
Subject: Externalizing/freeing music

Posted by [Taro_M](#) on Mon, 08 Aug 2016 08:36:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm trying to figure out how to change the music system, so all of the events (tension, battle, nothing etc.) have their own folders and the music system uses any .ogg or .wav files you place in those folders.

Right now I'm manually editing the "Music Control.cpp" and "Music Control.h" in order to add my own music, but I need to recompile the exe every time I do so.

Yeah, yeah you can just play the music using winamp, but I like when music fits the situation and game changes the music for me.

Currently in the "Music Control.cpp" there is a list "szMusicList" with contains the list of all of the music in the game. Later in the code for playing the music this list is called.

This is the battle music bit:

```
else if ( MusicMode == MUSIC_TACTICAL_BATTLE )
{
    sprintf( zFileName, "%s\\BATTLE_%d.ogg", szMusicList[15], uiNum);
    if ( !FileExists( zFileName ) )
        sprintf( zFileName, "%s\\BATTLE_%d.wav", szMusicList[15], uiNum );
}
```

From this the bit:

```
sprintf( zFileName, "%s\\BATTLE_%d.ogg", szMusicList[15], uiNum);
```

is what I'm interested in (the other one is for .wav files).

At lines 594, 708 and 769 there is also this bit:

```
bBattleModeSong = (INT8) (BATTLE_A_MUSIC + Random(2));
```

With is basically a randomization of the battle music if I understand this correctly. Notice the hardcoded "2" with means there are only 3 battle themes allowed.

So, I think that if we would have something like this:

```
sprintf( zFileName, "%s\\Battle_music\\*.ogg", uiNum);
```

and:

```
bBattleModeSong = (INT8) (BATTLE_A_MUSIC + Random(battle_music_count - 1));
```

battle_music_count = function counting of .ogg files in the new Battle_music folder.

I just want a simple to use system, similar to TES Oblivion and Fallout 3, where you could just place music files in the correct folders and game played them all as it should.

Subject: Re: Externalizing/freeing music
Posted by [Flugente](#) on Mon, 08 Aug 2016 10:08:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Assuming you'd number all your music files - so BATTLE_1.ogg, BATTLE_2.ogg, BATTLE_234.ogg etc. - you could use this very simple (but also very dumb) solution:

```
...
BOOLEAN wefoundmusicstoplay = FALSE;
while ( !wefoundmusicstoplay )
{
    UINT32 randomnumber = Random( 1000 );

    sprintf( zFileName, "%s\\BATTLE_%d.ogg", szMusicList[15], randomnumber );
    if ( !FileExists( zFileName ) )
        sprintf( zFileName, "%s\\BATTLE_%d.wav", szMusicList[15], randomnumber );

    if ( FileExists( zFileName ) )
    {
        wefoundmusicstoplay = TRUE;

        // do music stuff here..
    }
}
...
```

It's not smart and won't work on different names, but is a first step. Keep in mind that a few of the music numbers are likely stored in INT8, so you might need to change that.

Subject: Re: Externalizing/freeing music
Posted by [DepressivesBrot](#) on Mon, 08 Aug 2016 12:42:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

What you'll want to do for a fully drop'n'play system is to generate file lists from your folders in some way (SO has some ideas as usual) and then pick a random element to play.

Subject: Re: Externalizing/freeing music
Posted by [Taro_M](#) on Mon, 08 Aug 2016 16:49:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm no C coder sadly.

Why is the "szMusicList" called? Would it be necessary if there were separate folders for each music type?

As for the names: that why I'm suggesting the use of separate folders for all music types, this way you can tell the game to select any random music file from required folder.

Also, this may require an INI setting with would have something like new_music_system with could be set to true or false. On false music is played as it was, on new the folder system would be used.

Subject: Re: Externalizing/freeing music
Posted by [Flugente](#) on Mon, 08 Aug 2016 17:26:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Eh? szMusicList just contains a few strings - specific music files and the folder name itself.

Subject: Re: Externalizing/freeing music
Posted by [Taro_M](#) on Fri, 12 Aug 2016 12:31:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

It seems that after patch with allows ogg files I cannot add my music the way I did before. Now when I try to add music to the "MusicList" and increase the random number, with I shown in the first post. I get "too many integers" error during compilation I tried upping the szMusicList[15] to 52 (with was the correct number fow the ammount of music I tried to add), but I kept getting the same error, even when I tried changing the number up and down. I looked it up on the internet and changing the number should fix the error I was getting.

