напишу в этой книге нужны чтобы с пониманием создавать демонстрацию. Даже если Вы ничего не умеете, но есть желание сделать демонстрацию для ZX Spectrum, у Вас должно получиться после прочтения этой книги. Все необходимые программы я дам. Программа **Zapilyator** позволяет выводить четыре .gif объекта монохромной (два цвета черный 0,0,0 и белый 255,255,255) анимации стандартного формата IBM PC .gif. При этом совершенно не имеет значения, сколько одновременно прокручивается объектов внутри одной анимации при одном количестве фаз спрайта, предел только в скорости прерывания компьютера, памяти и размере экрана в 256 на 192 точек. Так же можно проигрывать до 10 мелодий формата .pt3 с переключением цифровыми кнопками от 1,2,3,4,5,6,7,8,9,0. По умолчанию автоматически проигрываются все мелодии одна за другой с затуханием в конце. Эта функция вшита в **Zapilyator**, поэтому её не надо настраивать она будет всегда работать, когда в демонстрации несколько мелодий. Количество мелодий, которые влезут, зависит от оставшейся памяти после заполнения анимацией, скроллами, текстом и прочими желаниями автора демонстрации. Помним, что на ZX Spectrum у нас есть только 128 кб и в этих ограниченных рамках мы делаем шедевр, оптимизируя спрайты и графику. Программа сама расставляет все загружаемые в неё объекты по памяти. Так же **Zapilyator** скомпилирует единый блок готовой программы, чтобы на дискете не было много файлов, и готовая демонстрация занимала меньше секторов. Ещё есть возможность запуска горизонтального регулируемого по длине скролла на 16 точек или скролла на 8 точек, как горизонтального, так и вертикального, регулируемого во все стороны и по длине. Шрифт скролла можно менять на любой 8 точечный стандартный, например сделанный в Art Studio. Или же взятый из моей подборки, которую я приложил на страничку **Zapilyator** в паке «вспомогательные программы». 16 шрифт в теории так же можно менять, но я этого не делаю, потому что мне это неинтересно, потому что три готовых набора 16 шрифтов в **Zapilyator** имеется. Таким образом, Вам не надо уметь раскидывать что-либо по страничкам памяти ZX Spectrum, не надо уметь подгружать, инициализировать и выводить на экран и сжимать. Всё это делается автоматически. Ваше время высвобождается под рисование, написание музыки или если Вы и этого делать не умеете, тогда просто высвобождается время на жизнь. Ведь всё можно украсть и переправить под свои нужды, а зачем что-то создавать, если можно просто украсть?

# Старт программы.

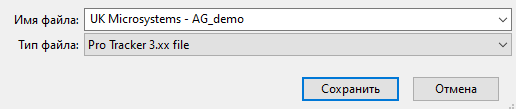
Осуществляется файлом **win-server-start.exe**, работоспособность проверена на **Windows 10,11**. По идее на 98, XP и выше он должен работать, но я не проверял, т.к. у меня нет таких версий **Windows**. Интернет для запуска программы и сборки демонстрации **не нужен**! Он нужен только для кражи картинок и гифок, их последующей обработки. По идее все вспомогательные программы можно заранее скачать, установить и работать без Интернета «по взрослому» на результат. Допустим нам отключили Интернет, делать в жизни совсем скучно, и мы все сидим, тупим, делаем демонстрацию, а когда наши придут и включат Интернет, мы выложим свои работы делавшиеся в момент апокалипсиса и кайфанем от диких комментариев. Этот редактор надо держать на компьютере, на случай отключения Интернета и всеобщего безделья по этому случаю. Поэтому давайте начнем внимательно изучать программу, следуем по меню сверху вниз.

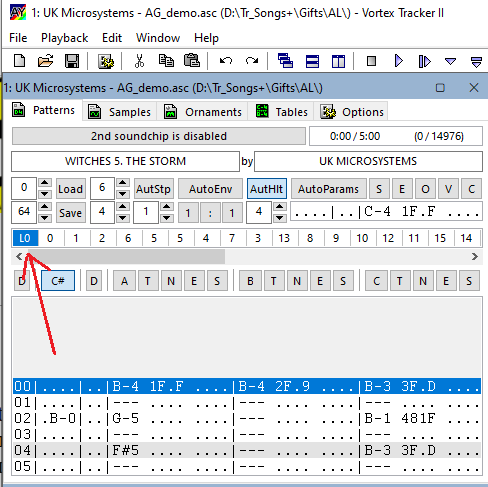
# **Меню: PT2/PT3 music file**

В **Zapilyator 3** можно вставить одну мелодию или несколько единым .zip паком. Это меню сделано для установки одной мелодии. Причём в это меню можно вставить очень длинную мелодию, например мелодии таких музыкантов, как Pator, Kuvo, Push и прочих, у кого огромные мелодии до 15 кб. Если мы работаем с этим меню и в демонстрации одна мелодия, тогда не надо ничего вставлять в меню для нескольких мелодий, и наоборот, программа будет ругаться по этому поводу. Формат мелодий только .pt3, причём желательно предварительно конвертировать их в программе **Vortex Tracker.**

<https://ay.strangled.net/vortex.htm>

Я это делаю всегда на всякий случай, чтобы впоследствии не ловить глюки и «зависоны» от кривых плееров переделанных вручную нерадивыми программистами, типа introspec. А ещё не всегда спектрумисты корректно выдирают мелодию, иногда автор мелодии или программист вручную изменяет плеер, программа Vortex гарантировано подставляет тот плеер, который точно будет работать и не глючить в Zapilyator. Всё что можно конвертировать в **Vortex Tracker** в формат.pt3 можно использовать в Zapilyator по умолчанию. То есть форматы: st, asc, sq, pt 1,2,3 – всё это можно загружать и сохранять как .pt3 в **Vortex Tracker**, а это значит, что мелодия будет гарантировано играть в Вашей демонстрации. Естественно кроме мелодий в форматах: 2AY, Turbo Sound, FM, SAA, Moon Sound, SAA. Atari, YM, и формат С.Бульбы - .AY. Для конвертации мелодии, её надо сначала загрузить в **Vortex Tracker**, а затем сохранить, как название мелодии, при этом не забываем установить в меню сохранения - формат файла .pt3.

  
меню сохранения мелодии, я выставил .PT3, вместо .ASC.

При просмотре мелодии обязательно обратите своё внимание, чтобы в ней был знак зацикливания, в любом, только не последнем паттерне должно стоять L0. Музыканты обычно ставят в первый или близком к началу паттерне (куплете), хотя в редком случае не ставят или ставят в середину, по их слуху. А нам главное чтобы такой знак просто стоял, и если его нет, то можно самим установить, например в 1 паттерне (куплете) мелодии кнопкой «L».  
  


Стрелкой показано меню в котором кнопкой «L» устанавливается на какой куплет зацикливается мелодия, то есть после того как вся мелодия проиграет, куда она будет возвращаться для нового проигрывания. Если там не будет стоять L, то музыка проиграет один раз и в демонстрации будет тишина. Вот так довольно подробно описано лишь одно меню **Zapilyator**. Почему так подробно? Потому что при любом шаге не в ту сторону, демонстрация будет виснуть. Когда проблема с мелодией, как правило, поломка сопровождается гудением, или же быстрым сбросом. Кстати с сайта-хранилища zxart.ee стандартные AY мелодии в формате .pt3 будут гарантировано играть и «бубны» с конвертацией в **Vortex Tracker** не нужны. А вот AY музыку с сайта Сергея Бульбы, лучше прогонять через **Vortex Tracker**, у него среди 36000 мелодий встречаются некорректно играющие. Но последнее время он с этим активно борется. Кстати Бульба С. всегда на связи и помогает спектрумистам с проблемами. Но тут уж кто как хочет, есть время рисковать, психовать, ничего не прогоняйте - ловите глюки. Кстати, если Вы музыкант, то можно мелодию самому написать и сделать свой сборник мелодий, это как по кайфу конечно. Я всё это к тому пишу, что нужно все нюансы соблюдать, чтобы демонстрация собралась и работала. Далее буду описывать всё так же подробно, чтобы у Вас заработало. Я люблю смотреть демонстрацию, и очень хочу, чтобы Вы после моей инструкции смогли её написать, а не просто скачать книжку и положить в папку ZX Spectrum на долгие века. Поймите, работа в ассемблере, это все равно что есть пищу ложкой ртом на кухне, а в **Zapilyator** это через другое место без ложки, не рукой и не на кухне. Тут мы за счёт того что ассемблером не пользуемся, ничего не считаем, мозгами не думаем - всё делаем через заднее место! Мы как бы пытаемся добраться до сердца ZX Spectrum-изма через то место где очень темно, страшно и пахнет плохо. Но мы же не хотим учить ассемблер, тратить своё время впустую на глупую математику, а раз не хотим думать, за нас будет думать компьютер Илона Маска и Активатор в его лице, но все ингредиенты ему надо всё равно выкладывать на блюдечке с голубой каемочкой, так как хочет только он. У нас нет шанса на ошибку. Фактически как в сказке «каша из топора.» Искусственный интеллект Активатора конечно есть, но если Вы не разовьете свой, то ничего не получится. Активатор кашу сварит, только ему нужна ключевая вода цветом белый и черный, крупа только гречневая, масло сливочное и соль нужного помола.

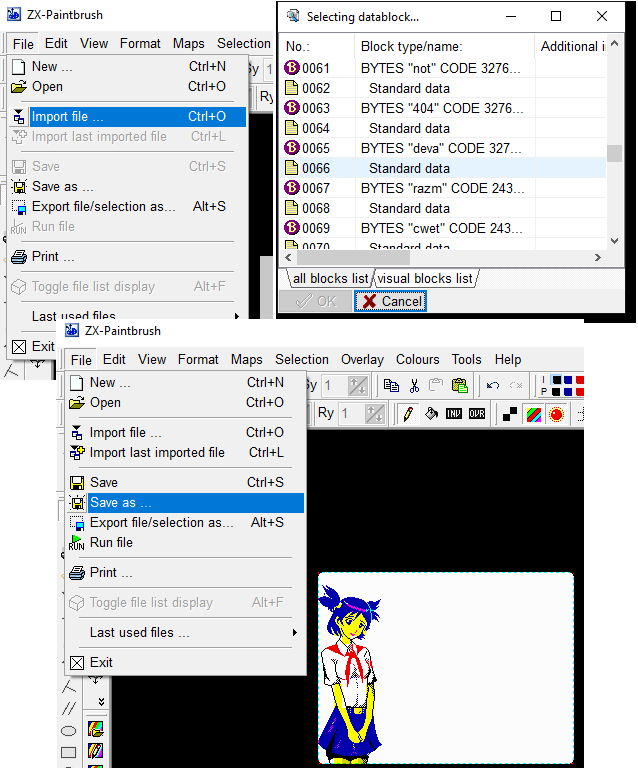
# Меню: **Splash delay**

***(****Количество куплетов музыки проигрываемых при демонстрации заставки регулируется под меню вставки картинки заставки Splash file*.)

В этом меню можно менять до 5 первых куплетов музыки. То есть играет, к примеру музыка, и заставку показывает долго, 5 куплет прошёл, началось демонстрация. Можно 1,2,3,4 куплета, музыка началась, первый куплет проиграл, и вдруг начинается демонстрация. Это подбирается по вкусу, в зависимости от целей и задачи демонстрации и самой мелодии. Этим как бы достигается некий первоначальный ритм трашмо. Мол переключение в такт музыки и тому подобное баловство. Кстати, Вы можете из музыки автора удалять любые куплеты, как Вам по вкусу. Например если Вам не нравится начало мелодии, слишком затянутое. Это делается просто правой кнопкой мыши на куплете и удалить. Если что инструкция к Vortex Tracker приложена на сайте Сергея Бульбы.

# Меню: Splash file, Background

*(заставка, внутренняя картинка)*

**Splash file -** это меню загрузки заставки в демонстрации. Заставка нужна в формате ZX Spectrum картинки – формата .scr. К сожалению не все .scr одинаково полезны, например с сайта zxart.ee прекрасные .scr, и они сразу будут работать. А с программы ESN из под образа TRD сохраненные как хоббета из родной ART Studio - нет. Я не разбираюсь в чём дело. И почему файл сохраняемый в хоббету как .scr, не работает при том что длинны 6912 одинаковые. Есть один способ, и он точно рабочий. С сайта zxart.ee можете сразу вставлять, если Вам редакция картинки не нужна, с BMP2SCR конвертора тоже сразу встанет, а вот с ART Studio способ немного затяжной, но рабочий. Всё что в этом абзаце написано, относиться и к внутренней картинке **Background** меню. Итак опишу способ правильного создания картинки формата .scr из TRD образа, берём немецкую старую программу **ZX-Paintbrush** для **Windows** с сайта [**https://vtrd.in/pcutilz/ZX-PAINT1.zip**](https://vtrd.in/pcutilz/ZX-PAINT1.zip) **Устанавливаем её в Windows.**

**Import file** открыть образ **TRD с ART Studio** и картинками подготовленными в демонстрацию. **Save AS** – сохранить в формате **.scr**. На картинке - примере, всё показал, вот так подготовленные картинки формата **.scr**, будут работать точно. К сожалению, программа ZX PAINT для WINDOWS глючная или не корректно работает в Windows 10,11, я делаю по одной картинке и всякий раз её перезапускаю, возможно из-за проблем совместимости с Windows 10,11. Я работаю в ней так: конвертировал 1 картинку, вышел, снова зашёл выбрал другую, и так все картинки поочередно. Иначе она наслаивает старую картинку на новую. При переводе начинает выводить глючные картинки с наслоением.  
 Иногда нужно многое менять в **Background** (внутренней картинке демонстрации), например, когда пытаешься попасть в цвет знакомест или же просто когда что-то рисуешь и проверяешь как оно вышло в демо. Тогда я не отключаю эмулятор **Unreal** с загруженным редактором **ART Studio**, сразу в нём редактирую картинку и сразу её конвертирую **ZX Paint**, затем сразу в **Zapilyator**, который так же не выключаю, проверяю (когда там все уже подгружено, так удобнее). И так пока не добьюсь результата, например, надо попасть в раскраску знакомест анимацией. Это сильно экономит моё время и ускоряет работу над демонстрацией, всё-таки не как слепой котёнок, а постоянно имеешь возможность запустить свою демонстрацию со всеми спрайтами, эквалайзерами и отточить графику экрана до совершенства. Можно так же редактировать в стандартном **PC Paint** из под **Windows**, сохранять и перегонять в **BMP2SCR**, и так же сразу закладывать в ячейку **Zapilyator** для проверки. В этом плане **Zapilyator** даёт истинный кайф в работе. Ты не мучаешься, а получаешь удовольствие от создания демонстрации. Я даже пиво в такие моменты «шлифовки» наливаю. Пью, и в кайфе подгоняю графику под анимацию. Выбор инструментов зависит от ситуации. Мне удобнее мелкие детали и раскраску делать в ART Studio, а крупные объекты стыковать в Paint сразу из под Windows. Таким образом, я сейчас подробно объяснил про то, что необходимо строго соблюдать формат **.scr** для заставки и внутренней картинки демонстрации – **Background,** и в последствии для цветных фаз спрайтов, которые так же делаются для **Zapilyator** через эти программы. Почему это важно? Потому что это тот глюк, который невиден и непонятен, но из-за него не работает собранная демонстрация – она просто виснет. Казалось бы, всё сделал правильно, а она не работает, оказывается что .**scr** формат внутри может содержать разный код, который **Zapilyator** не понимает. Я описал способ того как готовить картинку чтобы **Zapilyator** её точно понял. Ещё раз напишу, что с сайта мороза99 – zxart.ee все картинки **Zapilyator** понимает сразу, у него исторически верный .scr формат, однако после Вашей редакции в ART Studio с образа TRD **Zapilyator** уже не понимает, но мой способ рабочий. Хотя - я не исключаю, что есть множество других программ и способов. Есть люди, которые умеют прекрасно рисовать в BGE или немецкой Windows Paint для ZX, но я не умею.Мне некогда заниматься изучением корня проблемы и кучи других программ. Я пишу в квартал по 10 демонстраций, занимаюсь работой на сайте, форуме и занимаюсь спортом. Поэтому я делаю так, как быстро и точно работает. Забыл написать, что на **Zapilyator** я работаю более 7 лет, и что мной создано более 300 программ в нём, включая несколько крупных проектов – мега демонстраций.  
<https://zxaaa.net/AAABAND2.html>

Прислушайтесь к моим советам, чтобы не терять своё время, которое я потерял изучая методику запила методом «все научного тыка». Однако, если у Вас есть время, тогда находите новые способы запиливания демонстраций, в новых, неведомых мне подсобных программах. С удовольствием почитаю и возьму на вооружение в своей работе.

# Подготовка графики в редакторе BMP2SCR:

Графика, как же её сделать? Не рисовать же самому в конце то концов! Ни в коем случае, мы же все ничего делать не умеем. Можно что-то нарисовать в программе Art Studio, можно самому перевести с рисунка из сети Интернет или нарисовать с помощью Paint для Windows, и конвертировать в ZX Spectrum в BMP2SCR, а можно украсть и переделать под себя готовую с сайта художников: <http://zxart.ee/> Для новичка этот способ предпочтительней. 90% графики с этого сайта кроме авторов и десяти матёрых Спектрумистов поросших пиксельным мхом - никто не видел!

Подводных камней на этом участке работы на самом деле не так много, это обычный формат ZX Speccy картинки, размером 6912. Правда многие программы приделывают к картинке заголовки, и она меняется в размере, строго следите за её размером, он должен быть равен 6912 байт. Размер имеет значение, иначе **Zapilyator** не проглотит картинку, а сделает её сдвинутой и кривой, поэтому внимательно следите за её правильным форматом: 6912 .С. Этого можно добиться многими способами я Вам расскажу про самый распространенный способ с помощью программы BMP2SCR. Без этой программы на ZX Spectrum работать нельзя. Это основная программа Zapilyator-щика и её надо знать на зубок. Изучается только тыком в процессе жизни, исследуя все её функции на практике. Объяснить невозможно, только пробовать. Инструкции к ней нет. Каждая картинка сделанная в ней, это как скульптура – она индивидуальна. <http://vtrdos.ru/pcutilz/BMP2SCR.ZIP>

Любое сохранение в этой программе как **Save** сразу даст нужный результат в виде .scr. Что касается кражи картинок (конверсии) не забывайте что для цветной картинки необходимо помимо параметра цвет (Color Solid) в программе BMP2SCR, переходить в параметр регулирования цветов (Conversion Setting) и там отмечать в меню Bright Method подменю **Intellegent**, далее заработает двигалка Bright, которой вы и налаживаете бригхт, такой как у автора. Помните, воруя графику, не надо её портить, её нужно улучшить, но не делать хуже, иначе Fuck «жыво творящий» от автора не минуем. А если Вы сделаете картинку художника лучше, он скажет, вот чёрт, а он прав, как же я этого не заметил! Обычно параметр яркости (**bright**) держится в цифрах от 20-40. Остальное сравнивайте визуально. Более подробную методику работы с программой **BMP2SCR** я давать не буду, ибо это искусство и в этом суть написания демонстрации. Необходимо самому постичь эту программу, прочувствовать и выйти на свой новый уровень. Я до сих пор постигаю возможности этой программы, не смотря на то, что являюсь самым главным и узнаваемым «крадуном» графики в мире и работал с этой программой около 20 лет. За это время я её научился чувствовать как свою собаку.

# Меню: border, paper, ink, bright.

**Border –** бордер (рамка вокруг картинки) вокруг заставки. Цифры соответствуют цифрам цветов на клавиатуре ZX Spectrum. Это меню находится под вкладкой загрузки заставки.

**Animation Border** – рамка вокруг картинки 256 на 192 точек внутри демонстрации.  
*Не забываем!* На ZX Spectrumе рамка бордера всегда определяется не яркими цветами. Потому что яркости брайта на border у нас нет и не будет – мы не заслужили, дедушка Клайв Синкляр хотел ведь подешевле. Клайв, у нас бордер не яркий получился, что делать? - А зачем?

**INK –** цвет пикселя внутри картинки, используем только тогда, когда Вы не используете в демонстрации внутреннюю картинку backgraund, на которой уже определен цвет самим рисунком, за исключением случаев применения Вами цветных фаз спрайтов экранов спектрума в виде .gif. (как в примере, который я описываю в этой книге в демо котята).

**PAPER -** цвет внутри картинки, используем только тогда, когда Вы не используете в демонстрации внутреннюю картинку backgraund, на которой уже определен цвет самим рисунком, за исключением случаев применения Вами цветных фаз спрайтов экранов спектрума в виде .gif. (как в примере, который я описываю в этой книге в демо котята).

**BRIGHT -** используем галкой только тогда, когда Вы не используете в демонстрации внутреннюю картинку backgraund, на которой уже определен цвет яркости самим рисунком, за исключением случаев применения Вами цветных фаз спрайтов экранов спектрума в виде .gif. (как в примере, который я описываю в этой книге в демо котята).  
  
**ВНИМАНИЕ!** При использовании в демо в одной из банок для спрайтов цветных картинок .scr, которые Вы запиливаете как спрайты, описано ниже. Тогда обязательно выставляем параметры paper, inc, bright в меню под **BACKGRAUND**, чтобы они соответствовали цвету фона картинки и спрайта, об этом я скажу несколько раз в этой книге. Без выставления этих параметров при выводе цветного спрайта любым способом, картинка бакграунд (внутренний экран) будет затираться цветом фона цветных спрайтов.  
Вот алгоритм **Zapilyator**, который описал Илон Маск, изучив код программы:

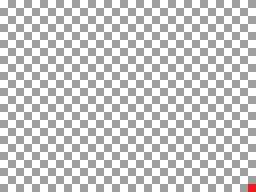
Чтобы initialColor = 0x38, установите:main\_ink = 0 (чернила: черный)main\_paper = 7 (бумага: белый)main\_bright = 0 (яркость: обычная)Итог: main\_color = 0 + 7\*8 + 0\*64 = 56 = 0x38Если в SCR файле атрибуты равны #38, они не попадут в diff (так как совпадают с initialColor), и #38 не будет выводиться на весь экран.

