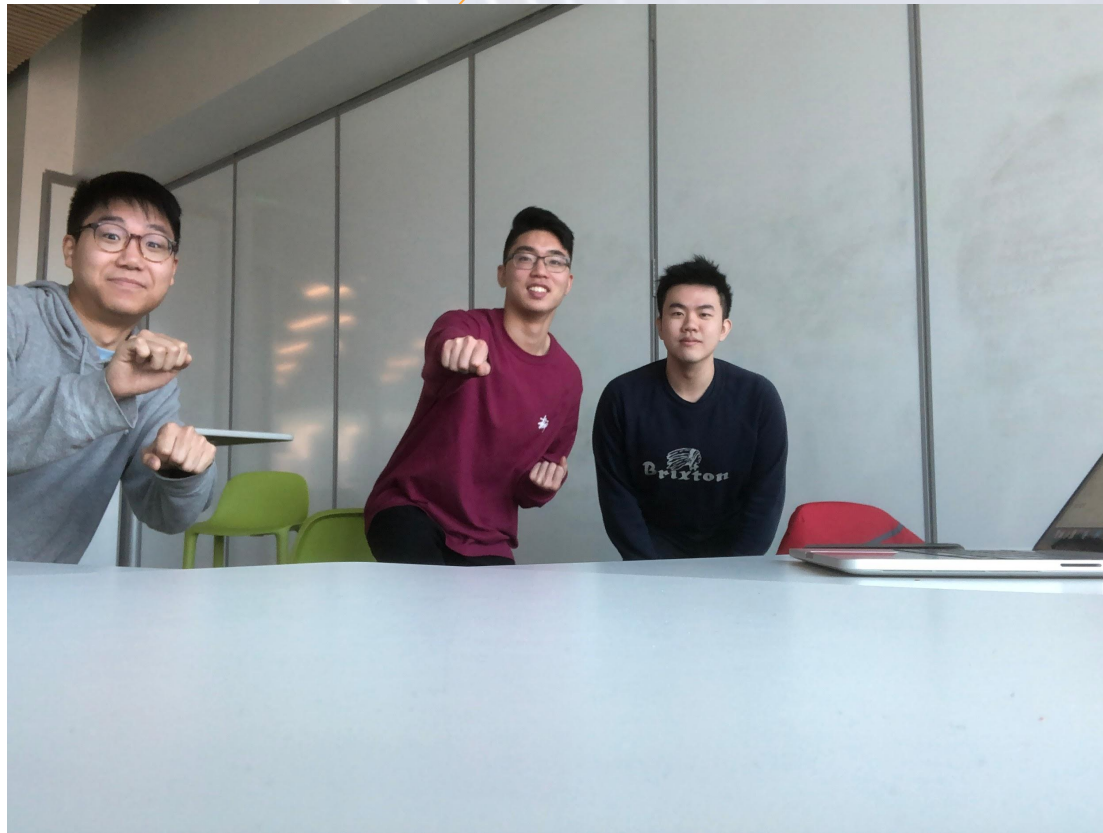




# **Caprice:** **A motion-controlled synthesizer**

Team E3: Michael Wang, Jeffrey Li, Jason Hsu

# The Team





# Use Case

What do we want to address?

- More freedom of expression for a beginner musician
- Lower learning curve vs. traditional instruments
- ECE Areas: Software, Signals



# Use Case

## Our Idea

### Controller (Left)

- 8 buttons toggles different notes
- Bluetooth

### Controller (Right)

- Controls note duration with motion
- Toggle filters
- Octave shift
- Major/Minor mode
- Chromatic shift
- Bluetooth

### Laptop

- Runs custom software for processing, real-time sound, filters, etc.
- Key selection

# Solution Approach

## Controller (Left)

- Smartphone application (React Native)
- Custom controller hardware/grip
- Bluetooth connection

## Controller (Right)

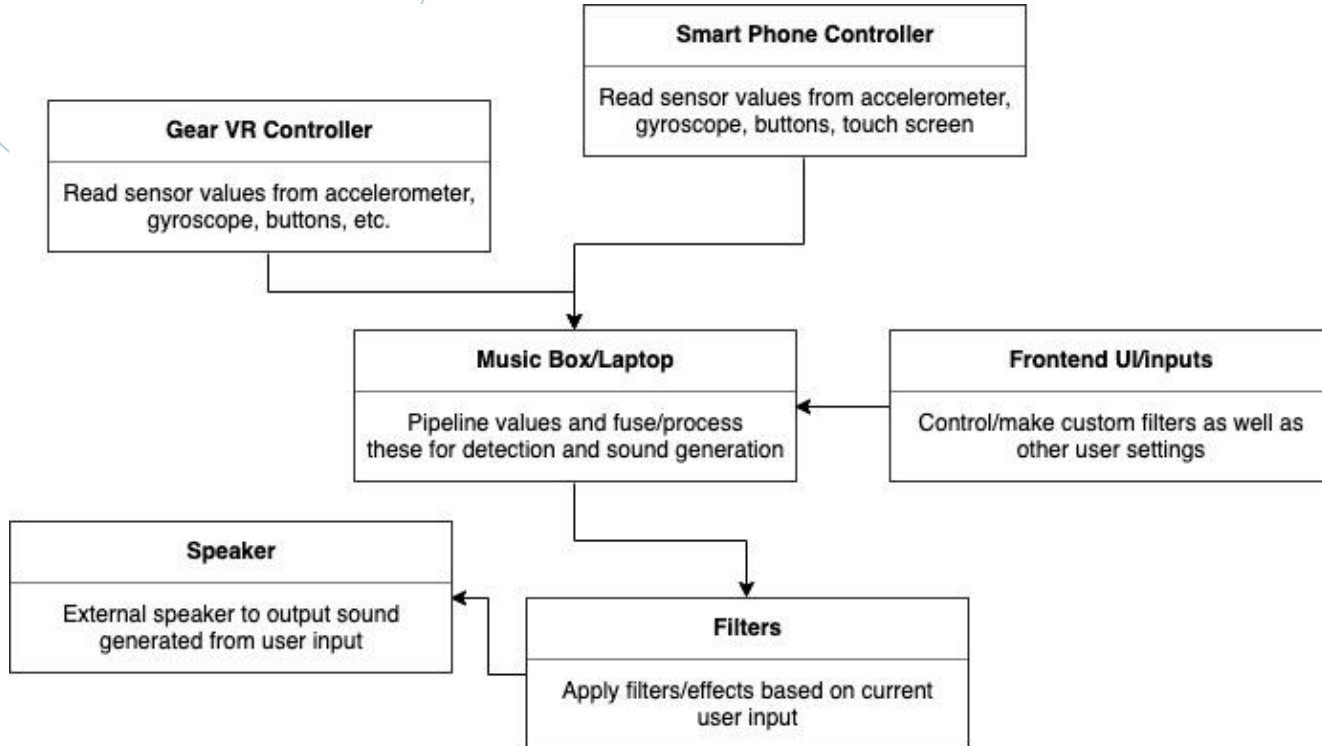
- Samsung GearVR controller
- Reverse engineer the GearVR communication protocol to read controller bluetooth service identifiers
- Store identifiers/device address for future connections

