Task plan for a beginner Android developer

Mentorship System

by Bartek Pacia Google Code-in 2019

task link

This document contains hyperlinks – it is intended to be read on an electronic device with Internet access. Do not print.

Introduction

I decided to create a relationship example between people focusing on entry-level Android development. I chose it because I'm programming on Android since 2016 and I feel pretty confident with it.

Assumptions about mentee's level

I have made a list of assumptions I made about the mentee's level of skills/knowledge so that the tasks make sense.
Mentee:

- Has some basic experience in Java programming, though hasn't got contact with Kotlin yet
- Has basic knowledge of Object-Oriented Programming
- Does not have a GitHub account
- Cannot use git
- Does not have Android Studio installed
- Knows the basics of using terminal

Introductory message to a mentee

I think that establishing a good relationship between a mentor and a mentee is very important to make learning pleasant and efficient. They should feel good and comfortable talking to each other.

That's how my first, "welcoming" message to my hypothetical mentee would look like:
"Hello! I'm Bartek and I've been programming Android apps for more than 3 years now. I'm very happy to be your mentor, I look forward to helping you with all your

question, concerns and issues. Remember, stupid questions don't exist, so if you feel like you don't understand something well, ask, ask and ask :)
I have prepared some tasks for you to jump start your journey with Android development. I recommend you to complete them so you'll have strong fundamentals that you'll build upon in the future.

Looking forward to hearing from you!
Bartek"

Tasks

1. Introduction to Android development

   Description

   Android is the world's most popular operating system. It is used by more than 2.5 billion people[1]. For years it was considered hard to create high-quality apps for it because of its complexity, but it's getting easier and easier since 2017[2]. To sum up, it's never been a better time to develop for the Android platform.

   To do

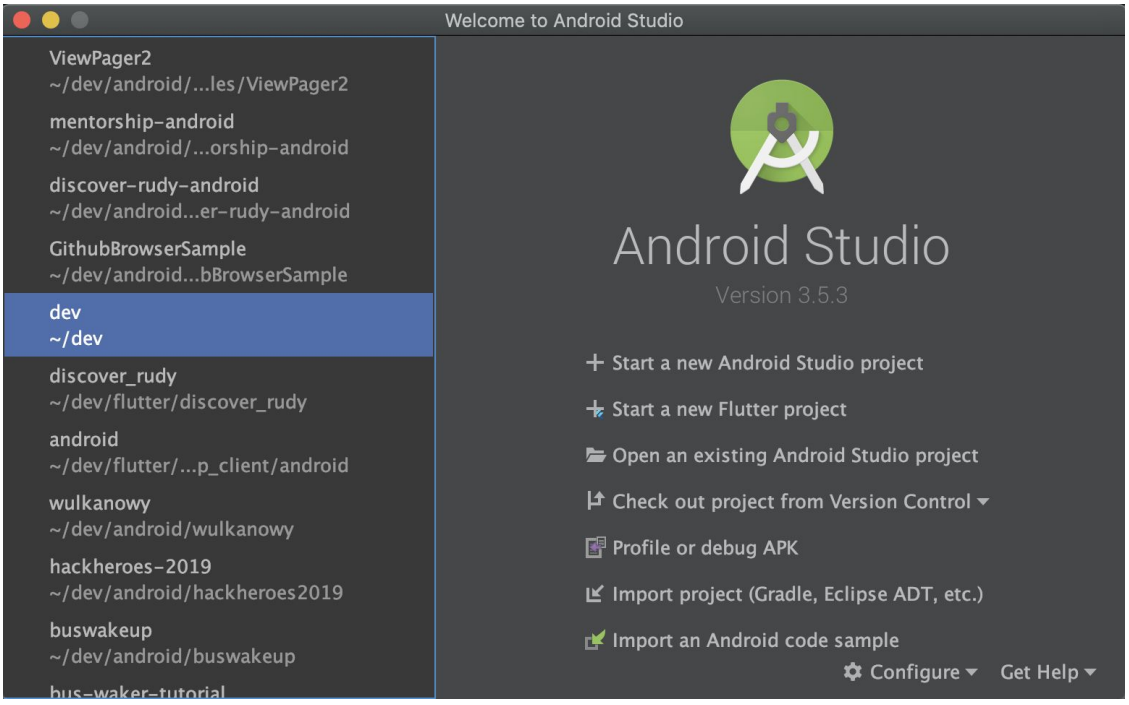
   Your first task is to head over to developers.android.com and install the newest version of Android Studio, the official Integrated Development Environment (IDE) for creating Android applications.

