Guide to setting up an emulator

This installation guide document is not a substitute but a complementary resource for the videos provided in the course.

Please note that this is a working document and that errors may remain.

There are lot of things that are discussed in the videos that are not discussed here. We highly recommend you viewing the videos to understand the process fully. The setting up process is similar for all the Operating Systems we support. We have used illustrations from a Windows environment.

1. Open Android Virtual Device Manager

   Clicking on the Android Virtual Device Manager or AVD Manager (Figure 1) will open the AVD Manager.

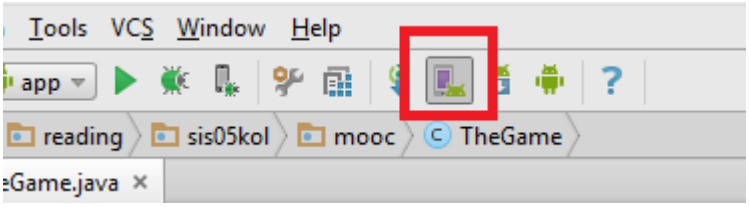

Figure 1: Android Virtual Device Manager

It may take a little time to load. Once loaded up it will look similar to Figure 2.

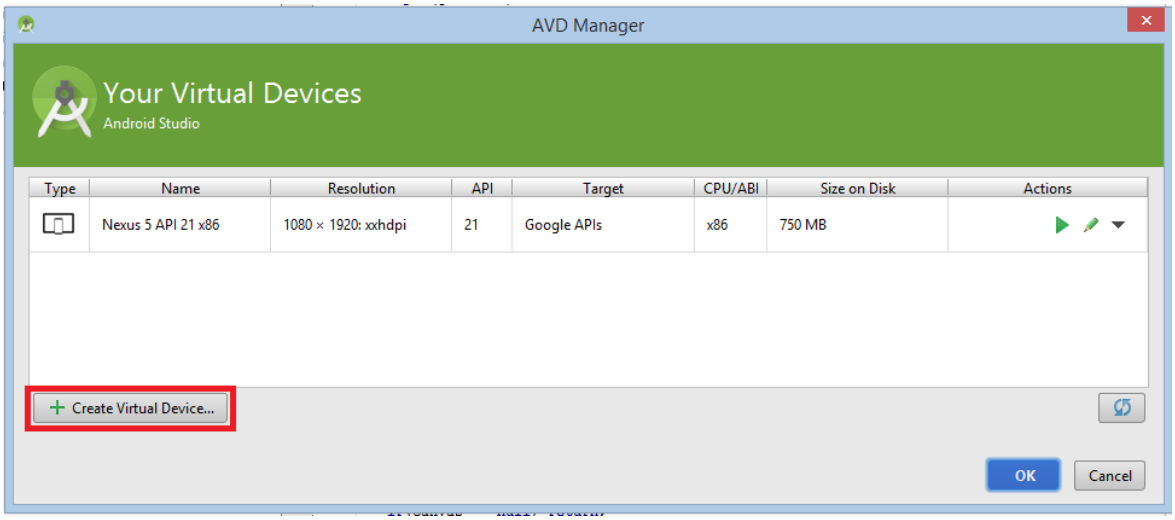

Figure 2: Android Virtual Device Manager Dialog

2. Create a new AVD

Click the 'Create Virtual Device' button (Figure 2).

