**CS407 Fall 2023**

Pair Programming: **NOT ALLOWED**
Due: **12:00PM CT on September 18$^{th}$**

# Lab01 Introduction to Android Studio and Git

## Overview

In this lab, you are going to set up your Android development environment. We will be using Android Studio which is the official integrated development environment for Google's Android operating system. This lab consists of the following **two main tasks**.

- **The first one** is to successfully install Android Studio on your personal computer and run a basic "Hello World" application along with setting up `VCS (git)` and pushing some changes to `Github`.
- **The second one** is to make a clone of our class GitHub repository and make a few changes that will be described in more detail below. The successful completion of this lab will help you get more comfortable with Android Studio and version control with Git.

This lab will help you complete more complex Android projects in the future.

## Learning Objectives

By completing this lab, a cs407 student will be able to:

1. Successfully **install** Android Studio on their personal machine,
2. **Get** familiar with the Android Studio interface,
3. Successfully **run** a first simple Android application,
4. **Get** familiar with Git and GitHub.

## Directions

### Milestone 1

#### A Download and Install Android Studio

The first step is to **download** the Integrated Development Environment (IDE) called **Android Studio** that we are going to use to develop our mobile applications and do the necessary set up. Android Studio is a free software development tool provided by Google.

Follow the instructions below to download and install it on your own personal machine.

1. First, follow this link provided by Google.

2. Then, navigate to the **Android Studio Download** section and **download** the Android Studio package that matches your platform. The most recent **stable** version is actually **Android Studio Giraffe**: android-studio-2022.3.1.18. It should show up for you at the top of the download page.

3. **Accept** the terms and conditions and follow the instructions to download and install Android Studio.

   Note that since Android Studio 4.0, Android Studio does NOT support 32 operating systems. If your computer's operating system is 32 bits, download Android Studio 3.6.3 through the download archives link.

4. **Accept** the **default configurations** for all steps.

5. Make sure that **ALL components** are selected for installation as shown below in Fig.1.

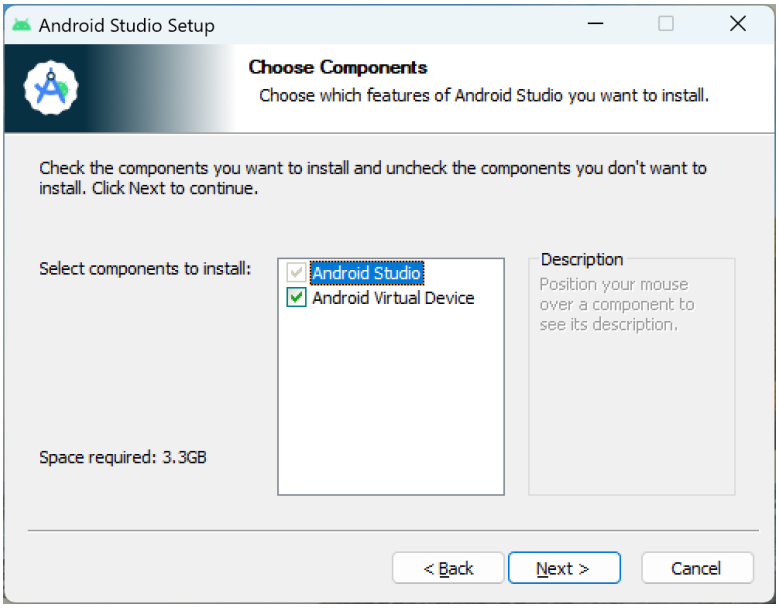



Figure 1: Android Studio Setup Window

6. After finishing the install, the Setup Wizard will download and install some additional components.

7. When the download completes, Android Studio will start, and you are ready to create your "Hello World" project.

## B Create the "Hello World" app

1. **Launch** Android Studio if it is not already opened.

2. In the main Welcome to Android Studio window, **click** "**Start** a **new** Android Studio project".

3. Customize the **Activity window** under the **Phone and Tablet** section. Every app needs at least one activity. An **activity** represents a single screen with a user interface and Android Studio provides templates to help you get started. Android Studio offers many different templates under the Phone and Tablet tab. For the Hello World project, **choose** the **Empty Views Activity template**. If you installed an old version of Android Studio (prior to the Flamingo one), you can select the "Empty Activity" template.

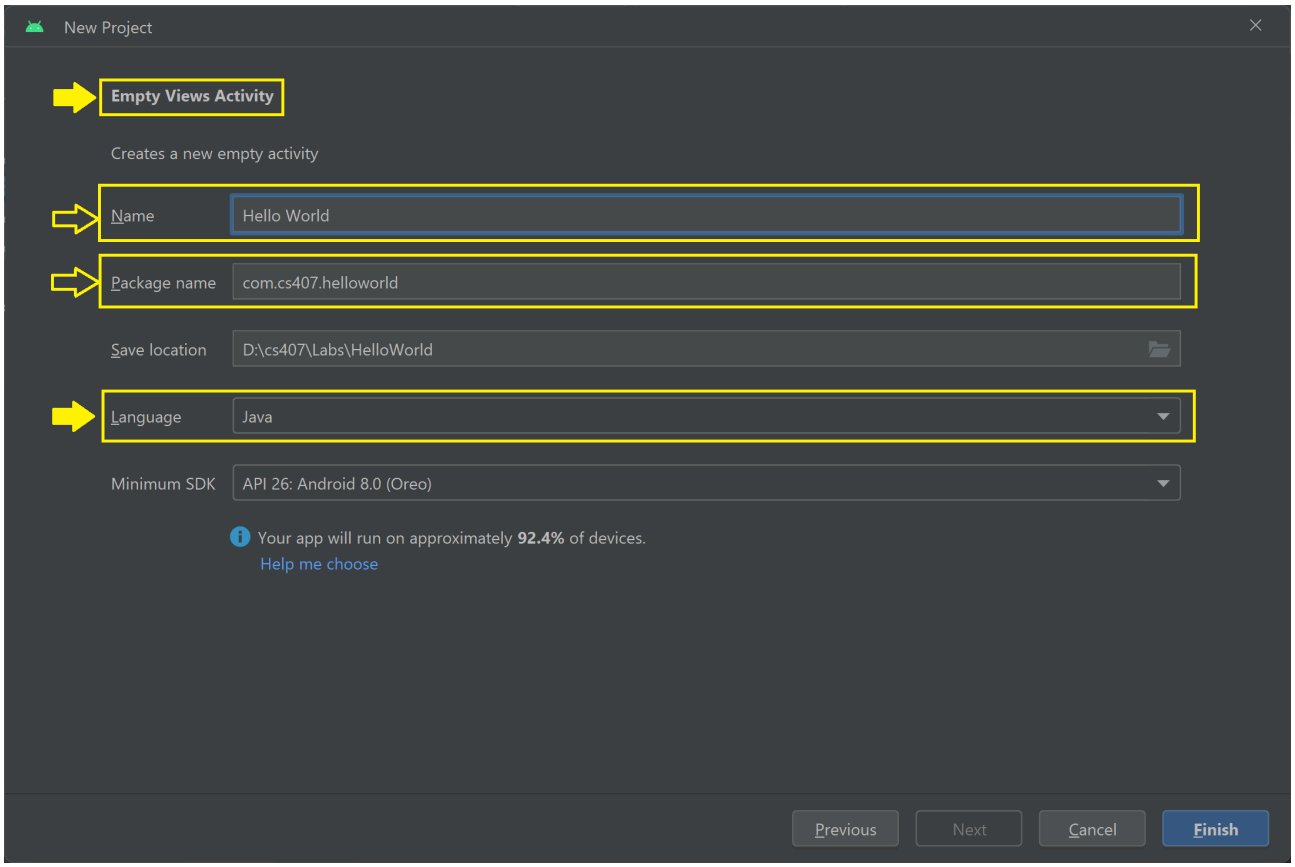



Figure 2: New Project Window

4. As shown above, in the New Project window (Fig. 2), give your application a **Name**. Use the **exact** name "Hello World".

5. Edit the **Package name** by setting it to the exact value `com.cs407.helloworld`

6. Make sure that you select the language as **Java**. If the Language selection option does not appear for you, go Previous and be sure to select the Empty Views Activity. Note that the ***Empty Activity*** template, for instance, does not support java as programming language in the latest stable version called Flamingo of Android Studio.

7. Verify that the default Project location is where you want to store your Hello World app and other Android Studio projects, or **change it to your preferred directory**.

8. Select **API 26: Android 8.0 (Oreo)** as the **Minimum SDK**. This defines the minimum Android version that your application is going to run on. As we select Android 8.0, this means that the application is going to work on Android 8, Android 9, 10, through Android 14 (the latest version of Android).

9. If your project requires additional components for your chosen target SDK, **Android Studio will install them automatically**.

10. Last step is to verify all the details that you have provided and click **Finish**.

After these steps, Android Studio:

- **Creates** a folder for your Android Studio project called `HelloWorld` in the folder you selected as your default Project location.
- **Builds** your project (this may take a few moments). Android Studio uses **Gradle** as its build system. You can follow the build progress at the bottom of the Android Studio window.
- **Opens** the code editor showing your project.

## C Create an Android Virtual Device (AVD) Emulator

Now that you have started an Android project, you will need a **virtual device** to **run** the app. To create an Android Virtual Device follow these steps:

1. In Android Studio, select **Tools** > **Device Manager**. Then, select the **Virtual** tab.

2. Click the **Create Device** button. The **Virtual Device Configuration** window appears.

3. Select **Phone** as category. A screen showing a list of preconfigured hardware devices. For each device, the table shows its diagonal display size (Size), screen resolution in pixels (Resolution), and pixel density (Density).

4. Choose the Pixel 6 Pro hardware device and click **Next** as shown in Fig. 3. If Pixel 6 Pro device is not shown, select any other device.

5. On the **System Image** screen, from the **Recommended** tab to **Select a system image**, choose which version of the Android system to run on the virtual device. You can select the latest system image, and click Next.

6. On the **Verify Configuration** screen, you can select a name for your AVD (all the other settings are default) and click **Finish**.

