

Introduction to Android Studio and Version Control ¹

1 Check Out an Android Tablet

A technician should be here at the beginning of lab. She/he will check out an Android tablet to you, which you will use during the entire semester. Treat your Android tablet kindly, and do not lose any parts.

2 Getting Started With Building and Running Projects Under Android Studio

Checkpoint 1 (10 points): Create a project from scratch and run it on your tablet. The instructor will step you through this process as part of the intro lecture.

3 Git Version Control: Commit and Push Projects

We will be using the *Git* version control system (VCS). Using Git will allow you to make a local "snapshot" of your project, and then to revert back to previous snapshots. Local means it is saved on the computer you are working on. (For example if you delete some code that you later realize that you need, you can recover it.)

- 1) Tell Android Studio that you will be using Git, and have it create a local Git repository:
 - a) On the menu select VCS -> Enable Version Control Integration ...
 - b) select 'Git' from the drop-down menu
 - c) Press the 'OK' button
- 2) Tell Git what source and configuration files you want to keep track of.
 - a) Right-click on the project name (e.g., Lab01, or whatever you named it), then choose the menu-item. The key is that you right click on the program name (for example, AnotherHello in the screen shot below) to assure that all the relevant project files are added. **Fair warning:** If you fail to do this correctly, you may have to start over on this lab.
 - b) Git -> +Add (You can also find Git through the VCS menu (VCS -> Git -> +Add))

c) If you are asked to commit vcs.xml, say No.

3) Commit those files into the repository

a) In the VCS menu, select Commit. A Commit Changes dialog will appear.

b) Commit message: "Initial commit."

c) Press the 'Commit' button.

d) If any warnings occur, a dialog will appear. Just press 'Commit' on these as well.

4) You should get a message to the effect that the *commit* was successful, and that it committed some number of file. You have now taken a "snapshot" of the current software. You can go back to this version at any time.

5) Verify that your commit was successful:

a) Right click on your project name as you did in step 2b above

b) Select Git -> Show History This should bring up a Version Control tab at the bottom of your Android Studio window. The history should contain only one item: your "Initial Commit".

c) Double click on the commit to see a dialog with a list of all the files you committed.

Checkpoint 2 (10 points): Have your instructor or lab assistant verify that you have done a commit.

Checkpoint 3 (10 points): Change the text property in the layout file to include your name in the hello message with text size 60dp. "Hello <your name>." Commit your changes and have your instructor or lab assistant verify the change.

Now that you've made a change, you will practice reverting to your original snapshot version:

1) Right click on the name of your project and select *LocalHistory -> Show History*. This should bring up a window that shows the differences between the current state of the project and your last commit. Specifically, activity_main.xml has changed.

- 2) Right-click on `activity_main.xml` in the window and select Show Differences. A window should appear that shows you specifically what changes you have made. Verify those changes are consistent with your expectations. Then close this window.
- 3) In the left pane of the current window is a list of changes. This should include all the files in your initial commit. At the very top of this list your more recent change to `activity_main.xml` should be listed. Below is an oval that is labeled something like "Commit Changes: Initial commit". Right click on this and select "Revert".
- 4) Close the window and rerun the app on your tablet. It should have the original behavior (Hello, World!).
- 5) In a similar manner, go back to the newer version.
- 6) Rerun the app on your tablet. It should again have the new behavior.

Checkpoint 4 (10 points) Demonstrate to the instructor or lab assistant that you can move between versions.

So far, your repository is on your local P:-drive, but you'd really like it in a location that you can easily access it from anywhere in the world. You'll use the repository service provided by GitHub.com. Unless it already exists, please visit the website and create a new account *with your UP email address*.

Once that is done, the software for our application will be in three places:

- The current version will be in the Android Studio project on your P:-drive.
- All *committed* versions will be in the local Git repository on your P-drive.
- All *pushed* versions will be accessible on the GitHub site.
 - Take care, because by default GitHub projects are public, so that anyone with an internet connection can download them. (You will later be instructed on how to use private repositories.)

