```cpp
#include <iostream>
#include "../hdr/SoundBoard.h"

int main(int argc, char* argv[]) {
    SoundBoard sound_board;

    sound_board.loadSound("audio/BombHasBeenPlanted.wav");
    sound_board.loadSound("audio/RecordScratchSoundEffect.wav");
    sound_board.loadSound("audio/ThrowingFlashbangSoundEffectCSGO.wav");
    sound_board.loadMusic("audio/RatsBirthdayMixtape.mp3");

    sound_board.playMusic(0);

    int choice = 0;
    while (choice != -1) {
        choice = 0;
        std::cout << "\nEnter a sound to play using num input\n";
        std::cout << "-1 to exit...\n";
        std::cout << ">> ";
        std::cin >> choice;

        if (choice == 9) {
            sound_board.playPause();
        }

        sound_board.playSound(choice - 1);
        std::cin.clear();
    }

    return 0;
}

```

```cpp
#include <vector>
#include "SDL_mixer.h"
#pragma once

class SoundBoard {
public:
    SoundBoard();
    ~SoundBoard();

    void 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();

private:
    // These pertain to music not sounds
    bool _is_playing = false;
    bool _is_paused = false;

    std::vector<Mix_Chunk*> _sound_effects;
    std::vector<Mix_Music*> _music_tracks;
};
```

```cpp
#include <iostream>
#include "../hdr/SoundBoard.h"
#include "SDL.h"

SoundBoard::SoundBoard() {
    if (SDL_Init(SDL_INIT_AUDIO) == 0) {
        int audio_rate = 22050;
        Uint16 audio_format = AUDIO_S16SYS;
        int audio_channels = 2;
        int audio_buffers = 4096;

        // Err out on bad return value
        if (Mix_OpenAudio(audio_rate, audio_format,
            audio_channels, audio_buffers) != 0) {
            SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad audio mixer init: %s",
                Mix_GetError());
            exit(-1);
        }
    }
}

SoundBoard::~SoundBoard() {
    SDL_Quit();
}

void SoundBoard::loadSound(const char file_path[]) {
    Mix_Chunk* new_chunk = Mix_LoadWAV(file_path);

    if (new_chunk != nullptr) {
        _sound_effects.push_back(new_chunk);
        std::cout << "Sound " << _sound_effects.size() << ": " << file_path
            << " load success...\n";
    }
    else {
        SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad sound chunk load: %s",
            Mix_GetError());
    }
}

void SoundBoard::loadMusic(const char file_path[]) {
    Mix_Music* new_track = Mix_LoadMUS(file_path);
    if (new_track != nullptr) {
        _music_tracks.push_back(new_track);
        std::cout << "Track " << _music_tracks.size() << ": " << file_path
            << " load success...\n";
    }
    else {
        SDL_LogError(SDL_LOG_CATEGORY_APPLICATION, "Bad track load: %s",
            Mix_GetError());
    }
}

void SoundBoard::playSound(const int sound_pos) const {
    // Check input is legal
    if (sound_pos < _sound_effects.size()) {
        Mix_PlayChannel(-1, _sound_effects[sound_pos], 0);
        std::cout << "Playing track: " << sound_pos+1 << "\n";
    }
}

void SoundBoard::playMusic(const int track_pos) {
    if (track_pos < _music_tracks.size()) {
        Mix_PlayMusic(_music_tracks[track_pos], -1);
        _is_playing = true;
    }
}

void SoundBoard::playPause() {
    if (_is_playing && !_is_paused) { // Is playing
        Mix_PauseMusic();
        _is_paused = true;
    } else if (_is_playing&& _is_paused) { // Is Paused
        Mix_ResumeMusic();
        _is_paused = false;
    }
    else if (!_is_playing && !_is_paused) { // No track has been played yet
        playMusic(0);
    }
}
```