Mobile Application Development
Aileen Pierce

ACTIVITIES AND INTENTS

Model View Controller

- Android also follows the model view controller (MVC) architecture
- Model: holds the data and classes
- View: all items for the user interface
- Controller: links the model and the view together. The backbone or brain of the app.
- These categories should never overlap.
- The goal of MVC is to have any object be in only one of these categories.

Activities

- An activity is a single, specific task a user can do
- Each activity has its own window for its view The window typically fills the screen, but may be smaller than the screen and float on top of other windows
- An app can have as many activities as needed
- Each activity is listed in the AndroidManifest.xml file

Intents

- An intent tells the app you're about to do something such as start a new activity
  - An explicit intent tells the app to start a specific activity
  - An implicit intent tells the app to start any activity that can handle the action specified

```
Intent intent = new Intent(this, Target.class);
startActivity(intent);
```

Passing Data

- You can add extra information to your intent to pass data to the new activity

```
intent.putExtra("message", value);
```

  - The `putExtra()` method is overloaded so you can pass many possible types
  - Call `putExtra()` as many times as needed for the data you're passing

Receiving Data

- When a new activity starts it needs to receive any data passed to it in the intent
- Get access to the intent

```
Intent intent = getIntent();
```

- Retrieve the data sent in the intent

```
String msg =
intent.getStringExtra("message");
```

- You aren't limited to strings, there are many other getxxxExtra methods

Intents

- An activity can also pass an intent to Android as an implicit intent
- Implicit intents tell Android what type of action you want to perform

```
Intent intent = new Intent(action);
```

`Intent.ACTION_DIAL` to dial a number

`Intent.ACTION_WEB_SEARCH` to perform a web search

`Intent.ACTION_SEND` to send a message

and many others

Intents

- Android checks to see what activities can handle the intent
- If just one activity can handle the intent, that activity is chosen
- If there is more than one activity that can handle the intent, the user is prompted to chose
- Android tells the activity to start even though it's in another app and passes it the intent

Intents

- Android uses intent resolution to figure out which component are able to receive an action
- An intent filter specifies what types of intents each component can receive
- The intent filter also specifies a category
- Each activity has intent filters defined in the AndroidManifest.xml file

Intents

- An intent filter must include a category of **`android.intent.category.DEFAULT`** if it's to receive implicit intents
- If an activity has no intent filter, or it doesn't include a category name of **`android.intent.category.DEFAULT`**, it means that the activity can't be started with an implicit intent. It can only be started with an *explicit* intent using the fully qualified component name.

Intents

- When you use an implicit intent, Android compares the information given in the intent with the information given in the intent filters specified in the AndroidManifest.xml file
  1. Android looks for intent filters with the category **`android.intent.category.DEFAULT`**
  2. Android then looks for intent filters with the action and mime type that match the intent
- Those intents are then presented to the user