



# Programming Composers and Composing Programmers

Victoria Dorn – Sony Interactive Entertainment





# About Me

- Berklee College of Music (2013) – Sound Design/Composition
- Oregon State University (2018) – Computer Science
- Audio Engineering Intern -> Audio Engineer -> Software Engineer
- Associate Software Engineer in Research and Development at PlayStation
- 3D Audio for PS4 (PlayStation VR, Platinum Wireless Headset)
  - Testing, general research, recording, and developer support





# Agenda

- Programming tips/tricks for the audio person
- Audio and sound tips/tricks for the programming person
- Creating a dialog and establishing vocabulary
- Raise the level of common understanding between sound people and programmers
- Q&A





# Media Files Used in This Presentation

- [Can be found here](#)
- <https://drive.google.com/drive/folders/1FdHR4e3R4p59t7ZxAU7pyMkCdaxPqbKI?usp=sharing>





# Programming Tips for the Audio Folks



"Binary Code" by [Cncplayer](#) is licensed under [CC BY-SA 3.0](#)





# Music/Audio

DAWs

Musical Motives

Instruments

=

# Programming

Programming Language(s)

=

Programming Logic

=

APIs or Libraries





# Where to Start??

- Learning the Language
- Pseudocode
- Scripting





# Learning the Language

- Programming Fundamentals

- Variables (a value with a name)

```
soundVolume = 10
```

- Loops (works just like looping a sound actually)

```
for (loopCount = 0; while loopCount < 10; increase loopCount by 1){  
    play audio file one time  
}
```

- If/else logic (if this is happening do this, else do something different)

```
if (the sky is blue){  
    play bird sounds  
}  
else{  
    play rain sounds  
}
```







# Learning the Language

- Programming Fundamentals
  - Data structures (Describe how data is organized)
  - Algorithms (Interesting things to do with our data)
    - Sort of like recipes (**cake** recipes)



```
typedef struct waveFile{
    /*RIFF-WAVE chunk*/
    unsigned char    chunkID[4];
    unsigned int     chunkSize;
    unsigned char    format[4];

    /*format subchunk*/
    unsigned char    subChunkID[4];
    unsigned int     subChunkSize;
    unsigned short   audioFormat;
    unsigned short   numChannels;
    unsigned int     samplingRate;
    unsigned int     byteRate;
    unsigned short   frameSize;
    unsigned short   sampleBit;

    /*data chunk*/
    unsigned char    dChunkID[4];
    unsigned int     dChunkSize;
    void*            dataArray;
}waveFile;
```





# Programming Logic Resources:

- [Tutorials Point](https://www.tutorialspoint.com/computer_programming/index.htm)
  - [https://www.tutorialspoint.com/computer\\_programming/index.htm](https://www.tutorialspoint.com/computer_programming/index.htm)
- [Crash Course](https://www.youtube.com/playlist?list=PL8dPuuaLjXtNIUrzyH5r6jN9ulIgZBpdo)
  - <https://www.youtube.com/playlist?list=PL8dPuuaLjXtNIUrzyH5r6jN9ulIgZBpdo>





# Pseudocode

- What is Pseudocode?
  - A simplified notation to describe a program's logic
  - Everyday words used to describe a solution to a programming problem

```
if we hit something  
    play a crash sound
```

```
if a weapon's velocity is > 10  
    play a loud crash sound  
else  
    play a quiet crash sound
```

```
if a weapon's velocity is > 10  
    if there are more than 15 enemies  
        play a loud crash sound  
    else  
        play a quiet crash sound
```





# Pseudocode

- What is Pseudocode?
  - A simplified notation to describe a program's logic
  - Everyday words used to describe a solution to a programming problem
- Start writing down your sound design and music playback ideas as pseudocode
  - Helps to find the parameters you need exposed from the game side
  - Gives an idea of how you want to branch your sounds/music

```
if a weapon's velocity is > 10
    if there are more than 15 enemies
        play a loud crash sound
    else
        play a quiet crash sound
```





