

Developer Guide

The following serves as a guide to future developers working on these apps. This is meant to be a **continuous** guide, meaning that every developer should add or make changes to this guide to help better serve future developers. This guide is by no means a thorough explanation of Unity, C# scripts, or GoogleARCore/Apple's ARKit. Rather, this guide specifically looks at these projects and serves to help developers as they navigate it.

Installing JDK and Android Studio

Currently, the floor plan app and ARSHM run best on an Android device. This requires the appropriate JDK, as well as an installation of Android Studio.

The JDK must be downloaded first! Otherwise, you run into issues.

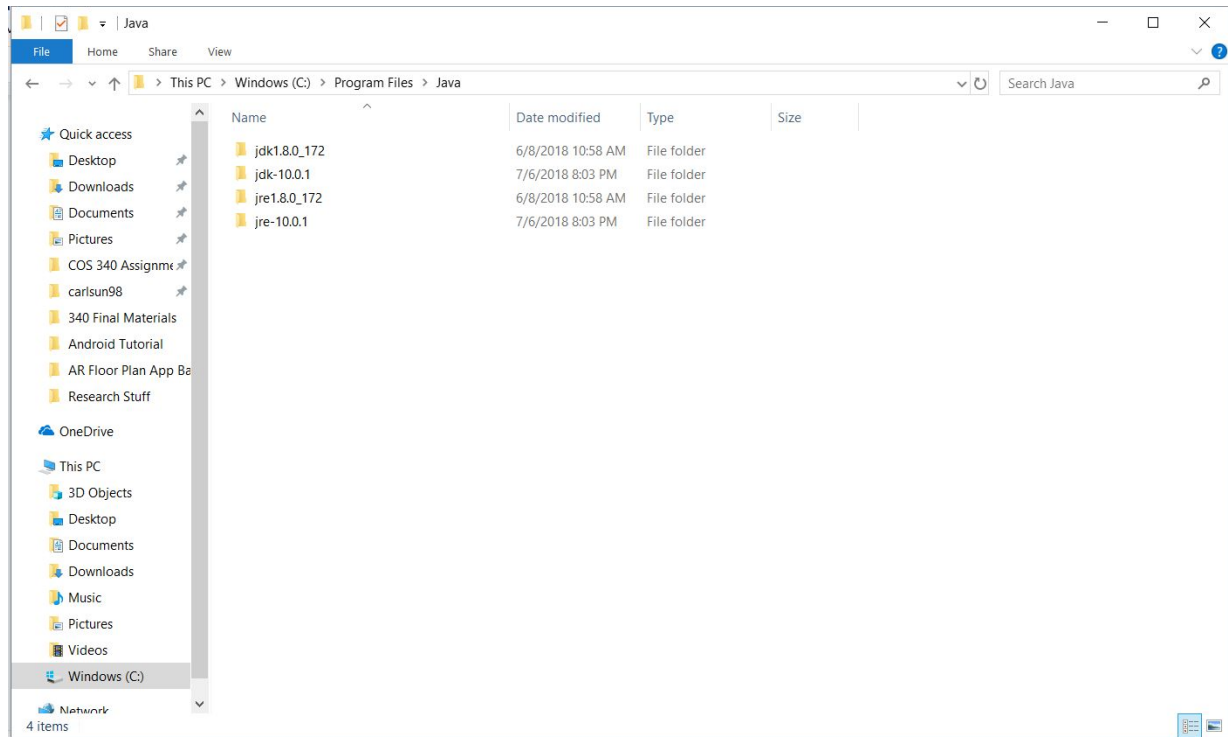
The JDK must be version 8. Otherwise, you run into issues.

The download link for JDK Version 8 Update 181 can be found here:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Simply click on the .exe and follow through the installation.

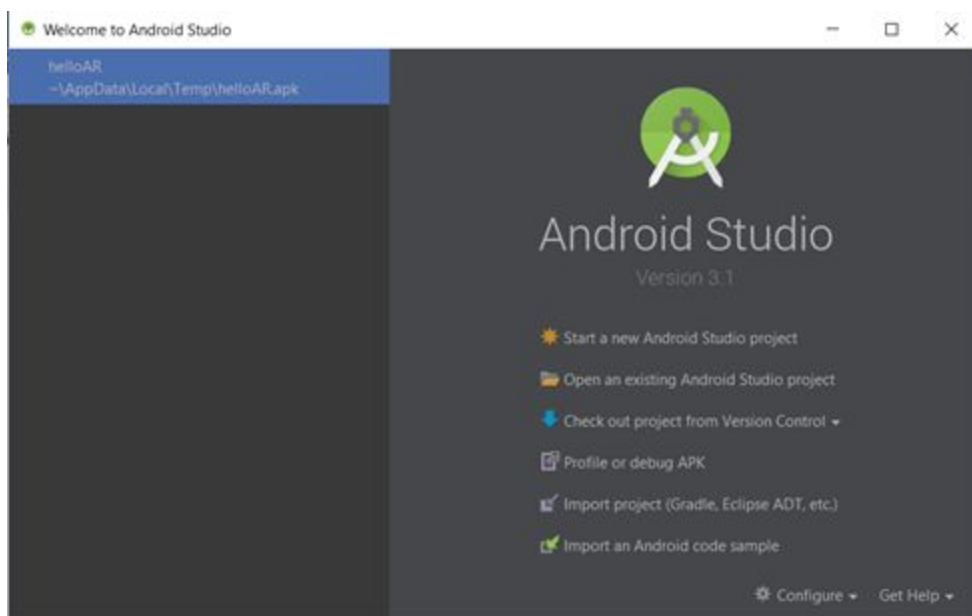
Once downloaded, you can double check that you have the JDK and JRE through your file system explorer. In Windows, the folders should be along this path:



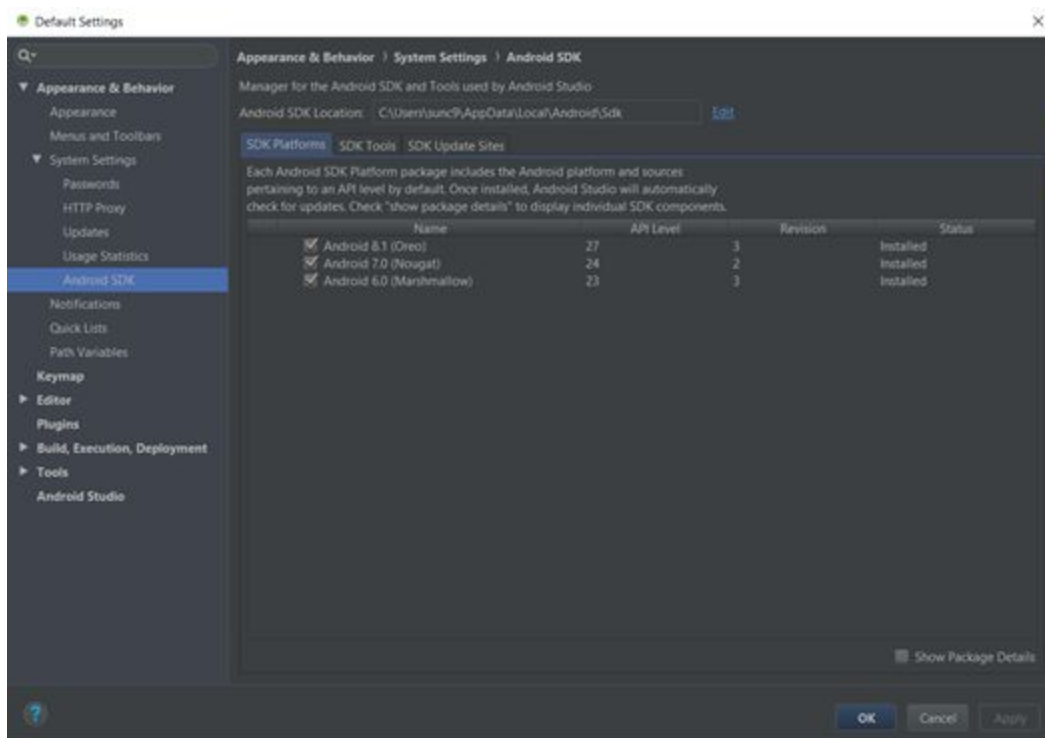
Next, Android Studio. The following link brings you to the Android Studio installation page:

<https://developer.android.com/studio/>

Once you've gone through the installation process, open up Android Studio. You should see this:

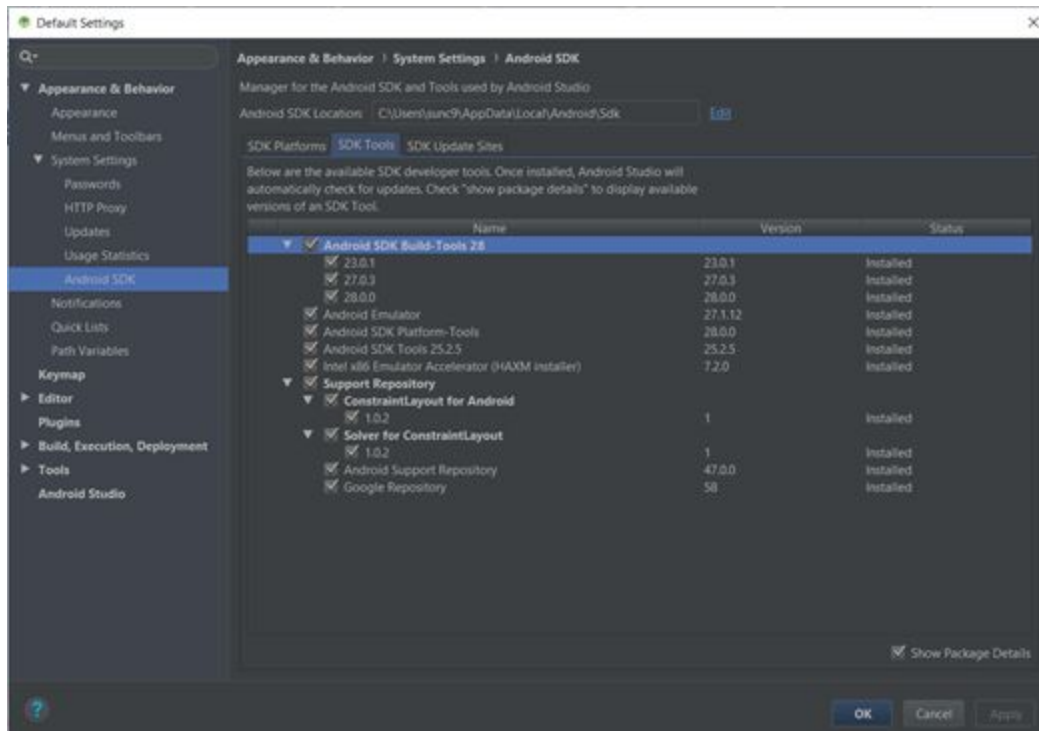


Go to Configure -> SDK Manager. There you should see the SDK platforms available for download.



Make sure you download 8.1, 7.0, and 6.0 by selecting the check boxes next to them. Deselect any other ones.

Next, go to the SDK Tools tab. Click on “Show Package Details” in the bottom right. The content should change to something similar to this:



Check off 23.0.1, 27.0.3, and 28.0.0 under Android SDK Build-Tools 28. The reason you need these is because the minimum SDK of the project is 25 (so the 23.0.1 most likely isn't necessary, but just in case). Make sure to check off the other items listed in the picture above. Anything else is extraneous, so either deselect it or leave it be. Apply these changes, and then close out.

Next, you need to check that your PATH variables are set up correctly. Visit this link:

<http://www.automationtestinghub.com/setup-android-environment-variables/>

This will guide you through adjusting your environmental variables so that the JDK can be reached. I worked through this part with Becca, so hopefully she will be able to assist you if need be.

Installing Unity

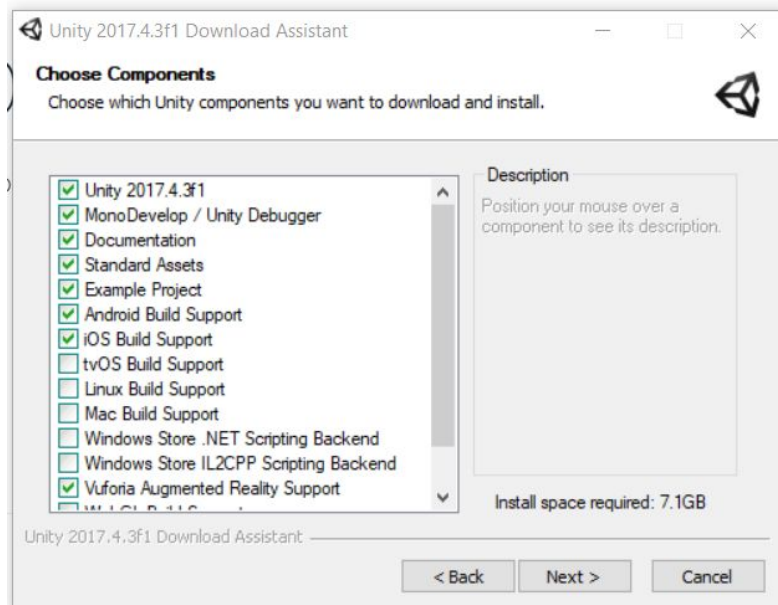
To install Unity, go to this link:

https://unity3d.com/get-unity/download/archive?_ga=2.7663922.1414526224.1532614159-1155665432.1528226331

This should lead you to a page containing older versions of Unity. The floor plan app and ARSHM app were written in Unity 2017.4. Version 4.3 should be a stable download for opening

and working on these apps. Attempting to open these apps in Unity 2018 results in errors that I have yet to resolve, so this is one of the challenges you can work on.

As you go through the installation procedure, you will eventually find this window:



The components with check marks are ones you need to install, especially the iOS and Android Build Supports, as these allow you to test the app on iOS or Android devices (testing on iOS still requires a Mac however, since the app needs to be run through XCode, and testing on Android requires Android Studio).

