Setting up Sudoku example on Android Studio

Installing Android Studio

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

If you didn't download Android Studio, go **download Android Studio now**.

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher. To check if you have JDK installed (and which version), open a terminal and type `javac -version`. If the JDK is not available or the version is lower than 6, go download JDK.

To set up Android Studio on Windows:

1. Launch the .exe file you just downloaded.
2. Follow the setup wizard to install Android Studio and any necessary SDK tools.

   On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

   Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable `JAVA_HOME` that points to your JDK folder, for example `C:\Program Files\Java\jdk1.7.0_21`.

The individual tools and other SDK packages are saved outside the Android Studio application directory. If you need to access the tools directly, use a terminal to navigate to the location where they are installed. For example:

```
\Users\<user>\sdk\
```

To set up Android Studio on Mac OSX:

1. Launch the `.dmg` file you just downloaded.
2. Drag and drop Android Studio into the Applications folder.
3. Open Android Studio and follow the setup wizard to install any necessary SDK tools.

   Depending on your security settings, when you attempt to open Android Studio, you might see a warning that says the package is damaged and should be moved to the trash. If this happens, go to **System Preferences > Security &**

Privacy and under **Allow applications downloaded from**, select **anywhere**. Then open Android Studio again.

If you need use the Android SDK tools from a command line, you can access them at:

```
/Users/<user>/Library/Android/sdk/
```

To set up Android Studio on Linux:

1. Unpack the downloaded ZIP file into an appropriate location for your applications.
2. To launch Android Studio, navigate to the `android-studio/bin/` directory in a terminal and execute `studio.sh`.

   You may want to add `android-studio/bin/` to your PATH environmental variable so that you can start Android Studio from any directory.

3. Follow the setup wizard to install any necessary SDK tools.

Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

Adding SDK Packages

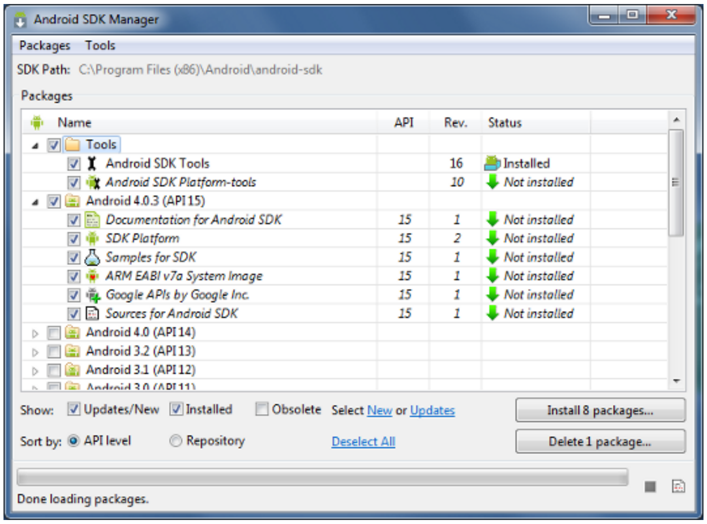
By default, the Android SDK does not include everything you need to start developing. The SDK separates tools, platforms, and other components into packages you can download as needed using the Android SDK Manager. So before you can start, there are a few packages you should add to your Android SDK.

To start adding packages, launch the Android SDK Manager in one of the following ways:

- In Android Studio, click **SDK Manager** in the toolbar.
- If you're not using Android Studio:
  - Windows: Double-click the `SDK Manager.exe` file at the root of the Android SDK directory.
  - Mac/Linux: Open a terminal and navigate to the `tools/` directory in the Android SDK, then execute `android sdk`.

When you open the SDK Manager for the first time, several packages are selected by default. Leave these selected, but be sure you have everything you need to get started by following these steps:

1. Get the latest SDK tools



SDK Path: C:\Program Files (x86)\Android\android-sdk

Name	API	Rev.	Status
☑ Tools			
☑ Android SDK Tools		16	Installed
☑ *Android SDK Platform-tools*		*10*	*Not installed*
☑ Android 4.0.3 (API 15)			
☑ *Documentation for Android SDK*	*15*	*1*	*Not installed*
☑ *SDK Platform*	*15*	*2*	*Not installed*
☑ *Samples for SDK*	*15*	*1*	*Not installed*
☑ *ARM EABI v7a System Image*	*15*	*1*	*Not installed*
☑ *Google APIs by Google Inc.*	*15*	*1*	*Not installed*
☑ *Sources for Android SDK*	*15*	*1*	*Not installed*
□ Android 4.0 (API 14)			
□ Android 3.2 (API 13)			
□ Android 3.1 (API 12)			
□ Android 3.0 (API 11)			

Show: ☑ Updates/New ☑ Installed □ Obsolete

Sort by: ● API level ○ Repository

Done loading packages.

As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:

1. Open the Tools directory and select:
   - **Android SDK Tools**
   - **Android SDK Platform-tools**
   - **Android SDK Build-tools** (highest version)
2. Open the first Android X.X folder (the latest version) and select:
   - **SDK Platform**
   - A system image for the emulator, such as **ARM EABI v7a System Image**

2. Install the packages

Once you've selected all the desired packages, continue to install:

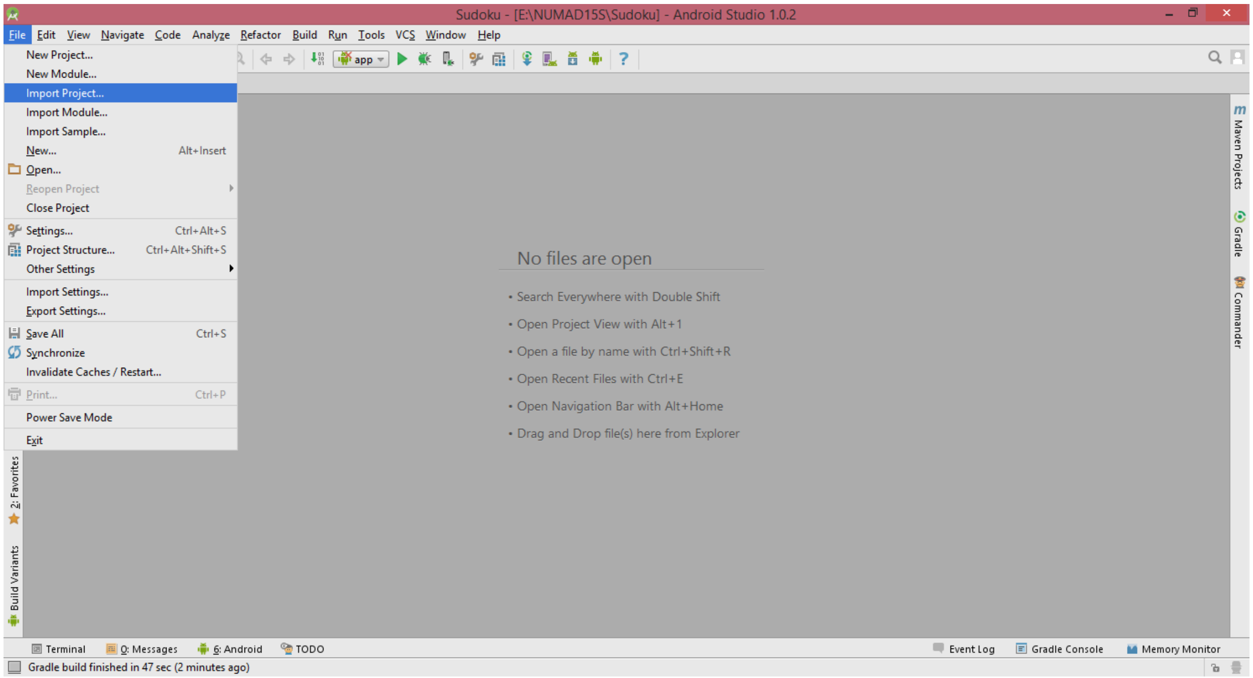
1. Click **Install X packages**.
2. In the next window, double-click each package name on the left to accept the license agreement for each.
3. Click **Install**.