I also tried this:

```
sprintf( zFileName, "%s\\BATTLE\\BATTLE_%d.wav" );
```

but music was playing as in vanilla, with was weird as I was sure this would at least break the music system for combat.

Subject: Re: Externalizing/freeing music
Posted by [Julix](#) on Fri, 12 Aug 2016 13:50:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

DepressivesBrot wrote on Mon, 08 August 2016 14:42What you'll want to do for a fully drop'n'play system is to generate file lists from your folders in some way (SO has some ideas as usual) and then pick a random element to play.

If you randomized which file the game is looking for (like Taro suggested) why would you still need to have it look for the files in the folder? -- Although it would have the advantage of perhaps allowing for any kind of name, as long as it's in the appropriate folder... that would make it really easy to change... If you want extra music, throw those files into this folder. Done. Very neat.

Taro_M wrote on Mon, 08 August 2016 18:49

Also, this may require an INI setting with would have something like new_music_system with could be set to true or false. On false music is played as it was, on new the folder system would be used.

If it works well, not throwing additional music files in that folder would mean nothing changes, right? Why the support for the old system? Is it for use with Mods that changed the music? - Or would it be a precaution to keep ini setting in case it doesn't work well?

Subject: Re: Externalizing/freeing music

Posted by [DepressivesBrot](#) on Fri, 12 Aug 2016 15:37:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Julix wrote on Fri, 12 August 2016 15:50DepressivesBrot wrote on Mon, 08 August 2016 14:42What you'll want to do for a fully drop'n'play system is to generate file lists from your folders in some way (SO has some ideas as usual) and then pick a random element to play.

If you randomized which file the game is looking for (like Taro suggested) why would you still need to have it look for the files in the folder? -- Although it would have the advantage of perhaps allowing for any kind of name, as long as it's in the appropriate folder... that would make it really easy to change... If you want extra music, throw those files into this folder. Done. Very neat.If you want to force users to adhere to special naming conventions, then you can go with the generated filenames. If you just want to drop any random wav/ogg file from your library into say "BattleMusic", then you'll have to have the game check what files are in there and pick from those.

Subject: Re: Externalizing/freeing music

Posted by [Taro_M](#) on Fri, 12 Aug 2016 16:19:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

All my music was named battle_1.ogg, battle_2.ogg and so on (the music had the right format FYI).

As I said, when I tried to add my music files to the list I got an error during compilation.

Subject: Re: Externalizing/freeing music
Posted by [Flugente](#) on Fri, 12 Aug 2016 16:50:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Taro_M wrote on Fri, 12 August 2016 12:31 It seems that after patch with allows ogg files I cannot add my music the way I did before.

What patch? What I posted was a code bit, in no way finished code.

Taro_M wrote on Fri, 12 August 2016 12:31

I also tried this:

```
sprintf( zFileName, "%s\\BATTLE\\BATTLE_%d.wav" );
```

but music was playing as in vanilla, with was weird as I was sure this would at least break the music system for combat.

That line needs more arguments. Of course it won't work this way.

http://www.tutorialspoint.com/c_standard_library/c_function_sprintf.htm

Subject: Re: Externalizing/freeing music
Posted by [Taro_M](#) on Sat, 13 Aug 2016 08:06:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

At some point the musiccontrol.hpp has been changed.

I had older version saved (it had my changes), now when I compared it with the newest version it is very much different.

The old version: <http://pastebin.com/Wy4C2YLN>

The old version did not have lines calling for the list with is why my cludge code was working then.

Subject: Re: Externalizing/freeing music
Posted by [silversurfer](#) on Sat, 13 Aug 2016 08:31:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Someone tried to change the music system long time ago but didn't finish it as far as I remember. If you used some old code first and now downloaded the current build it is expected to have changed a lot.

Subject: Re: Externalizing/freeing music
Posted by [Taro_M](#) on Sat, 13 Aug 2016 11:58:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

I understand, I just wanted to say that the changes caused the music system to behave differently.

Could you guys help me break this down for me?

```
sprintf( zFileName, "%s\\BATTLE\\BATTLE_%d.wav", szMusicList[15], uiNum );
```

sprintf is the command that is used to get the name of the music file to be played

zFileName is the formatted file name of the music that will be played, it is formatted by the string

"%s\\BATTLE\\BATTLE_%d.wav" is the formatting string, the % symbols are places where specifiers/arguments are inserted

szMusicList[15] is an argument referencing the music list and what appears to be total number of entries in the list

uiNum not sure, the number of the song?

So what should be printed if, let's say, game is supposed to play "Tensor B.wav"? Maybe I could wrap my head around this, if I knew what is the end result.