# **Меню эквалайзеров:**

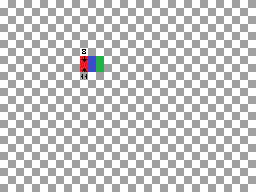
**Goba Left -** справа налево длинный **Goba Right -** слева направо длинный  **Goba UP -** маленький вверх три лепесточка (3 в длину на 2 в высоту знакоместа), 1 знакоместо 8 на 8 точек.

**Goba Down** - маленькийвниз три лепесточка (3 в длину на 2 в высоту знакоместа), 1 знакоместо 8 на 8 точек.  
 Когда Вы будете первый раз делать один из анализаторов (эквалайзеров), попробуйте сначала сделайте его методом «тыка», попытайтесь понять суть, прочтите ещё 100 раз нижний абзац про то, как на ZX Spectrum организована сетка экрана. Это, между прочим суть всего живого на спеки. И можно сказать если Вы это поймете, то после этого Вы уже почти программист, ну не то чтобы, но в грудь кулаком можете себя избивать по пьяни на тусовках с тещей.

**Analyzer –** анализаторы. Весь экран делите в уме сеткой по 8 точек или не в уме, а на Яву, как я. Можете на этой сетке делать расчёты, и только потом уже в **Zapilyator** писать значения цифр по X и Y.

****сетка экрана ZX Spectrum 256 на 192 точки. 32 на 24 квадратика 8 по 8 точек. **!АЗЫ ZAPILYATORСТВА! – знакоместа это суть всего живого, без понимания этого, мы далее не двинемся…**

1 линия по горизонту это цифра **ОДИН** в любом меню выбора **Zapilyator**. 1 линия 8 точек в высоту по вертикали это **ОДИН** в меню Zapilyator. (имеется в виду цифра по X,Y у эквалайзеров и скроллов) Итого 24 линии по вертикали и 32 линии по горизонтали. Давайте ещё раз заново, как для блондинок: экран ZX Spectrum содержит 256 на 192 точки. 256 делим на 8 (одно знакоместо 8 на 8 точек) = 32 линии (32 квадратика 8 на 8 точек). 192 делим на 8 = 24 линии. Эта формула для всего, что существует на ZX Spectrum. Это как ДНК у человека, его не надо менять, иначе будет даун. Запомните мои слова, а лучше прочтите 10 раз абзац и если Вы его не поняли, давайте ещё раз. Всё что делается в Zapilyator мы ставим четко по границам 8 на 8 точек (знакоместа). И эквалайзеры и спрайты и скроллы, везде знакоместо к знакоместу. Как в лего конструкторе, нельзя воткнуть иначе будет криво. Только там криво, а у нас повиснет. 8 на 8 точек, квадратик к квадратику, не пересекаясь границами, нельзя 7 точек,6,5,4,3,2,1. Всё соединяем границами знакомест и в горизонталь и в вертикаль. Иначе будут глюки в виде мерцающих спрайтов, сломанной графики, разбитых квадратов или просто повиснет с гудением мелодии. Всё поняли? Да ни хрена Вы не поняли! Эквалайзер сколько занимает знакомест? Например, берём **Goba Up** два знакоместа в высоту и три в длину. Это значит 16 на 24 точки, а это 2 квадратика вверх и 3 квадратика 8 на 8 точек по горизонтали. Что это значит? Жду любой умный ответ от спектрумистов…   
  
 Ладно, смотрите, вся длинна ZX Spectrum экрана по горизонтали это 32 квадратика (256 точек). Это значит, что по горизонтали по **X** мы не в 31 и 32 квадратик не можем поставить этот анализатор, потому что последний лепесток эквалайзера вылезет за пределы экрана. А где он вылезет? А он вылезет аккурат с другой стороны экрана и заглючит Вашу работу (потому что он влезет в другую экранную область, где уже что-то есть нарисованное) Куда мы его ставим? Крайняя точка 30. Тогда 31, 32 это последние два знакоместа в горизонте. Понятно теперь о чём я? Давайте, ещё раз прочтите этот абзац, потому что это основополагающий абзац для всего: спрайтов, скроллов. Чтобы потом Вы уже на опыте были. И главное с пониманием, а не методом тыка и психоза. Давайте пример:



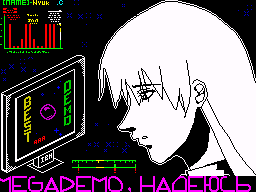
Нам надо поставить эквалайзер сюда. Тогда сверху по оси Y это 9 цифра, а по горизонтальной оси X это 11 квадратик. То есть эквалайзер Goba Up 11, 9. Почему 9, а не 8? Потому что нижний левый угол эквалайзера, это точка отсчета. Поняли? Кстати, не смотря на свой опыт, эти цифры до сих пор по этой сетке высчитываю руками, так же как на примере. И это гораздо быстрее метода перебора номеров знакомест. Но если на одно знакоместо ошиблись не беда, в следующий компиляционный запил переставьте цифру на 1 значение туда, куда Вам надо. И всё. Показания уже выверенного анализатора или скролла, когда это наконец то случается, я записываю в реадми файл, который сопровождает проект. Это удобно для последующих переделок, чтобы не вспоминать всякий раз, что там было. Можно и в бумажный блокнот записывать, как делали в былые времена наши деды. Для ловли эффекта ностальгии, лучше всё записывать в тетрадку. Большие проекты компилируются долго, секунд 10, это зависит от скорости ноутбука на котором Вы работаете. Одно из хороших при имуществ запилятора, это быстрая проверка чего либо переделанного. Это как заряженный станок. Заменил кассету, посмотрел, как будет выглядеть: новый спрайт, новая картинка или новая мелодия. Можно считать знакоместа иначе, например мышью в Paint Windows растягивать границы от края картинки до объекта, получать значение в пикселях, допустим цифру 80 точек, делим на 8 получаем 10 знакомест. Это как по X-оси так и по Y-осям. В качестве **лайFuck** скажу, что я в Windows, на панель задач вынес две программы: PAINT PC, калькулятор. Чтобы быстро переходить для рисования картинок, и расчетов знакомест. Это удобно. Считать Вам всё равно придется постоянно. Потому что все компоненты демонстрации на экран мы выкладываем с расчетом этих координат по X и Y. Поэтому картинку с сеткой ZX Spectrum в формате 256 на 192 точек я тоже храню на рабочем столе, чтобы быстро выводить и считать.

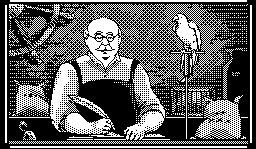
# Подготовка спрайтов неровной формы, их стыковка.

Да, можно резать, хоть лесенки, хоть лошадок, главное всё в знакоместах, то бишь можно спрайт буквой T, но знакоместами.

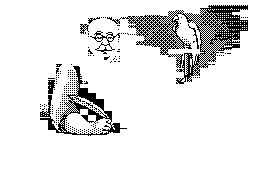


Вот пример на картинке, представьте, что цветное-кривое это разные спрайты внутри каждого что-то есть. Я могу их стыковать как угодно лишь бы по знакоместам. Но это очень трудно, потому что придётся скорее всего ручками такое делать, в каждой фазе стыковать. Я такое делал, но когда заморачивался сильно. Это высшая математика Zapilyator-ства. Можно месяц одно демонстрацию делать. Как пример, волосы девушки в последней части демонстрации Best Annimation MD, фактически руками сделал маску, знакоместами. Скачайте, прокрутите, всё это мегадемонстрацию, оно сделано в 1 версии

Zapilyator в 3 банках памяти.   
<https://zxaaa.net/view_demo.php?id=8466>   
  


Причём, посмотрите эту часть, это на самом деле просто высший пилотаж. А она написана в Zapilyator v.1. И так то это моя где то 10 по счету работа.  
  
Но было и трудней случай, когда я делал по маске деда в **KEY DEMO**.  


Маска выводила только движение руки и попугая из большого спрайта, а саму картинку отдельно делала, для экономии памяти. Это высший пилотаж. Смотрите последнюю часть нереально трудного в изготовлении **KEY Demo**:  
<https://zxaaa.net/view_demo.php?id=8814>

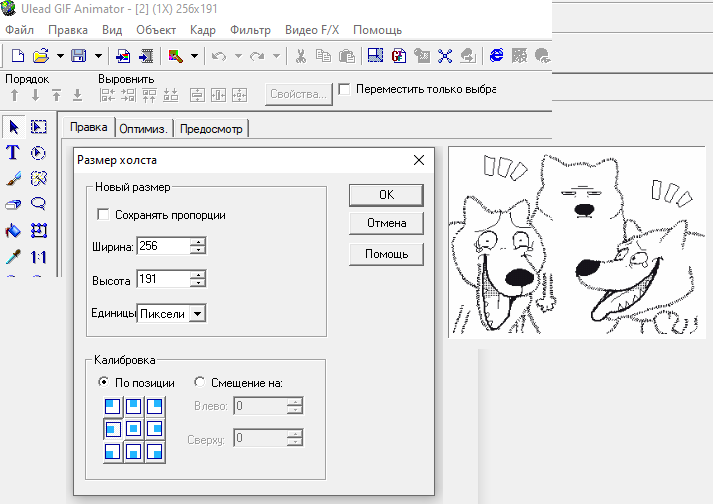
  
 Вот так выглядел спрайт после обработки и выноса за свой фон. (её фон выше на фото)

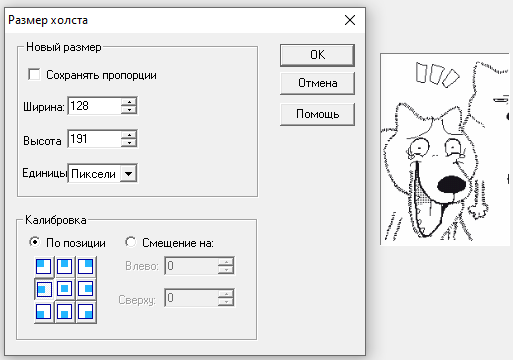
Это делалось в программе Какоса, которая позволяла отделять фон спрайта от движения в нём в рамках знакомест. Фон отдельный я выводил как бакграунд, то есть внутренняя картинка демонстрации, а выделенный программой движемый спрайт отдельно. К сожалению, я уже не помню, в чём это делалось, и Какос забыл, я его спрашивал что за программа, но если вспомню, напишу. По-моему где-то в журнале «За Рулем» есть описание процесса в моей статье про создание KEY Demo. Поищу, как-нибудь. Самому интересно как я это сделал. Человек ли я?

# **Основы работы в Unlead Gif Annimator 5.0.**

Какой либо иной версии **Gif Annimator** мне не удалось найти. К сожалению, программа старая и почти перестала работать в Windows 11, и я переустановил себе Windows 10, где она, хоть и глючно (проблемы к доступу к каталогу, иногда программа вываливается в Васик Виндовса), но всё же работает. Потому что без неё или подобной программы в Zapilyator делать нечего. Можно и в Photoshop и кучи других программ, но я научился в Gif Annimator 5.0 Crack про него и расскажу.

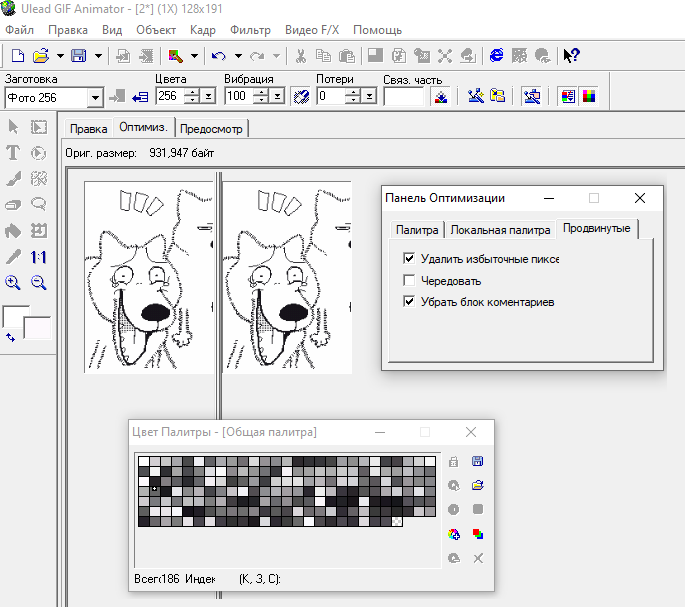
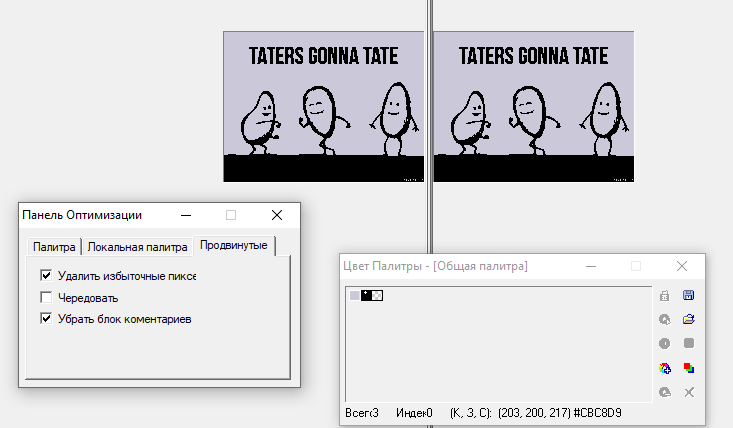
# **Нарезка, обрезка спрайта.**

Меню | ПРАВКА | РАЗМЕР ХОЛСТА  
В нём осуществляется нарезка, обрезка.  


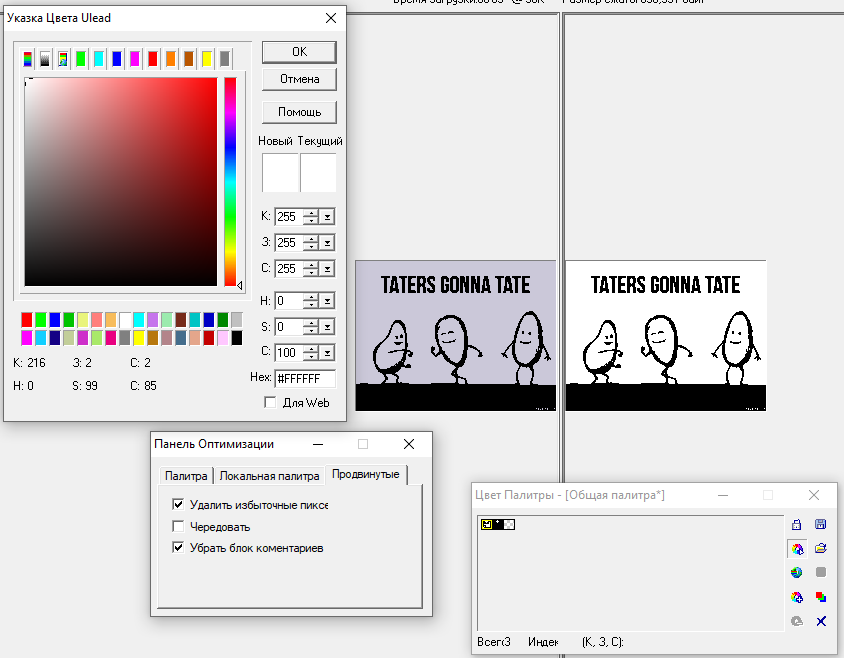
Видите 9 квадратиков с синим цветом, как джойстик? Это меню нарезки. Чтобы понять как, представьте что вы в комнате квадратной, а с другой стороны мужик с ножом, который хочет Вас убить, чтоб его отрезать, и он Вас не убил надо прижаться к противоположной стенке. Запомните это правило!!! Чтобы отрезать Волка справа, мы ставим курсор как на картинке слева. И раз мы режем справа, то значит это по горизонтали расчет, значит где 256 мы пишем цифру, подбирая её, например так: 256 разделить на два будет 125+3=128. Так мы отрежем пол экрана с правой собакой. Режем:  


Правило страшного мужика с ножом действует в любую сторону. Я накидываю текстом примеры. Нам надо отрезать сверху, значит курсор мы ставим вниз и режем по ВЫСОТЕ горизонтали. Надо отрезать в середине, ставим в середину и все что с краев воображаемой фигуры, которую мы очерчиваем по X и Y своими цифрами координат отрежется. Помните, что все отрезания должны быть кратны 8 (то есть знакоместу) И по высоте и по длине. После того как Вы сделали спрайт, проверьте, делятся ли его стороны на 8, если нет, значит надо переделать, тут Вы лучше сами пробуйте, объяснять это как то иначе сложно. В дизайне и графике, многое надо пробовать и нарабатывать свой навык. Это не программирование, где можно тупа переписать с тетрадки, в графике и музыке надо творить!

# **Подготовка стандартного спрайта.**

Спрайт на выходе должен быть только двух цветным. Черный 0,0,0 или белый 255,255,255. Никакие другие варианты не допускаются, **Zapilyator** будет глючно выводить иное. Так же не забываем цветные спрайты сначала приводить к двух цветному виду, а затем, если там будет конечно с чем работать, и оно ещё смотрится, уменьшать кратно сторонам. (это и для чёрно белых спрайтов и простых картинок подходит) Не забываем, кратность сторон желательно делать 1024, 512, 256. Если, допустим Ваш первоначально украденный спрайт по одной из сторон 1200 точек, то логичней его сделать 1024, ещё до того момента как начали сжимать, потому что потом в программах, где мы будем сжимать, проще и лучше жмется кратное нашим родным 256 точкам по горизонтали. Потому как 1024 это 512, а 512 это 256, если давить по 50%. Тоже самое относиться и к вертикалям, но как правило вертикаль соответствует горизонтали. Если жать с 1200, то спрайт при сжатии сильно исказит само изображение. Эти правила относится и к простым картинкам, которые мы воруем под конверсию и дальнейшую обработку. В общем, первоначальный размер сжатия надо соблюдать кратно сторонам 256 на 192 для лучшего результата.  
  
  
 Вот пример того как спрайты украденные в Интернет выглядят при вспарывании кишков. Нам кажется, что они белые и чёрные, а на самом деле нет. Видишь зайца? А он есть! Можно ли руками это сделать? Можно, но это ведь долго каждый такой квадратик цвета нажимать и ставить (подменять) нужные цвета: черный и белый вместо оттенков. А бывают пограничные между белым и черным цифры, и что туда ставить трудно понять что ставить, а если ошибиться рисунок кардинально изменится, например у собаки на лице появится прыщ или линия лишняя. Поэтому есть программы высоко-интеллектуальные, например **unoptimizeGIF** (реоптимизатор) Какоса. (она лежит в общем паке с инструкцией внутри) В неё вы загружаете спрайт и выгружаете его в таком виде:  
  
  
Как видим всего два цвета. (фиолетовый в примере не типичен, просто оригинал такой взял, у Вас будет там черный, если возьмете темные оттенки) Фиолетовый или любой другой, нам не подходящий можно сменить на белый. Нажмите мышью на фиолетовый квадрат ЦВЕТ ПАЛИТРЫ. Появится цветовая палитра, выберите белый взамен фиолетовому чёрный. **unoptimizeGIF** (реоптимизатор) сам определяет, что делать чёрным, а что белым. Он всё приводит к одинаковому значению, по своему алгоритму, но бывает и он не справляется, если спрайт слишком замыленный размытием цветов. Программа не способна понять что человек имел в виду, поэтому она может сделать чёрным, то что должно быть белым. Это очень плохо и такие спрайты, я обычно разбиваю на фазы, предварительно, естественно сделав их кратно 256, а лучше не более 512 размера, и подсовываю по одной фазе в BPM2SCR. А там уже по одной фазе конвертирую, с нужной мне на мой глаз, градацией перевода, чтобы как можно больше деталей объекта сохранилось. В этой книге есть описание этого процесса в главе **подготовка графики в редакторе BMP2SCR** и **Уменьшение размера .gif кратно сторонам.** Как уже видно, многое постоянно пересекается в книге, это общие принципы работы с графикой при конверсии, переводах и обработке краденного контента. Это и знакоместа 8 на 8 точек, это и размеры 256, 512 на 384, 1024… Всё это основные принципы работы без соблюдения которых, демонстрация будет работать, но криво, косо, с секущемся лучом и искаженной пропорцией объектов. Такие демонстрации лишь оскорбляют чувства верующих в демо на Zapilyator. Не забываем, какие шедевры можно делать, лишь приложив усилия и соблюдая основные принципы. Вам я всё разжёвываю, как детям малым, а сам до всего доходил методом проб и ошибок.

# Удаление избыточных пикселей, работа с цветом.

  
  
Чтобы зайти в это меню, надо нажать сначала: **ОПТИМИЗ**. На кромке рамки Gif Annimator. Затем откроется меню **ВИД**, а там **ПАНЕЛЬ ОПТИМИЗАЦИИ** и **ПАЛИТРА ЦВЕТА**. После того как спрайт сделали, Всегда необходимо убирать галку в меню **УДАЛИТЬ ИЗБЫТОЧНЫЕ ПИКСЕЛИ**. Я не знаю, что это такое, но без этого спрайт может глючить, хотя внешне он идеален и состоит из двух цветов.   
  
