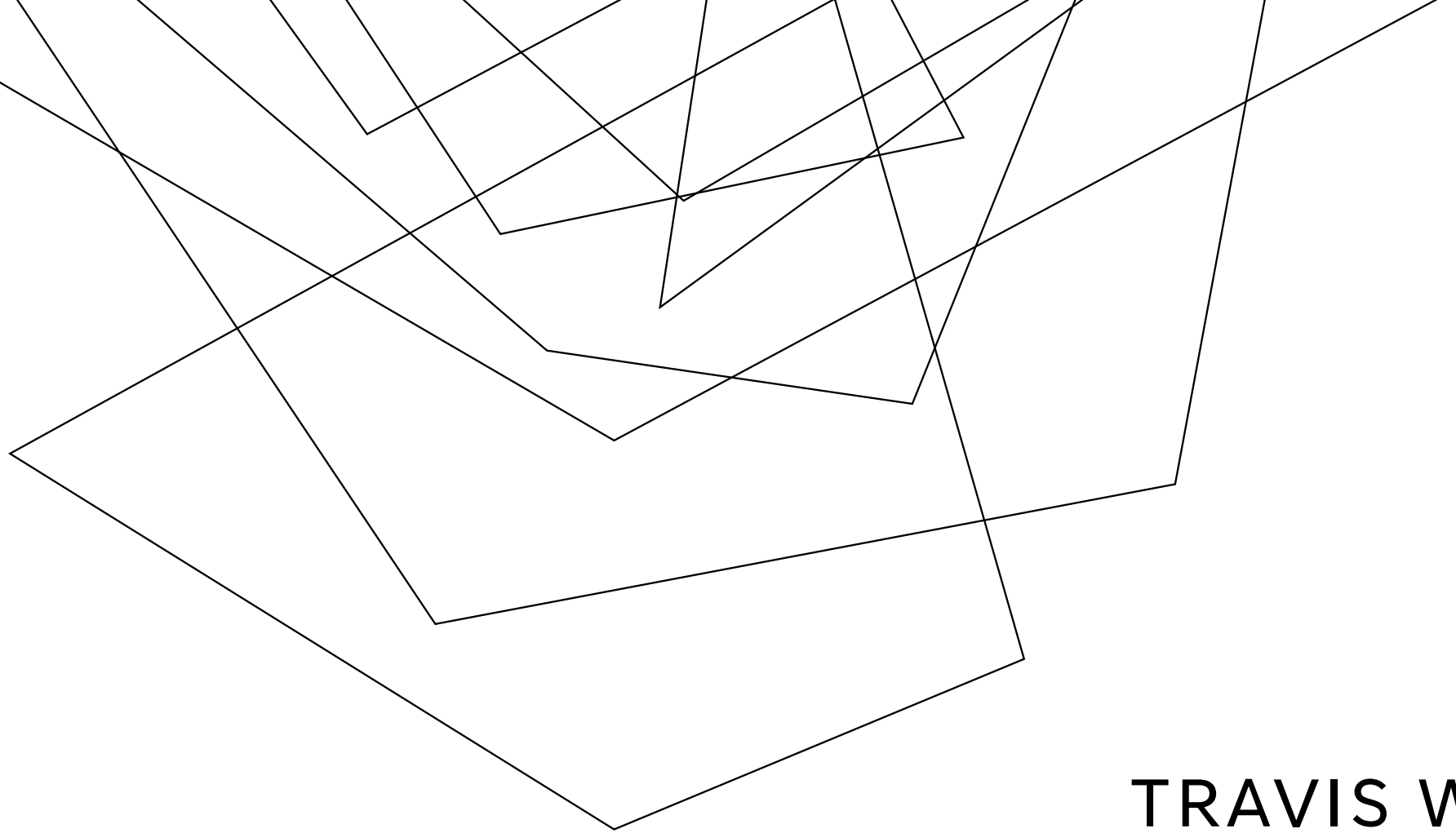
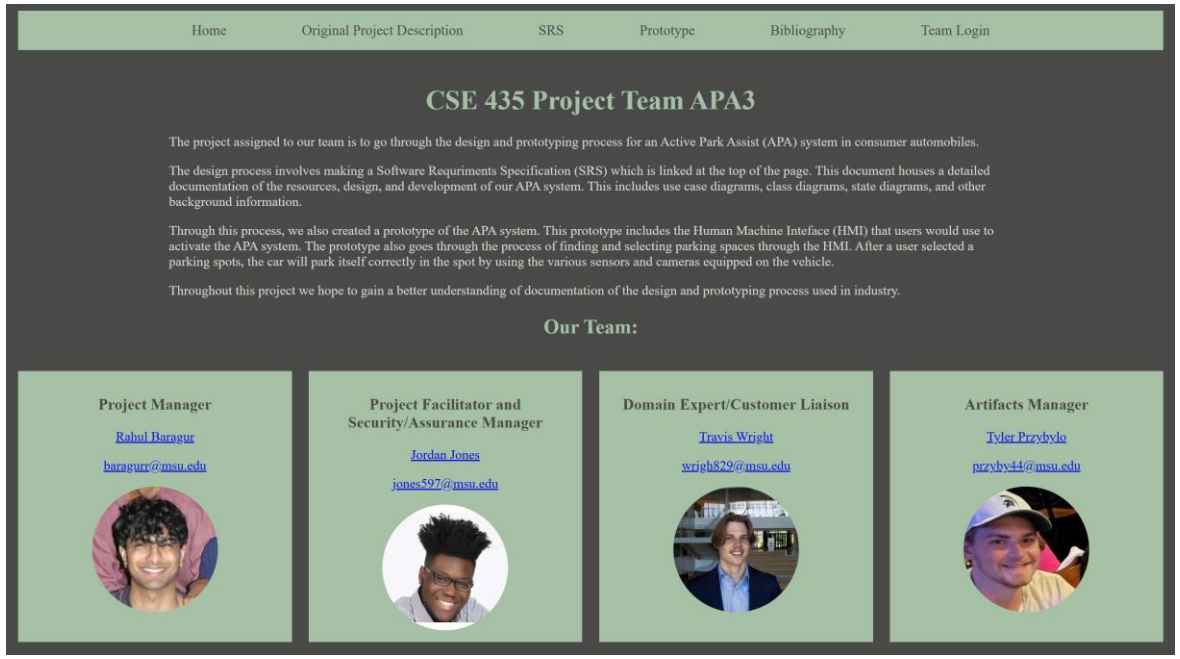