3. Select the type of device you want to create.

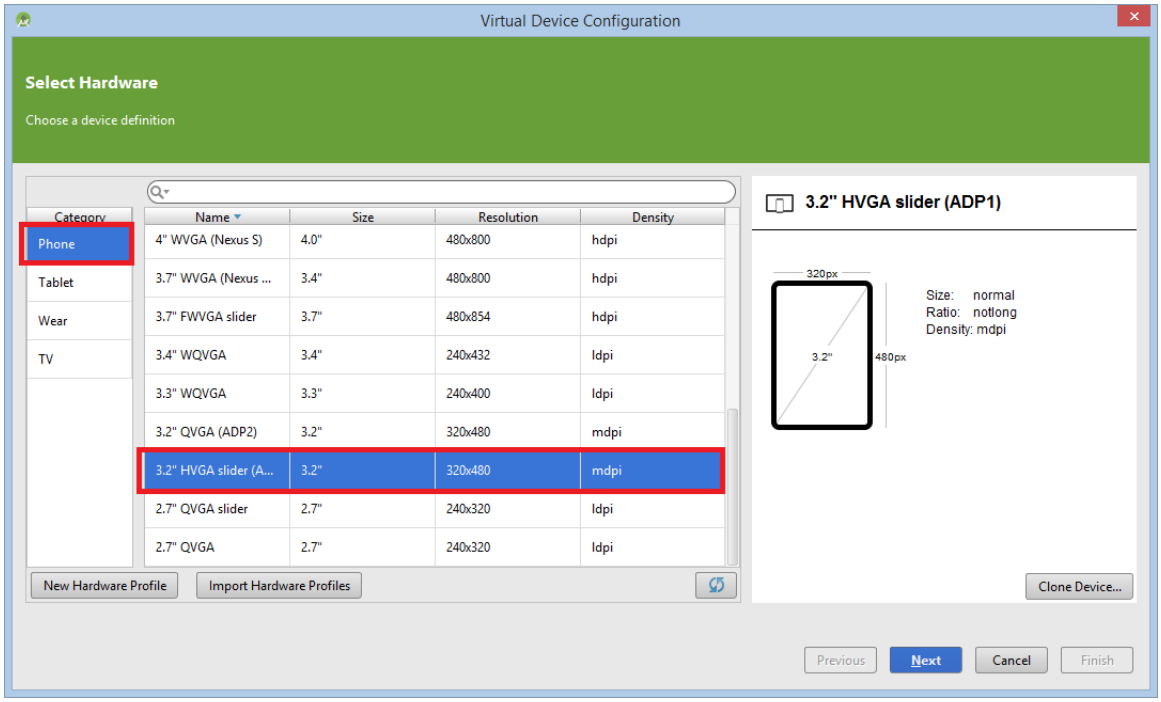

Figure 3: Virtual Device Configuration

4. Select the API Level

When you click 'Next' on Virtual Device Configuration dialog (Figure 3) it takes you to select a System Image. All installed API levels are shown in **Bold** while other available API levels are shown with a link to download them (Figure 4).

We advise using **Lollipop API level 21 armeabi-v7a** Target Android SDK Platform 5.0

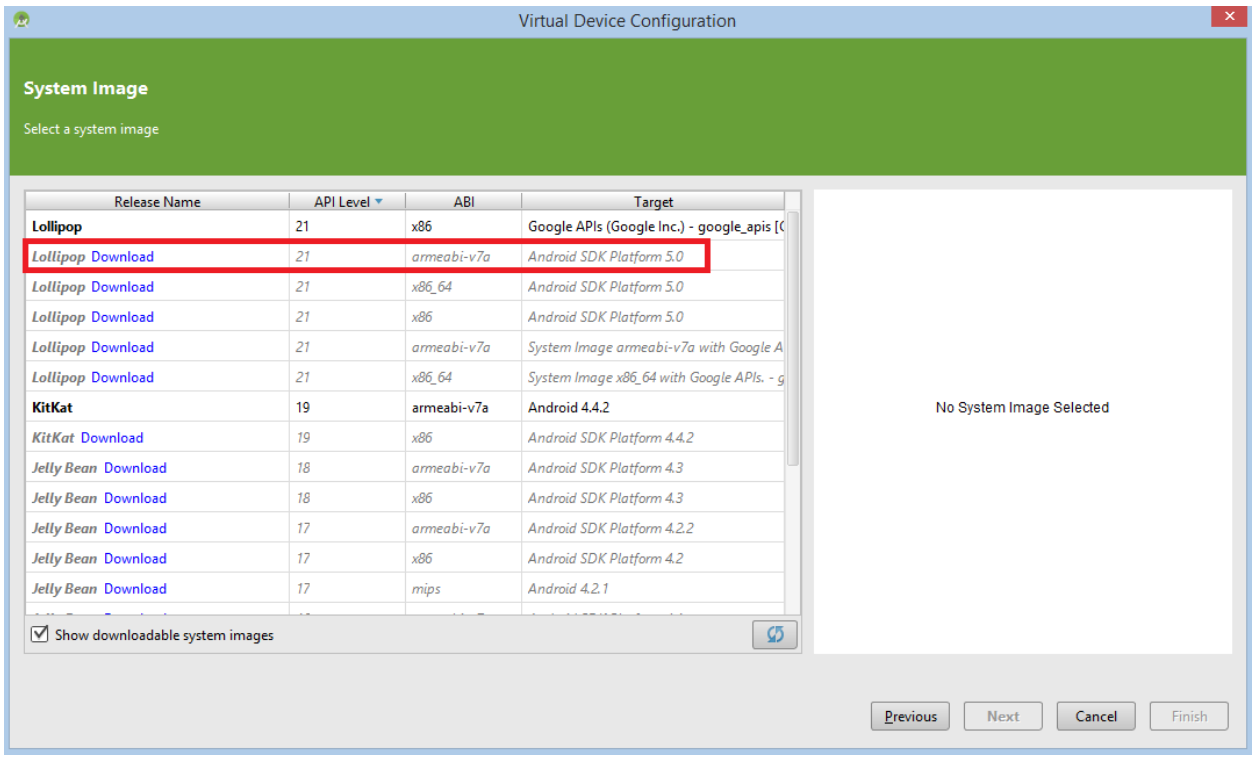

Figure 4: Virtual Device Configuration

However, this API level may not come installed; so we have to download and install it. Click on the 'Download' link for this API level.

