// Ejemplo 4  
//  
// Listado: main.cpp  
// Programa de pruebas. Añadiendo una BSO  
  
#include <iostream>  
#include <SDL/SDL.h>  
#include <SDL/SDL\_mixer.h>  
  
using namespace std;  
  
int main()  
{  
 // Iniciamos el subsistema de video  
  
 if(SDL\_Init(SDL\_INIT\_VIDEO | SDL\_INIT\_AUDIO) < 0) {  
   
 cerr << "No se pudo iniciar SDL: " << SDL\_GetError() << endl;  
 exit(1);  
 }  
  
 atexit(SDL\_Quit);  
  
 // Comprobamos que sea compatible el modo de video  
   
 if(SDL\_VideoModeOK(640, 480, 24, SDL\_HWSURFACE|SDL\_DOUBLEBUF) == 0) {  
   
 cerr << "Modo no soportado: " << SDL\_GetError() << endl;  
 exit(1);  
   
 }  
  
 // Antes de establecer el modo de video  
 // Establecemos el nombre de la ventana  
  
 SDL\_WM\_SetCaption("Prueba. SDL\_mixer", NULL);  
  
 // Establecemos el modo  
  
 SDL\_Surface \*pantalla;  
   
 pantalla = SDL\_SetVideoMode(640, 480, 24, SDL\_HWSURFACE|SDL\_DOUBLEBUF);  
   
 if(pantalla == NULL) {  
   
 cerr << "No se pudo establecer el modo de video: "  
 << SDL\_GetError();  
   
 exit(1);  
 }  
  
 // Inicializamos la librería SDL\_Mixer  
  
 if(Mix\_OpenAudio(44100, MIX\_DEFAULT\_FORMAT,\  
 MIX\_DEFAULT\_CHANNELS, 4096) < 0) {  
   
 cerr << "Subsistema de Audio no disponible" << endl;  
 exit(1);  
 }  
  
 // Al salir cierra el subsistema de audio  
  
 atexit(Mix\_CloseAudio);  
  
 // Cargamos un fichero para la BSO  
  
 Mix\_Music \*bso;  
  
 bso = Mix\_LoadMUS("./Sonidos/bso.mp3");  
  
 if(bso == NULL) {  
  
 cerr << "No se puede cargar el fichero bso.mp3" << endl;  
 exit(1);  
 }  
  
 // Variables auxiliares  
  
 SDL\_Event evento;  
 SDL\_EnableKeyRepeat(SDL\_DEFAULT\_REPEAT\_DELAY,\  
 SDL\_DEFAULT\_REPEAT\_INTERVAL);  
  
 int volumen = 100;  
  
 cout << "Pulse ESC para salir" << endl;  
 cout << "Pulse Q para subir el volumen" << endl;  
 cout << "Pulse A para bajar el volumen" << endl;  
 cout << "Pulse I para reproduccir la música" << endl;  
 cout << "Pulse P para pausar la reproducción" << endl;  
 cout << "Pulse R para reproduccir la música después del pause" << endl;  
 cout << "Pulse W para poner la música al inicio" << endl;  
 cout << "Pusel H para mostrar ayuda" << endl;  
  
 // Bucle infinito   
   
 for( ; ; ) {  
  
 // RECUERDA: -1 en SDL\_Mixer en las llamadas a función  
 // simboliza infinito  
  
 while(SDL\_PollEvent(&evento)) {  
  
 if(evento.type == SDL\_KEYDOWN) {  
  
 if(evento.key.keysym.sym == SDLK\_ESCAPE) {  
  
 // Cerramos el sistema de audio  
 // al terminar de trabajar con él  
   
 Mix\_FreeMusic(bso);  
   
 cout << "Gracias" << endl;  
   
  
 return 0;  
 }  
  
 // Manejo del Volumen  
  
 if(evento.key.keysym.sym == SDLK\_q) {  
  
 volumen += 2;  
  
 if(volumen < 128)  
 Mix\_VolumeMusic(volumen);  
 else  
 volumen = 128;  
  
 cout << "Volumen actual: " << volumen << endl;  
  
 }  
  
 if(evento.key.keysym.sym == SDLK\_a) {  
  
 volumen -= 2;  
  
 if(volumen > -1)   
 Mix\_VolumeMusic(volumen);  
 else  
 volumen = 0;  
  
 cout << "Volumen actual: " << volumen << endl;  
  
 }  
  
 // Control de la reproducción   
   
 if(evento.key.keysym.sym == SDLK\_i) {  
  
 Mix\_PlayMusic(bso, -1);  
  
 cout << "Música iniciada" << endl;  
   
 }  
  
 if(evento.key.keysym.sym == SDLK\_p) {  
   
 if(Mix\_PlayingMusic() == 1) {  
   
 Mix\_PauseMusic();  
   
 cout << "Música en pausa" << endl;  
  
 } else {  
  
 cout << "La música no está en reproducción" << endl;  
  
 }  
  
 }  
  
 if(evento.key.keysym.sym == SDLK\_r) {  
   
 if(Mix\_PausedMusic() == 1) {  
  
 Mix\_ResumeMusic();  
   
 cout << "Música en reproducción" << endl;  
 } else {  
   
 cout << "La música ya está en reproducción" << endl;  
  
 }  
 }  
  
 if(evento.key.keysym.sym == SDLK\_w) {  
   
 Mix\_RewindMusic();  
   
 cout << "Música al principio" << endl;  
  
 }  
  
 if(evento.key.keysym.sym == SDLK\_h) {  
  
 cout << "\n == AYUDA == " << endl;  
 cout << "Pulse ESC para salir" << endl;  
 cout << "Pulse Q para subir el volumen" << endl;  
 cout << "Pulse A para bajar el volumen" << endl;  
 cout << "Pulse I para reproduccir la música" << endl;  
 cout << "Pulse P para pausar la reproducción" << endl;  
 cout << "Pulse R para reproduccir la música después del pause" << endl;  
 cout << "Pulse W para poner la música al inicio" << endl;  
 cout << "Pusel H para mostrar ayuda" << endl;  
  
 }  
  
 }  
 }  
   
 }  
  
}