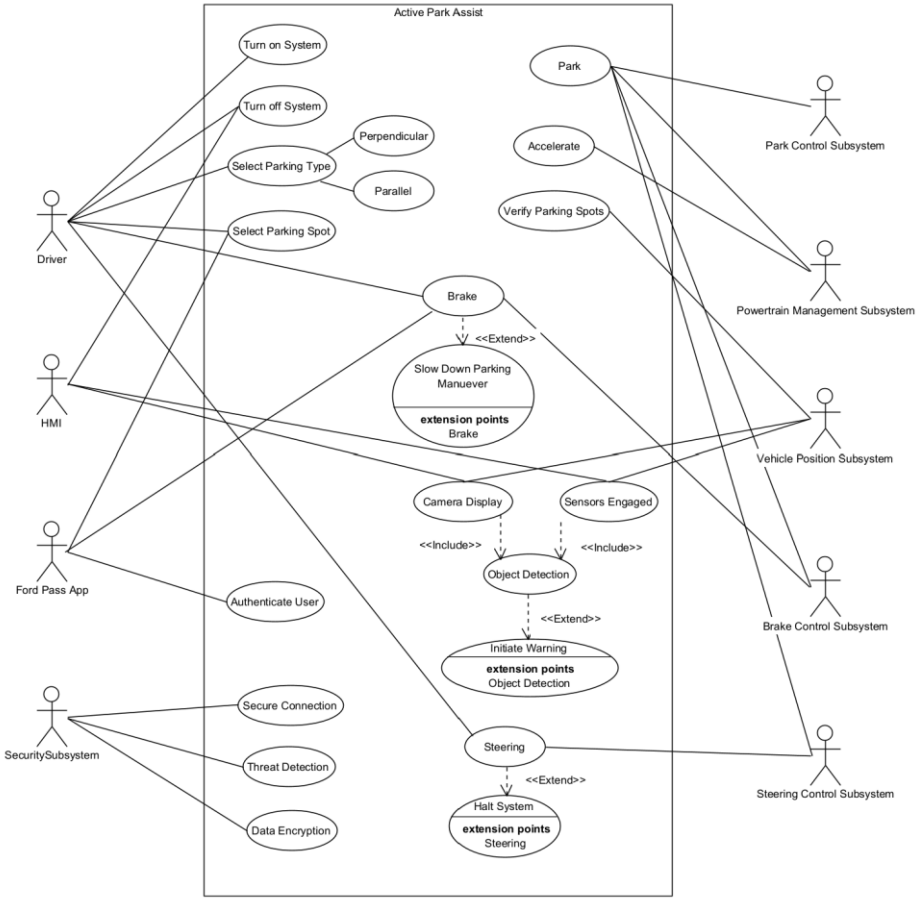
TRAVIS WRIGHT
COMP SCI - AUDIO - DATA
PORTFOLIO