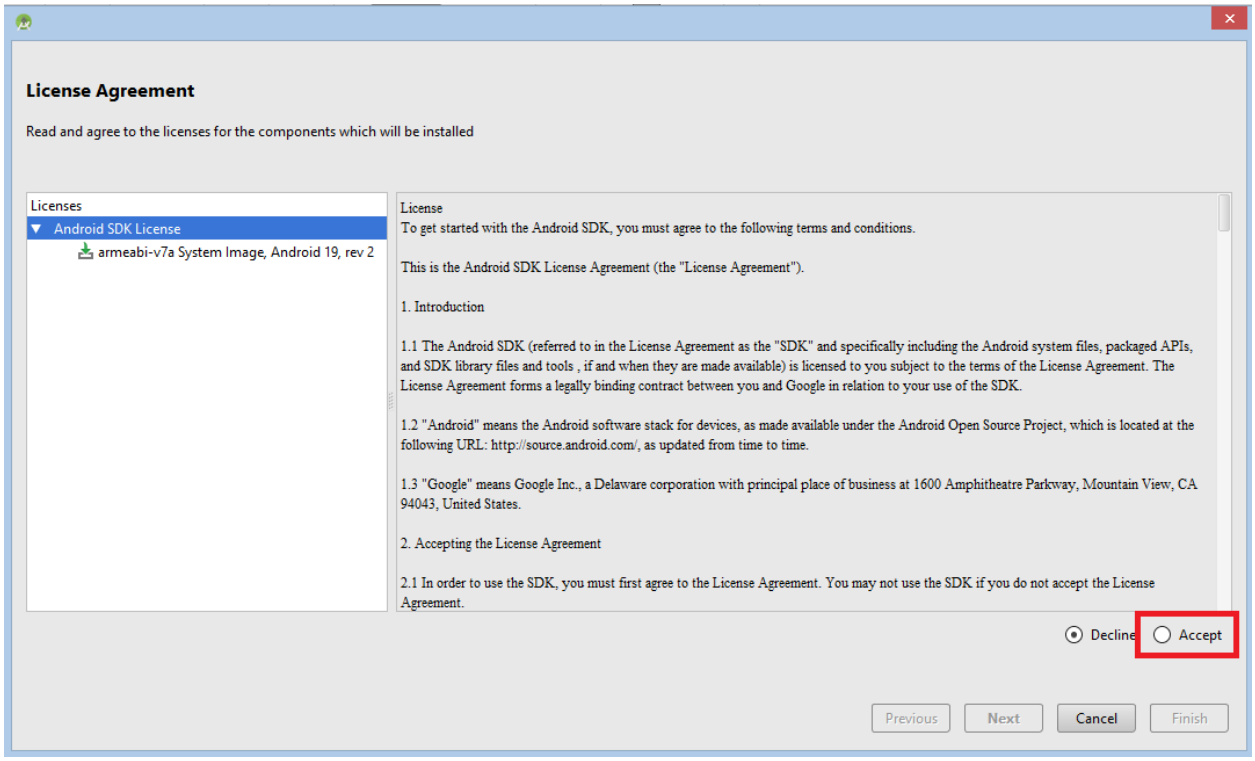

Figure 5: License Agreement

Accept the License Agreement (Figure 5).

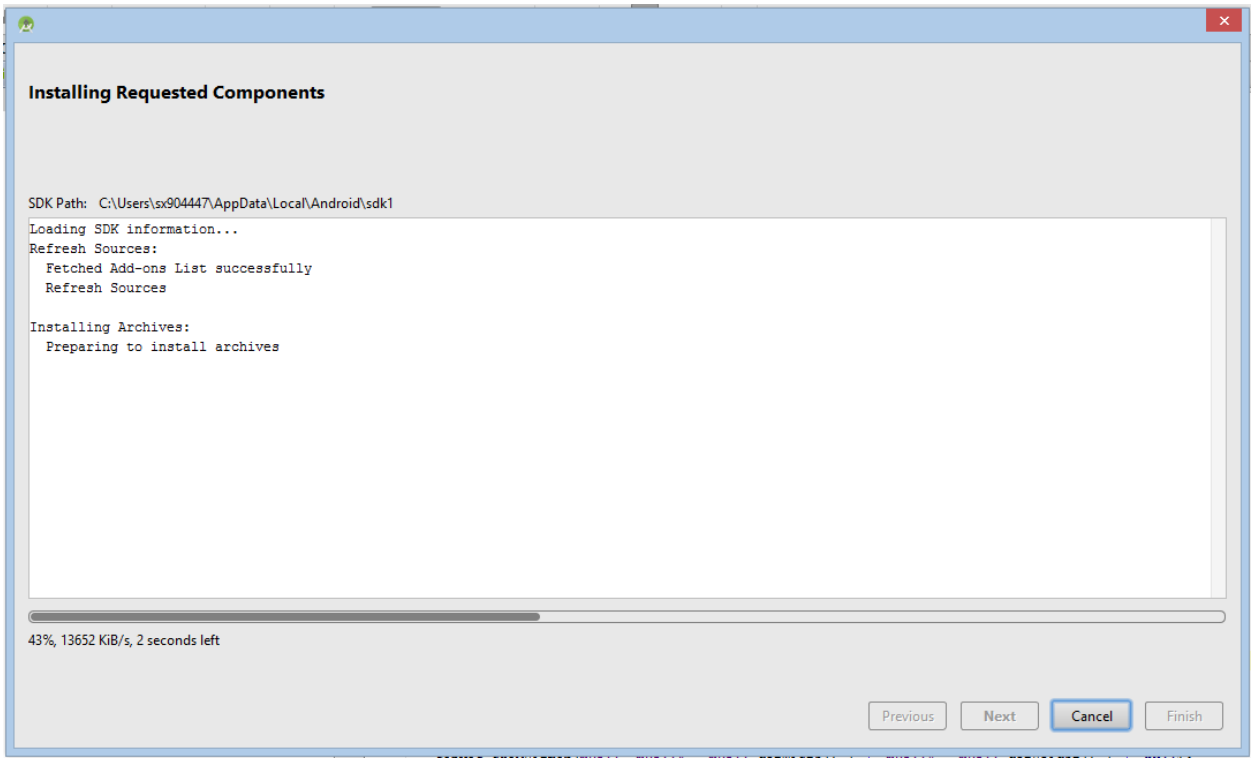

Figure 6: Installing the selected API Level

It will take some time to install the selected API level (Figure 6).