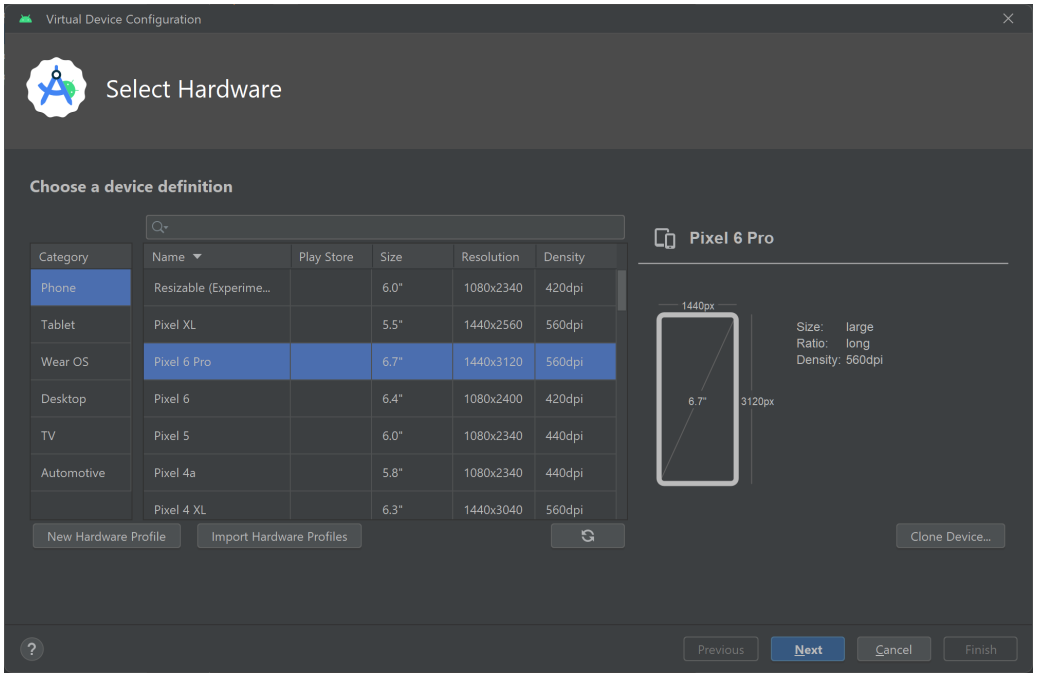



Figure 3: Select Hardware Window

## D Run the App

Now that you have setup an Android Virtual Device and a "Hello World" project all that remains to do is run this application on the Emulator.

1. In Android Studio, select **Run** > **Run app** or click the **Run icon** in the toolbar.

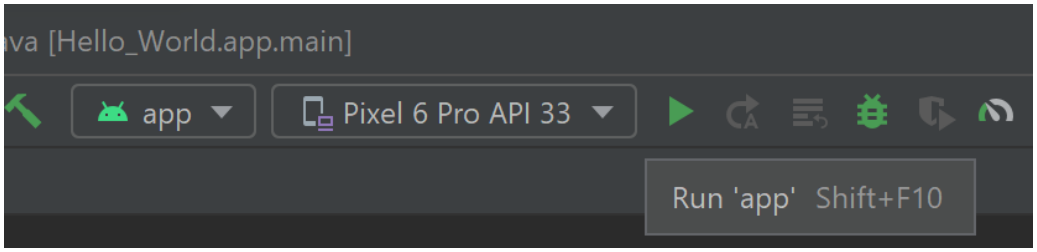



Figure 4: Run App Button

2. If you complete all these steps successfully you should see the emulator showing something like Fig. 5.

# Milestone 2: Successfully setup VCS(Git) and push changes to github

**Version Control Setup** - Now you will be learning how to use and integrate a **Version Control System (VCS)** with Android development. We will be using `Git` as our version


Figure 5: Hello World on Emulator

control software to track your progress. In this milestone you will **connect** your **GitHub account** to your Android Studio project.

- If you do NOT already have a GitHub account then you can **sign up** at this **link**.
- We will be using **GitHub classroom** in this course to monitor your progress. Below is an easy step-by-step tutorial to set up your first GitHub repository on Android Studio.
- **Install `Git` on your computer:** Visit this **official site** to download git on your computer. Once you do that, you can start using it with android studio.
- **Enable** Version Control Integration on android studio: **Go** to **Menu>VCS>Enable Version Control Integration**

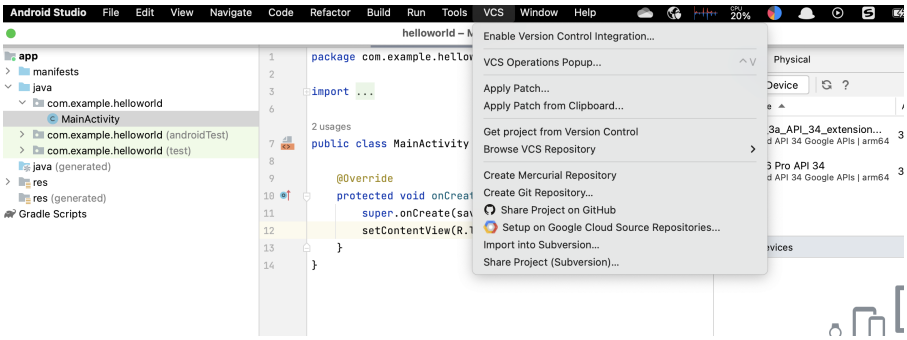



Figure 6: Enable Version Control Integration

- After that, **select** `Git` as your version control system.

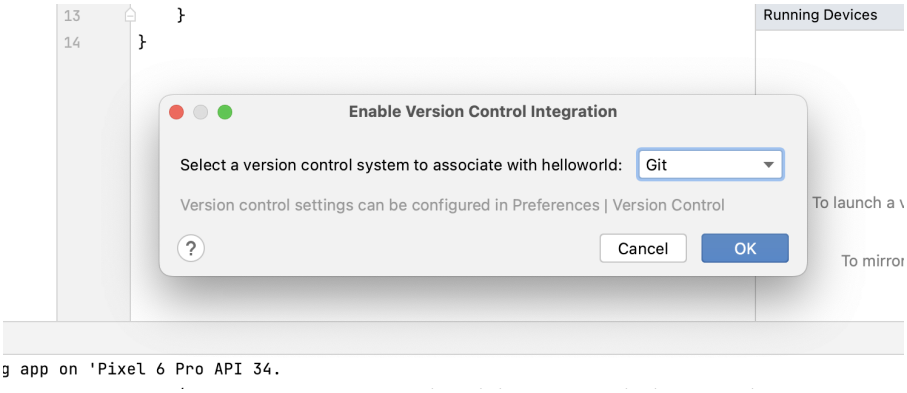



Figure 7: Select Git as Version Control System

On doing this, you will notice that all your files will turn red.

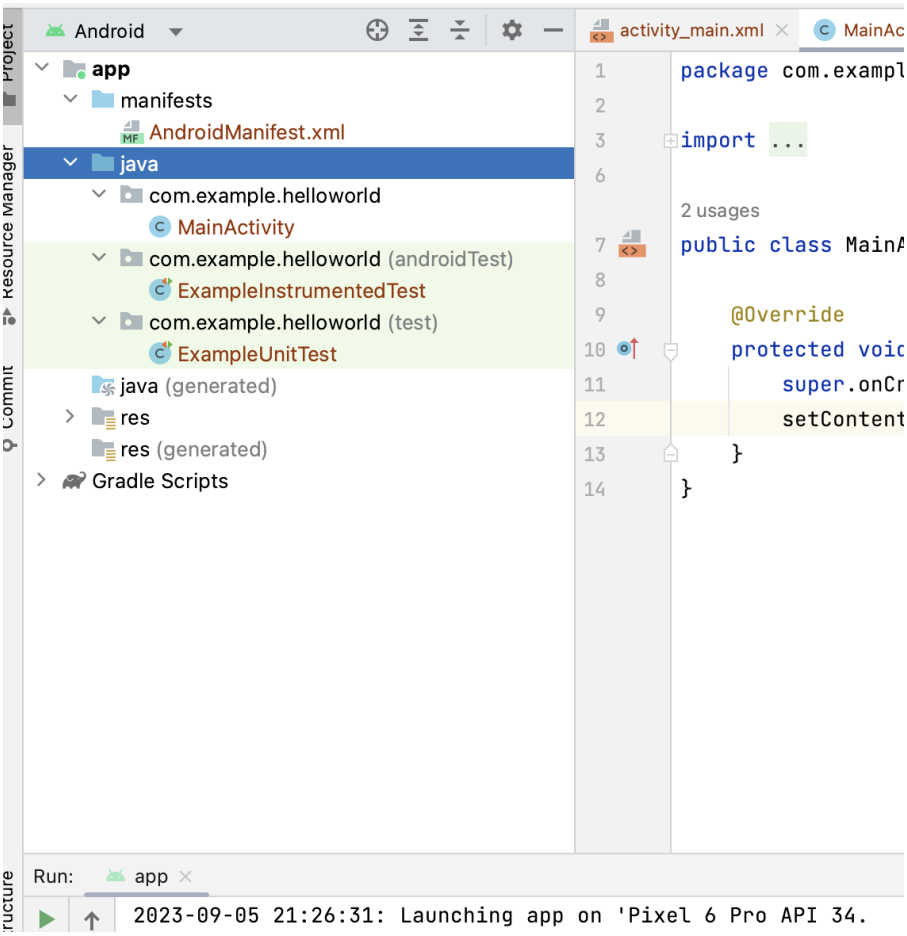



Figure 8: Files Turning Red

- **Share on Github:** Now, go to **VCS → Import into Version Control → Share project on Github or Git → GitHub → Share project on Github**.

