

## ¿Por que SDL?

### Créditos

- » **Autor:** Bob Pendleton (Robert C. Pendleton)
- » **Traductor:** Daniela López Seco.
- » **Fecha:** 4 de Octubre del 2005

### Portable, fácil (y así) poderoso

Como escribí en Agosto del 2002, estuve usando SDL (Simple DirectMedia Layer) casi un año. Antes de elegir SDL dedique años de mi tiempo libre mirando diferentes bibliotecas de desarrollo de juegos y probándolas.

Estaba buscando una biblioteca que tuviera una licencia razonable. Una biblioteca que me permitiera escribir código sin tener que preocuparme demasiado por asuntos legales y no me forzara a pagar una pequeña fortuna en derechos si desarrollaba algo que tuviera algún valor comercial. También quería una biblioteca que me permitiera escribir código para un gran rango de dispositivos y sistemas operativos. Con el crecimiento de las computadoras de mano y los teléfonos celulares con gráficos no parece razonable atar el desarrollo a un solo sistema operativo o arquitectura. Después de buscar mucho tiempo (admito que estaba buscando mas como un hobby que por negocios) encontré una lista de 3 sistemas que lucían interesantes.

El primero era Java. A mi me gusta Java como lenguaje y como sistema. Tiene todo lo que necesitas para escribir grandes juegos. Pero no esta disponible en todos lados y me dí cuenta que esperar que la gente bajara e instalara 20 (o mas) megas de código en su computadora antes de poder jugar mi juegos, limitaría seriamente la distribución online de cualquier juego que escribiera. También miré de cerca el nuevo sistema multimedia portable de Amiga. Pero, cuanto mas trabajaba con el, menos me gustaba. Después de dar una mirada rápida a varias bibliotecas libres para desarrollo de juegos decidí mirar detenidamente SDL. Lo que descubrí de SDL fue una sorpresa. A diferencia de la mayoría de las bibliotecas libres para desarrollo de juegos que se encuentran en la Web, SDL fue diseñada y escrita por un grupo de programadores de juegos experimentados y altamente profesionales. No por desprestigiar a los autores de las bibliotecas competidoras, pero SDL es soportada por una gran cantidad de buenos programadores. SDL fue escrito por programadores de juegos, para programadores de juegos.

Creo que lo que más me atrajo tanto de SDL cuando recién había empezado a leer sobre ella es que no trata de hacer todo. Solo hace las cosas que todos los juegos tienen que hacer en cualquier maquina o sistema operativos y después deja el camino libre para permitirte codificar. Deja que tomes un programa que escribiste para Windows y lo recompiles para que corra en Linux, o Mac, hasta en palms y viceversa. Te provee lo que necesitas mientras esconde toneladas de detalles. Si lo corres en Windows, SDL se apoya sobre DirectX. Si lo corres en Linux, SDL se apoya sobre X y otras bibliotecas. La clave es "simplemente funciona". Pase varios meses escribiendo programas de prueba tanto como una manera de aprender SDL y para tratar de encontrar bugs en SDL. En todo ese tiempo solo encontré un par de documentos mínimos acerca de bugs y lo que yo llamaría faltantes, los cuales son fáciles de cubrir. Trate de hacer cosas ridículas con SDL, como escribir un programa que usaba 10000 timers, y no fui capaz que romperlo. Cada vez que encontraba algo que pensaba que era un bug, resultaba ser o un resultado de errores en la documentación o mi mala interpretación de la documentación. La única vez que creí encontrar un bug serio en SDL resultó que ya había sido corregido.

SDL es pequeño, portable y confiable. Hace lo que necesitas y después deja el camino libre. La filosofía básica de SDL es una combinación de flaco y avaro (lean and mean) y mantenerlo simple (Keep it Simple).