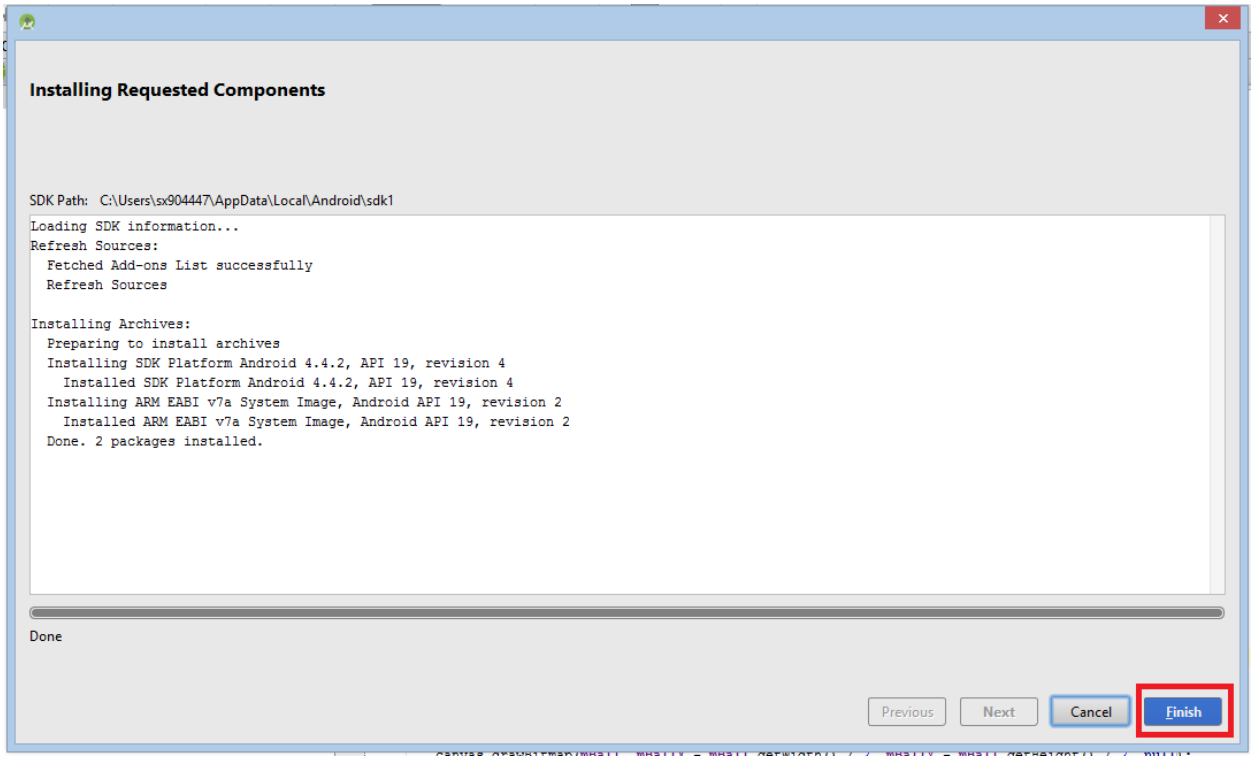

Figure 7: Installing API Level Completed

Click 'Finish' to go to the next step after finishing the API level installation (Figure 7). Now the installed API level will be shown in **bold** (Figure 8)

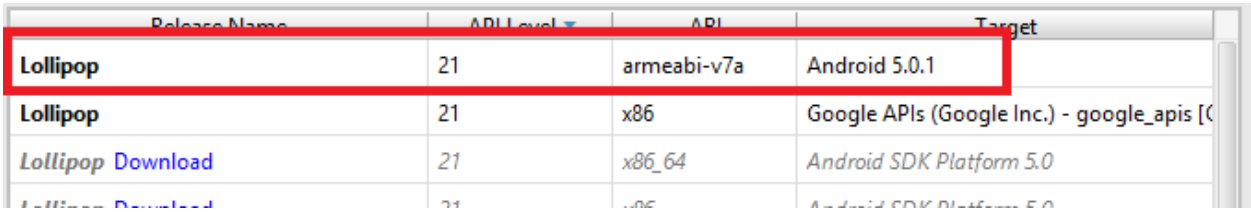

Figure 8: Installed API Level

Now select the required API Level (Lollipop API Level 21 armeabi-v7a Android 5.0.1) and click next.