To push your repository to GitHub:

- 1) Select VCS -> Import into Version Control -> Share Project on GitHub
- 2) Type in your GitHub user id and password. A new dialog will appear asking for information about the new repository.
 - a) Repository Name: pick something appropriate. Do not use the default
 - b) Do not make it a private repository. (You may do this later.)
 - c) Remote: origin

d) Description: optional

Press the Share button when done.

3) A dialog will appear asking you what files to commit. Select OK.

a) If you are asked to commit vcs.xml, say No.

Return to your web browser and locate the repository that you just created on GitHub.

Checkpoint 5 (10 points): Demonstrate to the instructor or lab assistant that the project has been pushed and that you can navigate through the online file system to find your resource files.

4 Git Version Control System of the Birthday Cake App

In this section you will check out an existing project (BirthdayCake) from GitHub.

This download process gives you a copy of a project from a repository using this process you get your own version. There is no connection between the modifications you make to this copy and the original.

1) Create a folder named cs371 somewhere on your P: drive.

2) Find and clone the starter-project repository by:

a) Go to the project on github.com

`https://github.com/cs301up/BirthdayCake`

b) Download the zip-file for the project by clicking the “Clone or Download” button on the right hand side of the page and then selecting Download ZIP.

c) Unzip this starter file into a subfolder of your c371 folder on your P-drive.

3) Open the downloaded project in Android Studio

a) Close open Android Studio projects (File -> Close Project). This should bring you back to the “Welcome to Android Studio” screen.

b) Open in Android Studio the project that was downloaded and unzipped in “Preliminaries.

c) Press “Open an existing Android Studio project”

- 4) Browse to the project you just unzipped (P:\cs371\...); open the project.
 - a) **IMPORTANT:** Select the file that has the icon matching the one for Android Studio
- 5) If it asks you “Would you like to remove VCS root ...”, say ‘Yes’.

Note: You may see an error that says “Invalid VCS root mapping.” This is normal as Android Studio is expressing confusion that the project came from Git but is not checked in anywhere. This should go away once you check your project in.

Once you have the app downloaded, verify that you can run it on your tablet. Then, create a new repository of your own for this project on GitHub.

Checkpoint 6 (10 points): Have your instructor or lab assistant verify that the BirthdayCake app is running on your tablet and that you have pushed it to your repository.

Now you want to modify the app. In the project explorer tree (left hand side) locate the file named activity_main.xml. This file describes the user interface of the app and shows you a preview of what it looks like.

The application currently has a TextView at the top that says “Welcome to CS371”. Modify this TextView to say “Happy Birthday!” instead. Then make the font much larger and a pumpkin orange color.

Checkpoint 7 (10 points): Have your instructor or lab assistant verify that you have successfully made the changes.

Now, you want to check this revised version into GitHub. Follow these steps:

- 1) Commit the project to your local repository
 - a) On the menu, select VCS → Commit... A “Commit Changes” dialog will appear. The only file that should have changed is activity_main.xml.
 - b) Enter a commit message describing what you did
 - c) Hit the Commit button. You may be asked to review warnings as Android has (rightly) identified things that could be improved with the layout. For now, it is safe to ignore these and confirm the commit.
- 2) Now push your changes to GitHub
 - a) On the menu, select VCS → Git → Push... A “Push Commits” dialog will appear.

b) Press the Push button

In a web browser, verify that the change has appeared on GitHub and view the modifications to the `activity_main.xml` file there.

Checkpoint 8 (10 points): Show your instructor or lab assistant the modified app running on your tablet and also show the changes on GitHub.

The app currently has two buttons at the bottom. Add a Switch control between the two buttons. The label on this switch should be labeled “Frosting”.

Commit your changes and push them to your project on GitHub.

Checkpoint 9 (10 points): Show your instructor or lab assistant the modified app running on your tablet and also show the changes on GitHub.

Locate the file in your project that is named `CakeView.java`. It contains the `CakeView` class. The code there is what draws the cake on the screen.

Modify this Java code so that there is a second candle on the cake.

Commit your changes and push them to your project on GitHub.

Checkpoint 10 (10 points): Show your instructor or lab assistant the modified app running on your tablet and also show the changes on GitHub.

Congratulations. You have completed your first CS371 lab.