Lappeenrannan teknillinen yliopisto

School of Software Engineering

Sofware Development Skills

<Hyoseo Jang>, <000719647>

LEARNING DIARY, <Software Development Skills: Mobile> MODULE

**16/01/2022**

Today, I learned how to start up the first Android studio project. For the first lecture, I got to know that the purpose of this course is to develop my project and learn about Android studio. In this lecture, I tried to make my environment, but because of the version, I struggled hard time. Also, I learned about the components of Android studio and use it.

**23/01/2022**

I learned the core elements of Android studio, and develop the project. I learned about Broadcast receiver and I faced errors of null point exception at that time. Finally, I found that I need to include localbroadcastmanager in the code. While handling the unexpected errors, I found that debugging is a necessary part of coding. After that, I continued to watch the second part till the end.

**30/01/2022**

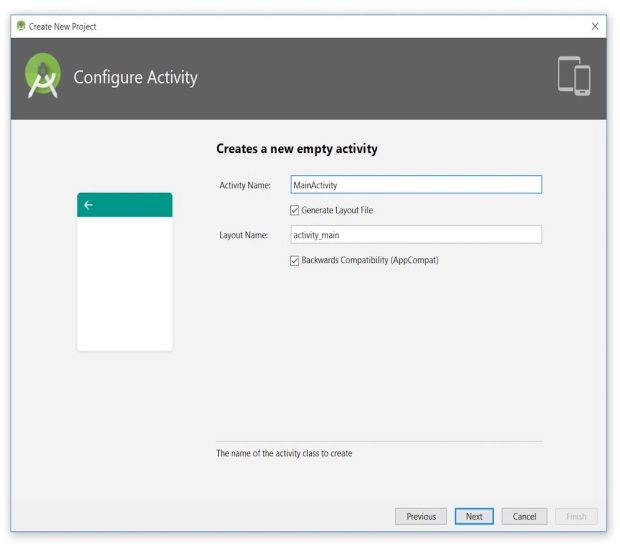
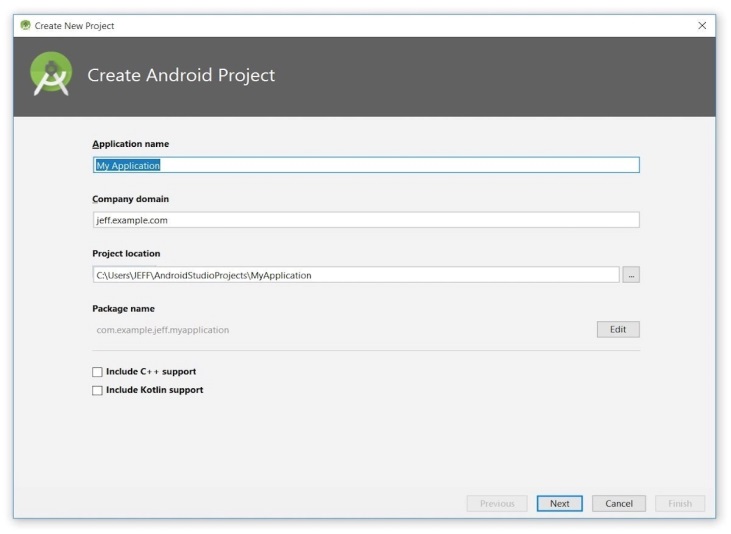
Today, I learned about ‘list view’ and it is a type of a View Group. To display data that users define, I need to use Adapter. And in the third part, I got to know how to use Adapter View, including ListView, ExpandableListView, GridView, Spinner, Gallery. Except for this lecture, I tried to make other examples using list view, and it was helpful to develop the skills of java.

**<Android Studio For Beginners Part 1>**

I started up my first Android project and got to know Android Studio's main window.

**Starting a new project**

I clicked Start a new Android Studio project. Android Studio responded with the Create New Project dialog box.



1) **manifests** stores AndroidManifest.xml, which is an XML file that describes the structure of an Android app. This file also records permission settings (where applicable) and other details about the app.

2) **java** stores an app's Java source files according to a package hierarchy, which is ca.javajeff.w2a in this example. It also organizes files for testing purposes.

3) **res** stores an app's resource files, which are organized into drawable, layout, mipmap, and values subbranches:

4) **drawable** is a mostly empty location in which to store an app's artwork; initially, the XML files for the launcher foreground and background adaptive icons are stored here.

5) **layout** is a location containing an app's layout files; main.xml (the main activity's layout file) is initially stored here.

6) **mipmap** is a location containing various ic\_launcher.png files, which store launcher screen icons of different resolutions.

7) **values** is a location containing colors.xml, strings.xml, and styles.xml.

8) **The Gradle Scripts** branch identifies various .gradle (such as build.gradle) and .properties (such as local.properties) files that are used by Android Studio's Gradle-based build system.

**23/01/2022**

Today, I learned how to start up the first Android studio project. For the first lecture, I got to know that the purpose of this course is to develop my project and learn about Android studio. In this lecture, I tried to make my environment, but because of the version, I struggled hard time. Also, I learned about the components of Android studio and use it.

**<Android Studio For Beginners Part 2>**



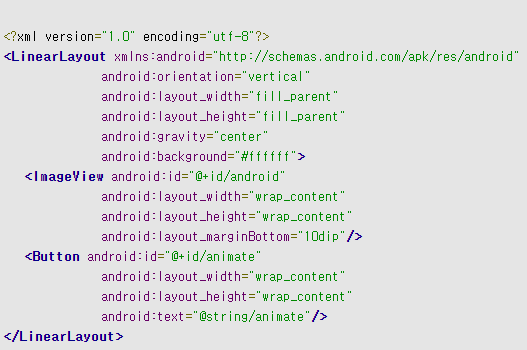
The W2A.java file starts with a package statement, which names the package (ca.javajeff.w2a) that stores the W2A class.

