

Como crear un personaje para Pocket Volleyball

Créditos

- » **Autor:** Hugo Ruscitti
- » **Fecha:** 2 de Marzo del 2006

Introducción

Pocket Volleyball ha sido diseñado con la intención de facilitar la inclusión de nuevos personajes, no necesitará preocuparse por reconstruir el programa o configurarlo.

El proceso es muy sencillo, necesitaremos crear 2 imágenes y un archivo de configuración donde especificaremos información acerca del personaje y su autor.



El Colo, un personaje de Pocket Volleyball

Para comenzar adoptaremos de mascota al personaje Colo realizado por Walter Velazquez 1, miembro del grupo LosersJuegos. Puede descargar la imagen original de la que he partido para desarrollar esta guía.

Tenemos la posibilidad de utilizar gráficos de cualquier tamaño, aunque recomiendo utilizar un tamaño cercano a 50x130 píxeles (ancho x alto) para mantener una proporción de tamaño en todos los personajes2.

¿ Como funciona ?

Cuando ejecuta el videojuego se realizará una búsqueda completa de personajes. Dependiendo del sistema operativo que emplee, Pocket Volleyball podrá analizar diferentes directorios:

En sistemas GNU/Linux

Existen dos locaciones por defecto:

- » `$DATADIR/characters` - donde residen los personajes que forman parte del repertorio oficial.
 - » `$HOME/.volleyball/characters/` - se utiliza para incluir personajes adicionales, ya sea propios o descargados de la Web. Estos personajes solo estarán disponibles para el usuario residente en `$HOME`.
- Cabe destacar que las variables `HOME` y `DATADIR` varían conforme a cada equipo y usuario. `HOME` será su

directorio principal, en mi caso: HOME es "/home/hugo". DATADIR, en cambio, apuntará al directorio donde ha instalado el videojuego, por ejemplo DATADIR podría valer "/usr/share/volleyball".

El proceso de búsqueda se notifica en la misma terminal:

```
hugo@sarge:~/proyectos/juegos/volleyball-0.81/src$ ./volleyball

Starting: Pocket Volleyball (volleyball 0.82)
Copyright (C) 2005-2006 Hugo Ruscitti
Web site: http://www.losersjuegos.com.ar

This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.

Remember run this program with --help for more options

Searching characters in `~/home/hugo/test2//share/volleyball//characters`
    pingu: OK
    tux: OK
    test: ERROR
    can't find 'speed' option
```

En este caso se han encontrado 3 personajes, pingu y tux funcionarán sin problemas, mientras que test no podrá seleccionarse ya que contiene un error.

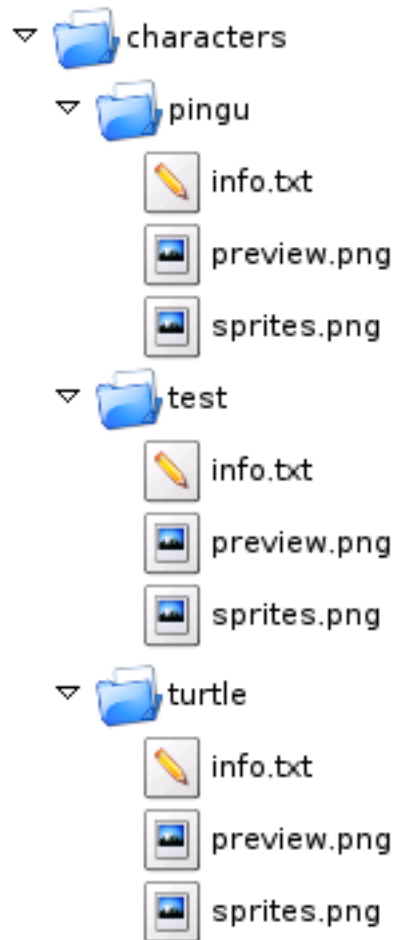
En sistemas Windows

En los sistemas Windows se utilizará el directorio principal del programa, donde residen los archivos ejecutables. Si instaló el videojuego en "c:\volleyball" los personajes se buscaran en "c:\volleyball\data\characters".

Para observar la salida del programa deberá abrir y examinar el archivo stdout.txt que se generará al ejecutar el programa.