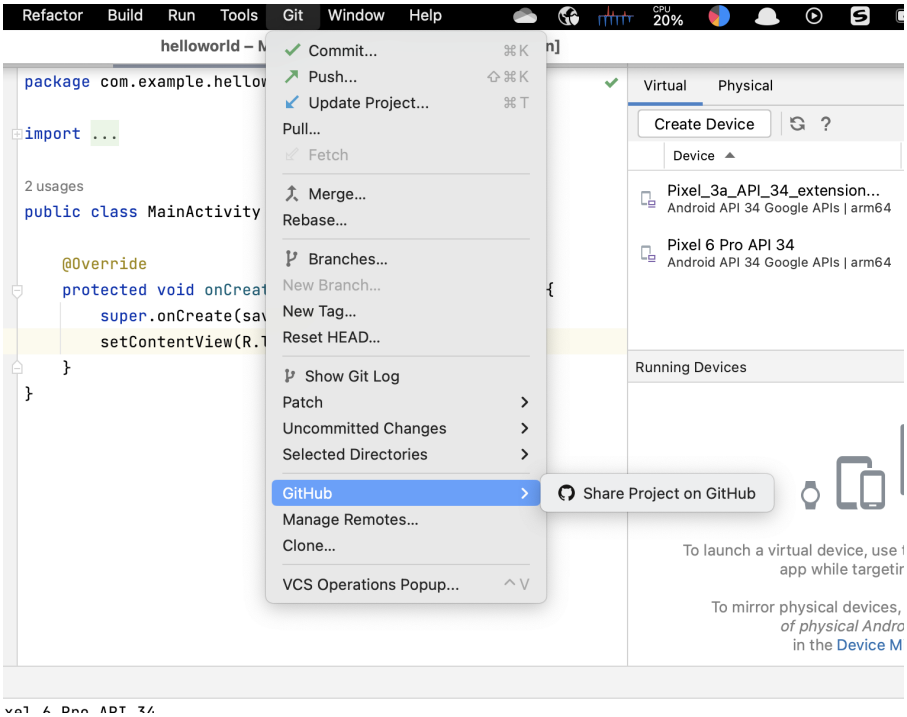



Figure 9: Share Project on GitHub

- Now **give a name** to your repository and write a description if you want. This step may require you to enter **github credentials**.

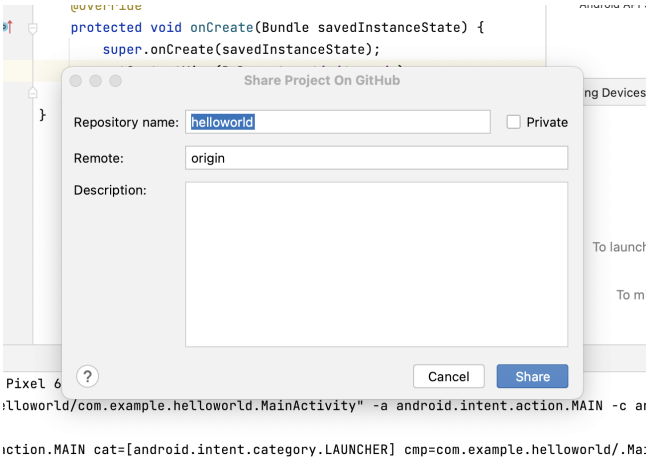



Figure 10: Name Your Repo

- **Commit and Push your project files on Github:** Now select the files you want to

share and press Add. You can also write a specific commit message and after you do so, all your files will turn white.

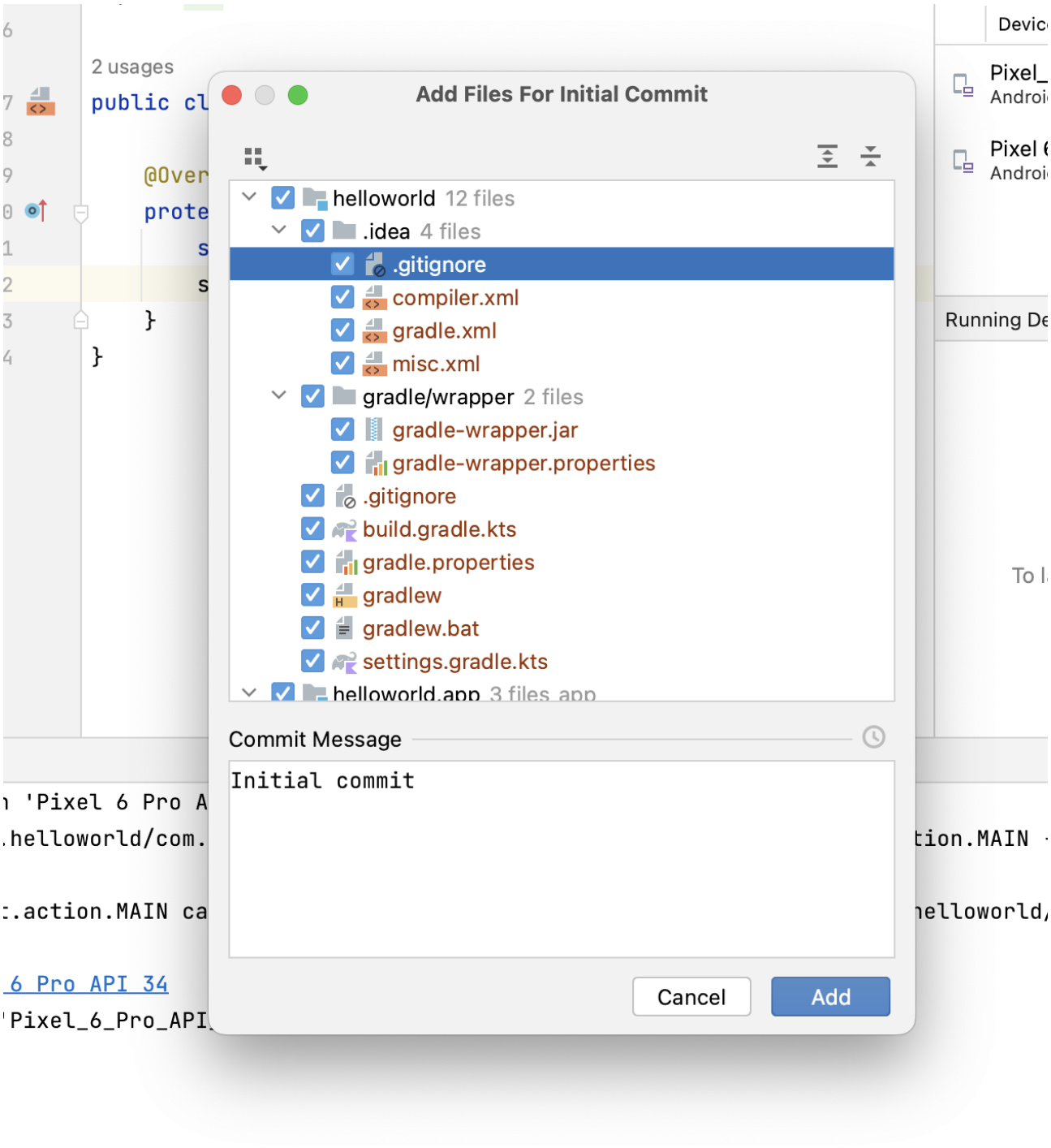



Figure 11: Select Files to Push to GitHub

- Your project is now under version control and shared on Github, you can start making changes to **commit** and **push**.
- Add a new `ImageView` reference to the MainActivity.java. Notice that after the changes are made, the MainActivity.java file turned blue. That means the file now has uncommitted changes (and it doesn't match the file in the Github repository).

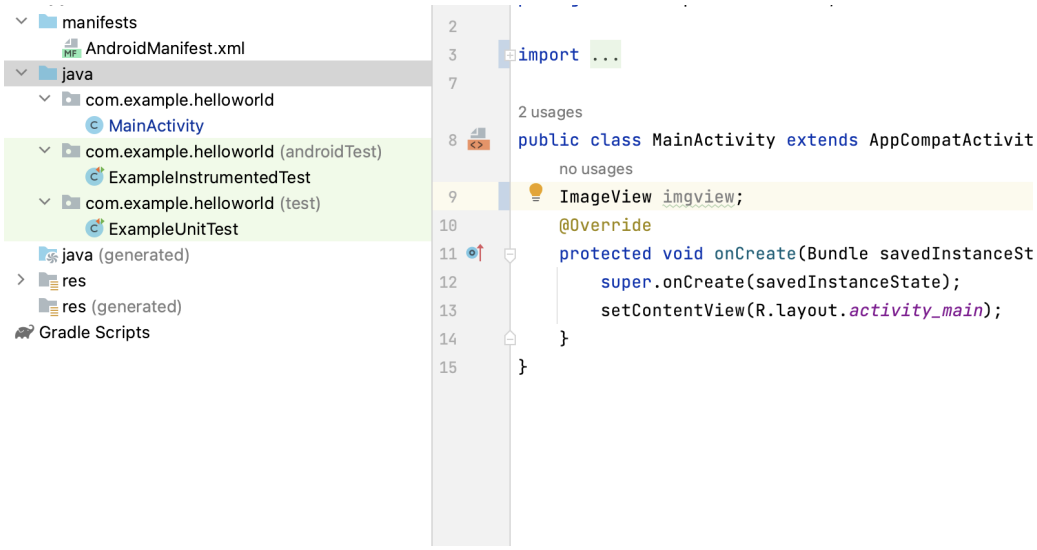



Figure 12: Reference of ImageView in MainActivity

- It is up to you now when you want to commit and push. You can do it after every new change or after a lot of changes. To commit and push, go to Menu → VCS.

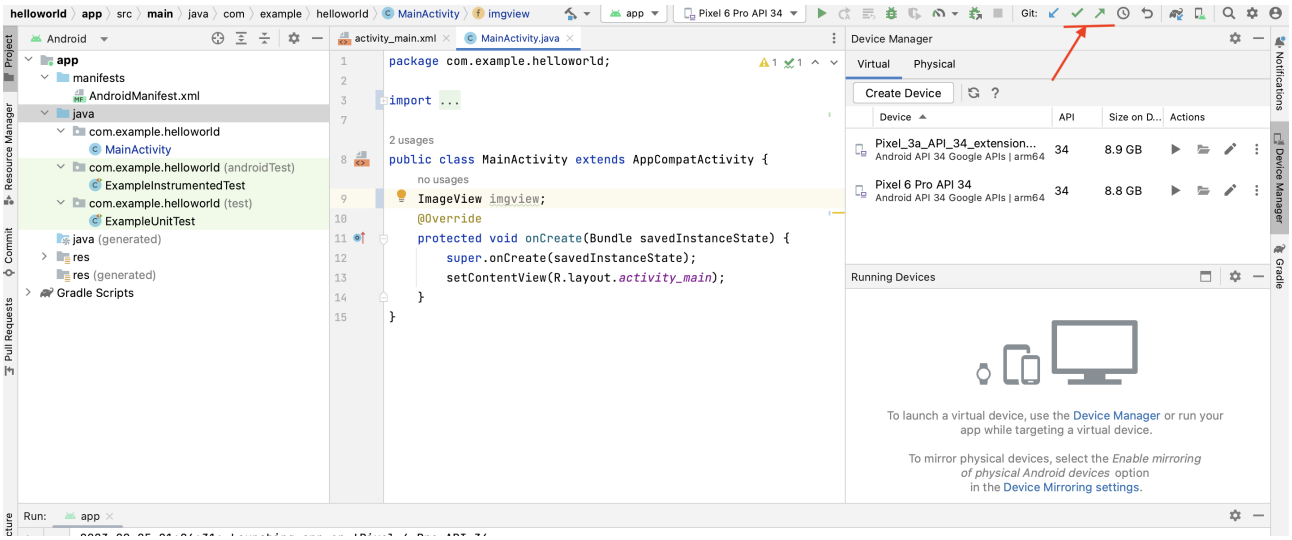



Figure 13: Git Toolbar at the Upper-Right Corner of Android Studio

- After that, you will see a commit changes window/panel and you have to write a commit message (mandatory) and then press on commit and push.

