RPG Nexus

Revista sobre RPG Maker

Número 2

Índice

Contenidos

Sobre la Revista

Acto 2

"No es fácil pero vamos a seguir adelante"

Esto de montar una "revista casera" no es fácil (como ya se preveía), sobretodo en cuanto a tiempo y disponibilidad de internet para poner un poco de control entre los colaboradores y contenidos. Aún así, seguimos adelante.

A partir de ahora, para la revista, habrá que ver qué contenidos metemos.
En cuanto a tutoriales de RPG Maker seguirá habiendo cosas más o menos fáciles y básicas.
Para grafismo, el mismo nivel de los tutoriales que se han publicado ya.
Y sobre los scripts, una vez esté explicado el tema del SBP (que tanta gente pide), habrá que meter un tutorial general para los que quieran aprender RGSS.

Esto significa que habrá más números, así que esto marcha bien.

Contacto: ***rpgnexus@gmail.com***

Noticias sobre RPG Maker

Novedades en el "panorama"

RPG Maker XP 1.03

Enterbrain anunció la salida de la versión 1.03 del RPG Maker XP.
Esta versión elimina los problemas de RMXP en Windows Vista y problemas menores.

De momento no hay versión oficial en inglés ni traducción a español, pero no es una actualización imprescindible en caso de utilizar el RPG Maker bajo Windows XP.

¿Posible regreso de Radio Maker?

Hace un tiempo **Makerhack** planteó la posibilidad de que volviese Radio Maker, y por otro lado **Lampard** lleva tiempo estudiando la posibilidad de llevar adelante una radio.

Puede ser que cada vez esté más cerca la vuelta de un programa radiofónico sobre RPG Maker, habrá que esperar a ver si la idea sale adelante, quien sabe.

Rincón Maker desaparecido...

Desde hace unas cuantas semanas no se puede acceder a Rincón Maker, si intentamos acceder acabamos en el blog personal de uno de sus administradores (Gash).

Se desconoce el motivo de la caída o si sus administradores (Gash, Ichinose y Zero) volverán a levantar la comunidad una vez más, pero si finalmente no vuelve (como parece) se habrá perdido una comunidad interesante, heredera de la vieja web de Uva_r.

Concurso de juegos de verano

Se ha organizado un concurso de juegos, en **RPGmakerXP.com**.
Actualmente las inscripciones están cerradas, y hay 15 juegos inscritos.
Para más información: >> **Enlace al tema del concurso**

El proyecto EasyRPG

Pequeño resumen general

Introducción

EasyRPG pretende ser un programa similar a «RPG Maker», una herramienta de creación de videojuegos de rol (RPG) que no requiere conocimientos de programación.

Será un RPG Maker **libre y multiplataforma** (también gratuito), su código podrá ser modificado libremente para dotar al programa de mayor flexibilidad y potencia.

El Proyecto

El primer paso es crear un **clon del RM2000**, una vez conseguido eso, habrá tiempo de sobra para convertirlo en una herramienta que se ajuste a las necesidades que surjan.

Algunas personas ya han tenido dudas al ver una muestra del aspecto de la interfaz, que es clavada a la del RM2000 y en inglés, pero no es algo definitivo ni mucho menos, es sólo una base, además para los que aún no lo tengan claro, EasyRPG será **multilenguaje** (incluyendo español, por supuesto).

En principio, EasyRPG se divide en varios subproyectos:

- Desarrollo del editor
- Desarrollo del intérprete
- Material/Arte
- Documentación
- Internacionalización/Tradución

El Editor

Actualmente, se está diseñando toda la interfaz gráfica del EasyRPG (en principio, una copia idéntica al RM2000).
Está bastante avanzado, la mayor parte de las ventanas están ya diseñadas utilizando **wxGlade**, un “diseñador de interfaces” para **wxWidgets**.

El Intérprete

Hace tiempo se buscó información y se “estudiaron” los archivos de mapas de RPG Maker 2000 y 2003 (.LMU), fue un avance interesante a partir del cual, Damizean desarrolló un pequeño **“emulador”** de esos mapas.
Esto permitirá que en un futuro se puedan importar proyectos hechos en RPG Maker.

De momento se sigue estudiando varios aspectos que ayuden a entender mejor el funcionamiento del intérprete del RPG Maker y poder aplicarlo al EasyRPG.

A continuación dejo la descarga del "emulador" de mapas de Damizean, y también un mapa y un chipset para probarlo.