# Pseudocode Resources:

- [Khan Academy](#)
  - <https://www.khanacademy.org/computing/computer-programming/programming/good-practices/p/planning-with-pseudo-code>
- [Wikipedia](#)
  - <https://en.wikipedia.org/wiki/Pseudocode>

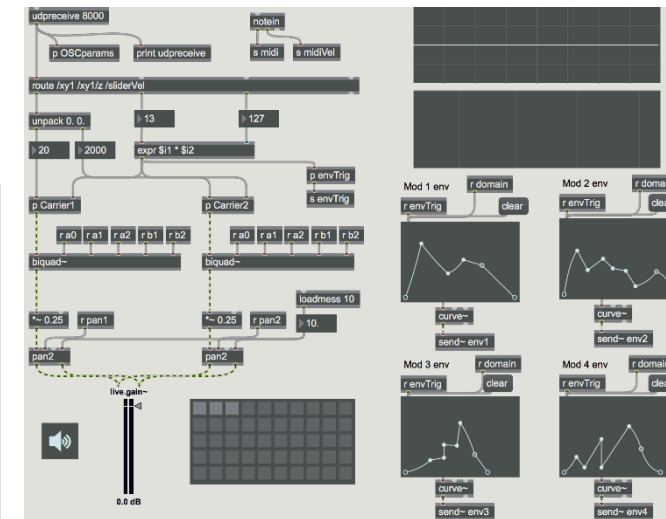
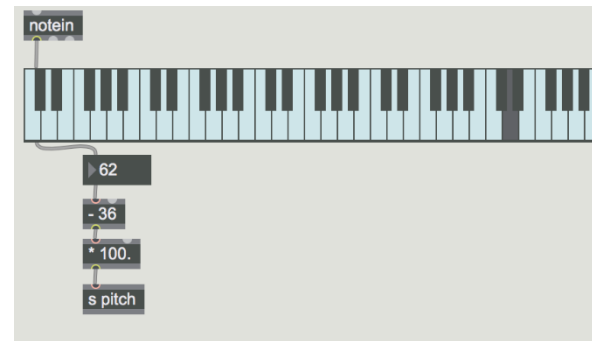




# Scripting!!

- Scripts are small sections of code used to pull together larger, more complex features
  - Cool results without too much technical know-how
- Common Languages used for Scripting:
  - Lua, Python, C# (Unity), MaxMSP (Visual Scripting)
- C# examples...

```
thePrint = "Hello"  
thePrint = thePrint + " World!"  
print thePrint
```



MaxMSP Patches by Brandon Sangston





# Play A Sound:

```
audioSource.Play();
```





# Play A Sound with a Pitch and Volume:

```
float vol = 0.5;
```

```
float pitch = 0.8;
```

```
audioSource.volume = vol;
```

```
audioSource.pitch = pitch;
```

```
audioSource.Play();
```







# Play A Sound with a Random Pitch and Volume:

```
float vol = Random.Range(.5f, .75f);
```

```
float pitch = Random.Range(.80f, 1f);
```

```
audioSource.volume = vol;
```

```
audioSource.pitch = pitch;
```

```
audioSource.Play();
```





# Play A Sound When a Button is Pressed:

```
if (Input.GetKey(button)) {  
    float vol = Random.Range(.5f, .75f);  
    float pitch = Random.Range(.80f, 1f);  
    audioSource.volume = vol;  
    audioSource.pitch = pitch;  
    audioSource.Play();  
}
```





# Production Script:

```
if (Input.GetKey(button)) {  
    if (audioSource != null) {  
        float vol = 0.5;  
        float pitch = 0.8;  
        audioSource.volume = vol;  
        audioSource.pitch = pitch;  
        audioSource.Play();  
    }  
}
```





# Scripting Resources:

- [Unity C# Tutorials](#)
  - <https://unity3d.com/learn/tutorials/s/scripting>
- [Learn Python the Hard Way](#)\*
- <https://learnpythonthehardway.org/>
- [LearnPython.org](#)
- [Tutorials Point](#) (Scripting)
  - [https://www.tutorialspoint.com/scripting\\_language\\_tutorials.htm](https://www.tutorialspoint.com/scripting_language_tutorials.htm) \*\*

\*Not actually that hard

\*\*Typo in link is supposed to be there.





# Programming

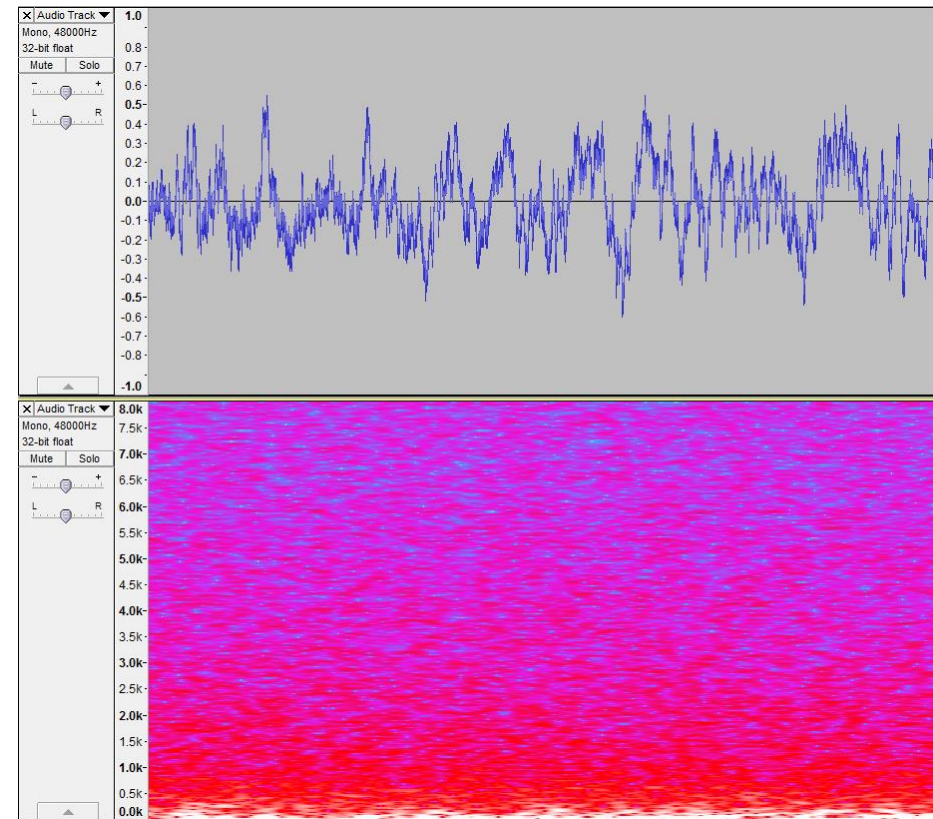
- Not that hard to get started
- Dive in and create

```
if you try programming
    if you really like it
        you will have tons of fun
    else
        you had fun trying
else
    you might have fear of missing out
```





# Audio Tips for the Programming Folks





# Where to Start??

- Learn the Language
- Listen to audio
- Learn audio middleware and the audio side of game engines





# Learning the Language

- A Mix
- Stems
- Distortion (Unwanted Distortion)
  - CPU peaking versus Amplitude peaking (example causes)
  - Crackling versus Popping (example effects)
- Effects
  - Manipulating the way audio sounds
  - Reverb
  - Low-Pass Filters
- Audio Quality Keywords
  - “Tinny”, “bass-y”, “not enough winds”, “this sounds blue...”