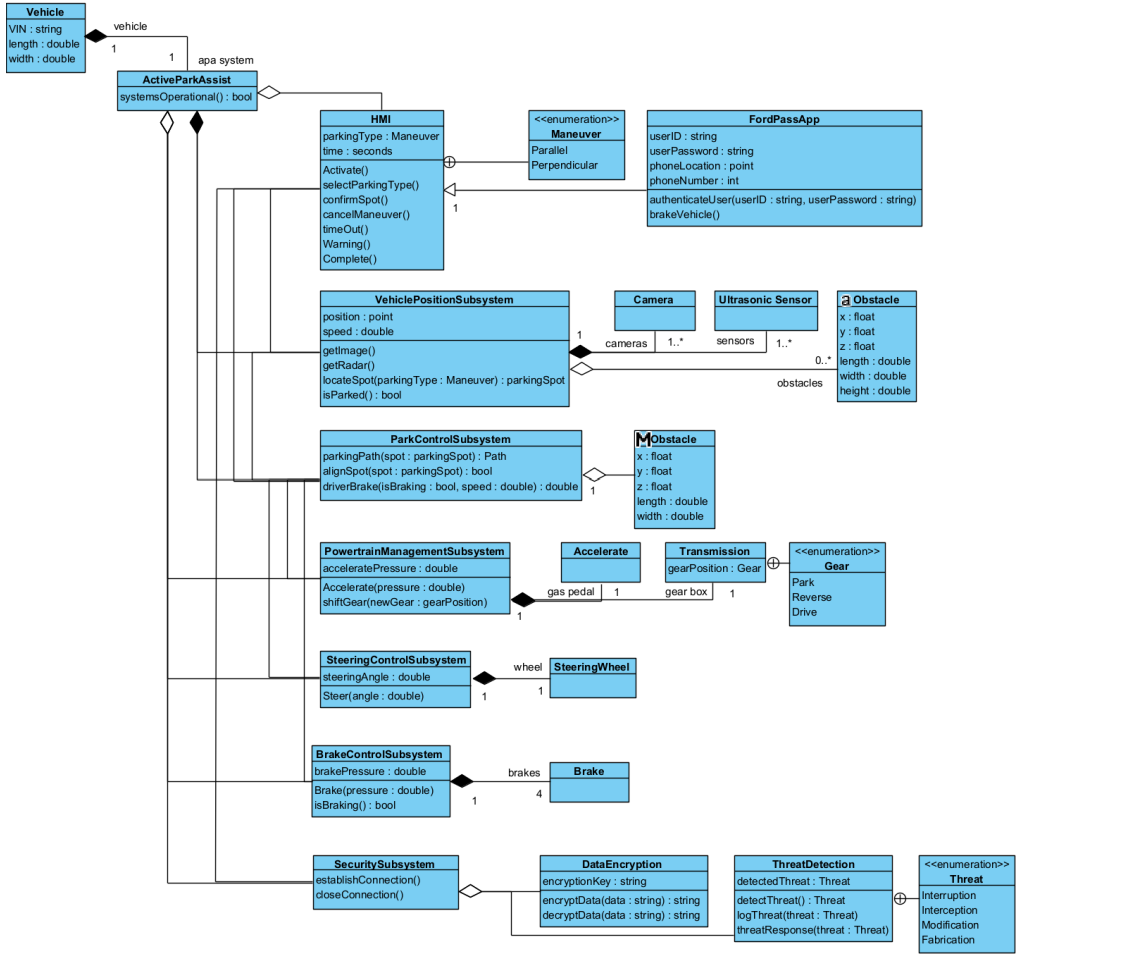
CSE435: ACTIVE PARK ASSIST

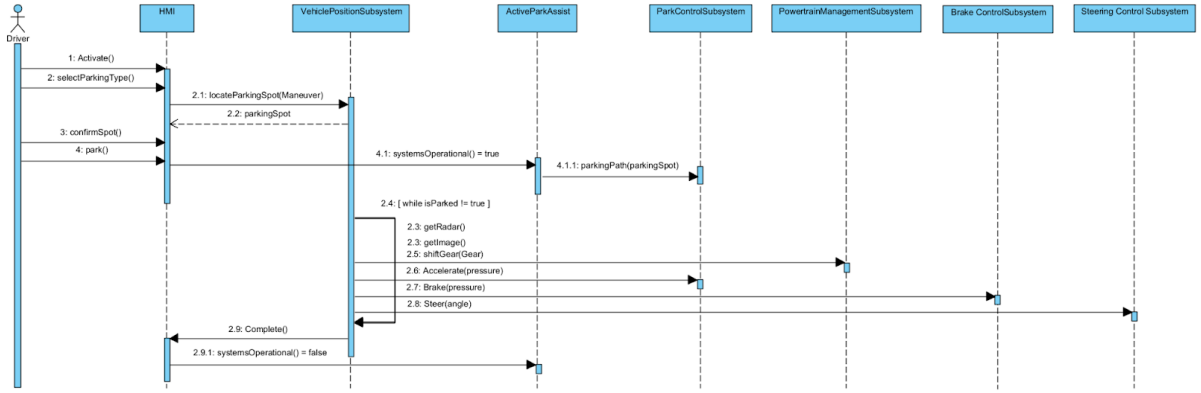
- Designed and developed the Active Park Assist system, a safety-critical feature enabling vehicles to identify parking spaces and perform automated parallel or perpendicular parking maneuvers.
- Created a detailed Software Requirements Specification (SRS) document by eliciting customer feedback, gathering and refining system requirements, and outline project constraints, details, and goals.
- Developed comprehensive Use Case, Class, State, and Sequence diagrams to model system behavior, design, architecture, and interactions providing varying perspectives for ourselves and our customer.
- Built an intuitive and interactive prototype in Unity, simulating the Active Park Assist functionality to demonstrate system capabilities and user experience.

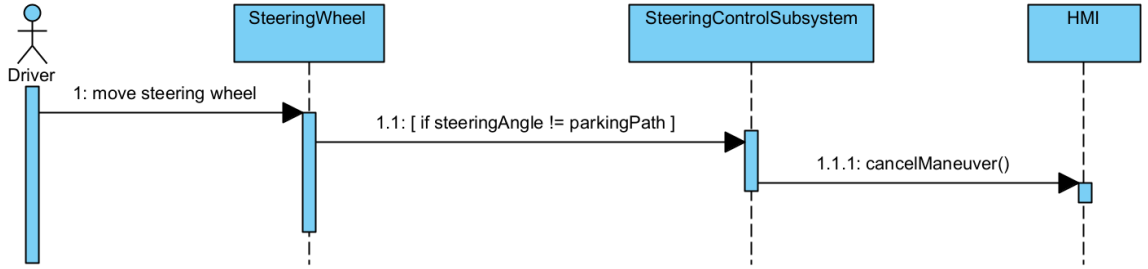
visit our website: https://cse.msu.edu/~przyby44/