   Submission

[1] Source: https://www.androidpolice.com/2019/05/07/there-are-now-more-than-2-5-billion-active-android-devices/

[2] Precisely, since Google I/O 2017, where Android Architecture Components were announced and Kotlin was made an official programming language for Android.

Show screenshots of opened Android Studio, like below (don't worry that the list on the left is empty – we will work on that :))



2. First app

Description

Now that you have Android Studio up and running, let's create something very simple. You can get additional help and guidance from Creating a project guide on Android Developers site.

To do

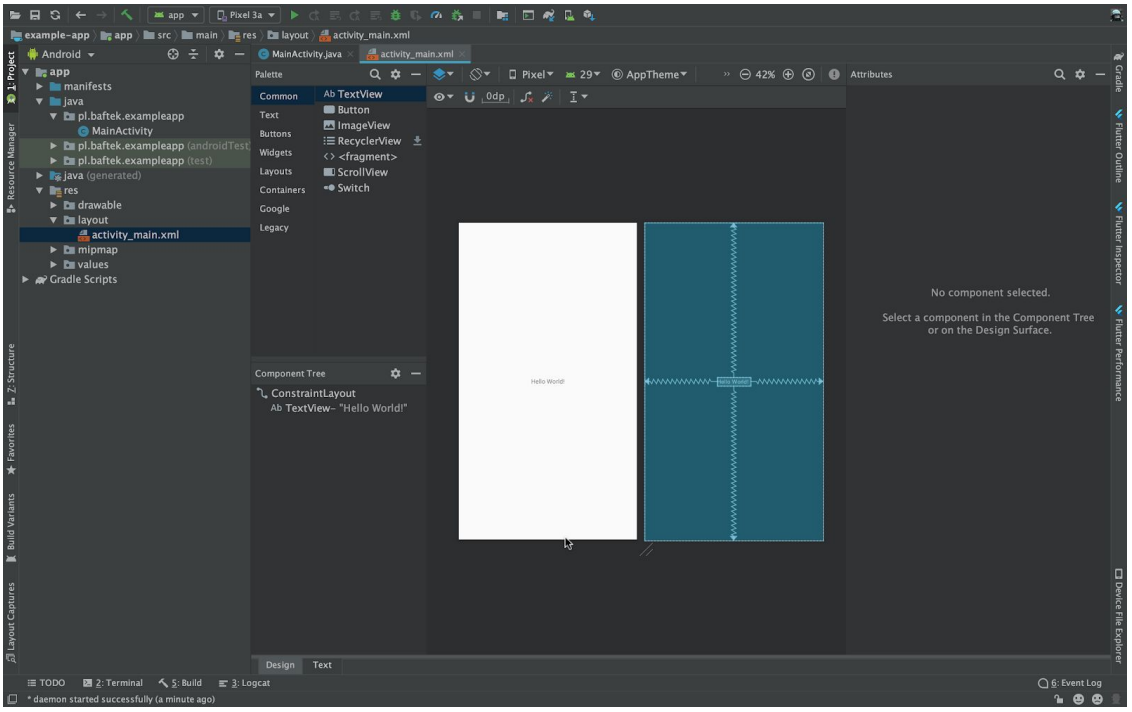
Click on 

and follow the instructions on screen. Select Empty Activity on the first screen, Java as the language and API 21 as the minimum API level.

After your project is set up successfully, explore the directory structure on the left. Open files and see what they contain. Pay special attention to app/java and app/res/layout folders – that's where you'll spend most of the time coding.

Submission

Show screenshots of a newly created project displaying file app/res/layout/activity_main.xml, like below:



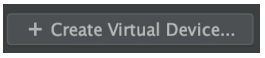
3. Creating a virtual device

Description

Now that you have a simple project, you have to run it somewhere. The good news is that you don't need to own a real, physical Android phone – you can use a program that *emulates* the behavior of such on your computer. Such a program is called an emulator.

To do

Click on this icon  on the Android Studio toolbar. A window will open up.

Click on 

button at the bottom. Select Pixel 3a as the device and select the highest possible API level. Leave the rest of the options, they aren't relevant to us on this stage. When you're ready, hit "Finish".