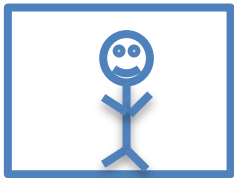




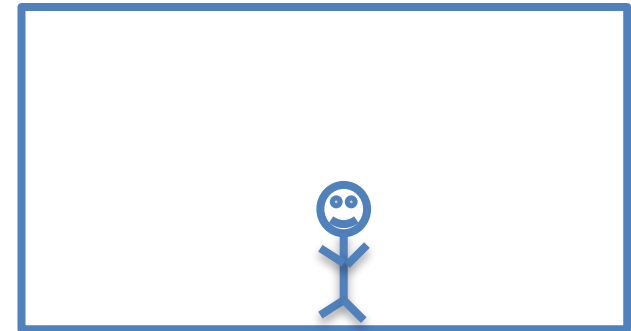


# Reverb Example

Small Room



Big Room





# Tinny Versus Not Tinny Example

Tinny



Not Tinny





# Learning the Language

- And many more...
- Learn from each other
- Good resource:
  - <https://www.soundonsound.com/sound-advice/glossary-technical-terms>





# Listen to Audio





# No, really...



Bang!

# Listen to Audio



Pow!





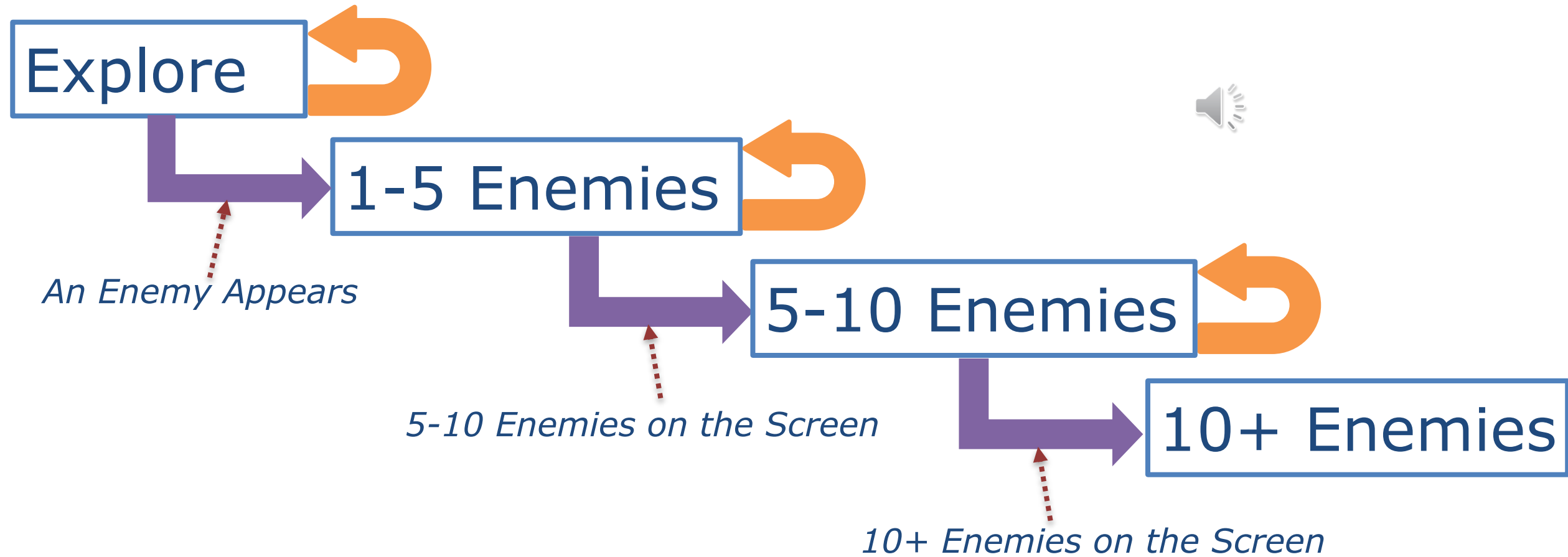
# Listen to Audio

- Play games, listen to audio transitions
  - How is the game you are playing creating interactive music/sound?
- Interactive music paradigms
  - Branching music
  - Layering based on state changes
- Interactive sound design
  - Game parameter “hooks”
  - Variation creation through sound “grains” and elements
  - Reverb