Active Park Assist
Turn on System
Turn off System
Perpendicular
Select Parking Type
Parallel
Select Parking Spot
Park
Accelerate
Verify Parking Spots
Brake
<<Extend>>
Slow Down Parking Manuever
extension points
Brake
Camera Display
Sensors Engaged
<<Include>>
<<Include>>
Object Detection
<<Extend>>
Initiate Warning
extension points
Object Detection
Steering
<<Extend>>
Halt System
extension points
Steering
Authenticate User
Secure Connection
Threat Detection
Data Encryption
Driver
HMI
Ford Pass App
SecuritySubsystem
Park Control Subsystem
Powertrain Management Subsystem
Vehicle Position Subsystem
Brake Control Subsystem
Steering Control Subsystem

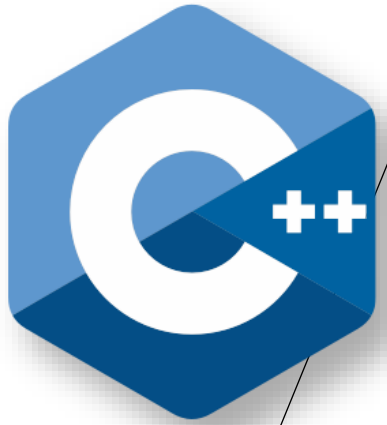

Vehicle
VIN : string
length : double
width : double
vehicle
1
1
apa system
ActiveParkAssist
systemsOperational() : bool
HMI
parkingType : Maneuver
time : seconds
Activate()
selectParkingType()
confirmSpot()
cancelManeuver()
timeOut()
Warning()
Complete()
<<enumeration>>
Maneuver
Parallel
Perpendicular
FordPassApp
userID : string
userPassword : string
phoneLocation : point
phoneNumber : int
authenticateUser(userID : string, userPassword : string)
brakeVehicle()
VehiclePositionSubsystem
position : point
speed : double
getImage()
getRadar()
locateSpot(parkingType : Maneuver) : parkingSpot
isParked() : bool
Camera
Ultrasonic Sensor
Obstacle
x : float
y : float
z : float
length : double
width : double
height : double
cameras
sensors
obstacles
ParkControlSubsystem
parkingPath(spot : parkingSpot) : Path
alignSpot(spot : parkingSpot) : bool
driverBrake(isBraking : bool, speed : double) : double
PowertrainManagementSubsystem
acceleratePressure : double
Accelerate(pressure : double)
shiftGear(newGear : gearPosition)
Accelerate
Transmission
gearPosition : Gear
<<enumeration>>
Gear
Park
Reverse
Drive
gas pedal
gear box
SteeringControlSubsystem
steeringAngle : double
Steer(angle : double)
wheel
SteeringWheel
BrakeControlSubsystem
brakePressure : double
Brake(pressure : double)
isBraking() : bool
brakes
Brake
SecuritySubsystem
establishConnection()
closeConnection()
DataEncryption
encryptionKey : string
encryptData(data : string) : string
decryptData(data : string) : string
ThreatDetection
detectedThreat : Threat
detectThreat() : Threat
logThreat(threat : Threat)
threatResponse(threat : Threat)
<<enumeration>>
Threat
Interruption
Interception
Modification
Fabrication


Driver
HMI
VehiclePositionSubsystem
ActiveParkAssist
ParkControlSubsystem
PowertrainManagementSubsystem
Brake ControlSubsystem
Steering Control Subsystem
1: Activate()
2: selectParkingType()
2.1: locateParkingSpot(Maneuver)
2.2: parkingSpot
3: confirmSpot()
4: park()
4.1: systemsOperational() = true
4.1.1: parkingPath(parkingSpot)
2.4: [while isParked != true]
2.3: getRadar()
2.3: getImage()
2.5: shiftGear(Gear)
2.6: Accelerate(pressure)
2.7: Brake(pressure)
2.8: Steer(angle)
2.9: Complete()
2.9.1: systemsOperational() = false


Driver
SteeringWheel
SteeringControlSubsystem
HMI
1: move steering wheel
1.1: [if steeringAngle != parkingPath]
1.1.1: cancelManeuver()

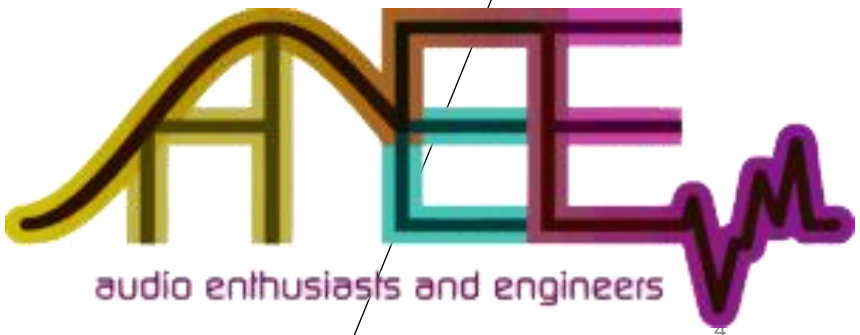
C++ EQUALIZER

JUCE



AEE Software Project Team-Leader

- Utilized the JUCE framework, an open-source C++ library for audio application and plugin development.
- Led a team of audio enthusiasts coordinating tasks and ensuring a timely delivery while maintaining and teaching coding standards.
- Implemented user interface components for real-time audio adjustments, enabling seamless interaction and precise control over sound input and output.
- Conducted thorough testing and debugging to ensure stability and performance across various audio environments and platforms.