## ¿Que se obtiene de SDL?

### Comunidad

La comunidad de SDL es una de las comunidades más positivas y más responsables que he visto en la Red. Dirigida por Sam Latinga con la ayuda de un grupo de voluntarios dedicados, la comunidad crea un flujo estable de nuevos proyectos, nuevos agregados a SDL, continuo mejoramiento del código, y soportes de SDL para nuevas plataformas. La lista de correo de SDL es una fuente constante de información acerca de SDL y de programación de juegos en general con un mínimo de peleas y discusiones. Los archivos de la lista de correo representan un recurso enorme, lleno de información sobre como usar SDL para resolver problemas al programar juegos.

### Iniciar y terminar

Todos los sistemas operativos hacen que el programador escriba algo de código para averiguar que es lo que el sistema operativo te deja hacer y lo que el hardware puede hacer. Después, hay que escribir mas código para preparar las cosas del modo que las quieres. Hay que establecer los modos gráficos, inicializar el sistema de sonido, chequear si hay mouse o joystick, etc, etc. Luego, cuando el juego esta terminado usualmente hay que ordenar y decirle al sistema operativo que terminaste de usar cualquier cosa que estuvieras usando. Toda esta codificación es casi igual en todos los proyectos. SDL te da una manera simple de descubrir que es lo que el hardware puede hacer y después, con par de líneas de código podes dejar todo listo. Limpiar al final necesita aun menos código. Y la pequeña cantidad de código de SDL que necesitas, puede ser copiada y reutilizada en proyecto tras proyecto.

### Eventos basados en Entrada

SDL provee entrada por teclado, mouse y joystick, usando un modelo basado en eventos que será familiar para todos los programadores que hayan escrito código en X11, Windows, o Mac OS. La diferencia es que SDL te dará los mismos eventos sin importar en que sistema operativo lo estés usando y SDL hace que no tengas que manejarte con todos los eventos específicos del sistema operativo que solo complican el trabajo de escribir código.

### Tiempo y relojes

Todos los videojuegos necesitan saber que hora es y necesitan ser notificados cuando hay que realizar una acción. SDL provee una API de tiempo y relojes sencilla, limpia y confiable que es independiente de la máquina y del sistema operativo. Las APIs de tiempo de SDL permiten la creación de miles de relojes y pueden ser usadas sin las preocupaciones de otras APIs de tiempo que usé en el pasado.

### Sonido

SDL provee una API simple para averiguar las capacidades de la placa de sonido y todo lo que se necesita para producir sonido de baja latencia en cualquier dispositivo. Aunque el soporte es mínimo, cubre todo lo que realmente se necesita para generar sonido. En este sentido concuerda con la filosofía Flaco y Avaro (lean and

mean) de SDL. Por otro lado, existen bibliotecas de sonido de alto nivel para SDL, y de la manera en la que trabaja la comunidad de SDL, hay bibliotecas nuevas apareciendo todo el tiempo.

## **CDROM**

SDL provee una API para leer y reproducir pistas en un CDROM. No se puede superar.

## **Gráficos**

SDL te deja decidir como hacer los gráficos. Podes empezar con píxeles crudos y hacer casi todo lo que quieras con ellos, o puedes usar OpenGL para hacer gráficos 2D y 3D acelerados por hardware. El primer método funciona en cualquier dispositivo, y como OpenGL esta disponible para todas las placas de video 3D, es una manera segura y portable de hacer gráficos 3D de alto rendimiento en un dispositivo, independientemente del sistema operativo. SDL se ocupa de configurar los modos de video, acceder a la memoria de video, si lo usas, e inicializar OpenGL. SDL te ahorra el sufrimiento de preparar gráficos de pantalla completa, tan usados en juegos, y también te permite preparar juegos en ventanas.

## **Red**

SDL incluye una API para trabajar en red de bajo nivel, que permite iniciar y controlar sockets TCP/IP y UDP/IP. La API es similar a las APIs para trabajar en red de Unix y Windows. La diferencia es que solo te permite usar utilidades que son comunes a todos los sistemas operativos que soporta. Y, la API se ocupa de muchas tareas molestas de bajo nivel que hacen que preparar y manejar las conexiones sea tan tedioso.

