UNIVERSITY of WASHINGTON

Android 210 - Lecture 3

Data Input, Dialogs & ListView

Margaret Maynard-Reid
January 26, 2015

Agenda

- Homework 1 Solution
- UI Basics Part 2
  - ScrollView
  - EditText, Checkbox, Toggle/Switch
  - Toasts, Popupwindows & Dialogs
  - Adapters, Spinner & ListView
- Git Version Control
- Homework 2 requirements

Android Stories

- Cyanogen is heading to create "Android without Google"
- 10 Material Design Android apps you should be using right now

Review Homework 1 Solution

Walk through code

Review of last week

- When do we use a LinearLayout?
- When do we use a RelativeLayout?
- When do we use a FrameLayout?
- What is an ImageButton?

UI Input

- EditText: take user input data
- SeekBar: a ProgressBar with a dragable thumb
- Spinner: select from a dropdown
- Compound Buttons: choices
    - Switch
    - Check Box
    - Radio Button
    - Toggle or Switch Button

EditText - hint

EditText hint is shown in the textbox before editing the field.
It is a way to replace labels for Edit Fields.

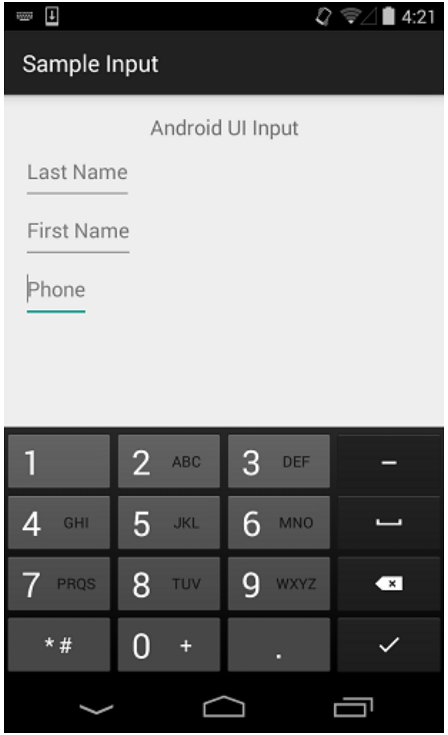
```
<EditText
android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Name:"
/>
```

EditText - inputType

- Combinatorial value of an editable object that controls the type of input expected
- Set in XML via inputType with constants like:
  - "text"
  - "phone"
  - "textEmailAddress"
- Set in Code via EditText.setInputType() with constants like:
  - TYPE_CLASS_TEXT
  - TYPE_CLASS_PHONE
  - TYPE_TEXT_VARIATION_EMAIL_ADDRESS

EditText - Input Type in XML

```
<EditText
        android:id="@+id/editTextPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_margin="10dp"
        android:hint="Enter your Phone Number"
        android:inputType="phone">
 </EditText>
```

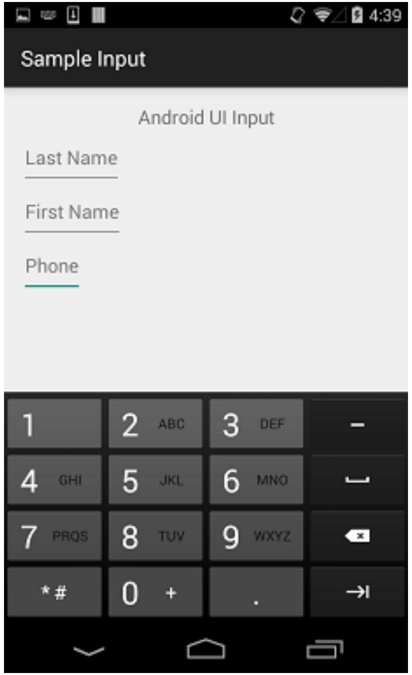


EditText - windowSoftInputMode

- Combinatorial value of an activity that controls display of the soft keyboard
  - stateUnspecified
  - stateUnchanged
  - stateHidden
  - stateAlwaysHidden
  - stateVisible
  - stateAlwaysVisible
  - adjustUnspecified
  - adjustResize
  - adjustPan
- Set on the Activity in AndroidManifest.xml