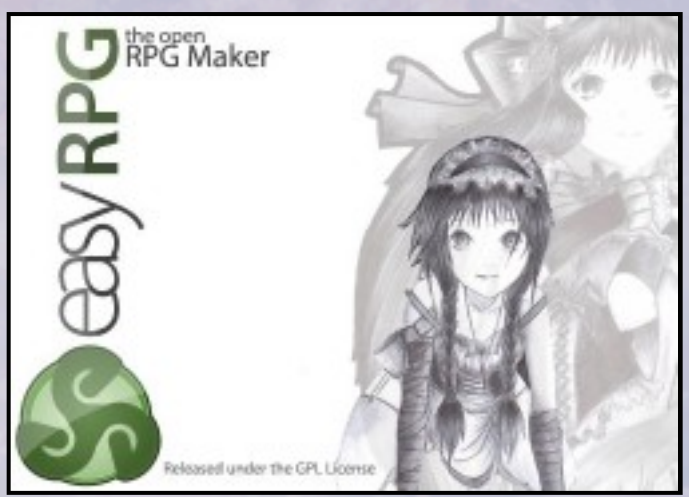
>> **Enlace al emulador**

Material/Recursos

Es la parte del proyecto que más está avanzando gracias a la ayuda de mucha gente que colabora constantemente, así que resumiendo un poco el asunto:

Ha habido algunas propuestas para el logotipo, splash y la identidad visual, por ejemplo estos trabajos de Enigmart y Benbeltran, que han gustado bastante.

easyRPG

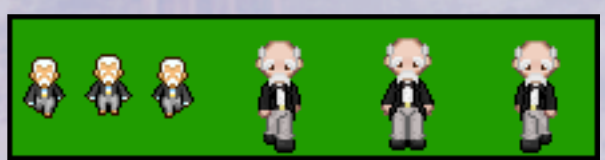


OpenRTP

El proyecto OpenRTP pretende desarrollar charas que se publicarían bajo licencia **GPL** (libres), a partir de los charas que trae el RTP del RPG Maker.

Digamos que es una **adaptación del RTP del RPG Maker** bajo otro estilo, para tener una base de gráficos para EasyRPG, y que a la hora de importar un juego, se mantengan en el juego charas similares.



Pero no sólo está el OpenRTP, hay un montón de charas bastante interesantes que siguen distintos estilos, y que también forman parte de la **gran colaboración** que hay en cuanto a material gráfico, donde cualquiera puede ayudar.

A continuación, una muestra de distintos recursos hechos hasta ahora.

Ideas / Propuestas

Desde hace meses han ido apareciendo propuestas e ideas para las **funcionalidades** del programa y demás, desde luego en ese aspecto está más que cubierto.

Aunque más adelante será momento de revisarlas todas y aplicar las más interesantes al proyecto, se puede seguir debatiendo sobre distintos puntos (en el foro o en el wiki).

EasyRPG es un proyecto bastante ambicioso, no es algo que se pueda finalizar en sólo un año, y aunque parezca que avanza muy lento, está llevando un buen ritmo gracias a la gente que colabora actualmente.

Cualquiera puede colaborar, creando material, con la documentación, programación... lo que sea, es un proyecto abierto a cualquiera, y toda ayuda será bienvenida.

A continuación hay varios **enlaces** para cualquier interesado en colaborar con el proyecto, y para los que quieran más información sobre el proyecto.

>> Enlace al wiki

>> Enlace a easy-rpg.com

>> Enlace al Almacen de Recursos

Para este artículo se han utilizado imágenes sacadas del Almacén de Recursos del proyecto.

Entre los autores de los recursos mostrados están Astral, Bolt, Chaleeman, DarkChanT, Forest, slash128, seroc, step_puki y endless_dark.

El Libro

Autor: Eldark Analizado por: Nirok RM2003

Historia

El protagonista de la historia es **Grohen Mitch**, un abogado al que últimamente le suceden cosas extrañas.

Un día recibe la llamada de su esposa en medio de una noche pidiéndole que acuda a una antigua mansión, la mansión de su padre.

En todo momento nos moveremos por esa mansión, un lugar abandonado y siniestro, perfecto para el modo en el que se desarrolla la historia.

Desde el principio hasta casi el final pensaremos que es una historia llena de **paranoias y locuras**, pero todo cobrará sentido dando lugar a un final gustoso que sin duda dejará satisfecho a más de uno.

Este juego posee una historia buena y original, nueva en el genero "survival".

Gráficos

En este punto destaca por los acertados mapeados que se lograron, **ambientando perfectamente** el lugar.

Da vida a una mansión llena de maldad donde la historia y el ambiente se desarrollan perfectamente.

Aunque los gráficos no son 100% originales, están bien elegidos y bien editados. También hay que destacar los efectos de luz, que perfeccionan la ambientación.

Hay que comentar que el hecho de ver casi siempre un mismo gráfico de mapa puede hacer que el juego en este aspecto sea un poco agobiante o repetitivo, se podría haber incluido un sótano u otro tipo de mapa para dar mayor variedad.

Música y Sonido

No hay muchas variantes musicales pero sí son las justas, que ambientan a la perfección los escenarios, nos avisan del peligro y nos mantienen atentos a lo que pueda suceder.

Los sonidos están muy bien colocados, mejoran la ambientación, hay risas malévolas, chapoteo en charcos de sangre, gritos, voces... en conjunto dan mucho realismo.

Además la música no se hace nada repetitiva, acompaña al desarrollo sin estorbar.

Entretenimiento

El juego en sí es lineal (habitual en aventuras de tipo "survival"), cada acción nos habilita otras, siguiendo un desarrollo constante, al más puro estilo *Resident Evil*.

El hecho de **"volver atrás"** para realizar algo que antes no podías también es característico, pero la mansión no es muy grande y el juego corto, no agobia demasiado.