Downloading Our Projects

The following links are to my and Zach's GitHubs:

<https://github.com/carlsun98/ARFloorPlanApp>

<https://github.com/zacharyliu/arshm-release>

Download our projects as ZIP files, and extract them (using 7-Zip or whatever file extraction program you may have) to the Documents part of your computer (unclear if extracting to Downloads caused problems). You should then have two folders, each containing a project.

Now open Unity. In the window, there should be an "Open" option in the top right, and through there you can open up my project and Zach's project.

To build our projects, take a look at the readme on my GitHub.

To set up the server for Zach's project, take a look at the readme on his Github.

I attempted to comment both my code and Zach's code to the best of my understanding, and I hope you can draw some help from them.

While Becca probably went over with you what tasks need to be done, here's my current list to help you stay focused:

- Floor plan app
 - Find a way for the overhead camera to follow the user's location. My idea was to spawn in and continuously update the position of a GameObject representing the user, and having an overhead camera rendering to the Minimap Render Texture follow it.
 - Sub-problem: Fix the issue of the camera changing the object it centers on to the last one placed.
 - Figure out why the minimap does not display the line connecting the last and first spheres when you tap the first sphere. All other lines are displayed.
 - Figure out how to display the measurements on the minimap. This is mildly optional, since the display may be very small on the minimap.
- Integrate the floor plan app and ARSHM together. This will be difficult, as there already exists a point cloud in Zach's app, so piggy backing off of it could cause problems. Alternatively, it may work better.
- Add buttons to Zach's app. Becca can tell you more about this. Locations of the buttons and relevant scripts are listed below.

Buttons: In the hierarchy, ARSHM -> UI -> ControlCanvas -> Toolbox -> Button Layout contains the Anchor, Surface, and Draw buttons. Toolbox also contains other buttons that pop up after pressing Anchor, Surface, and Draw. ControlCanvas -> Capture is the capture button on the lower right corner.

Associated Script: The Toolbox Controller Script manages the buttons. Found in UI -> ControlCanvas -> Toolbox

How to Recreate the Floor Plan App

If you feel inclined, you can recreate the floor plan app relatively easily. It is based off of GoogleARCore's sample project. Here are the steps:

1. Visit the following link:

<https://developers.google.com/ar/develop/unity/quickstart-android>

Set up the project according to the guide.

2. Click on the Example Controller and locate the Andy value next to "Andy Android Prefab" on the right hand side. Click on this to locate the Andy object in Project. Drag and drop it into the Scene tab, and press F to focus on it.
3. Andy is the object that is spawned when the user taps on the plane, so let's change the object that is spawned. At the top, go to GameObject -> 3D Object -> Sphere. Adjust the scale to 0.1 in X, Y, and Z. Delete Andy from the scene.
4. Right click in your Project files and go Create -> Material. Name the material something (Point Color is what I used), and change the shader of the material (on the right hand side when the material is selected, towards the top) from Standard to Unlit -> Color. Change the color to whatever you want your points to be. Then, select the sphere in the Hierarchy and drag + drop the material from the Project files onto the Mesh Renderer -> Materials -> Element 0 slot.
5. While you have the sphere selected, click on "Add Component" (on the right hand side towards the bottom) and search up "Line Renderer". Add it to the sphere. In the line renderer component, change the Width property (just below Loop) to 0.09. Change the color of the line by doing the same process in step 4 (Create a new material, change the shader, select the sphere and drag + drop the material onto Line Renderer -> Materials -> Element 0). Set the Positions -> Size to 0. Drag + drop the Sphere in the Hierarchy onto the Andy prefab in the Project files. This will replace Andy with the sphere object. You can then delete the sphere from the Hierarchy.
6. Go to Example Controller and click on the gear on the right hand side in "Hello AR Controller (Script)." Click Edit Script. This lets you edit the code in the script. Check the code in my project to see what you need to add.

7. Once that's done, you need the text object. Go back to the Unity editor, and go to the GameObject tab at the top -> 3D Object -> 3D text. Select it in the Hierarchy, and change the Scale to 0.005 for X, Y, and Z. Then change the Text Mesh -> Font Size to about 100 (you can change this stuff around after you've gone through this to see what sizes work best). Then set the Text Mesh -> Anchor to Middle Center. Drag + drop it into the Project files to make it a prefab, and rename it to "Text".
8. Select the Example Controller. Drag + drop the Text prefab from the Project files onto the value next to "Text"
9. Go to File -> Build Settings. Press the "Add Open Scenes" button and then click "Build and Run" (make sure you've done the steps in the ARCore tutorial!)
10. When you're testing it on your phone, you can move around to generate a plane, and then click anywhere on the plane to place a point.