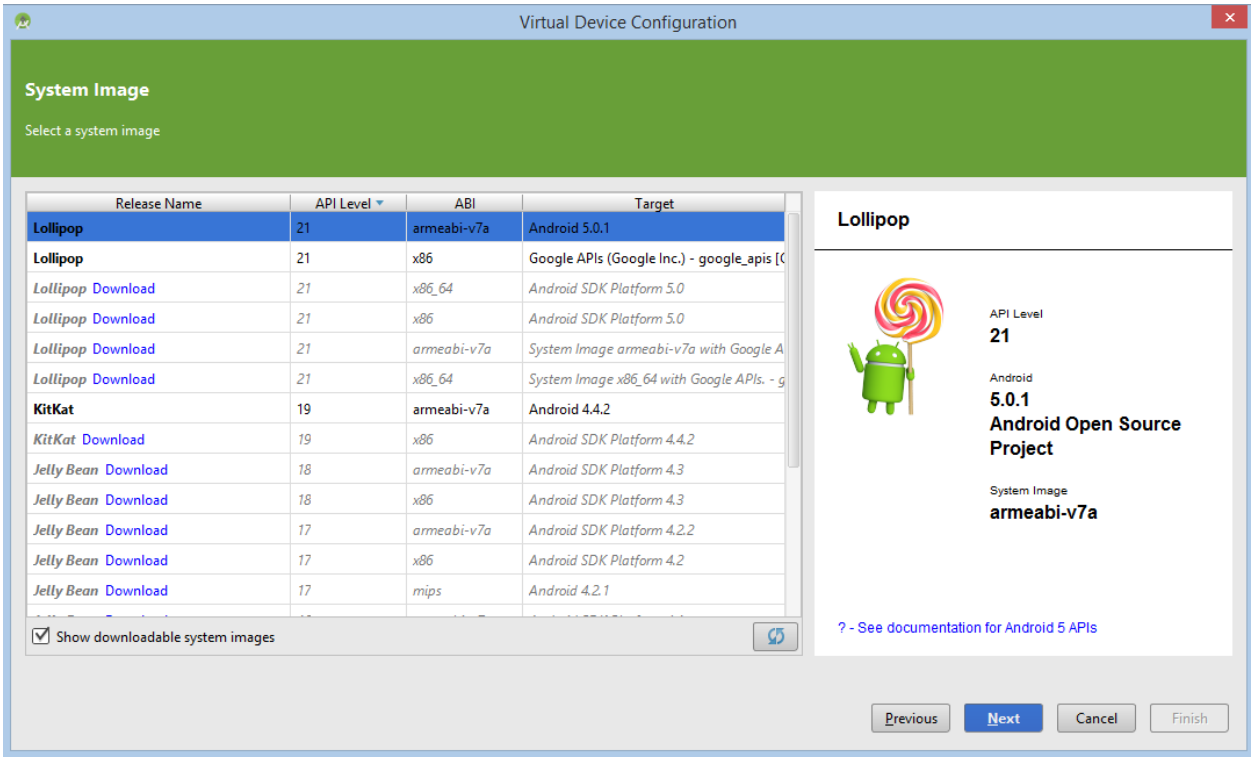

Figure 9: Select API Level 19

5. Give a name to the AVD and uncheck ‘Use Host GPU’ check box (Figure 10). Click ‘Finish’.

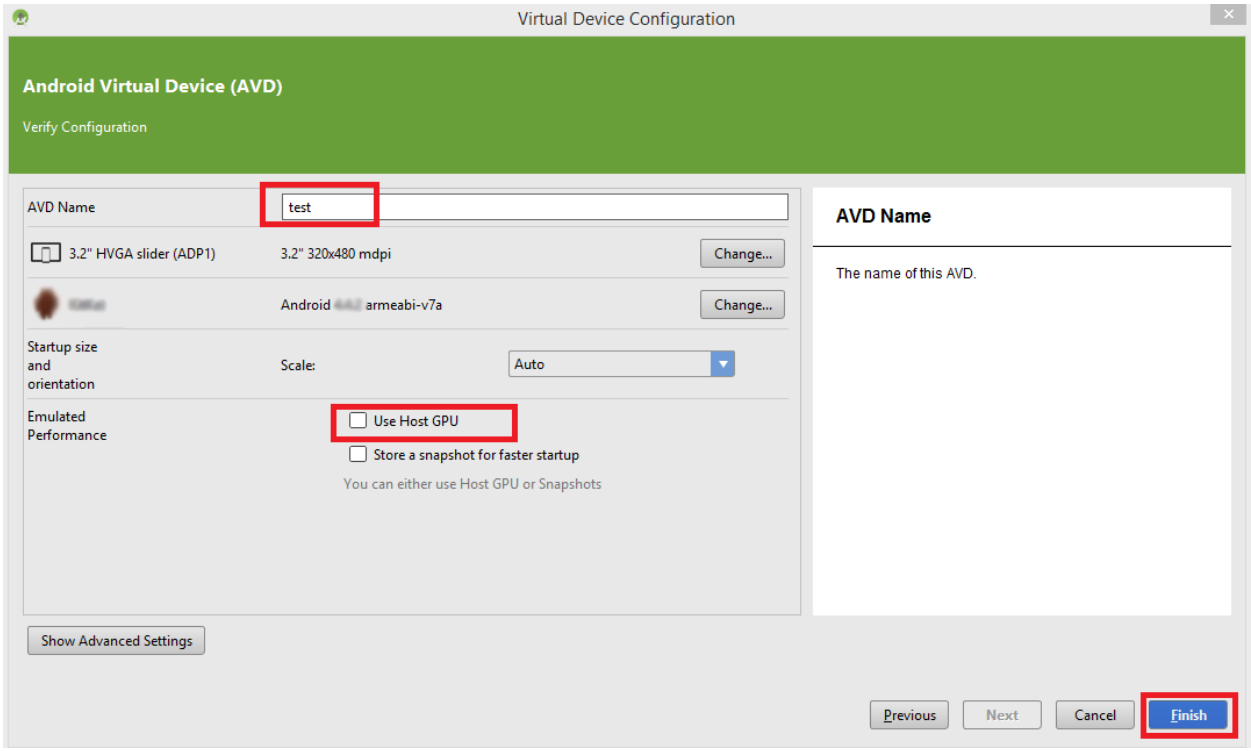

Figure 10: Verify AVD Configuration

6. Now the AVD Manager will show the newly created AVD (Figure 11).

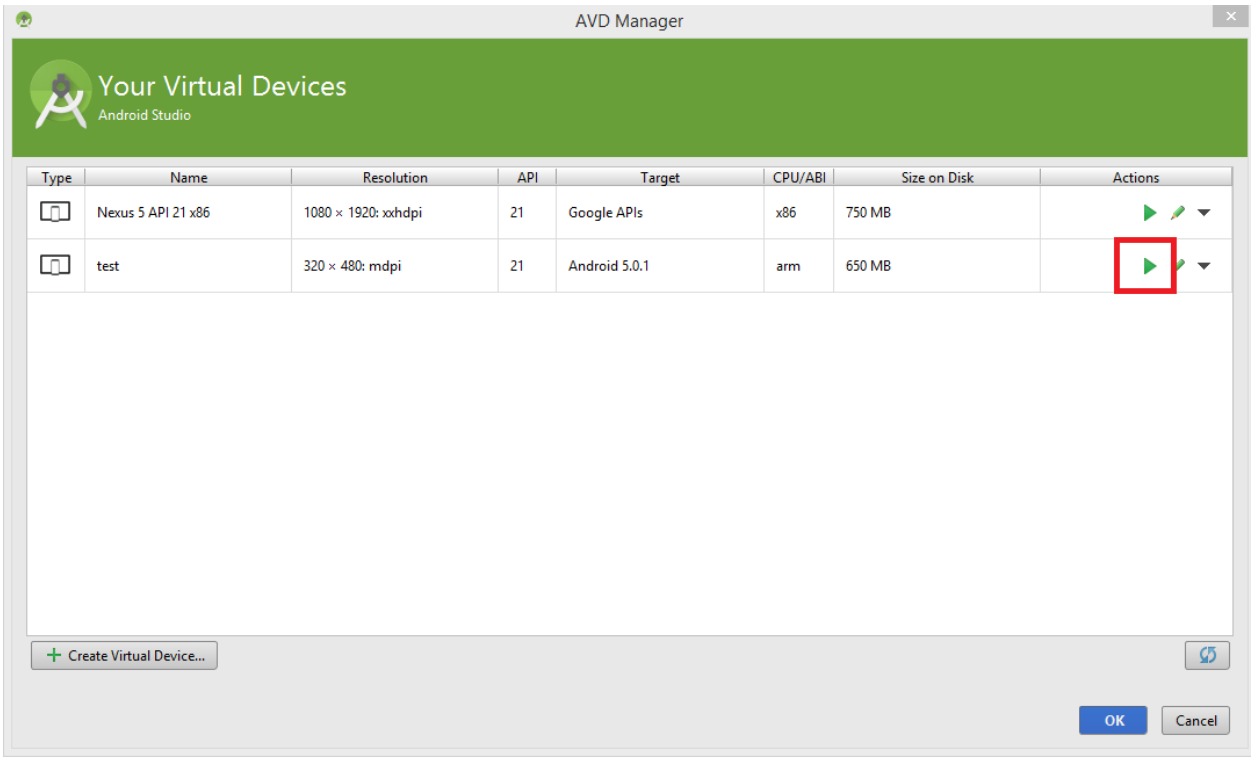

Figure 11: AVD Manager showing the newly created AVD

7. To start the AVD, select the created AVD and click the start button (the green triangle) to launch the AVD (Figure 11).

Figure 12: Launching AVD