The download progress is shown at the bottom of the SDK Manager window. **Do not exit the SDK Manager** or it will cancel the download.

Note: *You should ideally download the latest version against which you compile your application and the version which you intend to target the application. In this case - Android 5.0.1 (API 21) and Android 2.3.3 (API 10) respectively.*

Importing Sudoku code

1. In Android Studio, click on **Import Project** from the **File** menu.

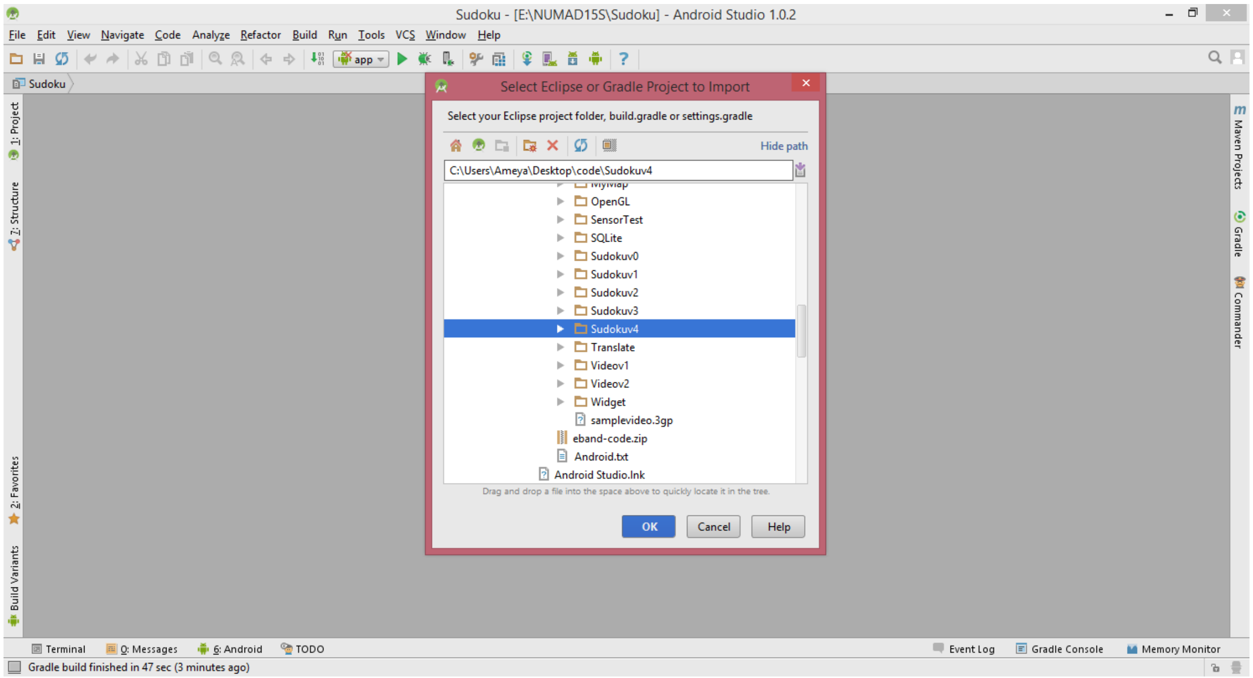


No files are open

- Search Everywhere with Double Shift
- Open Project View with Alt+1
- Open a file by name with Ctrl+Shift+R
- Open Recent Files with Ctrl+E
- Open Navigation Bar with Alt+Home
- Drag and Drop file(s) here from Explorer

Gradle build finished in 47 sec (2 minutes ago)

2. Browse to the directory where you extracted the code and select **Sudokuv4**



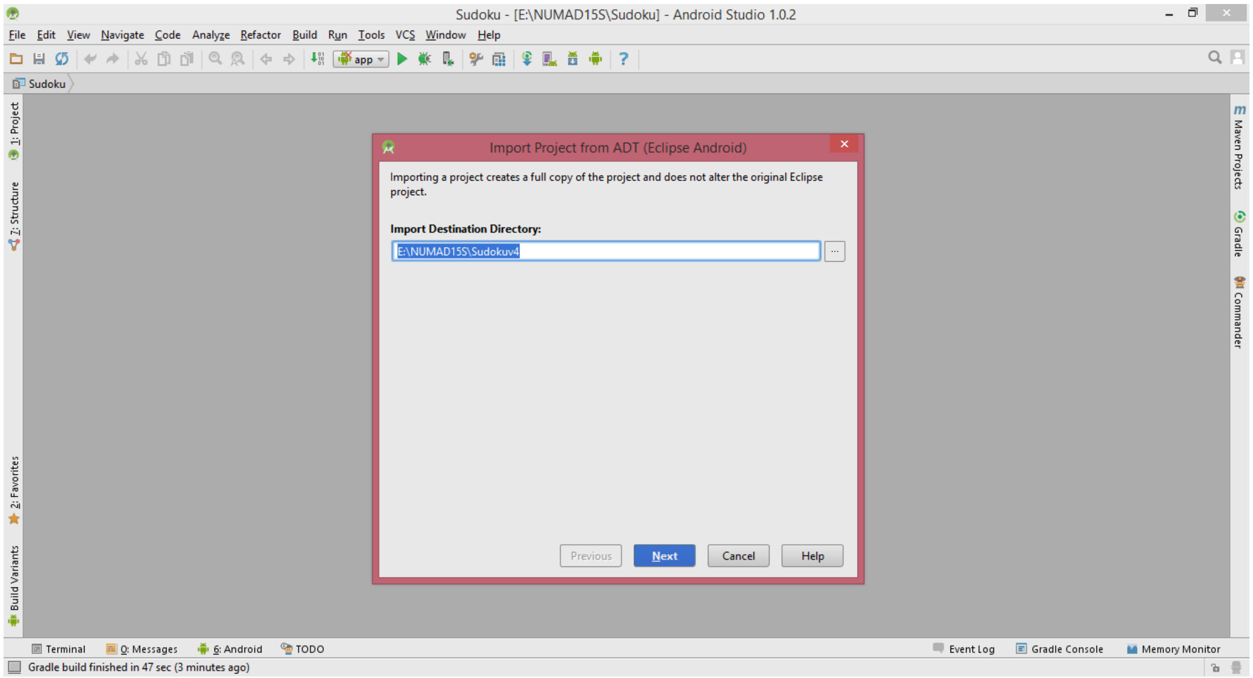
Select your Eclipse project folder, build.gradle or settings.gradle

C:\Users\Ameya\Desktop\code\Sudokuv4

Drag and drop a file into the space above to quickly locate it in the tree.

