

# Создание трасс для Rock'n'Roll Racing (SMD)

Трасса описывается блоком данных размером 1236 (\$4D4) байт.

Распределение данных о трассе:

\$0 - \$3FF – блоки (участки) трассы

\$400 - \$43F – вэйпоинты (направление движения, которым руководствуются компьютерные противники при езде по трассе)

\$440 - \$47F – неизвестно

\$480 - \$4BF – неизвестно

\$4C0 - \$4CF – чекпоинты

\$4D0 – стартовая позиция

\$4D1 – расстановка машин на старте

\$4D2 – количество кругов

\$4D3 – неизвестно

## \$0 - \$3FF – блоки (участки) трассы

Вся трасса делится на 64 блока. Каждый из этих блоков может быть прямой, поворотом, развилкой, и так далее.

Номера блоков трассы (в шестнадцатеричной системе):

00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F

Во время игры блоки видны игроку так:

00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F










Каждый из блоков описывается 16 байтами. Начало описания каждого из блоков относительно начала описания всей трассы (в шестнадцатеричной системе):










000	010	020	030	040	050	060	070
080	090	0A0	0B0	0C0	0D0	0E0	0F0
100	110	120	130	140	150	160	170
180	190	1A0	1B0	1C0	1D0	1E0	1F0
200	210	220	230	240	250	260	270
280	290	2A0	2B0	2C0	2D0	2E0	2F0
300	310	320	330	340	350	360	370
380	390	3A0	3B0	3C0	3D0	3E0	3F0

### Структура блока

Блок описывается 16 байтами.

1 байт – тип блока. Пусто, прямая, поворот, и так далее (см. таблицу)

Вид блока	Код блока	Описание
	17	пустой блок
	04	прямая
	05	прямая
	0A	поворот
	08	поворот
	0B	поворот
	09	поворот
	15	стартовая полоса
	16	стартовая полоса

	0F	развилка
	0E	развилка
	0C	развилка
	0D	развилка
	14	перекрёсток
	06	прямая с трамплином
	07	прямая с трамплином
	02	правый бортик
	03	нижний бортик

2 байт – неизвестно, обычно во всех трассах 00

3 байт – неизвестно, 00 если тип блока 17 (пустой), или 03 во всех остальных случаях

4-10 байты – карта высот блока. Применяется только для блоков 04, 05, 06, 07 (остальные не проверены). Стандартное значение – 00.

Блок делится на 6 участков.

Увеличивая следующий байт относительно предыдущего, на трассе в том месте будет возвышение дорожного полотна. А уменьшая – будет понижение.

Например, код 00 01 02 05 04 02 00 в этом месте будет означать сначала небольшое возвышение по трассе, потом понижение. Два соседних числа не должны отличаться больше чем на 2. 03 после 01 возможно, но 04 после 01 нет. Понижение аналогично – можно использовать 02 после 04, но 01 после 04 нельзя.

У соседних блоков должны совпадать крайние высоты. (4 байт следующего блока должен совпадать с 10 байтом предыдущего. Или наоборот, зависит от расположения)

Для всех остальных блоков (кроме 04, 05, 06, 07) 4 байт это высота всего блока, и изменяя его поднимается-опускается полностью весь блок. 5-10 байты на высоты блока не влияют. И должны быть заполнены нолями

11-14 байты – неизвестно, во всех трассах просто 00

15-16 байт – препятствие на дороге.

Если препятствия нету, оба должны быть 00

15 байт – расположение препятствия. Может быть 01, 02, 03 (правая часть дорожного полотна, середина, и левая часть)

16 байт – тип препятствия:

01 – масляная лужа

02 – липкая лужа

03 – ускоритель вправо или вниз (зависит от типа блока)

04 – ускоритель влево или вверх (зависит от типа блока)

Ещё надо протестировать все возможные варианты, но, похоже, что препятствие может быть только в блоках с типом 04, 05, 02, 03 (прямой участок)

## **\$400 - \$43F – вэйпоинты**

Противники во время езды по трассе руководствуются вейпоинтами (указатели движения). Для каждого из блоков трассы указывается, на какой блок нужно ехать дальше.

