

**Spike:** Task 16.P

**Title:** Sound Board

**Author:** Mitchell Wright, 100595153

**Goals / deliverables:**

To demonstrate an understanding of frameworks and sound/music functionality within SDL2.

Items created during task:

- Code, see: \16 - Spike – Sound Board\SDL2

**Technologies, Tools, and Resources used:**

- Visual Studio 2022
- SourceTree
- GitHub
- SDL2 Development Library
- SDL2\_mixer Development Library
- Lecture 3.2 – Data Structures

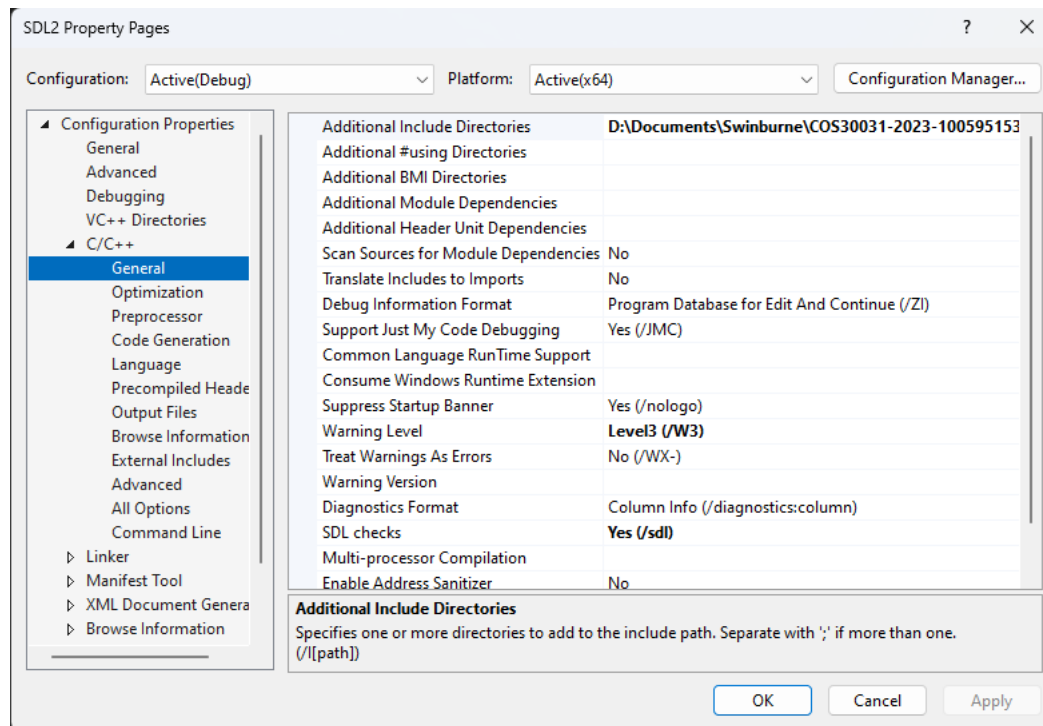
**Tasks undertaken:**

- Implementing SDL2\_mixer
- Playing Sounds
- Playing and Pausing Background Music
- Commit to Git

## What we found out:

### 1. Implementing SDL2\_mixer:

Implementing the mixer library isn't too dissimilar to implementing SDL2. All that was needed to get it running was to add in the include directory and dependencies.



Once implemented, an additional Init was needed to allow the audio systems to function.

```
//Init, init check, error message.
if (SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO) != 0)
{
    SDL_Log("SDL init failed: %s", SDL_GetError());
    return 1;
}
```

Additionally, the following line was needed in order to output any audio.

```
Mix_OpenAudio(44100, MIX_DEFAULT_FORMAT, 2, 2048);
```

The parameters, in their respective order, are the bitrate of the audio, audio format, number of channels and buffer size.

### 2. Playing Sounds:

Playing sounds was incredibly easy. Firstly, in order to have a key press be registered, an `SDL_Event` object must be created, then the type needs to be checked. In this case, we're wanting to check for `SDL_KEYDOWN` events.

```
if (event.type == SDL_KEYDOWN)
{
    switch (event.key.keysym.sym)
    {
        case SDLK_1:
            playSound("potion_seller.wav");
            break;
        case SDLK_2:
            playSound("im_going_into_battle.wav");
            break;
        case SDLK_3:
            playSound("i_want_your_potions.wav");
            SDL_Log("You can't handle my potions. My potions are too strong for you, traveller.");
            break;
    }
}
```

From there, I've used a switch block in order to find which key was pressed. Once a 1, 2 or 3 are pressed, the `playSound` function is called, with a filename as a parameter. This then plays a sound. Real easy stuff here.

### 3. Playing and Pausing Background Music:

Unlike sounds previously, a music object needs to be created, though still with a filename as a parameter.

```
Mix_Music* bgm = Mix_LoadMUS("oblivion.wav");
```

Once set, the music can be played, paused and resumed as below.

```
case SDLK_0:
    if (Mix_PlayingMusic() == 0)
        Mix_PlayMusic(bgm, -1);
    else if (Mix_PausedMusic() == 1)
        Mix_ResumeMusic();
    else
        Mix_PauseMusic();
```

Again, just really simple stuff here. I did try for a while to use the inbuilt SDL2 functions to do this, but uh, it was complicated, and tutorials all referenced `SDL2_mixer` instead.

4. Commit to Git:

