Android Sensors and Communication: Part 2

Recap

- We saw a high-level overview of Android sensors.
- We saw the different categories of sensors and sensor data.
- We saw the basic workflow for sensor programming.
- We played with the light sensor and listed all the sensors on our device.

Today

- We’ll focus on vector-valued sensors.
- Specifically, we’ll talk about the position and motion sensors.

Sensors: Gyroscope

- The gyroscope sensor measures the rate of rotation of the phone along each axis.
- A very simple and elegant use for the gyroscope is to detect when the phone has moved.
- If the phone is still, the gyroscope returns a rate of rotation that is close to zero along all axes.
- If the phone is picked up or moved, the rates of rotation shoot up.
- We only care about the magnitude of the rates, and if that magnitude has gone up.
- Let’s see how to do this...

Sensors: Example 38 (Gyroscope)

- Remember, the only thing we have to do is override the onSensorChanged() method.
- We want to detect if the total rate of rotation exceeds some user-defined threshold.
- The gyroscope gives us a **vector** whose components are the rates of rotation.
- We only care about the magnitude of this vector to see how **much** the phone is rotating.
- Recall that the magnitude of a vector is the square root of the sum of squares of its components.

Sensors: Example 38 (Gyroscope)

- Let’s change the color of the app screen when the phone is rotated.
- This is straightforward.
- Interestingly, the gyroscope doesn’t detect translation, only detects rotation.
- In practice, only a robot would be able to move the phone in just one plane of motion without any small rotations.
- You can imagine more complicated uses for the gyroscope, such as a video game or a flight simulator app.
- In general, the gyroscope is far too sensitive for most apps, and sucks up a lot of battery.

Sensors: Accelerometer

- The accelerometer is ideal for gestures, and for measuring specific types of motion.
- As the name implies, the accelerometer measures acceleration (experienced by the sensor itself).
- Acceleration is a **vector**, which means that we get 3 values back from the sensor: one along each axis.
- There's one major issue with the accelerometer precisely because of how it works.
- It also measures acceleration due to gravity, which is always tugging downwards!

Sensors: Accelerometer

- If you want to ignore gravity, you'll have to subtract out its effects.
- The accelerometer uses far less battery than the gyroscope, but is also far less accurate.
- Let's see how to do a color change on phone translation using the accelerometer.
- We'll assume that rotation is ignored, but in practice, you'll have to detect and get rid of that yourself.

Sensors: Example 39 (Accelerometer)

- The linear accelerometer ignores gravity for us so we don’t have subtract it out explicitly.
- Let’s make it so that moving the phone a little bit changes the color, but moving it too much does not.
- Let’s also make it so that only moving in the y and z directions changes color.
- Pretty straightforward.
- Meta note: if you find this easy, it’s because you’re now comfortable with event-driven programming.

Sensors: Gesture Detection

- How would you build in gesture detection in your app?
- In practice, see if Android already has a way to do it, or look for a library.
- For learning, we'll implement our own shake detection.
- This time, the color will only change on a shake.
- Great! But... what is a shake, exactly?
- Think about it...

Sensors: Gesture Detection

- It’s a quick move to one side, or to the front/back.
- So, can we use that to build our gesture detector?
- Yes!
- Build on the previous example.
- Note that acceleration is the relative quantity here, not velocity.

Sensors: Example 40 (Shake Detection with the Accelerometer)

- We'll need to pick a threshold. A slow move shouldn't register.
- If we exceed the threshold on at least two axes, that should register as a shake.
- Note that you can train people to learn the right gestures by setting thresholds carefully; this is an open research question at the intersection of design and psychology.
- In practice, you also want to force some time interval between correct "shakes".
- You can use the system clock to achieve this effect.

Sensors: Step Counter and Detector

- With a lot of careful processing, it’s possible to write a good step counter or step detector using the accelerometer.
- Fortunately, Android gives us a way of doing this for free (since Android KitKat).
- Two synthetic sensors: the step counter and the step detector.
- The step detector has lower accuracy, but lower latency, and is a trigger sensor.
- The step counter is higher latency but higher accuracy.

Sensors: Example 41 (Step Counter)

- This is a fairly standard example.
- Your screen will display the number of steps taken since last (re)boot.
- To try this out, run the example on your phones, no easy way to emulate this (afaik).
- Let’s stop here.

Summary

- We learned how to work with the accelerometer, gyroscope, and step counter.
- We learned how to implement rudimentary gestures.