Вэйпоинты указываются так – сначала пишется номер блока, на который нужно ехать из блока 00, потом номер блока, на который нужно ехать из блока 01, далее из 02, 03, ..., 3E, 3F. Всего 64 байта. Если какой-то блок не используется (пуст, тип 17) – вэйпоинт из него указывается FF.

## **\$440 - \$47F – неизвестно**

Пока неизвестно, в игровых трассах заполнено байтами 03.

## **\$480 - \$4BF – неизвестно**

Тоже неизвестно, в игровых трассах заполнено байтами 02.

## **\$4C0 - \$4CF – чекпоинты**

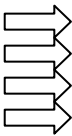

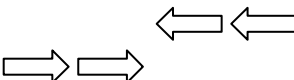
Проезжая трассу, нужно обязательно проехать по нескольким блокам. Указывается максимум 16 блоков, которые нужно проехать для того чтобы засчитали круг. Проехать нужно все, пропустить разрешается всего 1.

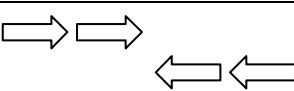
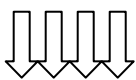

Указываются подряд номера блоков, если их меньше 16 – неиспользованные чекпоинты обозначаются FF

## **\$4D0 – стартовая позиция**

Номер блока, на котором будут стоять машины на старте (желательно, чтобы тип того блока был 15 или 16)

## **\$4D1 – расстановка машин на старте**

Код	Расстановка
01	
02	
03	

04	
05	
06	

## **\$4D2 – количество кругов**

Какое число – столько кругов надо проехать (01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, ...)

## **\$4D3 – неизвестно**

Или 05, или 00

## Добавление трассы в игру

Итак, изменённая/новая трасса готова. Теперь надо добавить её в игру. Просто взять и что-то куда-то записать не получится, придётся взять какой-нибудь HEX-редактор, оригинальный ром, калькулятор, и учесть, что заменить можно будет только первую трассу, а если не повезет, то потом вторая трасса даже не запустится и игра повиснет. Вот так жестоко, да.

Всё нижеперечисленное тестировалось на роме `Rock_n'_Roll_Racing_(U)_[!].bin`  
Что получится с другими – я не знаю, так что ковыряйте лучше этот ром.

### Шаг 1

Нужен конвертор, скачать можно, например, отсюда:

[http://homepage.corbina.net/~lich/rrr/rrr\\_decompressor.zip](http://homepage.corbina.net/~lich/rrr/rrr_decompressor.zip)

Качаем, запускаем, File -> Open Data -> Выбираем файл, в котором содержатся данные трассы (файл должен быть размером 1236 байт) -> Open

Если всё успешно нашли и открыли, жмём кнопку Compress. Рядом с оригинальным файлом появится новый файл с именем, заканчивающимся на «(CmprsdVDD)».

### Шаг 2

Открываем HEX-редактор. В нём открываем ром RRR, и файл созданный конвертером.

Копируем всё из файла созданного конвертером, и вставляем в ром RRR по офсету \$AA8C0 (в ром нужно не просто вставить, а так чтобы переписалось всё предыдущее, после ваших операций размер рома не должен измениться!).

Далее берём калькулятор. К размеру созданного конвертером файла в байтах (тут будем считать в десятичной системе счисления, чтобы не путаться) прибавляем 260288. И переводим в шестнадцатеричную систему счисления.

Например:  $260 + 260288 = 261156 = \$3FC24$

Далее в роме по офсету \$6B1B0 вписываем то, что у нас там получилось, дописав спереди нулей так чтобы было 8 цифр: 0003FC24

(это стандартные значения. Если ваша трасса будет в сжатом виде более 260 байт – она частично перезапишет вторую трассу и после первой гонки всё зависнет)

Описание недоделал: SnowWorm

<http://www.emu-land.net/forum/index.php/topic,2076.0.html>

thanks to: Ti\_