8. It will take some time to launch the AVD. Some machines can take about 10min to launch it. While launching it will show AVD window loading resembling a loading Android device (Figure 12).

Figure 12: AVD is loading

9. Once fully loaded AVD look similar to the Android device you selected in the AVD configuration. The one selected in this tutorial is shown in Figure 13.

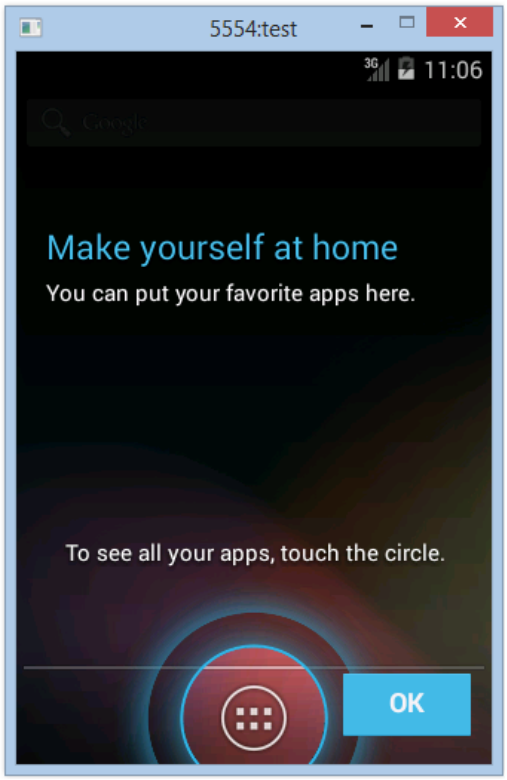

Figure 13: AVD is loading

10. You can explore the AVD by clicking 'OK'. It is actually a virtual Android device.

Figure 14: Various displays

NOTE: If you leave your Emulator without interacting with it for a while the screen locks itself similar to an Android phone. Locked screen will have a closed padlock (Figure 15).