EditText - windowSoftInputMode

stateVisible



4:39

Sample Input

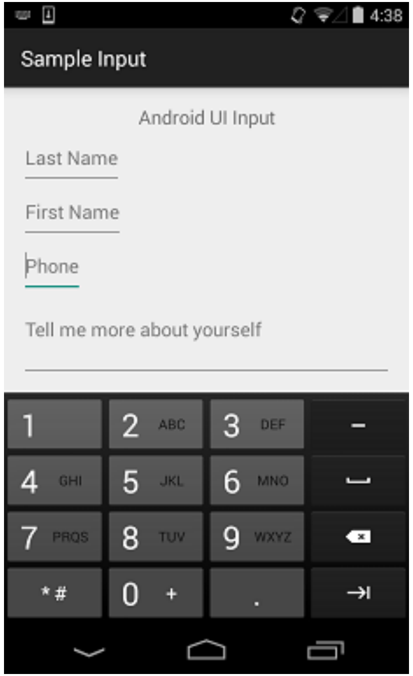
Android UI Input

Last Name

First Name

Phone

stateVisible|adjustResize



4:38

Sample Input

Android UI Input

Last Name

First Name

Phone

Tell me more about yourself

EditText - multiline

Make the Edit field multi-line:

android:singleLine="false"

android:lines="2"

or use this:

android:inputType="textMultiLine"

EditText - Best Practices

- Be mindful of input size – especially for localizability
- Always set the inputType and windowSoftInputMode explicitly

EditText - get input data

- Create EditText in xml:

  EditText editText = (EditText)findViewById(R.id.editText1);

- Get user input

  editText.getText().toString();

- On TextChanged...
- Validate EditText, show UI cue where there is error
- Handle saving data when screen orientation changes

EditText - save state

- Activity is destroyed when screen rotates
- Use **onSaveIntanceState()** to save data. ← called between onPause & onStop()
- Data is restored it in either
  - **onCreate()** or
  - **onRestoreInstanceState()** ← called after onStart() so no need to check if bundle is null
- Android automatically saves UI states as long as a valid id has been assigned to it
- http://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)

ScrollView

- To add scrolling wrap the whole layout into a scrollview

```
<ScrollView
      xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="match_parent"
     android:layout_height="match_parent" >
...
</ScrollView>
```

- Now you can scroll down to see the rest of the content.

Passing data to another activity

1. Sending Activity - put the data in the intent

```
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
intent.putExtra("firstname", mEditTextFirstMame.getText().toString());
startActivity(intent);
```

2. Receiving Activity - retrieve the data from intent

```
Bundle bundle = getIntent().getExtras();
String firstName = bundle.getString("firstname");
```

Hands-on Coding

Walk through **SampleInput.java**

*** Note: AVD modify emulator to enable keyboard if you want to type in EditText when using an emulator.

Break

Homework 2 requirements

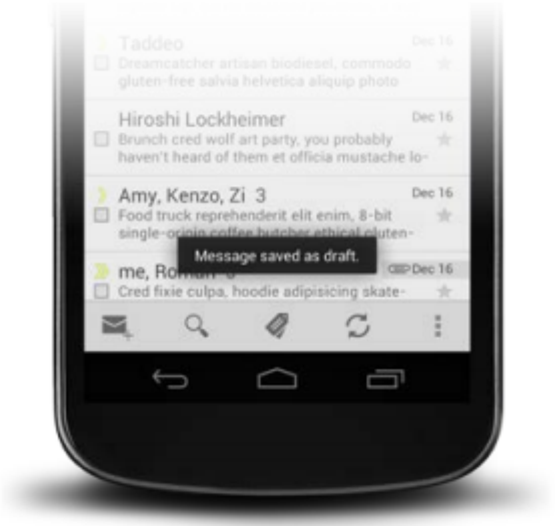
- Data input
- Preserve data on screen rotation
- Pass data from screen 1 to 2

Input

Toasts, PopupWindows, and Dialogs

Toasts

- Simple visual notifications across the bottom of the screen that capture no user input



Taddeo Dec 16
Dreamcatcher artisan biodiesel, commodo gluten-free salvia helvetica aliquip photo

Hiroshi Lockheimer Dec 16
Brunch cred wolf art party, you probably haven't heard of them et officia mustache lo-

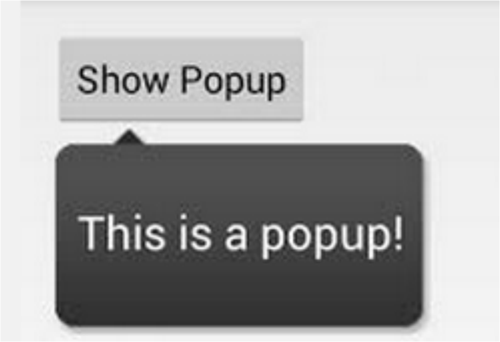
Amy, Kenzo, Zi 3 Dec 16
Food truck reprehenderit elit enim, 8-bit single-origin coffee butcher ethical gluten-

me, Ro Dec 16
Cred fixie culpa, hoodie adipisicing skate-

Message saved as draft.