Estructura de un personaje

Cada personaje se almacenará en su propio directorio, siempre dentro de "characters":



Árbol de directorios

La imagen muestra la estructura del directorio `characters` que incluye 3 personajes: `pingu`, `test` y `turtle`. Note que cada directorio incluye 3 archivos obligatorios:

preview.png

Debe ser un gráfico de tamaño exacto, 64x64 píxeles, y de fondo transparente2.



preview.png

Este gráfico se utilizará para representar al personaje en la pantalla de selección, en la barra de puntajes y en la pantalla de créditos.

sprites.png

`sprites.png` contiene todos los cuadros de animación del personaje en sentido hacia la derecha.



Si quiere, puede acceder al fichero en tu tamaño original.

Todos los cuadros de animación deben tener la misma longitud horizontal, como si se tratara de una grilla o vector.

Puede obtener algunos consejos para desarrollar la grilla mas adelante en el documento.

info.txt

Dado que cada personaje tiene sus propias características, como por ejemplo su velocidad, existe un archivo de texto donde se especifican todas estas características.

La estructura del archivo contiene atributos y valores, separados por una igualdad (=):

```
# Character
info_version = 1
nick = colo
description = a freak

# Some date of author
author = Walter Velazquez
email = wgv_tattoo@yahoo.com.ar
web = www.losersjuegos.com.ar

# Sprites drawing
frames = 10
delay = 7

# Collisions
hotspot_x = 45
hotspot_y = 137
collision_x = 50
collision_y = 30
ratio = 25

# Logic
width = 40
jump_power = 25
speed = 2

# Animation states
stand = 5
foward = 3 5 4 5
backward = 3 5 4 5
```

```
jump = 6
select = 3 5 4 5

touch_back = 0 0
touch_up = 1 1
touch_front = 2 2
touch_back_in_air = 7 7
touch_up_in_air = 8 8
touch_front_in_air = 9 9
```

El recuadro anterior es un extracto del archivo info.txt de uno de los personajes incluidos en el juego. Note que las líneas comenzadas por # (numeral) representan comentarios y no se procesarán.

Encontrará una descripción detallada de cada atributo al final de este artículo.

Creando a nuestro personaje

Dado que ya contamos con los cuadros de animación de un personaje, nos dedicaremos a completar una serie de pasos:

- » Crear un directorio dentro de characters.
- » Armar una imagen de 64x64 y llamarla preview.png.
- » Generar la grilla de celdas con los cuadros de animación y guardar los resultados como sprites.png.
- » Armar el archivo info.txt con los atributos del personaje.

Crear el directorio

El primer paso es trivial, si utiliza GNU/Linux y ha ejecutado el videojuego, encontrará un directorio llamado .volleyball en su directorio principal:

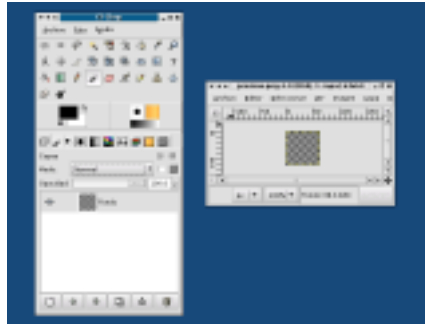
```
cd ~
cd .volleyball
cd characters
mkdir colo
```

Si alguno de los directorios no existe, no se preocupe, ejecute mkdir con cada uno de ellos y regrese al paso anterior.

```
mkdir .volleyball
mkdir .volleyball/characters
```

Generar preview.png

Luego podría crear la imagen del cuadro de selección, preview.png. Para ello le resultará muy útil el programa que utilizamos mas adelante en este artículo: El Gimp!:



Puede generar la imagen agregando parte del personaje, como el rostro, a una imagen de tamaño fijo (64x64 pixels). Luego deberá guardar la imagen como preview.png dentro del directorio que creó anteriormente.



preview.png

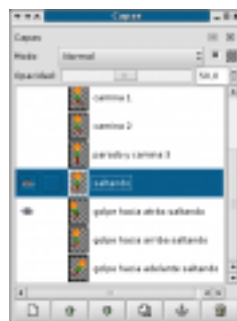
La imagen muestra nuestro resultado, note que utilicé como fondo el color (r = 255, g = 0, b = 255), como representación de transparencia.

Generar sprites.png

Crear la grilla de cuadros es algo mas complicado, aquí hemos partido de una grilla completa, aunque podría no ser su caso.

