1. Document Revision

Version	Author	Date (YY-MM-DD)	Comment
1.0.0	Lukas Heise	12-02-29	First revision

2. Overview

2.1 About

FreeSLW stands for Free Sound Library Wrapper. It is written in C/C++ and the intention is that FreeSLW is to be used in computer games, providing easy to use functions for sound playback, positioning in a 3D environment and overall management of sound effects. The predecessor to the library was called FreeSL and was originally developed for the openFrag (originally an open source first person shooter game) project back in 2006. The original FreeSL library was very popular and was used in several games including some AAA games.

2.2 Features

- Management of audio buffers and sources
- Handling of 2D and 3D sound effects
- Static playback of Wav and Ogg files
- Stream playback of Ogg files
- Environmental audio support
- Static playback of sounds from Zip file packages
- Organizing sound effects into groups
- Managing sound effects based on priority (*under development*)

2.3 Contributors

Name	Contact	Role
Lukas Heise	http://www.lukasheise.com	Original code and design, project initiator

2.4 License Overview

FreeSLW is free software released under the GPL 3 license. For more information please go to:
http://www.gnu.org/licenses/gpl-3.0.html

3. User Guide

3.1 Interface

Function Name	Return Value	Input Value	Description
GetSubsystem	The audio subsystem used for rendering sound	None	Returns the audio subsystem used for sound rendering
Update	None	None	This function should be called once each frame in your program loop.

GetListener	A pointer to the listener class	None	This function will retrieve the listener. Using the listener you can specify where in the environment the listener of the sound is positioned and in what type of environment it is positioned.
GetStatistics	General debug statistics	None	Returns general statistics for the audio engine.
GetSupportedStaticSoundFormats	A comma separated string	None	Returns the supported file formats for static sound playback
GetSupportedStreamSoundFormats	A comma separated string	None	Returns the supported file formats for stream sound playback
SetVolume	None	Volume [0,1]	Sets the master volume
GetVolume	The volume	None	Returns the master volume
SetMetersPerUnit	None	Meters per unit	Sets the scale of the world.
GetMetersPerUnit	Meters per unit	None	Returns the current scale of the world
SetGroup	None	The group number and if it should be active or not	Used to activate and deactivate groups
IsGroupActive	True if the group is active	The group number	Check if a specific group is active or not
StopAllSounds	None	None	Calls the stop function on all sounds that are playing
DuplicateSound	Sound object or null on failure	The sound object that will be duplicated	Duplicates a sound.
LoadSound	Sound object or null on failure	File path to the sound, group number and the priority level of the sound	Loads a static sound from file
LoadSoundFromZip	Sound object or null on failure	File path to the sound in the zip package, the zip file path, group number and the priority level of the sound	Loads a static sound from a zip file
LoadSoundFromData	Sound object or null on failure	Name of the sound, file type, file data, data length, group number and the priority level of the sound	Creates a static sound object from file data. The formats supported are the same that the LoadSound function supports
StreamSound	Sound object or null on failure	File path to the sound to be streamed, group number and the priority level of the sound	Loads a sound as a stream. Streamed sound objects will only load the portion of the sound from file that is currently playing. This can generate a lot of IO if many sounds are streamed but the advantage is that they take less memory and are faster to load
CreateSound	Sound object or null on failure	Name of the sound, sound data, data length, sound format, sound frequency, group number and the priority level of the sound	Creates a static sound object from the sound data provided
ReleaseSound	None	Sound object	Frees the sound object

3.2 Sound

Function Name	Return Value	Input Value	Description
GetName	The name of the sound	None	Returns the name of the sound (in most cases the file name)
GetFrequency	Sound Frequency	None	Returns the frequency of the sound
GetFormat	Sound format (stereo, mono)	None	Returns the format of the sound
IsStream	True if this is a streamed sound	None	Used to determine if this sound source is a streamed sound or a static sound source