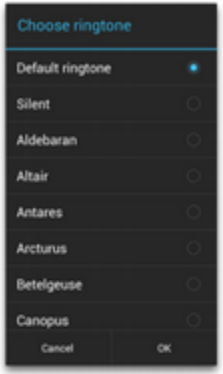
PopupWindows

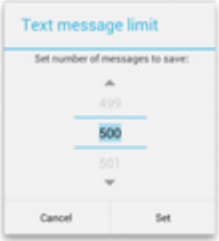
- A floating window over the top of an Activity
- Used to display additional info

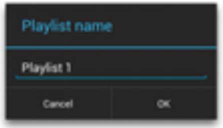


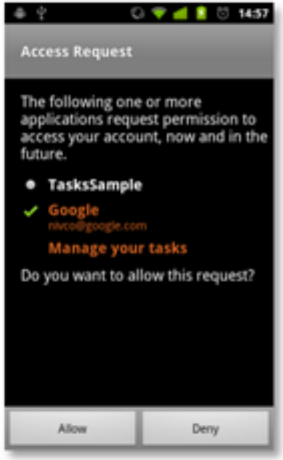
Dialogs

- A window over the top of an Activity that usually gathers input from or alerts a user









The following one or more applications request permission to access your account, now and in the future.

- TasksSample
- ✓ Google
  nivco@google.com
  Manage your tasks

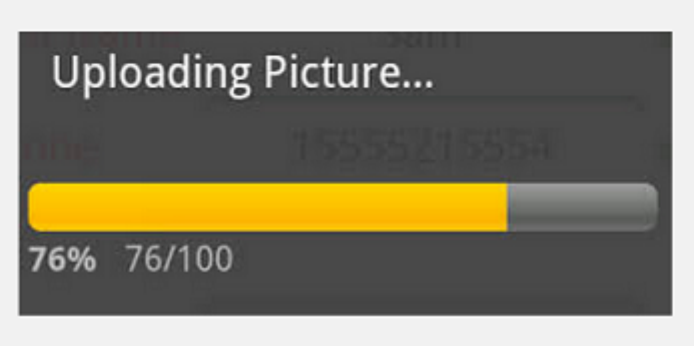
Do you want to allow this request?

Dialog - how to create it

1. Use AlertDialog.Builder to set dialog properties:
   - Title
   - Message
   - Buttons & button clicks
2. Create the dialog
3. Show the dialog

ProgressBar

- Keep users informed of the progress during a long running operation (i. e. uploading an image)
- Will go over this when discussing threading and AsyncTask



Uploading Picture...