Правило это запомните для всех спрайтов в Zapilyator, которые Вы подготовили где либо. Их все надо прогонять через Gif Annimator 5.0 Crack и убирать эти избыточные пиксели, (меню ВИД, а там **ПАНЕЛЬ ОПТИМИЗАЦИИ**) даже когда Вы их не видите, они есть.

Вот что мне ответил **Илон Маск** по этому вопросу:

Избыточные пиксели в GIF-анимации - это **пиксели, данные о цвете которых хранятся в файле, но которые не меняются от кадра к кадру** или не видны зрителю. Их наличие приводит к неоправданному увеличению размера файла.

**Детали:**

* **Как это работает:** Анимированный GIF состоит из последовательности кадров. Изначально, каждый кадр может хранить полную информацию обо всех пикселях изображения. Если на протяжении нескольких кадров фон остается неизменным, то информация об этих неизменяющихся пикселях дублируется в каждом новом кадре, создавая избыточность.
* **Проблема:** Эта избыточность увеличивает общий вес (размер) GIF-файла, что замедляет его загрузку и потребляет больше пропускной способности сети.
* **Оптимизация:** Современные программы для создания и оптимизации GIF-анимаций (например, Photoshop, Gifsicle, онлайн-сервисы) используют специальные методы для удаления этих избыточных пикселей. Вместо сохранения полной картинки в каждом кадре, они сохраняют только те пиксели, которые фактически изменились по сравнению с предыдущим кадром, используя прозрачность, чтобы показать неизменившиеся части предыдущего кадра.
* **Результат:** Оптимизация за счет удаления избыточных пикселей позволяет значительно уменьшить размер файла без потери качества изображения или плавности анимации.

**ВСЕГДА НАДО убирать галку в меню УДАЛИТЬ ИЗБЫТОЧНЫЕ ПИКСЕЛИ.**

**Zapilyator** зачастую не понимает что в .gif космический мусор, а что астронавт Гагарин. Представьте Вы встретили девушку, она прекрасная, но Вы не видите её избыточные пиксели, а потом в процессе общения понимаете, что она тварь. Это и есть избыточные пиксели, их обязательно надо убирать в Gif Annimator или реоптимизаторе Какоса, или ещё проще простого сделать это в бесплатной OnLine программе: <https://ezgif.com/webp-to-gif>   
 В разделе оптимизация есть функция монохромного рисунка и там то же уберутся избыточные пиксели. Но я по старинке всегда пользуюсь Gif Annimator. В Photoshop тоже есть функция «убрать избыточные пиксели», но я в нём ни бум-бум. unoptimizeGIF (реоптимизатор) Какоса, который я приложил в паке программ тоже прекрасно удаляет избыточные пиксели.

# Подмена цвета одного на другой в картинках:

Ещё один **ЛайFuck** от **ААА.** В программе GIF Annimator 5.0 Crack, ровно таким же способом можно менять в краденных под конверсию картинках один цвет на другой и сохранять результат как .gif картинку, потом уже её переводить в .png для дальнейшей редакции, например в Paint. Но это отступление для работы с графикой. Например, я конвертирую графику HiRes c Commodore 64, по своему вкусу меняю один некрасивый, тусклый, унылый цвет коммодора на другой яркий жизнерадостный - спековский. Цифры цветов беру пипеткой из шаблона ZX Spectrum цветов, который сам сделал в ART Studio, этот шаблон я вместе с редактируемой мной картинкой делаю в PAINT на PC. Вначале, этот шаблон помещаю на экран около картинки сбоку например. Это для чтобы цвета сразу присутствовали в общей пипеточной палитре в гиф аниматоре:  
  


В принципе у каждого цвета ZX Spectrum есть свой цифровой код, и зная его, или же прощупав пипеткой, можно менять своими руками без обращения к палитре:

**НЕ ЯРКИЕ**

белый - 159,159,159

Желтый - 159,159,0

Голубой – 0,159,159

Синий – 0,0,159

Красный – 159,0,0

Фиолетовый – 159,0,159

Зеленый -0,159,0

**ЯРКИЕ**

белый – 254,254,254

Желтый – 254,254,0

Голубой – 0,254,254

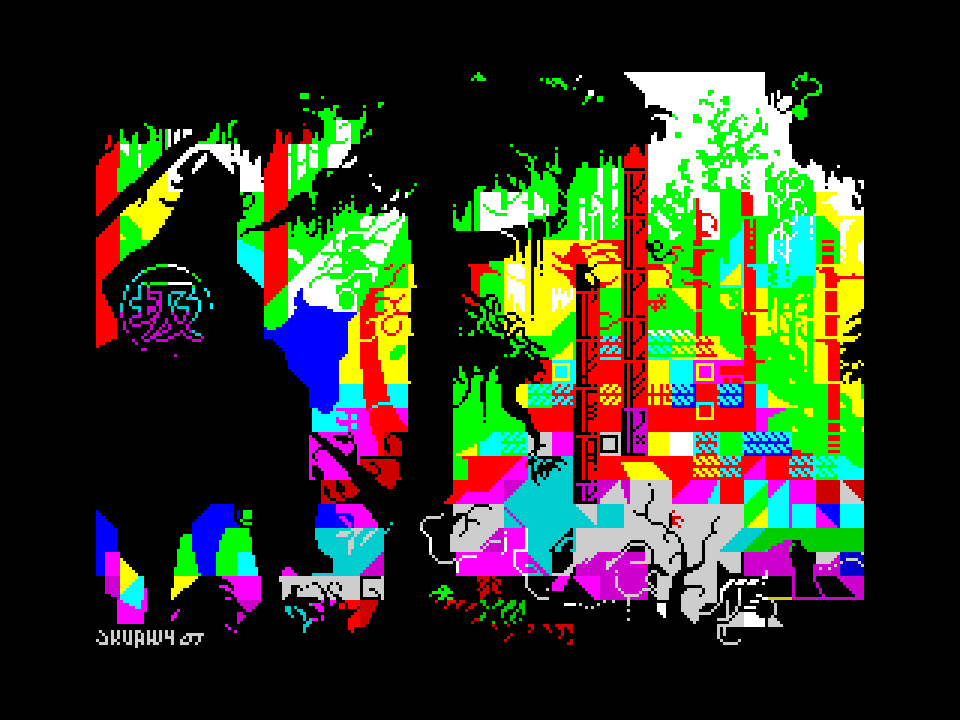
Синий – 0,0,254

Красный – 254,0,0

Фиолетовый – 254,0,254

Зеленый – 0,254,0

Почему в примере разняться цифры цвета, ведь я говорил 255,255,255 это белый. Да, но у каждого эмулятора свои оттенки, я для этого примера снял цвета с экрана **Spectaculator**. Возможно, в другом эмуляторе будет 159 и 254 цвета, ярче на 1 оттенок (160, 255), но BMP2SCR эту разницу сожрёт и не заметит, так что не переживайте. Это для общего образования. Не удивляйтесь, что где то 253, где то 254, а где то 255, это просто авторы эмуляторов как правило кодеры-дураки, им всё равно какие у ZX Spectrum правильные данные цветов. **BMP2SCR** это сожрёт, и выставит необходимый близкий, который точно загрузиться в ART Studio для дальнейшей обработки руками. Но я всегда делаю по феншуюу, то есть 0,0,0 и 255,255,255 уже на вход к Zapilyator, но это касается только gif.

Вот пример конверсии в **Gif Annimator 5.0 Crack** простой заменой или подменой цветов. Да, не удивляйтесь, эта картинка переведена на спектрум только в Gif Annimator. Там я легко поменял цвета, ничего не рисуя и не сдвигая, заменил некоторые цвета коммодора спековскими. А знакоместа спека и коммодора полностью совпадают. Что там 8 что тут. Поэтому надо просто заменить цвета и найти размер 256 на 192, у коммодора он чуть больше, надо просто по вкусу отрезать лишнее до размера 256 на 192 или просто в **BMP2SCR** подвигать готовую картинку и там определить нужный участок который оставляем как 256 на 192. Переводить картинки с коммодора 64 одно удовольствие.  
  
  
Работа со спрайтами это адский труд, программы для работы со спрайтами ужасно трудные в логическом понимании, особенно всякие там пхотошопы. Я сейчас попытался объяснить только два аспекта, это обрезка и замена цвета, а их сотни. В этом и есть труд Zapilyator-щика - это умение искусно работать в сторонних графических программах. Можно ли сделать в Photoshop? Да! Но я не умею. Я сам постигал Gif Annimator методом ненаучного «тыка». Можете где угодно работать. Но на выходе два цвета: 0,0,0 и 255,255,255. И оптимизация избыточных пикселей.

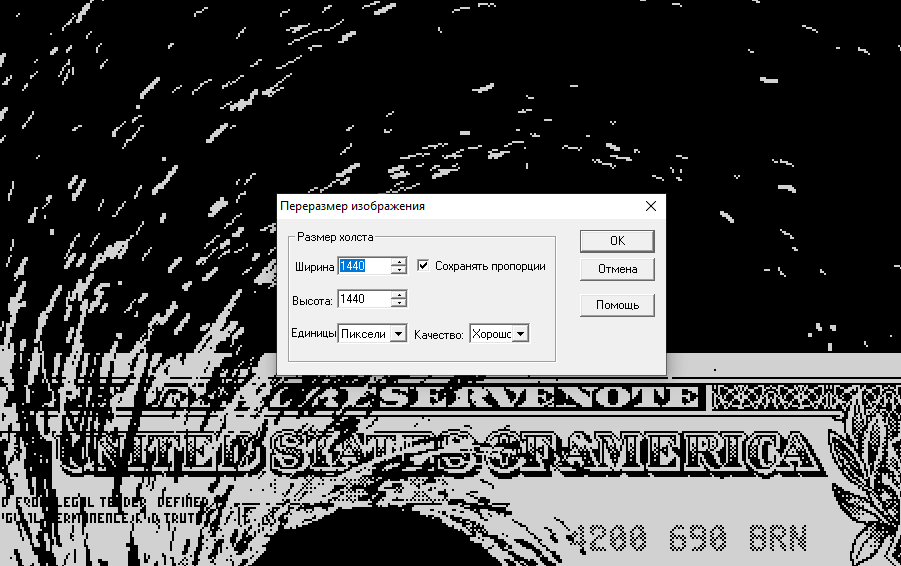
# Инверсия .gif.

Инверсия спрайта иногда необходима. Её можно сделать как на **background** картинке - внутренней картинке демонстрации, но можно и сам спрайт инвертировать. В принципе как Вам удобнее. Она делается так же заменой цветов в Gif Annimator 5.0 Crack. Грузим спрайт, там два цвета: черный и белый. Мы просто назначаем чёрному временно, допустим красный, потом меняем белый на чёрный, а красному ставим белый. Прочтите ещё раз. Или же можно просто онлайн программой сделать инверсию спрайта: <https://ezgif.com/>

Так то ещё можно все фазы скинуть по одной и каждую фазу загружать в Paint и инвертировать руками )) Кому как нравится. Можно демонстрацию делать год, а можно час. Руками я делал в начале пути, это прикольно и долго, но от таких запилов в глазах сосуды лопаются от напряжения. Особенно когда правишь фаз 200 у спрайта.

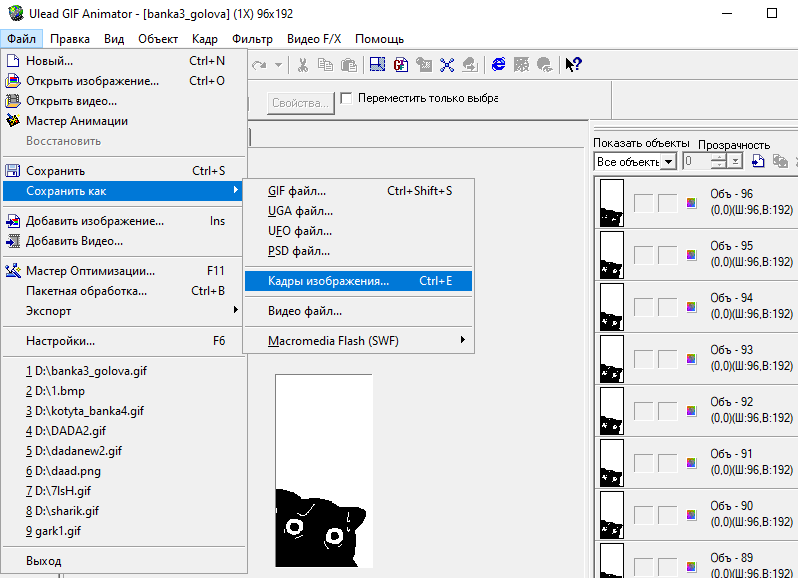
# Уменьшение размера .gif кратно сторонам.

Лучше всего и проще это делать в онлайн программе <https://ezgif.com/resize>. Картинки приведенные к размеру 512 на 384 мы жмем как 50%, остальное сами считайте. Просто 256 на 2 это 512, 512 на 2 это 1024. Все размеры картинок, в принципе однообразны и сводятся к ZX Spectrum-овскому 256 на 192. Просто делите правильно.

Но можно и в Gif Annimator в меню:  
ПРАВКА, ПЕРЕРАЗМЕТИТЬ ИЗОБРАЖЕНИЕ  
  
Доллар сейчас размером 1440 на 1440, а нам надо 256. Ставим галку сохранять пропорции и выбираем в верхнем окошке 256  


Так как квадратный .gif, то оно и стало квадратным 256 на 256. Видите, спрайт стал мутный? Эта «размытость цвета» появляется при уменьшениях .gif, как её убрать? Тут я могу сказать что в данном примере никак, но в целом если не с такого размера, а с 512 на 384 до 256 уменьшать, то можно попробовать выгружать все фазы готового мыльного спрайта, как кадры и потом ручками каждую править подбирая единую для всех чёрно-белую градацию перевода в BMP2SCR, это долго, но это лучший результат. Там кстати 3 вида серой градации в меню сверху первые три строчки. Не буду раскрывать все секреты, скажу так, я Вам азы дал, а дальше сами, ищите свой способ. **Единственный совет**: при обработке большой группы фаз спрайта, кидайте их на сохранение фаз в одну папку, под именами допустим a1, они будут a001, a002, a003 и.т.п.. И в эту же папку кидайте программу BMP2SCR, запускайте и она по умолчанию будет сохранять, подменять эти же фазы на себя, в меню выставляем, **сохранять как PNG**, а не scr, т.к., удобнее редактировать фазы спрайтов в Paint на IBM PC, а не в ART Studio. Их и видно сразу в папке Windows, и менять что-то там удобнее, например дорабатывать или обрабатывать каждую фазу, удалять глюки перевода. И потом удобнее смотреть готовый результат просмотром картинок на Windows в одной папке. При этом как мультик картинки будут показываться в нужной Вам ручной скорости. И нет потери времени на копирование в ART Studio и обратно. Всё равно, в итоге, Вы будете собирать .gif на PC.

# Сохранение отдельных фаз спрайта.

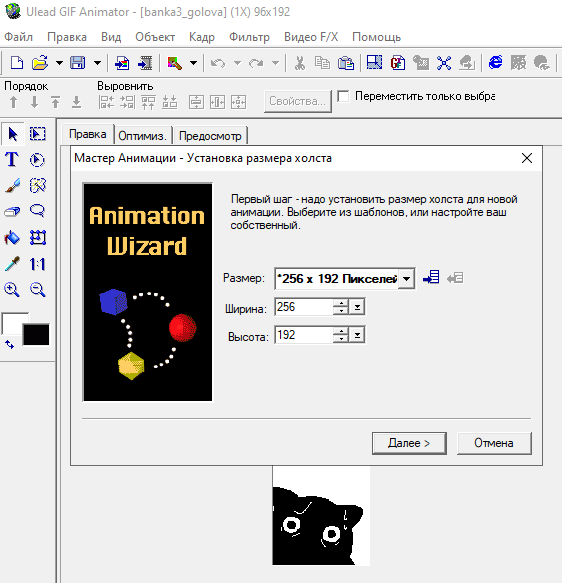
****

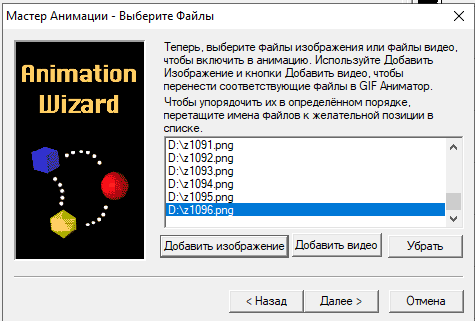
Тут всё просто и понятно. Меню ФАЙЛ, СОХРАНИТЬ КАК, кадры изображения. Название пишем , например 1. Сохранить первую фазу как 001, и далее по номерам 002, 003.. Сохранять фазы необходимо для их редакции в сторонних графических редакторах.

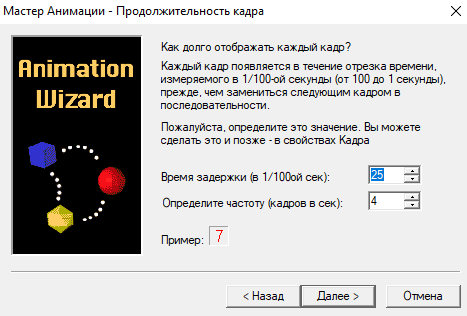
# **Сборка отдельных фаз спрайта в .gif.**

Способов два. **Первый** в ручном режиме, методом перетаскивания фаз из папки где они сохранены, выделяются мышью и тупа переносятся в нижнее окно редакции спрайтов Gif Annimator в первую фазу пустую, которая там всегда есть. К сожалению, этот способ не рабочий на Windows 10,11 у меня сбрасывается из-за несовместимости программы Gif Annimator 5.0 Crack с новыми версиями Windows 10,11 (проблема работы с обращением к папкам или диску или каталогам). Раньше, на XP я этим способом делал. Зато мне пришлось научиться работать вторым, более цивилизованным способом.

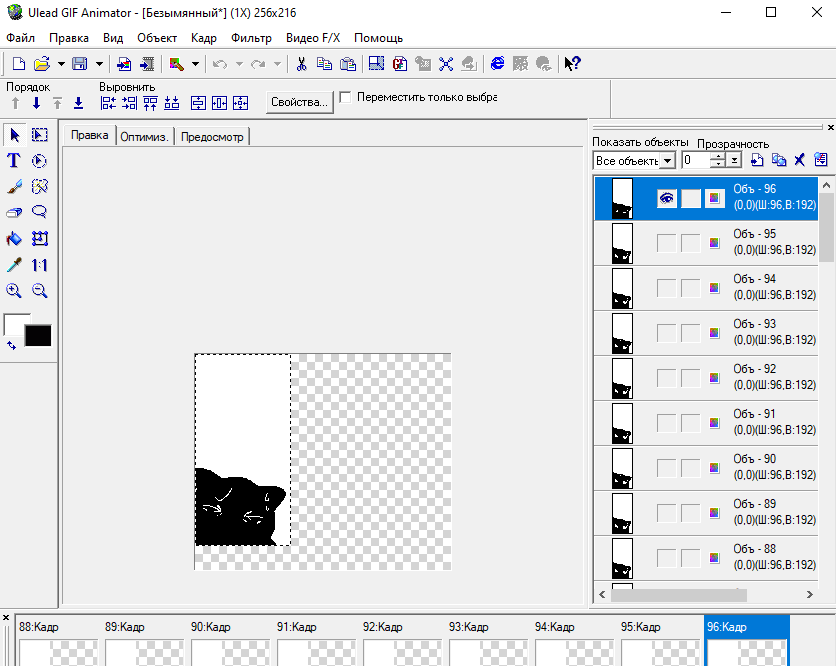
**Второй способ (совершенный).**

Заходим в меню: **ФАЙЛ, МАСТЕР АНИМАЦИИ**.  
  
  
Выставляем там размер Вашего бедующего спрайта, для этого сначала смотрим фаза спрайта какого размера, допустим 256 на 192. Выставили. Весь проект будет 256 на 192. В любом случае анимация всегда одного размера заданного в начале (все потом можно обрезать и поменять). **Жмем далее…**  
  
ДОБАВИТЬ ИЗОБРАЖЕНИЕ   
И с папки, где у Вас фазы анимации заносим их все:  
  
  
путем выделения всех фаз мышкой от первой до последней.



Следим, чтобы они все располагались от первой до последней, и если это не так, мышкой можно передвинуть прямо в этом меню по местам и жмем **далее**…  


выставляем в меню **ВРЕМЯ ЗАДЕРЖКИ** скорость прокрутки спрайта. Её берем из оригинала, если Вы его украли, или ставьте свою, если свой спрайт. В любом случае потом в программе можно регулировать все заново. Жмём далее…

  
Все готов Ваш спрайт, теперь выставляем скорость, если она разная у фаз, на каждой фазе правая кнопка мыши. И не забываем удалить избыточные точки в меню ОПТИМИЗАЦИЯ, проверяем заодно цвета, не намешали ли Вы там какие-то левые цвета в процессе редактирования, бывает не заметишь, как в Paint нажмешь и размоешь белый. Глазу не видно, а спрайт будет глючить. Приводим все к единому цвету черный и белый. Этот обряд мы всегда делаем, при любом редактировании спрайта. Вот так легко спрайты разбираются и собираются. Мне бы кто это объяснил, как я Вам.

