Introduction to Version Control and Making an App

# Setting Up Developer Mode (Done in Lab 1a)

# Getting Started with Building and Running Projects in Android Studio (Done in Lab 1a)

# 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 you need, you can recover it from the Git repository.)

1. Tell Android Studio that you will be using Git, and have it create a local Git repository:
   1. On the menu select VCS -> Enable Version Control Integration …
   2. select ‘Git’ from the drop-down menu
   3. Press the ‘OK’ button
2. Tell Git what source and configuration files you want to keep track of.
   1. The key is that you right click on the program name 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. In order to see the project, change the dropdown above the listed files from Android to Project.
   2. Right-click on the project name (e.g., Lab01 or HelloWorld or whatever you named it), and choose Git -> +Add. (You can also find Git through the VCS menu (VCS -> Git -> +Add).)
   3. If you are asked to commit vcs.xml, choose No.
3. Commit those files into the repository
   1. In the VCS menu, select Commit. A Commit Changes dialog will appear.
   2. If you scroll to the bottom of the files list in the upper window, you should see just 1 Unversioned File, which is vcs.xml.
   3. Enter the commit message: "Initial commit".
   4. Press the 'Commit' button.
   5. 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 files. 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:
   1. Right click on your project name as you did in step 2b above
   2. 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”.
   3. Double click on the commit to see a dialog on the right with a list of all the files you committed (that dialog might already be showing).

**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>.” Run your program to make sure the changes took effect. Commit your changes with a descriptive commit message and have your instructor or lab assistant verify the commit and the running app.

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 right pane and select Show Difference. 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. It should have the original behavior (Hello World!).
5. Now go back to the newer version by right-clicking on the line that says “Reverted to…” Make sure that you look at the differences in the activity.xml before you revert to ensure you’ve selected the correct version. On the left is the Base Version, meaning the version that you have selected to revert to. On the right is your current, existing version. The Base version should say “Hello <your name>!” and Your Version should say “Hello World!” Once you’ve verified you have the correct version selected, right-click and select Revert. You should see “Reverted to ‘Reverted to…’”.
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 (or your computer’s C: 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 you already have an account (or want to create a new one), please visit the website and create a new account *with your UP email address*.

Once we post the project to GitHub, the software for our application will be in three places:

* The current version will be in the Android Studio project on your P: drive (or C: drive).
* All *committed* versions will be in the local Git repository on your P: drive (or C: 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. If this doesn’t work, follow the steps below.

Log in via token:

Log in to github.com on the browser

Click on your profile icon in the top right corner and select Settings. Then Developer Settings > Personal access tokens.

Click Generate new token.

Give it a name. Check the boxes for ***repo***, ***admin:org***, and ***gist***. Click the Generate token button at the bottom.

Copy the value of the token.

Go back to Android Studio. Select the option in the dialog to use token. Paste in your token.

Once you’ve successfully logged in:

* 1. Repository Name: Pick something appropriate (the default may be ok if it is sufficiently descriptive). There cannot be spaces in the name. Use underscores instead.
  2. Do not make it a private repository. (You may do this later.)
  3. Remote: origin
  4. Description: optional
  5. Press the Share button when done.

1. A dialog may appear asking you what files to commit. Select OK.
   1. 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. You should be able to find it under Your repositories under your profile menu.

**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 (in other words, a folder called “res”).

# Git Version Control System of the Birthday Cake App

In this section you will check out an existing project (BirthdayCake) from GitHub. **It is important that you complete all the following checkpoints for this app because the next two labs build on the work you do here.**

This download process gives you a copy of a project from a repository. 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: (or C:) drive. Do not use your Desktop or Downloads folders; choose a better location.
2. Find and clone the starter-project repository by:
   1. Go to the project on github.com