Además debe tener en cuenta que cada cuadro de grilla está alineado y utiliza las mismas dimensiones que el resto. Encontrará muchas maneras de realizar esta última tarea, aunque en este artículo veremos como solucionarlo con El Gimp! mediante capas:

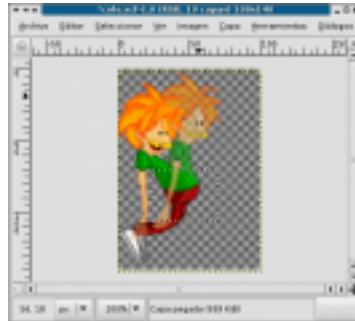
Antes de generar la grilla completa será conveniente separar los cuadros de animación en diferentes capas para poder alinearlos:



En la primer imagen podemos observar como se puede organizar un conjunto de capas. La segunda imagen nos muestra como se verá nuestra imagen si dejamos visibles 2 capas y aplicamos transparencia a la primera.

La utilidad primordial de dividir los cuadros de animación en capas es poder compararlos. Note que cualquier

animación requerirá que sus cuadros estén alineados con precisión, de otra manera el movimiento del personaje en pantalla presentará algunas inconsistencias de movimiento muy molestas. Veamos un ejemplo de capas correctamente alineadas:



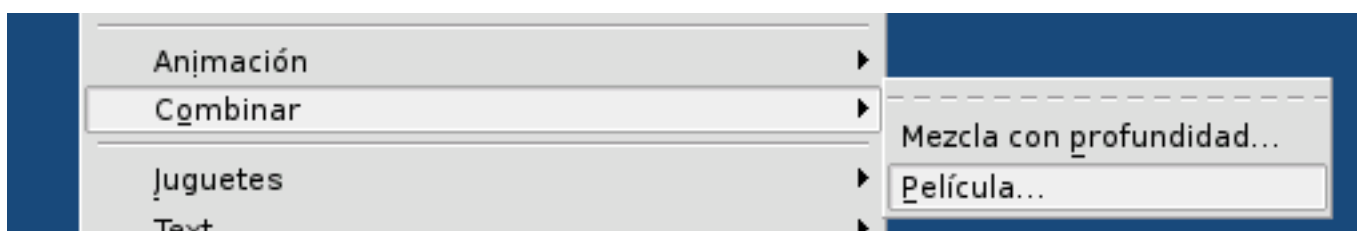
A modo de ejemplo, le recomiendo descargar y analizar estos archivos (colo.xcf y pingu.xcf) con el programa Gimp.

Una vez que tenga todos los cuadros alineados de manera correcta, necesitará "unificar" el tamaño de cada capa:



Realizamos esta tarea porque buscamos aplicar un filtro que nos permitirá generar una grilla de cuadros completa. No podremos aplicar el filtro si alguna de las capas tiene un tamaño diferente al resto.

El siguiente paso será convertir la animación de capas a una imagen dividida en grillas. Para ello utilizamos el filtro combinar -> película:



Luego definimos una serie de valores como indican las imágenes:



Las opciones mas importantes son Ajustar altura a las imágenes y reducir las barras de la solapa Avanzado cero, exceptuando a la primera (altura de la imagen).

Como resultado obtendrá una grilla de animación como esperábamos.

Ahora solo restará definir el color de transparencia para la imagen y guardar los cambios como sprites.png

Generar info.txt

Abra su editor de textos favorito y genere el archivo info.txt, dentro del mismo directorio del personaje.

La primer sección del archivo define datos del autor y el personaje en general:

- » info_version: La versión del archivo de configuración, este valor suele ser 1, aunque posiblemente en futuras versiones de Pocket Volley podamos cambiar a otra versión. Si no está seguro utilice 1 como valor.
- » nick: Es el identificador del personaje, en este caso lo llamaremos igual que su carpeta: "colo".
- » description: Un breve comentario acerca del personaje, o una descripción de su origen.
- » author: El nombre del autor que ha creado el personaje o lo adaptó al juego.
- » email: Dirección de correo electrónico del autor.
- » web: El sitio web del autor, o bien la dirección desde donde se puede obtener mas información.