**The onCreate() method**

1) **R** is the name of a class that's generated when the app is being built. This class is named R because its content identifies various kinds of application resources, including layouts, images, strings, and colors.

2) **layout** is the name of a class that's nested within R. An application resource whose ID is stored in this class describes a specific layout resource. Each kind of application resource is associated with a nested class that's named in a similar fashion. For example, string identifies string resources.

3) **main** is the name of an int-based constant declared within layout. This resource ID identifies the main layout resource. Specifically, main refers to a main.xml file that stores the main activity's layout information. main is W2A's only layout resource.



1) **orientation** identifies the linear layout as horizontal or vertical. Contained widgets are laid out horizontally or vertically, and the default orientation is horizontal. "horizontal" and "vertical" are the only legal values that can be assigned to this attribute.

2) **layout\_width** identifies the width of the layout. Legal values include "fill\_parent" (to be as wide as the parent) and "wrap\_content" (to be wide enough to enclose content). (Note that fill\_parent was renamed to match\_parent in Android 2.2, but is still supported and widely used.)

3) **layout\_height** identifies the height of the layout. Legal values include "fill\_parent" (to be as tall as the parent) and "wrap\_content" (to be tall enough to enclose content).

4) **gravity** identifies how the layout is positioned relative to the screen. For example, "center" specifies that the layout should be centered horizontally and vertically on the screen.

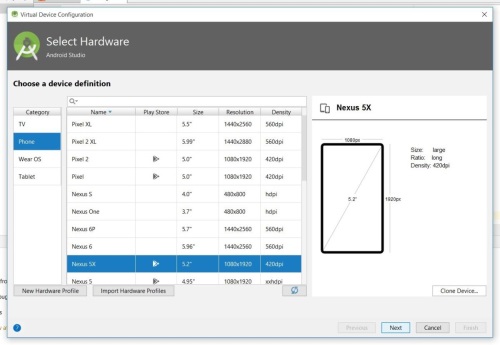
5) **background** identifies a background image, a gradient, or a solid color. For simplicity, I've hardcoded a hexadecimal color identifier to signify a solid white background (#ffffff). (Colors would normally be stored in colors.xml and referenced from this file.)

**23/01/2022**

Today, I learned how to start up the first Android studio project. For the first lecture, I got to know that the purpose of this course is to develop my project and learn about Android studio. In this lecture, I tried to make my environment, but because of the version, I struggled hard time. Also, I learned about the components of Android studio and use it.

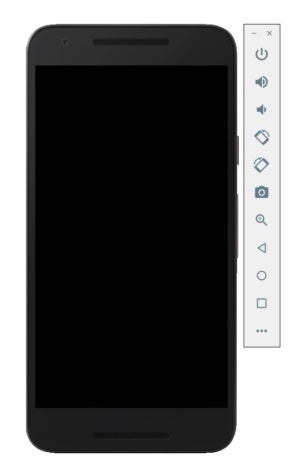
**<Android Studio For Beginners Part 3>**

**Building your Android app**

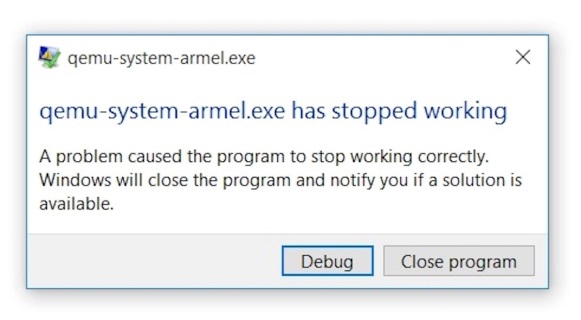


I selected whatever device you want to emulate. For this example, I selected the highlighted (default) Nexus 5X.

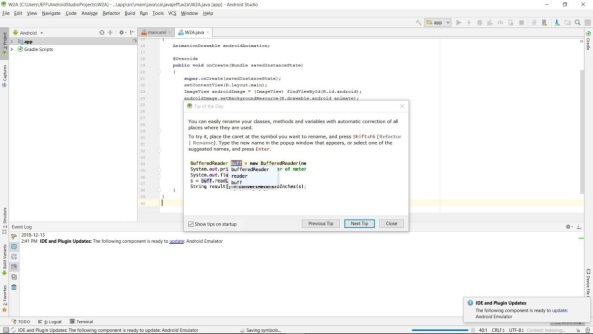
The resulting System Image panel invites me to select a system image for this device emulation. I clicked the Other Images tab followed by IceCreamSandwich.



Shortly after the emulator window appeared, the Windows-based qemu-system-armel.exe program that's responsible for creating this window crashed.



At first, I wasn't sure how to fix this problem. Fortunately, the next time I ran Android Studio 3.2.1, I observed an **IDE and Plugin Updates** message in the lower-right area of Android Studio's main window.

[[](https://images.idgesg.net/images/article/2019/02/figure12-100788050-orig.jpg?auto=webp&quality=85,70)](https://images.idgesg.net/images/article/2019/02/figure12-100788050-orig.jpg?auto=webp&quality=85,70" \o "<div class='credit'>Jeff Friesen</div> <p>Figure 12. Click the Update link and follow directions to update the Android emulator (click to enlarge)</p> )

I clicked the **update** link and followed directions to update the emulator, which resulted in a new qemu-system-armel.exe file. No more crashes, but I quickly hit another snag.