



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Reproductor multimedia básico con SDL para Sony/PS3

Apellidos, nombre	Agustí i Melchor, Manuel (magusti@disca.upv.es)
Departamento	Departamento de Informática de Sistemas y Computadores
Centro	Universidad Politécnica de Valencia



1 Resumen de las ideas clave

En este artículo vamos a presentar las características básicas de una aplicación que de servicio de reproductor multimedia básico a la plataforma *Sony PlayStation 3* [1] (en adelante PS3) en la que se ha procedido a instalar una versión de GNU/Linux Ubuntu 9.10 (Karmic) y 10.4 (Lucid). Se quiere aprovechar la capacidad de cómputo de esa plataforma para darle una utilidad más allá de servir de plataforma de juegos y se ha pensado en que pueda mostrar y reproducir una variedad de ficheros de diferentes media. Además se hará uso del mando a distancia tradicional de esta plataforma: el *Sixaxis* [2]. Asumimos que ya se dispone de una versión de GNU/Linux operativa instalada en la plataforma PS3.



Figura 1: Fotografía de la plataforma de ejecución (a) y del mando a distancia (b) tomadas de la Wikipedia.

En algunos productos comerciales se utilizan denominaciones como media center, media player o disco multimedia. Nuestra realización tiene un carácter básico, elemental. Esto es muestra cómo acceder a ejemplos de cada tipo de media con una biblioteca de funciones. La elegida ha sido *Simple DirectMedia Layer* [3] (en adelante SDL) por que permite fácilmente su portabilidad a otras plataformas. Así que lo que estamos desarrollando para la PS3 es válido para cualquier otra plataforma basada en computadores y con un sistema operativo instalado que tenga soporte para SDL.

2 Objetivos

Una vez que el alumno haya leído con detenimiento este documento y experimentado con los ejemplos de código, será capaz de:

- Reproducir una serie de formatos de ficheros de imágenes, sonido y vídeo.
- Generar una salida gráfica con visualización de imágenes, vídeo y texto.
- Gestionar los eventos de un dispositivo remoto nativo de la plataforma final escogida.



3 Introducción

Un reproductor multimedia es una herramienta que se utiliza en la mayoría de los dispositivos informáticos tanto portables como fijos. Se decidió crear uno desde cero con SDL que está principalmente enfocada al desarrollo de aplicaciones gráficas y de videojuegos.

SDL es una biblioteca de funciones de naturaleza multiplataforma, libre y de código abierto escrita en C. Ofrece un interfaz para para manipulación de datos multimedia: texto, sonido, gráficos, vídeo y gestión de eventos de dispositivos. Los formatos soportados en la versión 1.2, que ha sido la utilizada aquí, se resumen en la tabla 1.

TIPO	NOMBRES	Formatos soportados
Vídeo	SMPEG	MPG
Imagen	SDL_IMAGE	BMP, JPEG, TIFF, PNG, PNM, PCX, GIF y TGA
Sonido	SDL_MIXER	WAVE, MP3 y OGG
Texto	SDL_TTF	Fuentes <i>TrueType</i>

Tabla 1: Formatos de ficheros soportados por SDL.

Como se ha escogido una consola de videojuegos que no suele llevar un teclado ni ratón, sino algún tipo de mando para los juegos, el tipo de interacción que puede realizar el usuario es ligeramente diferente de la "habitual" en un computador, pero es totalmente habitual para los usuarios de esta plataforma. En este sentido es necesario explicar cómo se ha conseguido el soporte para este dispositivo en la plataforma PS3.

3.1 Instalación del QtSixA

Podemos navegar por el sistema como si de un ratón se tratase con el mando *Sixaxis* de PS3. Para ello hay que instalar los *drivers* necesarios y el programa *qtsixa*. La fig. 2 muestra el aspecto en pantalla de dicho programa en ejecución durante la configuración del mando. Como respuesta a las pulsaciones de los diferentes elementos del mando *SixAxis*, *qtsixa* genera eventos de teclado o de ratón que son asignados por el usuario durante la configuración de *qtsixa*. Con lo que una aplicación que utilice este *driver*, verá una secuencia de eventos típica como la generada por un ratón y un teclado.

