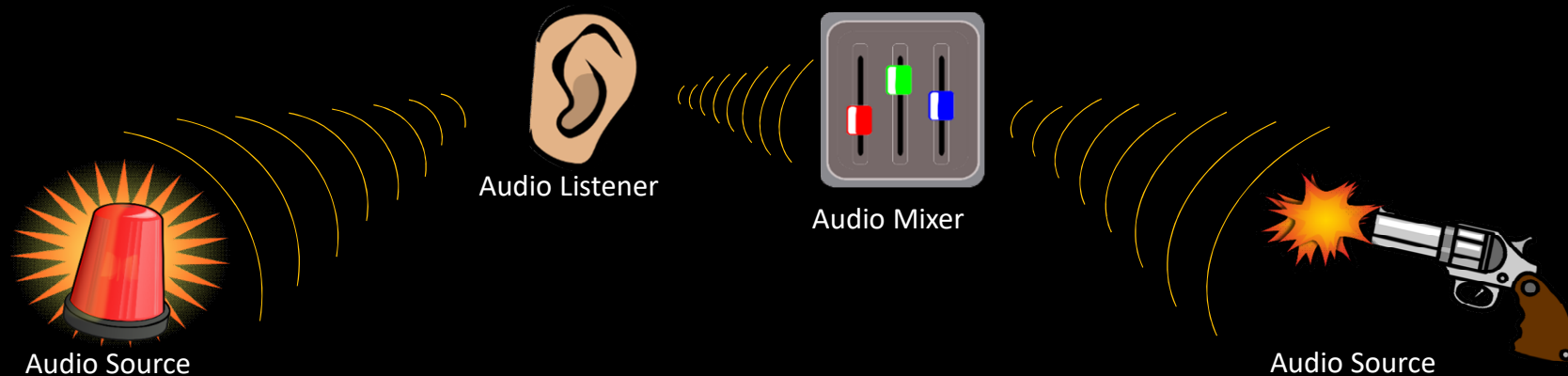


# Game Music Concepts

## Game Music Concepts

# Recap – Audio Concepts

- In a previous section of the course we looked in detail at ‘Sound Effects And Audio Concepts’, including Audio Listeners, Audio Sources, and Audio Mixers used in Unity.
- Feel free to watch that video again as a recap, since we’ll be using many of the same concepts when we implement music for the game.