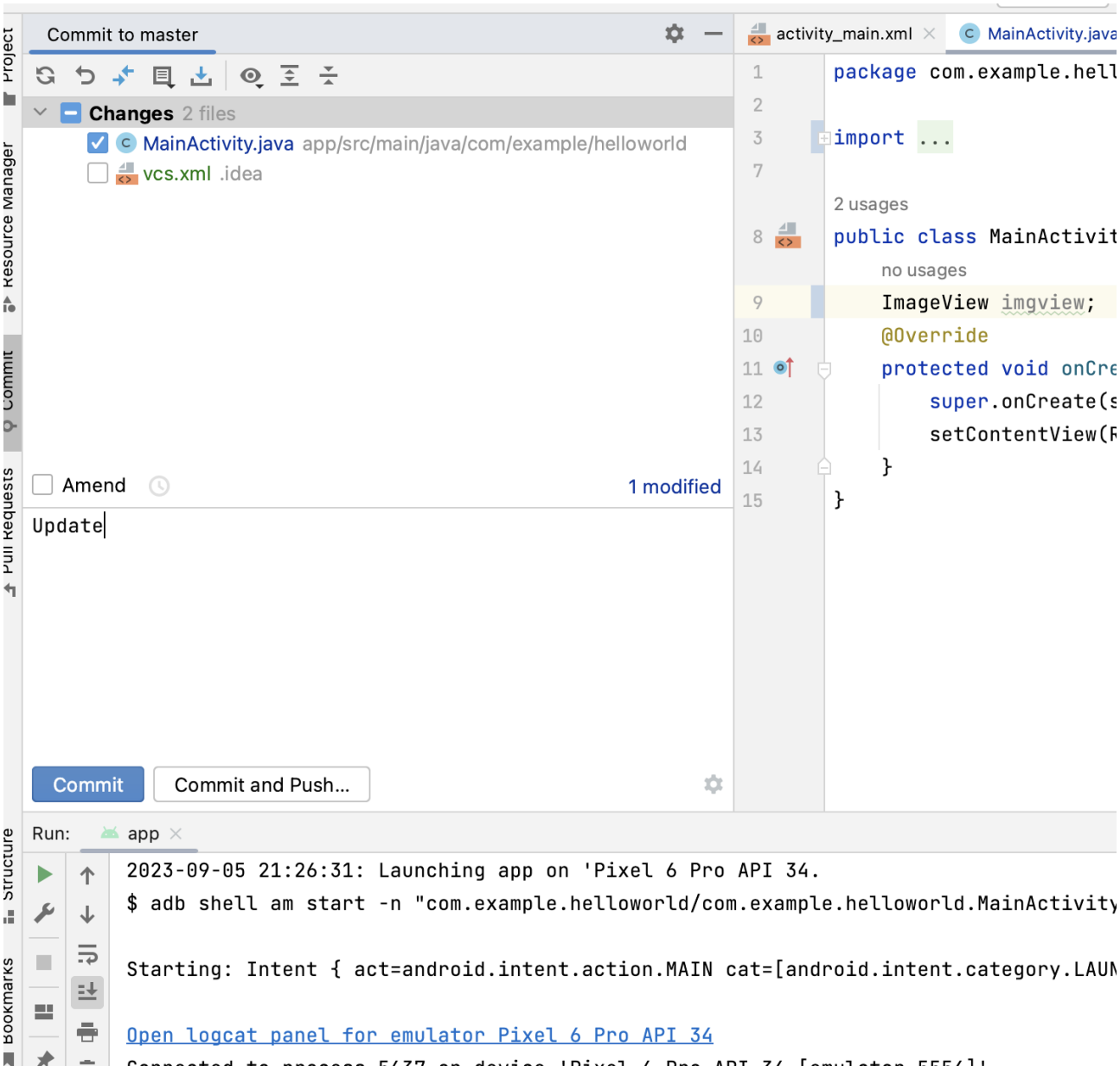



Figure 14: Commit window/side-panel with files modified

- After this step you will now push your changes to the GitHub repository. You will do this by navigating to VCS → Git → Push / Git → Push.

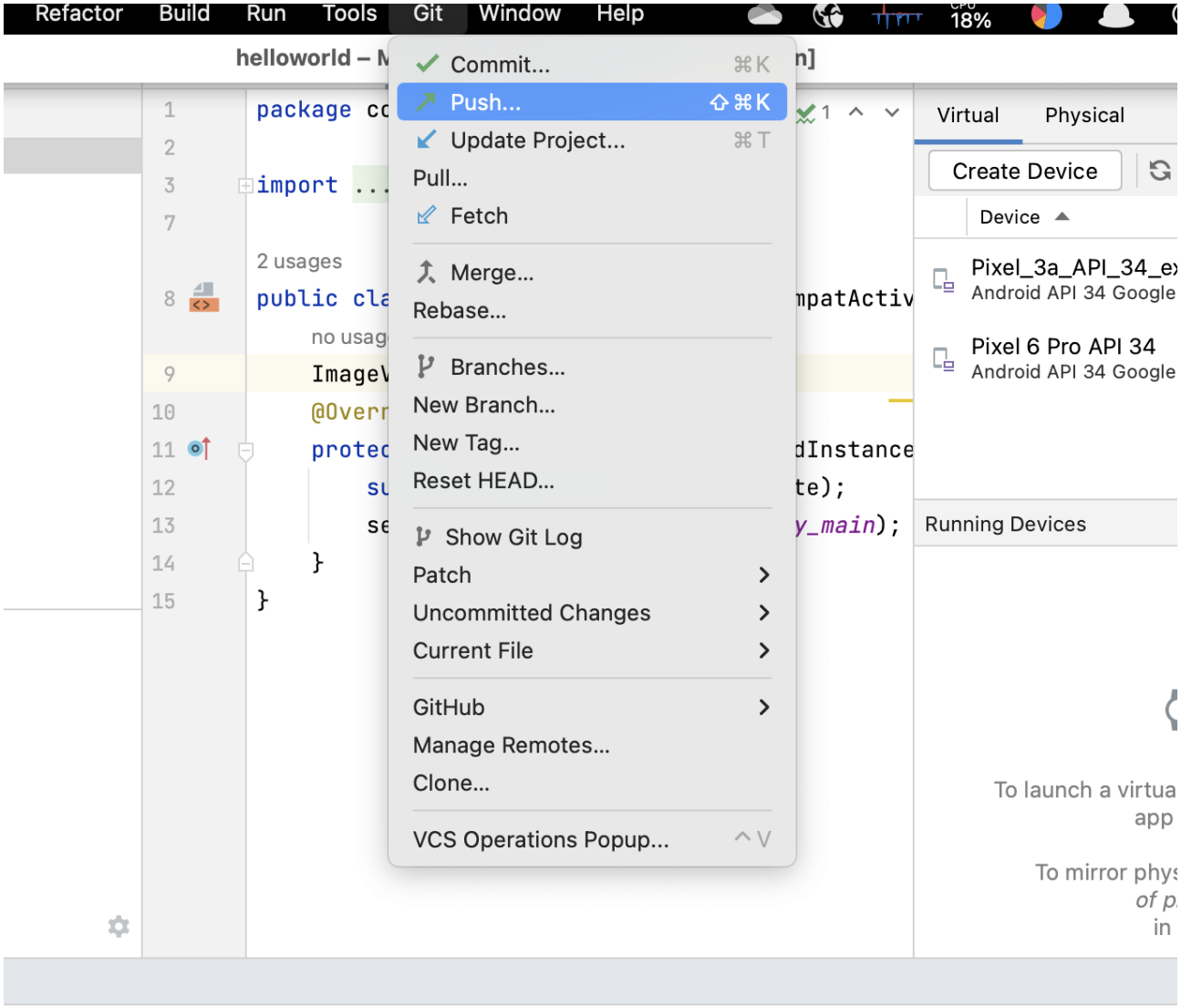



Figure 15: VCS → Git → Push / Git → Push

- Finally, you will see another push window. Just press on push button and all your changes will be reflected in your repository.

