# The Audio System

What is it’s purpose?

To play audio.

In the game sound will be utilized through sfx, such as actions that the player performs and actions that the enemies perform. Sounds will also be played during certain menu effects as well.

Music will be played based on the level as well as state of the game.

How to organize sounds.

When the player uses an action such as the acceleration action, the audio system will be notified and will play the proper sound in relation to the player. When the player is no longer using that action, the sfx will be stopped.

OK WHAT IS THE PROBLEM

We will need an individual buffer for each sound

A hashmap will hold a buffer per sound type and when a sound needs to be played, we’ll look up the buffer and create a sound instance play it and throw it on the stack

Maybe we should have a stack per sound type.

Ex. Menu sound stack, Actor sound stack, Object sound stack

The menus have their own sounds

Entities have their own sounds

Levels have their own music

Let’s distinguish music/audio streams from sound.

Sound covers sfx and short durations of audio.

Music covers audio of longer duration

When a level loads each level will play a background track.

Each track can range from music to background ambience. ~~And as the player goes along in the level the tracks will change based on various triggers (possibly)~~

Combat sequences or even triggers can cause a change in the background music. This can result in overlaying of sounds or simply switching the track gradually

Narrative sequences can cause an audio file to be streamed

## AudioSystem Class

Objects:

* soundQueue
* BackgroundMusic
* SoundBuffer(sound effects)per effect type
* NarrationMusic(narrative sounds)

Functions:

* update()
* setMasterVolume()
* setSoundEffectsVolume()
* setMusicVolume()
* setDefault()