## Laptop

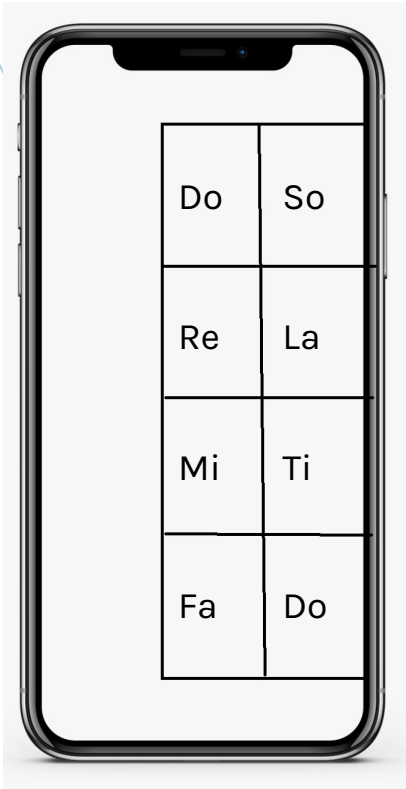
- pybluez: connect to both devices, subscribe to services
- mido: generate real time MIDI instructions
- kivy: GUI framework
- Custom code to process MIDI to sound wave
- Apply filters



# Solution Approach

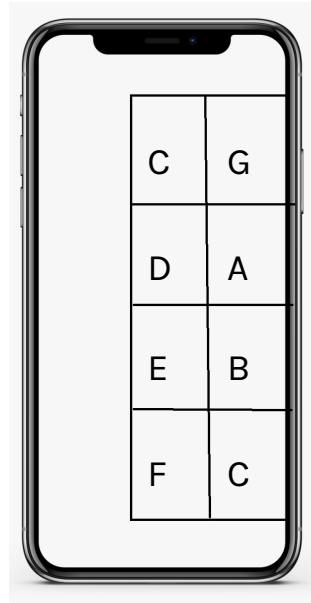


# Left Hand Gestures

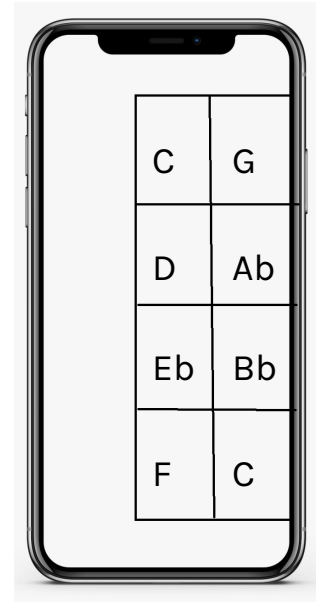


Solfege Representation

- Grid of 8 buttons representing solfege notes in the selected key



Key of C Major



Key of C natural minor

# Right Hand Gestures



Chromatic Shift






Filter Toggle



Octave Shift



Volume

-  Click down
-  Hold finger (no click)
-  Swipe



# Requirements

## Latency

- <100ms latency from controller action to audio

## Pitch

- 8.18 Hz to 12500 Hz (MIDI range)

## Volume

- 0-75dB (Living room music)

## Effects & Filters

- Vibrato
- Delay
- Fuzz/Overdrive

# Testing, Verification, & Metrics

## Latency

- Time audio processing code

## Pitch

- Compare note with known frequency to played frequency, assure played frequency is within 5 Hz of the desired note

## Volume

- Measure volume with measurement device, compare with expected volume.

## Effects & Filters

- Vibrato:  $2 * \text{amplitude} \leq 1$  semitone
- Delay: 30ms - 1000ms

# Tasks & Division of Labor

## Michael

- Sensor data pipeline
- UI/Integration

## Jason

- Real-time music generation
- Signals processing

## Jeffrey

- Sensor data pipeline
- Bluetooth stack