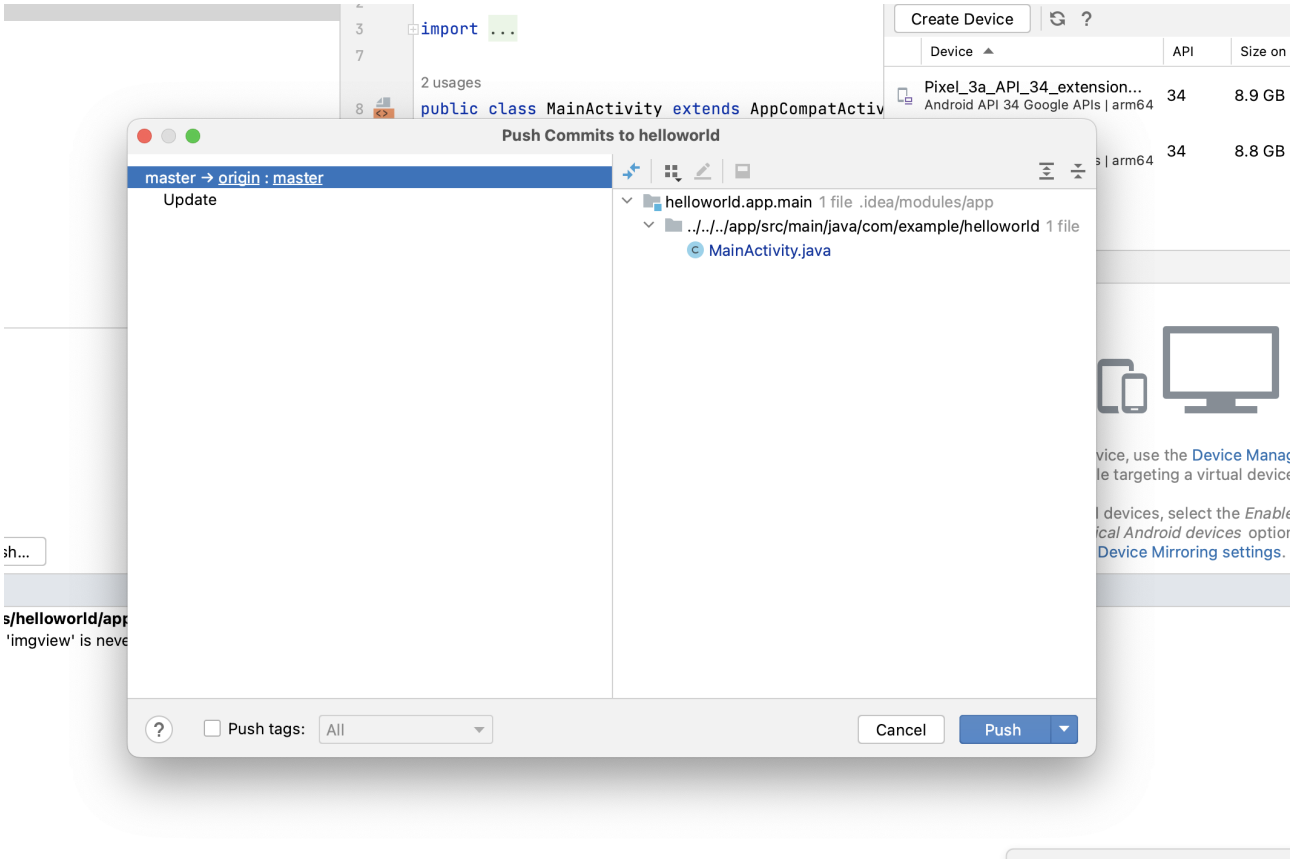



Figure 16: Click the Push Button

**Congratulations** on setting up your first project with version control.

## Milestone 3: Share your project on Github classroom

In this milestone you will have to successfully **share** your **local repo** in the `github classroom`. Once you have the most updated copy of the code you will have to make a few changes to the code which will be described below.

1. Start a **new project** by selecting **File** → **New** → **New Project**

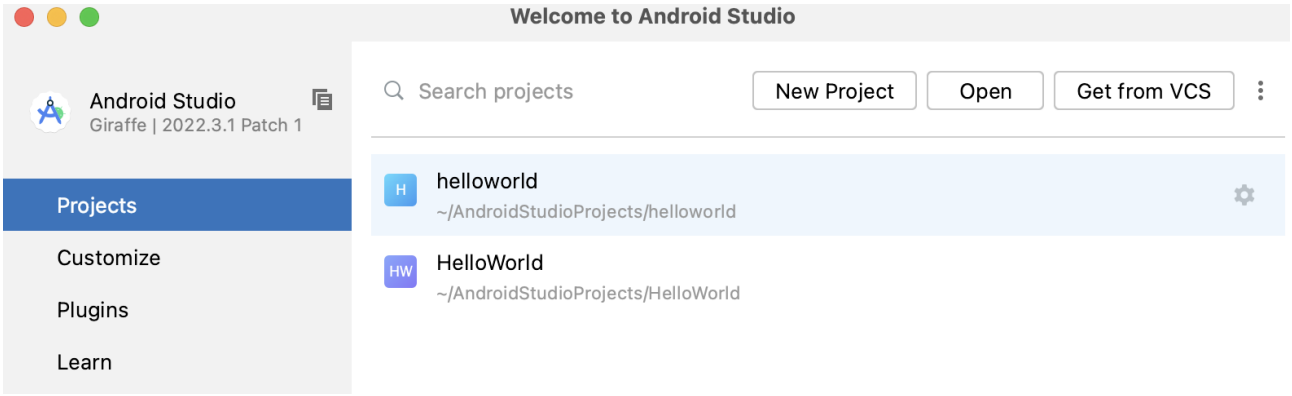



Figure 17: Starting new project in Android Studio

2. Then click the following link: https://classroom.github.com/a/amwsx2Js to accept the invitation for this assignment.

3. Once you click **authorize** you will get the **repo** with a link (it will be something like https://github.com/cs407uw/lab1-yourgithubnamehere) .

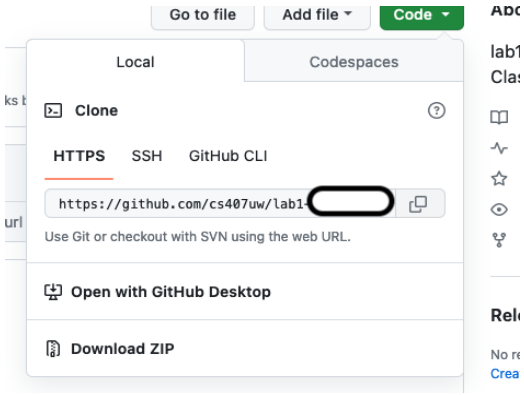



Figure 18: Sample github repository setup with link to repository

4. Then, you will need to **initialize** this project with **git** as done in Milestone 2.

5. Next, **set up** the **remote repository** with the ssh link above by clicking Git → Manage remotes → add.

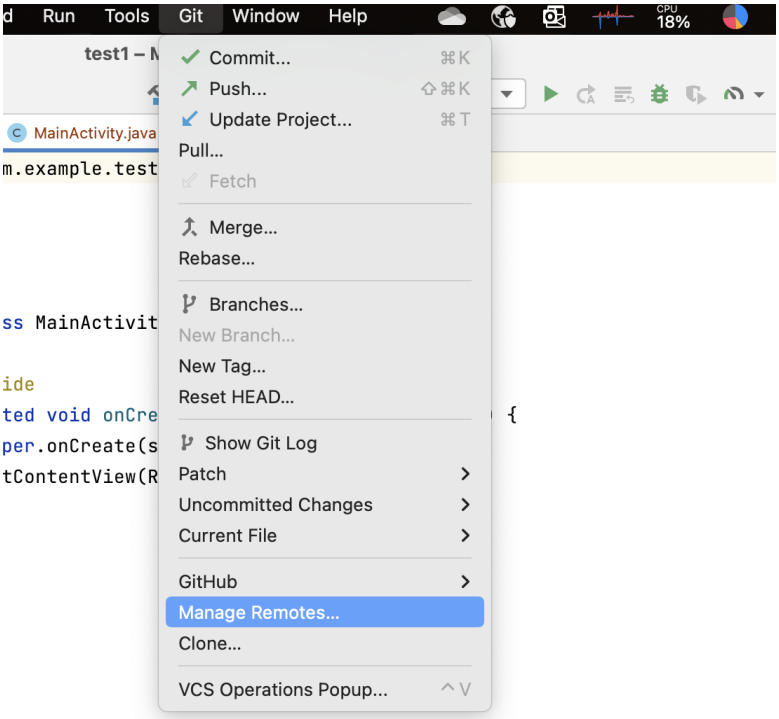



Figure 19: Setting up remote repository

6. Type in your `git url`, please use the https link. Then it may trigger github login. For authentication, you will see Fig. 20

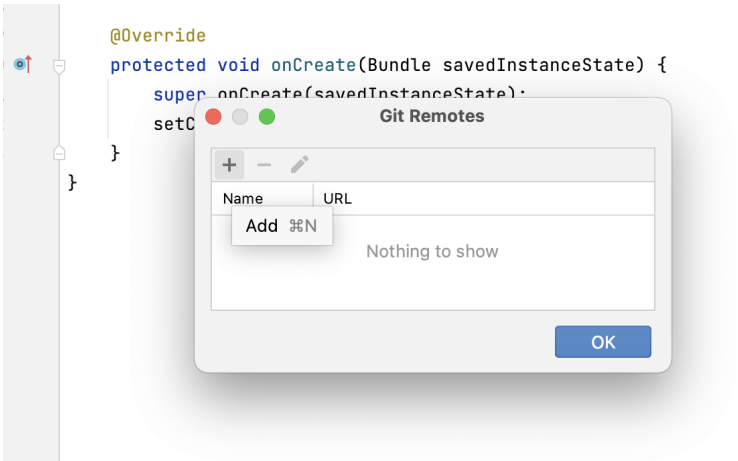



Figure 20: Github Authentication Step 1

7. Then click on the use token, as shown in Fig. 21 below.

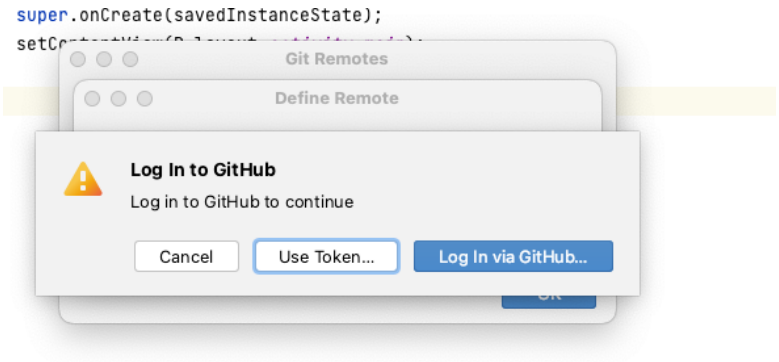



Figure 21: Github Authentication Step 2

8. Click on generate to get your personal access token generated from github. You can configure your access right given by the token.