Veamos como proceder a su instalación para poder interactuar con el mando de la videoconsola. En su primer paso, el proceso de instalación es ligeramente diferente para cada distribución de GNU/Linux:

- En Ubuntu, tanto en la distribución *Karmic* como *Lucid*, se añade el repositorio mediante la orden:
\$ sudo add-apt-repository ppa:falk-t-j/qtsixa
- Para versiones anteriores (*Jaunty* o *Intrepid*) hay que añadir la URL del PPA, añadiendo a las fuentes de software (cambiando `UBUNTU_VERSION` por "jaunty" o "intrepid" respectivamente):
\$ deb http://ppa.launchpad.net/falk-t-j/qtsixa/ubuntu UBUNTU_VERSION main

Después, en cualquiera de estas versiones, desde un terminal se puede terminar el proceso de instalación con:

```
$ sudo apt-get update && sudo apt-get install qtsixax
```

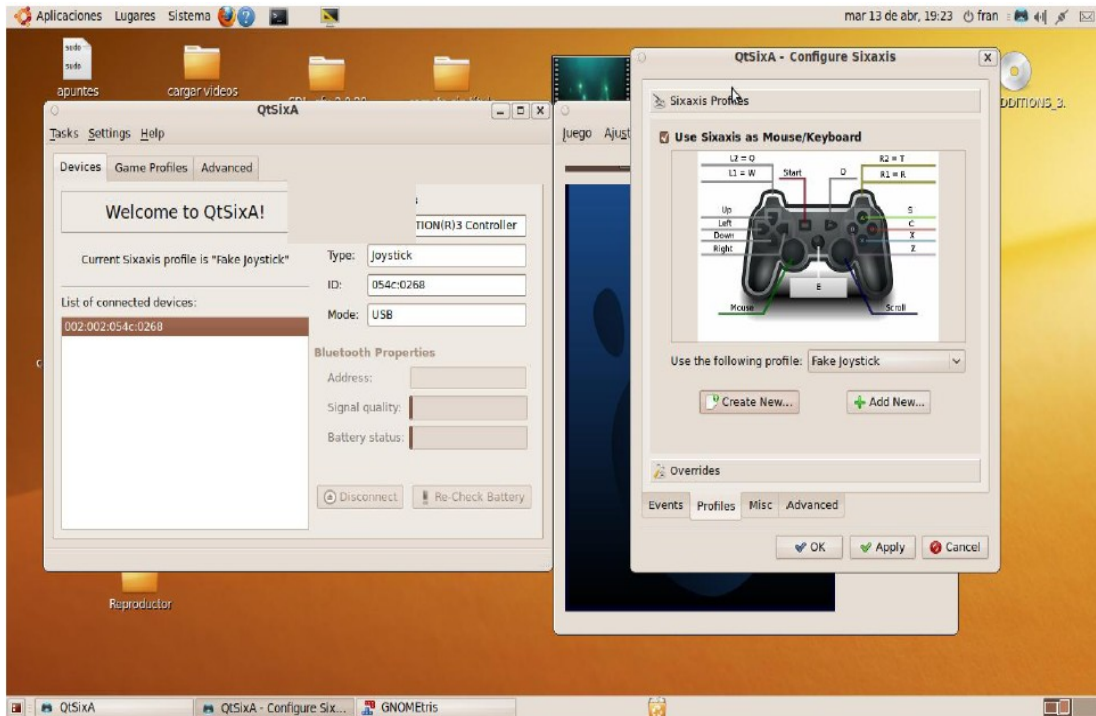


Figura 2: SixAxis en ejecución.

4 Desarrollo

Por brevedad en la exposición, este apartado muestra detalles de programación relativos a las funciones más importantes del proyecto. En concreto, los referidos a: el acceso al sistema de ficheros y la manipulación de los media (audio, imagen y vídeo).