En este caso los valores nos han quedado:

```
# Character
info_version = 1
nick = colo
description = a freak

# Some date of author
author = Walter Velazquez
email = wgv_tattoo@yahoo.com.ar
web = www.losersjuegos.com.ar
```

En la siguiente sección especificamos cómo se procesará la imagen sprites.png:

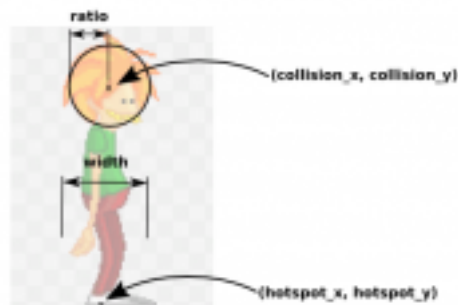
- » frames: Cantidad de columnas de la grilla
- » delay: pausa entre cada cuadro de animación

En nuestro caso ha quedado:

```
# Sprites drawing
frames = 10
delay = 7
```

Luego tendremos que especificar el área de colisión y los puntos de control.

Dado que los atributos de colisión se encuentran muy relacionados con la forma del personaje, he incluido una imagen que resume los valores a evaluar:



Las colisiones entre el jugador y la pelota se determinan mediante una circunferencia, de centro (collision_x, collision_x) y de radio igual a "ratio". El punto de control (hotspot_x, hotspot_y) determina el eje del personaje, debe elegir un punto centrado en la figura y cercano al suelo. Por último se encuentra el valor "width" que especifica el ancho del personaje.

Importante: Ejecute el videojuego con el parámetro -d y podrá corroborar de manera visual si sus parámetros de colisión reflejan lo que buscaba.

Evaluando la imagen obtuvimos:

```
# Collisions
hotspot_x = 45
hotspot_y = 137
collision_x = 50
collision_y = 30
width = 40
ratio = 25
```

Luego tiene la posibilidad de alterar:

- » jump_power: Fuerza del salto, por defecto se utiliza el valor 25
- » speed: Velocidad del personaje al caminar

```
# Logic
```

```
jump_power = 25
speed = 2
```

Por último, definimos los cuadros de animación para cada acción del personaje, para ello utilizamos los índices de las columnas en la grilla sprites.png.

Por ejemplo:

```
foward = 0 1 2 3
```

Define la animación de "caminar hacia adelante".

- » stand: Mientras se encuentra parado en el lugar
- » foward: Camina hacia adelante
- » backward: Camina hacia atrás
- » jump: Mientras salta
- » select: Animación a mostrar en la escena de "selección de personajes"
- » touch_back: Golpea la pelota de espalda
- » touch_up: Golpea la pelota desde abajo
- » touch_front: Golpea la pelota de frente
- » touch_back_in_air: Como touch_back pero mientras salta
- » touch_up_in_air: Como touch_up pero mientras salta
- » touch_front_in_air: Como touch_front pero mientras salta

Lo último, lo que añadimos al archivo ha quedado:

```
# Animation states
stand = 5
foward = 3 5 4 5
backward = 3 5 4 5
jump = 6
select = 3 5 4 5

touch_back = 0 0
touch_up = 1 1
touch_front = 2 2
touch_back_in_air = 7 7
touch_up_in_air = 8 8
touch_front_in_air = 9 9
```

¿ Como puedo contribuir ?

Simplemente comprima el directorio del personaje (en tar.gz o zip), y envíelo a mi casilla de correo electrónico: hugoruscitti_arroba_gmail_dot_com (cambie _arroba_ y _dot_ por @ y . respectivamente).

A medida que reciba contribuciones para el juego las iré clasificando en la página principal del juego. De esta forma otros usuarios tendrán la oportunidad de extender el repertorio de personajes del juego.

Si tiene alguna consulta o sugerencia, no dude en comunicarse con nosotros. Gracias.

Notas

1 - En principio, Walter había desarrollado el personaje para otro videojuego que llamaría 2 slow 2 idiots (en alusión a la película 2 fast 2 furious). Lamentablemente, ese videojuego no se desarrolló. (ver origen de cita).

2 - No me haga caso, era solo una idea (ver origen de cita).

3 - El color (r = 255, g = 0, b = 255) será interpretado como transparente durante la ejecución del juego (ver origen de cita).

Este documento ha sido generado automáticamente a partir del archivo 'crear_personaje.xml' el Mon Jan 19 20:42:44 2009

La versión mas reciente de este documento se almacena en www.losersjuegos.com.ar. Visitenos para obtener mas recursos y actualizaciones.