76% 76/100

Hands-on Coding

Walk through **SamplePopup.java**

Break

Data-Adapter-View

The Data-Adapter-View Pattern

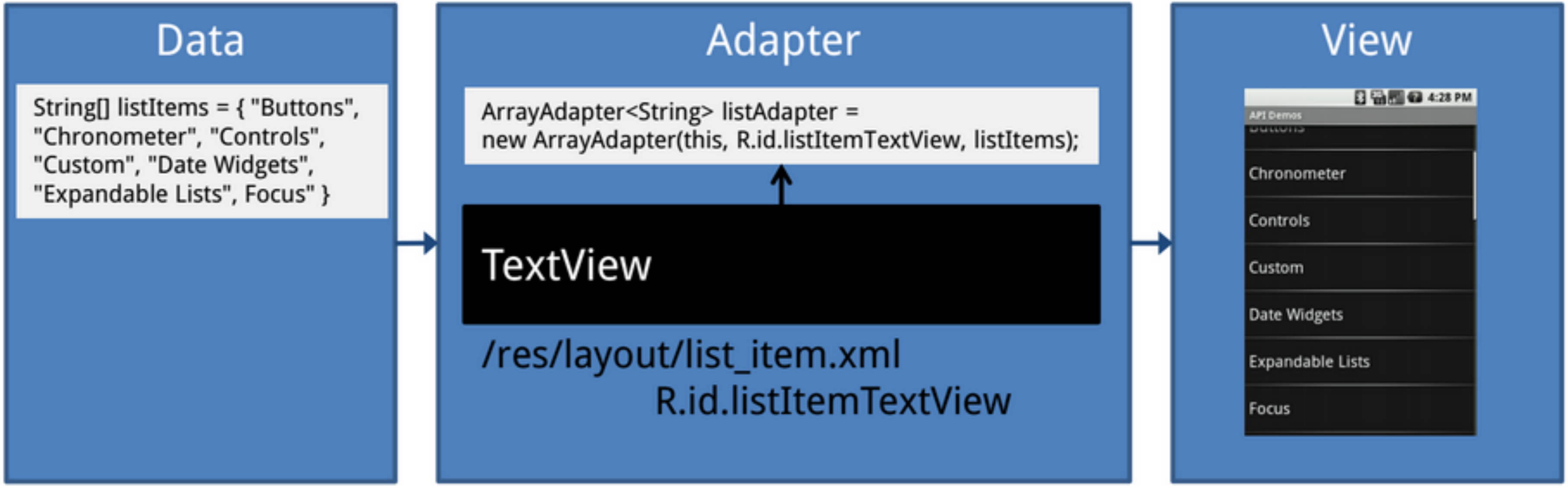
- Very similar to Model-View-Controller (MVC)
- A bridge between UI and data
- Used to fill content in Lists & Spinners, etc.

Adapter Binding

The process of associating your data or content to an AdapterView

- The Adapter is the mapping between an AdapterView and the Data that needs to be populated
- The Adapter provides access to the data items and describes the View needed to display each data item

Data-Adapter-View



Data

```
String[] listItems = { "Buttons",
"Chronometer", "Controls",
"Custom", "Date Widgets",
"Expandable Lists", Focus" }
```

Adapter

```
ArrayAdapter<String> listAdapter =
new ArrayAdapter(this, R.id.listItemTextView, listItems);
```

TextView

/res/layout/list_item.xml

R.id.listItemTextView

View

Types of Adapters

- ListAdapter - Generic data wrapper for a ListView
- ArrayAdapter - Wrapper for an array of arbitrary objects
- CursorAdapter - Wrapper for data from a Cursor that must include "_id" column
- SimpleAdapter - Wrapper for static XML data
- SimpleCursorAdapter - Wrapper for mapping columns from a Cursor
- SpinnerAdapter - Wrapper for Spinner data

Changes to Data

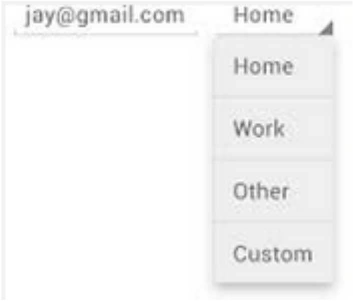
If you are making changes to the data underneath the view, you will need to call as appropriate:

- BaseAdapter.notifyDataSetChanged()
  - Notifies observers that the data has been changed
- BaseAdapter.notifyDataSetInvalidated()
  - Notifies observers that the data is no longer valid or available

Spinner

A drop-down widget View (pick-list)

- Backed by an Adapter



ListView

A View that vertically scrolls a list of items

– Backed by an Adapter

– You can find 3rd party implementations of horizontal ListViews on github, etc.

Hands-on Coding

Walk through **SampleList.java**

Git Version Control

GIT - Distributed Source Control

- Download it from here: http://git-scm.com/

Android Studio already has built in VCS integration

Common Commands:

- git status, git log
- git commit, git branch, git checkout, git merge
- git pull, git push

