

Como crear un contador sencillo para nuestros juegos

Créditos

- » **Autor:** Arturo Luna Contreras (sofoke)
- » **Fecha:** Enero del 2008

Resumen

Este es un pequeño tutorial de como crear con SDL y SDL_TTF un contador sencillo para nuestros juegos... Para este tutorial asumiré que el usuario ya tiene instalado SDL y todo lo necesario para leer fuentes trueype... También para hacerlo un poco más entendible para muchos lo realizaremos en C...

Un pequeño ejemplo de un contador de tiempo es:



Como podemos notar este se encuentra en la parte superior de un juego; y nos indica el tiempo restante para que termine el tiempo limite (en este caso la selección de la aeronave).

Comenzando...

Tener en cuenta que hay que iniciar TTF antes con `InitTTF()`; Y cerrarlo con `TTF_Quit()`; o el clásico `atexit(TTF_Quit)`; para que se cierre cuando ya no se ocupe...

Podemos crearlo de muchas maneras pero en esta ocasión lo realizaremos en dos archivos diferentes, el primero se llama 'contador.h' e incluye declaraciones, el segundo es 'contador.c' y tiene todo el código antes declarador. Esta distribución nos permitirá usarlo para futuros proyectos.

Lo primero es crear la estructura básica del archivo 'contador.h':

```
#ifndef __TIMER_H__
#define __TIMER_H__

/* código completo */

#endif
```

Estas líneas sirven para indicarle al compilador como trabajar con nuestro fichero, y no procesarlo mas de una vez en un proyecto mas grande.

Analicemos las dos primeras líneas: el `#ifndef __TIMER_H__` dice que si: `timer.h` (el nombre del archivo que estamos creando) no esta definido se defina mediante: `#define __TIMER_H__...` ahora lo que sigue a estas dos líneas es el código completo (que se ha omitido aquí, lo veremos luego), luego el `#endif` solo indica que es el fin del archivo.

Necesitaremos dos variables de tipo `SDL_Color` una para el color de la letra y otra para el fondo de la letra...

Ten en cuenta que aquí he utilizado variables globales para que resulte mas sencillo tener el primer ejemplo funcionando, y no entrar en otros detalles...

Necesitamos una variable de tipo `SDL_Surface` (superficie de SDL) aunque no es estrictamente necesario (se recomienda al lector que analicé el código terminado y pruebe a disminuir líneas lo más que pueda, para ahorrar recursos, espacio, etc...).

También ocuparemos una variable para la posición del contador de tipo `SDL_Rect` una variable de tipo `char` para la cadena (los números serán tratados como cadenas) dos variables tipo `int` para la posición en (x, y) de la pantalla y otras tres del mismo tipo que explicare después...

Por último (pero no menos importante) es la variable tipo `TTF_Font` para cargar la fuente (es apuntador).

Ahora las variables ya declaradas quedarían de la siguiente manera:

```
SDL_Color bgcolor, fgcolor;
SDL_Surface *TimerSDL;
SDL_Rect PosTimerRect;
TTF_Font *FuenteTimer;
char Num[1];
int Posx, Posy, Timer, y;
```

Ahora declararemos las funciones que vamos a usar primero la que va inicializar todo lo que necesitamos quedaría de la siguiente manera:

```
void InitTimerSDL( char FontTimer[40], int Tam, int x, int y, int Ini );
```

Analicemos: el `void` significa que no retorna nada... el argumento `FontTimer[40]` se encarga de indicar la ruta de la fuente que se va a usar; `'Tam'` es para tamaño de la fuente a usar, `'x'` e `'y'` es para la posición en pantalla e `'Ini'` es para el número con el que se desea empezar el contador.

Ahora el contenido de esa función sería:

```
void InitTimerSDL( char FontTimer[40], int Tam, int x, int y, int Ini )
{
    Timer = Ini;
    Posx = x;
    Posy = y;
    FuenteTimer = TTF_OpenFont(FontTimer, Tam);
    fgcolor.r = 900;
    fgcolor.g = 900;
    fgcolor.b = 900;
    bgcolor.r = 900;
    bgcolor.g = 0;
    bgcolor.b = 0;
}
```

Creo que lo único a explicar aquí es: que para las variables 'fgcolor' y 'bgcolor', los atributos 'r', 'g', y 'b' son los que representan el tono de color a utilizar (R{ed}, G{reen}, B{blue}) (se recomienda al lector interactuar con los valores para obtener colores diferentes). 'bgcolor' es para el color de fondo y 'fgcolor' es para el color del texto. TTF_OpenFont es para cargar la fuente, la ruta al archivo de la fuente se indica por 'FontTimer' y el tamaño por 'Tam', luego todo se guarda en FuenteTimer.

Ahora declararemos la función encargada de cambiar disminuir (en este caso) los números a mostrar.

```
void CambioTimer( void );
```

Bueno esto no es muy difícil solo la declara. Ahora esto es lo que contiene:

```
void CambioTimer( void )
{
    if( !( y%20 ) )
    {
        Timer--;
    }
    y++;
}
```

Es algo enredado, pero es la forma más resumida que encontré. Primero el `if(!(y%20)` es el encargado junto a `y++` de restarle -1 a Timer cada 20 ciclos del juego (hay que acomodar esto según se quiera que avanza el tiempo).