To run your app on the newly-created emulator, hit the green arrow button on the toolbar  .

If you need more guidance, check out the resources below or ask me :)

Resources

Run your app

Run an app on Android emulator

Submission

Send me a screenshot of the example app running on the Android emulator.

4. Displaying a simple message

To do

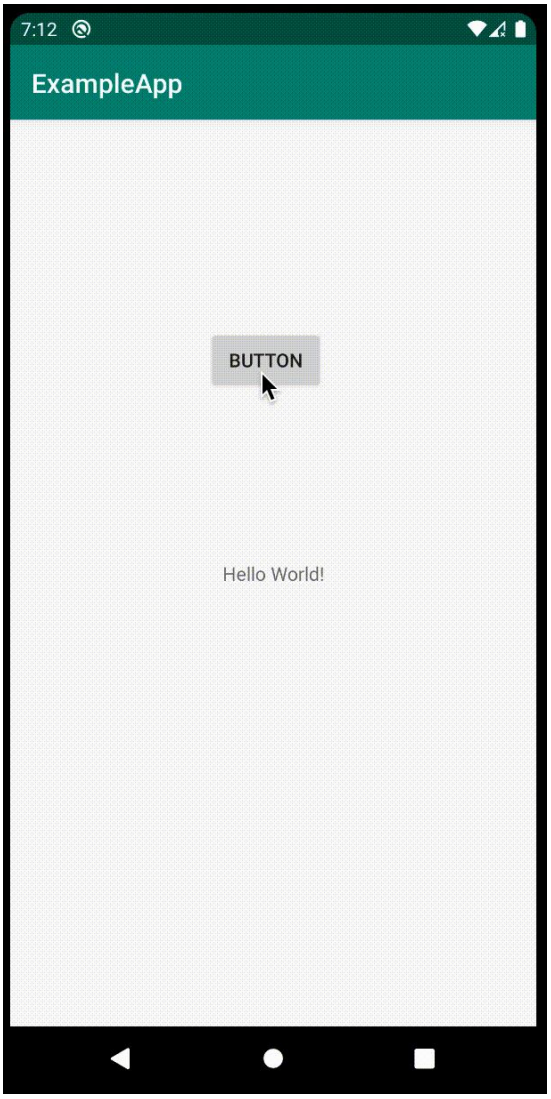
Let's get our hands dirty with code! In this task I want you to add a Button to the activity_main.xml file and make a pop-up message (called Toast on Android) when the user taps on it.

Resources

[Application fundamentals]
[Toasts overview]
[Building a simple user interface]

Submission

Use screen recording software of choice to record a short GIF demonstrating a popup message being shown when you click on the button. Upload it in the Systers Mentorship app. Example below:



5. Introduction to Git and GitHub

Description

Git and GitHub are essential tools when it comes to software development. Although they may intimidate you at first glance, you'll realize soon how helpful they are.

It is not a hard concept to grasp, but it does require some time to fully understand the basic. That's why I'm sending you a bunch of high-quality

articles and videos. Feel free to message me in case you find something to be unclear.

Resources

Difference between Git and GitHub
Git and GitHub tutorial for beginners
Git - installation, configuration and first commit
What is GitHub?

Submission

1. Send me a link to your GitHub profile in the Systers Mentorship app :)
2. Send me the screenshot of the output of running command `git --version` in your terminal

6. Uploading your project to GitHub

Description

As you learned in the previous task, git and GitHub are incredibly useful. In this task, you'll upload your simple project to GitHub to make our collaboration easier.

To do

1. Run `git init` in the root of your project.
2. Create a commit
3. Create a public GitHub repository
4. Add remotes to your local git repository
5. Do `git push` to upload your project to GitHub.

Resources

Create a GitHub repository
Adding an existing project to GitHub using the command line

Submission

Send me a link to the GitHub repository of your project in Mentorship app.

7. Adding second Activity

Description

Activity is the main building block of Android apps. It is exactly what the name suggests – an activity that the user can do. Login, Register, Friends, Feed – all of those features deserve to have a separate activity.

To do

1. Create `SecondActivity`.
2. Create a button in `MainActivity`.
3. Add an on click listener to that button which will open `SecondActivity`.
4. Make a commit and `git push` to GitHub.