El lector interesado puede encontrar mucha documentación al respecto de SDL, a parte de en el sitio web oficial, en [4, 5, 6, 7 y 8]. Si el lector está interesado en obtener una copia del listado integro del proyecto presentado aquí, no dude en ponerse en contacto con nosotros a través de la dirección de correo electrónico indicada en la portada.

4.1 Acceso al sistema de ficheros

Este acceso se realiza con llamadas al sistema, esto quiere decir que no es parte expresa de SDL y que se puede utilizar en cualquier otra plataforma. El listado 1 muestra un ejemplo de uso. El punto importante es la inclusión del fichero de cabecera *dirent.h* que es el que define las operaciones que permiten listar directorios.

La estructura de datos *struct dirent* sirve para declarar la ruta que deseamos listar. *DIR* representa el contenido de un directorio: es la lista con todas las entradas del mismo: ".", "..", etc. Con *opendir* abrimos el directorio y el ejemplo



recorre todo el directorio, cada entrada se obtiene con *readdir* y se muestra en el terminal. Al terminar se cierra el fichero de directorio con *closedir*.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
int main ()
{
    struct dirent *d;
    const char *direccion="/home";

    DIR *directorio = opendir(direccion);
    printf("Contenido de %s\n\n", direccion);

    while ((d=readdir(directorio)) != NULL) {
        printf("%s/%s\n", direccion, d->d_name);
    }

    closedir(directorio);
    return 0;
}
```

Listado 1. Recorriendo el contenido de un directorio.

4.2 Reproducción y visualización de los media

Veamos un ejemplo de la aplicación desarrollada para la visualización de imágenes y la reproducción de ficheros de audio y vídeo.

4.2.1 Visualizar imágenes

El listado 2 muestra un trozo de la función que hemos desarrollado para pintar la botonera que permite manipular la lista de ficheros de los diferentes medias. En este caso hay que cargar la imagen desde un fichero con *IMG_Load* a una *SDL_Surface*, asignar una paleta de colores opcionalmente y pintarla con *SDL_BlitSurface*. La ventana se refresca con el nuevo contenido con *SDL_Flip*. El listado también muestra como al inicio de la aplicación se ha creado una variable (pantalla) también de tipo *SDL_Surface*. También cómo se inicializa y configura SDL y la variable cuyo contenido define la ventana que se muestra en la pantalla.

```
...
void Dibujar_botones(SDL_Surface *pantalla)
{
    SDL_Rect rect;
    SDL_Surface *imagen;

    if(modo==3){
        imagen = IMG_Load("barra_imagenes.png"); }

    rect.x = 100;
    rect.y = 20;
    rect.w = imagen->w;
    rect.h = imagen->h;
```

```
SDL_SetColorKey(imagen,SDL_SRCCOLORKEY,SDL_MapRGB(imagen->format,0,0,0));
SDL_BlitSurface(imagen,NULL,pantalla,&rect);