# **Gif. не влазит в память, помоги А.А.А.!**

Проблема с «**у меня не влазит**» будет постоянно основной при работе с Zapilyator и не только. Если дома жене можно сказать, помоги, то тут придётся самим. Вариантов масса, как решить, но я бы хотел, чтобы Вы сами до всего доходили. Потому что Вы можете найти свои способы, и они могут быть лучше моих. Сделаю лишь наброски к тому, как решать проблему «не влазит» анимация .gif в память. Во первых, в .gif много фаз, удалите каждую вторую, проверьте, влазит теперь? Да для этого надо загнать спрайт в Gif Annimator и внизу мышкой выделить каждую вторую фазу и удалить, не забывайте, что у Вас есть скорость спрайта, допустим 10, и если Вы удалите каждую вторую фазу, значит надо её сделать чуть медленнее хотя бы 14, 16. Это на глаз, чтобы было плавно. Вы ведь удалили 50% фаз, значит и скорость надо менять. Во вторых проблему можно решать уменьшением спрайта в размерах. Я уже в этой инструкции описываю как менять размер спрайта кратно сторонам в Gif Annimator, та же это можно сделать в онлайн программе:  
<https://ezgif.com/resize>  
 Там это делается, как кратно процентами, так и вручную, на глаз. Кстати, эта программа прекрасно справляется с уменьшением. Ещё, когда Вы что-то уменьшаете, желательно чтобы оно перед этим уже было оптимизировано в чёрный и белый цвет. И так мыла в цвете прибавится, которое потом надо будет обрабатывать по фазам в программе **BMP2SCR**, так же как я описал ниже в этой инструкции обработку сторон векторного кубика с цветом.   
 Для того чтобы понять влезет .gif или нет, надо просто пробовать вставлять в банку памяти Zapilyator, допустим мы решили работать по быстрой памяти, в 3,4 банках, вставляем туда 1 gif. Потом компилируем проект без всего, только с этим одной анимацией .gif и смотрим сколько весит одна, потом вторая и так далее. Не факт что если на PC .gif 100 кб, он не влезет, иногда влезает и не факт что.gif на PC всего 20 кб, и что он сто процентов влезет. Там куча факторов, основной из которых это размер поверхности экрана который охватывает .gif. Как пример звездное небо 256 на 192. Дай дьявол влезет 12 таких экранов, и то вряд ли, зависит от количества кучности точек звездного неба и даже их разброса. Чтобы прочувствовать, надо пробовать поработать в программе. Необходимо с самого начала демо-творчества усвоить что готовый .gif обрезается по сторонам кратно 8, максимально рядом с рисунком, или даже впритык, поле мы оставляем только лишь в том случае, если впритык не делится на 8. Режем ровно на столько, насколько не делится. И только тогда оставляем белое поле. То есть в идеале, оно у Вас не может быть больше 8 точек.  
Чувство «влезет – не влезет» нарабатывается с годами, заодно прибудет 1001 способ, как сделать так, чтобы влезло. Но не всё влезет, бывает, хоть об стену головой не входит.

# **Рабочий стол Zapilyator-щика:**

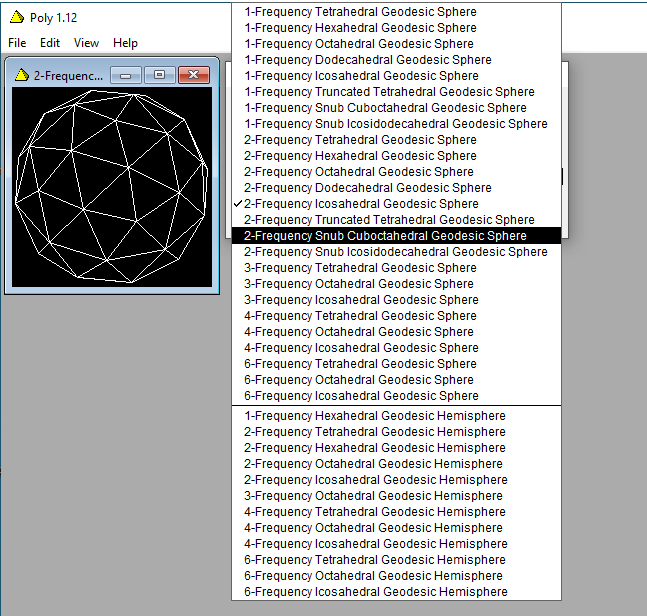
1. Gif Annimator 5.0 Crack – работа со спрайтами  
2. ZX-Paintbrush – конвертация в .scr  
3. ZX\_AAA\_Workshop – конвертация картинок с эмулятора в PNG  
4. DOSBox 0.74-2 – для запуска ESN (значок ESN перетаскивается мышью на значок DOSBOX. Это эмулятор MS DOS. ESN работает только в MSDOS среде.)  
5. ESN – для работы с образами, копирования  
<https://vtrd.in/pcutilz/SN_114.ZIP>  
**лайFUCK** – Папку с ESN я кидаю на рабочий стол, а в нее образ TRD с ARTStudio в которой всё рисую в эмуля торе UNREAL. Так при переносе значка ESN на значок DOSBOX для запуска, я сразу вижу файлы и мне не надо лазить искать, где файлы образа, т.к., DOSBOX на рабочем и ESN в нём определяет именно рабочий стол и папку на нём как рабочую. Это экономит тонны времени.  
6. Ay\_Emul – выбор AY мелодий, их подбор плеер + архив музыки  
Сайт Сергея Бульбы:   
<https://ay.strangled.net/emulator.htm>  
<https://ay.strangled.net/music.htm>

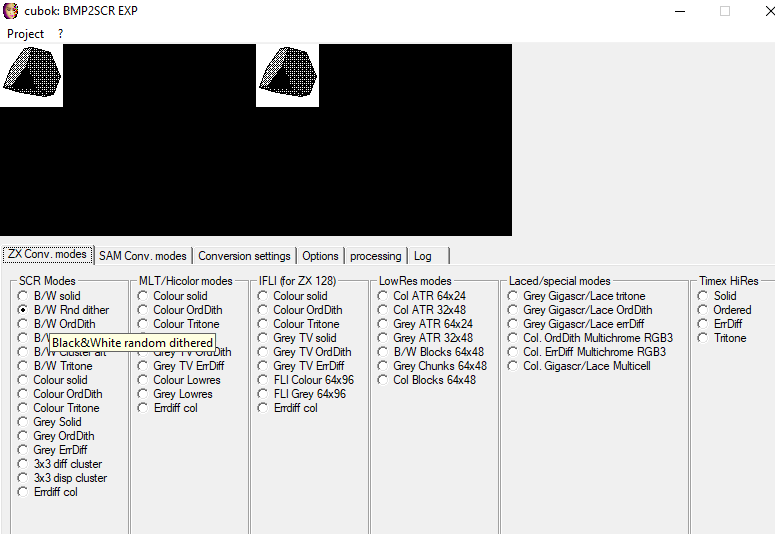
36000 AY мелодий для Ваших демонстраций.  
  
7. Unreal – эмулятор с сайта ААА. F9 там сохранить работу (например нарисовали картинку, сохранили как на ZX Spectrumе на дискету и ещё поверх жмем F9 и только тогда Ваша картинка сохраниться и запишется в образ, F3 загрузить образ. https://zxaaa.net/EMUL.html  
8. Total Commander, для всего на всякий случай.  
9. Vortex Tracker – для пере конвертации мелодий в формат pt3  
<https://ay.strangled.net/vortex.htm>

10. Сетка в виде картинки 256 на 192.png – для наглядного расчета  
11. PAINT PC – для рисования картинок  
12. BMP2SCR\_EXP\_2.11a или в папке где Вы будете обрабатывать картинки он сохраняет там где лежит по умолчанию.

# **Poly Pro 1.12**

Программа для преподавателей, которую я стырил в США из университета физиков-ядерщиков. Эта программа создана для обучения студентов разным геометрическим фигурам (программа платная, но бесплатно работает), и в ней можно делать абсолютно любую фигуру из геометрии векторами-линиями. Делать это не значит, что надо что-то считать, дьявол упаси от слова считать и думать, мы не для этого ничего не учили в школе, просто выбираете из списка готовый шаблон и мышкой растягиваете размер и крутите фигуру в любой плоскости. Сохраняется результат в 36 фазах, что смотрится довольно плавно, для глаза, и главное любым размером, какой Вам надо. Все это делается мышью. Думать абсолютно не надо. Знать математику, геометрию и прочую хрень не надо. Всё как мы любим – тупо! И ещё там можно крутить фигуру мышью в пространстве, куда закрутите, так она и будет крутится в демонстрации. Я сотни раз в ней делал фигуры и они прекрасно работают в **Zapilyator**, при этом места в памяти не кушают. Кстати сохраняется сразу в .gif и да, там сразу два цвета черный и белый, хотя можно цвета сделать любые, а потом в программе **BMP2SCR** сделать на их основе сетку градаций сторон. Для этого .gif спрайт из этой программы разбиваются на фазы, а потом каждую фазу обрабатывают специальными настройками, которые я описал ниже в BMP2SCR.

.   
  
  
  
Геометрическая фигура сделанная в Poly с цветом, градации сторон сделаны моим способом: ****

Это, то о чем я Выше писал, в программе **Poly** ставим разные цвета на стороны фигуры, далее сохраняем как .gif, далее способом описанном ниже разбиваем гиф на фазы, и загоняем каждую фазу **ЕДИНОЙ настройкой в BMP2SCR**. Там в меню выбираем этот пункт:  


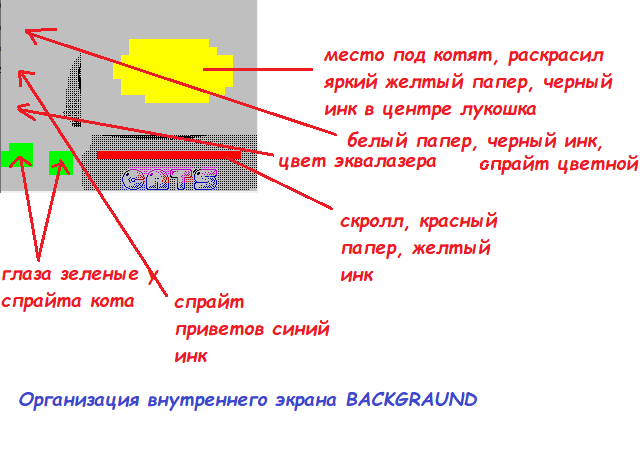
И переходим **в Conversion Setting,** где на одной фазе тренируемся двумя двигалками слоёв, выбирая нужные оттенки, и только потом этой единой настройкой прогоняем все фазы цветного кубика, сохраняя как PNG.(меню OPTION, там можно выбрать в какой формат сохранять фазы картинки, которую Вы обрабатываете) Почему единой настройкой? Потому что только так он потом будет крутиться единообразно и красиво. Тут **лайFuck** такой, сначала скидывайте в папку все фазы .gif, а потом туда же кидайте BMP2SCR.exe, тогда после того как в опциях Вы выставите «сохранять в PNG» он будет сохранять готовое обработанное в туже папку где и находится программа и фазы спрайта. Это сильно экономит время. Фактически 36 фаз можно сделать за 5 минут. То что на выходе будет размер 256 на 192 не страшно, т.к., в Gif Annimator после сборки спрайта размером 256 на 192 Вы легко обрежете до нужных Вам 8 кратных границ спрайта. Инструкция как собирать, разбирать и резать спрайты, мной дана в этой же книге. А так же их можно резать рукой с мышью проще в программе онлайн: <https://ezgif.com/resize>

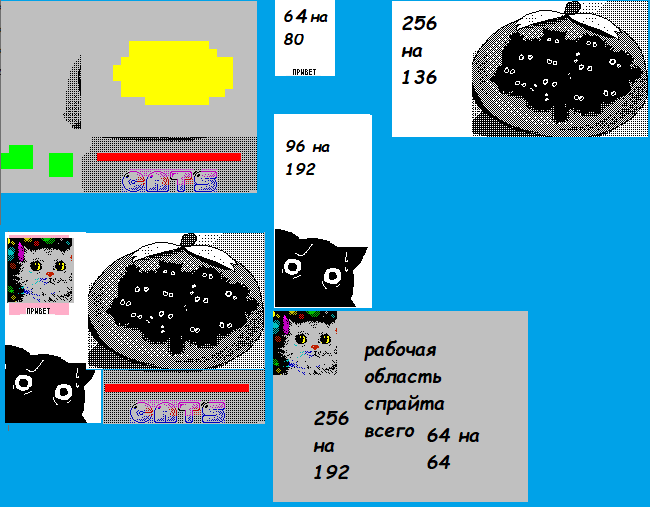
# Меню: банки памяти для анимации.

**Animation 1 (main/slow)** – банка медленных спрайтов **Animation 2 (main/slow)  
Animation 3 (int/fast)** – банка быстрых спрайтов **Animation 4 (int/fast)**

Можно я по анимации буду долго? Хотя я и так уже выше по инструкции написал нереально много и да, это всё относилось и к этому пункту меню. Если честно, то по большому счёту Zapilyator-ство (искусство делать демо без знаний программирования) - это прежде всего работа с анимацией и графикой. Анимацию можете делать свою полностью, рисуя каждую фазу, задавая скорость прокрутки в специализированных программах и собирая там же монохромные спрайты. (так например делал КАКОС НАНОС в своей знаменитой демонстрацию от которой все писили кипятком) А можете брать её из Интернета поиском по различным ресурсам. Главное требование, чтобы по итогу, после всей обработки и конвертации анимации она была пиксельной и монохромной, то есть состоящей из двух цветов: черного и белого и естественно умещалась в память и на экран 256 на 192 точки. Никаких розмытей и джипегов и лишних цветов! Итак, что такое медленное и что такое быстрое по отношению к анимации в Zapilyator? Медленная память - её я использую для вывода маленьких спрайтов, размером 1 знакоместо на 2, можно и больше, но тут действует принцип про «7 шапок», сшить можно, но они будут тормозиться, глючить при выводе лучом, гоблин заругает факом. Медленно это значит я выведу эту анимацию глючно, медленно с секущемся лучом, зато ее много влезет. Но если фаз много 200-300 но они махонькие, от 1 до 2 знакомест, то Zapilyator её в медленной памяти выведет так же круто как и в быстрой. Для больших спрайтов я использую только 3,4 банки, хотя иногда бывает и медленная, тут всё зависит от метода проб и ошибок. Так же как и эквалайзеры - спрайты должны стыковаться знакоместо к знакоместу и по горизонтали и по вертикали. Стыковаться не только между собой, но и знакоместами с рисунком бакграунда. Всё что вы рисуете на картинке в демо, должно стыковаться знакоместами. Помимо этого не забываем про инверсию спрайта, но её легко делать, я уже описывал это выше. Давайте сразу посмотрим на примере демонстрации, чтобы было ясно о чём я говорю. Мы будем делать кошку в демонстрации и её котят. В этой демонстрации будет один цветной спрайт, три чёрно белых. Это будет очень трудное демонстрация для Zapilyator-ства, там будет три мелодии с выбором, цветная заставка, картинка внутри, скролл восьмерка и эквалайзер. Полный, как говорится фарш. (на момент написания инструкции я кстати не могу сделать это демонстрацию, т.к., в Zapilyator-е последней версии что-то нарушилось и цветной спрайт не хочет работать )) ) Ещё по подготовке анимации я стал последнее время сначала пробовать загружать по одной готовой анимации в каждую банку, на пустом Zapilyator (без загрузки мелодий и картинок), чтобы понять размер каждой анимации в сжатом Zapilyatorом виде, а потом уже на основе собранной информации принимать решение, какую анимацию, в какую банку вставлять. Zapilyator позволяет загружать в него один объект, он его компилирует и в конце выдаст значение в килобайтах, сколько занимает анимация и сколько места осталось. Все значения выдаются в виде текста в специальном окне, сами там всё поймете. Для тех кто не хочет понимать, красный цвет текста говорит о том что что-то не так. Тогда надо принимать решение, либо сокращение фаз и увеличение времени воспроизведения оставшихся, или же выбор иной анимации, или же другой способ решения проблемы. Решений тысяча, вплоть до полного отказа от спрайта. Ах, да, сначала Вы делаете демонстрацию с одной мелодией и всеми спрайтами и прочим, пишите текст в скролл, и только потом, посмотрев цифру остатка памяти, принимайте решение сколько мелодий в демонстрации использовать. Допустим 40 кб в остатке, это точно 10 мелодий войдет. Остальное сами пробуйте на ощупь. Многое зависит от длинны мелодии, например. У каждого музыканта своя длинна, у музыканта Pator есть мелодии по 15 кб. А у Вас банка страниц памяти ZX Spectrum 16 кб. Оно туда не войдет. А есть мелодии по 2-3 кб, они и в 20 кб остатка войдут все вместе.

**Начинаем безобразничать на примере Кошки:**

 Эту картинку мы вставляем в формате .scr в background (внутренний экран демонстрации, который мы готовим отдельно в ArtStudio, например). Теперь я покажу, как стыкуются на ней наши спрайты. Их можно делать по одному, и постепенно наращивать сложность, добавляя по одному раз за разом, соединяя спрайты вместе на одном экране, если конечно одинаковое количество фаз у обоих спрайтов и одна прокрутка по скорости. На уровне «Профи» я делаю иногда сразу все спрайты и сразу собираю демонстрацию под ключ. Но я, как шахматист Zapilyatorщик научился всё держать в голове, расставлять их на сетке в уме во снах. Всё это придёт со временем, и кто будет работать в этой программе больше 2-3 лет, также сможет сделать. Кстати ZX Spectrum-исты дали мне имя «БОГА ZAPILYATOR», после побед на всех конкурсах ZX Spectrum-изма. Правда потом изгнали из тусовки за то что я беру графику и музыку без спроса авторов живых и мертвых, хотя честно скажу я смотрел на могильный камень Алекса Филимонова (по фото) и спросил, можно мне твою музыку использовать, она сказал бери... )) Но это ZX Spectrum-исты, они такие суровые. Существует масса способов, как решать проблему нехватки банок памяти для анимации, например объединение 2 спрайтов в один, трёх или даже 4. Если у одного спрайта 24 фазы, а у другого 12, а скорость их одна, можно 12 фаз сделать 24, повторив каждую и всунуть их в 24 фазный спрайт. Это чтобы освободить банку памяти под другой спрайт. Возможностей полно, и решение таких проблем это и есть кайф от работы в Zapilyator. Однообразия тут не будет. Постоянно новый вызов и решение художественно-анниматорских задач. Впихать не впихуемое в не впихуемое – истинное наслаждение для настоящего мужчины спектрумиста!



Страшно? Мне тоже. Итак, как видите четыре спрайта лежат чётко в границах знакомест:  
64 на 80 – спрайт скролла «Hello» фактически 1 знакоместо в ширину и 64 точек в длину, поэтому его кладём в медленную банку №2. Вы края спрайта белого не смотрите, края я делаю, уже когда спрайт готов. Читайте далее сейчас поймете зачем там края такие огромные.  
96 на 192 – в скоростную банку №4 т.к. он находится ниже чем котята, тяжелый спрайт много фаз. А вывод спрайта идёт сверху вниз.  
256 на 136 – в скоростную банку №3, много фаз, тяжелый спрайт, он выше по экрану чтобы не было эффекта сечения луча, лучше выставить его в 3 банку, а не в 4. Помните, если у Вас сечется луч, значит Вы плохой человек, переделывайте, ищите другие способы. Луч сечься не должен, это все равно что ребенка обидеть. Такой эффект противоречит всем законам демо-сцены. Если сечется луч, считайте что Вы не сделали демонстрацию, а пришли к выводу, что надо что то переделать. Это не вариант такое собирать.

# Цветной спрайт. Эффект двойного экрана.

256 на 192 – это очень интересный спрайт цветные картинки формата .scr размером 64 на 64 точки их 10 штук. Все они в верхней части экрана, и я буду их выводить с замедлением 255, которое максимально на данное время разрешает делать Zapilyator. Поэтому я это кладу в медленную память банку один, и выставляю 255 скорость вывода одной фазы. (под загрузкой спрайта есть меню скорости, его не надо трогать если Вы скорость выставляли в гиф аниматоре для не цветного спрайта например, программа сама будет проигрывать спрайт по умолчанию, так как его задумывал автор, задавая скорость фаз вручную) Рисуются картинки цветных фаз в ART Studio, далее сохраняются, как я описывал выше. Цветные картинки-спрайты можно делать двумя способами. Ах, да эти картинки не обрезаются как gif, они все-таки стандартные экраны ZX Spectrum, их фон белого цвета, это как фон спрайта белого цвета.  
  
 **Делаем первым способом, деревянным как рубль**:

Сохраняем их в правильном формате .scr и кладём в папку с именами 1,2,3 и.т.п называем в зависимости от того как надо их показывать поочередно цифрами от 1 до… . И сжимаем их в этой папке, как zip папку. Подсовываем в Zapilyator эту сжатую папку, как обычную gif-ку. Да, Zapilyator это сожрёт, он понимает что .scr в папке .zip архива - это фазы спрайта, это в нём предусмотрено искусственным интеллектом ещё первого создателя программы Nyuk. Кстати создатель программы лично мне запретил в ней работать. Но так вышло, что последние лет 7 только я один в не работаю, а все остальные нет. ВНИМАНИЕ! Обязательно выставляем параметры paper, inc, bright в меню под **BACKGRAUND**, чтобы они соответствовали цвета фона картинки и спрайта, об этом я скажу несколько раз в этой книге. Без выставления этих параметров при выводе цветного спрайта любым способом, картинка бакграунд (внутренний экран) будет затираться цветом фона цветных спрайтов.

**Второй ТРУДНЫЙ способ в осознании, но более передовой способ вывода цветных спрайтов, если есть возможность постичь, пользуйтесь только им:** Это сборка в программе **ZX Animator (KAKOSA)** специализированой .gif с зашифрованными цветовым фоном. При этом можно задавать в **Gif Annimator 5.0** разную скорость вывода фаз спрайтов, что не позволяет делать 1 способ, там скорость фиксируется сами Zapilyator-ом от 1 до 255 и всё. А если надо медленно выводить картинки, то 1 способ этого не позволяет.   
  