El juego daba para más en cuanto a duración, pero no la historia, está suficientemente explotada como para terminarla en el momento justo.

Dificultad

El tema de los combates (ActionRPG) está **bastante equilibrado** en todos los aspectos, además hay suficientes balas y sprays curativos, si se usan correctamente.

El puzzle al que nos enfrentamos es fácil de seguir, sabremos dónde está todo si vamos tocando todo lo que hay en las habitaciones según avanzamos.

A pesar de ser un juego corto, es entretenido, muy bueno en todos los aspectos y genialmente ambientado.

Sin duda cumple todas las expectativas, aunque tiene algunos bugs que se corregirán en los próximos meses.

>> **Descargar "El Libro"**

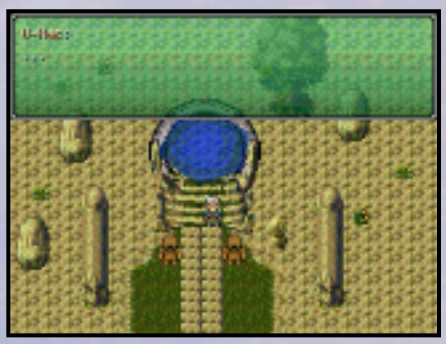
El Legado de las Estrellas

Autor: Biohazard Analizado por: PsYcO RM2003

Historia

Ark, un héroe de un planeta muy lejano al nuestro es engañado por los malvados alienigenas Goa´Uld para que éste abra el **Stargate**, un portal interestelar.

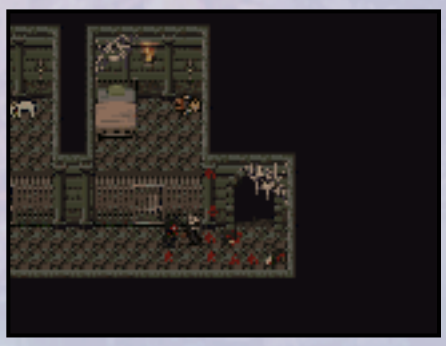
Así, los crueles Goa´Uld llegan hasta la tierra, donde esclavizan a los humanos haciéndose pasar por dioses.



Tras estos sucesos Ark despierta desorientado, sin saber cómo ha sobrevivido, en un planeta con unos avances muy inferiores a los suyos.

Allí conoce a unos seres de una antigua raza enemiga de los Goa´Ulds, que le ayudarán y le guiarán en su lucha para enmendar el terrible error que cometió.

Gráficos



La **homogeneidad** es total en todos los gráficos, todo sigue una línea con estilo similar al RTP, con un buen mapeado y una ambientación más que decente.

Hay armonía entre charas, chipsets, facesets y demás gráficos de conjunto, por lo que el autor se ha arreglado muy bien con el material que tenía.

El colorido difiere bastante entre algunos charas, unos muy vivos y otros muy apagados.

Además algunas imágenes desentonan al no seguir el estilo RTP del juego (suelen ser imágenes "reales" sacadas de algunas series y películas).

Al principio del combate contra algunos enemigos se muestran charas de éstos que, desgraciadamente, no siempre se corresponden con su gráfico de "monster".

Música y Sonido

Hay gran diversidad de estilos entre las músicas, desde películas míticas como *Rocky*, hasta videojuegos populares como *Metal Gear*, pasando por música clásica.

La mayoría de los temas son "midis", aunque hay un par de temas en formato MP3.
En cuanto al sonido, es casi todo RTP y hay algún rip.

En general **todas las músicas encajan** con la situación y no desafinan.

Entretenimiento

Hay puzzles y minijuegos interesantes, podemos pescar y tocar la ocarina al más puro estilo *Zelda*.

También hay otros engines como el de reloj (además de los dos que ya se han comentado), e incluso tenemos ActionRPG para algunos combates.

Los combates son abundantes y utilizan el sistema de batalla que trae por defecto el RPG Maker 2003, el menú también es el que trae el Maker por defecto.

Dificultad

La dificultad de los combates es **media tirando a fácil**.

Respecto a los puzzles hay variedad, desde los más lógicos y simples hasta otros que requieren romperse un poco la cabeza para resolverlos.

Los minijuegos tienen una dificultad medida con maestría, lo justo para que sea un desafío y que a la vez no resulten fáciles ni desesperantes.

Como conclusión se puede decir que tiene un argumento interesante ambientado en el **universo Stargate**, con desafíos y juegos de lógica.

A pesar de que los combates y la armonía de algunos gráficos es mejorable, en conjunto es un buen juego.

>> **Descargar "El Legado de las Estrellas"**

Tutorial de Variables

RM2000
RM2003
RMXP

Dificultad:

¿Qué son las variables?

Una variable es un elemento que **puede guardar datos**, en este caso siempre serán datos numéricos (únicamente enteros, el Maker no usa decimales), y se puede hacer que esos valores cambien a nuestro antojo en el juego (desde el modo de edición).

Un interruptor sólamente puede guardar el valor "activado" y "desactivado", mientras que una variable puede guardar un valor número, por ejemplo: -5, 0, 17...

