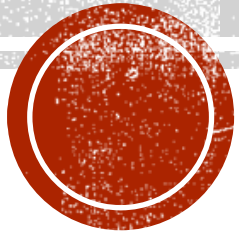
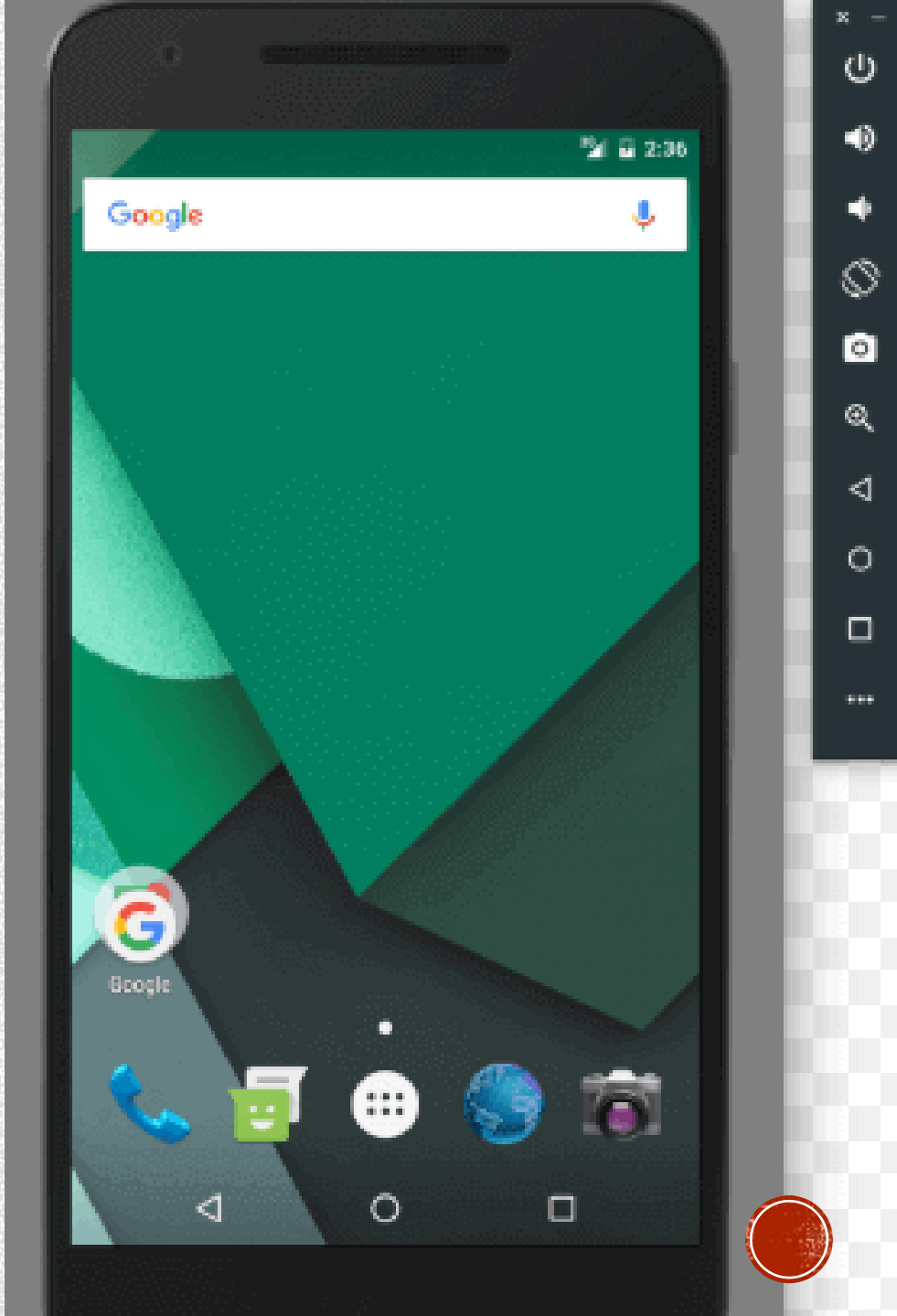
LECTURE # 02
INTRODUCTION TO AVD AND ADB

Software Application & Mobile Devices



ANDROID VIRTUAL DEVICE

- The **Android emulator** is an **Android Virtual Device (AVD),** which represents a specific Android device. We can use the Android emulator as a target device to execute and test our Android application on our PC. The Android emulator provides almost all the functionality of a real device. We can get the incoming phone calls and text messages. It also gives the location of the device and simulates different network speeds. Android emulator simulates rotation and other hardware sensors. It accesses the Google Play store, and much more