Ahora hay que declarar y usar la función que se encargara de imprimir todo en pantalla:

```
void DrawTimer( SDL_Surface* Screen );
```

Ahora su código:

```
void DrawTimer( SDL_Surface* Screen )
{
    CambioTimer();
    sprintf( Num, "%d", Timer );
}
```

```
TimerSDL = TTF_RenderText_Shaded( FuenteTimer, Num, fgcolor, bgcolor );
PostimerS.x = Posx;
PostimerS.y = Posy;
PostimerS.w = TimerSDL->w;
PostimerS.h = TimerSDL->h;
SDL_SetColorKey( TimerSDL, SDL_SRCCOLORKEY | SDL_RLEACCEL,
SDL_MapRGB(TimerSDL->format, 255, 0, 0));
SDL_BlitSurface( TimerSDL, NULL, Screen, &PostimerS );
}
```

Ahora veamos una a una las sentencias:

```
sprintf( Num,"%d", Timer );
```

se encarga de que el valor de Timer sea copiado a la variable Num, pero considerando que 'Num' es una cadena y 'Timer' un número entero.

```
TimerSDL = TTF_RenderText_Shaded( FuenteTimer, Num, fgcolor, bgcolor );
```

se encarga de generar una superficie y apuntarlo con la variable de tipo SDL_Surface. Recibe la fuente, el tamaño, el color de la letra y el color de fondo.

```
PostimerS.x = Posx;
PostimerS.y = Posy;
PostimerS.w = TimerSDL -> w;
PostimerS.h = TimerSDL -> h;
```

Se encarga de definir la posición y el tamaño de la superficie que se va a ocupar.

```
SDL_SetColorKey( TimerSDL, SDL_SRCCOLORKEY | SDL_RLEACCEL, SDL_MapRGB( TimerSDL ->
format, 255, 0, 0) );
```

Esta parte es la que se encarga de hacer en verdad transparentes los píxeles que no se quieren mostrar... si no se tiene esa línea podríamos obtener un resultado parecido al siguiente:



```
SDL_BlitSurface( TimerSDL, NULL, Screen, &PosTimerS );
```

Como se imaginan esta última parte es solo para volcarlo a la pantalla, se le pasa la superficie (TimerSDL), NULL para indicar que tome toda la región de la imagen, Screen es la superficie principal, y el &PosTimerS es la posición de impresión destino.

También no olvidemos que debemos invocar a una función para liberar los recursos ocupados. Para ello hay que escribir:

```
TTF_CloseFont( Fuente );
```

Ahora tenemos terminada nuestra mini-biblioteca para la creación de un contador. El código completo a continuación:

contador.h

```
#ifndef __TIMER_H__
#define __TIMER_H__

SDL_Color bgcolor, fgcolor;
SDL_Surface *TimerSDL;
SDL_Rect PosTimerRect;
TTF_Font *FuenteTimer;
char Num[1];
int Posx, Posy, Timer, y;

void InitTimerSDL( char FontTimer[40], int Tam, int x, int y, int Ini );
void CambioTimer( void );
void DrawTimer( SDL_Surface* Screen );

#endif
```

contador.c

```
#include "contador.h"

void InitTimerSDL( char FontTimer[40], int Tam, int x, int y, int Ini )
{
    Timer = Ini;
    Posx = x;
    Posy = y;
    FuenteTimer = TTF_OpenFont( FontTimer, Tam );
    fgcolor.r = 900;
    fgcolor.g = 900;
    fgcolor.b = 900;
    bgcolor.r = 900;
    bgcolor.g = 0;
    bgcolor.b = 0;
}

void CambioTimer( void )
{
    if( !( y%20 ) )
    {
        Timer--;
    }
    y++;
}

void DrawTimer( SDL_Surface* Screen )
{
    CambioTimer();
    sprintf( Num,"%d", Timer );
    TimerSDL = TTF_RenderText_Shaded( FuenteTimer, Num, fgcolor, bgcolor );
    PosTimerS.x = Posx;
```

```
PosTimerS.y = Posy;
PosTimerS.w = TimerSDL -> w;
PosTimerS.h = TimerSDL -> h;
SDL_SetColorKey( TimerSDL, SDL_SRCCOLORKEY | SDL_RLEACCEL, SDL_MapRGB( TimerSDL
-> format, 255, 0, 0) );
SDL_Blitsurface( TimerSDL, NULL, Screen, &PosTimerS );
}
```

Ahora, un ejemplo de utilización podría ser así:

```
#include "timer.h"

/* inicialización de SDL y creación de variable Screen */

InitTimerSDL("font.ttf", 20, 10, 10, 300);
DrawTimer(Screen);
SDL_Flip(Screen);
```



Este documento ha sido generado automáticamente a partir del archivo 'contador.xml' el Mon Jan 26 17:15:16 2009

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