Gradle build finished in 47 sec (3 minutes ago)

3. It will recognize the Project from ADT. Choose a destination directory for the import.



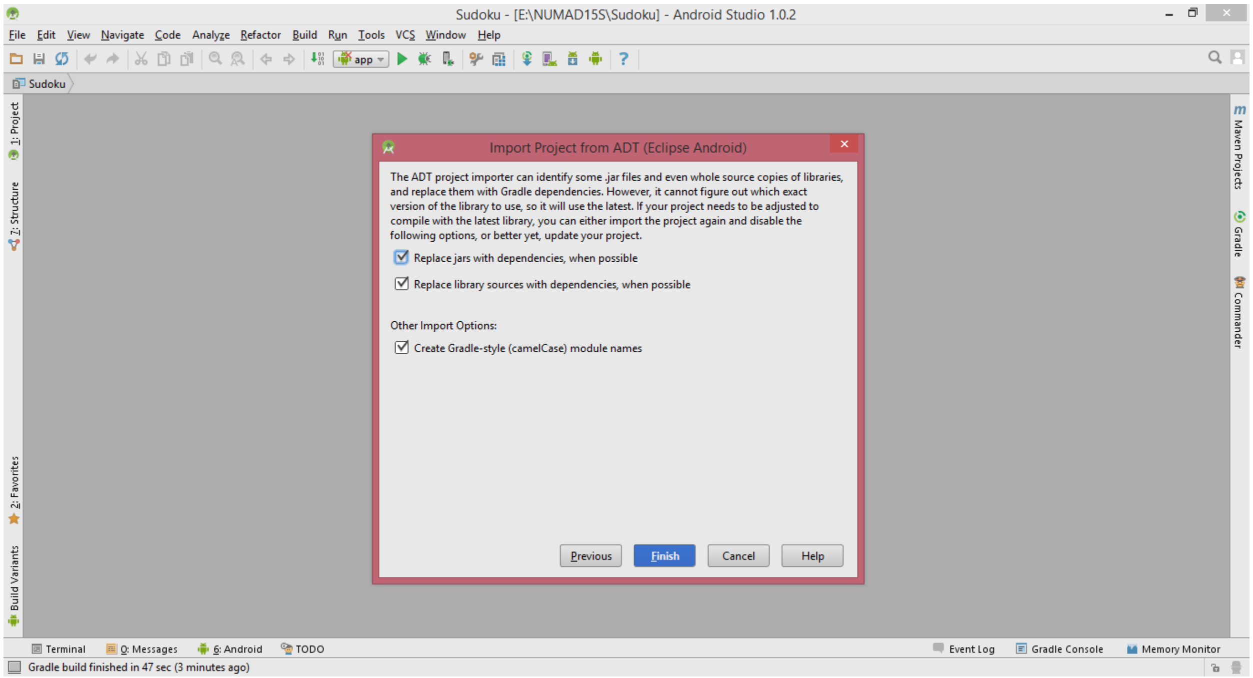
Importing a project creates a full copy of the project and does not alter the original Eclipse project.

Import Destination Directory:

E:\NUMAD15S\Sudokuv4

Gradle build finished in 47 sec (3 minutes ago)

4. Click on the Finish button.



The ADT project importer can identify some .jar files and even whole source copies of libraries, and replace them with Gradle dependencies. However, it cannot figure out which exact version of the library to use, so it will use the latest. If your project needs to be adjusted to compile with the latest library, you can either import the project again and disable the following options, or better yet, update your project.

☑ Replace jars with dependencies, when possible

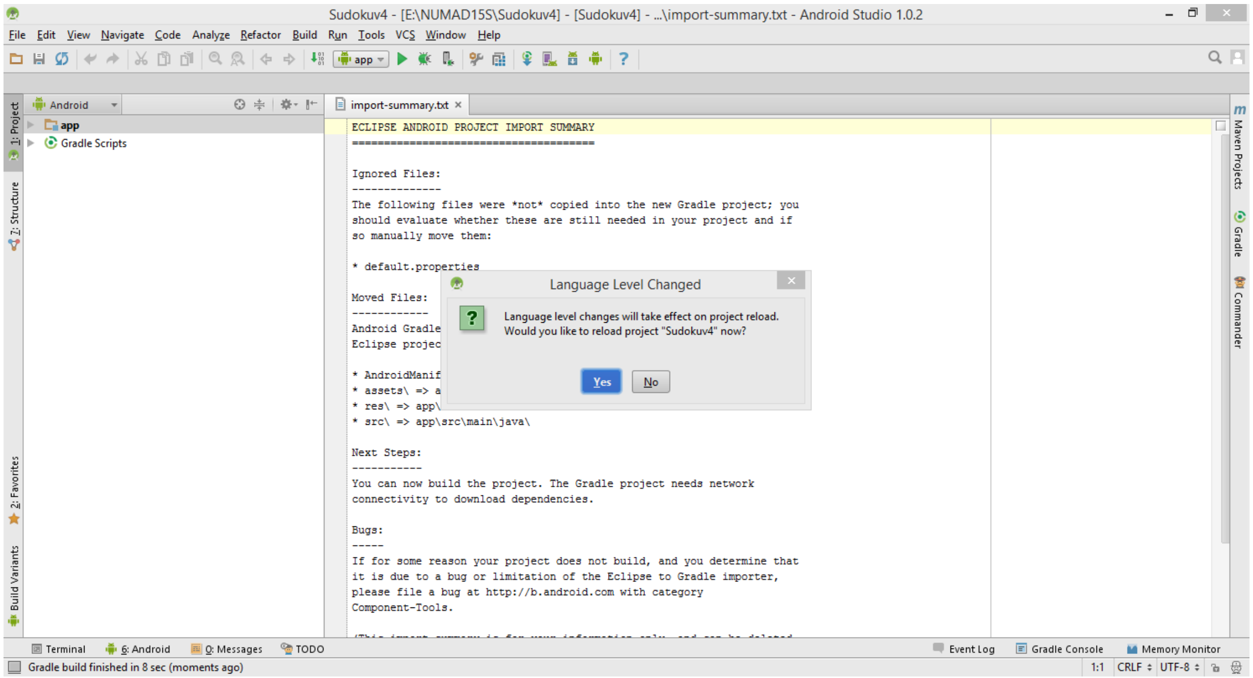
☑ Replace library sources with dependencies, when possible

Other Import Options:

☑ Create Gradle-style (camelCase) module names

Gradle build finished in 47 sec (3 minutes ago)

5. After importing the project, the configurations will be modified from ADT Eclipse project to Android Gradle as Android Studio uses Gradle build.



```
ECLIPSE ANDROID PROJECT IMPORT SUMMARY
======================================

Ignored Files:
--------------
The following files were *not* copied into the new Gradle project; you
should evaluate whether these are still needed in your project and if
so manually move them:

* default.properties

Moved Files:
------------
Android Gradle
Eclipse projec

* AndroidManif
* assets\ => a
* res\ => app\
* src\ => app\src\main\java\

Next Steps:
-----------
You can now build the project. The Gradle project needs network
connectivity to download dependencies.

Bugs:
-----
If for some reason your project does not build, and you determine that
it is due to a bug or limitation of the Eclipse to Gradle importer,
please file a bug at http://b.android.com with category
Component-Tools.

(This import summary is for your information only, and can be deleted
```