Итак:

1. рисуем картинки на спеке и сохраняем в формате .scr

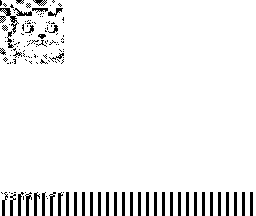
2. переименовать их в порядке демонстрации от 1 до.... СТРОГО 1,2,3,4 и.т.п. Никакие другие имена при переименовании не используются.

3. Скинуть в ту же папку где находиться ZX Annimator Kokosa

4. В ZX Annimator Kokosa нажать на 1, там в окне будут видны все названия файлов от 1 до....

5. Экспорт....

6. Автоматически соберется много файлов BMP весом 163 kb каждый. Картинка .scr преобразуется в не цветную без атрибутов в bmp **256 на 216**. Они не 256 на 192, а 256 на 216 (это + кодировка цвета по методу кокоса)



Изображение выглядит так. Странно, оно 256 на 216 точек, автоматически появляется нижнее не понятное, зашифрованное для инопланетян. Это и есть Ваши фазы будущего цветного спрайта, а низ это код цвета картинки понятный запиляторному «Богу Кузе».

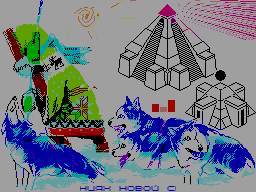
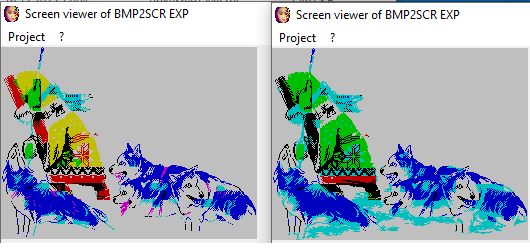
7. В GiF Annimator 5.0 собираем его как обычный .gif, задаем скорость показа фаз, гифка размером 256 на 216!!! 216, а не 192, т.к. в нижней границе зашифрован шифр с цветами для инопланетян.

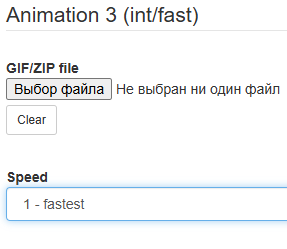
8. Кидаем её в 1 банку памяти Zapilyator.

Смотрим, сколько в остатке докидываем, дорисовываем картинки по максимуму.

Цвет появится в Zapilyator, он сам распакует кодировку Какоса. Zapilyator в курсе что это Какос. Какос это не какос, это программист и музыкант из города Туапсе. Вундеркинд ZX Spectrum-изма. Изобретатель программ, который в мире еще не существует. Возможно он всё таки инопланетянин.

Скачать Какоса ZX Annimator можно тут: <https://zxaaa.net/ARHIVE/zxaaanimator.zip>  
  
**А можно ли сделать в Zapilyator двумя тремя картинками цвета, например эффект двойного экрана?** Да можно, я это делал сотни раз. Берём две одинаковые картинки и раскрашиваем их по разному, например в ART Studio и тупа сохраняем как scr. Далее кладём их в пак .zip и подсовываем в любую банку памяти гиф Zapilyator. Устанавливаем максимальную скорость. Включаем в эмуляторе UNREAL noflic и у Вас двойной экран готов. Вот пример последней работы маэстро:  
<https://zxaaa.net/view_demo.php?id=15769>

  
Как видим тут неестественные для спека цвета у фона, на который я наложил другие спрайты, эквалайзер и скролл. Фон делался из двух одинаковых картинок разной раскраски:  
  
Две картинки .scr пакуем в единой папке в zip архив, суем в 1,2,3 или 4 банки, лучше все таки в скоростную чтоб стабильней были цвета. И ставим под меню загрузки gif показатель Zapilyator:

FASTEST  


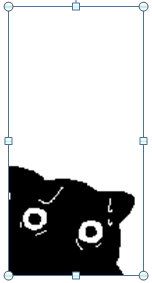
Если поставить скорость вывода цветного спрайта как 1,2,3 то будет просто мигать цветом, тоже как вариант такой эффект я делал в невероятном запиле:  
<https://zxaaa.net/view_demo.php?id=8657>  
На каждую 3 скорость фазы кубика я выставлял цвет, запуская сразу три картинки с разными цветами, создался эффект задержки цвета в движении. В общем, тут проводите свои эксперименты и добивайтесь дизайнерских высот.   
  
 **Кстати говоря, вот Вам ейсчё лайFuck!**

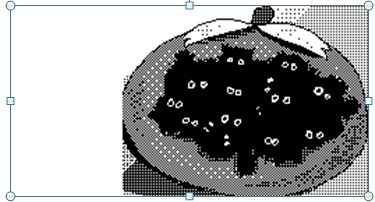
**Можно ли внутренний экран Backgraund (картинка самой демонстрацию), вставлять не в окошко Backgraund, а вставлять в виде фаз спрайта цветного например в одну из банок 1,2,3,4 ?**

Да можно! И даже скорее нужно если у Вас есть свободная ячейка памяти, желательно 1,3,4, а лучше 3,4, тогда делайте всегда два экрана, раскрасьте, там например, хотя бы название демонстрацию красиво, или ещё что то, раз всё равно у Вас две картинки сменяются и смело вставляйте. Но, я ещё 1 фазу как правило, всё равно дублирую и вставляю в меню background. У Zapilyator пока нет красивого перехода затухания от заставки перед демонстрацией в саму демонстрацию к картинке внутреннего экрана и если сразу вставить как zip папку с двумя цветными картинками **ZX Spectrum**, то будет небольшой скачок глюка перехода, в виде быстрой смены знакомест на экранах. Оно будет единожды, но всё равно неприятно, гоблин заругает. А если вставить 1 фазу цветной картинки в меню бакграунда, а затем 1,2 фазы этой же картинки как zip вставить в банку памяти для gif, то глючного перехода не будет, его сожрет 1 фаза цветной картинки, потому что в Zapilyator внутри кодом прописана верный переход со стиранием экрана. Опять же, тут всё зависит от оставшейся памяти, ячеек памяти и прочих аспектов. Но почти все демонстрации я делаю через zip и бакграунд одновременно. Это те демонстрации, где я использую 2 экран, для поиска новых дизайнерских цветов, как то красный и желтый даст оранжевый, ранее не доступный цвет **ZX Spectrum**. В общем, можно и даже нужно вставлять два бакграунд экрана для более крутой раскраски, если есть свободная банка памяти после всех Ваших желаний по анимации.   
Почему я советую вставлять цветные картинки для двойного экрана всё таки в 3 или 4 банку памяти – то есть быстрые? Потому что в медленных банках не будет двойного экрана, луч может сечься. Там просто не хватит скорости инта (скорость процессора Z80) для прокрутки двух-трех цветных картинок. А в 3,4 банке будет просто двойной экран. Тут надо Вам пробовать самим, в зависимости от задачи демонстрации. Всё о чём я Вам тут пишу, в принципе я изучал методом тыка, проб и ошибок в течении 7 лет ежедневной работы в Zapilyator. Всё это мои исследования в области Zapilyator-ства. Вы можете найти и другие методы и способы, о которых я дурачёк, даже не догадывался эти семь лет жизни. С радостью их изучу и внедрю.

# **Границы белого фона на спрайте.**

Теперь Вы скажете, а как же границы белого на спрайте? Они могут пересекаться? Да, но нельзя только чтобы пересекался сам рисунок спрайта (черный). По границам я делаю полные экраны всегда для цветных, т.к. его нельзя обрезать, и монохромные .gif спрайты я обрезаю по видимой части, чтобы экономить память.

тут до 256 точек не надо белого догонять. Хватит кратное 8. Экономим память везде где реально конечно.



Котят до 192 тоже не надо догонять, там нет ничего движущегося. В ширину 256, т.к., от левого края мерим, там нельзя белое убрать, эта граница нужна **ZX Spectrum** для понимания расстояния выставления.

Котэ с глазами я обрезал по длине, то есть по «X-оси», т.к., он мне там далее ничего не кажет из рисунка кота. Котят обрезал по высоте, то есть по Y оси. «Хелло скролл» я обрезал максимально и по длине «X оси» и по ширине «Y оси». Спрайт лишь упирается в левый верхний угол, т.к., он не в углу, а чуть ниже, ближе к середине, остальное мне нужно лишь для выдерживания расстояния от верхнего левого края. Кстати спрайт «хелло» я подкрасил на примере розовым, чтобы Вы понимали, что поверх него в его видимой белой границе лежит цветной спрайт, это допускается. Всё что белое в пределах знакомест - никак не глючит, но строго в пределах знакомест стыковка. Если граница одного спрайта будет, допустим 57, а стыкуемого к нему 103. То эта связка будет глючить, это будет видно. Как правило, эта проблема видна выводом не полного спрайта, а лишь его раз мыленной части. У человечка с ручками не будет ручек и.т.п. . Это надо просто понять. Как видим, все спрайты имеют белые поля. Расчет спрайта мы делаем всегда от левого верхнего угла по оси X и Y, а то что ниже и справа, без рисунка - выводить не надо, но помним что сам спрайт всегда по всем сторонам обязан быть кратен 8, то есть равен знакоместу. Нельзя выводить спрайт 81 точка, только 80 или 88, но не 81,82,83… Если Ваш спрайт не делится на 8 по какой либо из сторон, ищите вариант как это сделать, либо его укоротить перерисовав внутренний рисунок, либо удлинить до кратной 8 длинны и ширины, если позволяет рисунок Вашего бакграунда. Белое поле спрайта неминуемо закроет, что либо на рисунке, если в том месте вывода будет что-то нарисовано. Лавирование между рисунком бакграунда и границами спрайтов и есть сложность работы в Zapilyator. Или Вы станете немного дизайнером и художником, или же Вы не сможете делать демонстрацию в Zapilyator, просто сложите руки и скажете это не возможно, я лучше сделаю демонстрацию кодом, чем так мучится, и естественно ничего Вы не сделаете.

# Прозрачность в .gif

Раньше я делал вместо белого цвета фона спрайта (gif) прозрачность везде при финальной сборке, сейчас заметил, что из-за этого глючат спрайты, и я стал оставлять белый цвет. Если вы будете двигать спрайт в Gif Annimator 5.0, или подобных программах, то фазы спрайта в сдвинутом месте автоматически станут прозрачным. Тогда я выгружаю каждую фазу спрайта отдельным экраном и делаю загрузку этих фаз в PAINT на PC и оттуда CTRL + S сохранить, фаза спрайта быстро сохраняется как белый фон общий. Я заново собираю спрайт, расставив скорости фаз в соответствии с оригиналом. В общем, от прозрачности в фазах спрайта я ушёл. Память она не экономит (я проверял), а глюки добавляет.

**СПРАЙТ ВСЕГДА ДЕЛАЕМ (РАСЧИТЫВАЕМ) ОТ ВЕРХНЕГО ЛЕВОГО КРАЯ, ОБРЕЗАЕМ ПО ГРАНИЦЕ ЗНАКОМЕСТ КРАТНО ДЕЛЕНИЮ НА 8 ТОЧЕК!!!**

# Работа с готовыми gif, кража контента:

Повороты gif, оптимизация, монохромизация, уменьшение кратно процентам или границам, все это делается на бесплатном сайте:

<https://ezgif.com/>

Он руками русифицируется в браузере. Там всё понятно, пробуйте, программа мощнейшая. Работает в России только со шнурка. Это просто нереально крутая программа. Изучите её и половину того что тут написана она тоже делает. Я много не знаю, но она делает нереальные вещи автоматически. Я описал лишь то, что я в ней сам делал.  
 Так же монохромность .gif делается реоптимизатором Какоса-Наноса, (unoptimizeGIF) приложил по ссылке: <https://zxaaa.net/OTHER/zapilyator_programms.rar>  
Инструмент мощный и простой в освоении, там есть файл описатель, запускается в командной строке Total Commander или любого другого коммандера.  
  
 Фактически этот сайт <https://ezgif.com/> Zapilyator Photoshop для тупых, таких же как я - А.А.А.. То есть для нас всех раз Вы тут. Думать не надо, приложил-выгрузил. Даже я – Великай А.А.А., который не любит думать, там сам разобрался. Так что если нужно готовый .gif повернуть, обрезать и прочее, там это делается за пять секунд. Именно с этим инструментом я делаю за час два целиковое демонстрацию. Ведь самое ценное что есть в жизни - это наше время, не надо его тратить на рутину, делайте демонстрацию качественно, в кайфе от пива и быстро. Оставьте время на спорт и пресс на турничке, а так же просмотра сериала Fargo и прочего вкусного и интересного, а демонстрация это баловство.   
  
 **Где красть gif?**

Во первых google поиск, такие запросы как : 1bit .gif, monochrome .gif  
черно белые.gif и прочее сами думайте или спросите у ИИ.  
  
Во вторых сайты хранилища:   
<https://gifcities.org/search?q=pixel&offset=0&page_size=200>

<https://pixeljoint.com/>

Где красть картинки?  
<https://tomseditor.com/gallery/&lang=en>

<https://zxart.ee/>

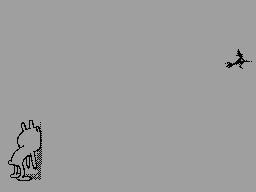
Какие картинки лучше красть? Новичкам, советую начать с монохромных 1bit, или картинок с Commodore 64, они 100% на ZX Speccy потом смотрятся даже в улучшенном виде. Труднее с MSX, но тоже реально. Проще всего брать готовые работы с сайта zxart.ee от гражданина мороза99, всё кромсать, соединять, как по душе и по кайфу, и делать красивые демонстрацию. Самое хреновое, когда Вы тупа конвертируете цветные JPG картинки PC в BMP2SCR с битыми атрибутами, это нубство. Это типа картинок AER или Дворкина:  
<https://zxaaa.net/view_demos.php?a=Dworkin&vers=23>  
Так не делайте, это идиотизм. Дураков на спектруме и так до хрена хватает, могу отсыпать. АЕR просто алкаш, ему можно, он демонстрацию после стакана водки делает. Смысла нет делать такое днище под пивом. Градус другой, можно и красиво сделать за 5 минут. В целом, зачем делать демонстрацию, если Вам не хочется рисовать и что то делать?  
  
 Есть куча художников уровня Grunge, кто рисует 100 картинок в секунду, берите их, кромсайте, делайте коллажи, соединяйте по вкусу. Главное? на самом деле классная идея.Даже мороз99 мне как то говорил, бери картинки делай что хочешь, лежат у меня пылятся. Потом конечно под воздействием Хайпа переобулся, «двуликий анус» ZX Spectrum-изма. Кактус перестройки, фак завернутый в фольгу ….

# Подготовка анимации (объединение двух спрайтов)

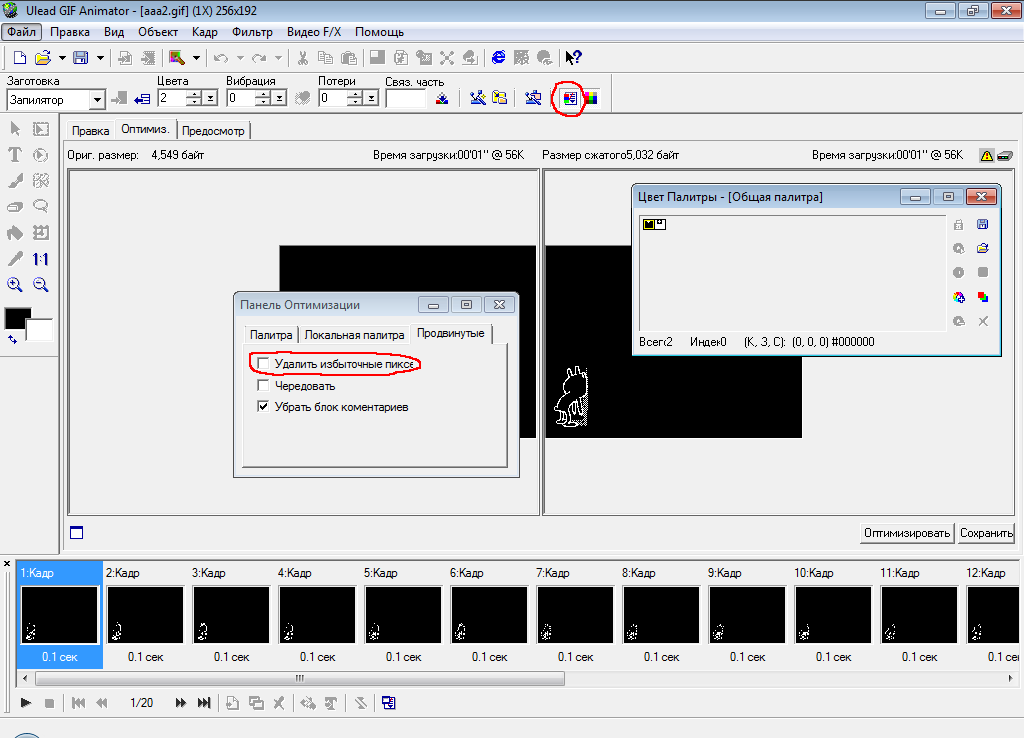
Самая сложная в человеческом понимании часть создания демонстрации. Адский труд, особенно, если Вы делаете это первый раз, и рядом нет взрослого. Я очень подробно всё объясню, постарайтесь прочитать этот абзац два, а лучше три раза в туалете, прежде чем начать делать демонстрацию, чтобы не терять своё время напрасно и не тратить нервные клетки. Анимацию вы можете взять, украсть в интернете, можете нарисовать, что вряд ли, ибо отсутствие творческого начала в серой массе не позволит. Помните, что Вам необходимо уместиться в экран 256 на 192 точки. Чем больше кадры анимации, тем меньше фаз. Конечный, подготовленный gif ,файл не должен превышать 12 килобайт, иначе депилятор напишет красную надпись что **Long**. (это единственная и самая страшная надпись) Обязательно установите себе программу **Unlead Gif Animator v5**, взломанных версий полно в Интернете. Другие аниматоры не подойдут, ибо процессы сжатия картинок в них не совершенны. Найденный **gif**, вы должны загрузить в **Unlead Gif Animator**, далее по тексту **UGF**. В UGF выбрать вкладку **ФАЙЛ-СОХРАНИТЬ КАК-КАДРЫ ИЗОБРАЖЕНИЯ**, далее указать папку сохранения, желательно отдельно создать и нажать в названии цифру 1. Теперь все кадры скинуться в папку автоматически в виде 1,2,3 картинок и так далее в зависимости от количества фаз. Далее работайте с фазами, обрабатывайте их где угодно (Art Studio, Paint PC, BGE) и приводите к надлежащему виду и размеру 256 на 192. Главное условие это то, что **gif должен быть 256х192, двухцветный**. Вы можете проигрывать и не один спрайт, а столько, сколько укажите на экране, например ААА сделал кадры человечка снизу слева и баба яги справа вверху. Единственное условие количество фаз спрайтов, даже разных, должно быть одинаковым. То есть 20 человечка, а значит и 20 фаз баба яги. А если баба яги всего пять от природы, а человечка 20? Тогда в каждый кадр человечка добавляйте баба ягу, по схеме:

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20  
  
1,6,11,16 - фаза баба яги №1  
  
2,7,12,17 - фаза баба яги №2  
  
3,8,13,18 - фаза баба яги №3  
  
4,9,14,19 - фаза баба яги №4  
  
5,10,15,20 - фаза баба яги №5

После этого у Вас будут 20 экранов с двумя спрайтами которые одновременно будут работать:



Затем все фазы заносите в UGF, где в самом конце перед сохранением обязательно сделайте: **В закладке "Оптимизация" Ulead GIF Animator'a открываем "Панель оптимизации", закладка "Продвинутые" и снимаем галочку "Удалить избыточные пиксели**.



После этого сохраняйте как **gif спрайт.** Теперь, считайте, что демонстрацию Вы уже сделали. Ибо подготовка спрайта - самое трудное. Я не касаюсь тут описания работы самого **Unreal Gif Annimator 5**.

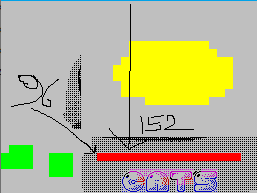
# Меню: Slide Show (ZIP of compressed images)

Эта функция пока ещё не работает в Zapilyator 3.0. В следующих версиях сделаем, если Активатор не умрёт, все предпосылки умереть от голода и холода у него есть. Это будет смена множества цветных картинок через различные эффекты гашения экрана – переходы. Для быстрого создания сборников картинок после демонстрации на демо-пати. Илон Маск уже в курсе. Активатор тоже. Переходы гоблина есть, всё как бы ясно, но сил делать нет. Последние правки Zapilyator, Илон Маск ругался матом, он постоянно говорит Активатору что на планета земля это сделать невозможно, а Активатор учит ИИ и в итоге он выдает верный код, так то мы уже Илона Маска всем азам написания демонстраций на ZX Spectrum научили. Но Маск он такой вредный…

# Меню: MusicBank (ZIP of PT2/PT3 files)

В эту ячейку мы кладем папку в ZIP архиве с музыкой формата .pt3, названой цифрами от 1 максимум до 10. Всего можно в зависимости от остатка памяти проиграть 10 мелодий, все они будут переключаться кнопками от 1 до 0, играть авто проигрыванием и затухать в конце. Мелодии называете цифрами указывая интеллекту, что за чем играть, то бишь последовательность 1,2,3,4,5…10 . (названия файлов должны быть как 1.pt3, 2.pt3,3.pt3,4.pt3,5.pt3,6.pt3,7.pt3,8.pt3,9.pt3,10.pt3.) А в память Zapilyator их загрузит сам в том порядке, который наиболее удобный для него. Допустим, в банке с анимацией 3 странички есть место в 5 кб, он туда загрузит 5 кб, а в другую 10 мелодию, и так далее, это Zapilyator сам решает. Если мелодии не влезут, то Zapilyator в конце напишет какая не вошла. Уберите её из архива и всё, или уберите другую, чтоб влезла эта. Если Вы вставляете несколько мелодий, то в 1 верхнюю ячейку для музыки не надо ничего вставлять. Оно и не вставится, Zapilyator напишет «ступид крези». Музыку мы готовим аналогично верхней ячейки для одной мелодии, прогоняя все через **Vortex Tracker** в формате .pt3. Да, в эти ячейки оооочень длинные мелодии вставить не получиться, ибо он раскидывает мелодии по 16 килобайтным ячейкам памяти, которые и без того заняты гифками и прочим. Но до 10 килобайт вполне влезут. Хотя мелодии музыканта Push по 8 минут я вставлял аж 10 штук без проблем. В общем, пробуйте методом личного исследования. **Zapilyator** надо познавать и изучать, я не исключаю того что в нём есть жизнь, как на Марсе. Что тут непонятного?