Git - create a new repo

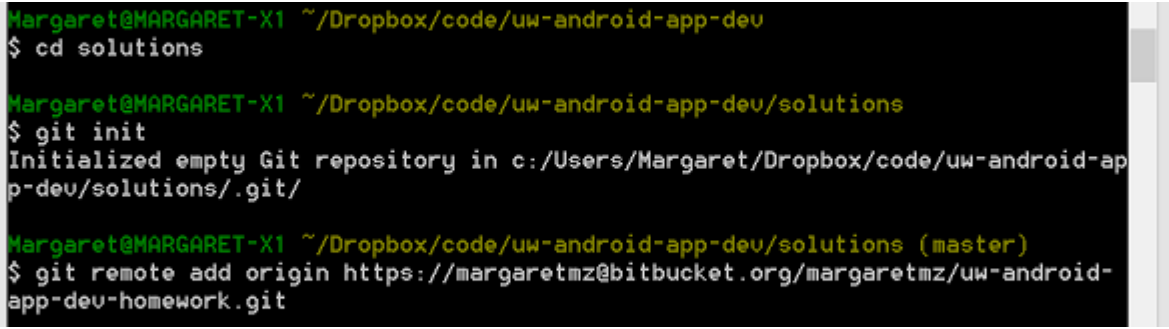
First create a new repository on Bitbucket or gitHub

- Bitbucket - free unlimited private repo, limited to 5 users
- gitHub - free unlimited public repo

Git - add remote

GitBash in Windows, Command Line in Mac

- *mkdir /path/to/your/project* (optional step)
- *cd /path/to/your/project, or a folder that contains multiple projects*
- *git init* ← initializes git repository for the folder
- *git remote add origin https://margaretmz@bitbucket.org/margaretmz/newrepo.git*

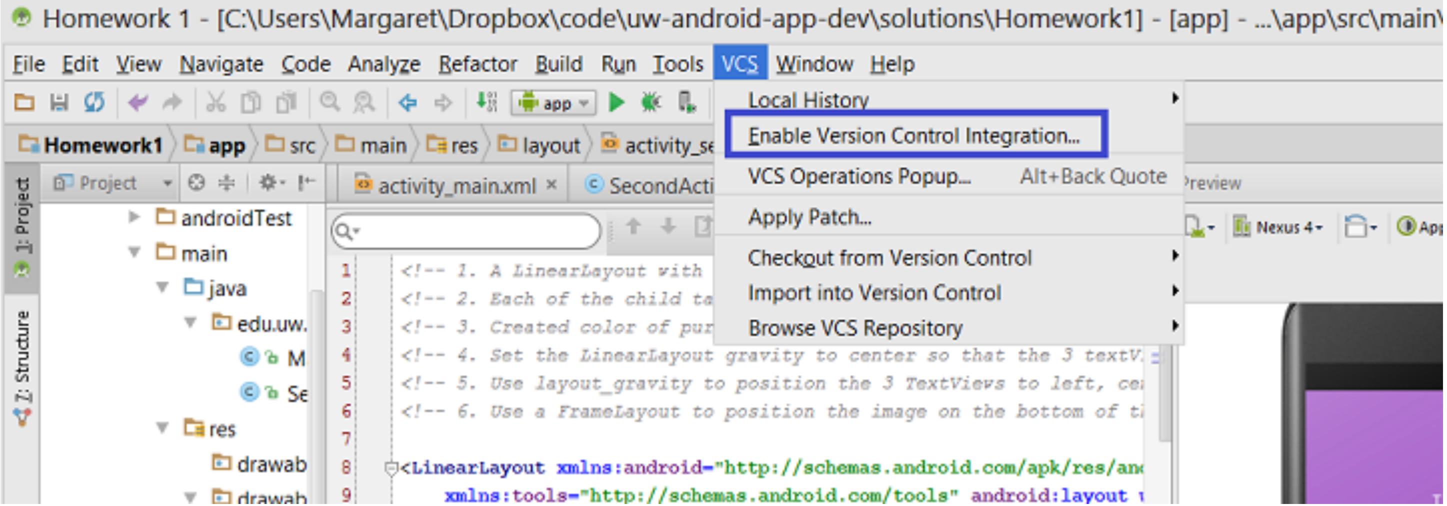


```
Margaret@MARGARET-X1 ~/Dropbox/code/uw-android-app-dev
$ cd solutions

Margaret@MARGARET-X1 ~/Dropbox/code/uw-android-app-dev/solutions
$ git init
Initialized empty Git repository in c:/Users/Margaret/Dropbox/code/uw-android-ap
p-dev/solutions/.git/

Margaret@MARGARET-X1 ~/Dropbox/code/uw-android-app-dev/solutions (master)
$ git remote add origin https://margaretmz@bitbucket.org/margaretmz/uw-android-
app-dev-homework.git
```

Git - VC integration in AS

In Android Studio:

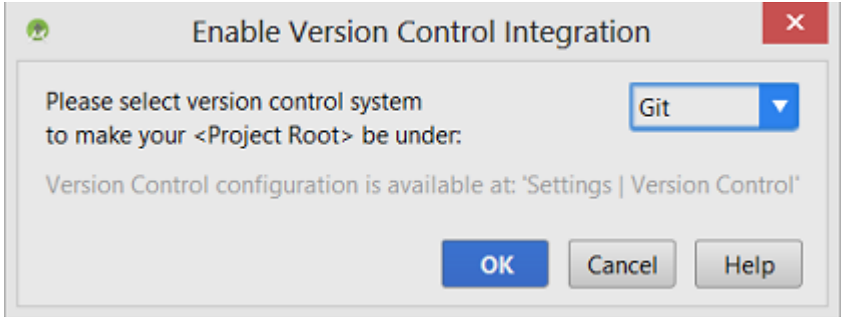
- VCS/Enable Version Control Integration


Homework 1 - [C:\Users\Margaret\Dropbox\code\uw-android-app-dev\solutions\Homework1] - [app] - ...\app\src\main\
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
Homework1 app src main res layout activity_se
Project
1: Project
7: Structure
activity_main.xml
SecondActi
androidTest
main
java
edu.uw.
M
Se
res
drawab
drawab
Local History
Enable Version Control Integration...
VCS Operations Popup... Alt+Back Quote
Apply Patch...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Preview
Nexus 4
App

```
<!-- 1. A LinearLayout with
<!-- 2. Each of the child te
<!-- 3. Created color of pur
<!-- 4. Set the LinearLayout gravity to center so that the 3 textV
<!-- 5. Use layout_gravity to position the 3 TextViews to left, ce
<!-- 6. Use a FrameLayout to position the image on the bottom of t

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/an
    xmlns:tools="http://schemas.android.com/tools" android:layout
```

Git - VC integration in AS

- Select Git



Enable Version Control Integration

Please select version control system to make your <Project Root> be under: Git

Version Control configuration is available at: 'Settings | Version Control'

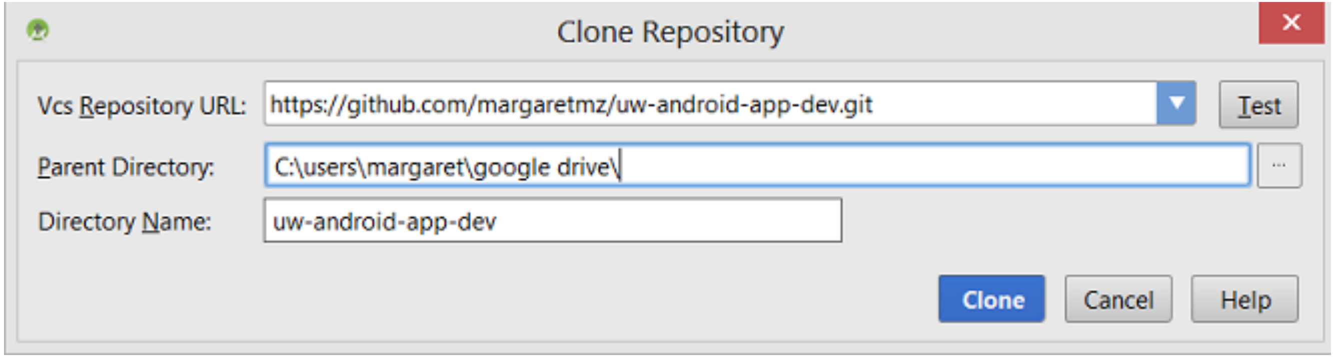
Git - commit changes

- Right click project,
  Git/Add(**Ctrl+Alt+A**) ← add the project to git
- Modify **.gitignore** if needed
- Alt+9 to see changes
- Select files to commit
- Push ← make sure to select the branch

Git - clone a repository

In Android Studio,

Git/Checkout from Version Control/Git



Clone Repository

Vcs Repository URL: https://github.com/margaretmz/uw-android-app-dev.git

Parent Directory: C:\users\margaret\google drive\

Directory Name: uw-android-app-dev

Upcoming

- Feb 2: ActionBar & Fragment
- Feb 9: Storage
- Feb 9: Homework 2 due
- Homework 1 should be graded within 1 week