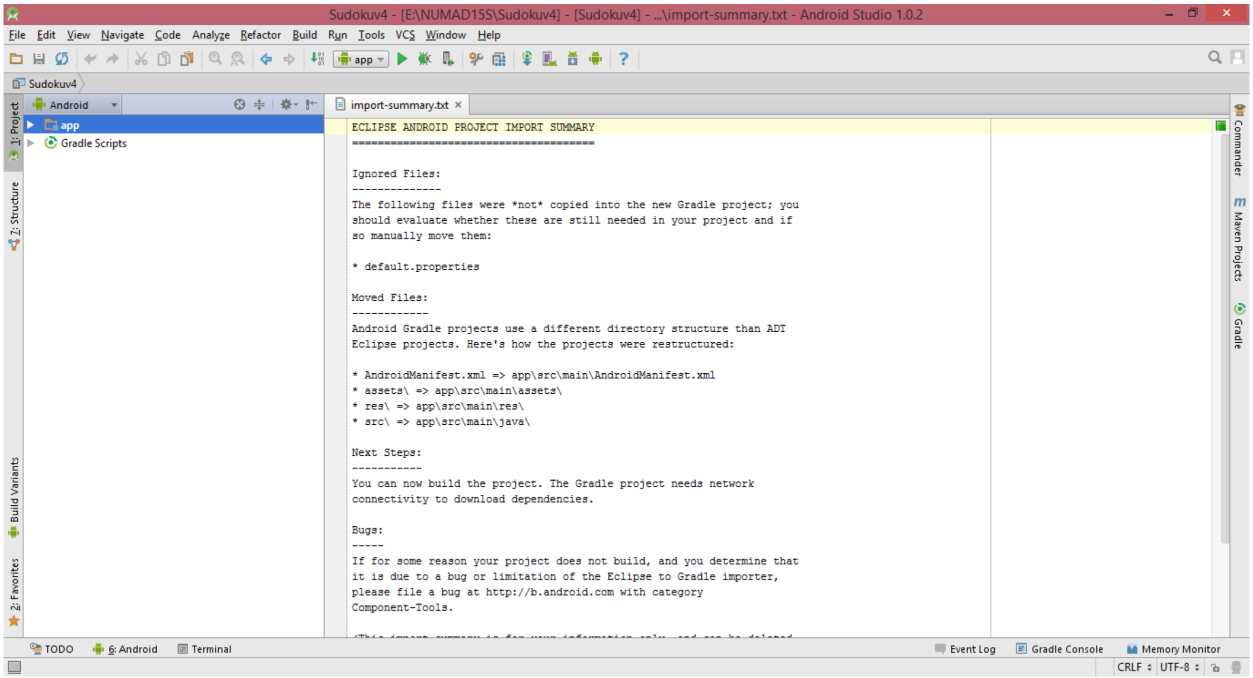
Language Level Changed

Language level changes will take effect on project reload.
Would you like to reload project "Sudokuv4" now?

Yes No

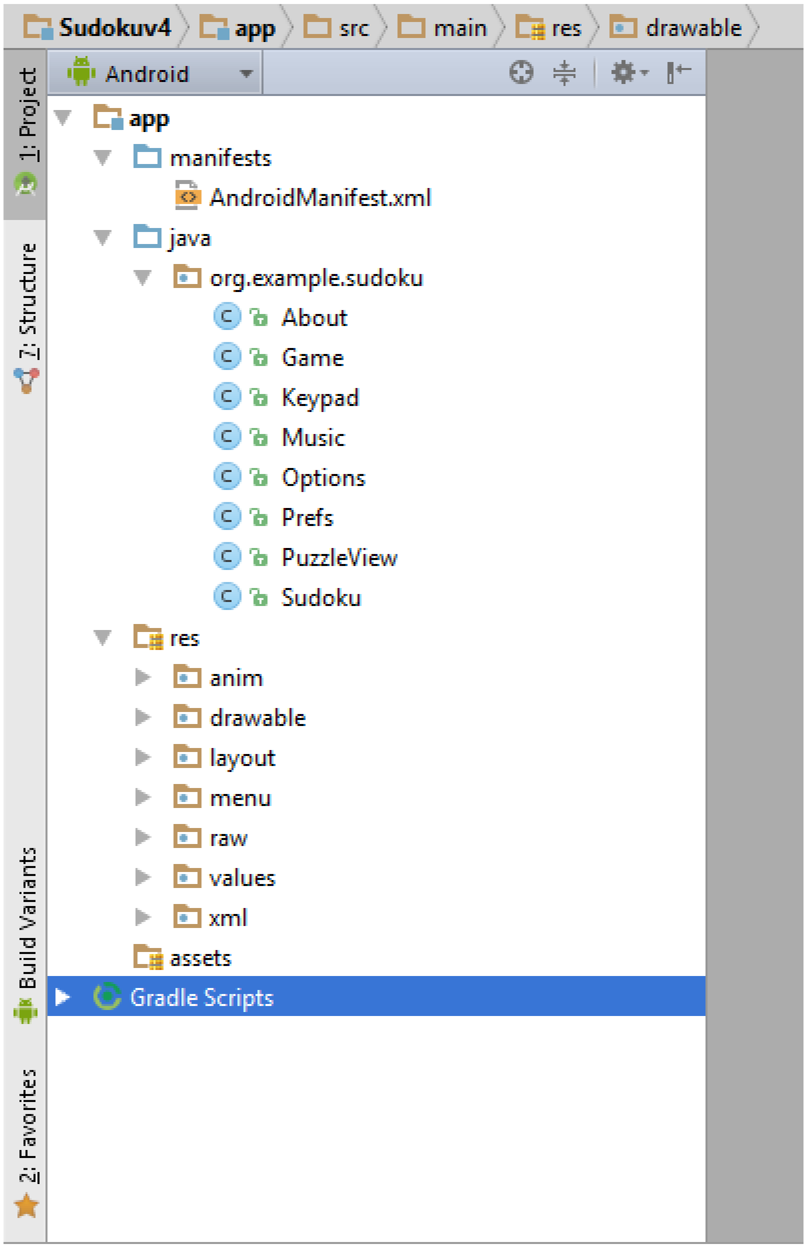
Gradle build finished in 8 sec (moments ago)

6. It will display a summary of the changes done as Android Gradle projects use a different structure than ADT Eclipse projects. Android Studio modifies the structure of the project accordingly, so you do not need to worry.



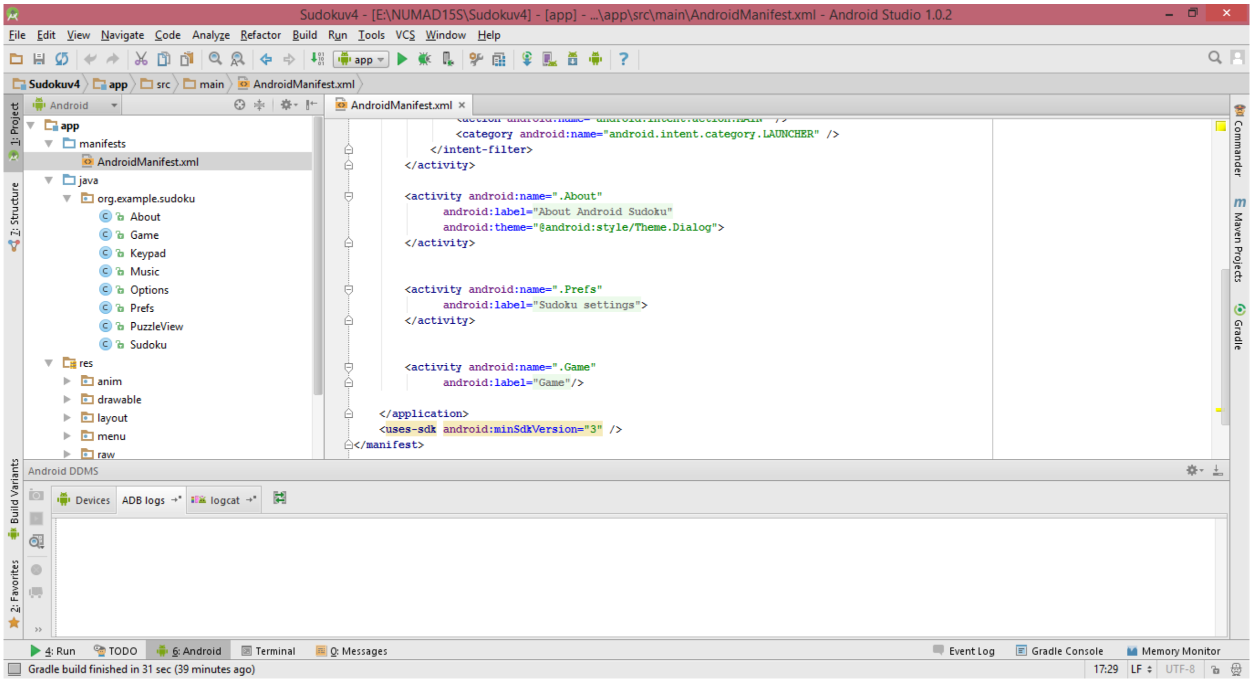
```
ECLIPSE ANDROID PROJECT IMPORT SUMMARY
======================================

Ignored Files:
--------------
The following files were *not* copied into the new Gradle project; you
should evaluate whether these are still needed in your project and if
so manually move them:

* default.properties

Moved Files:
------------
Android Gradle projects use a different directory structure than ADT
Eclipse projects. Here's how the projects were restructured:

* AndroidManifest.xml => app\src\main\AndroidManifest.xml
* assets\ => app\src\main\assets\
* res\ => app\src\main\res\
* src\ => app\src\main\java\

Next Steps:
-----------
You can now build the project. The Gradle project needs network
connectivity to download dependencies.

Bugs:
-----
If for some reason your project does not build, and you determine that
it is due to a bug or limitation of the Eclipse to Gradle importer,
please file a bug at http://b.android.com with category
Component-Tools.

(This import summary is for your information only, and can be deleted
```