C++ Equalizer

The plugin seamlessly integrates with Windows AudioFilePlayer and JUCE's Plug-In Host, enabling users to effortlessly select and play their desired tracks while unlocking a suite of advanced features:

- Real-time Spectrum Analyzer
- Custom Visual Sliders
- Bypass Buttons

Adjustable Parameters:

LowCut Filter

- Adjust slope (12, 24, 36, 48 dB/Oct)

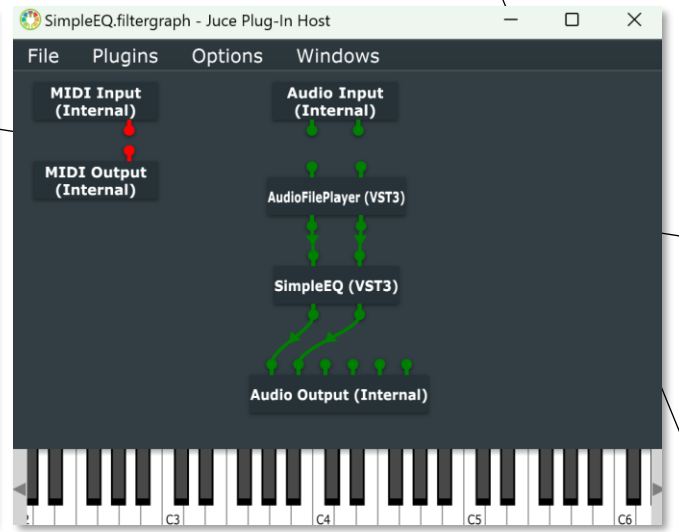
HighCut Filter

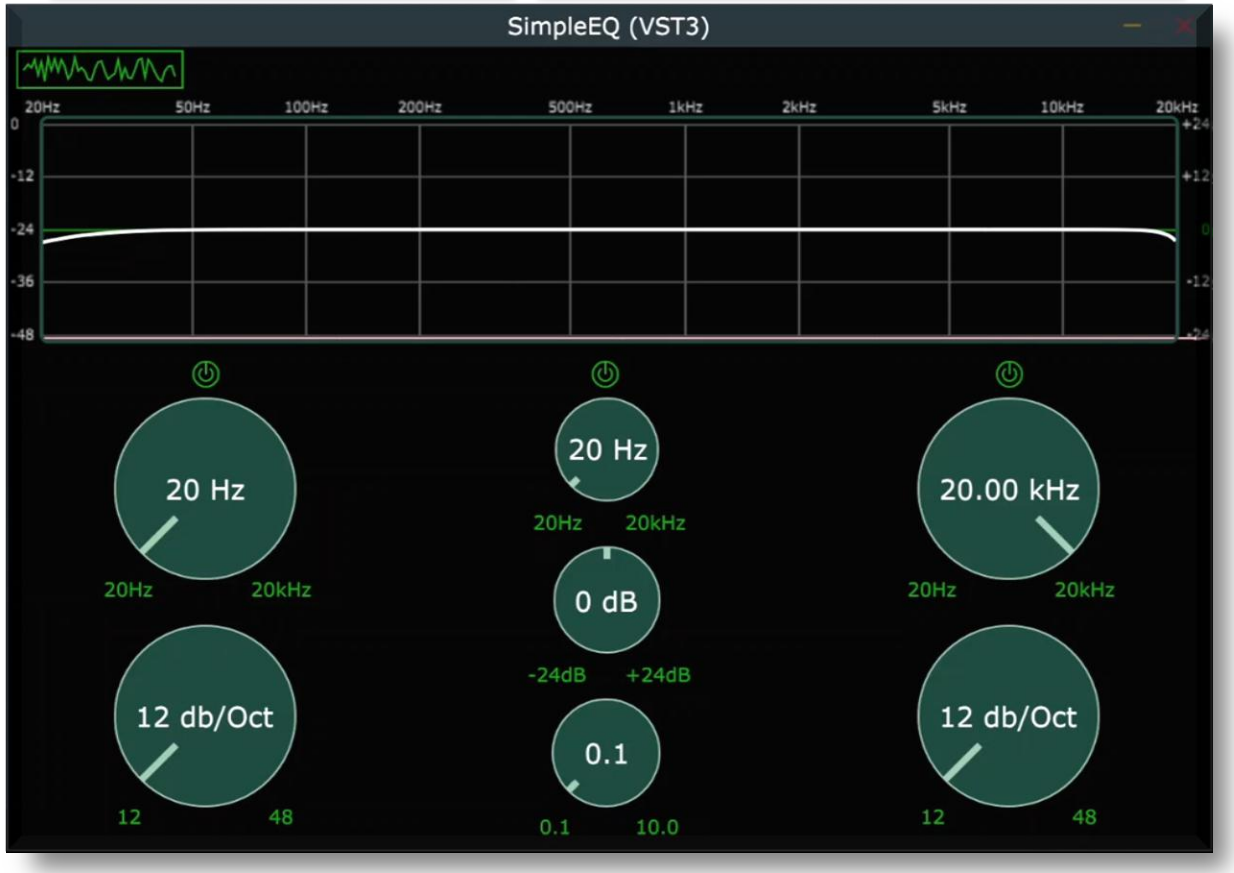
- Adjust slope (12, 24, 36, 48 dB/Oct)