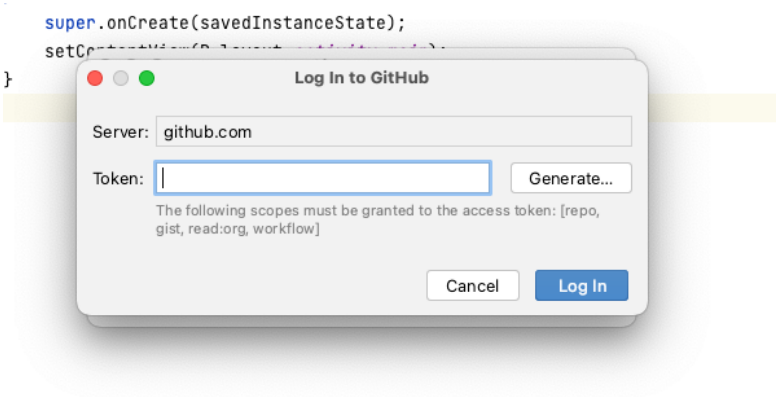
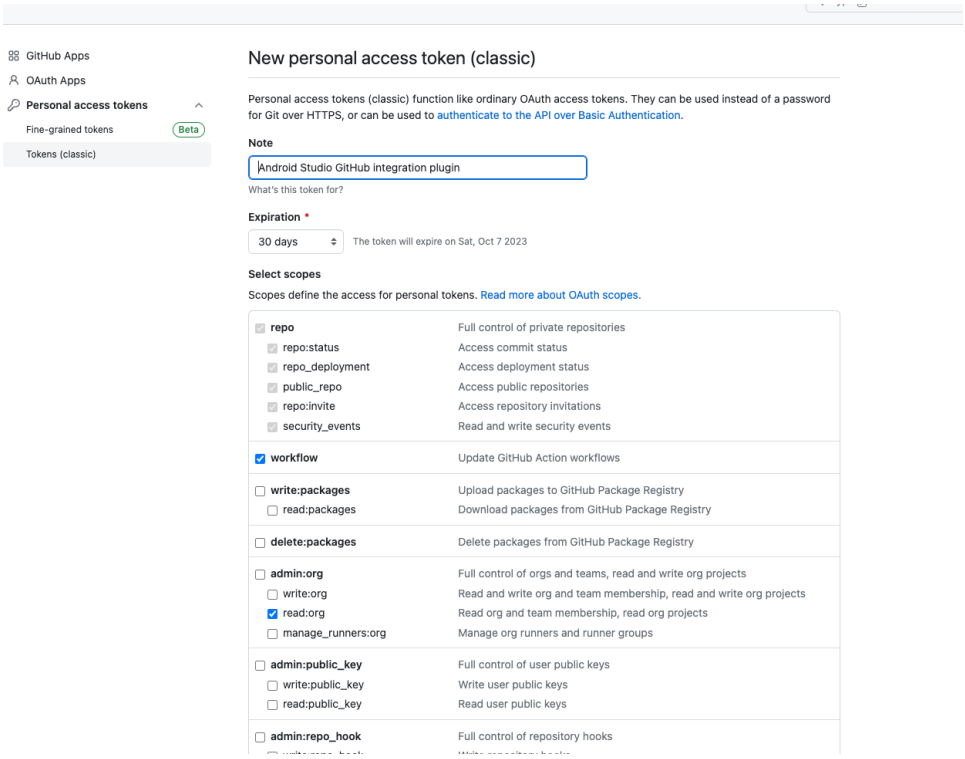
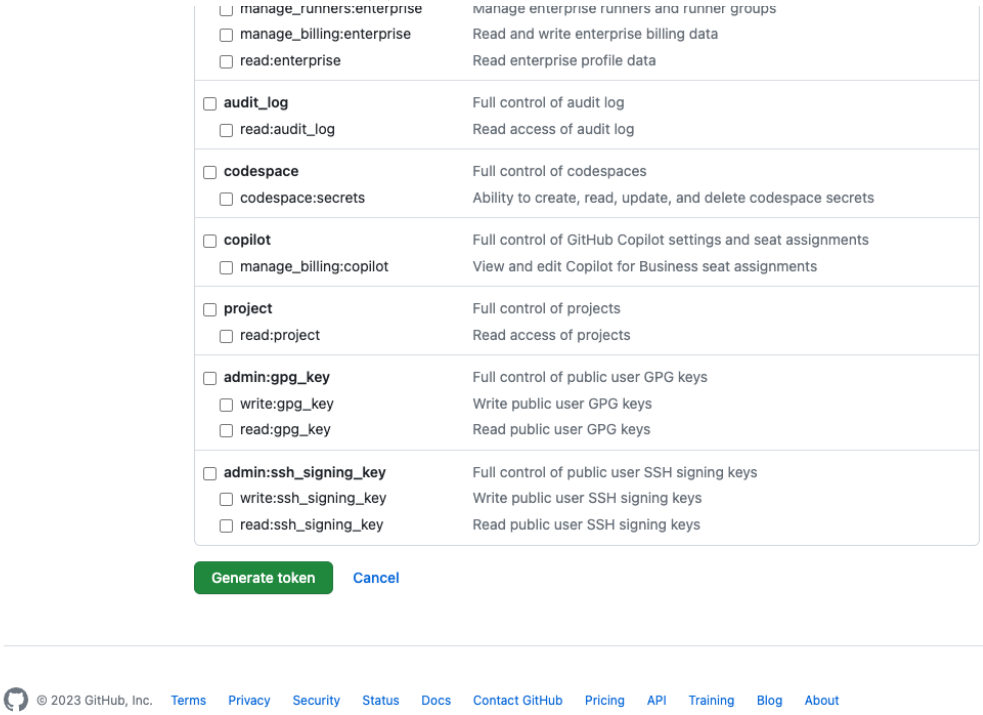



Figure 22: Github Authentication Step 3

9. Create the github personal access token from step shown in the image below. Click generate token to create the token and copy it for future use.




(a)




(b)

Figure 23: Creating personal access token

10. Then paste the token to the authentication of Android Studio.

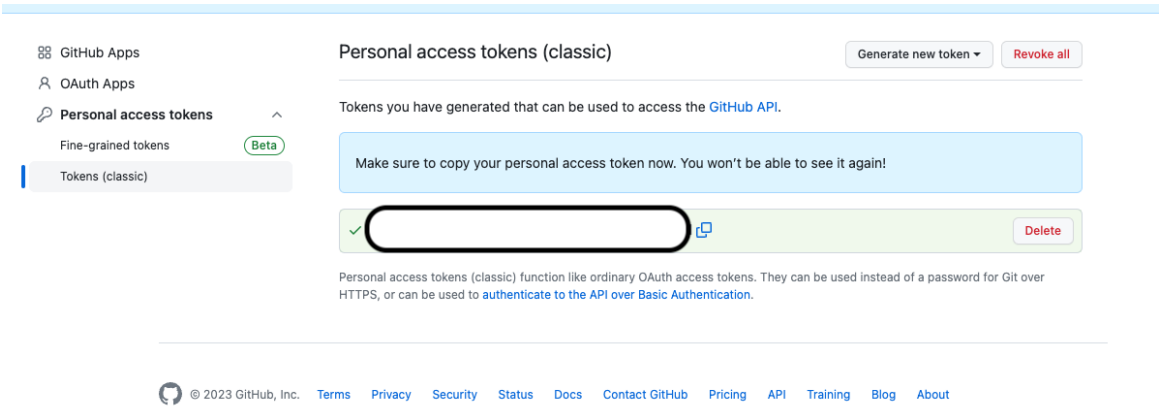



Figure 24: Copy the generated token

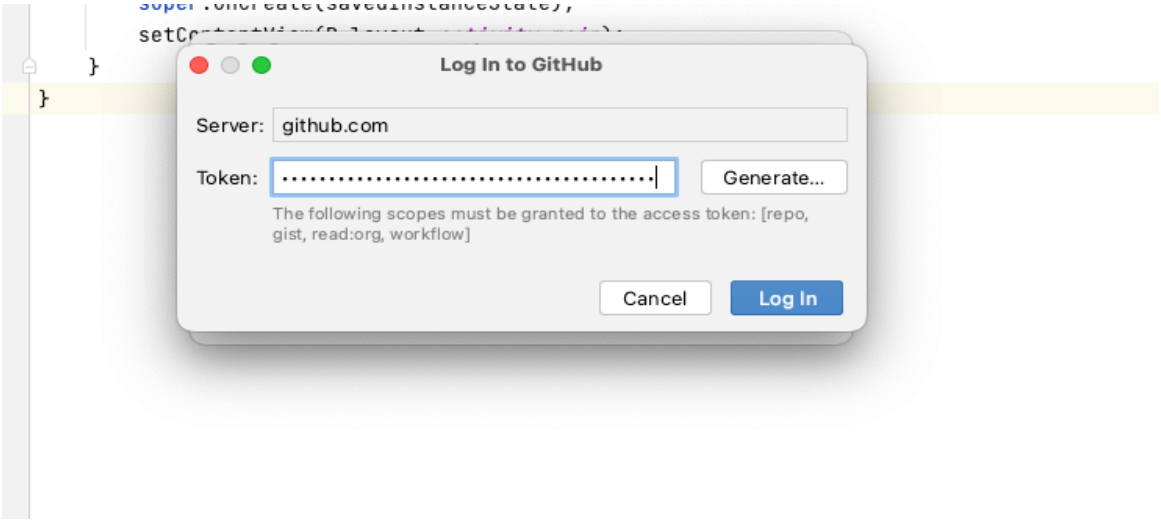



Figure 25: Paste the token and Complete Github Auth

11. This will allow the android studio to have access to your github repos. After the authentication is done you can find that your local project can be found in the commit panel/window. Then commit and push it.