7. Your project structure will look like this.



```
app
    manifests
        AndroidManifest.xml
    java
        org.example.sudoku
            About
            Game
            Keypad
            Music
            Options
            Prefs
            PuzzleView
            Sudoku
    res
        anim
        drawable
        layout
        menu
        raw
        values
        xml
    assets
Gradle Scripts
```

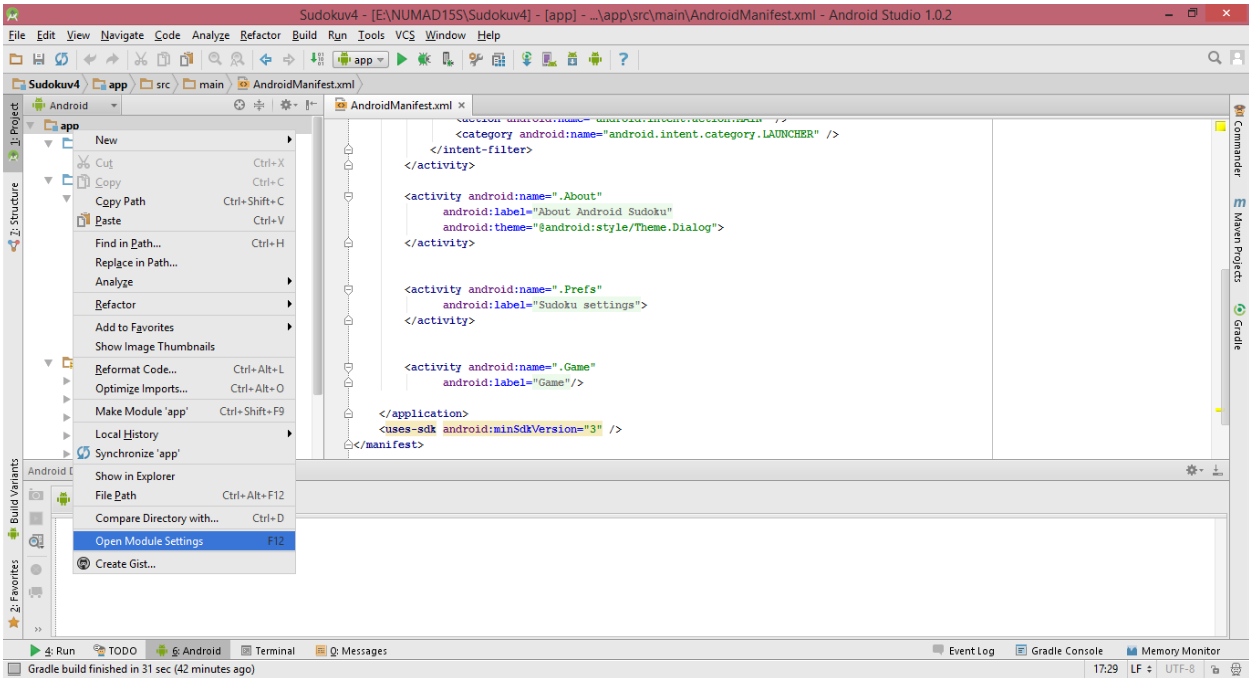
8. Open the AndroidManifest.xml and scroll to the bottom. The `uses-sdk` tag will have an `android:minSdkVersion="3"` declaration which is an integer designating the minimum API Level required for the application to run. Despite its name, this element is used to specify the API Level, *not* the version number of the SDK (software development kit) or Android platform. The API Level is always a single integer.



```
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".About"
            android:label="About Android Sudoku"
            android:theme="@android:style/Theme.Dialog">
        </activity>


        <activity android:name=".Prefs"
            android:label="Sudoku settings">
        </activity>


        <activity android:name=".Game"
            android:label="Game"/>

    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Gradle build finished in 31 sec (39 minutes ago)

9. To change the minimum version or the target version, right-click on the **app** module, and click on **Open Module Settings**.

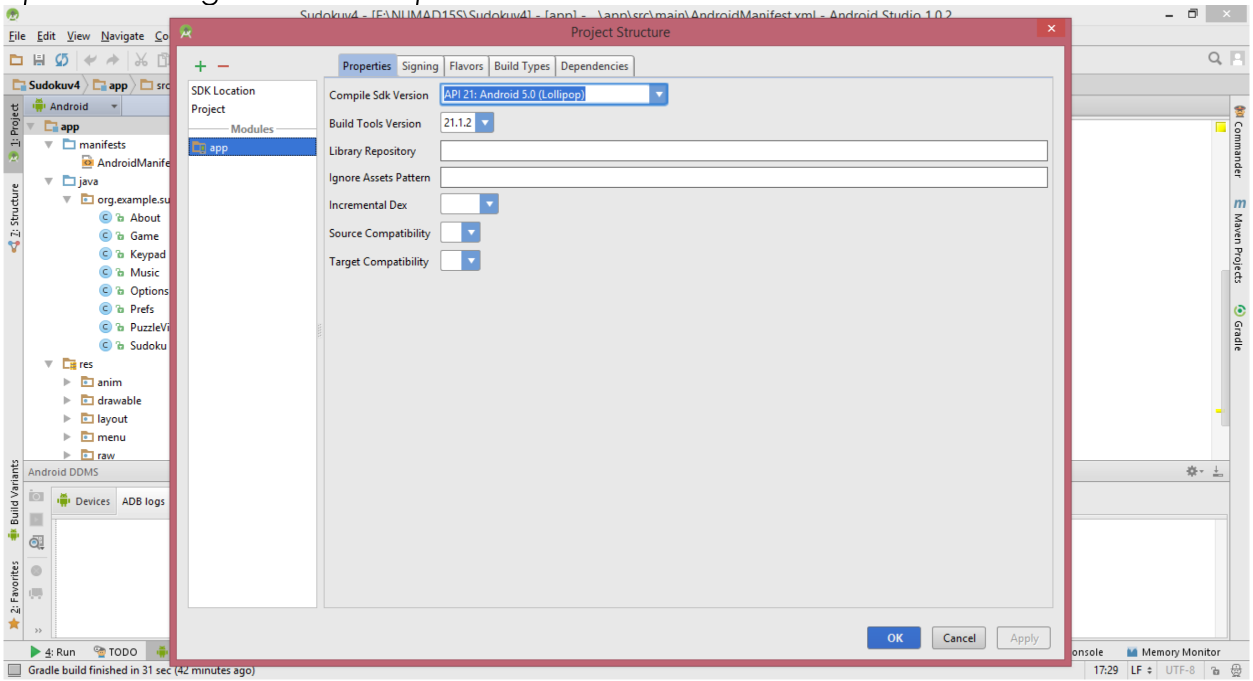