SDL_Flip(pantalla);
}
...
int main( int argc; char *argv[])
{
SDL_Surface *pantalla; //declaramos primero la pantalla base
if(SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO) < 0) { exit (-1);}
...
pantalla= SDL_SetVideoMode (800,600,16, SDL_HWSURFACE || SDL_DOUBLEBUF);
...
}
```

Listado 2. Parte de la función para pintar la botonera y del programa principal relativos al tratamiento de imágenes.

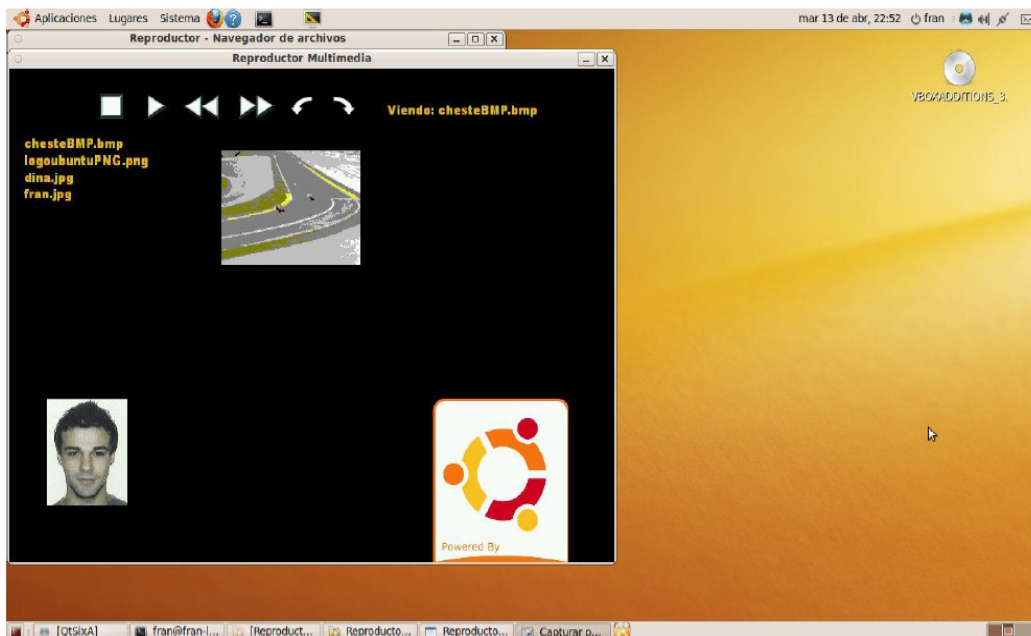


Figura 3: Ejemplo de visualización de imágenes.

En la visualización de imágenes hemos utilizado funciones de SDL como *SDL_rotozoom*, para voltear las imágenes y obtener las miniaturas que se ven debajo de la imagen que se está visualizando en tamaño original.

La fig. 3 muestra una imagen centrada que es la que se está visualizando junto a dos previsualizaciones, las que están abajo, a su derecha e izquierda. Un menú gráfico en la parte superior permite moverse por la lista de imágenes localizadas que se muestra en la parte central izquierda.

4.2.2 Reproducir audio

Nos centramos en la reproducción de ficheros de música, esto es ficheros cuyo contenido va a ser utilizado una vez y que pueden requerir cierta ayuda del sistema para desentrañar el formato en que están guardados en disco y descomprimirlos para su reproducción por el subsistema de audio.

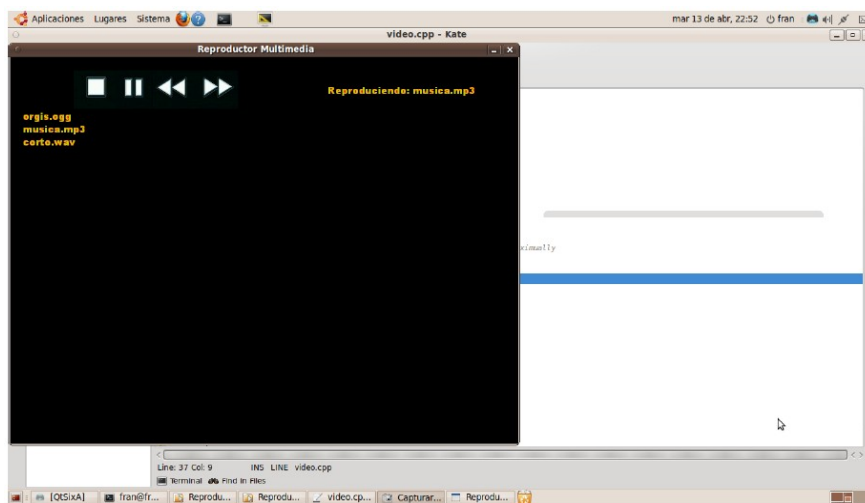


Figura 4: Ejemplo de captura durante la reproducción de un fichero de audio.

