More Android
Logging

- Dump messages to the log; see with Android Monitor

- Log messages have:
  - Priority (ERROR, WARN, INFO, DEBUG, VERBOSE)
  - Tag
  - Message

- Usage: (use logd shortcut)

```java
private final static String TAG = 
    ClassName.class.getSimpleName();
Log.d(TAG, "functionName: your message here");
```
Testing

- Two kinds of tests for Android Projects:
  - Normal non-UI tests (‘test’ directory)
    - Just use Junit as normal
  - User Interface tests (‘androidTest’ directory)
    - Use Espresso
Espresso Test Recorder

- Run ‘Record Espresso Test’
  - Click the series of actions
  - Insert Asserts as appropriate
    - Only a few Assert options currently available
    - Can later go back and extend/modify the code

- Run Tests (What I could get to work…)
  - Use Gradle Menu
  - Tasks -> verification -> connectedAndroidTest
Monkey Testing

- Random user interface testing
  - Mostly to check for crashes

- Implemented using the “Android Debug Bridge” (ADB)
  - Command line tool
  - Mine is in ~/Library/Android/sdk/platform-tools

- Run with:
  - `adb shell monkey -p <package.name.app> -v <num actions>`
  - `adb shell monkey -p ninja.zilles.hangman -v 5000`
  - Or ./adb shell .... (if not in your path on Mac)
Constraint Layout

- Beta Feature of Android Studio / Support Library
- Goal: Reduce hierarchy in Layouts
Using ConstraintLayout

- In app/build.gradle:

```groovy
dependencies {
    ...
    compile 'com.android.support.constraint:constraint-layout:1.0.0-beta1'
}
```

- In layout.xml file:

```xml
<android.support.constraint.ConstraintLayout ...
```
To Do for Thursday

- Watch Lesson 2 of the Google/Udacity Android Class
  - “Connect Sunshine to the Cloud”
Android Permission Model

- Users don’t want apps to violate their privacy
- Users grant apps permission to do things
  - Access the network, camera, calendar, phone book, etc.
  - Historically, these have been granted at install time
  - All or nothing model

- Starting in Marshmallow, incremental permission model
  - Request “mandatory” permissions at install time
  - Request other permissions as needed (for clarity)
  - Allow users to revoke permissions
    - App must check permissions before doing controlled things.
Threads

■ When you write code, you tell the machine what to do
  ▪ One thing at a time.
■ Hard/bad to interleave multiple things
  ▪ E.g., a user interface with long latency tasks
Threads, cont.

- Computer programs are made up of threads
- Each thread:
  - Performs a series of tasks
  - In the order specified by the code
  - One at a time
- **Hard/bad to interleave multiple things on a single thread**
  - E.g., a user interface with long latency tasks
- **Solution: use multiple threads; dedicate a thread to the UI**
  - So it is always responsive
  - Do slow stuff on other threads
  - Have to handle communication/synch between threads