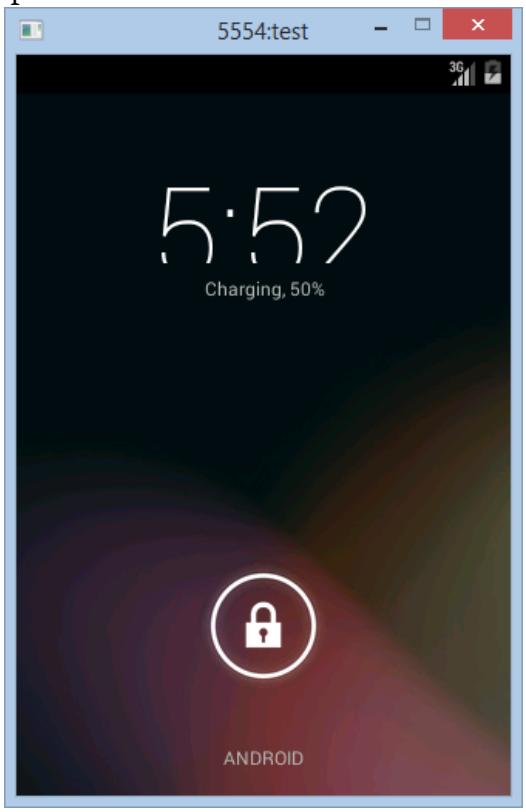

Figure 15: Emulator screen locked

If this happens you need to click on the padlock and drag outward (Figure 16) until it opens (Figure 17).

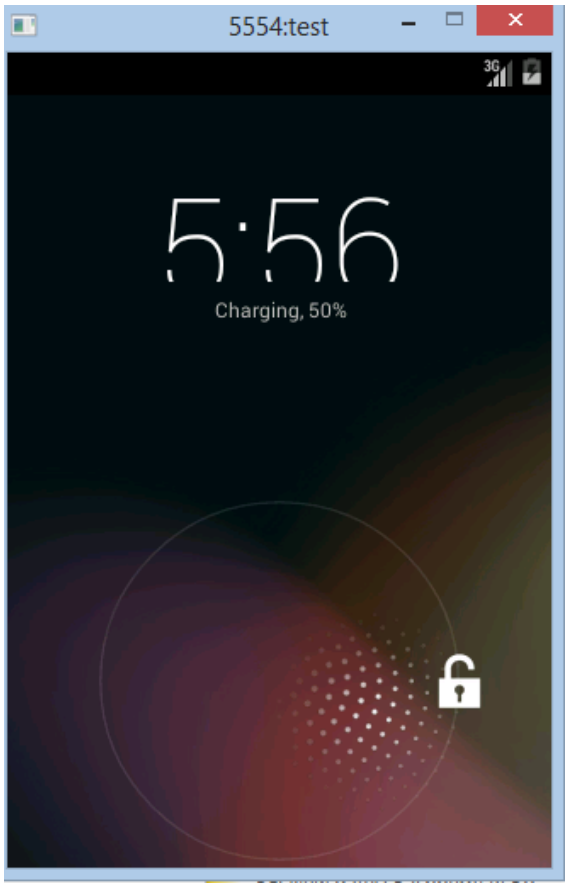

Figure 16: Emulator screen unlocking

Figure 17: Emulator screen unlocked

Comments

Comments are used in programs to describe complex logic in a program or source code to improve human readability. Comments are ignored by the computer when executing the code. In IDEs comments are highlighted in a different colour so that it is easier to see that they are 'comments' and not the source code or instructions in the program.

In Java we use two different types of comments: single line comments and multi line comment.

```
// this is a single line comment
```

Two forward slashes are used to distinguish single ling comments.

```
/* this is a
   multi line
   comment */
```

Anything that is between `/*` and `*/` is considered a multi line comment.

if you go down on TheGame.java file in the project you can see there is a section of code 'actionOnTouch' method that is being commented out.

```
//This is run whenever the phone is touched by the user
/*
@Override
protected void actionOnTouch(float x, float y) {
    //Increase/decrease the speed of the ball making the ball move towards the touch
    mBallSpeedX = x - mBallX;
    mBallSpeedY = y - mBallY;
}
*/
```

Figure 18: Commented out actionOnTouch method

Because the actionOnTouch method is commented out, if you run this code even when the phone is touched nothing will happen. If you want your phone to respond to the touch then we have to 'uncomment' this block of code by removing /* and */ symbols surrounding the actionOnTouch method.

```
//This is run whenever the phone is touched by the user
/*
@Override
protected void actionOnTouch(float x, float y) {
    //Increase/decrease the speed of the ball making the ball move towards the touch
    mBallSpeedX = x - mBallX;
    mBallSpeedY = y - mBallY;
}
*/
```

Figure 19: Remove /* and */ to uncomment

When uncommenting you will notice the change of colour in this block of code. This is because now the IDE identifies this block as Java code rather than a comment.

```
@Override
protected void actionOnTouch(float x, float y) {
    //Increase/decrease the speed of the ball making the ball move towards the touch
    mBallSpeedX = x - mBallX;
    mBallSpeedY = y - mBallY;
}
```

Figure 20: Uncommented block of code

Running the App

12. Click the Run 'app' button (the green triangle – Figure 21).

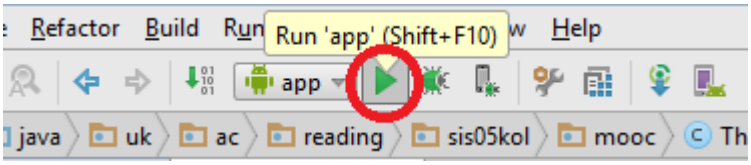

Figure 21: Click Run 'app'