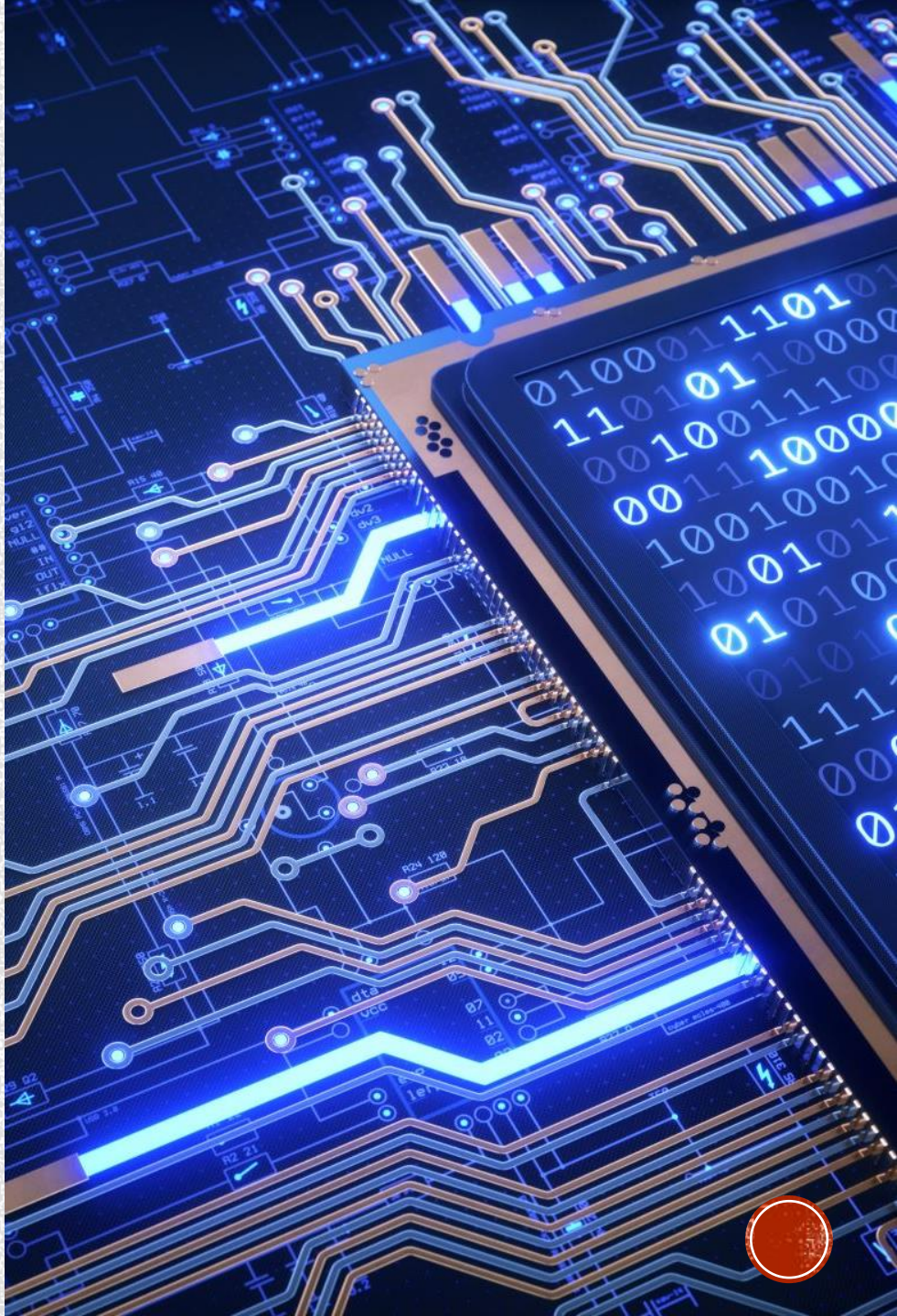


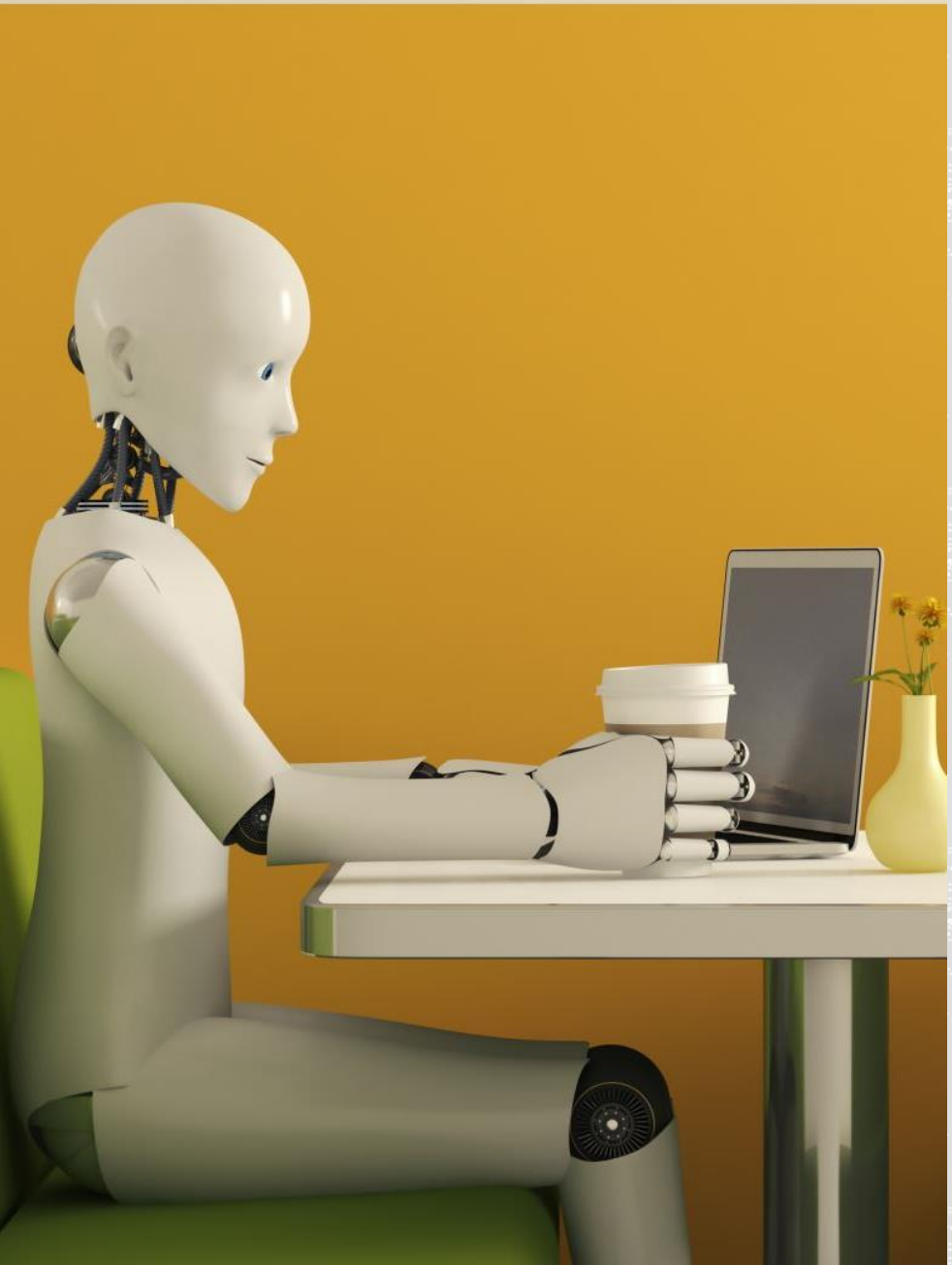
WHY WE USE ANDROID VIRTUAL DEVICES

- Testing Android applications on emulator are sometimes faster and easier than doing on a real device. For example, we can transfer data faster to the emulator than to a real device connected through USB.
- The Android emulator comes with predefined configurations for several Android phones, Wear OS, tablet, Android TV devices.

BENEFITS OF ANDROID VIRTUAL DEVICE

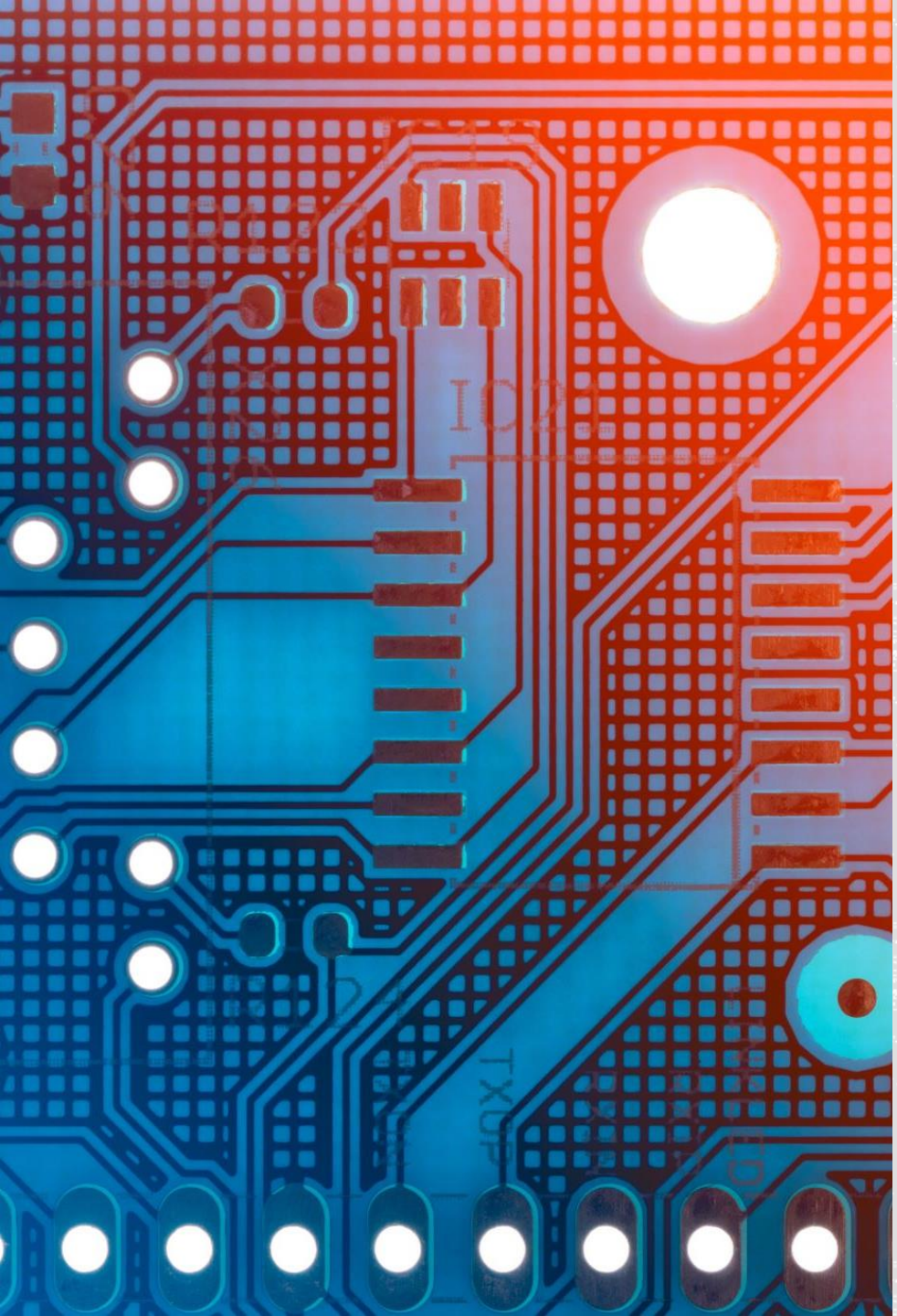
1. **Testing Convenience:** Easily test apps on different Android configurations.
2. **Cost Savings:** Avoid expenses of purchasing physical devices.
3. **Accessibility:** Available within Android SDK for easy setup.
4. **Flexibility:** Customize device configurations for thorough testing.
5. **Speed:** Quickly deploy apps for testing and debugging.
6. **Integrated Environment:** Seamlessly integrates with IDEs like Android Studio.
7. **Emulated Features:** Supports emulated hardware features for comprehensive testing.
8. **API Support:** Test apps against specific Android API levels.