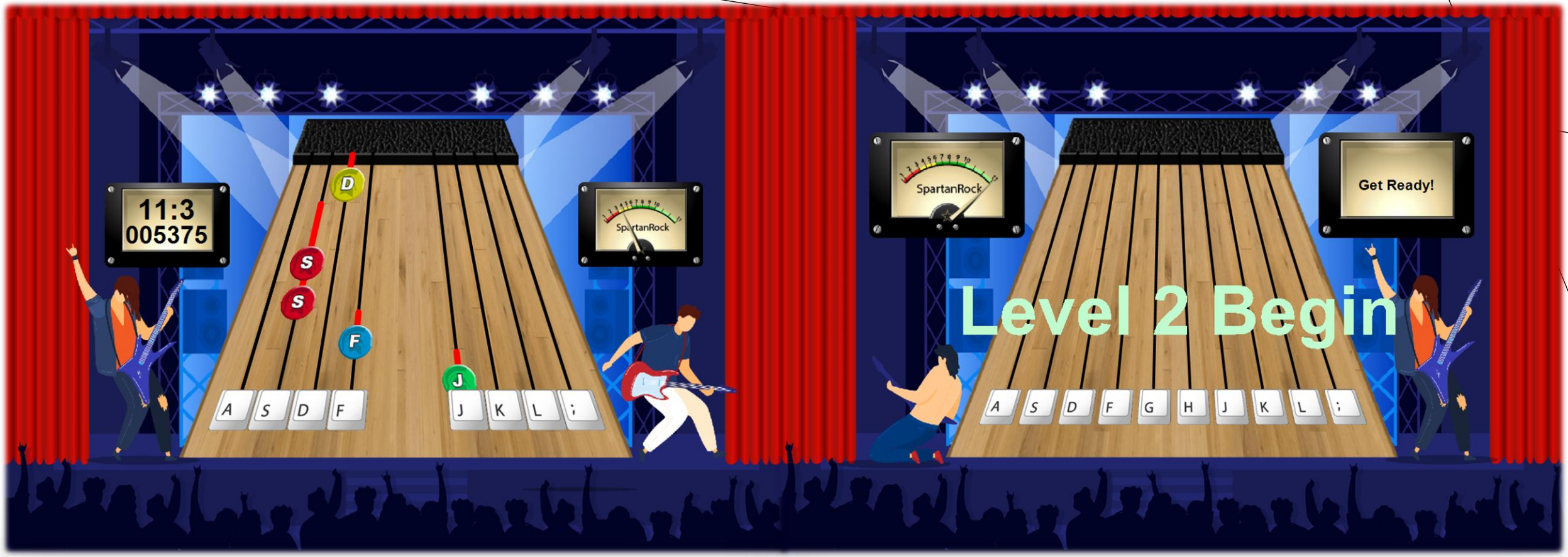
Peak Frequency

- Select specific frequency (20hz – 20kHz)
- Increase/decrease gain (-24dB - +24dB)
- Adjust quality factor (0.1 – 10.0)

demo video ➡







CSE335: SPARTAN HERO

Designed and developed an interactive music game reminiscent of Guitar Hero, featuring multiple levels and intricate mechanics. Created in C++ using ‘wxWidgets’ and ‘miniaudio’ libraries to manage audio playback, user interactions, scoring, and ensuring a fluid, immersive experience.

The month-long project was constructed using SCRUM techniques to manage sprints and hold regular team meetings, ensuring effective task coordination with an emphasis on robust testing and debugging:

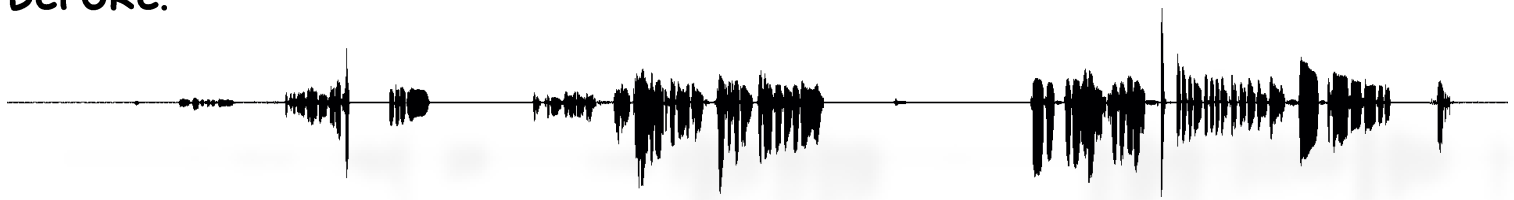
- Utilized Trello for team-productivity management
- Adhered to Google’s testing standards
- XML file load and save compatibility

NOISE REMOVAL SCRIPT - PYTHON

Project

- Developed a Python script using Pydub and Numpy to manipulate audio files and enhance the workflow at WLNS-TV Channel 6 News.
- The script intelligently identifies and removes background noise by dynamically detecting silence based on a customizable dB threshold, with buffer extensions to prevent abrupt cuts in audio.
- Ensures only meaningful audio is retained by defining minimum segment length and merging overlapping segments, enhancing the clarity and continuity of the final output.
- Utilized by news reporters and on-air anchors when recording long audio segments in the field. The script aims to assist in the final editing process of news casts, stories, and voice overs.