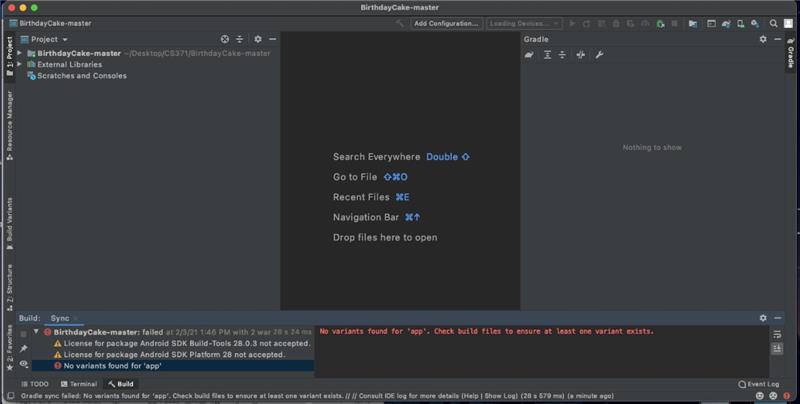
https://github.com/cs301up/BirthdayCake

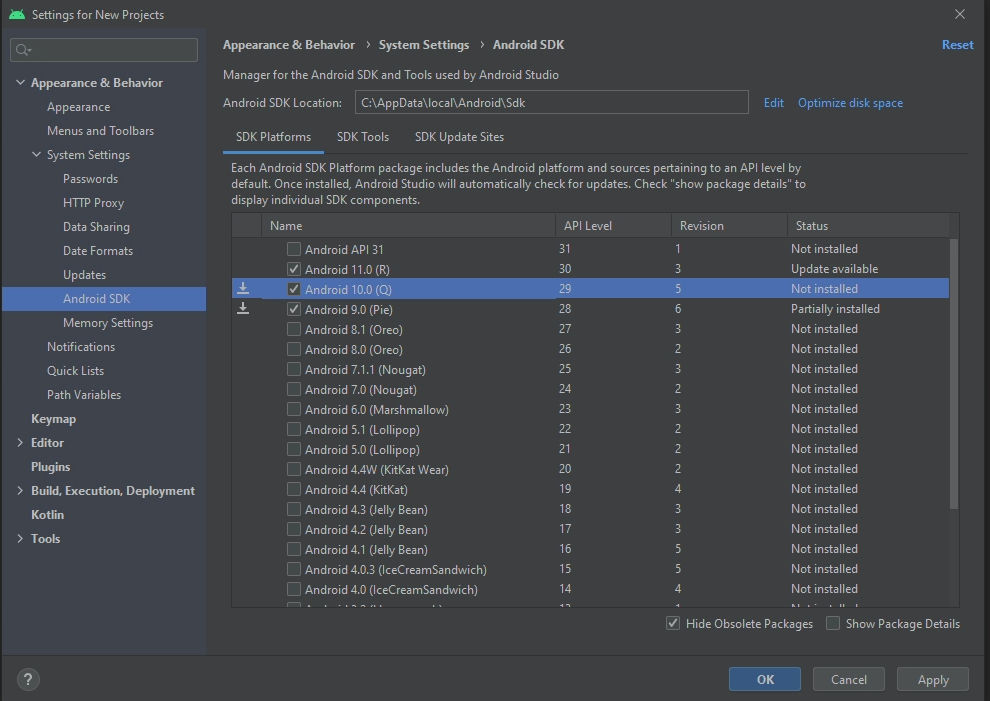
* 1. Download the zip file for the project by clicking the green “Code” button on the right-hand side of the page and then selecting Download ZIP.
  2. Extract this starter file into a subfolder of your c371 folder on your drive.

1. Open the downloaded project in Android Studio
   1. Close open Android Studio projects (File -> Close Project). This should bring you back to the “Welcome to Android Studio” screen.
   2. Press “Open an existing Android Studio project”.
   3. Browse to the project you just unzipped. IMPORTANT: Select the topmost file called BirthdayCake-master that has the icon matching the one for Android Studio.
   4. 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.