SetGroup	None	The group number	Set the group number this sound is in
GetGroup	Group number	None	Return the current group this sound is in
Play	None	None	Play the sound
Resume	None	None	Resume playing of the sound if it is paused
Rewind	None	None	Rewind the sound to the beginning
Stop	None	None	Stop the sound from playing
Pause	None	None	Pause the sound
IsPlaying	True if the sound is playing or paused	None	Can be used to determine if the sound is playing. If the sound is paused then the function will return true
IsPaused	True if the sound is paused	None	Returns true if the sound is paused
SetLooping	None	True if the sound should loop indefinitely	Enable or disable the looping of this sound
IsLopping	True if the sound is set to loop	None	Returns true if the sound is set to loop
SetPosition	None	3D position vector	Set the position of the sound
GetPosition	None	Reference values that will hold the position	Get the position of the sound
SetVelocity	None	3D velocity vector	Set the velocity of the sound in 3D space
GetVelocity	None	Reference values that will hold the velocity	Returns the velocity of the sound in 3D space
SetPitch	None	The pitch value [0,any]	Set the pitch shift of the sound
GetPitch	Returns the pitch of the sound	None	Returns the pitch of the sound
SetGain	None	The gain value [0,1]	Set the gain of the sound
GetGain	Returns the gain of the sound	None	Returns the gain of the sound
SetRolloff	None	Roll off factor	Set the roll off factor. This parameter is used to control the range of the sound
GetRolloff	Returns the roll off factor	None	Returns the roll off factor
SetDistance	None	Reference and max distance	Set the relative and max distance for the sound
GetDistance	None	Reference values that will hold the min and max distance values	Returns the currently active relative and max distance of the sound
GetPlayDuration	The length of the sound in seconds	None	Returns the length of the sound in seconds. The actual playtime is: GetPlayDuration() / GetPitch()
GetPlayTime	The pan position in seconds	None	Returns the the pan position of the sound (how many seconds the current loop has played)
SetPlayTime	None	The pan position in seconds	Set the pan position in seconds of the sound
FadeIn	None	Duration in seconds until full fade out	Begins to fade the sound out [0,gain]
FadeOut	None	Duration in seconds until full fade in	Begins to fade the sound in [gain,0]

3.3 Listener

Function Name	Return Value	Input Value	Description
SetPosition	None	Position of the listener in 3D (x,y,z coordinate)	Set the position of the listener in the 3D space
GetPosition	None	Reference values that will hold the position	Returns the position of the listener in the 3D space

SetOrientation	None	Direction vector and the up vector	Set the view orientation (camera rotation) of the listener in the 3D world
GetOrientation	None	Reference values that will hold the direction and the up vector	Returns the orientation of the listener in the 3D space
SetVelocity	None	Velocity of the listener in 3D (x,y,z coordinates)	Set the velocity of the listener in the 3D space
GetVelocity	None	Reference values that will hold the velocity	Returns the velocity of the listener
SetDistanceModel	None	The distance model that you want to use	The distance model types match the ones found in the OpenAL library
GetDistanceModel	The currently active distance model	None	Returns the currently active distance model
SetDopplerParameters	None	Doppler factor and velocity	Set the doppler parameters
GetDopplerParameters	None	Reference values to where the doppler parameters will be set	Returns the doppler parameters
SetEnvironment	None	The audio environment the listener is in.	Set the environment of the world for the listener
GetEnvironment	The current active environment	None	Returns the currently set environment of the listener

3.4 Code Sample 1 (Sample1.cpp): Init/Load/Play/Unload

In this first example we initialize the interface, load and play a sound and when it is finished we release the sound and unload the interface.

3.5 Code Sample 2 (Sample2.cpp): 3D Sound, Listener and Environments

In this sample we set a few parameters for the listener. In the program loop we rotate the listener and we hear that the sound moves from one speaker to the other. We also set an environment witch add a bit of eko to the sound.

3.6 Code Sample 3 (Sample3.cpp): Groups

In this sample we load two sound effects and put them into two different groups. Groups can be used to separate for example in a game environment game sound effects from menu sound effects. When the user goes into menu mode we deactivate the sound effects in the game group and activate the sound effects in the menu group. This leads to that the sound effects in the game pause and will be resumed when we return to the game and activate the game group again.

4. Dependencies

Here is a list of libraries that FreeSLW uses

- OpenAL
- ALUT
- Ogg Vorbis
- zlib
- mmgr (Memory manager & tracking software by Paul Nettle)
- Creative EAX 2.0

5. Future Releases

Here is a list of features that are planned for future releases.

- Create project files for other IDE's and OS's
- Remove dependency to STL (uses STL list and vector)
- Interface::ReleaseAllSounds
- OpenAL: Move error messages to a separate file
- OpenAL: Add Obstruction and Occlusion support
- OpenAL: Add support for passwordprotected zip files
- OpenAL: Paused sounds should not use a source
- OpenAL::Globals::AllocSource => stop sounds that have a lower priority