

# A2: Sliding Tiles Puzzle

---

**Due** Oct 12, 2018 by 11:59am      **Points** 10

---

## Sliding Tiles Puzzle App

### Learning objectives

- Use Android Studio to complete a Java application
- Understand the basics of Gradle, the Android SDK, Android emulator input and output and Android studio
- Practice more with Git in preparation for the project

### Getting the starter code

Your repo contains starter code for a sliding tiles puzzle game. Once you have it working, you will see something like this screenshot.



As you might guess, you can slide tiles that are adjacent to the blank area — the 8 can slide down, the 14 can slide right, and so on. The game concludes when a player successfully orders the tiles, with the blank tile at the bottom right. The game can also be paused partway through and saved if needed, and then loaded back from the previous save.

There is also a file called `.gitignore` that contains filename patterns that Git should ignore. This means that it won't be prompting you as much to add files that you shouldn't be adding.

### Installation and setup

Android Studio is installed in the Teaching Labs. If you would like to use your own laptop instead, follow these instructions.

1. [Install Android Studio](https://developer.android.com/studio/) [\(https://developer.android.com/studio/\)](https://developer.android.com/studio/) if you haven't already.
2. Get the A2 URL for your starter code repository from MarkUs and clone it using Android Studio.

Please be careful about the exact process!

1. Answer "Yes" to whether you want to create a new Android Studio project.
2. Select "Import project from external model" and click Next:

gradleimport.png

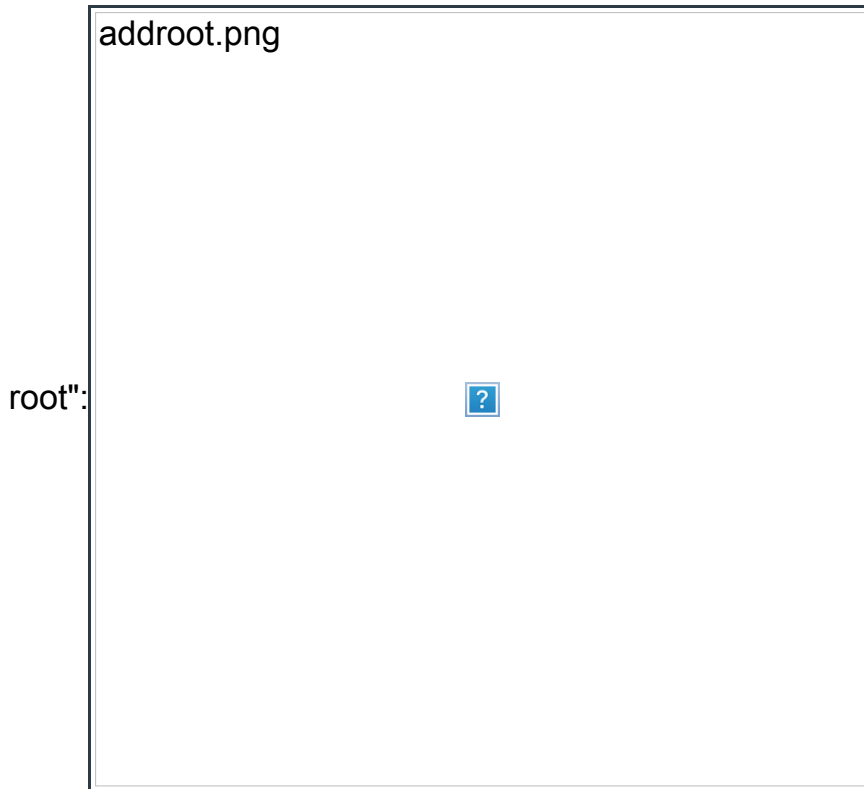


3. Add "A2" to the Gradle project path:

importproject.png



4. You might get the message "Unregistered VS root detected", and you should choose to "Add



3. The project should now build.
4. Create an Android Virtual Device within Android Studio. Select a Pixel2 smartphone as the device to emulate, specifying the device OS as Android 8.1 API 27.
5. Launch the virtual device and ensure it loads correctly.
6. Now run it, and you should see the puzzle. Nothing will happen when you click, because you need to fix the program.

## A few words about Android

In Android, an *Activity* is a screen in your application. The sliding tiles puzzle app has two Activities. One of them, *StartingActivity*, contains an *onCreate* method that is the entry point to your code. This is roughly comparable to method *main*.

Android Applications have many components, and each plays a pivotal role in ensuring the overall application works as intended.

- **Gradle:** The main build tool for Android applications. Gradle is like a programming language for building software applications. A typical Gradle script builds, compiles and links your application code, the layout files and the specific precompiled libraries and dependancies specified into a installable application. There are a number of scripts found in the Gradle Scripts directory of the project. **DO NOT EDIT THESE FILES.** They specify which version of Gradle to use for the application, dependencies and version numbers for things like the Android SDK, Android OS

version and JUnit version. Your code must match the versions provided in the starter code for it to be marked correctly.

- **Manifests:** The `AndroidManifest.xml` file contains information specific to your application. It allows you to specify an overall visual theme for the application, and list the activities/views you would like to have available to the application, and which one the application should start with.
- **Res:** The `Res` folder contains subfolders that specify resources used by your application. Here you will find the drawable folder, containing the images of each of the 16 tiles used in the game. The layout folder contains the layout xml files for each activity/view in the application.
- **Layouts:** Each Activity has its own XML layout file. Here individual components such as buttons, headers, panels and anything else displayed in the view can be shown.
- **Application Code:** The code is in the `java` folder. The main logic of the game, along with the controller and model code, live here. This is the section of the App that you will need to modify to complete this assignment.
- **Emulator:** The Android Virtual Environment emulates an Android device on your computer to safely test your application in a controlled setting. In order to run and test your application you will need to configure a virtual device to the required settings listed below under the setup instructions. We will use a virtual device with identical settings to ensure your app runs as required.

Please ensure that you do not modify anything outside the `java` folder for this assignment.

## A tour of the code

We recommend that you read the code in the following order. We have marked your tasks as TODO items directly in the code.

- **Class Board:** It keeps track of Tile objects in a 2D array. The constructor is given a 1D List of Tiles, and these are used to populate the 2D array. This class is also Observable, which means that other classes can sign up to be alerted when the contents change. This happens, as you'll see, when `swapTiles` is called. The call to `notifyAll` ends up letting the `GameActivity` know that the user interface needs to be updated.
  - Missing/incomplete: `numTiles`, `swapTiles`. Also, make this class implement `Iterable<Tile>`.
- **Class Tile:** this one is a bit weird. Board makes use of it, of course.
- **Class BoardManager:** BoardManager manipulates the Board, figuring out whether a tap is legal,

checking whether the puzzle has been solved, and so on.

- Missing/incomplete: `puzzleSolved`, `touchMove`.
- **GameActivity**: This is the main sliding tile puzzle. It uses a `BoardManager` to keep track of the board, and also manages all the user interface components.

We'll go over the user interface part in lecture.

## Submitting

As with A1, you will submit by pushing your changes.