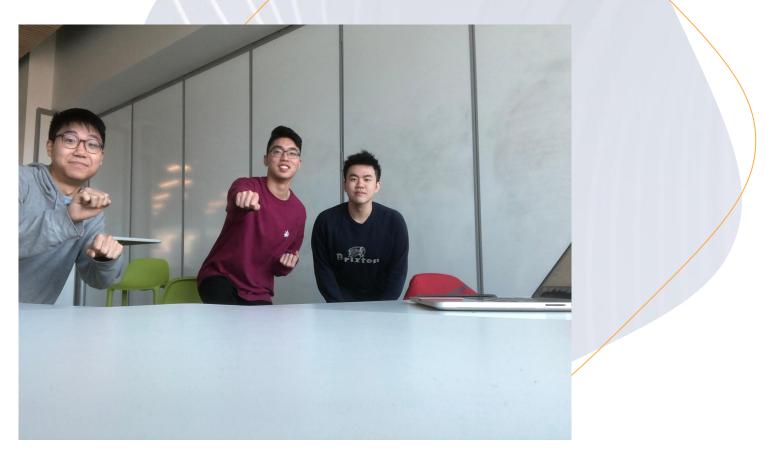
# Caprice: A motion-controlled synthesizer

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

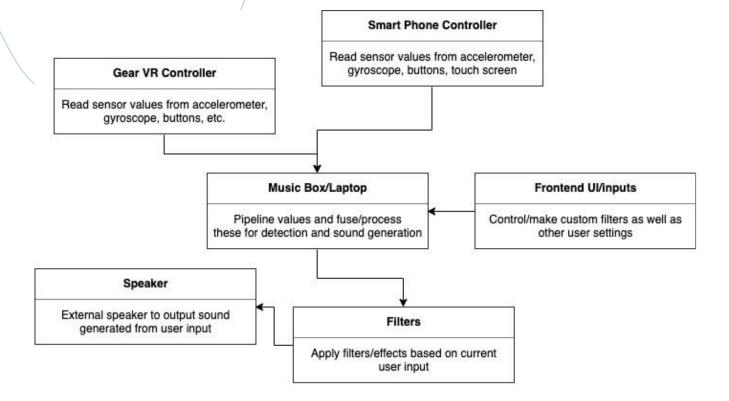
#### Laptop

- 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

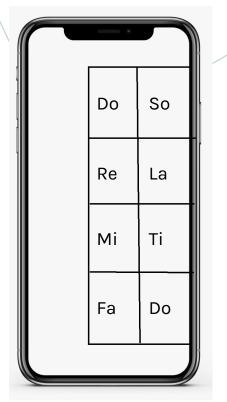




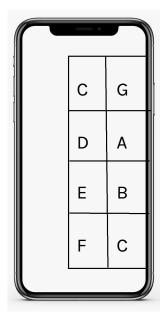
### **Solution Approach**

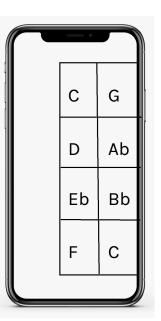


### Left Hand Gestures



Grid of 8 buttons representing solfege notes in the selected key





Solfege Representation

Key of C Major

Key of C natural minor

### **Right Hand Gestures**



Chromatic Shift

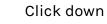
Filter Toggle

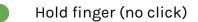




Octave Shift

Volume





Swipe

### Requirements

#### Latency

<100ms latency from controller action to audio</p>

#### 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\*amplitude <= 1 semitone</p>
- 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

### Schedule

		Week														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		1/13	1/20	1/27	2/3	2/10	2/17	2/24	3/2	3/9	3/16	3/23	3/30	4/6	4/13	4/2
esearch																
Blu	uetooth Stack Research															
So	ound Generation Research															
Se	ensor Fusion Research															
VP Building																
Let	ft/Right Hand Gesture Detection												Jeffrey			
Re	ead Sensor Values from GearVR												Jason			
De	etect and Label Gestures from GearVR					6							Michael			
Re	ead Sensor Values form Smart Phone												Jeffrey/Jason			
Blu	uetooth Protocol w Laptop/Smart Phone								5				Jeffrey/Michael			
Sla													Jason/Michael			
So	ound Generation/Filtering												All			
Ge	enerate MIDI signal in real time															
Co	onvert MIDI signal to audio output in real time															
	oply effects/filters to real time audio signal															
Sla	ack															
	nking Detection/Sound (Integration)															
	uetooth Connect w/both devices, process sensor dat	ta														
Re	eact Native Frontend for Smartphone															
Kiv	vy Frontend for Laptop								24							
Inte	tegrate Audio Processing Backend															
UI																
Sla	ack															
urse Logistics																
Ab	ostract															
Pro	roject Proposal															
De	esign Review												-			
Tes	sting Phase															
Pu	ublic Demo and Report															1