13. It will then present the Choose Device dialog (Figure 22).

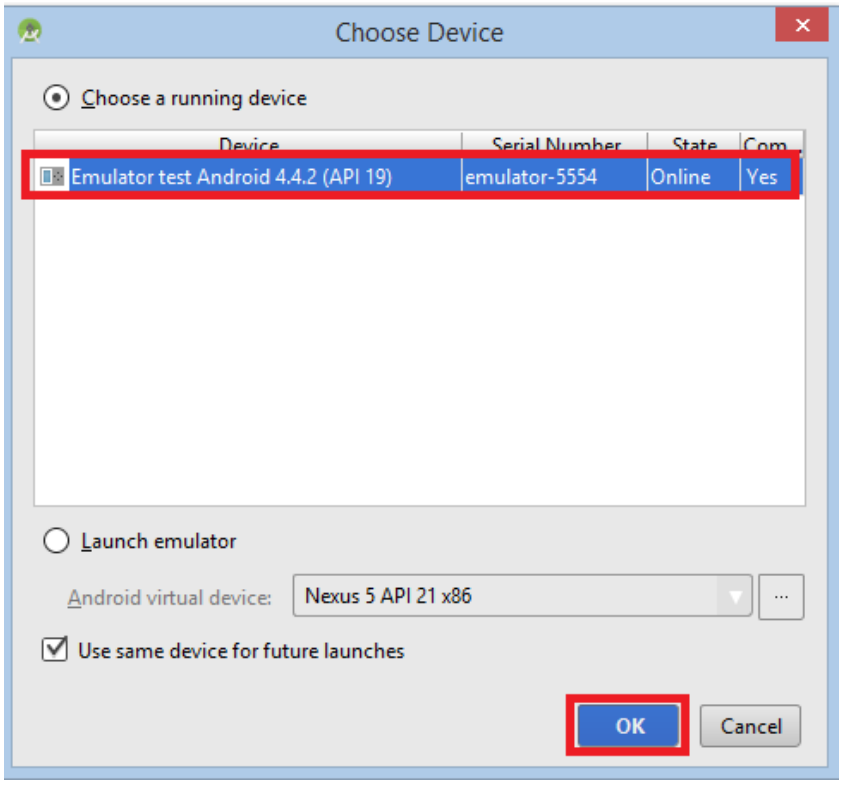

Figure 22: Choose Device Dialog

Select the emulator we have already launched and click OK.

14. It will take a bit of time and then will show the App on the phone (Figure 23)

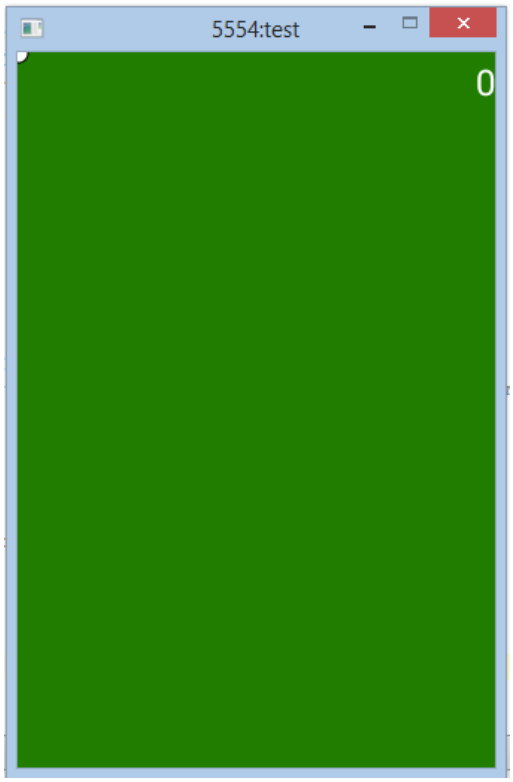

Figure 23: App running on the emulator

15. Click on the screen and you will see the white ball on the screen.

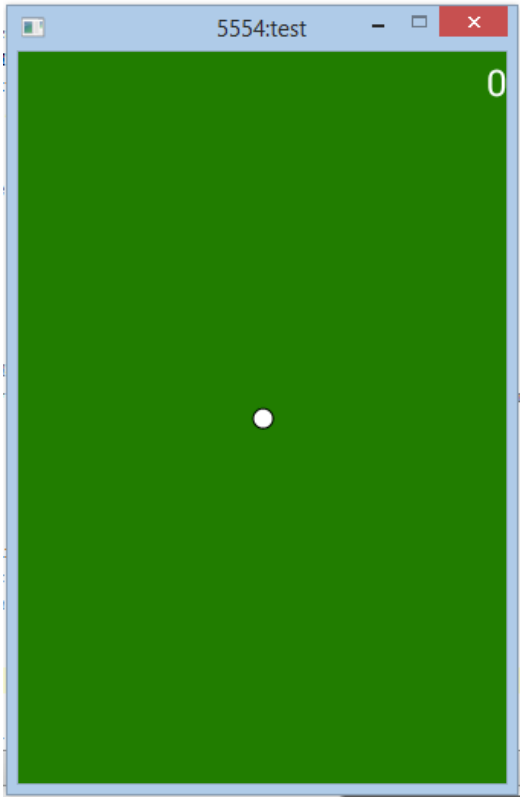

Figure 24: App responding to clicks