Sensors

Joe Rogers
Android 310

What are sensors?

- Hardware (and sometimes software), that provides special information to an app.
  - Pressure
  - Gyroscope
  - Accelerometer
  - Step counter
- Note: Must use a device for testing. Sensors are not available on the emulator.

Why use sensors

- Helps provide the app extra information that can not be determined by touch alone.
  - For example a game that allows the user to navigate a ball around the screen in a maze, might use the gyroscope to determine which direction the user is tilting the screen.
  - A fitness app may determine the user started jogging to automatically “track” the workout, in case the user forgot.

Why use sensors?

- A weather app may detect the user is driving and fetch current weather locations more frequently.
- An app may unlock a secret feature if the user shakes their phone on a particular screen.

Activity Detection

- Part of location apis in Google Play Services.
    - Google developed algorithms to help determine if the user is tilting the phone, walking, running, biking or riding in a vehicle.
    - Its a low cost way of gaining access to the sensor data without having to process the raw data.
    - More battery efficient than raw sensor data.

Activity Detection cont.

- In order to use this feature, the app must add the activity recognition permission
  - Required because indicating the user is driving, walking, etc is considered “personal” information especially when combined with location, etc.
- All results are sent via “pending” intents.
  - Advantage is easier for the app to be “asleep” and wait for a specific event to occur.

Google Fit

- New api in Google Play services which allows “fitness” applications to fetch data and share data.
- Allows access to body sensors on device and connected devices (ie wear, heart rate monitor, etc)

Google Fit

- Data is considered sensitive. User may be entering calories, height, weight etc.
- Therefore apps requesting this data will have to “authenticate” and ensure the user grants explicit permission to the app. User can revoke access without uninstalling the app.
- Not demoed in class.

Raw sensors

- Activity recognition and fitness is useful, but may not fit special cases required by the application.
- In those cases the raw device sensors may be a better choice.
    - Especially true if building a game or doing a custom gesture using motion.

Raw sensors

- Some sensors produce voluminous data. Therefore you should process it quickly and on a separate handler thread.
  - Especially if the data needs to be “transformed” in order to be consumed.
- Ensure you stop monitoring the sensor data when done as sensors will impact battery life.

Types of sensors

- Motion Sensors
  - These sensors measure acceleration forces and rotational forces along three axis. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.
- Position Sensors
  - These sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.

Types of sensors cont.

- Environmental Sensors
    - These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers

Considering a sensor

- Different devices use different hardware to provide sensor data. May need to adjust usage based on:
  - OS version (older OS may require estimation)
  - power usage
  - frequency of output
  - amount of “batched” operations supported.
  - Data delivery: continuous, on change, etc.

Gotchas

- Not every device has every sensor.
  - If critical may restrict app installations to only those devices via <uses-feature> in the manifest and making the sensor required. Ala a compass app without a magnetometer is not useful.
  - If optional, ideally hide the “UI” indicators supporting the feature and ensure <uses-feature> is marked optional.
  - Not all sensors have a corresponding feature.

Quick note on uses-feature

- Many permissions implicitly make features required.
  - For example CAMERA and LOCATION permissions.
  - If CAMERA is optional for your application, you still request the permission, but you add a “uses-feature” tag indicating the camera hardware is optional.
  - This does mean checking to see if the camera is available and disabling “camera” buttons if the feature is missing.

Uses-feature cont.

- Especially important for permissions for “Telephony” aka telephone access.
  - Easy to cause app to not be available on tablets which will not have telephony access.
- Best to verify in the play store what devices are supported and double check your “key” devices. If missing, may need to look for an implicit “uses-feature” via permission issue.

Uses-feature cont.

- Requiring a feature does not guarantee it is available. It only guarantees the device has the specific hardware.
- Should still “check” and handle when missing.
  - For example, device administration app, may disable cameras if the device is at a particular location.
  - Android M rumored to give permission control.

Resources

Resources

- Activity Recognition
- Google Fit
- Sensor Overview