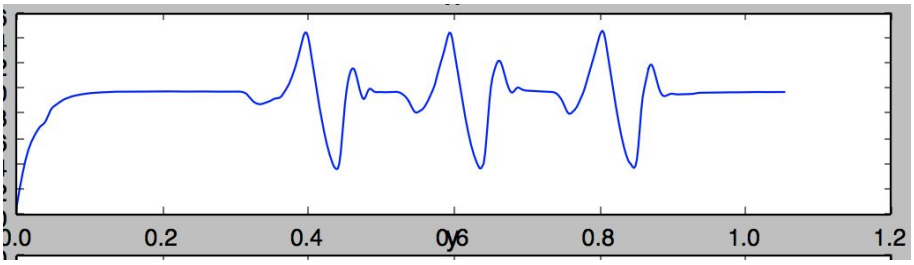
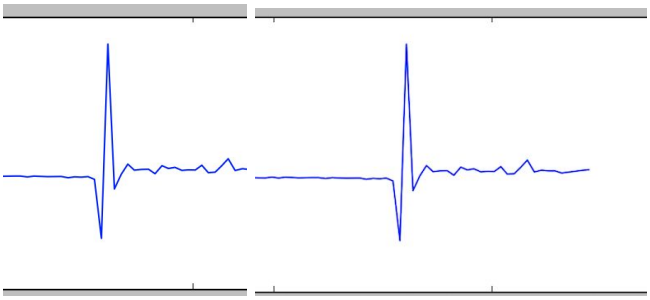
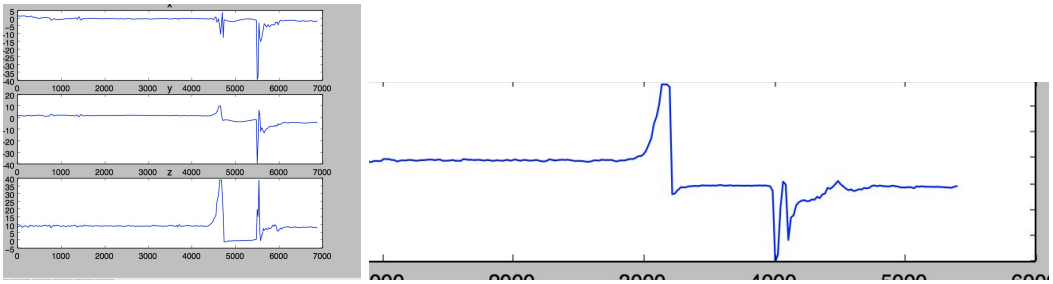
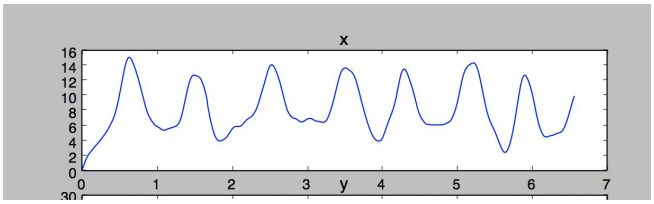
Group 5 Final Write Up

- We used given source code and created 4 games.
    - Jump: Test how high you can jump
    - Shot Put: Test how far you can throw your phone follow the rule of shot put
    - Phone in sky: Test how high you can throw your phone
    - Spin phone: count phone spinning times
- Jump
    - 

    - The moment a person left ground, the acceleration is the biggest
    - The moment a person at top, the acceleration is the smallest
    - When rising in air, the acceleration is -9.8m/s^2
    - We use this calculate speed and find distance by V*T(upward)
- Shot Put
    - 
    - Find the starting point when hand speed up
    - Find the max acceleration point where phone “leave” hand
    - Calculate average acceleration and assume the angle is always 45 degree
- Phone in sky
    - 

    - When you actually throw a phone, the phone reach State of weightlessness and we find z score is very close to zero
    - Similar to jumps, we find the time between highest acceleration and the moment z is close to zero
    - We use magnitude in the end(close to 1.3), it’s steadier this way.

- Spin Phone
  - 

  - Step detection is pretty good for this

- What's hard for us
  - We spend hours on understanding how xml file, fragments, main activity, algorithm and service work
  - It's hard to reach the ideal situation as we sampled in real life, some results are a little off. Just changing parameters is not enough to reach high accuracy.
  - In the end, we still couldn't make single accelerometer service work for every function in one build. We tried to make different service class for different activities, but it's still not working for every single one. In order to make all function work, we have to build 4 times with mSensorManager register to a different listener. We must have missed some part in our service code to connect every pieces.

```
//mSensorManager.registerListener(jumpDetector, mAccelerometerSensor, SensorManager.SENSOR_DELAY_GAME);
//mSensorManager.registerListener(stepDetector, mAccelerometerSensor, SensorManager.SENSOR_DELAY_GAME);
//mSensorManager.registerListener(throwDetector, mAccelerometerSensor, SensorManager.SENSOR_DELAY_GAME);
mSensorManager.registerListener(verticalThrowDetector, mAccelerometerSensor, SensorManager.SENSOR_DELAY_GAM
```

- What's fun
  - Testing our app is pretty fun, especially throwing phones in air.
  - Understand more on how android apps work.