# Меню: **Scroll setup**

В текст скролла пишем текст скролла. На русском и нерусском. Знаки препинания, какие то в шрифте есть, если чего-то нет, дорисуйте фонт и вставьте свой.   
Скролл бывает 16 и 8 точек. 8 точечный скролл бывает горизонтальный и вертикальный.  
  
**Scroll width** - длинна скролла в знакоместах  
**Horizontal offset**, - отступ от левого края в знакоместах  
**Position** - вертикальная позиция скролла Текст в скролла я пишу в самом конце, его длинна зависит от остатка памяти, если её мало много не напишешь. Можно писать в любом текстовом редакторе, затем перенести. Буквы «Ё» ни в одном шрифте нет, так же нет некоторых знаков.   
  
 8 скролл, это будет отдельное меню, и отдельные показатели длинны, высоты и отступа. Тут всё можно понять тыком, а так же сеткой знакомест. Я уже это объяснял на примере эквалайзеров, суть такая же, ничего не меняется. Единственное, поймите если скролл 32, а Вы делаете отступ от левого края, допустим 2 знакоместа тогда длинна скролла уменьшается на два знакоместа с конца скролла. Иначе будет глюк на экране. Это нужно просто представить у вас паровозик на 32 вагона и депо на 32 вагона-места. Если Вы решите загнать паровозик глубже в депо на два вагона, то вам надо с конца состава убрать сначала два вагона, иначе они пробьют депо. Аналогично и вертикальным скроллам. Если Вы отступаете слева и уменьшаете длину, то скролл можно располагать по средине в чем-то как небольшой отрезок. В общем смотрите пример, где красное это место для скролла:  
  
  
**  
  
96 точек слева и 152 сверху, значит у нас 96 разделить на 8 = 12 и 152 разделить на 8 = 19  
  
горизонт 12  
вертикаль 19**длинна скролла (красная область) 144 точки это равно 18  
  
Итого 12,19,18  
  
Вот и все расчеты. Всё начинается с бакграунда, как Вы его подготовите, так далее и пойдет. Естественно на нём всё должно попадать тоже в знакоместа.   
  
 **Надо ли скроллу выставлять цвета paper, ink, bright, которые есть под ним в меню Zapilyator?**  
 Если честно, лучше не надо, делайте это на бакграунд картинке внутреннего фона демонстрации. Если сделали, тогда просто не трогайте это меню, Zapilyator в первую очередь берёт цвета из бакграунда, и если есть бакграунд, он на все эти цвета в Zapilyator-ном меню не обращает никакого внимания. Но если у Вас совсем нет внутренней картинки в демонстрации, тупа всё на белом или на чёрном фоне, тогда выставляйте цвета скроллу и цвет папер и инка и брайта самого экрана в соответствующих меню. Хотя это бред, конечно, я даже в худшем варианте всё равно подгружаю хотя бы только с раскраской знакомест бакграунд, зачем делать демонстрацию, если лень её делать нормально?  
  
Выбор шрифта в 8 точечных видах скроллов происходит вручную путём следования из меню:  
**Scroll font**по пути: папка с Zapilyatorом:   
src  
public  
resources  
и там увидите папку с шрифтами, тыкните и шрифт подключится. Это мы доработаем в процессе.

# **Меню: Моноблок, Счётчик прерываний.**

**Создать моноблок** (объединить все файлы в boot.B)Объединяет boot.B, splash.C и все страницы данных в один файл  
тут все понятно, сделать демку под бейсик.

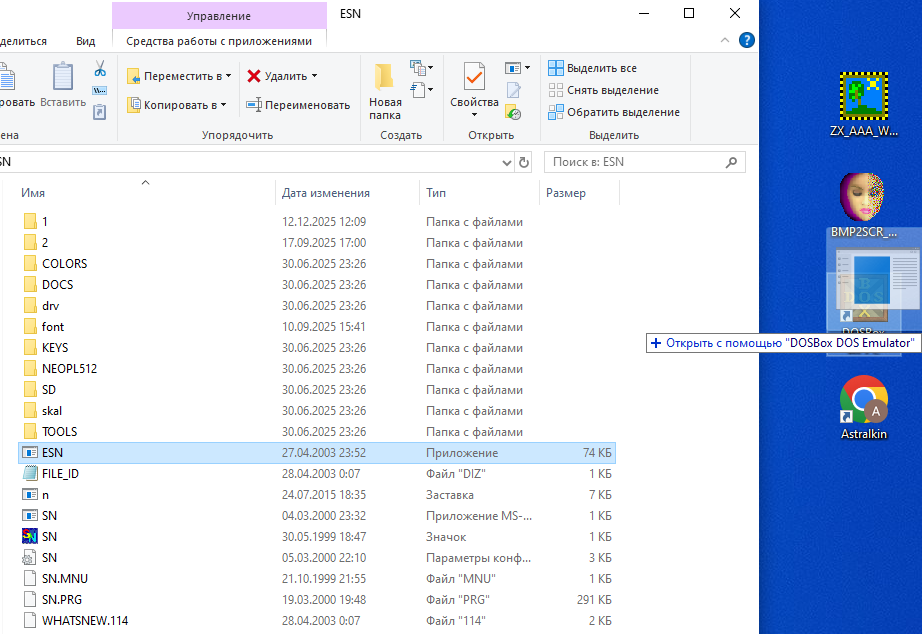
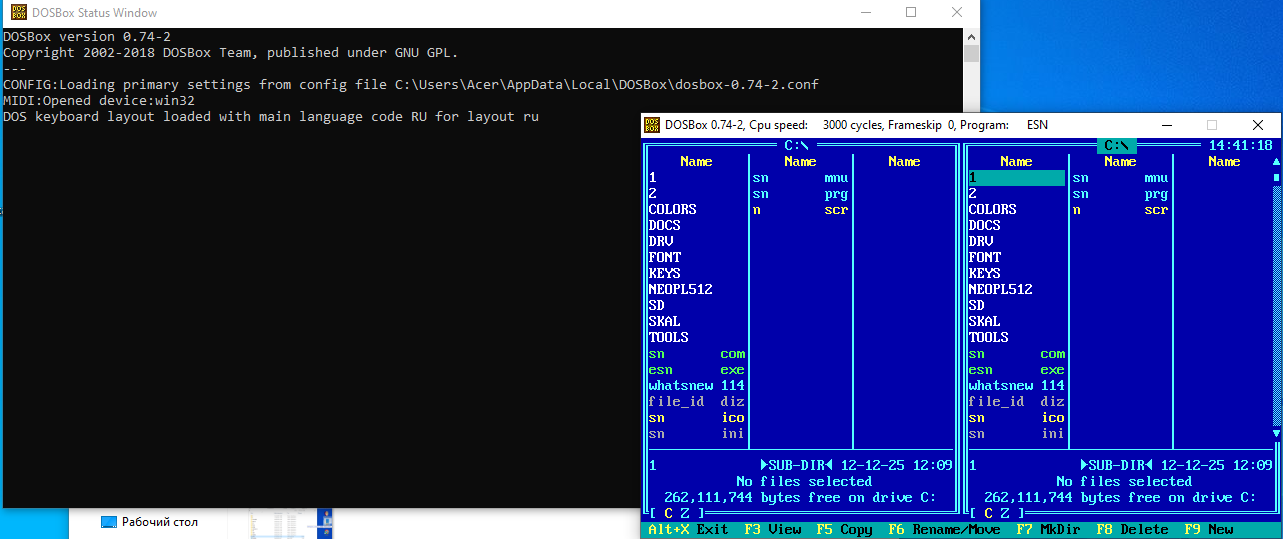
**Счетчик прерываний** (добавить переменную int\_counter)Добавляет переменную int\_counter для подсчета прерываний  
Счетчик прерываний, это для тех кто понимает о чем я.

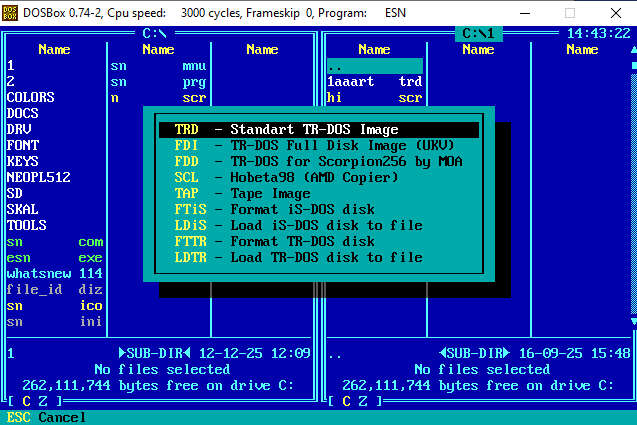
Можно создавать моноблок, а можно не создавать. От этого зависит размер файла в конце сборки (количество секторов и файлов на дискете). Это просто фича такая приятная.  
  
Счётчик прерываний сделан для умных программистов Zapilyator, кто хочет видеть влезет в ИНТ вся работа или нет. Тормозятся спрайты, музыка или нет.

# **ESN –копирования ZX файлов в образ .TRD.**

Этот абзац я пишу в основном для старых, древних как мамонты Спектрумистов, кто всё забыл, а так же для новичков кто ничего не знает, а очень хочет написать демонстрацию. Без этих знаний, нормальное демонстрацию не написать, потому что в ART Studio мы всё раскрашиваем и рисуем финальную версию. Копирование файлов .scr в образ TRD необходимо для того чтобы в этом потом рисовать в программе ART Studio из под эмулятора. Инструкцию к программе ART Studio я выложу в виде отдельной профессиональной книги на русском языке (перевод с Оксфордовского) в формате PDF. Программу ESN можно скачать на сайте Virtual TR-DOS тут: <https://vtrd.in/pcutilz/SN_114.ZIP>  
Там же можно скачать абсолютно любой софт для ZX Spectrum, может тот на котором Вы работали в детстве и по которому скучаете. Не удивлюсь если кто-то захочет спрайты делать в какой-то древней программе спрайтов с ZX Spectrum. Тут, как говорится, ностальгия у каждого своя. Я даю лишь основные знания, а инструментарий можете пользовать любой.   
  
Далее следите за руками ААА:  
На рабочем столе создаем папку с программой **ESN** внутри. Устанавливаем программу **DOS BOX** с официального сайта:  
<https://www.dosbox.com/>

У Вас на рабочем столе должна появится иконка DOSBOX и папка c ESN. Перетаскиваем ярлык ESN с папки где ESN на ярлык DOSBOX, мышкой. Вот так:

  
Запускается ESN, появляется знакомое по фильмам из прошлого меню НОРТОН КОММАНДЭР, считайте оно и есть!  


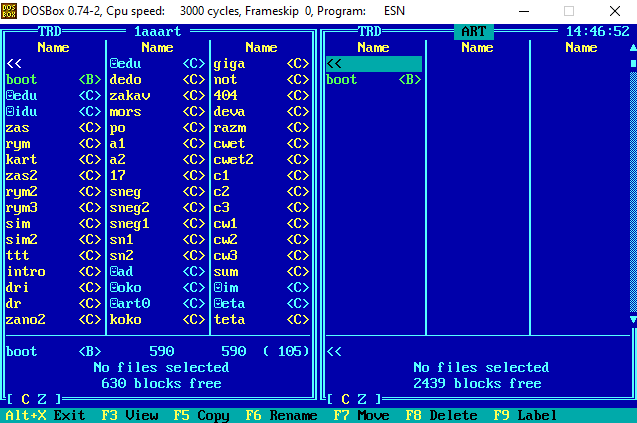
Управление простое. Можно постичь даже методом перебора всех кнопок. F6 переименовать файл, F8 удалить, F9 создать образ, SPACE – выделение файлов в каталоге. ESC - Выйти, стрелочки перемещение, табуляция перемещение между окнами.  
  
**F9 – создание образа, нас интересует только TRD!!!**  
Выбираем место на диске, где создаём образ TRD и нажимаем кнопку Inter тут:  


Образуется TRD это и есть дискета 720 кб. Уже отформатированная. Пустая. Записываем туда ART Studio скачанyю с сайта VIRTUAL TR DOS, какая Вам нравится:

* [**Art Studio v1.6RD (Rus)**](https://vtrd.in/system/ART_R_OV.zip) by Oxford Computer Publishing'85, Oleg Vdovin
* [**Art Studio v2.01**](https://vtrd.in/system/ARTSTUCG.zip) by Oxford Computer Publishing'86, United Coders Group'96
* [**Art Studio v2.01**](https://vtrd.in/system/ARTPHANT.zip) by Oxford Computer Publishing'86, Phantom Group'94
* [**Art Studio v2.01**](https://vtrd.in/system/ASTUD_RS.zip) by Oxford Computer Publishing'86, Real Software'98
* [**Art Studio v2.01**](https://vtrd.in/system/ARTSRLDG.zip) by Oxford Computer Publishing'86, Red Limited Drunkards Group'95
* [**Art Studio v2.01**](https://vtrd.in/system/ASTUDSUR.zip) by Oxford Computer Publishing'86, Artem Surunov'96
* [**Art Studio v2.01**](https://vtrd.in/system/ASTPROFI.zip) by Oxford Computer Publishing'86, TOO Profi'94
* [**Art Studio v2.01**](https://vtrd.in/system/ASTUD_HW.zip) by Oxford Computer Publishing'86, Hardware Wizard'97
* [**Art Studio v2.01**](https://vtrd.in/system/ASTUDPHF.zip) by Oxford Computer Publishing'86, Phantom Family'96
* [**Art Studio v2.01**](https://vtrd.in/system/ASTUSMCG.zip) by Oxford Computer Publishing'86, Sinc Masters Creative Group'98
* [**Art Studio v2.01 (for AY-Mouse)**](https://vtrd.in/system/ARTSTCEG.zip) by Oxford Computer Publishing'86, Enigma Group'96
* [**Art Studio v2.01 (Rus)**](https://vtrd.in/system/ART_R_OI.zip) by Oxford Computer Publishing'86, O. Imanov'94
* [**Art Studio v2.01c**](https://vtrd.in/system/ART+MUS.zip) by Oxford Computer Publishing'86, Seamans
* [**Art Studio v2.3**](https://vtrd.in/system/ARTST_23.ZIP) by Oxford Computer Publishing'85, Viator'99
* [**Art Studio v3.0**](https://vtrd.in/system/AS128V30.zip) by Oxford Computer Publishing'86, Cybertronic Soft'97
* [**Art Studio v3.04D**](https://vtrd.in/system/ARTSKIDS.zip) by Oxford Computer Publishing'85, Kidson'94
* [**Art Studio v3.08**](https://vtrd.in/system/ARTSTINT.zip) by Oxford Computer Publishing'86, Intruders Corp'99

Art Studio, их уже как грязи, пока я пишу, архиспекуколог tiboh ещё пять версий найдет и выложит. Мне нравится такая, в которой можно рисовать мышкой и одновременно кнопками и которая не глючит и быстро работает. Я вложу образ с ней сразу в папку общую с программами для Zapilyator. Ссылка со страницы Zapilyator:   
<https://zxaaa.net/OTHER/zapilyator_programms.rar>

Должно появиться имя ARTStu чего-то там в имени и .B файл. Кнопкой F6 я его переименовываю в boot, чтобы образ автоматически запускался в эмуляторе, где я буду впоследствии рисовать.



Всё Ваш образ готов для работы. Для входа внутрь образа нажмите на его названии файла Enter. Можно в него таким же способом копировать по кнопке F5 (SPACE) – выделение файла, при этом около файла выделенного появляется желтая Галя. Ваши будущие картинки которые, например Вы нарисовали в Windows в программе PAINT, а затем сохранили как PNG и конвертировали в BMP2SCR в .scr, уже далее загоняете в образ простым переносом файла в Windows в папку на рабочем столе, где у Вас ESN, и далее запускайте ESN и там его увидите и скопируете F5 в образ с ART Studio, как в Total Commander или Norton Commander. А оттуда, уже работаете из под эмулятора Unreal в ART Studio. Да, кстати есть FAR Commander с плагинами копирования дискет ZX Spectrum и даже говорят Total Commander, но я так не умею. Делаю всё топором по старинке.

**лайFuck от ААА**, делайте имена файлов .scr своих картинок попроще, чтоб запоминать их и чтобы проще набирать их названия в ART Studio, помните наверное что меню набора имени в программе ART Studio глючит)) Это исторический факт. Смотрите пример названий картинок ААА, sneg, cwet, итп. Как для дауна. Зато понятно.



Нарисовали, сохранили так же как Вы это делали в 1991 году в самой программе, и потом нажимаете в эмуляторе UNREAL кнопку на клавиатуре Вашего ноутбука - **F9** – это перезаписать образ дискеты в эмуляторе. Всё. Ваша картинка с изменениями сохранена в Art Studio. Вы молодец! Если Вы не нажмете в эмуляторе после создания чего-либо и записи на дискету ZX Spectrum кнопку F9, Вам эмулятор разок напишет, что Вы выходите, а данные не сохранятся, надо выходить? Короче, не забывайте сохраняться, а то нарисуете, в ART Studio, запишите там и будете думать что эмулятор знает что Вы там в нём делали ))) Не забывайте что Вы делаете в эмуляторе, а не на реале и дисковод Ваш эмуляторный не в курсах о том что Вы есть на белом свете со своими Windows 10,11. Поэтому кнопкой **F9** по завершении всего Вы ему как бы говорите, запиши, то что я сделал. Спасибо большое. Теперь Вы можете всё и даже более, это не инструкция, а библия демостроителя!

# Не собирается демо, а только что собиралась.

Если демонстрация вдруг не собирается, а минуту назад собиралась. Проверьте пожалуйста ещё раз, а лучше заново загрузите в каждую ячейку сделанное: музыку, заставку, спрайты и картинку бакграунд. Если Вы хоть что-то где-то поменяли в процессе работы, хоть точечку, Zapilyator новый сохраненный файл даже под тем же именем что и было - уже не загрузит. Т.к. длина файла изменяется, и он автоматически тупит. Надо просто заново загрузить в ячейку то, что Вы меняли. При запарке в работе с ним часто это забываешь и сидишь думаешь, переживаешь, не понимаешь почему не компилируется. А я, например, поменял бакграунд и забыл об этом. В общем у **Zapilyator** есть такая функция при перемене чего либо, перезаписи в тот же файл, неважно, даже перемещении с папки, в папку, он уже не загружает файлы. Не переживайте, успокойтесь, и загрузите все заново. Будет работать!

# Пример сборки демонстрации Cats:

(папка cats)  
Запускаем **Zapilyator** файлом **win-server-start**

1. Кладем в ячейку: **Splash file** заставку демонстрации **zastavka.scr**

2. Выставляем **Border - WHITE**, т.к., рамка у заставки белая, то и border эстетичный белого цвета.

3. Выставляем параметр **Splash delay** - проигрывание музыки 3 куплета, до перехода в демонстрацию, потому что так под музыку переход. Это под первую мелодию расчёт на переход в демонстрацию.

4. **backgraund\_kotiki.scr** вставляем в **Background** – это наша внутренняя картинка демонастрации, заранее мной нарисованная в графическом редакторе **ZXSpeccy**.

5. **Animation Border** – **white** –снова белый бордер, т.к., картинка требует белого цвета.

6. **Ink** – **black** – мы его выставляем, потому что далее вставляем цветные картинки как .gif, если бы мы не использовали цветные картинки, которые перебираются поочередно, то можно все эти параметры не выставлять, т.к., цвет будет браться из картинки внутренней, которая у нас в демонстрации. Если мы не поставим правильные параметры цвета, то в нашем примере будет глюк в виде отсутствия цветов на картинке внутри демонстрации. Можете попробовать не выставлять.

7. **Paper** – выбираем белый цвет - **White**, потому что в демонстрации картинка белая, в цветном спрайте тоже соответственно дизайну белая. Описания причин описано выше.

8. **Bright** – выставляем не яркий, убираем галку. Потому что в демонстрации у нас картинка белая и не яркая, в цветном спрайте белая и не яркая. Описания причин описано выше. Это мы выставляем только потому что мы используем цветной спрайт в качестве .gif

9.**Analyzer** – анализатор, мы будем использовать, вправо длинный **Goba Right** с параметрами 1 к краю картинки и 12 по вертикали вниз, того требует дизайн картинки. То есть, так я задумал место под скролл.