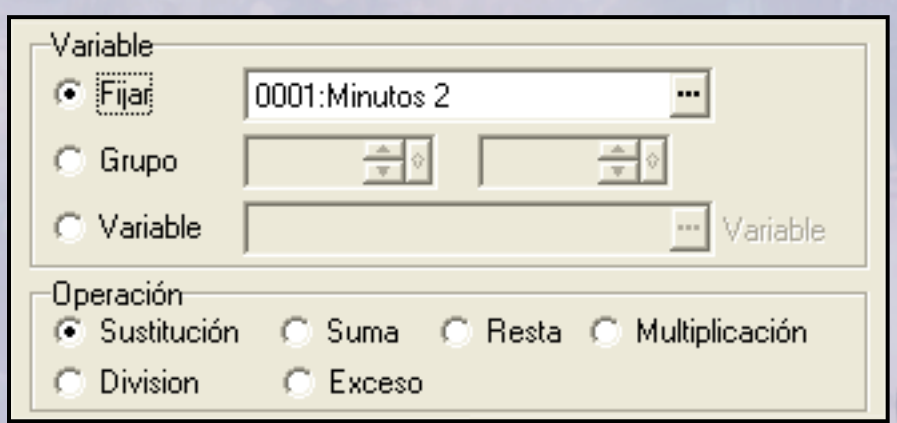
¿Para qué sirven las variables?

Una variable es mucho más flexible que un interruptor, podemos almacenar distintos valores (en un interruptor sólo dos), por lo cual podemos hacer que para cada valor que puede tomar la variable, ocurran cosas distintas.

Podemos crear distintas situaciones:

- Que un evento se ejecute cuando una variable tome un determinado valor.
- Que un evento se ejecute cuando una variable tome un valor dentro de un rango.
- Que un evento realice una cosa distinta para cada valor.
- ...

Además de esto podemos utilizar las variables como contadores, para saber qué opción estamos seleccionando en un menú... en fin, tiene muchos usos.



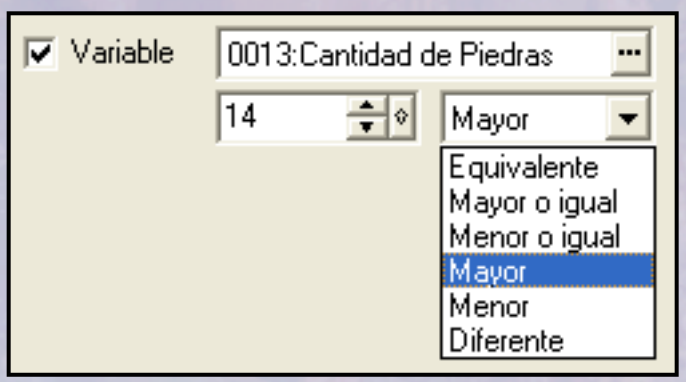
¿Cómo se usan las variables?

Al igual que los interruptores, se utilizan como **condición de situaciones**.
Se pueden utilizar para que en un evento puedan suceder varias cosas distintas, o si utilizamos la variable como contador, podemos usarla para saber si ese contador ha llegado a una cifra determinada (por ejemplo un contador de monedas, el valor de la variable sería la cantidad de monedas), y podemos operar con ella.

Para modificar el valor de una variable tenemos el comando **Operaciones de Variable**, que permite hacer operaciones sobre la variable, con un número (sumar, restar, multiplicar, dividir...), incluso podemos operar con los valores de dos variables.

Las variables se usan para las condiciones de un evento, de dos maneras:

1) Condición de inicio en un evento
Un evento puede tener varias páginas (cada página representa una situación).
A cada página podemos asignarle una variable como condición, así dependiendo del valor de una o varias variables, se ejecutará el contenido de una u otra página del evento.



2) Condiciones y Efectos
Esto ya funciona dentro de un evento, entre el "código".
En este caso sirven para algo más concreto como por ejemplo, que dependiendo del valor de una variable se muestren unos mensajes u otros (en general, que se ejecuten distintos comandos dependiendo del valor de la variable).

Para ello contamos con el comando **Condiciones y Efectos**, podemos hacer que si el valor de una variable es mayor/menor/igual/distinto a un número determinado, se ejecute un "código", y si activamos la opción de Excepción podemos hacer que si la condición no se cumple, suceda otra cosa.

El tutorial original se publicó en www.tutoriales-maker.com

Engine de Linterna

RM2000
RM2003
RMXP

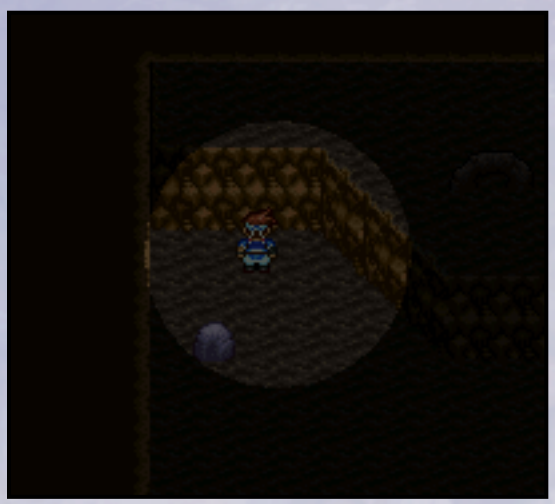
Dificultad:

Imágenes necesarias

Para hacer este efecto se necesita, como mínimo, una imagen que ocupe toda la pantalla (320x240 pixels en RM2000/2003 y 640x480 pixels en RMXP).