```
...
int main (int argc, char *argv[]) {
...
Mix_Music *musica; //declaramos la musica de fondo
...
if (Mix_OpenAudio(22050, AUDIO_S16SYS, 2, 4096) < 0)
// iniciamos el audio a la frecuencia 22khz, 16 bits, 2 canales y un buffer de 4096 (segun el
//sonido poner mas o menos cantidad de buffer)
{ exit(-1); }

musica = Mix_LoadMUS("sonido.mp3"); //no creo q necesite explicacion

Mix_PlayMusic(musica, -1); //hacemos sonar la musica, 0 normal -1 infinito
...
Mix_HaltMusic(); //paramos la musica
Mix_FreeMusic(musica); //liberamos la musica
...
}
```

Listado 3. Ejemplos de uso de funciones relativas al manejo del audio.

La fig. 4 muestra, en la parte superior, la “botonera” que permite realizar acciones típicas de reproducir/parar, pausar, avanzar y retroceder en la lista de ficheros que se muestra debajo de la botonera, en la parte izquierda. En la parte superior derecha se indica el archivo en curso. La lista de ficheros se crea a partir del conjunto de ficheros de un directorio, cuyo formato es reconocido por la aplicación.

El listado 3 muestra un ejemplo de uso de ficheros de audio, empezando por la declaración de una variable de tipo *Mix_Music*; la inicialización del sistema de audio con *Mix_OpenAudio*. En este caso a 22Khz, 16 bits, con 2 canales y un tamaño de *buffer* para almacenar parte de la secuencia de audio en memoria de 4K bits. Se puede cargar desde un fichero MP3 con *Mix_LoadMUS* y lo hacemos sonar con *Mix_PlayMusic*. Al terminar liberamos recursos con *Mix_FreeMusic* y el hardware con *Mix_HaltMusic*.

4.2.3 Reproducir vídeo

Es posible reproducir vídeo en formato MPG con la clase *SDL_Movie* en la que hay que declarar objetos de tipo *SDL_Movie*, que se cargan y asocian a un área en

pantalla con el método *Load*. Se empieza a reproducir con *Play* y se pide que se refresque el área de la pantalla asociada con *SDL_Flip*, como se muestra en el listado 4 y con el resultado que muestra la fig. 5.

```
...  
    SDL_Movie mov1;  
  
    // Load the specified file and draw it to the 'screen' surface  
    mov1.Load( ruta_peliculas, pantalla, 1, 1 );  
...  
    mov1.Play();  
    SDL_Flip( pantalla );  
...
```

Listado 4. Ejemplos de funciones para reproducción de vídeo.

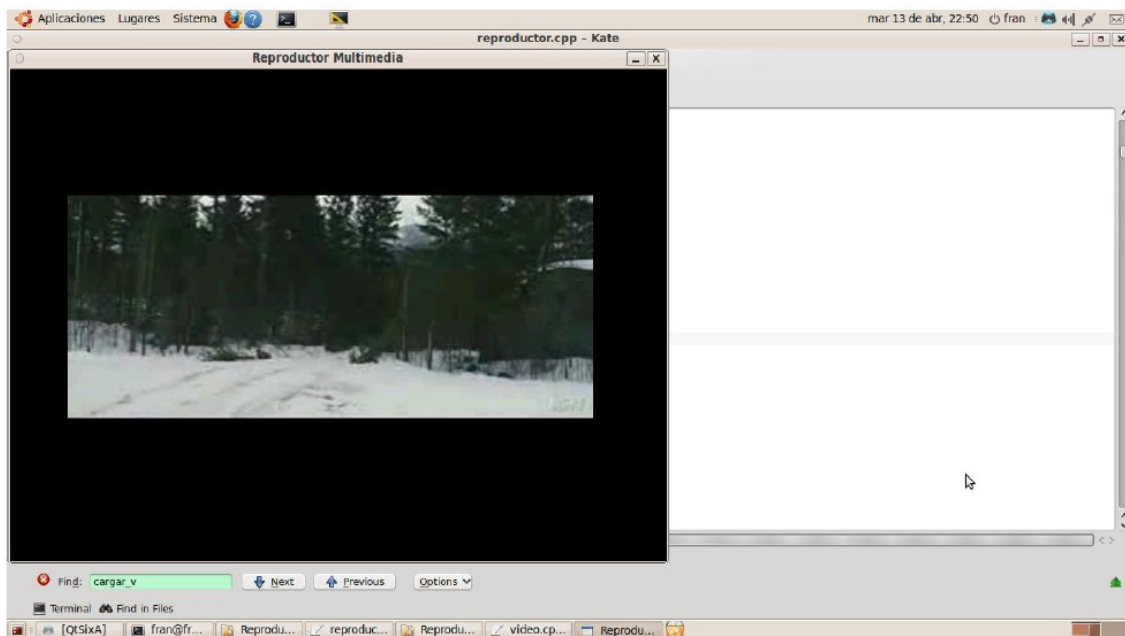


Figura 5: Ejemplo de reproducción de un fichero de vídeo con SDL.

5 Consideraciones para el diseño de la aplicación

La instalación del programa consiste en la del ejecutable resultado de compilar todos estos apartados y los ficheros de imágenes de la botonera para permitir, en un futuro siguiente paso, la personalización del aspecto del programa. Usando simplemente el terminal se puede compilar el programa con la orden:

