



SDL:Tutorials:Complete 2D Engine Sound Core

From GPWiki

 **The Game Programming Wiki has moved!** 

The wiki is now hosted by GameDev.NET at wiki.gamedev.net. All gpwiki.org content has been moved to the new server.

However, the GPWiki forums are still active! Come say hello.

Contents

- 1 Sound Core Overview
 - 1.1 What Should the Sound Core do?
 - 1.2 How the Sound Core is Laid out
- 2 Explanation of the Subsystems
 - 2.1 cSound
 - 2.2 cSoundManager
- 3 Conclusion
 - 3.1 Final Notes
 - 3.2 About the Source

Sound Core Overview

What Should the Sound Core do?

As always think of what you want it to do. In my case I want to have streaming music and sound effects. Since this is a 2D engine it wouldn't make much sense to have position based

sound so I will be leaving that out.

How the Sound Core is Laid out

Following on the same strategy as the graphics core I want to have a resource class and a manager class to handle all the resources.

- cSound - A single sound resource
- cSoundManager - Manages all sound files

Explanation of the Subsystems

cSound

cSound is a class to represent a sound resource, it should handle loading, playing and all of the functionality we wish to have for a sound file.

```
class cSound
{
    public:
        cSound();
        ~cSound();

        bool Load(std::string filename);
        void SetRepeat(int count);

        void Play();
        void Pause();
        void Stop();

    private:
        Aluint m_Buffer;
}
```

For the most part the sound resource is straight forward. 'Load' will open a file and put it into the sound buffer, false will be returned on a failure to load the file. 'SetRepeat' is handy for music and some effect files that you wish to be repeated, I have defined -1 as an infinite loop, 0 as not repeat and anything else as that number of times to repeat. 'Play', 'Stop' and 'Pause' do what you think they would. 'Aluint' is an openAL buffer type in case you are wondering.

cSoundManager

The sound manager handles all sound resources, to keep consistency this class will look very similar to the animation manager.

```
class cSoundManager
{
    public:
        SoundManager();
        ~SoundManager();

        uint CreateSound( std::string Filename);

        void Remove(uint Sound);
        void RemoveAll();

        void Pause(uint Sound);
        void PauseAll();

        void Stop(uint Sound);
        void StopAll();

        void Play(uint Sound);
        void PlayAll();
        void SetRepetitions(uint Sound, int count);

    private:
        std::map< uint, cSound > m_Sounds;
};
```

If you can't remember how the animation manager was laid out go take a look at it and compare it to the sound manager. Isn't it great to have a unified approach to both sound and graphics? Anyways everything here should be self explanatory.

Conclusion

Final Notes

The sound core can be a very complex beast but I decided to make it very simple. Maybe next time I make an engine I will include 3D sound and then the real fun can begin.

About the Source

As will all the other tutorials you can get the whole source code [here](#).

This tutorial was written by [Seoushi](#). Do you like this tutorial? Have any questions or comments? Let me know or you can ask the forums.

Retrieved from "http://gpwiki.org/index.php/SDL:Tutorials:Complete_2D_Engine_Sound_Core"

Categories: [C and SDL](#) | [All C articles](#) | [All SDL articles](#)

- Content is available under [GNU Free Documentation License 1.2](#).
- [Link to Us!](#)