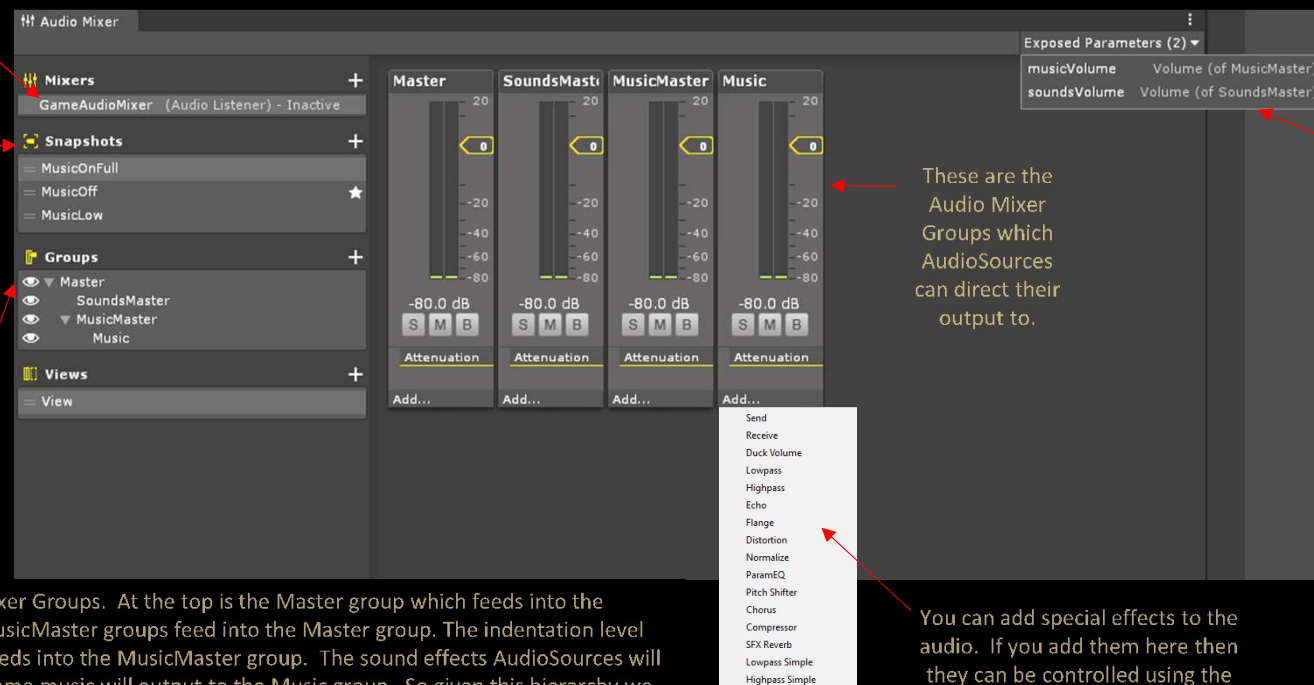
# Game Music Concepts

## Recap – Audio Mixer

- In that previous lecture we looked in detail at the Audio Mixer asset that's supplied in the course assets, and how it's used to take the sounds from audio sources, and then pass that audio through to the audio listener in the scene.

This is the asset name of the Audio Mixer

These are the Snapshots – they capture a specific configuration of an Audio Mixer Group (the Music group in this case)



These are exposed parameters – so here we can see that the volume of both the MusicMaster and SoundsMaster Audio Mixer Groups are exposed to code (using the parameter names musicVolume and soundsVolume). You actually expose the parameters in the Project Assets through the inspector.

In the future, we'll develop a menu to allow us to control the game music volume level by changing the MusicMaster volume level, and the game sound effects volume level by changing the SoundsMaster volume level.

This is the 'hierarchy' of the Audio Mixer Groups. At the top is the Master group which feeds into the AudioListener. The SoundsMaster and MusicMaster groups feed into the Master group. The indentation level shows the relationship. The Music group feeds into the MusicMaster group. The sound effects AudioSource will output to the SoundsMaster group. The game music will output to the Music group. So given this hierarchy we can fade the music in and out using the Music group, without affecting the overall MusicMaster group volume. We'll actually fade the game music in and out using the Snapshots we've defined, and transitioning between them in the code.

You can add special effects to the audio. If you add them here then they can be controlled using the 'snapshots'

# Game Music Concepts

## Audio Mixer

- We'll be using the same Audio Mixer asset that we used for sound effects for our game music.
- The output from the game music Audio Source component will be directed through the 'Music' group in the audio mixer, so that we can control the music fading in and out using snapshots.



## Game Music Concepts

# Game Music Classes - MusicTrackSO

### MusicTrackSO

```
Header MUSIC TRACK DETAILS  
Tooltip  
public string musicName;  
Tooltip  
public AudioClip musicClip;  
Tooltip  
[Range(0,1)]  
public float musicVolume = 1f;
```

- A scriptable object asset will be created for each music track that we want to play.
- The MusicTrackSO will contain details of the Audio Clip to be played and the music volume between 0 and 1 (this can be used to make loud music recordings quieter).

## Game Music Concepts

### Game Music Classes – Music Manager

#### MusicManager

```
public void PlayMusic(MusicTrackSO musicTrack,  
    float fadeOutTime = Settings.musicFadeOutTime,  
    float fadeInTime = Settings.musicFadeInTime)
```

- The MusicManager class is a Singleton component that will be added to the MusicManager gameobject in the scene. This singleton can be accessed through a static instance variable.
- It has a publicly facing method to play a music track. This method will fade out any music that's already playing, and then fade in the new music track specified. The fading in and out of music will be done through the 'snapshots' defined in the Audio Mixer asset.

# Game Music Concepts

## Game Object And Component Structure