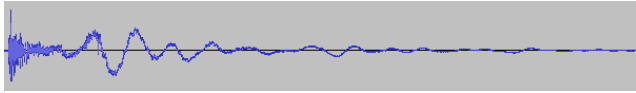


# Music Branching Example





# Grain/Variation Example



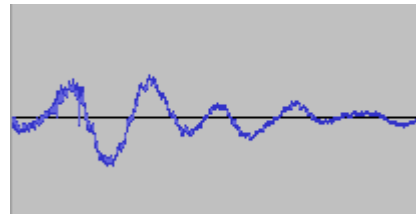
Normal footstep sounds

Footstep1
Footstep2
Footstep3
Footstep4



Heel	Toe
------	-----

Break apart heel and toe sounds



Granularized footstep sounds

Heel2	Toe1
Heel4	Toe2
Heel1	Toe3
Heel3	Toe4







# Listen to Audio

- Take what you have learned and try to analyze games
  - Play your favorite RPG
    - What are they doing with their sounds?
    - Are they doing anything atypical but cool?
    - Use these techniques





# Resources

- [Designing Sound](#) (Audio Implementation Greats)
  - <http://designingsound.org/tag/audio-implementation-greats/>
- [Audio Coding Perception Examples](#) (Audio Engineering Society)





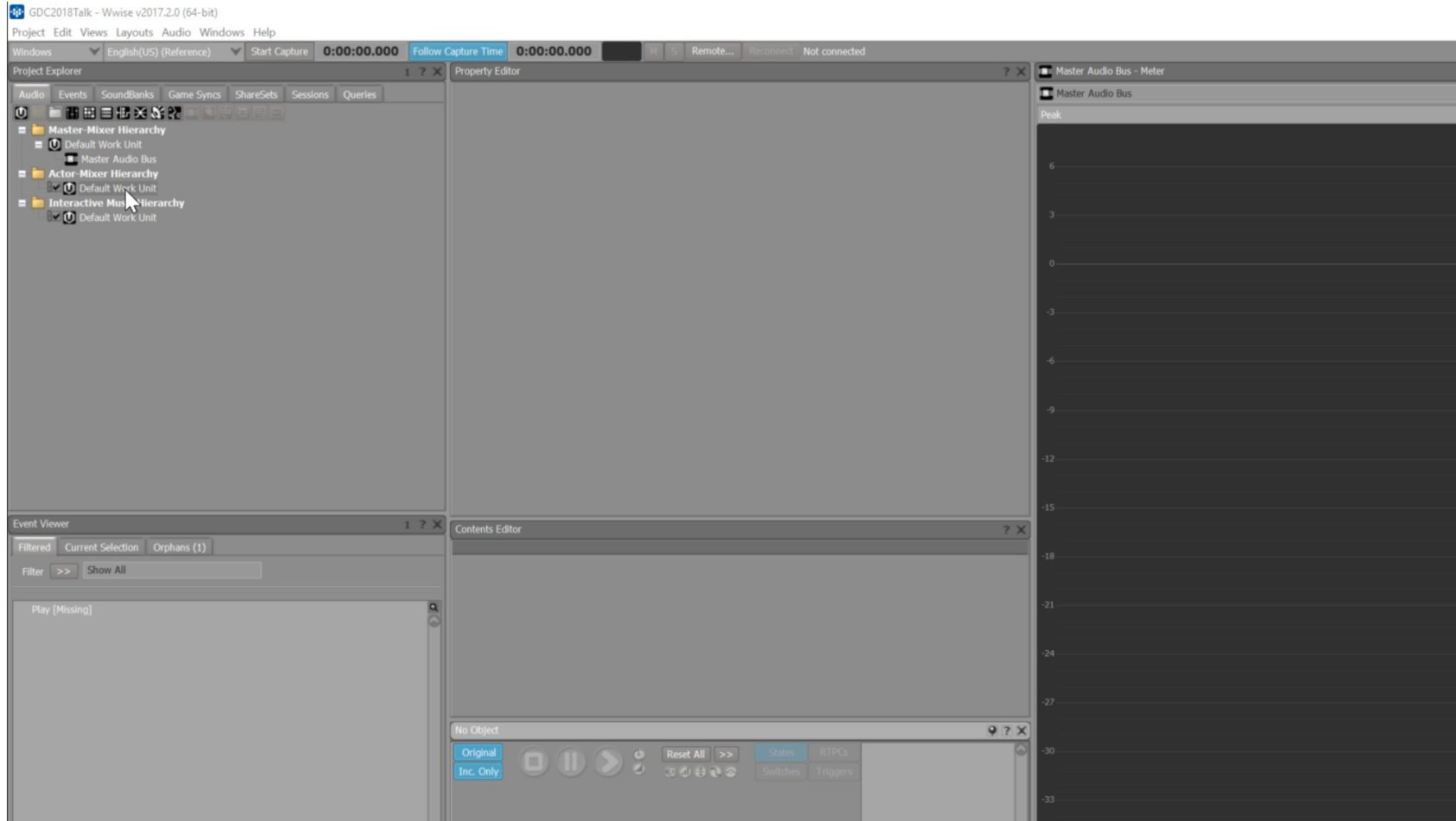
# Audio Middleware / Game Engine Audio

- Try out one of the many audio middleware engines
  - For example, Wwise or FMOD
  - Learn how to implement some basic audio behaviors using their interface
- Try figuring out how to add audio to a Unity or Unreal Engine project
- Generate ideas
- Generate curiosity