```
$ gcc -o nombreEjecutable nombreFicheroDeCodigo.c `sdl-config --cflags --libs` - ISDL
```


Como se muestra en la fig. 6, para poder reproducir los distintos archivos de vídeo,

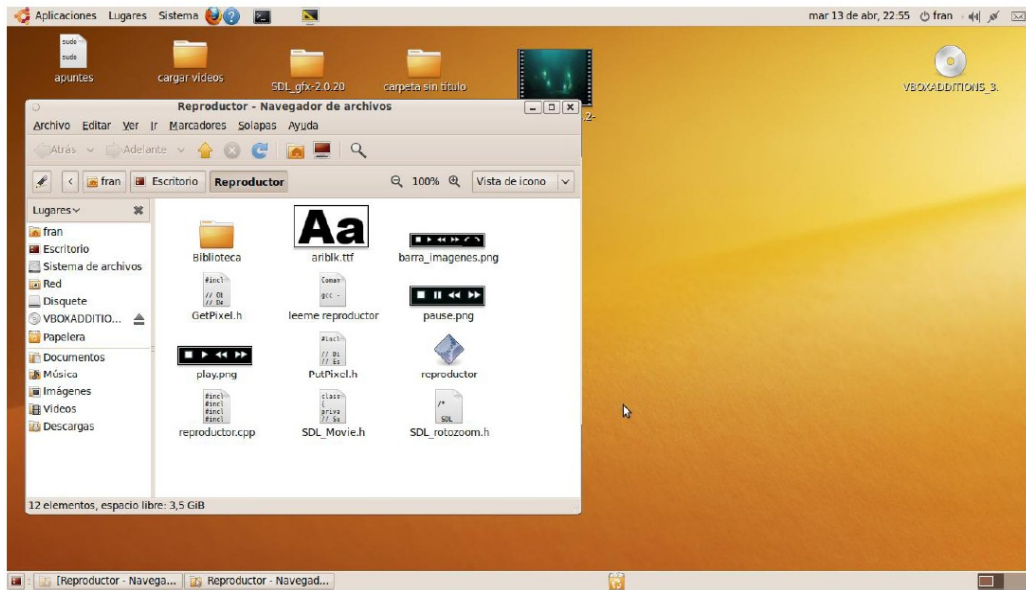


Figura 6: Ejemplo de contenidos tras la instalación.

música o imágenes, es necesario introducirlos previamente en el directorio “Biblioteca”. Este ha de estar dentro del directorio en que se ha copiado el reproductor, que al ejecutarse mostrará el menú inicial que se puede ver en la fig. 7.

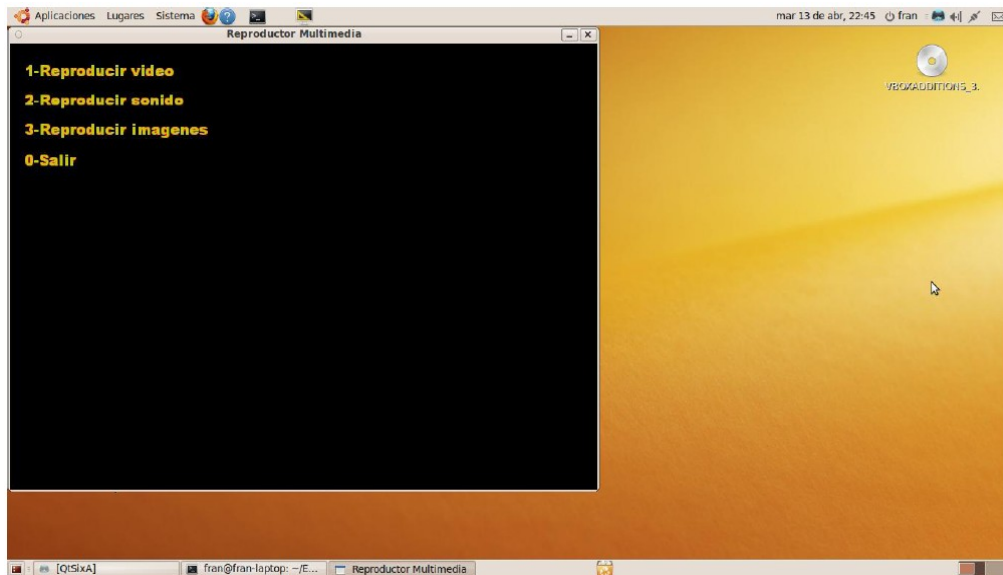


Figura 7: Menú principal de la aplicación.

Desde aquí entramos a los diferentes menús mediante los números del teclado o con los botones triángulo, cuadrado y equis del mando SixAxis para entrar en modo vídeo, sonido e imagen respectivamente. En cualquiera de los tres modos podemos navegar con el ratón si se ejecuta sobre un computador, pero en caso de hacerlo con el mando de la PS3, estos son las funciones asignadas a cada botón en cada caso: equis para reproducir y pausar, los pulsadores frontales (R1 y L1) para cambiar de fichero y círculo para salir.



6 Conclusiones y trabajos futuros

Bueno, no todo acaba aquí. Queda mucho por añadir a este ejemplo básico para que sea como las utilidades que acompañan a uno de los productos que están en el mercado dentro de un televisor o acompañando a un “disco multimedia”, pero hemos abierto el camino. De momento el lector podrá analizar las sugerencias de cómo se manipulan los diferentes medios utilizando la biblioteca de funciones SDL.