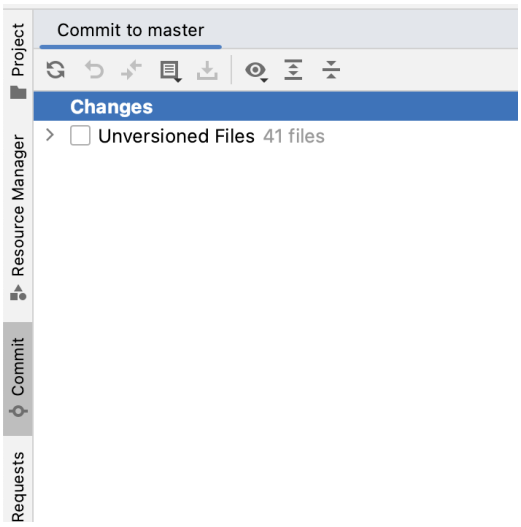



Figure 26: You can see uncommitted changes here

12. You will find your update is checked in the github classroom in branch master!

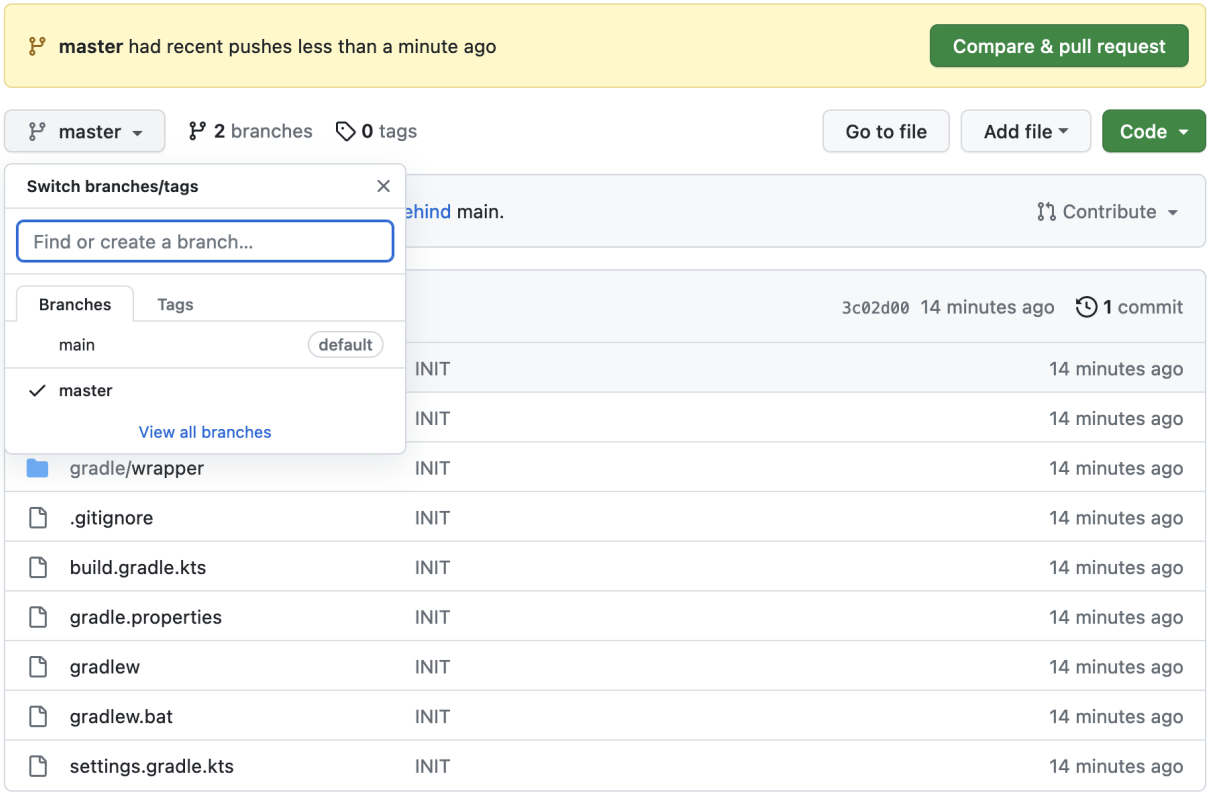



Figure 27: Github classroom local clone

Once you have completed this step you will be left with a local clone of our GitHub classroom

repository. Now you have the most updated copy of the code you will have to make the following changes to the code given to you:

- In the *activity_main.xml* file, ound in the files on the left hand side : app → res → layout. Once you open this file from the file explorer you should be looking at some code like the one in the screenshot below.

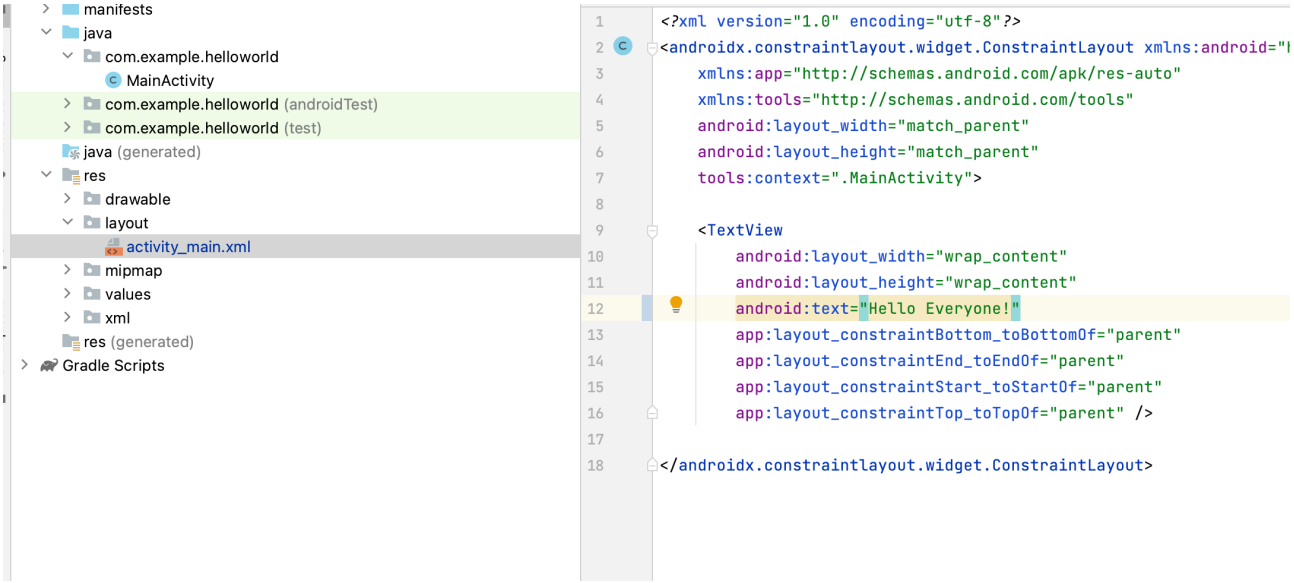



Figure 28: Starting code for step 1

- Your first task is to change message from the original text to "Pass!" You will need to make this change in the following location.

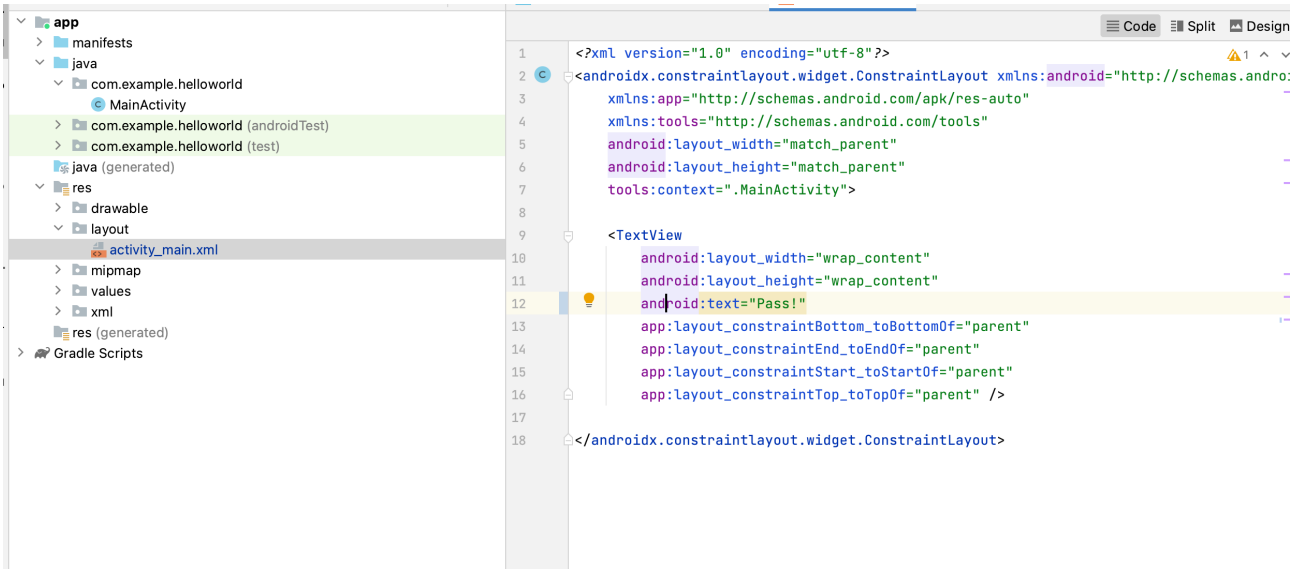



Figure 29: Changing message text

- Your second task is to add a Button by using the design tab. You can use this design tab to drag and drop various components onto your app. For this task you should select Common → Button and drag and drop it onto the app.
- You will now be able to see it on the app preview. You can move it around and place it wherever you'd like as shown in the screenshot below.