When you open up the birthday cake app in Android Studio, you may get an error that looks like the one in the screenshot below. This means you need to download another version of the SDK from the SDK manager, which is this icon  in the top right corner of the Android Studio menu bar. Check the box for the API Level that is mentioned in your error message. I've seen it be 28 for some and 29 for others (I have no idea why it is inconsistent). When you hit OK, Android Studio will ask you to confirm the installation of that SDK.

[](https://us-prod.asyncgw.teams.microsoft.com/v1/objects/0-wus-d9-a096c362c9f534ec8b9f18b56f8ad317/views/imgo)



Once the app is properly loaded, verify that you can run it. 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 make it orange in color.

These changes may not be immediately visible in the Design view of activity\_main.xml. You can apply the changes by selecting Run > Apply Changes and Restart Activity. Alternatively, if you Run the program (with or without selecting Apply Changes first), you should see the updates on your tablet (or the simulator) as well as in the Design view.

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

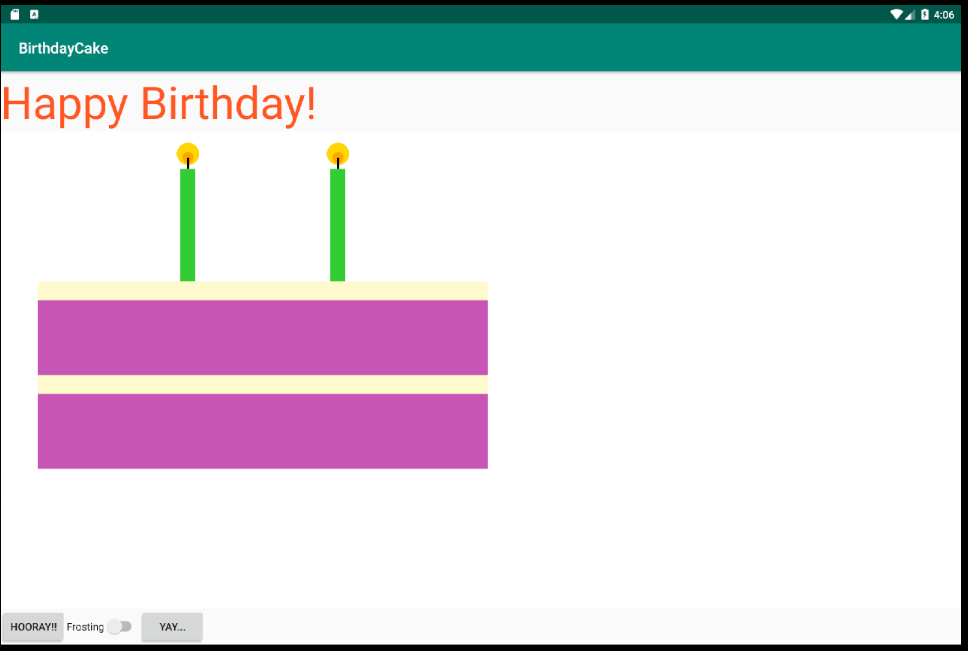
1. Commit the project to your local repository
   1. On the menu, select VCS → Commit… A “Commit Changes” dialog will appear. The only file that should have changed is activity\_main.xml. (It’s possible that files called misc.xml and jarRepositories.xml changed too. It’s ok to keep them selected.)
   2. Enter a commit message describing what you did
   3. 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
   1. On the menu, select VCS → Git → Push… A “Push Commits” dialog will appear.
   2. 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 7 (20 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 (not a button) between the two buttons. The label on this switch should be “Frosting”. The switch will take up most of the screen, **so update it so that it is a far more reasonable size**, like in the screenshot below.

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 there are two candles on the cake equidistant from each other and the edges of the cake, as shown below.



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

**Checkpoint 8 (20 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. Lab 2 will pick up from here, using the same project that you have modified. **If you are using the Shiley virtual machine and did not originally save your work to your P: drive, be sure to do so now before you log off, otherwise all your work will be lost!**