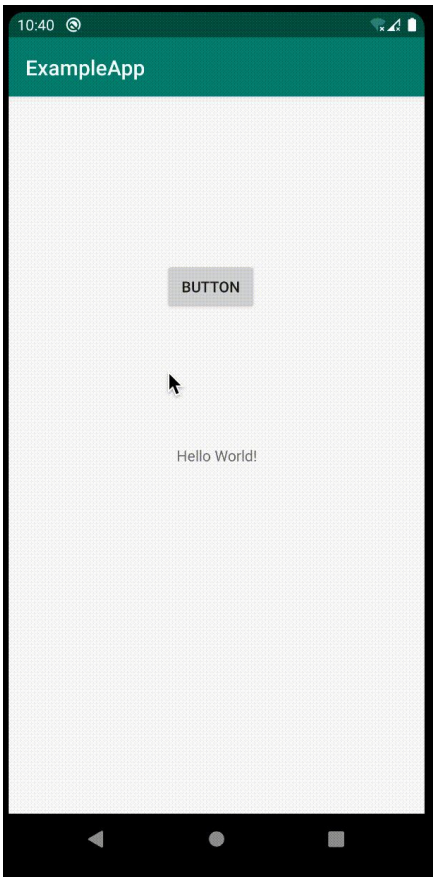
Resources

[Start another Activity on Android Developers]

Submission

Send me a link to the commit containing the changes.

Example



8. Passing data between Activites

Description

Situation when the app asks for user input in Activity1 and does something with that data in Activity2 is a very common is a common and recurring pattern in many Android apps.

In this task, you'll learn how to do this.

To do

1. Create an EditText in MainActivity asking for the user's name.
2. In MainActivity, create a Bundle object and attach it to the intent which starts SecondActivity.
3. Display the user's name in SecondActivity.
4. Make a commit and push it to GitHub.

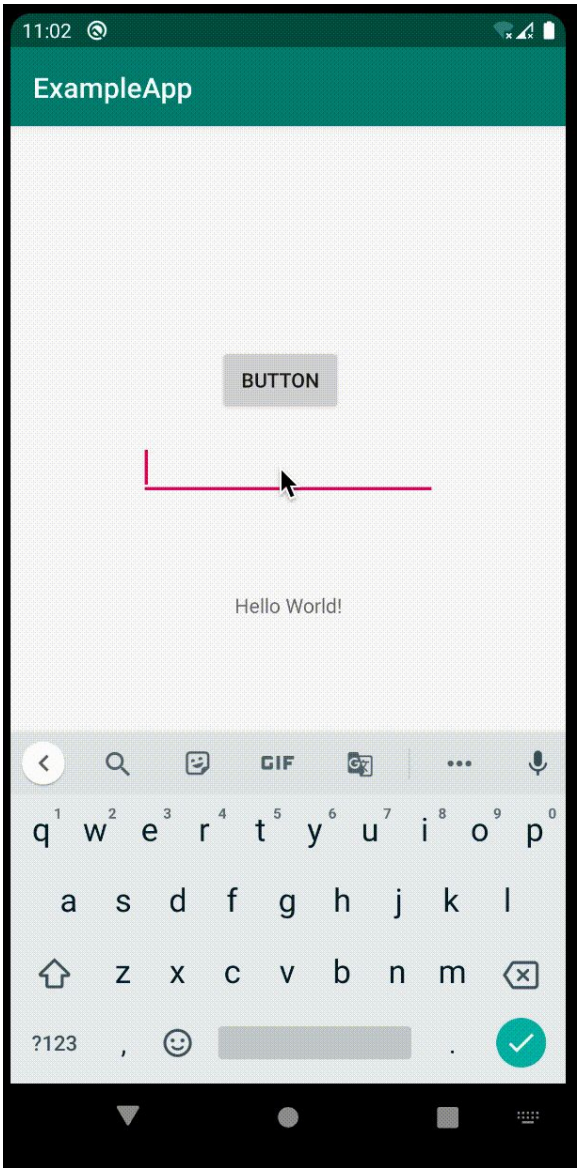
Resources

[Passing data between Activities]

[Passing data between Activities tutorial]

Submission
Send me a link to the commit which contains the required changes.

Example



Wrapping up
Creating those tasks was a very fun and interesting Activity for me :)

I felt like a tutorial creator and I felt responsible for my hypothetical mentee, responsible to find the best way to pass my knowledge to them.