SCENARIO

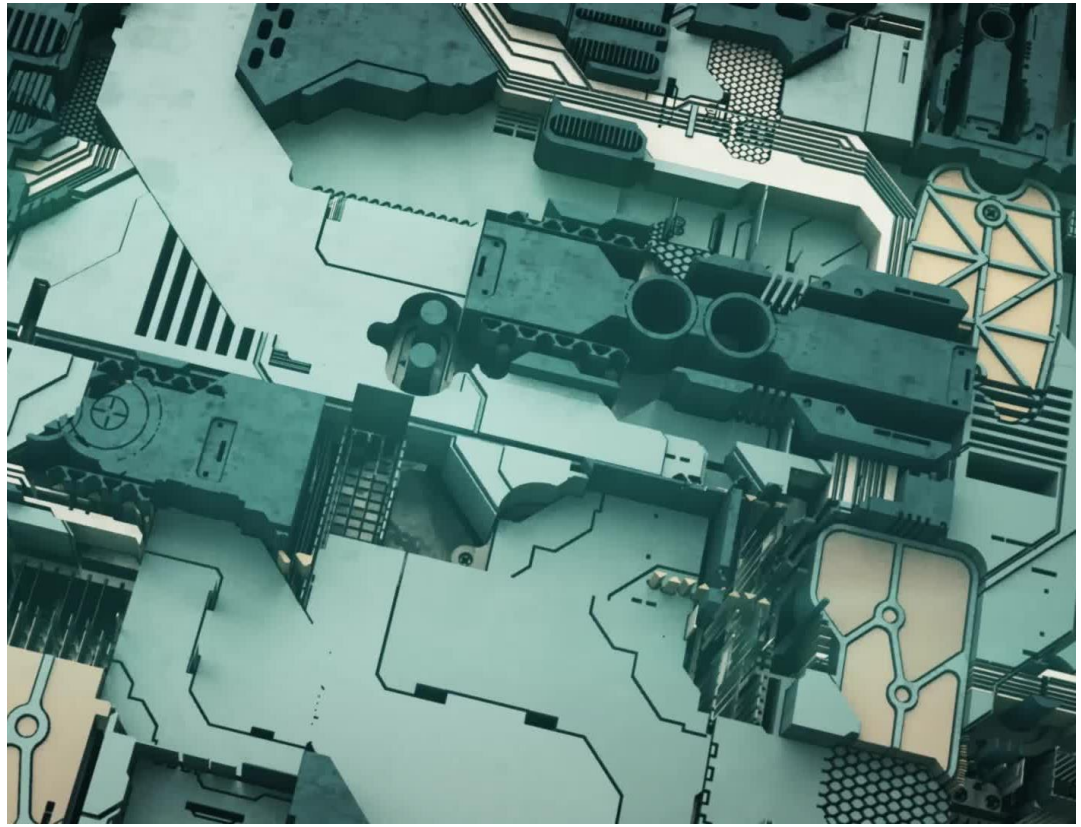
- Imagine you're a developer working on a new Android app. As you progress with your development, you encounter the need to test your app thoroughly across various device configurations and scenarios. This is where Android Virtual Device (AVD) comes into play.
- With AVD, you can simulate different screen sizes, resolutions, and device specifications right on your computer. This allows you to ensure that your app's user interface adapts correctly to various screen sizes, maintaining a consistent and responsive experience for users across different devices.
- Additionally, AVD supports testing against different Android API levels, enabling you to verify that your app functions properly on different versions of the Android platform. You can take advantage of new features introduced in newer API levels while ensuring backward compatibility with older versions, ultimately reaching a wider audience.



REQUIREMENTS AND RECOMMENDATIONS

- The Android emulator takes additional requirements beyond the basic system requirement for Android Studio. These requirements are given below:
- SDK Tools 26.1.1 or higher
- 64-bit processor
- Windows: CPU with UG (unrestricted guest) support
- HAXM 6.2.1 or later (recommended HAXM 7.2.0 or later)