```
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".About"
            android:label="About Android Sudoku"
            android:theme="@android:style/Theme.Dialog">
        </activity>


        <activity android:name=".Prefs"
            android:label="Sudoku settings">
        </activity>


        <activity android:name=".Game"
            android:label="Game"/>

    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Gradle build finished in 31 sec (42 minutes ago)

10. In the Properties tab, the compile version is set to the latest version of Android available in your SDK.

Tip: Choose the highest platform version available so that your application will be optimized for greater user experience on the latest devices.



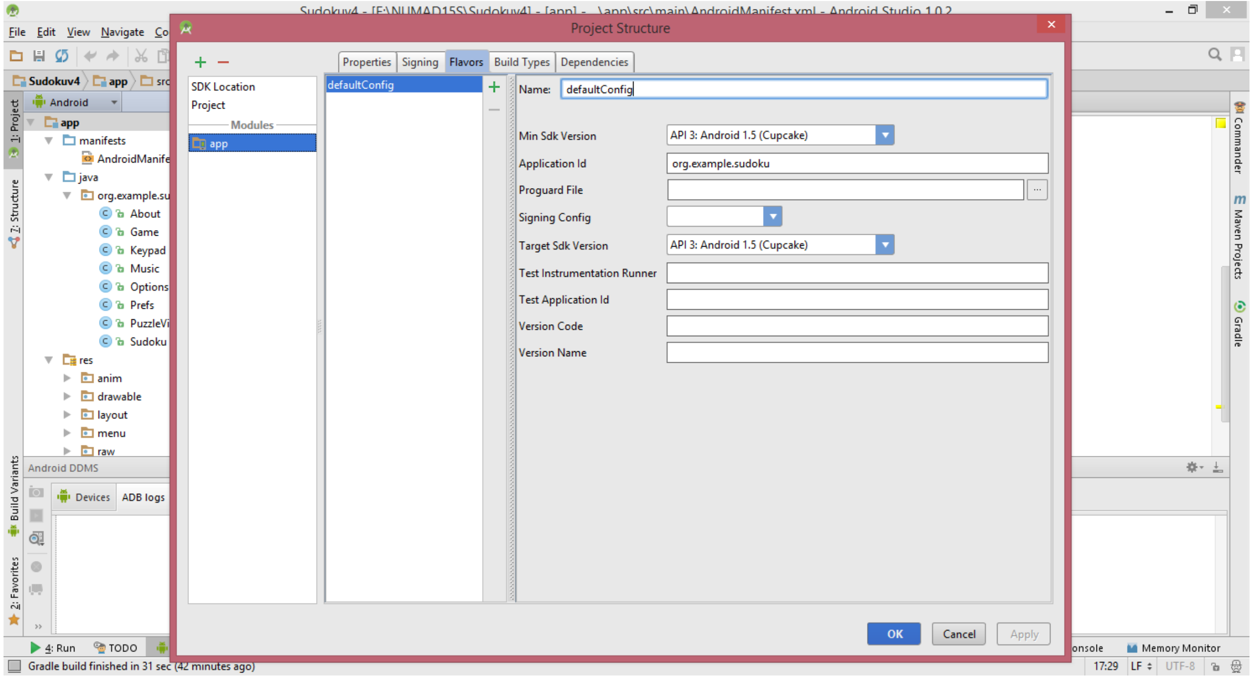
Sudokuv4 - [F:\NUMAD15S\Sudokuv4] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 1.0.2

Project Structure

Compile Sdk Version: API 21: Android 5.0 (Lollipop)
Build Tools Version: 21.1.2
Library Repository:
Ignore Assets Pattern:
Incremental Dex:
Source Compatibility:
Target Compatibility:

Gradle build finished in 31 sec (42 minutes ago)

11. In the Flavors tab, you will see the minimum version as well as the target version. For the Sudoku example, the minimum version is set to "3" by default.



Sudokuv4 - [F:\NUMAD15S\Sudokuv4] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 1.0.2

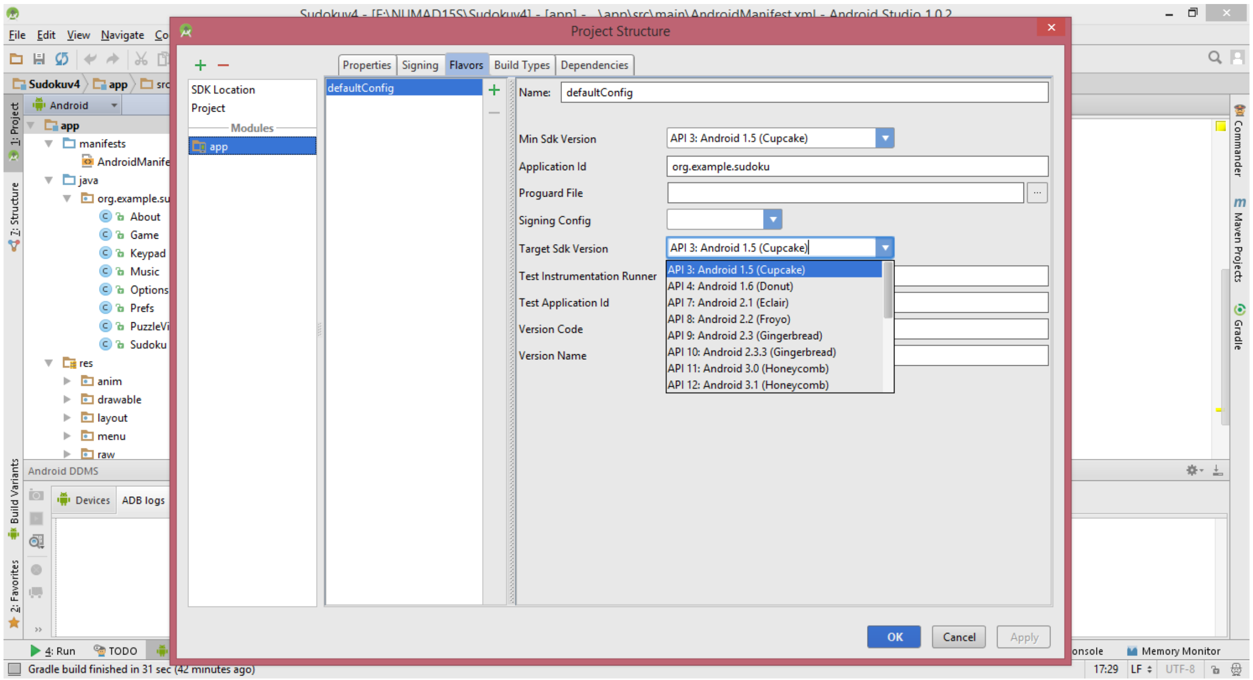
Project Structure

defaultConfig

Name: defaultConfig
Min Sdk Version: API 3: Android 1.5 (Cupcake)
Application Id: org.example.sudoku
Proguard File:
Signing Config:
Target Sdk Version: API 3: Android 1.5 (Cupcake)
Test Instrumentation Runner:
Test Application Id:
Version Code:
Version Name:

Gradle build finished in 31 sec (42 minutes ago)

12. For the target version, set the Android version to 2.3.3 (Gingerbread) which is API Level 10.



Sudokuv4 - [F:\NUMAD15S\Sudokuv4] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 1.0.2

Project Structure

Name: defaultConfig

Min Sdk Version: API 3: Android 1.5 (Cupcake)

Application Id: org.example.sudoku

Proguard File:

Signing Config:

Target Sdk Version: API 3: Android 1.5 (Cupcake)

Test Instrumentation Runner

Test Application Id

Version Code

Version Name

API 3: Android 1.5 (Cupcake)
API 4: Android 1.6 (Donut)
API 7: Android 2.1 (Eclair)
API 8: Android 2.2 (Froyo)
API 9: Android 2.3 (Gingerbread)
API 10: Android 2.3.3 (Gingerbread)
API 11: Android 3.0 (Honeycomb)
API 12: Android 3.1 (Honeycomb)

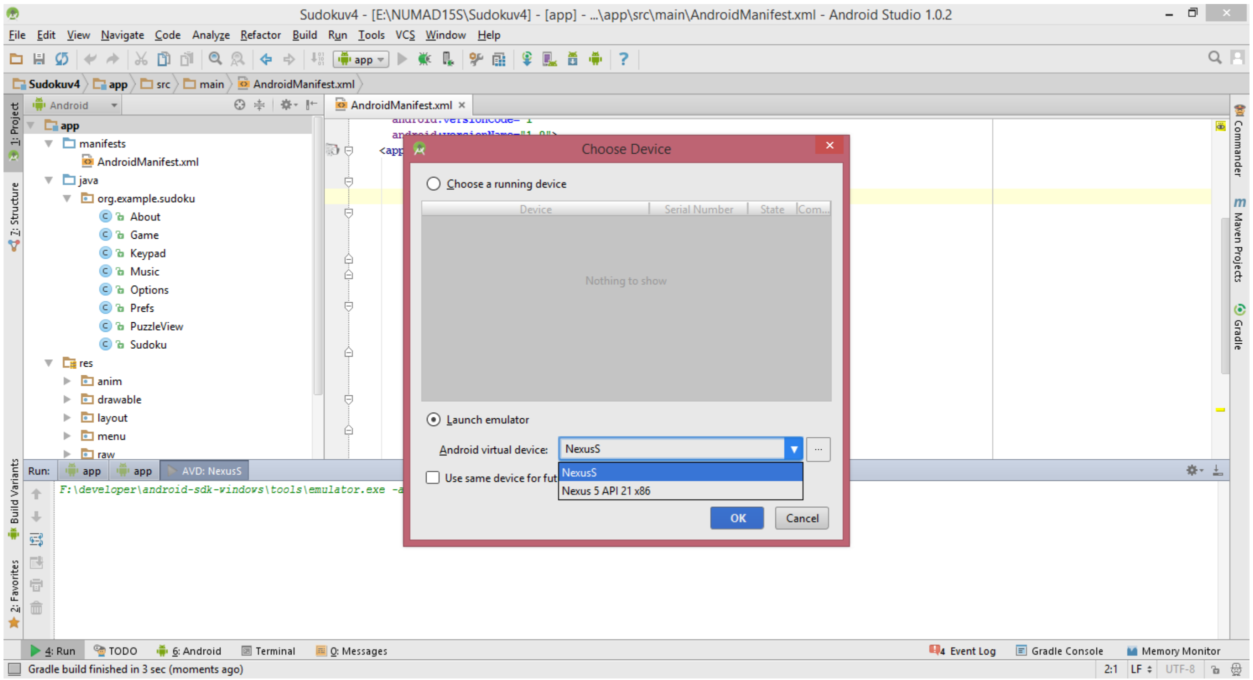
Gradle build finished in 31 sec (42 minutes ago)

android:minSdkVersion
An integer designating the minimum API Level required for the application to run. The Android system will prevent the user from installing the application if the system's API Level is lower than the value specified in this attribute.

android:targetSdkVersion
An integer designating the API Level that the application targets. If not set, the default value equals that given to `minSdkVersion`.

This attribute informs the system that you have tested against the target version and the system should not enable any compatibility behaviors to maintain your app's forward-compatibility with the target version. The application is still able to run on older versions (down to `minSdkVersion`).

13. Run the application by clicking on the ‘Run app’ (Alt + Shift + X) from the Run menu.



Sudokuv4 - [E:\NUMAD15S\Sudokuv4] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 1.0.2

Choose Device

○ Choose a running device

Device	Serial Number	State	Com...

Nothing to show

● Launch emulator

Android virtual device: NexusS

NexusS
Nexus 5 API 21 x86

□ Use same device for fut

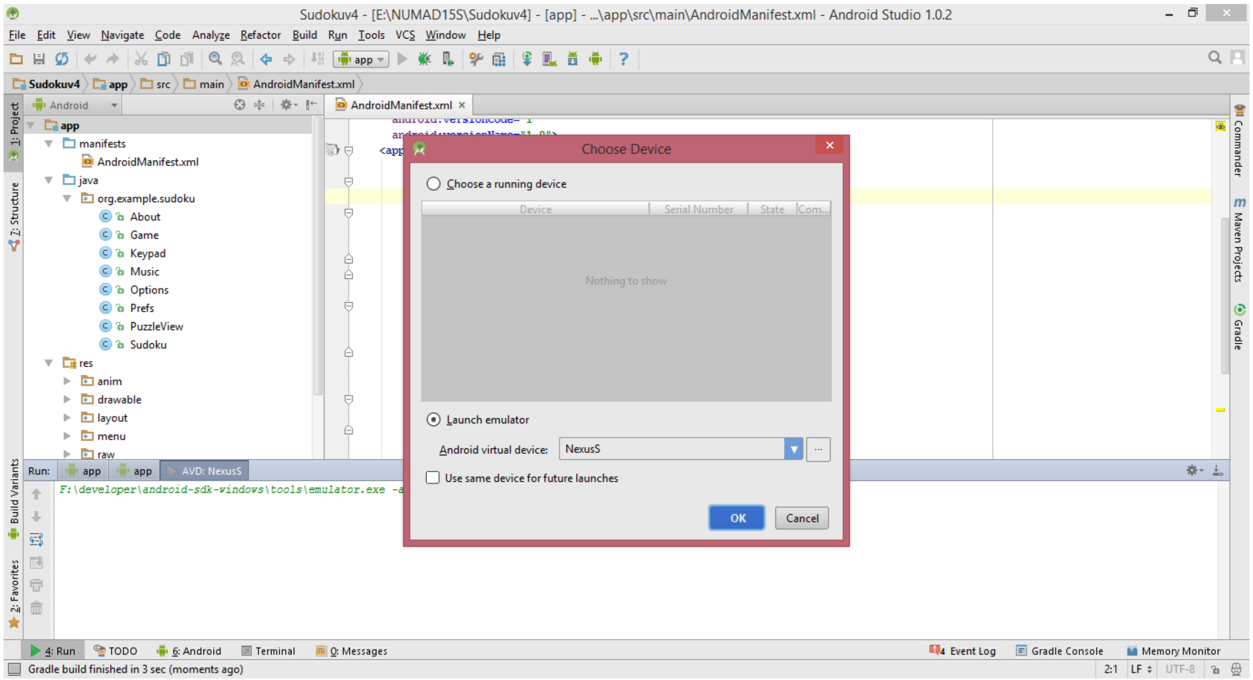
OK Cancel

```
F:\developer\android-sdk-windows\tools\emulator.exe -a
```

Gradle build finished in 3 sec (moments ago)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Sudokuv4 app src main AndroidManifest.xml
Android
app
manifests
AndroidManifest.xml
java
org.example.sudoku
About
Game
Keypad
Music
Options
Prefs
PuzzleView
Sudoku
res
anim
drawable
layout
menu
raw
Run: app app AVD: NexusS
4: Run TODO 6: Android Terminal 0: Messages
4 Event Log Gradle Console Memory Monitor
2:1 LF UTF-8
1: Project 7: Structure Build Variants 2: Favorites Commander Maven Projects Gradle

14. Select the ‘Launch emulator’ option and choose a Android virtual device.



Sudokuv4 - [E:\NUMAD15S\Sudokuv4] - [app] - ...\app\src\main\AndroidManifest.xml - Android Studio 1.0.2

Choose Device

○ Choose a running device

Device	Serial Number	State	Com...