10. **Animation 1 (main/Slow).** Это медленная ячейка памяти вывода спрайтов. Вставляем спрайт: **KAKOS\_10screen\_Banka1**.gif. Мы собираем демонстрацию 2 трудным способом, первая .gif у нас подготовлена в специализированной программе **ZX Animator (KAKOSA).** Пример того как в ней работать я описал в главе:Цветной спрайт. Эффект двойного экрана. Поэтому мы ничего не выставляем в меню **Speed**. Если бы мы брали 1 способ, это когда мы кладем картинки .scr в папку, которую вставляем как .zip, тогда надо выставить скорость прокрутки эти картинок от быстрой до медленной (255). Для примера я приложил пак c zip картинками, 1 способом. Потом соберите вторым способом: через скорость спрайта. Файл для второго примера: **1banka\_kotiki\_cwet.zip**, так и вставляете его в эту же банку один, тогда спрайт Какоса не надо трогать.

11. **Animation 2 (main/Slow).** Это ещё одна медленная ячейка памяти вывода спрайтов. Вставляем простой спрайт 1 знакоместо видимого показа, это типа скролл с приветами, который я нарисовал. Вставляем его: **hello\_2banka.gif** (прозрачность не убрал, потому что мне лень. Он вроде бы с ней не глючит и ладно. Раньше я наоборот специально делал прозрачность.

12. **Animation 3 (int/Fast).** Сюда вставляем: **banka3\_golova.gif**, память быстрая, кошка большая, фаз много.

13. **Animation 4 (int/Fast).** Сюда вставляем: **kotyta\_banka4.gif**, память быстрая, котятки много требуют INT сигнала, то есть скорости процессора для обработки всех данных.

14. **SlideShow (ZIP of compressed images)** – ничего не вставляем, функция пока не доступна и демонстрация того не требует. Для чего это меню я описал в соответствующей главе книги.

15. **MusicBank (ZIP of PT2/PT3 files)** – сюда мы вставляем папку **3music.zip** с музыкой, всего в этот проект влезло 3 мелодии. Их я заранее прогнал через Vortex Tracker и привёл к формату .pt3. Назвал 1,2,3, чтобы Zapilyator понимал какую мелодию за какой играть (очередность). Их можно в демонстрации вызвать кнопками 1,2,3, так же они играют авто проигрыванием.

16. **Scroll Setup** в меню Text пишем текст:   
КОТЯТА ВЫ ПОСЛУШАЙТЕ, РЕБЯТА, Я ХОЧУ ВАМ РАССКАЗАТЬ! РОДИЛИСЬ У НАС КОТЯТА - ИХ ПО СЧЕТУ РОВНО ПЯТЬ. МЫ РЕШАЛИ, МЫ ГАДАЛИ, КАК ЖЕ НАМ КОТЯТ НАЗВАТЬ? НАКОНЕЦ МЫ ИХ НАЗВАЛИ: РАЗ, ДВА, ТРИ, ЧЕТЫРЕ, ПЯТЬ! РАЗ - КОТЕНОК САМЫЙ БЕЛЫЙ, ДВА - КОТЕНОК САМЫЙ СМЕЛЫЙ, ТРИ - КОТЕНОК САМЫЙ УМНЫЙ, А ЧЕТЫРЕ - САМЫЙ ШУМНЫЙ. ПЯТЬ - ПОХОЖ НА ТРИ И ДВА - ТОТ ЖЕ ХВОСТ И ГОЛОВА, ТО ЖЕ ПЯТНЫШКО НА СПИНКЕ, ТАК ЖЕ СПИТ ВЕСЬ ДЕНЬ В КОРЗИНКЕ. ХОРОШИ У НАС КОТЯТА - РАЗ, ДВА, ТРИ, ЧЕТЫРЕ, ПЯТЬ! ЗАХОДИТЕ К НАМ, РЕБЯТА, ПОСМОТРЕТЬ И ПОСЧИТАТЬ. СЕРГЕЙ МИХАЛКОВ. КИСКИНО ГОРЕ ПЛАЧЕТ КИСКА В КОРИДОРЕ, У НЕЕ - БОЛЬШОЕ ГОРЕ: ЗЛЫЕ ЛЮДИ БЕДНОЙ КИСКЕ НЕ ДАЮТ УКРАСТЬ СОСИСКИ! БОРИС ЗАХОДЕР. КОТЕНОК КОТЕНОК ВОЗИТСЯ С КЛУБКОМ: ТО ПОДПОЛЗЁТ К НЕМУ ТАЙКОМ, ТО НА КЛУБОК НАЧНЁТ КИДАТЬСЯ, ТОЛКНЕТ ЕГО, ОТПРЫГНЕТ ВБОК... HИКАК НЕ МОЖЕТ ДОГАДАТЬСЯ, ЧТО ЗДЕСЬ НЕ МЫШКА, А КЛУБОК. АГНИЯ БАРТО. ВЫШЕЛ КОТИК ПОГУЛЯТЬ РАЗ, ДВА, ТРИ, ЧЕТЫРЕ, ПЯТЬ - ВЫШЕЛ КОТИК ПОГУЛЯТЬ! ПОТЯНУЛСЯ, ОГЛЯДЕЛСЯ, СЕЛ, НА СОЛНЫШКЕ ПОГРЕЛСЯ. ПОДРЕМАЛ НА ТРАВКЕ КОТИК, ПРОБУДИЛСЯ, ВЫМЫЛ РОТИК, НЕ СПЕША ДОМОЙ ВЕРНУЛСЯ, НАД ТАРЕЛКОЙ ОБЛИЗНУЛСЯ, СЪЕЛ КОТЛЕТКУ, И ОПЯТЬ ВЫШЕЛ КОТИК ПОГУЛЯТЬ. СВЕТЛАНА ПШЕНИЧНЫХ. КОТЕНОК ЕСЛИ КТО-ТО С МЕСТА СДВИНЕТСЯ, НА НЕГО КОТЕНОК КИНЕТСЯ. ЕСЛИ ЧТО - НИ БУДЬ ПОКАТИТСЯ, ЗА НЕГО КОТЕНОК СХВАТИТСЯ. ПРЫГ - СКОК! ЦАП - ЦАРАП! НЕ УЙДЕШЬ ОТ ЦЕПКИХ ЛАП! ВАЛЕНТИН БЕРЕСТОВ. ВЫШЕЛ КОТИК ПОГУЛЯТЬ РАЗ, ДВА, ТРИ, ЧЕТЫРЕ, ПЯТЬ - ВЫШЕЛ КОТИК ПОГУЛЯТЬ! ПОТЯНУЛСЯ, ОГЛЯДЕЛСЯ, СЕЛ, НА СОЛНЫШКЕ ПОГРЕЛСЯ. ПОДРЕМАЛ НА ТРАВКЕ КОТИК, ПРОБУДИЛСЯ, ВЫМЫЛ РОТИК, НЕ СПЕША ДОМОЙ ВЕРНУЛСЯ, НАД ТАРЕЛКОЙ ОБЛИЗНУЛСЯ, СЪЕЛ КОТЛЕТКУ, И ОПЯТЬ ВЫШЕЛ КОТИК ПОГУЛЯТЬ. СВЕТЛАНА ПШЕНИЧНЫХ. ВЕЖЛИВЫЙ ХВОСТ ВOCПИТАННЫЙ ХВОСТИК У КОШКИ МОЕЙ: ОН ВСЮДУ ПОЧТИТЕЛЬНО ХОДИТ ЗА НЕЙ. И В ДВЕРИ ОН ПРЕЖДЕ НЕЕ НЕ ВОЙДЁТ - ОН ВЕЖЛИВО КОШКУ ПРОПУСТИТ ВПЕРЕД. ВЛАДИМИР СТЕПАНОВ. КОТЕНОК НА ДИЕТЕ БРОДИЛ КОТЕНОК ВОЗЛЕ МИСКИ И, ПРИОТКРЫВ ОДИН ГЛАЗОК, С ВОЛНЕНЬЕМ ГЛЯНУЛ НА СОСИСКИ И ЯЗЫЧКОМ ЛИЗНУЛ РАЗОК. - ВРАЧИ ПРИДУМАЛИ ДИЕТЫ, ПРОДУКТАМ ЧТОБЫ ПРОПАДАТЬ, А САМИ КУШАЮТ КОТЛЕТЫ, - ОН ПРОВОРЧАЛ И ЛЕГ В КРОВАТЬ. ПРИЯТНЫЙ ЗАПАХ ШЕЛ ИЗ МИСКИ, КОТЕНОК МУЧИЛСЯ, ТЕРПЕЛ. И АППЕТИТНЫЕ СОСИСКИ ЧЕРЕЗ МИНУТУ ВСТАЛ И СЪЕЛ. ТАМАРА ЛОГАЧЕВА. Дема переписана заново в новой версии запилятора, она улучшена по графике, добавлена музыка. Переписана А.А.А. в 2025 году в декабре.  
  
17. Выбираем **8px height** – скролл 8 точек в высоту..

18. **Scroll direction** – выбираем в левую сторону.

19. **Шрифт** по умолчанию, что предложил Zapilyator, про выбор шрифта я написал в соответствующей главе книги.

20. Цвета скролла **INK, PAPER, BRIGHT** – не указываем, т.к., у нас будет браться с внутренней картинки этот цвет.

21. **Scroll width** ДЛИННА СКРОЛЛА - **18** знакомест, это я рассчитал по сетке сам

22. **Horizontal offset** - отступ от левого края картинки - **12** знакомест

23. **Position** – высота по вертикали картинки - **20** знакомест.

24. Создать моноблок галку ставим, чтобы был один файл сжатый на дискете.

25. **Счетчик прерываний – не ставим галку, нам это не надо. Это для отладки демо, а не для финальной сборки.**

Нажали **собрать**: ждем сборки, потом жмём **Download**, и в меню загрузки появится файл в нём будет демонстрации и её финальная сборка. По итогам сборки программа выдаст чек, в котором вс что произошло расписано.

Result

Loading data...

MusicBank parsed and saved!

Starting animation parsing...

---

Parsed: **0 - 9** (10 total).

Done!

Animation size: **7 079** bytes

Bytes affected: **4 527** bytes

Data ratio: **1.56** bytes

---

Parsing animation 2...

Parsed: **0 - 29** (30 total).

Done!

Animation size: **1 505** bytes

Bytes affected: **760** bytes

Data ratio: **1.98** bytes

---

Parsing animation 3...

Parsed: **0 - 95** (96 total).

Done!

Animation size: **36 264** bytes

Bytes affected: **10 595** bytes

Data ratio: **3.42** bytes

---

Parsing animation 4...

Parsed: **0 - 23** (24 total).

Done!

Animation size: **43 350** bytes

Bytes affected: **13 792** bytes

Data ratio: **3.14** bytes

---

Generating sources...

Free space: **8.40** kb (**8 604** bytes)  
page 0: 8 528 байт  
page 1: 1 байт  
page 3: 25 байт  
page 4: 21 байт  
page 6: 24 байт  
page 7: 5 байт

Success!

---

Compiling sources...

First compilation:

SjASMPlus Z80 Cross-Assembler v1.18.3 (https://github.com/z00m128/sjasmplus)

Pass 1 complete (0 errors)

Pass 2 complete (0 errors)

include data: name=res/main\_bg.pck (539 bytes) Offset=0 Len=539

include data: name=res/scroll (2349 bytes) Offset=0 Len=2349

include data: name=res/8x8font (769 bytes) Offset=0 Len=769

include data: name=res/music1.pt3 (7451 bytes) Offset=0 Len=7451

include data: name=res/music2.pt3 (7059 bytes) Offset=0 Len=7059

include data: name=res/music3.pt3 (2753 bytes) Offset=0 Len=2753

> Page 0 free: 8528

> Page 1 free: 1

> Page 3 free: 25

> Page 4 free: 21

> Page 6 free: 24

> Page 7 free: 5

include data: name=dehrust.bin (256 bytes) Offset=0 Len=256

include data: name=res/splash\_bg.pck (689 bytes) Offset=0 Len=689

Pass 3 complete

Errors: 0, warnings: 0, compiled: 45860 lines, work time: 0.156 seconds

Packed page0.c: 32431 -> 14973 bytes (53.8% compression)

Packed page1.c: 16382 -> 9983 bytes (39.1% compression)

Packed page3.c: 16358 -> 10835 bytes (33.8% compression)

Packed page4.c: 16362 -> 9661 bytes (41% compression)

Packed page6.c: 16359 -> 1776 bytes (89.1% compression)

Packed page7.c: 16378 -> 2592 bytes (84.2% compression)

Replaced 6 savetrd commands total

Included 6 packed files in TRD

SjASMPlus Z80 Cross-Assembler v1.18.3 (https://github.com/z00m128/sjasmplus)

Pass 1 complete (0 errors)

Pass 2 complete (0 errors)

include data: name=res/main\_bg.pck (539 bytes) Offset=0 Len=539

include data: name=res/scroll (2349 bytes) Offset=0 Len=2349

include data: name=res/8x8font (769 bytes) Offset=0 Len=769

include data: name=res/music1.pt3 (7451 bytes) Offset=0 Len=7451

include data: name=res/music2.pt3 (7059 bytes) Offset=0 Len=7059

include data: name=res/music3.pt3 (2753 bytes) Offset=0 Len=2753

> Page 0 free: 8528

> Page 1 free: 1

> Page 3 free: 25

> Page 4 free: 21

> Page 6 free: 24

> Page 7 free: 5

include data: name=page0p.c (14973 bytes) Offset=0 Len=14973

include data: name=page1p.c (9983 bytes) Offset=0 Len=9983

include data: name=page3p.c (10835 bytes) Offset=0 Len=10835

include data: name=page4p.c (9661 bytes) Offset=0 Len=9661

include data: name=page6p.c (1776 bytes) Offset=0 Len=1776

include data: name=page7p.c (2592 bytes) Offset=0 Len=2592

include data: name=dehrust.bin (256 bytes) Offset=0 Len=256

include data: name=res/splash\_bg.pck (689 bytes) Offset=0 Len=689

Pass 3 complete

Errors: 0, warnings: 0, compiled: 45898 lines, work time: 0.156 seconds

Total compression: 114270 -> 49820 bytes (56.4% saved)

TRD merge config found: [{"files":[{"name":"boot.B","start":23552},{"name":"splash.C","start":24576},{"name":"0.C","start":24576},{"name":"1.C","start":49152},{"name":"3.C","start":49152},{"name":"4.C","start":49152},{"name":"6.C","start":49152},{"name":"7.C","start":49152}],"output":"boot.B","address":23552,"remove\_original":true}]

Attempting to merge 8 files...

TRD files merged: 8 files -> boot.B

Removing 7 original files: splash.C, 0.C, 1.C, 3.C, 4.C, 6.C, 7.C (removed 7 original files)

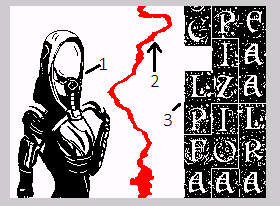
Code compilation success

На этом моя краткая инструкция на несколько листов закончена.   
Всегда Ваш А.А.А. 2025 год. Как говорится инструкции писать, не мешки второчить.  
**Пожертвования тут:**  
<https://www.tinkoff.ru/rm/r_lBRrykOmfd.DUMScRuHlg/v2vfJ54782>  
Это моя карточка Тбанк. Переводите деньги, если они у Вас много, а я их переведу Активатору у него их мало, даже скорее нет. Всех с Новым 2026 годом, счастья, радости, детишек побольше и многодемия. Готовые демонстрации, которые Вы сделаете, присылайте мне в почту с именем письма DEMO для ZXAAA.  
aaa@zxaaa.net

# Часто совершаемые ошибки при работе с Zapilyator.

Эта статья была написана мной для Zapilyator 2 в журнал «За Рулём», однако спустя 10 лет она не утратила своей обучающей сущности. Конечно в 2026 году я немного ее поправил для Zapilyator 3. Итак приступим изучать секреты демо строительства. Годы работы с профессиональным инструментом Zapilyator 2 научили меня некоторым хитростям, которые помогают выпускать за сутки 2-3 части мегадемо или 5 простых демонстраций. Начнём пожалуй по порядку. Для примера возьмём весьма неудачную демонстрацию от группы NOT Soft:

https://zxaaa.net/view\_demo.php?id=9014  
  
**Lyric Demo**

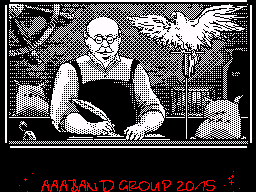
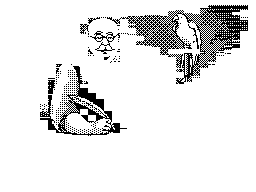


Как мы видим и слышим музыка – ужасно тормозится, спрайт тетриса кривой и дерганный. Есть глюки в красной полоске - молнии. Тут явно что-то не так! Автор пьян... Я попросил автора демонстрации прислать мне исходники спрайтов. Итак, что же мы видим внутри организма демонстрации?

**Рука, фон...**

Спрайт руки кидается 4 экранами спектрума (scr), причем женщина - мать не стёрта, а кидается то же вместе с рукой. Автор видимо не в курсе, что женщина-картинка занимает 1/3 экрана и получается, что она ест 4 места картинки 6912 байт. К тому же автор отдельно от этого кидает Background экрана 6912 этой же женщины. Удивителен мир приматов, но наши спектрумовские это самые первые на земле индейцы!   
  
 Исправляем этот недостаток. Тут два пути либо это сделать автоматически используя программу Какоса "Ficha" (программа есть по ссылке и внутри этой инструкции)

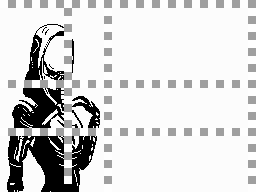
<https://zxaaa.net/OTHER/zapilyator_programms.rar>  
  
 Для работы с фичей все картинки-фазы сначала переводятся в .scr экраны и именуются 1,2,3.. Можно сделать это самым быстрым способом, сначала загружаев GiF Annimator, в него загружаем спрайт, и затем сохраняем как картинки в папку. В эту же папку кидаем программу BMP2SCR, и по одной картинке переводим в формат scr. Затем эти готовые .scr (фазы гифки) переименованные как 1,2,3,4…. и так далее, кладутся в ту же папку где находится программа ficha. Нажимается пуск фичи (в ней указываем сколько фаз спрайта обрабатываем) и указывается путь к 1 картинке, затем нажимается старт. Прямо при Вас сделаются картинки фаз спрайта в формате .bmp. Вы это даже немного увидите. А так же отделяется спрайт движения от самого недвижимого бакграунда. (создается картинка фона под названием zadnik). Для трудных спрайтов это единственный способ изготовления подобных фаз спрайтов. Например дема Мопас делалась фичей. Моя дема KEY DEMO, часть с пиратом и попугаем делалась фичей! Это наисложнейшая работа, которую нельзя сделать руками. Просто надо быть невероятным гейм див художником анимации по маске. Например, в части с попугаем было 130 полноэкранных фаз спрайта. И выделить их руками не смог бы даже piesu )) Но ААА смог фичей какоса!

Фон деда пишущего пером и попугаем выделенный фичей спрайт

А я ручками хАААчу!

В нашем случае проще работать руками "на глаз" потому что фаз не много. Я тоже, чаще всё делаю тупа, как буратино – руками. Тем паче у нас всего 4 фазы картинки и в принципе это не трудно. Сначала я должен сделать разметку с помощью сетки:

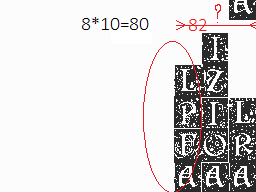
Разметка сделана. На глаз определена граница движения руки, теперь просто обрезаю сам экран от руки по границам знакомест. То есть кратное 8. Главное, чтобы спрайт совпадал по отмеченным границам знакомест. В результате мы получаем 4 маленькие ручки, которые влезут в первую банку памяти и не будут глючить. Это хорошо! Потому что нам понадобится много места чтобы сделать эту демонстрацию. Одновременно отрезаю лишнюю руку от картинки - фона женщины. На экране она нам не нужна, потому что выводится спрайтами, а лишние байты памяти не помешают!



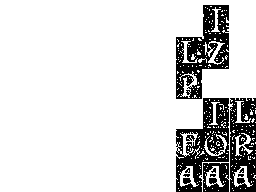
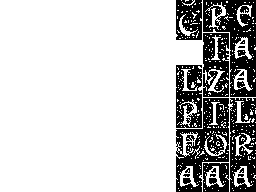
Эту картинку переводим в спектрумовский формат .scr и загоняем в Art Studio. Где нам предстоит фон раскрасить под знакоместа спрайтов молнии и тетриса. Это как правило делается после черновой сборки всей демонстрации. Или в тот момент, когда нам известны размеры спрайтов и мы понимаем места их расположения. Сейчас я понимаю, что тетрис 80 точек в ширину, а молнию не понимаю, надо примерять на месте...

**Спрайт «Тетрис».**

Теперь переходим к спрайту тетриса и ужасаемся авторской работе! Перед нами предстаёт 45 кадров, имеющих в каждом кадре глюки. Более того в ширину он 82 точки и это означает рисовать всё-всё заново руками!



