GitHub Username: Kimo9985

Swing Tracker

Description

Swing Tracker will allow a tennis player to track forehand, backhand and overhead swings by combining sensor readings from the wearables Accelerometer and Gyroscope. This tool will allow recreational players to get a better estimate of their complete workout and serious players a means to quantify practice.

Intended User

Tennis Players

Features

List the main features of your app. For example:

- Combines the Accelerometer and Gyroscope to determine type of swing
- Utilizes a low-pass filter to reduce noise
- Counts forehand, backhand and overhead smash
- Allows user to select whether they are left handed or right handed
- Allows user to reset counts
- Battery level on main screen for convenience

User Interface

Screen 1

Main Screen - Counter Fragment

Screen 2

Settings Fragment

Key Considerations

How will your app handle data persistence?

Swings collected from the sensors will be saved as SharedPreferences.

Describe any corner cases in the UX.

The UX will be created using ViewPager. There will be 2 pages (Counter Screen and Settings Screen). This wearable will be compatible with Round and Square Watches.

Describe any libraries you'll be using and share your reasoning for including them.

I may need to include the org.apache.commons.math3 library if a more advanced filter is needed to reduce noise and smoothness.
http://mvnrepository.com/artifact/org.apache.commons/commons-math3/3.0

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create new project in Android Studio
- Setup Gradle builds:
    - Add Google Play Service Wearable
    - Add Google Support Wearable
    - Configure Apache Commons Math3 Library (if required)

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - Retrieve sensor readings from Accelerometer and Gyroscope
    - Implement low-pass filter to reduce noise
    - Determine type of swing with Accelerometer and Gyroscope readings
- Build UI for PagerAdapter

    - Allow app to support multiple page fragments
- Build UI for SwingCounterFragment
    - Setup xml for Screen 1
    - Update swing counts
- Build UI for SettingsFragment
    - Setup xml for Screen 2
    - Allow reset button
    - Allow left/right handed user profile
- Build UI for Utilities
    - Setup SharedPreference to remember count and user profile

Task 3: Motion Sensor & Low-Pass Filter

- Implement motion sensor to detect Accelerometer and Gyroscope signals
    - Only use Accelerometer signals that surpass given threshold parameters
    - Use y and -y axis of Gyroscope to determine arc of swing
    - Combine signals to determine type of swing
- Implement Low-Pass Filter to reduce signal noise and prevent false readings