La imagen debe tener un fondo negro y una figura más o menos centrada (generalmente se usa un círculo), del color que queramos, pues ese color luego actuará como transparente.

La idea es que la imagen quedará anclada a la pantalla, aunque nos movamos por el mapa, la imagen siempre estará centrada y será visible, dejando la zona del círculo (o la figura que usemos) más iluminada, de manera que alrededor del personaje habrá iluminación, y en el resto del mapa se verá todo más oscuro.



¿Cómo hacer el sistema?

Para crear el efecto sólo hay que situarse en los mapas donde vayamos a usarlo.

Creamos un evento en cualquier punto del mapa donde no nos moleste, por ejemplo arriba a la izquierda del todo.

El evento tendrá dos páginas, la primera en **Proceso Paralelo** y la segunda como **Pulsar Aceptar** o **Pulsar Tecla de Decisión** (depende de la traducción).

La idea es la misma tanto en RPG Maker 2000/2003 como en XP y se hace casi igual, sólo hay que oscurecer la pantalla cuando queramos utilizar la linterna, y mostrar la imagen, pero voy a explicar cómo se hace en RM2000/2003 y en XP.

En RPG Maker 2000 y 2003

En la primera página del evento sólo hacen falta tres comandos:

Primero ponemos un comando **Cambiar Tono de Pantalla**, y ajustamos los parámetros para oscurecer la pantalla, por ejemplo (25,25,25) (Rojo, Verde, Azul).

Ponemos un comando **Mostrar Imagen**, usamos la imagen de linterna que hemos hecho antes y la situamos en el centro (X=160, Y=120).
Se le puede poner algo de **transparencia** también, por ejemplo un 25% o 35%, aunque si lo queremos hacer oscuro de verdad, ponemos un 0% de transparencia.

Para acabar con la primera página, activamos el interruptor *Linterna*.

La segunda página del evento la dejamos vacía, pero ponemos como condición de comienzo, que el interruptor *Linterna* esté activado.
Esto último sirve para que el juego no esté continuamente oscureciendo la pantalla y poniendo la imagen, sólo hace falta una vez, si no, puede provocar alguna ralentización.

En RPG Maker XP

Ponemos un comando de **Cambiar Tono de Pantalla**, ajustando los parámetros para oscurecer la pantalla, por ejemplo (-110, -110, -110) (Rojo, Verde, Azul).

Ponemos un comando **Mostrar Imagen**, elegimos la imagen de linterna, la situamos en el (0, 0), tomando como referencia Arriba-Izquierda, en vez de Centrado.
Como en el otro caso, ajustamos la **Opacidad** (transparencia), que podemos dejarla como queramos, por ejemplo a 180.

Para acabar con la primera página, activamos el *interruptor local A* (para cosas que sólamente ocurren en un mismo evento es más cómodo usar un interruptor local, así nos evitamos utilizar uno de los interruptores normales).

La segunda página del evento la dejamos vacía, pero ponemos como condición de comienzo, que el *interruptor local A* esté activado.
Esto lo hacemos por el mismo motivo que en el otro método.

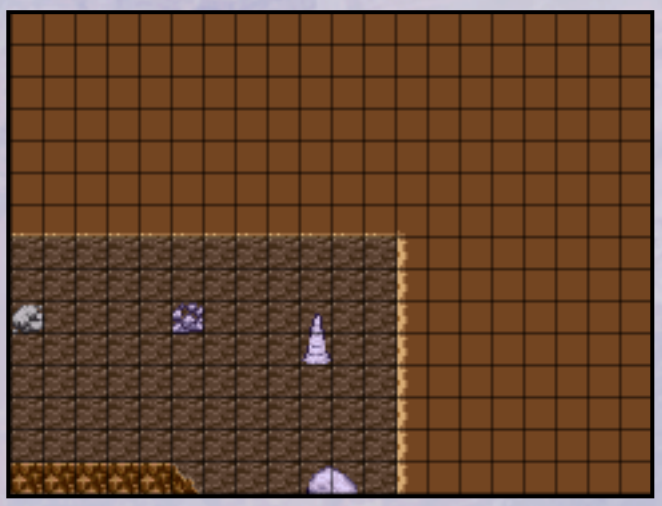
Aunque parece que ya está todo, no hemos acabado aún, queda algo pendiente, y es que se nos presenta un problema que tendremos que solucionar.

Solucionando un problema

Una vez hemos hecho el sistema podemos darnos cuenta de un problemilla que aparece si nos movemos hasta los extremos del mapa, el jugador sale de la posición iluminada.

Para solucionarlo sólamente tenemos que hacer el mapa más grande y centrarlo, digamos que habría que meter tiles por los 4 lados del mapa, tiles a los que no se puede acceder ni servirán para nada más.