Animamos al lector a empezar por añadir un sistema de listas de reproducción que el usuario pueda personalizar y otras sugeridas como “lo más visto”, “lo menos escuchado”, “aleatorio”, agrupado por metadatos o asociando un sistema de puntuación que el usuario valora en algún momento. Es el momento de que te pongas manos a la obra e intentes elaborar uno. Ya verás que enriquecedor te resulta. ¡¡ÁNIMO!!

No quiero acabar sin antes agradecer a F. J. Alonso y F. Gramuntell su colaboración en la elaboración del material gráfico que acompaña a este artículo como resultado de su trabajo en la asignatura de Integración de Medios Digitales.

7 Bibliografía

[1] *Wikipedia, The Free Encyclopedia* (Última consulta en abril 2012). PlayStation 3. Disponible en <http://en.wikipedia.org/w/index.php?title=PlayStation_3&oldid=489304716>.

[2] *Wikipedia, The Free Encyclopedia* (Última consulta en abril 2012). SixAxis. Disponible en <<http://en.wikipedia.org/w/index.php?title=Sixaxis&oldid=488127819>>.

[3] *Simple Direct Media Layer*. (Última consulta en abril 2012). Disponible en <<http://www.libsdl.org/>>.

[4] Antonio García Alba (2008). Tutorial de libSDL para la programación de videojuegos.

[5] Fco. Javier Pérez Pacheco. SDL: Primeros pasos. Programación de Vídeo Juegos Disponible en <<http://www.javielinux.tk>>

[6] Fco. Javier Pérez Pacheco. SDL: Segundos pasos. Programación de Vídeo Juegos Disponible en <<http://www.javielinux.tk>>

[7] Javier Martínez Baena. Introducción a SDL (Simple Directmedia Layer)

[8] Alberto García Serrano. Programación de Videojuegos con SDL