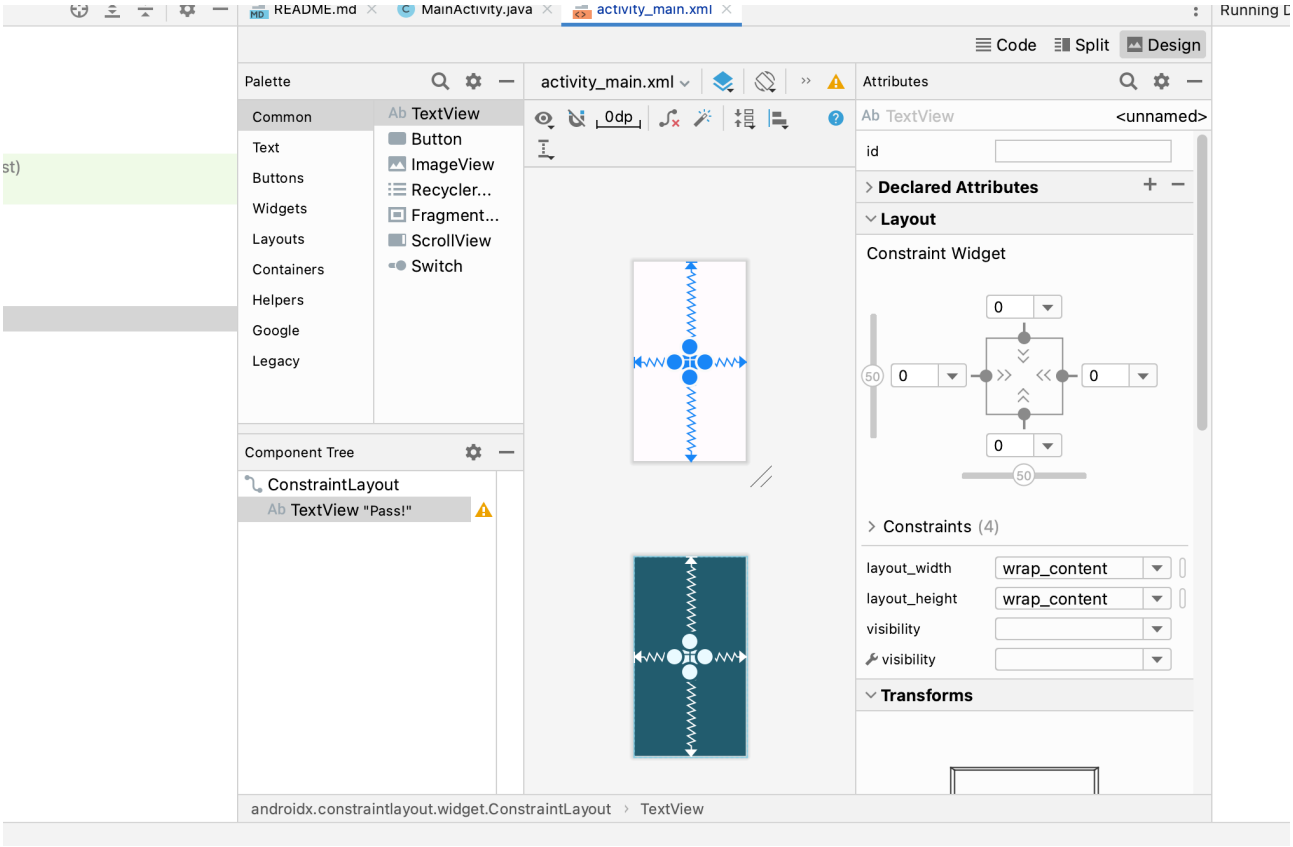



Figure 30: Inserting Button

- Click on "Infer Constraints Symbol" shown by arrow in the below screenshot to create automatic constraints.
- Change the text that the Button displays by going back to the text/code tab or you can change it in the design tab by first selecting the button and then editing the common attributes (common attributes → text) as shown below.

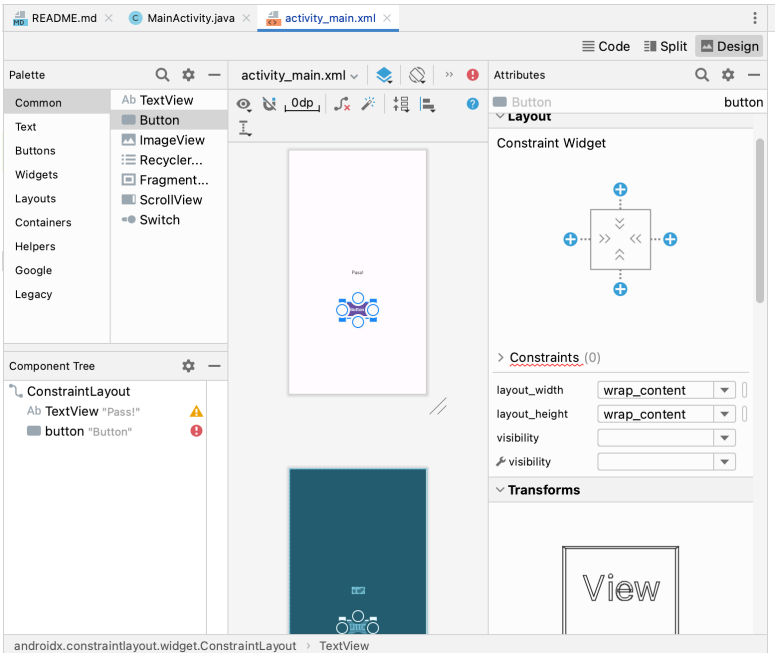



Figure 31: Constraint Layout

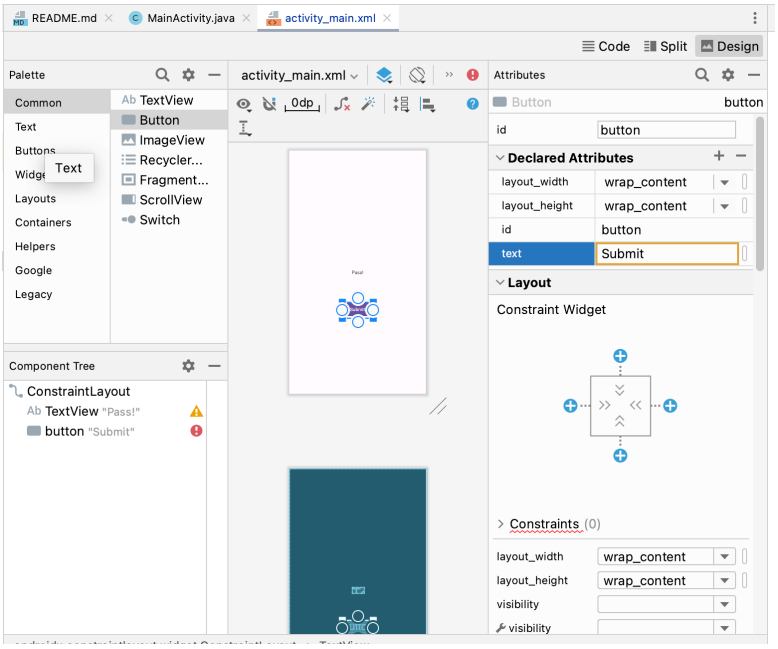



Figure 32: Changing text for button - step 1

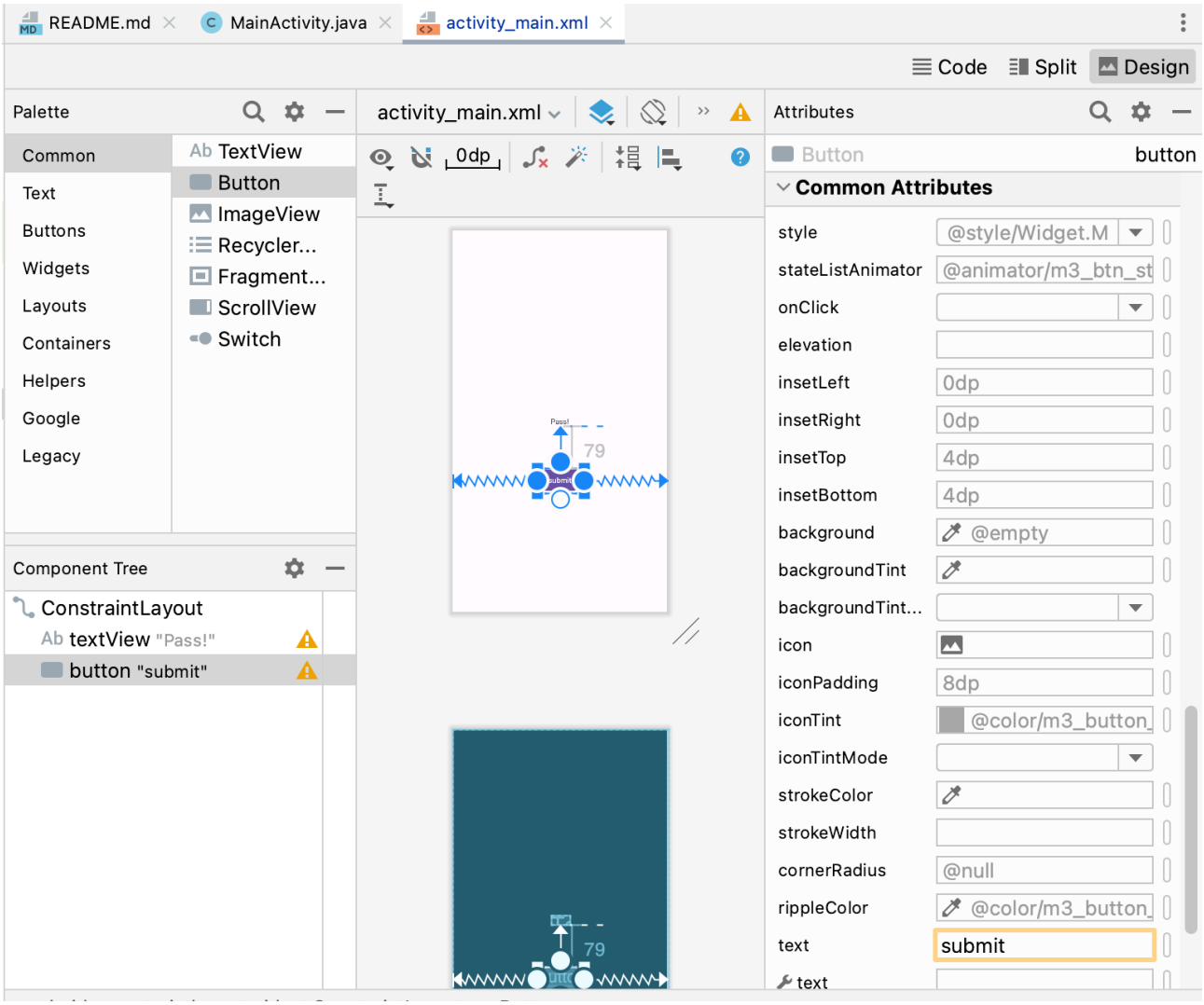



Figure 33: Changing text to button - step 2

- As the semester progresses, you will get more and more comfortable with the various resources that Android Studio provides.
- Now that you have made these changes, `commit` **these changes** to your **local version control repo** as you have learned from the git tutorial above.
- After you have made these changes to your android project, the last milestone is to `Git push` **all these changes** to your **remote GitHub classroom repository**. We will be checking the time and date of the last push to your individual repositories and grading you accordingly.

# Conclusion

You have done a **great job** to get this far. Generally, the setup is the most boring part of any software development. Now that you have successfully completed this you are ready to take on more complex challenges that are waiting for you in the next many weeks of this semester.

# References

- Google has great documentation for Android and this is a good link to help get you started with this project.
- Github set up tutorial.

# Gradescope Submission

**Congratulations on finishing this CS407 Lab!** The only TWO files that you must submit to Gradescope are:

- `MainActivity.java`.
- `activity_main.xml`.

Your score for this assignment will be based on your **"active"** submission made prior to the deadline of Due: **12:00PM CT on September 18**$^{th}$.

The second portion of your grade for this lab assignment will be manually graded by our TAs/graders (Demo sessions). Penalties to your lab's grade are applied if your lab is not manually graded before **5:00PM CT on September 18**$^{th}$.