Como se ve en estas imágenes de abajo, he metido tiles de separación para cada borde del mapa, así se soluciona.



Cosas a tener en cuenta

Para desactivar la linterna, simplemente hace falta borrar la imagen, poner el tono de pantalla normal y desactivar el interruptor que activamos antes.

No tiene por qué ser una sóla imagen, podemos crear linternas con varias imágenes, e incluso puede ser otra forma que no sea un círculo, hacerlo mas complejo ya depende de lo que quiera cada uno, es cuestión de probar.

En el tutorial original (de **Tutoriales-Maker**) hay dos ejemplos que se pueden abrir con el RPG Maker para ver cómo están hechos.

Entre los extras de la revista se adjunta la imagen de linterna utilizada en el tutorial.

Charasets a partir de plantillas (Parte 2)

Grafismo

Dificultad:

En esta segunda parte del tutorial de creación de charas a partir de plantillas vamos a tratar el pelo, su silueta, brillo, sombreado...

La silueta

Lo primero es dibujar una silueta alrededor de la cabeza del personaje, ese será la forma del peinado.

Conviene que esa silueta esté un poco por encima de la parte alta de la cabeza, para que quede bien.

El relleno

Ahora tenemos que rellenar el interior de forma que hemos dibujado, y lo haremos con un color más claro que el que utilizamos para la silueta.

Siempre es bueno aclarar ciertas zonas de los bordes para lograr un efecto más suavizado, en especial en las partes que no conforman la silueta del personaje (como el flequillo).

Sombreado básico

Ahora aplicamos un tono más claro sobre todo el pelo, usando el tono de relleno como un degradado que suaviza el contraste entre este nuevo color y los bordes.

Los brillos

Para agregar los brillos hay que tener en cuenta que el pelo está formado por mechones, y cada mechón se deriva del centro de la cabeza. Así que añadiremos uno o dos píxeles de tono claro sobre los distintos mechones y oscurecemos un poco algunas zonas de la cara cubiertas por el flequillo.

Creando Facesets

Grafismo

Dificultad:

Este tutorial de creación de facesets es bastante directo, la explicación va paso por paso ayudándose de imágenes.

Paso 1

Se dibuja la silueta de la cara y el cuello, con un "color piel" más o menos oscuro.

Es importante practicar mucho para que nos salga bien la silueta, a la primera es difícil.

Paso 2

Se rellena el interior con un color más claro que el que hemos usado para el borde, será el tono de piel del face.

Luego utilizamos un tono un poco más claro que el del borde, hará la unión entre el color del borde y el del interior.

Paso 3

Con un color un poco más claro que el anterior, hacemos lo mismo que antes y lo usamos para combinarlo y hacer la silueta más gruesa.

Ahora podemos sombrear la cara con un color claro, teniendo en cuenta de donde viene la luz.

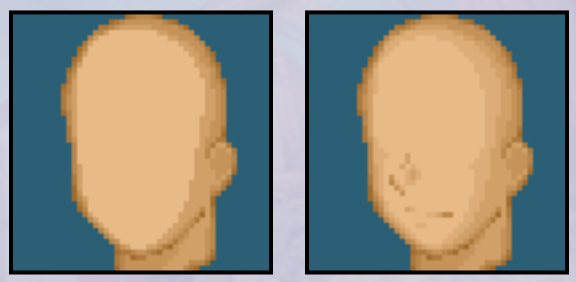
Hasta ahora hemos estado dando volumen a la cara, utilizando varios tonos para darle un toque suavizado, desde el color del borde de la silueta hasta el color de relleno.

Se han utilizado tonos en progresión, cada vez más claros teniendo como límite de claridad, el tono utilizado en la cara.

Paso 4

En el ejemplo hemos ido utilizando hasta 7 tonos (siendo el del borde el más oscuro, y el de relleno el más claro), para darle volumen.

Después de eso podemos dibujar fácilmente la nariz y la boca (lo podíamos haber hecho al principio).



Paso 5

Podemos hacer los ojos y cejas, teniendo en cuenta el color de pelo que tendrá el face.

En principio las cejas son una línea gruesa más o menos diagonal, según hacia donde mire la cara, un ojo deberá ser un poco más grande que el otro.

Para la piel se puede utilizar un octavo tono para darle brillos, y en cuanto al pelo, es complicado, primero deberemos dibujar la silueta y rellenarlo de un color base.

Paso 6

Para sombrear el pelo utilizamos la misma técnica que en la cara, utilizar varios tonos que difieran un poquito para dar sensación de suavizado.

Es cuestión de practicar mucho.

Paso 7

A partir de ahí podemos añadirle accesorios como la parte superior de la camisa o chaqueta, que al igual que la cara y el pelo, se basa en un color de borde, un color de relleno, y varios tonos.

Podemos aprovechar para hacer retoques en el pelo, ojos, orejas y cuello, para que queden mejor.

Sistema de Batalla de Cybersam (Parte 2)

Scripts RGSS

Dificultad:

Animaciones

En las dos últimas de código del tutorial anterior vimos dos métodos, **pose()** y **enemy_pose()**, que manejan las animaciones (posiciones) que tendrán los personajes. Vamos a ver estos métodos en profundidad.

