



Task 16 - Spike Summary Report



Spike: Task_16

Title: Sound Board

Author: Thomas Horsley, 103071494

Goals & Deliverables

Aim: Use SDL2 has a framework to load, play and pause sound effects and music.

Deliverables:

- Functional Code
- Spike Summary Report

Technology, Tools and Resources

Tech and Tools



The project was scripted in C++ 17 using Visual Studio Community 2022.

UML's and charts are made with www.Lucidchart.com

Source control is handled using Git.

Resources

- SDL2 Mixer Documentation:
https://wiki.libsdl.org/SDL2_mixer/FrontP
- Sound With C++:
<https://www.youtube.com/watch?v=q06uSmkiqec>
- Music With C++:
<https://www.youtube.com/watch?v=DbZUftR6-yU&t=847s>

Tasks Undertaken

Planning & Implementation

SoundBoard		
- _is_playing	:	bool
- _is_paused	:	bool
- _sound_effects	:	std::vector<Mix_Chunk*>
- _music_tracks	:	std::vector<Mix_Music*>
<hr/>		
+ SoundBoard()	:	SoundBoard
+ ~SoundBoard()	:	SoundBoard
<hr/>		
+ loadSound(const char file_path[])	:	void
+ loadMusic(const char file_path[])	:	void
+ playSound(const int sound_pos) const	:	void
+ playMusic(const int track_pos)	:	void
+ playPause()	:	void

The soundboard will be made using the SDL2 developer libraries and the SDL2_Mixer extension libraries for SDL2.

The event handling functionality is handled in the main().

Git Commit History



Commits

History for [COS30031-2023-103071494](#) / 16 - Spike - Sound Board

Commits on Sep 27, 2023

Started and Finished Task 16 - Spike 'Sound Board' ...



unknown committed 3 days ago

df5c803  

Commits on Sep 18, 2023

Started Task 17 - Spike 'Sprites and Graphics' ...



unknown committed 2 weeks ago

2d858b8  

Commits on Sep 17, 2023


Completed playing audio on keypress ...



unknown committed 2 weeks ago

04b4165  

Commits on Jul 30, 2023

Initial Commit ...

 KingSchlock committed on Jul 31

0d3f501  

End of commit history for this file

Newer

Older

Code

```
4
5 SoundBoard::SoundBoard() {
6     if (SDL_Init(SDL_INIT_AUDIO) == 0) {
7         int audio_rate = 22050;
8         Uint16 audio_format = AUDIO_S16SYS;
9         int audio_channels = 2;
10        int audio_buffers = 4096;
11
12        // Der out on bad return values
13        if (Mix_OpenAudio(audio_rate, audio_format,
14            audio_channels, audio_buffers) != 0) {
15            SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad audio mixer init: %s",
16                Mix_GetError());
17            exit(-1);
18        }
19    }
20
21
22 SoundBoard::~SoundBoard() {
23     SDL_Quit();
24 }
25
26 void SoundBoard::loadSound(const char file_path[]) {
27     Mix_Chunk* new_chunk = Mix_LoadWAV(file_path);
28
29     if (new_chunk != nullptr) {
30         _sound_effects.push_back(new_chunk);
31         std::cout << "Sound " << _sound_effects.size() << ": " << file_path
32             << " load success...\n";
33     }
34     else {
35         SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad sound chunk load: %s",
36             Mix_GetError());
37     }
38 }
39
40 void SoundBoard::loadMusic(const char file_path[]) {
41     Mix_Music* new_track = Mix_LoadMUS(file_path);
42     if (new_track != nullptr) {
43         _music_tracks.push_back(new_track);
44         std::cout << "Track " << _music_tracks.size() << ": " << file_path
45             << " load success...\n";
46     }
47     else {
48         SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad track load: %s",
49             Mix_GetError());
50     }
51 }
```

Soundboard construction and sound loading

```
53 void SoundBoard::playSound(const int sound_pos) const {
54     // Check input is legal
55     if (sound_pos < _sound_effects.size()) {
56         Mix_PlayChannel(-1, _sound_effects[sound_pos], 0);
57         std::cout << "Playing track: " << sound_pos+1 << "\n";
58     }
59 }
60
61 void SoundBoard::playMusic(const int track_pos) {
62     if (track_pos < _music_tracks.size()) {
63         Mix_PlayMusic(_music_tracks[track_pos], -1);
64         _is_playing = true;
65     }
66 }
67
68 void SoundBoard::playPause() {
69     if (_is_playing && !_is_paused) { // Is playing
70         Mix_PauseMusic();
71         _is_paused = true;
72     } else if (!_is_playing && _is_paused) { // Is Paused
73         Mix_ResumeMusic();
74         _is_paused = false;
75     }
76     else if (!_is_playing && !_is_paused) { // No track has been played yet
77         playMusic();
78     }
79 }
```

Soundboard play / pause functionality.

What was Learned?



After completing the spike, I'm able to play, pause, load and free music and sound chunks on event triggers.