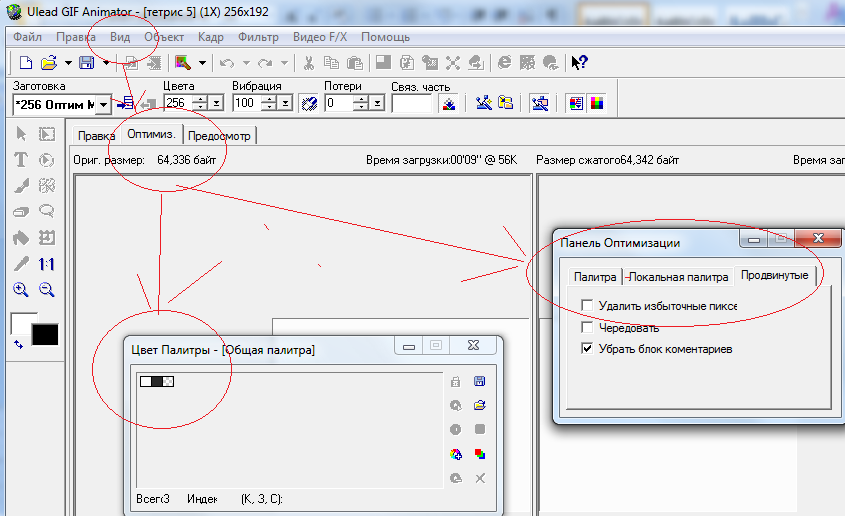
Помимо глюков по ширине, мы имеем, неровное сложение, неровный край. Неправильное закрепление цвета. На спектруме допускается только черный и белый. Соответственно белый это 0 и черный это 255. Тут есть даже серый оттенок и чего только нет. Спрашивается, как это поймет запилятор при компресии, он же не человек и у него нет допуска «на Емелю»? Да никак ребята! Это глюки, сбросы, зависоны, нехватка памяти, да что угодно! Я переделал спрайт руками каждую фазу, сделав его в 80 точек шириной. Именно 80 точек это ровно 10 знакомест. Нельзя в запиляторе спрайты соединять неровным количеством пикселей 82 или 84... Лучше всего ориентироваться на знакоместа по сетке спектрума 256 на 192. И делать их кратным 8. Ведь одно знакоместо это 8 на 8 точек.

Теперь спрайт тетриса готов, а ААА потерял фактически час жизни. Спрайт тетриса огромный, ему нужна только 3-4 банка памяти, а он ест много памяти. Загнав спрайт руки, музыку, заставку и 2 картинки .scr фона у меня осталось всего 36 килобайт на молнию. Ужас, как с этим жить!

**Спрайт «Молния».**

Раскрыв молнию ужаснулся ещё больше. У автора кидается 50 экранов .scr формата! Уже хотел бросить это дело... Ведь чтобы перевести 50 экранов спектрума в 50 PNG экранов необходимо каждый из них ткнуть рукой и дважды указать путь в программе bmp2scr. Я потерял еще 30 минут времени. Потом закатал молнию в спрайт и прогнал в **GifAnnimator**, удалив лишние точки в меню **Оптимизация**. Я напишу это дважды! Я прогнал спрайт в **GifAnnimator, после чего удалил лишние точки в меню оптимизация**! Это самое важное и самая распространенная ошибка всех запильщиков на этой планете, кроме ААА. Вот пример того что надо сделать, на рисунке показано где надо убрать галку, каждый раз когда вы сохраняете спрайт для запилятора, даже если вы делали это до этого момента, всё равно делайте. При повторной загрузке спрайта под правку снова и снова! После реоптимизатора Nyka то же делайте!



Только убрав галку вы гарантируете, что спрайт не будет глючить и в памяти он будет занимать столько места, сколько весит, а не с учетом Ваших глюков. к тому же в меню **Цвет палитры** следите за тем, чтобы у Вас был цвет черный и белый. Причем черный 255 и белый 0, без оттенков.

**Теперь пробую впервые запилить демонстрацию по параметрам:**

Музыка Fatal Snipe 1,7 килобайт

Картинка заставка 6912 байт

1 банка медленной памяти: две картинки scr фона с разными знакоместами цвета в zip архиве;

2 банка медленной памяти: спрайт руки (до 2на 2 знакоместа) глючить не будет!;

3 банка быстрой памяти - спрайт тетриса;

4 банка быстрой памяти - спрайт молнии...

**Дехайпинг...**

Не хватает памяти. С болью в сердце сокращаю спрайт молнии в минус 25 фаз, то есть сокращаю ровно в два раза от 50 фаз, увеличив скорость прокрутки, для того чтобы видимая разница была минимальна. Снова закатываю в банки по тем же параметрам и получаю в остатке 3 килобайта. Дема готова, всё летает и не глючит:

<https://zxaaa.net/view_demo.php?id=9017>



Спрайт тетриса работает как часы, спрайт молнии стал еще лучше, за счет того что я меняю цвета, кидая в zip архиве две картинки фона. Для справки 1,2,3 и даже 4 картинки scr в zip архиве как спрайт, занимают меньше места чем 1 картинка подсоединяемая как фон демы в меню backgraund. Это секрет запилятора, пользуйтесь им для экономии памяти. Если не хватает 5-6 килобайт, кидайте фон в банку №1,2,3 или 4 в zip архиве. Экономия ощутима. Там же можно менять скорость прокрута картинок и добиваться нужного цвета мультиколора. Можно сжимать даже одну .scr картинку в zip архив и кидать её как спрайт в первую банку. Ваш фон будет занимать 1 килобайт, а не 6.

**Как ещё экономить память в Zapilyatior ?**

1. экспериментируйте с банками памяти, меняйте местами спрайты, даже если Вам сначала кажется что так не заработает, пробуйте, нарабатывайте практику. Проверяйте дему в спекакуляторе. У Nyuk есть один глюк!!! При первом запуске демы в Унреал срабатывает Турбо режим и все что вы сделали видите в режиме 7 мегагерц. соответственно не видны ваши глюки в спрайтах. И мультиколор отображается не так как в реале. Обязательно запускайте после сжатия демы в trd образ в эмуляторе спекакулятор!!! Три раза говорю, в спекакулятор запускайте! Там сразу видно как работают Ваши спрайты и какого цвета мультиколор. Не забываем режим эмулятора noflic, это когда визуально эмулятор показывает очень быстро смену картинок. Смотрим дему 4yk4a там как раз в режиме мультиколора надо смотреть, то есть поставив галку noflic в эмуляторе UNREAL, в других эмуляторах нет такой функции.

2. Иногда для экономии памяти можно отрезать кусочки - вступления мелодий. Экономит немного, но все же. Резать в Vortex Tracker. выделяете что отрезаете, правая кнопка мыши и delete. Для постановки первого куплета мелодии надо нажать L, на этот L куплет будет закругление проигрывания мелодии.

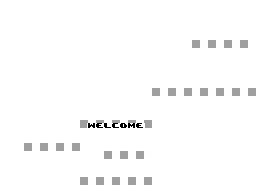
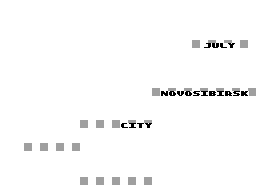
3. Не забывайте удалять лишние точки в меню оптимизация в GifAnimator

4. Убирайте заставку, она много ест памяти, фактически 6912, т.к. не сжимается Нюком. И потом запуск через заставку уже утомил.

5. Скролл запилятора (его текст) ест беспредельно много памяти. Проще нарисовать его фазами в 1 знакоместо. При этом текст можно писать бесконечно. Все подсовывается в 1,2 банки памяти и летает. Я использую шрифты Art Studio,

Они есть в запиляторе 3 или же их можно брать стандартные из ART Studio и переводить в запилятор3.

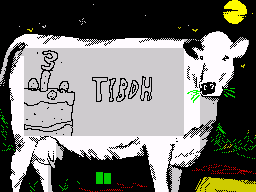
пишу там необходимые фразы, а затем по специальной сетке накладываю слова. Сохраняя фазы - фразы. Потом в гиф анниматоре удаляю, вернее меняю серый цвет на белый и остается только белый фон и черные буквы. Сетка наносится на экран бакраунд со всем рисунком, позволяет точно определять места где скролл не будет пересекаться с спрайтами эффектами и картиночным рисунком. Эта хитрость позволяет писать скроллы в 100-200 фаз всего за 1 час вместо потери на это суток.

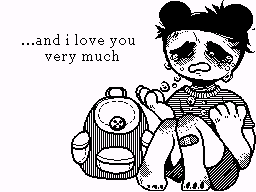
 

6. Меняйте черный с белым в спрайте. Иногда в Вашем спрайте черного больше чем белого и лучше кинуть инвертированный спрайт изменив цвет фона. Тут то же есть секрет работы повышающий скорость обработки спрайтов по инвертации цвета. Это возможность изменения цвета есть в гиф аниматоре. Например надо поменять черный с белым. Фаз аж 300 штук... Руками вы это будете делать сутки. В гиф аниматоре войдите в меню Оптимизация - Цвета. Там на черном цвете нажмите Enter и поставьте, например красный (не важно какой). Теперь можете белый сделать черным, он не сольется с черным фоном, а затем верните красный на белый. Вот и всё. 30 секунд и спрайт инвертирован. Иногда инвертация спрайта это вообще единственный способ его вставить в демо. Экономия поразительная, до половины памяти от начального варианта. Определяется на глаз. Если первоначально черного больше, меняйте его на белый.

7. В спрайтах с огромным количеством фаз, особенно «АНИМЕ» направленности, где мелкие чёрные рисунки перемежаются со взрывами и вспышками, с большим количеством чёрного, можно применять «через-строчность», то есть из этого:  
  
  
делать это:  
  
Казалось бы, что я тут сэкономил? Но на самом деле почти килобайт!!! Запилятор любит через строчку и не любит полностью залитое черным, особенно, когда все остальные фазы спрайта преимущественно белые, как это:

   
  
Этот небольшой трюк, однажды позволил мне запилить этот гифт:  
<https://zxaaa.net/view_demo.php?id=15960>

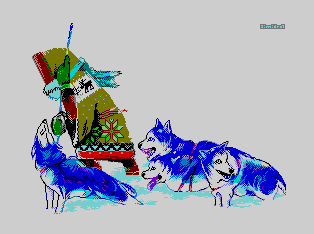
  
Tiboh Gift.  
В общем экспериментируйте и не плачьте, что памяти не хватает…



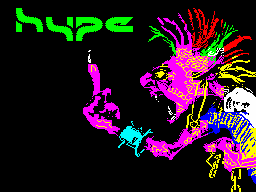
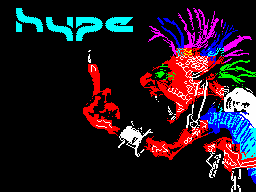
# Создание мерцающих картинок в запиляторе.

Сразу оговорюсь, что никакого мультиколора в его высоконаучном «алониевом» понимании в программе запилятор - не существует. Поэтому умники и умницы, могут пройти мимо этой статьи, со словами, что запилятор это антинаучно. Зато у нас есть честная поочередная смена цветных атрибутов внутренних картинок – «Background». Причём это может быть 2-3 и даже более смен картинок с различными цветами атрибутов. При этом картинки хранятся в виде **zip** архива. Располагаются в одной из 4 банок памяти, желательно в 3 или 4 быстрой памяти запилятора. Чем меньше знакомест красим, тем быстрее скорость вывода и равномерней цвет.

Сделать полноценный 256x192 точки экран мультиколором, вернее двойным экраном не получиться. У меня не хватило тактов сделать картинку чукчи Dimidrola. После компиляции картинка глючила, цвета не успевали сменяться. Хотя, казалось бы - тут свободно половина экрана! Связано это ещё и с изменениями пикселей. Если пиксели не меняются, то есть две картинки идентичные и меняются только атрибуты, то можно вместить гораздо больше смены или мерцания цвета.



Для того чтобы сделать «псевдо-мультиколор» мы должны в первую очередь собрать саму демонстрацию на одном экране полностью под «ключ». Разложить все спрайтовые эффекты в нужных местах. То есть сделать полностью демонстрацию и определиться с цветами на экране и проверить работоспособность самой программы. И только потом мы снова берем наш экран Background и меняем цвета по вкусу. Я делаю это в программе Art Studio, открыв её дважды на экране. То есть у меня на рабочем столе располагаются перед глазами сразу две картинки. Одна слева, другая справа. Работают два эмулятора Unreal. Таким образом, я сразу вижу, что менять на картинках.

1.scr 2.scr

На выходе у нас получается две «Background» картинки с разными цветами. Эти картинки мы сохраняем в виде двух отдельных .scr и вместе архивируем в zip архив. Экраны для удобства лучше назвать именами 1, 2. В дальнейшем мы присоединяем наш zip архив с этими картинками в 3 или 4 банку спрайтов запилятора. Первоначальный «Background», который ранее присоединяли в верхнее меню запилятора «Background» мы убираем насовсем. При этом! И это очень приятно, две картинки в zip архиве почему то займут меньше места в памяти чем одна картинка вставленная в меню Background. Конечно, мы немного памяти высвобождаем, но всё равно это приятно. Теперь запускаем команду **Make**, и радуемся мультиколорной демонстрации ))

**Каверзные, тупые вопросы:**

***- Можно ли присоединять zip архив в 1 или 2 банки медленной памяти запилятора***?

В принципе можно, но надо смотреть на окончательный вариант работы после сборки и компиляции. Скорость смены атрибутов будет чуть медленней, хотя иногда это смотрится красивее. Не забываем при этом, что первый запуск программы после команды **MAKE** проходит в турбо режиме эмулятора Unreal и для более точного определения результата, запускайте программу отдельно в эмуляторе Spectaculator.

***- Можно ли использовать больше экранов со сменой цветов?***

Да можно. Оптимально 2-4. Чем дальше в лес, тем не понятней эффект смены атрибутов. Две картинки дают эффект слабой размывки цвета, можно сказать мы получаем новый цвет на спектруме, он будет более блеклый или смешается в соответствии с правилами смеси цветов. Другие варианты заставляют цвета мерцать и от большего количества экранов мерцание только усиливается.

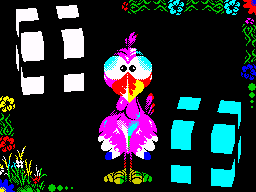
***- Возможно каким либо другим способом добиваться того же результата для получения цвета?***

Да возможно, есть второй способ через программу Какоса Наноса, которая называется **Cwetka**. Я даже не буду пытаться объяснить, как это трудно. Это вообще за гранью понимания всего человеческого. Представьте картинки которые рисует Какос, так вот это еще хуже. Хотя все свои старые мегадемо с мерцающими цветами я делал через неё. Но человек ко всему привыкает… Даже к Какосу))

**Наиболее интересные примеры использования цвета в готовых продуктах от ААА:**

**Deluxe**

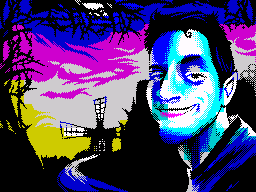
<https://zxaaa.net/view_demo.php?id=8762>



Перекрасил пространство под кубики, птицу, цветы. Приятно получилось, всего 2 мя экранами.

**Exocube**

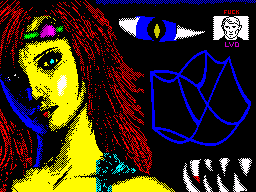
[**https://zxaaa.net/view\_demo.php?id=8659**](https://zxaaa.net/view_demo.php?id=8659)



Раскрасил всю картинку, сделал мягкий голубой цвет кубиков.

**Hype Fuck Magnetik**

[**https://zxaaa.net/view\_demo.php?id=8657**](https://zxaaa.net/view_demo.php?id=8657)



Сделал на каждую 4 фазу кубика 1 фазу цвета (всего 4 картинки цвета), появился эффект тени с цветом на линиях кубика. Игрался с настройками скорости спрайта цвета и скорости кубика. Удалось найти такие настройки, когда на каждую фазу кубика выскакивал своя фаза цвета. Отчего и появился радужный эффект за выводом спрайта. В принципе случайно вышло.

# Использование ONLINE комплекса EZGIF

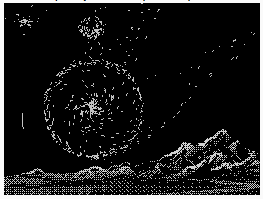
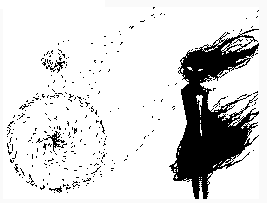
<https://ezgif.com/>

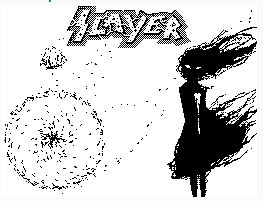
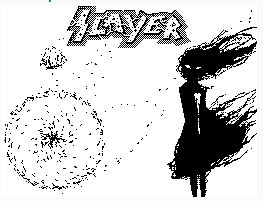
Вы думаете это очередная онлайн программа сделанная на коленке? Нет, ребята, Вы ошибаетесь. Это мощнейший инструмент. И он сравним с Photoshop, только понятный и удобный, и даже пока бесплатный. Познать его методом «тыка» можно. ИИ по нему проинструктирован и поможет в любом вопросе. Задаете вопрос как мне сделать (что то в ezgif). Получаете пошаговый ответ. Я лишь дам необходимые азы.   
  
 Итак, этой программой можно одним кликом делать прозрачный gif, то есть убрать фоновый цвет. Всего в наших gif два цвета это черный – рабочий и белый это фон.

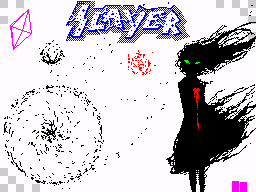
<https://ezgif.com/effects>

Загружаем не прозрачный gif с белым фоном, затем в меню:   
**Replace color with transparency: ставим галку на белом цвете. Далее жмем Apply Selected и забираем готовый прозрачный гифт. Для чего мы его сделали?**Изготовление демонстрации без соблюдения границ 8 точек и знакоместа в программе EZGIF.  
  
**Первый gif в который мы вставляем другие гифты обязательно должен быть прозрачным. Желательно сначала сделать больший гифт в размере поля спектрума 256 на 192 в прозрачном виде. А затем на него выкладывать меньшие гифты (прозрачные) или изображения графики с прозрачным фоном.  
Для соединения гифта в гифт существует инструмент:**

<https://ezgif.com/add-image>

 **Берем гифт. Делаем его прозрачным. Меняем белый на черный. Я делаю это в Gif Annimatore, способ описывал выше, но можно и в ezgif там есть такая возможность замены цвета.  
  
Получаем вот такой шаблон:** **Теперь на этом квадратике можно выкладывать (накладывать) другие спрайты, это могут быть спрайты или рисунки переведенные в gif и так же сделанные прозрачными. Главное чтоб размер вставляемого был менее чем тот в который вставляете. А так же естественно должно быть одинаковое количество фаз спрайта и скорость. Если Вам надо разную, значит делаете два спрайта в разные банки. Например, давайте вставим название:** **название**

**Получаем готовый результат**Мышкой можно перетаскивать вставляемый гифт (картинку) в любую точку экрана 256 на 192. Так же можно вставлять и гифты в гифты. Главное условие скорость и количество фаз должно быть одно. И прозрачные обязательно. А когда Вы будете вставлять готовое в демо, там уже прозрачность убирайте. Запилятор как то плохо к прозрачности относится. Этот способ не требует соблюдения четкого размера 8 на 8 точек, потому что мы вставляем в больший размер где 8 на 8 соблюдено. А вставляемый нами прозрачный гифт, то есть у него нет границ. Это намного удобнее чем я описывал ручной способ выше с соблюдением границ 8 на 8. Ограничение только в одном, у спектрума всего 3,5 мегагерца, нужно помнить, что заполнить весь экран движением не выйдет, будет тормозиться демонстрация, чаще всего это выражается в неправильном проигрывании AY музыки, как будто бы инструменты начинают скрипеть. Пропадает басовитость мелодии. Но это все в процессе работы познается. Нужно искать четкий баланс, естественно в ущерб музыке не делаеться музыкальная демонстрация. Музыка долна играть как у автора, а то что происходит на экране Вами оптимизируется. В принципе этой онлайн программой можно делать свои интересные гифты, которые затем надо распределять по банкам памяти в запиляторе.   
  
  
Если в примере у меня 12 фаз и в каждой девочка картинка, на которой нет движения, тогда Вы ее можете оптимизировать в программе Kakos Fisha, ее я приложил в архив. Все 12 фаз Вы скидываете на винт как картинки. Затем программой BMP2SCR переводите в scr картинки. И переименовываете их в цифры от 1 до 12. 1,2,3,4,5,6,7,8,9,10,11,12. В папку где находиться фича их скидываете. Нажимаете пуск фичи и выбираете сколько фаз, у нас 12. Пуск. Фича выдаст 12 картинок в формате BMP. И одну картинку ZADNIK в формате scr. Не пугайтесь, собирайте эти BMP прямо в GIFAnnimator, как будто это обычный gif. Получите вот это:  
  
  
Задник это и есть Ваш BACKGROUND, а гифка это гифка. Как видим программа какоса из 12 фаз выбрала то изображение котороене двигается и вынесло его в бакграунд. Таким образом спрайт облегчился значительно. Это и такты дает и одновременно освобождает память. Именно так я смог собрать вот эту демонстрацию:



<https://zxaaa.net/view_demo.php?id=16029>

Раскраску я проводил так. Сначала я нераскрашенную дему запустил, естественно скролл и анализатор выставил, затем готовое скинул как черно белый экран. Его я сохранил на дискету с ARTStudio. Раскрасил. Удалил черный рисунок, оставил только цвета. И командой MERGE в ARTStudio подгрузил в цвета данные задника. Но это уже детали. На самом деле я выйграл лишь скорость работы спрайта и 2,8 кб. Потому как Zapilyator по умолчанию сам жмет спрайты прекрасно. Но ребята, скорость работы никто не отменял. Мы боремся за каждый недо-мегагерц.   
  
На этом сегодня все. Инструкцию я дописываю по мере обретения какого либо нового навыка в работе.

**Творите демонстрацию, экспериментируйте и у Вас обязательно выйдет что то случайно ))**