INSTALL THE EMULATOR



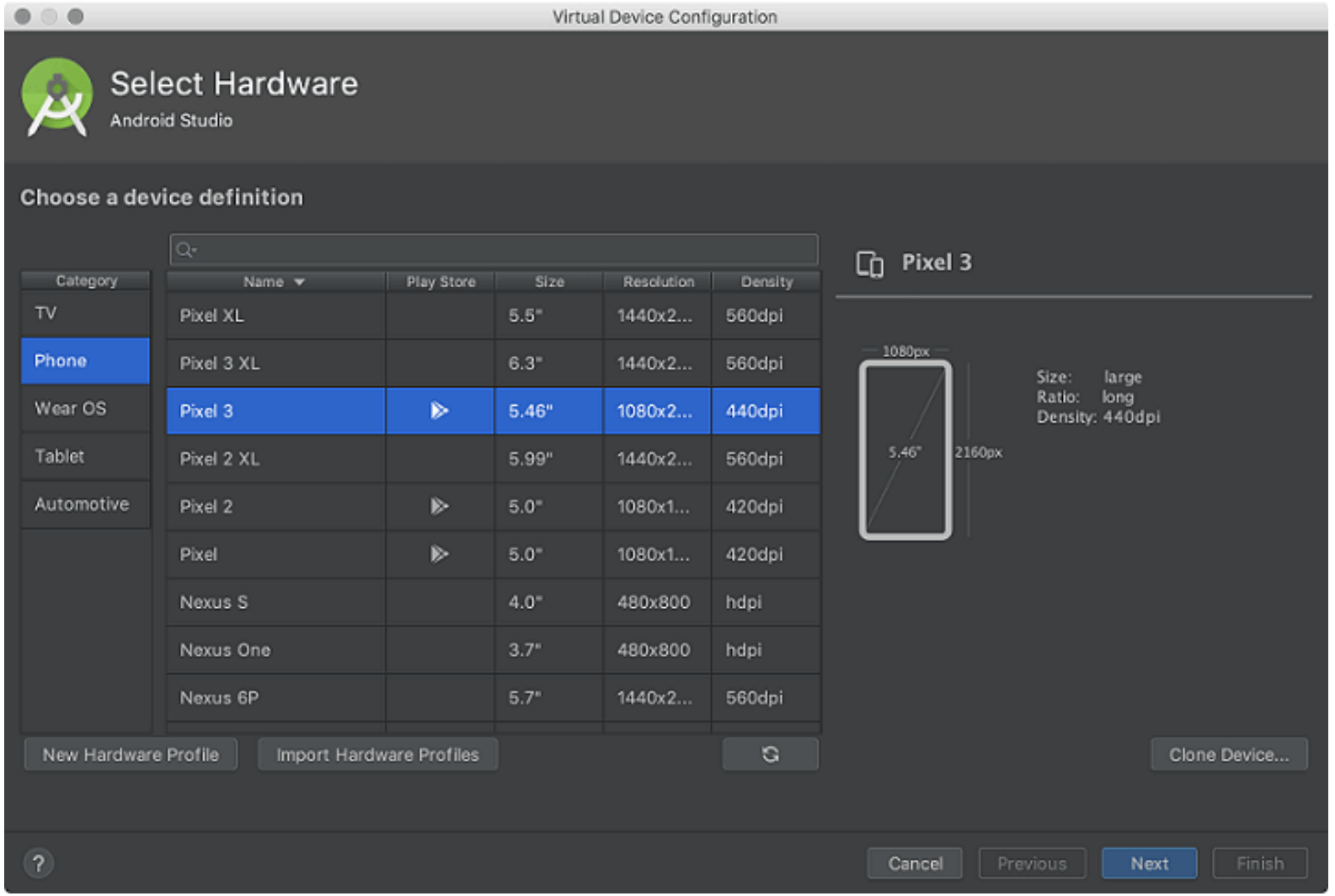
- The Android emulator is installed while installing the Android Studio. However, some components of emulator may or may not be installed while installing Android Studio. To install the emulator component, select the **Android Emulator** component in the **SDK Tools** tab of the **SDK Manager.**

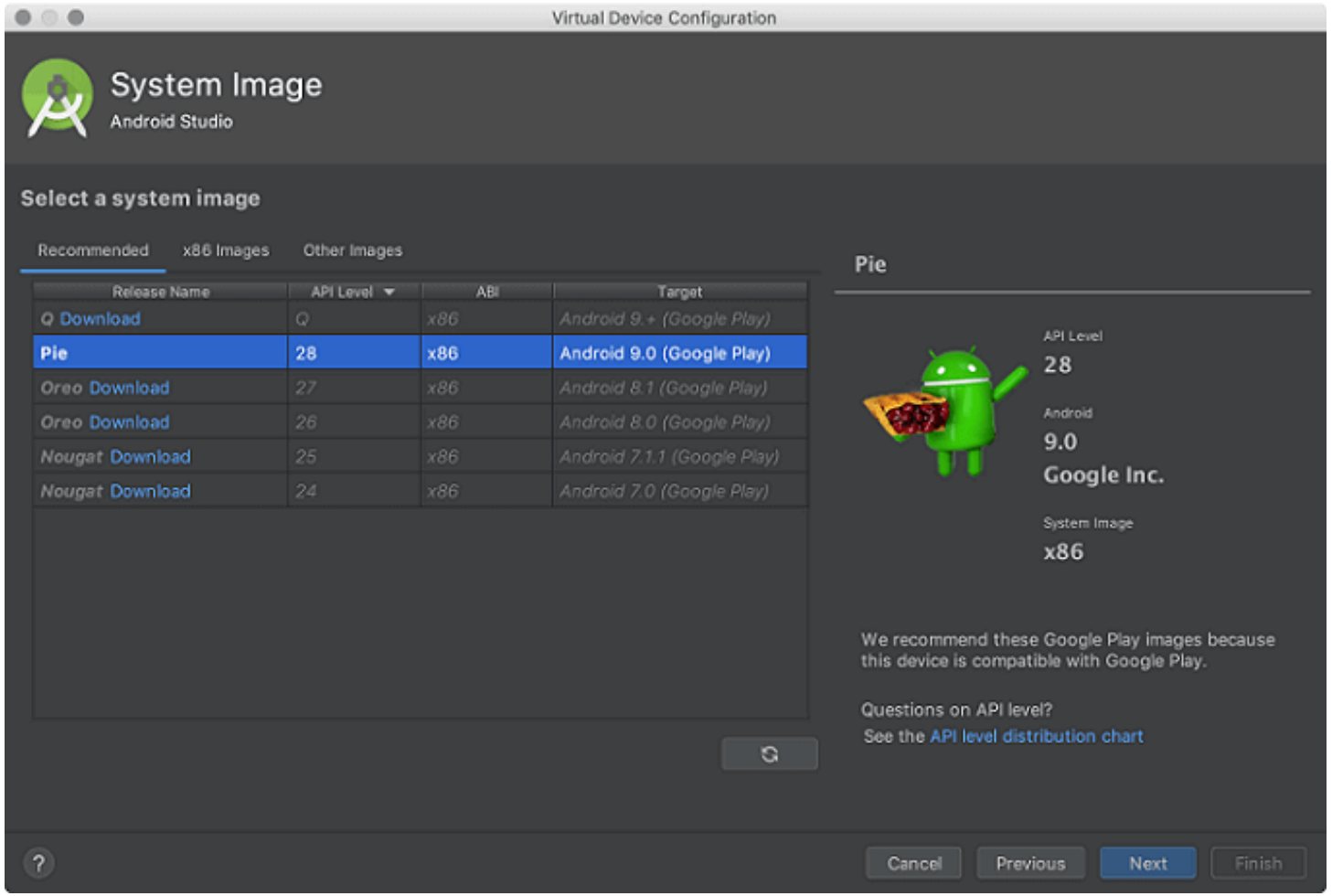
RUN APP ON EMULATOR:

- Run an Android app on the Emulator
- We can run an Android app form the Android Studio project, or we can run an app which is installed on the Android Emulator as we run any app on a device.
- To start the Android Emulator and run an application in our project:
- **1.** In **Android Studio,** we need to create an Android Virtual Device (AVD) that the emulator can use to install and run your app. To create a new AVD:-
- **1.1** Open the AVD Manager by clicking **Tools > AVD Manager.**

CONTINUE:

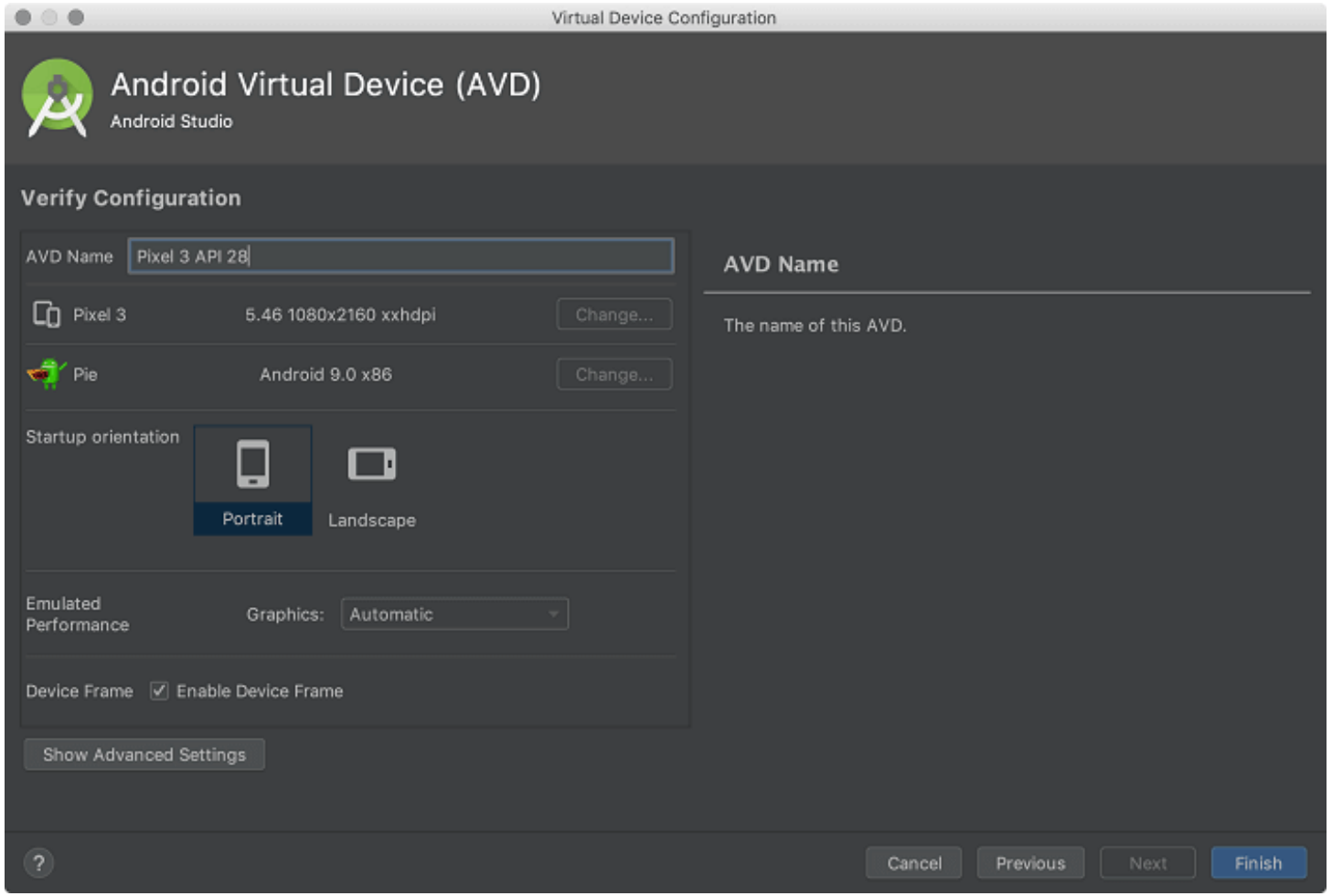
- **1.2** Click on Create **Virtual** Device, at the bottom of the AVD Manager dialog. Then **Select Hardware** page appears.





CONTINUE:

- **1.3** Select a hardware profile and then click **Next.** If we don't see the hardware profile we want, then we can create or import a hardware profile. The **System Image** page appears.