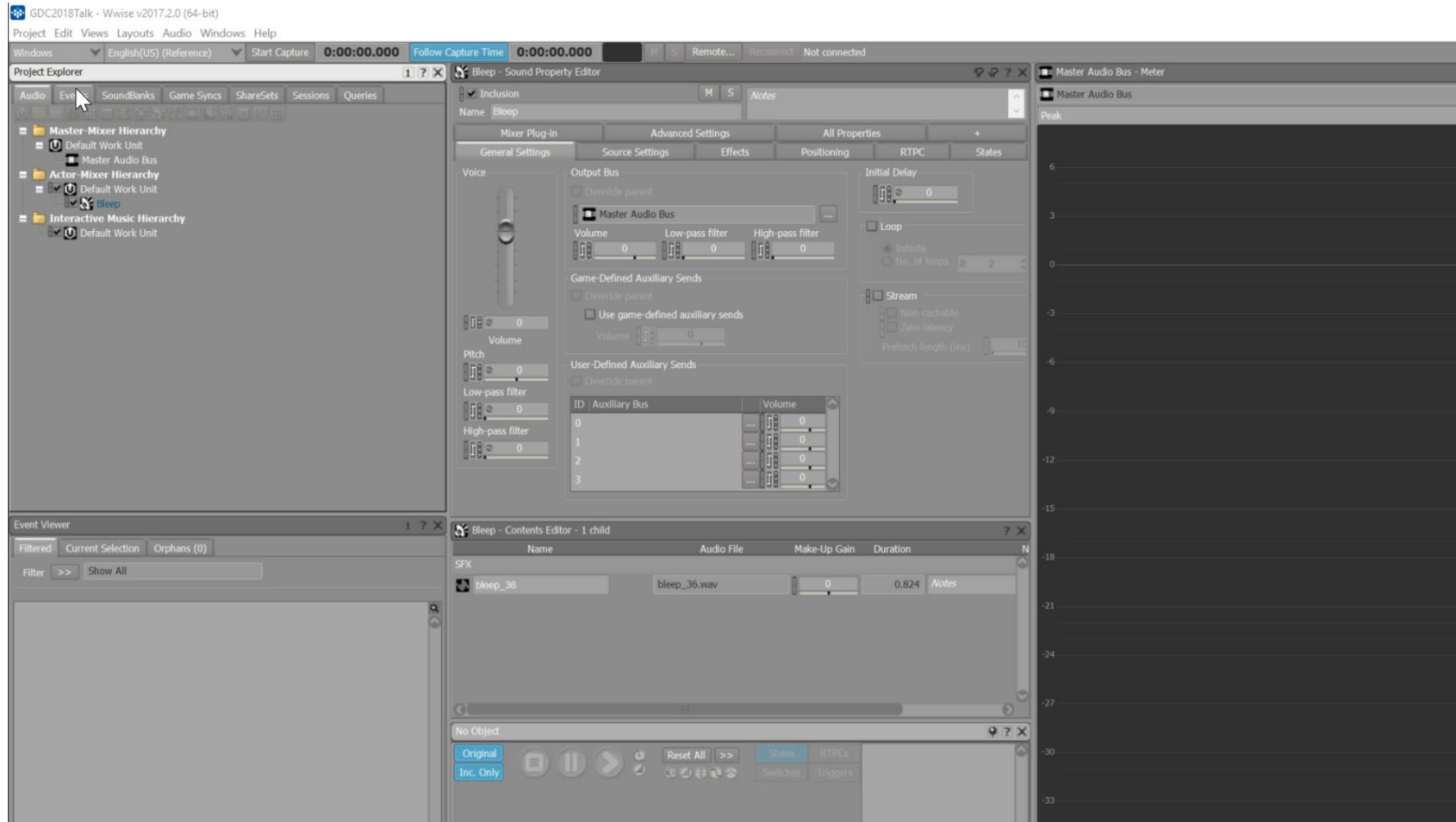


# Creating an SFX in Wwise





# Creating an Audio Event in Wwise





# The Code to Call

```
uiID = soundEmittingButton.GetID();
```

```
AK::SoundEngine::PostEvent("Play", uiID);
```





# Audio Middleware / Game Engine Audio

- [Unity audio tutorials](#)
  - <https://unity3d.com/learn/tutorials/s/audio>
- [Unreal Engine audio tutorials](#)
  - <https://docs.unrealengine.com/latest/INT/Engine/Audio/Overview/index.html>
- [Wwise tutorials](#)
