Outline
Device: Pebble, Nexus 7
Function:

- Pebble
  - use embedded accelerometer to grab ax,ay,az
  - detect the largest magnitude value in every 10 seconds, smooth, find peak?
  - send to Android via bluetooth
  - **Need to disable vibration**
- Android
  - “Listener Interface” to receive acce data from bluetooth
    - on receiving the data:
      - trigger a func to play a piece of music
      - light up some area in the activity
  - A func to decide which
  - UI mock up
  - Add on features
    - different drums/whips/anything related to a HIT
    - various number of drums

Left/right hand player

1. **DISTRACT BACKGROUND**
   10s
2. **store waveform and analysis**

Pebble Air Drum App Description

I. Description

Pebble Air Drum is an Android app. It uses Pebble Watch as an input of user gestures, and let the user to play simulated drums displayed on smartphone.

When the user launches an app, he first chooses which hand wearing pebble watch. Then the user will go to the screen with simulated drums.
In playing the drum, to obtain accuracy, the user should keep the pebble watch face upward. If the user wants to beat a specific drum, he can turn the watchface slightly toward the direction of the intended drum when beating.
There's background beats for user to keep track of, which automatically starts when enterring the game screen.

II. Implementation

Pebble Air Drum contains two parts, the Pebble part and the Android part.
The Pebble part program is written in C, it reads and processes the data from accelerometer sensor, and send processed results to Android phone. The sampling rate of the sensor is 10 Hz, and every 5 samples forms a batch, and calls a data handler. The handler judges whether there is a beat by calculating the combined acceleration. When the watch is in static, there is a stable value of accelerometer sensor, which is the gravity. The beat gesture will cause a weight loss, and the combined acceleration will decrease sharply. After Then the handler judges a beat, it calculates the offsets of the sampled data and adjust the data by subtracting the offsets. The adjusted data then integrated and used to judge whether the user is beating towards front, back, left or right. In the end, the handler sends the result to Android phone.

The Android application targets API level 19 (Android 4.4) and the UI interface was designed for running on Nexus 7 or other device with same resolution. Pebble watch and Android application communicates in AppMessage.

III. Test Results

Latency:
There is a perceptible latency in the Android app reaction. Since the data is processed in batch, not strictly real time, there will be a delay in Pebble watch process the data, and send the data to Android phone. However, the latency is acceptable.
Accuracy:
The user could obtain a good accuracy if performing "as expected".


CLOUDPEBBLE
WITHDISTANCE
SETTINGS
COMPILATION
GITHUB
SOURCE FILES
ADD NEW
main.c
RESOURCES
ADD NEW
IMAGE_DRUM.png
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:106: front!
[DEBUG] main.c:128: right!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! 1 -1
[DEBUG] main.c:54: next data discarded **
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:135: stay!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! 0 0
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:106: front!
[DEBUG] main.c:120: left!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! 1 1
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:128: right!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! 0 -1
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:113: back!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! -1 0
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:113: back!
[DEBUG] main.c:120: left!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! -1 1
[INFO] main.c:252: Outbox send success!
[DEBUG] main.c:54: next data discarded **
[DEBUG] main.c:106: front!
[DEBUG] main.c:194: sent!
[DEBUG] main.c:195: sent! 1 0
[DEBUG] main.c:54: next data discarded **
[INFO] main.c:252: Outbox send success!


4:14
Pearl
PebbAirDrum