Sensors

CE881: Mobile and Social Application Programming

Spyros Samothrakis

Febrary 08, 2015

Interesting Cultural Artefacts

Some interesting questions

Sensors

Discussion

Theme: “Sensors”

- Almost every sci-fi film ever made
- Ian Bank’s “Culture” series (books)

Sensors Apps

- AndroSensor
- Sensor Kinetics

IDE Tips (Again!)

- ctrl + left click
- Takes you to method/class/whatever definition
- Use it!!!

Question that have popped up in the course so far

- Java annotations
- Supported since Java 5
- Some annotations are used by the compiler or the IDE (e.g. “@Override”, “@Deprecated”)
  - You can remove them and the compiled code will do exaclty the same thing
- Runtime annotations
  - Change the behaviour of the code
  - e.g., create an annotation to retry
  - http://aspects.jcabi.com/

Example

```java
public class MyResource {
  @RetryOnFailure
  public String load(URL url) {
    return url.openConnection().getContent();
  }
}
```

Testing

- You don’t have to unit test
- You have to have a software testing schedule
- Even if manual
- State what you will test and how
- Unit tests should help catch errors

Model-View-Controller

- The default architecture
- You need to update the model!
- More on this later

Sensors

- Control Engineering
- What are sensors for?

Running on the device directly (1)

- Sensors don't make much sense in the emulator
- But you can debug directly in your device

Running on the device directly (1)

1. Enable developer mode on the device (device specific)
2. Connect your device to your computer's USB port
3. Setup your computer
   - Install Drivers (if on windows)
   - Run adb server as root / check lsusb for device in linux
4. run "adb devices"
5. Use the IDE to launch your app for the device

Android Sensor Categories

- Motion sensors
- Environmental sensors
- Position sensors
- All sensors types defined in android.hardware.Sensor

http://developer.android.com/guide/topics/sensors/sensors_overview.html

Motion Sensors

- TYPE_ACCELEROMETER
- TYPE_GYROSCOPE
- TYPE_ROTATION_VECTOR
- TYPE_GRAVITY
- TYPE_LINEAR_ACCELERATION

Environmental Sensors

- TYPE_AMBIENT_TEMPERATURE
- TYPE_LIGHT
- TYPE_MAGNETIC_FIELD
- TYPE_PRESSURE
- TYPE_RELATIVE_HUMIDITY
- TYPE_TEMPERATURE

Position Sensors

- TYPE_ORIENTATION
- TYPE_PROXIMITY

Finding available sensors

```
// global
private SensorManager sensorManager;
private Sensor accelerometer;
....
onCreate() {
    ...
    // within method
    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

    // Get all sensors
    List<Sensor> deviceSensors = mSensorManager.getSensorList(Sensor.TYPE_ALL);
    // Iterate over sensors, find sensors you like etc,

    // get the accelerometer
    accelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

}
```

Listening to sensor events

- Within an activity that implements *SensorEventListener*

```
@Override
 public final void onSensorChanged(SensorEvent event) {
   float[] acceleration = event.values;
   // do something with this, same as getting any other event
 }


@Override
 protected void onResume() {
   super.onResume();
   sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_NORMAL);
 }

 @Override
 protected void onPause() {
   super.onPause();
   accelerometer.unregisterListener(this);
 }
```

Handling multiple event types

- One could possibly do
  - “SenserEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER”
  - Use if/switch statements
- Or register multiple listeners
- Use-case specific
- Group similar events together

How/when to use sensors

- Sensors drain battery
- Some sensors drain more than other (e.g. Gyroscope vs Accelerometer)
- Not all devices have all kinds of sensors
- Device does not have a a type of sensor, **getDefaultSensor** returns null

Discussion

- Android devices sensors
- They can be used easily
- Debug on a real device
- Questions?