Nothing to show

● Launch emulator

Android virtual device: NexusS

□ Use same device for future launches

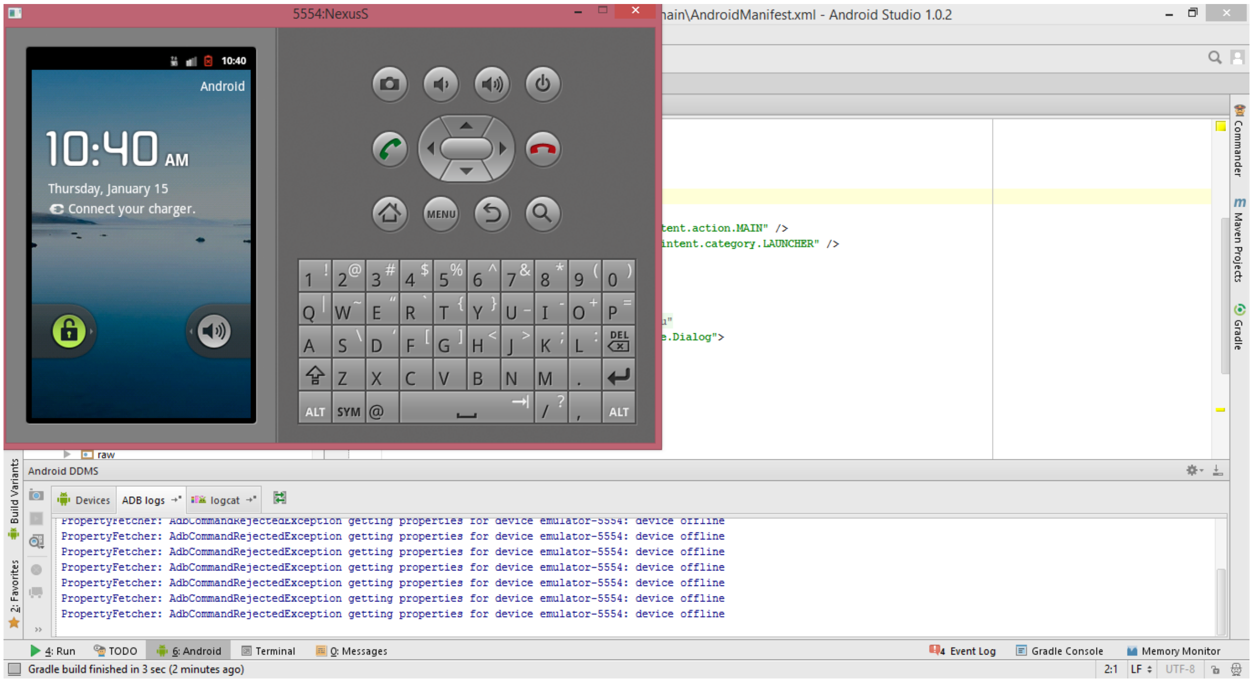
OK Cancel

```
F:\developer\android-sdk-windows\tools\emulator.exe -a
```

Gradle build finished in 3 sec (moments ago)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Sudokuv4 app src main AndroidManifest.xml
Android
app
manifests
AndroidManifest.xml
java
org.example.sudoku
About
Game
Keypad
Music
Options
Prefs
PuzzleView
Sudoku
res
anim
drawable
layout
menu
raw
Run: app app AVD: NexusS
4: Run TODO 6: Android Terminal 0: Messages
4 Event Log Gradle Console Memory Monitor
2:1 LF UTF-8
1: Project 7: Structure Build Variants 2: Favorites Commander Maven Projects Gradle

15. The AVD launches in a separate window.

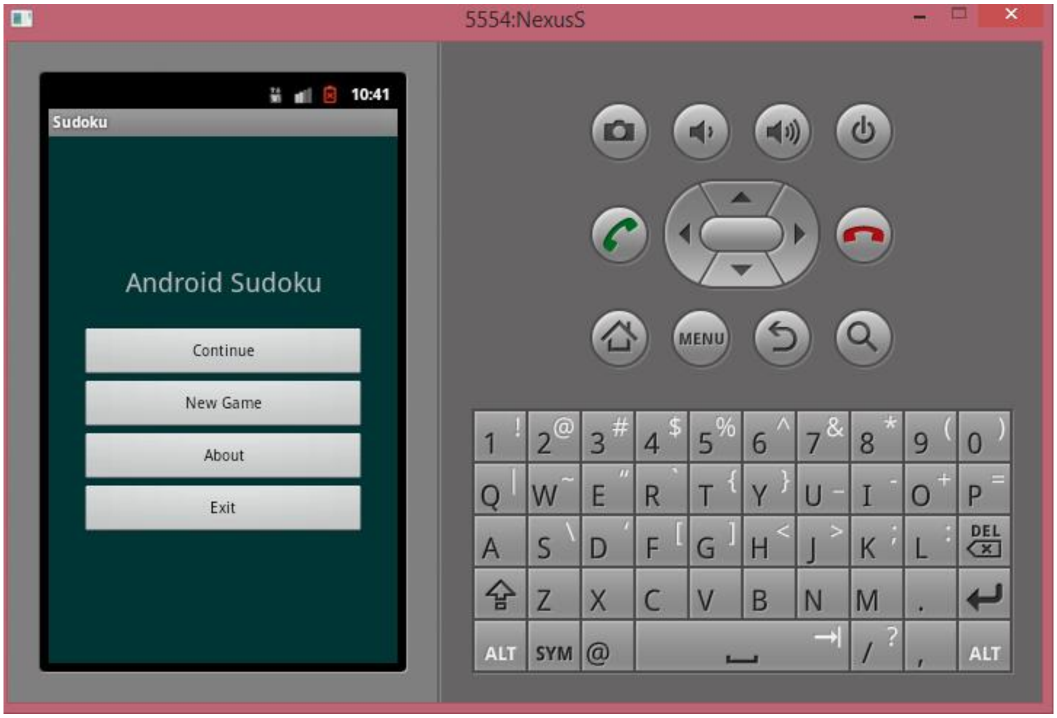


```
tent.action.MAIN" />
intent.category.LAUNCHER" />
a"
e.Dialog">
```

```
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
PropertyFetcher: AdbCommandRejectedException getting properties for device emulator-5554: device offline
```

Gradle build finished in 3 sec (2 minutes ago)

16. The Sudoku application will run like below in the AVD. You're all set!



References:

https://developer.android.com/sdk/installing/index.html?pkg=studio
https://developer.android.com/sdk/installing/adding-packages.html
https://developer.android.com/guide/topics/manifest/uses-sdk-element.html