La primera línea (en la primera imagen) llama la atención:
def pose(number, frames = 4)

```
def pose(number, frames = 4)
  case number
  when 0  # run
    change(frames,5,0,0,0)
  when 1  # standby
    change(frames,5,0,@frame_height)
  when 2 # defend
    change(frames,5,0,@frame_height*2)
  when 3 # Hurt, loops
    change(frames,5,0,@frame_height*3)
  when 4 # attack no loop
    change(frames,5,0,@frame_height*4,0,true)
  # ...etc.
  else change(frames,5,0,0,0)
  end
end
```

En esta línea podemos modificar el número de frames que tendrá la plantilla del personaje, que en la imagen es **4**.

Como podemos ver en las siguientes líneas se define el orden y las características que tendrán cada una de las posiciones de nuestra plantilla (corriendo, defensa...).

Nos quedamos con **change(argumentos)**, que está en el script "Animated Sprite".

En la primera línea (de la segunda imagen) vemos los argumentos que define cada pose de la plantilla.

```
def change(frames = 0,delay = 0,offx = 0,offy = 0,
  startf = 0, once = false)
  @frames = frames
  @delay = delay
  @offset_x, @offset_y = offx, offy
  @current_frame = startf
  @once = once
  ...
```

frames – Nº de frames de la animación
delay – Intervalo de tiempo entre frames
offx – Distancia del eje X
offy – Distancia del eje Y
startf – Frame inicial de la animación (pose)
once – Sólo se reproduce una vez

Agregando posiciones

Una vez hemos comprendido como funcionan los métodos para las animaciones (poses) vamos a ver como podemos añadir más o cambiar las que ya hay.

Tomamos como ejemplo la siguiente línea:

```
when 2
    change(frames, 5, 0, @frame_height * 2)
```

En este caso la animación 4 frames por defecto (frames hace referencia al valor por defecto), el tiempo de espera es 5 y la pose empieza en las coordenadas:
(0, @frame_height * 2)

Frame_height se puede definir (se vió en la pimera parte del tutorial), por defecto es **64**.

Por lo tanto, si ya tenemos 9 poses hechas, y queremos hacer una nueva (la 10), tendremos que seguir la plantilla (visto en la primera parte del tutorial).
La posición 0 corresponde a la primera animación de la plantila.

Si queremos que la animación no sea contínua, tendríamos que añadir en que frame debe comenzar la animación y poner true en el argumento once. Nos quedaría así:

```
when 9
    change(frames, 5, 0, @frame_height * 9, 0, true)
```

Vamos a ver un ejemplo de plantilla.

Hay 3 poses que sólo tienen un frame y que por tanto no sufrirán un loop.

Por lo demás, en este ejemplo cada frame es de 64x64 pixels, así que para referirnos a un frame, usaremos las coordenadas:
(0, frame_height * posición)

Recordemos que la posición 0 se refiere a la animación 1 de la plantilla.

Pose			
Quieto	Frame 1		
Correr	Frame 1	Frame 2	
Herido	Frame 1	Frame 2	Frame 3
Ataque	Frame 1	Frame 2	Frame 3
Habilidad	Frame 1	Frame 2	Frame 3
Muerto	Frame 1		
Victoria	Frame 1	Frame 2	Frame 3
Ataq. Especial	Frame 1	Frame 2	
Est. Alterado	Frame 1		

```
def pose(number, frames = 3)
  case number
  when 0  # Quieto
    change(1, 5, 0, 0, 0)
  when 1  # Correr
    change(2, 5, 0, @frame_height)
  when 2 # Herido
    change(frames,5,0,@frame_height*2)
  when 3 # Ataque
    change(frames,5,0,@frame_height*3,0,true)
  when 4 # Habilidad
    change(frames,5,0,@frame_height*4,0,true)
  when 5 # Muerto
    change(1, 5,0, @frame_height * 5)
  when 6 # Victoria
    change(frames,5,0, @frame_height*6)
  when 7 # Ataque Especial
    change(2, 5, 0, @frame_height * 7)
  when 8 # Estado Alterado
    change(1, 5,0,@frame_height * 8)
  else
    change(frames, 5, 0, 0, 0)
  end
end
```

Este sería el código correspondiente a cada posicion de la plantilla anterior.

Vemos que para las animaciones de ataque y de habilidad, tenemos el valor true para el argumento once, ya que sólo se reproducen una vez.

También vemos que en la plantilla utilizamos 3 frames como máximo para cada pose (algunas sólo tienen 1 o 2).

Para las poses que utilizan 3 frames, nos referimos al valor frames, pero si tienen sólo 2, pues ponemos un 2, como en la línea de **Ataque Especial**.

Por lo demás, siguiendo las explicaciones anteriores y trasteando un poco, se consigue fácilmente personalizar la plantilla y ajustar el código.