  - <https://www.audiokinetic.com/resources/videos/>





# Audio

- Audio is fun! Explosions are fun!
- Great game audio community to help you learn!





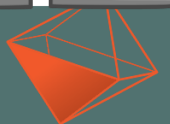
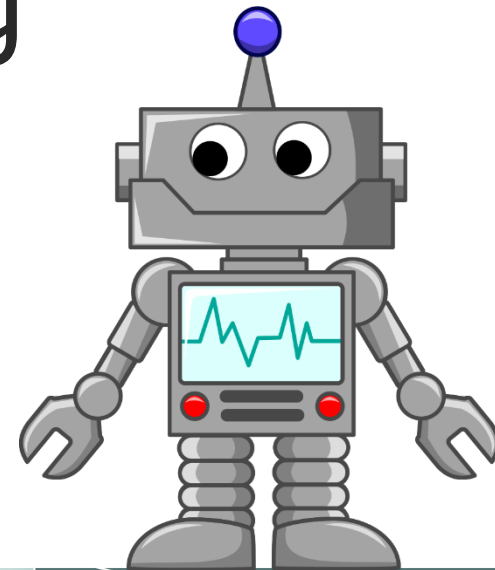


\*Plays some music stuff\*



01101000 01100101  
01101100 01101100  
01101111

# Creating a Common Dialog





# Cross Disciplinary Communications

- See your project from a different angle
- Learn new skills
- Make friends 😊





Thank you!

Twitter: @VictoriaDorn