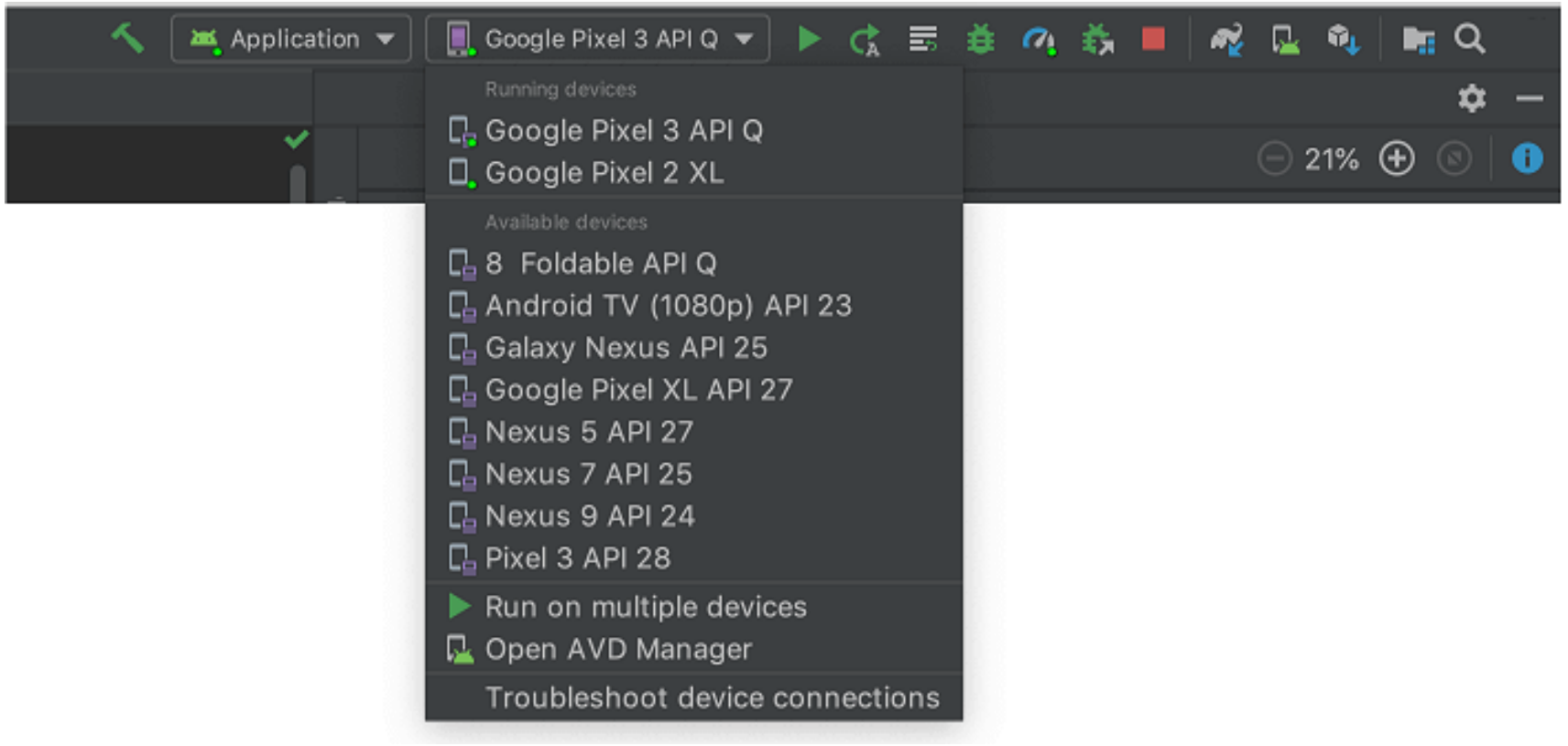


CONTINUE:

- **1.4** Select the system image for the API level and click **Next.** This leads to open a **Verify Configuration** page.

CONTINUE:

- **1.5** Change AVD properties if needed, and then click **Finish.**
- **2.** In the toolbar, choose the AVD, which we want to run our app from the target device from the drop-down menu.