Pose inicial y Estados alterados

La pose inicial es la animación a la que vuelven los personajes después de realizar una acción. Ya se dijo que si se modificaba **pose(X)** o **enemy_pose(X)** había que modificar el método **def default_pose**, que controla la pose inicial y los estados alterados.

```
def default_pose
    pose(1)
  if (@battler.hp*100) /@battler.maxhp < 25
    pose(9)
  elsif (@battler.hp*100) /@battler.maxhp < 50
    pose(9)
  elsif (@battler.hp*100) /@battler.maxhp < 75
    pose(9)
  end
  if @battler.state?(3) # poison
    pose(2)
  elsif @battler.state?(7) #sleep
    pose(6)
  end
end
```

Lo primero es la pose por defecto **pose(1)**.

Con **pose(X)** estamos llamando al método que hemos visto antes, donde X es la nueva posición en la plantilla.

Siguiendo nuestra plantilla tendriamos que sustituirlo por **pose(0)** ("quieto" es la pose 0, la que queremos que se muestre por defecto.

Sólo afecta a los héroes.

En el código Cybersam incluye tres ejemplos de estados alterados, el primero cambia la pose cuando el personaje tiene determinada cantidad de vida (menos del 25%).

```
# if HP is too low (- 25%)
if (@battler.hp * 100) /@battler.maxhp  < 25
    pose(9)
```

Con esto podriamos crear nuestras propias condiciones, por ejemplo, si el personaje se queda sin PM podemos cambiar la pose por una en la que se muestre cansado.

```
if @battler.sp  == 0
    pose(8)  #Esta es la posición de cansado en nuestra plantilla.
```

Los otros dos ejemplos son estados alterados propiamente dichos, es decir, cuando el personaje tiene un estado de la base de datos (veneno, sueño...).

```
if @battler.state?(3) # poison
    pose(2)
```

Esta línea comprueba si el héroe tiene el estado alterado nº3 de la base de datos (veneno) y si se cumple cambia la pose.
Para añadir más estados simplemente tendriamos que ir añadiendo condiciones.

```
elsif @battler.state?(7) # sueño
   pose(8)
elsif @battler.state?(6) # confusión
   pose(8)
```

Es importante destacar que el orden es muy importante, la última condición que se cumpla será prioritaria sobre las demás, de forma que si nuestro personaje tiene la vida por debajo del 25%, los PM a 0 y el estado alterado "confusión", la pose cambiará a esta última, que tiene mayor prioridad que las otras dos por estar más abajo.

Para poner la pose de muerte (cuando la vida llega a 0) vamos al método "def update", y en las ultimas líneas encontramos:

```
$game_system.se_play($data_system.actor_collapse_se) unless @dead
@dead = true
pose(6)
```

Con nuestra plantilla el valor correcto seria **pose(5)**, y no pose(6).

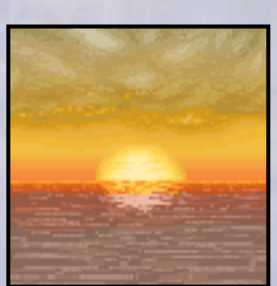
Continuará en el próximo número...

Galería de Imágenes

Marina - Hibari

Rasec - Aizen

Pems - Sprite2



Ito - Atardecer

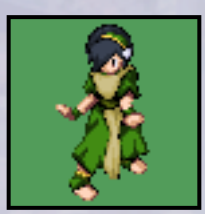
Nyaro! - Shiria-chan

Siremtukbac - Vurotráx

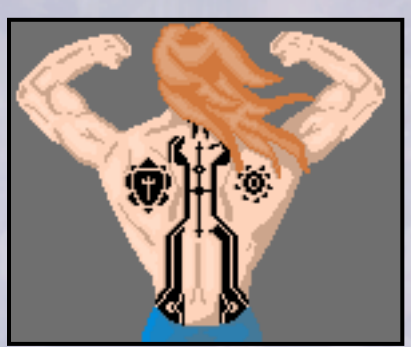
Ito - Link

Pickspy - Nintendo DS Tan

OZiriZO - Medaman



OZiriZo - Pixel Troph



Rasec - El Mazas

Despedida y agradecimientos

Información sobre RPG Nexus

RPG Nexus es un proyecto creado en **AldeaRPG** y dirigido por **BlueFalcon**.
En la creación y distribución de la revista colaboran varios usuarios y comunidades:

Gracias a las personas que han colaborado y han hecho posible este número:
(Testament, Kotfire, Nirok, Rune/Heike, Ito, OZiriZO, Marina, PsYcO, Moxy_Ragnarok, delaPipol, Rasec, Pems, Nyaro!, Siremtukbac y Pickspy)

Información

Existe una dirección de contacto de email donde podéis enviar vuestras opiniones, sugerencias, correcciones... (**rpgnexus@gmail.com**)

Además, **cualquiera puede participar** en la revista haciendo análisis, tutoriales, reportajes... para participar o recibir información sobre ello, podéis contactar con *BlueFalcon*, o enviar un email a la dirección mencionada.

La revista sale cada dos meses aproximadamente (no se puede dar una fecha exacta), se puede descargar en varias comunidades (que colaboran), o por eMule.
Hay información sobre la revista en AldeaRPG y algunas webs colaboradoras.

Es difícil que la revista guste a todo el mundo, pero ojalá podáis entreteneros con su lectura tanto o más de lo que nos entretenemos nosotros al hacerla.

¡Hasta el próximo número!