## **Hilos**

SDL provee una API de hilos que luce como una versión simplificada de pthreads. SDL provee toda la funcionalidad básica que un programador necesita en un paquete de hilos, mientras que esconde los detalles de bajo nivel que confunden a mucha gente y hacen que el desarrollo de código con hilos sea tan difícil. Los hilos de SDL se soportan en todos los sistemas operativos que soporten hilos y SDL.

## **¿Cuales son las desventajas?**

La única desventaja que pude encontrar - soy tan cínico que busque las desventajas primero - es la documentación. Encontré unos pocos errores y alguna de la documentación es algo escasa. Pero, es el pequeño precio que hay que pagar por esta gran biblioteca. Los problemas con la documentación son más que compensados con la habilidad de leer y leer el código de programas de ejemplo escritos por varios profesionales.

Algunas personas consideraran a las reglas que hay que seguir para obedecer la licencia LGPL como una desventaja. Yo no considero que lo sea. Obedecer las reglas de LGPL es simple y agrega poco o nada al costo del desarrollo o al costo de soportar un juego.

## ¿Cuanto cuesta?

SDL es libre. Es libre como la cerveza y es libre como la libertad. SDL se distribuye bajo LGPL que requiere que sigas algunas reglas simples para asegurar que todo el mundo tenga acceso libre a SDL. Mientras sigas esas reglas, puedes usar SDL.

## ¿Donde lo puedo conseguir?

[www.libsdl.org](http://www.libsdl.org) es la página principal y el punto de distribución. Todo lo que necesitas para empezar con SDL esta referenciado desde el sitio. Empezá por leer el archivo de las FAQ de la versión de SDL con la que quieres empezar y continua explorando el sitio desde ahí. Si no puedes encontrar lo que estas buscando usa la función de búsqueda de la página.

## Que gano con esto?

Para los programadores profesionales de juegos, SDL provee una biblioteca limpia que te puede ahorrar tiempo de desarrollo, incrementar la confiabilidad del producto, y darte la oportunidad de vender tu producto para más de un tipo de máquina o sistema operativo. Pero además de todas las ventajas que SDL le da a los programadores profesionales, los que mas ganan son los programadores que recién empiezan, o los que lo hacen por hobby, porque SDL te permite concentrarte en escribir el juego, mas que en los detalles del sistema operativo. Y, porque puedes empezar a usar SDL sin tener que gastar cientos o incluso miles de dólares en herramientas de desarrollo de software.

Todos los días, recibo pedidos en mi lista de correo o en mi correo privado de gente de todo el mundo que quiere escribir juegos, pero no tiene el dinero para comprar herramientas de desarrollo de software comerciales. Usando SDL y Linux + GCC + OpenGL se puede desarrollar un juego sin tener que gastar ni un dólar en software de desarrollo. Hasta en Windows, se puede usar SDL + GCC + OpenGL para escribir juegos de alta performance sin tener que pagar nada, más que tu computadora y la copia de Windows que corres (Nota del Traductor: jua!). SDL permite al programador principiante y al que lo hace por hobby, desarrollar juegos usando herramientas libres y después, si desarrollan algo con potencial comercial, migrarlo a otro sistema operativo como Windows o MAC OS, y llegar a una audiencia mayor. No solo el código es portable, sino también las capacidades que desarrolla.

En otras palabras, SDL te puede ahorrar tiempo y dinero y ayudarte llegar a una audiencia mayor con tus juegos.

---

*Este documento ha sido generado automáticamente a partir del archivo 'why\_sdl.xml' el Sat Nov 29 13:23:15 2008*

*La versión mas reciente de este documento se almacena en [www.losersjuegos.com.ar](http://www.losersjuegos.com.ar). Visitenos para obtener mas recursos y actualizaciones.*