RUN AND STOP AN EMULATOR & CLEAR DATA

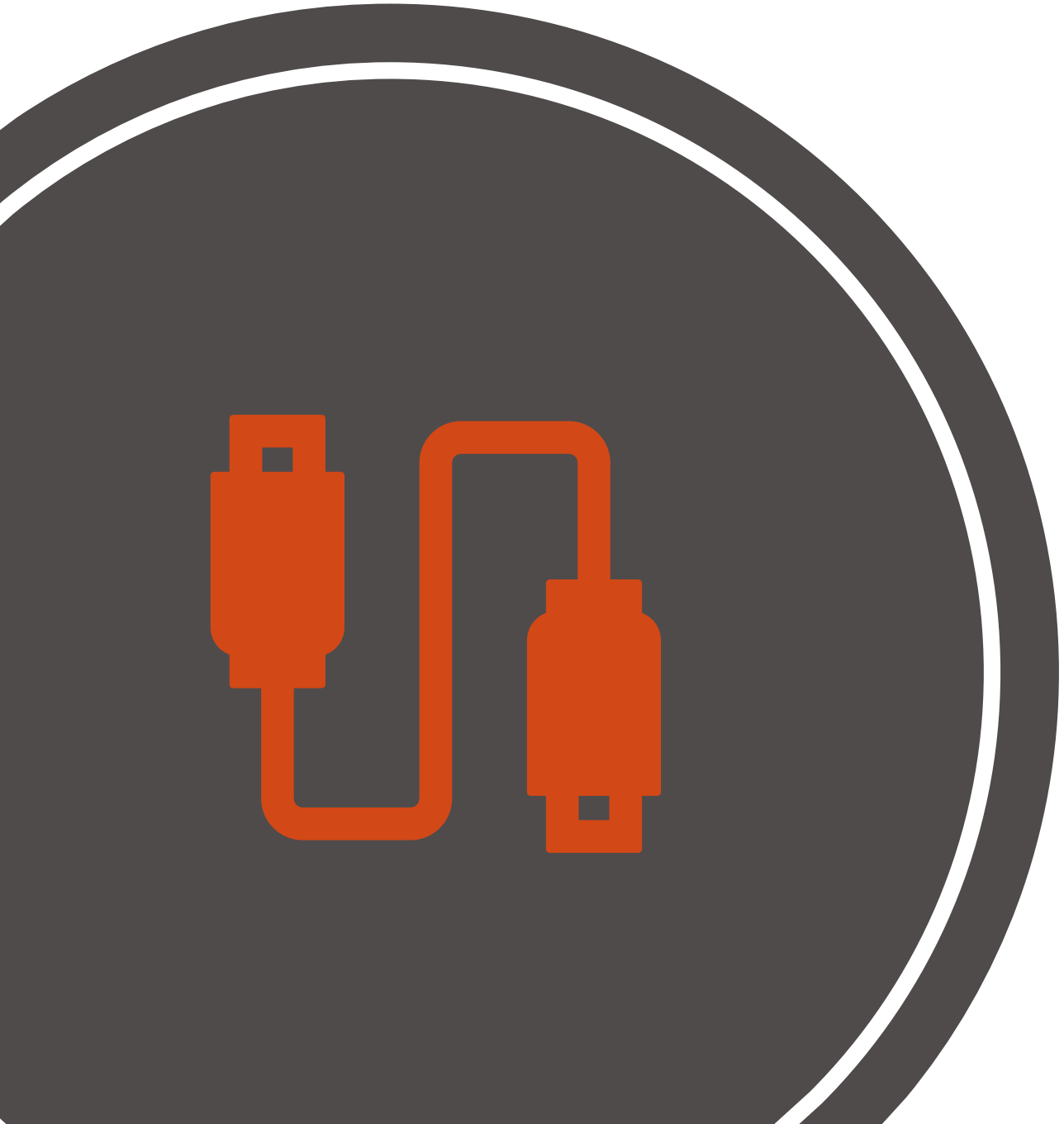
- To run an Android emulator that uses an AVD, double-click the AVD, or click **Launch**
- To stop the running emulator, right-click and select **Stop,** or click Menu ▼ and select Stop.
- If we want to clear the data from an emulator and return it to the initial state when it was first defined, then right-click an AVD and select **Wipe Data.** Or click menu ▼ and select **Wipe Data.**

INTRODUCTION TO ADB

- **Introduction to ADB (Android Debug Bridge):**
- Description: ADB, or Android Debug Bridge, is a command-line tool that allows developers to communicate with Android devices or emulators from a computer. It facilitates various tasks such as installing and debugging apps, accessing device shell, transferring files, and more.

INSTALLING APP USING ADB:

- **Installing Apps Using ADB:**
- Description: Installing apps using ADB provides a convenient way for developers to deploy and test their applications on Android devices or emulators directly from their development environment. Below are the steps to install an app using ADB:



ENABLE USB DEBUGGING:

On the Android device, navigate to Settings > Developer options and enable USB debugging.

CONNECT DEVICE:

- **Connect Device:** Connect the Android device to the computer using a USB cable. If prompted, allow USB debugging on the device.

VERIFY CONNECTIONS:

- **Verify Connection:** Open a command prompt or terminal window on the computer and run the following command to verify that the device is connected:

```
adb devices
```

- This command should display the connected device along with its unique identifier.

INSTALL APP:

- **Install App:** Use the following command to install the app on the connected device:

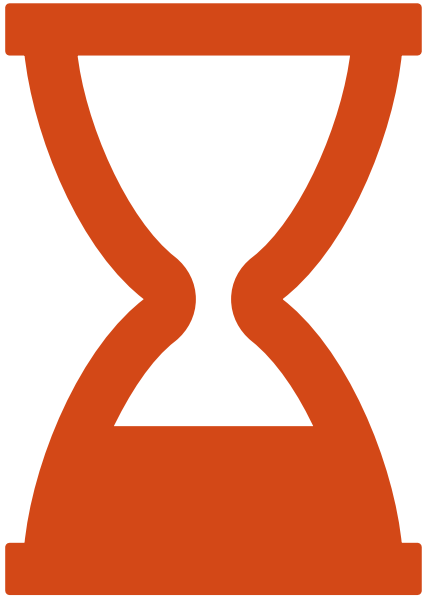
```
adb install path_to_your_app.apk
```

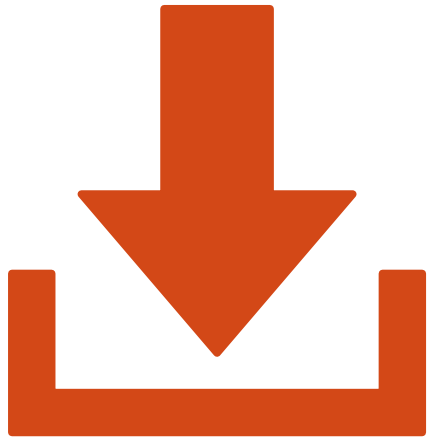
- Replace path_to_your_app.apk with the file path of the APK file you want to install.

WAIT & VERIFY FOR INSTALLATION

1. **Wait for Installation:** Wait for the installation process to complete. You'll see a success message in the command prompt or terminal window once the installation is finished.
2. **Verify Installation:** Check the device's app drawer or launcher to verify that the app has been successfully installed.

- Installing apps using ADB streamlines the deployment process and allows developers to quickly test their apps on real devices or emulators.





TASK: 1

- Setup AVD in your Device and Attach Screenshot in the file

TASK 2:

Install the Birthday Card Application in your Mobile devices using Android Debug Bridge