По нажатию на запуск, переходим в движок игры.

Сегодняшний исходник: zx.kaniv.net/src/cds2.asm

Start\_game.

Первым делом переключаем обработчик прерываний в обычный игровой режим, в котором будут обрабатываться события клавиатуры и мыши.

В зависимости от переключателя мыши настраивается способ ввода (кнопки от мыши идут как инвертированные биты – 0-вкл, 1-выкл.)

Фейдим экран, чистим его, переходим к загрузке карты - level\_choice

Уровня лежат в третьей паге, на уровень положено 512 байт, перемещаем нужный уровень в рабочий буфер для карт в нижней памяти.

Уровень представляет собою следующую структуру:

Сначала описан путь, по которому будут бежать враги. Путь описывается просто – в какую сторону будут проходить крипы от начальной точки (way\_begin), и закодирован просто: 1-right, 0-left, 2-down, 3-up, пока не достигнем конца списка - #ff.

;way\_begin\_adr

way\_begin dw #0800 - адрес стартовой точки в тайлах, 0 сверху и слева

Следующим в памяти расположена карта уровня, сгенерированная редактором уровней (TileEd). Карта имеет размер 16х12 тайлов, каждый 16х16 точек, и закодирована байтом на тайл.

После карты захардкожен перечень волн для данного уровня, об этом позже.

В самом конце 512-байтового блока лежат данные о начальной и конечной точке пути на карте,

Перечень доступных башен, закодированных битами для списка tower\_list, количество денег при старте уровня, а так же имя уровня.

Итак, уровень выбран, в память перенесён, данные разбросаны и некоторые декодированы.

Устанавливаем признак того, что башен у нас, устанавливем начальный спрайт внешнего вида базы (он будет показывать степень повреждения базы) , при выводим имя уровня и отображаем внешний вид карты уровня - map\_view.

Для этого, используя данные построенные в начальном ините (scr\_adr – начальные адреса для строк тайлов - #4000,#4040 и т.д.) отображаем тайлы на экран, отдельно складывая атрибуты цвета в отдельный блок памяти. Это пригодится нам для фейд-ина и фейд-аута карты уровня.

Ну и, естественно, проявляем уровень фейд-ином (метка map\_clr2). цвета уровня опять-же сохраняем для последующего использования в игре.

Переходим к созданию внутреннего представления карты для крипов и башен, map\_create.

Во первых, заливаем в память метку #ff, обозначающую что мы можем ставить в эту позицию башню. Далее, исходя из начальной позиции читаем путь и отрисовываем его в этой внутренней карте. Каждая позиция пути будет пронумерована от 1 и до n-ой конечной позиции, это будет использовано как следующий пункт при движении крипа. Последнюю позицию сохраняем как позицию нашей базы (смотри map\_create\_ex), она же будет у нас и концом пути для крипов.

Устанавливаем цвета, выводим жизни, деньги, очки, текущую волну/количество волн, уровень, отображаем доступные башни. Отрисовка происходит уже в оба экрана 128го спека.

Скопируем экран в теневой.

Так как крипы у нас выводятся по маске, необходимо сохранять данные фона.

Для этого вызываем store\_view, которая сохраняет экран в два выделенных буфера. Два буфера используется из-за 2х экранов и постоянного переключения между ними.

Сохраняем графику под курсором мыши.

Игра уже почти началась! Стоит показать путь, по которому пройдут волны врагов.

way\_view – пробегаемся по карте с поиском необходимого номера тайла пути и подсвечиваем его, а через некоторое время – возвращаем его цвет обратно. Выполняем это до тех пор пока не дойдём до последней точки пути, цели крипов – базы.

Сохраняем атрибуты для последующего вывода башен, заодно инициализируя память под это. Восстанавливаем цвета.

Инитим переменные количества волн, счётчика волн, количества башен, служебных переменных.

Переходим к иниту крипов (enemy\_create)

Начинаем с первой волны. Получаем количество крипов, базовую энергию (16 бит), период начального ожидания перед стартом волны, собираем таблицу крипов которые будут участвовать в волне, и начинаем инитить крипов: позиция на пути, начальный номер спрайта, первоначальное направление в котором они будут перемещаться, начальный адрес в экране при их появлении, общая задержка перед стартом крипов, добавляем небольшой рандом при появлении следующего крипа в волне, учитываем что крип может быть один, выбираем количество жизней для каждого крипа по его типу добавляя сюда немного рандомного значения для пущего интереса, инитим флаг замедления крипа.

Увеличиваем счётчик волн и выводим его значение, как и количество волн вообще, показываем сколько вообще крипов в волне будет, трынькаем и выводим таблицу – НОВАЯ ВОЛНА!

Игра началась.

После карты захардкожен перечень волн для данного уровня:

org levels\_mem+#130

; kolvo vragov, base energy, begin wait,0, wait ,enemy\_bit\_list - from left

db 12, 4, 0, #a0, 0, #14,0,%00000001

или db 20,#f0,1, #a0, 0, #13,0,%00001101

Важные пункты: количество крипов, Базовая энергия крипа (при запуске волны будет немного изменятся в небольших пределах для каждого), ожидание перед запуском волны, таймаут перед выпуском следующего крипа, биты крипов, которые будут выпущены. Здесь отметим, что врагов может быть лишь 8 типов.

Регулировать сложность/плаябельность уровня можно только переигрывая волну за волной, где-то повышая скорость выхода крипов, где-то понижая