BEFORE:

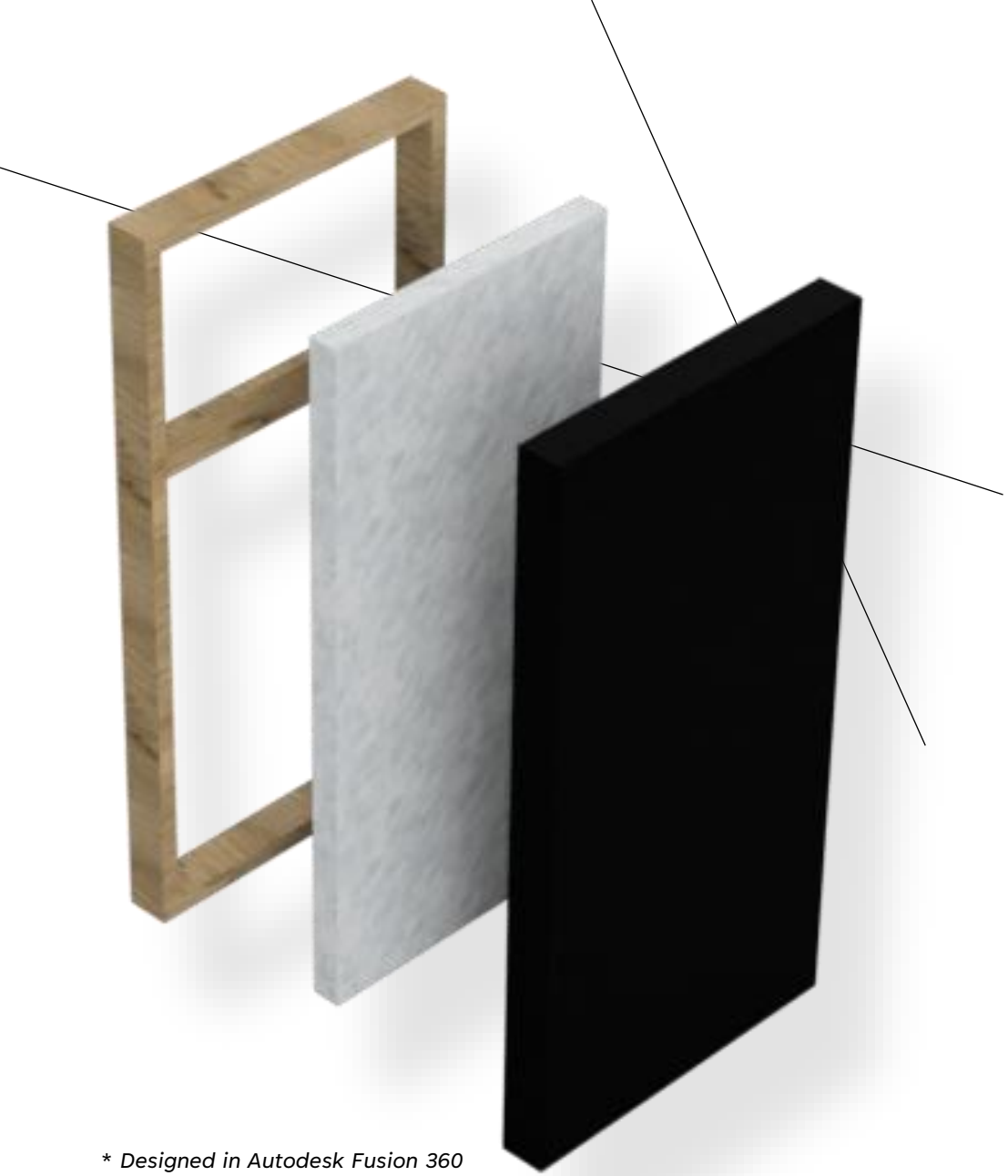


AFTER:

Pydub

NumPy

WLNS



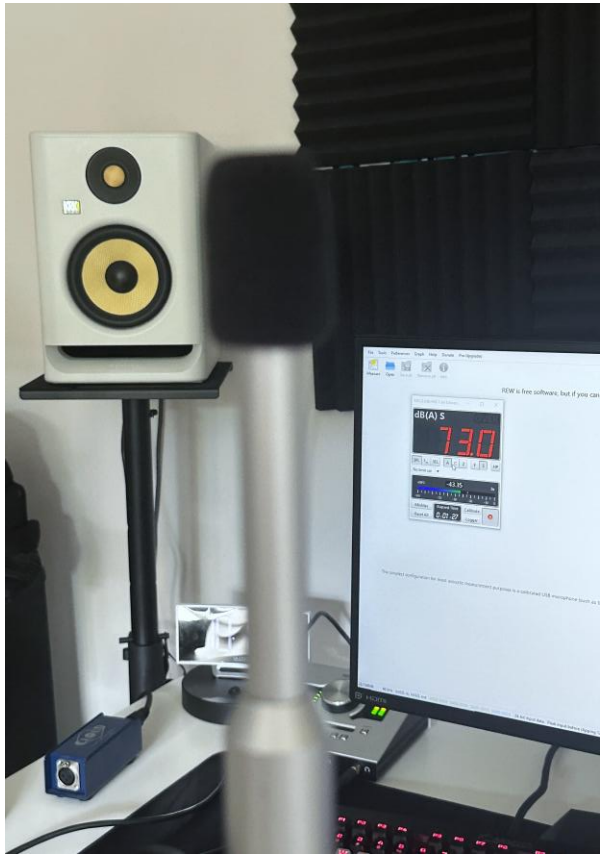
** Designed in Autodesk Fusion 360*

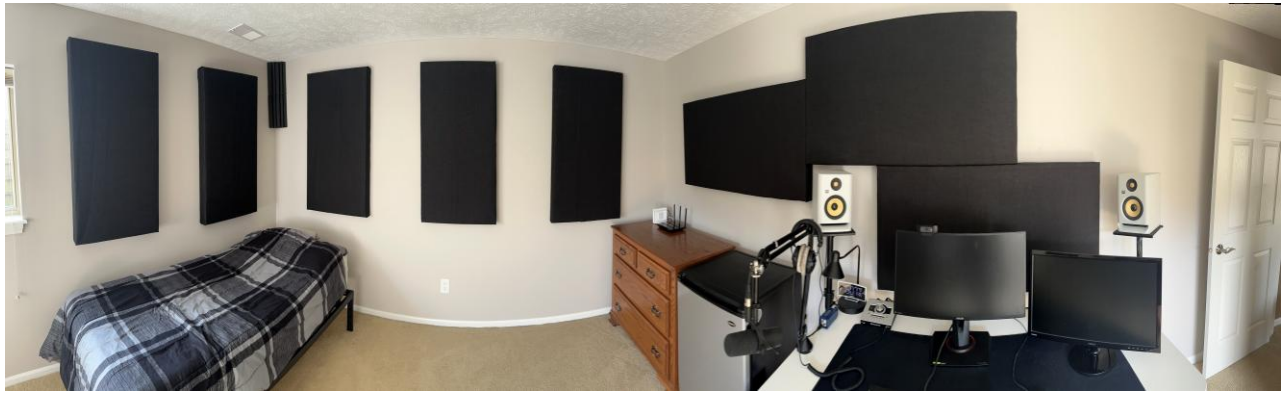
ACOUSTIC PANELS

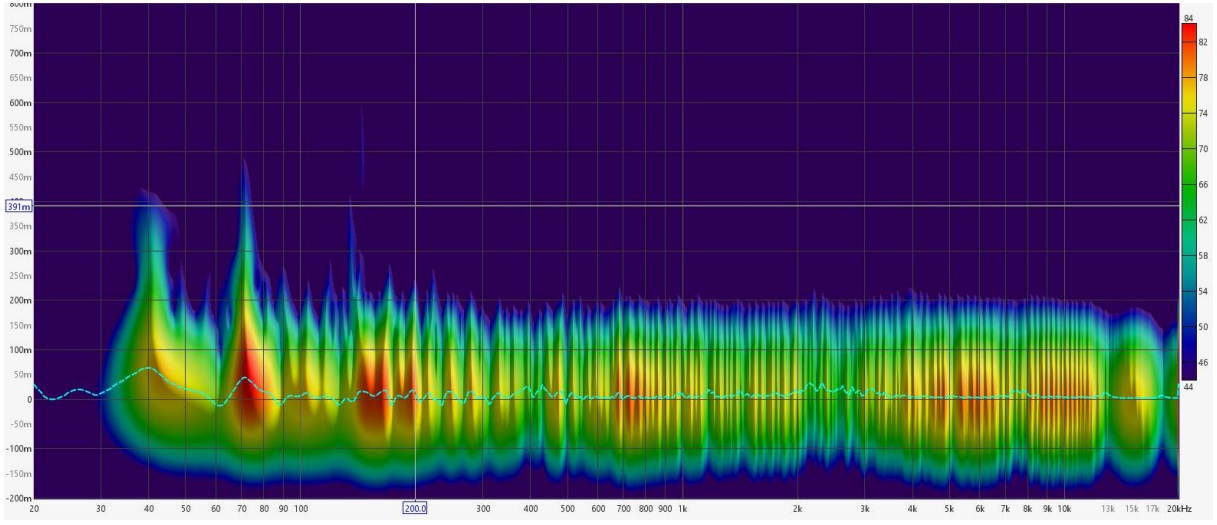
I designed and **hand-crafted eight** custom **acoustic panels** using Nu-Wool ECOCELL blankets: a **sustainable** product made from at least **70% recycled materials**, including post-consumer recycled newspaper, which helps reduce waste in landfills. Each panel measures **4ft x 2ft x 4in** providing excellent **sound absorption**. The construction includes a sturdy pine frame and black burlap fabric.

After installing the panels, I utilized **Room EQ Wizard (REW)** to assess the effectiveness of the acoustic treatment. Using a **Behringer ECM8000** condenser microphone, I **conducted detailed measurements** of the room's acoustics, focusing on decay time and reflections.

This process allowed me to quantitatively validate the impact of the panels and fine-tune the setup for optimal acoustic performance, and gaining valuable experience in **data analysis** and **data acquisition**.


dB(A) S
73.0




750m
700m
650m
600m
550m
500m
450m
391m
350m
300m
250m
200m
150m
100m
50m
0
-50m
-100m
-150m
-200m
20
30
40
50
60
70
80
90
100
200.0
300
400
500
600
700
800
900
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
13k
15k
17k
20kHz
84
82
78
74
70
66
62
58
54
50
46
44

GOOGLE DATA ANALYTICS

Overview:

Completed the Google Data Analytics Certificate, mastering essential **data analytics** concepts and techniques. This comprehensive program provided **hands-on experience** in **preparing**, **processing**, and **analyzing** large datasets to uncover actionable insights.

Skills:

Developed expertise in **data cleaning**, **analysis**, and **visualization**, with a focus on **data-driven** storytelling and interpreting **trends** to support business objectives. Enhanced ability to create impactful **dashboards** and reports that drive **informed decision-making**.

- Excel: data analysis and organization using *formulas* and *pivot tables*
- SQL: *queried*, *analyzed*, and *managed* data in *Google's BigQuery*
- R: *manipulated* and *displayed* data using the *tidyverse* package
- Tableau: developed interactive *dashboards* and compelling *visualizations*

THANK YOU

traviswright131@gmail.com
in/travisdwright/