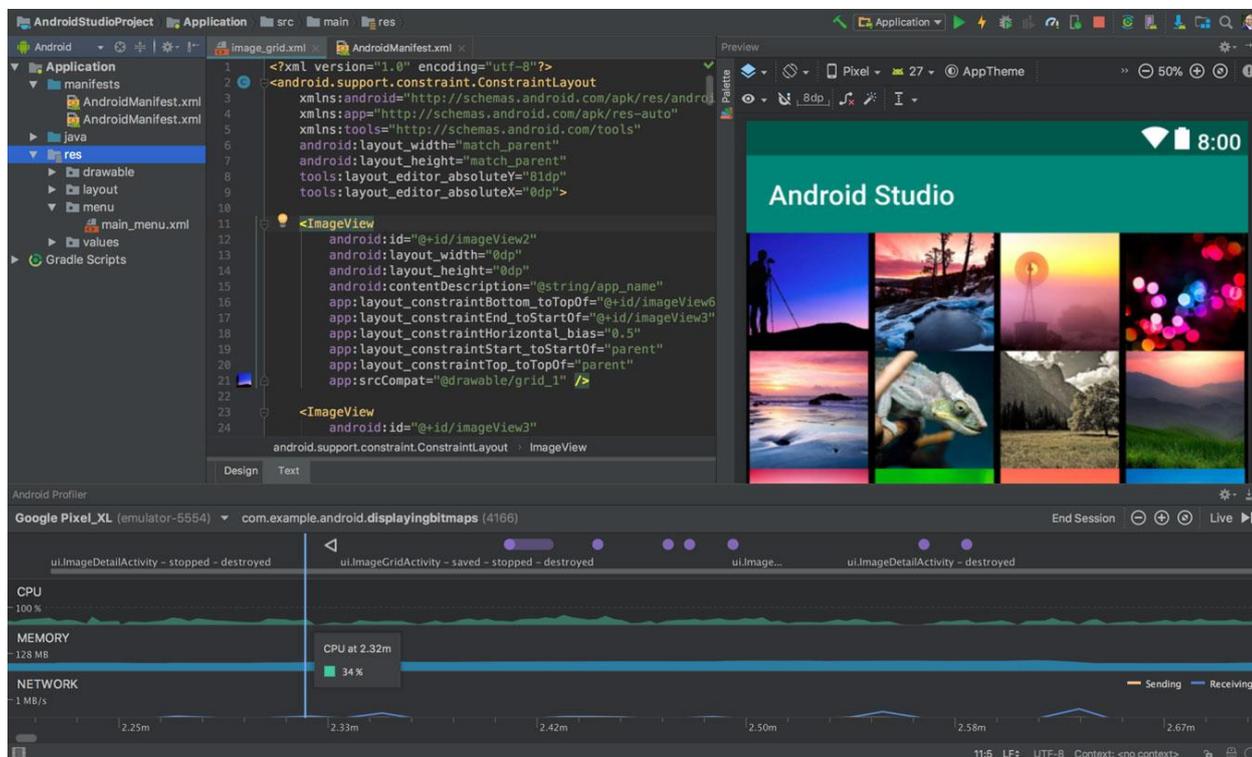


Создание простейшего приложения состоит из нескольких этапов:

- проект в Android Studio;
- создание пользовательского интерфейса;
- добавление активностей, навигации и действий;
- тест-драйв приложения в эмуляторе.

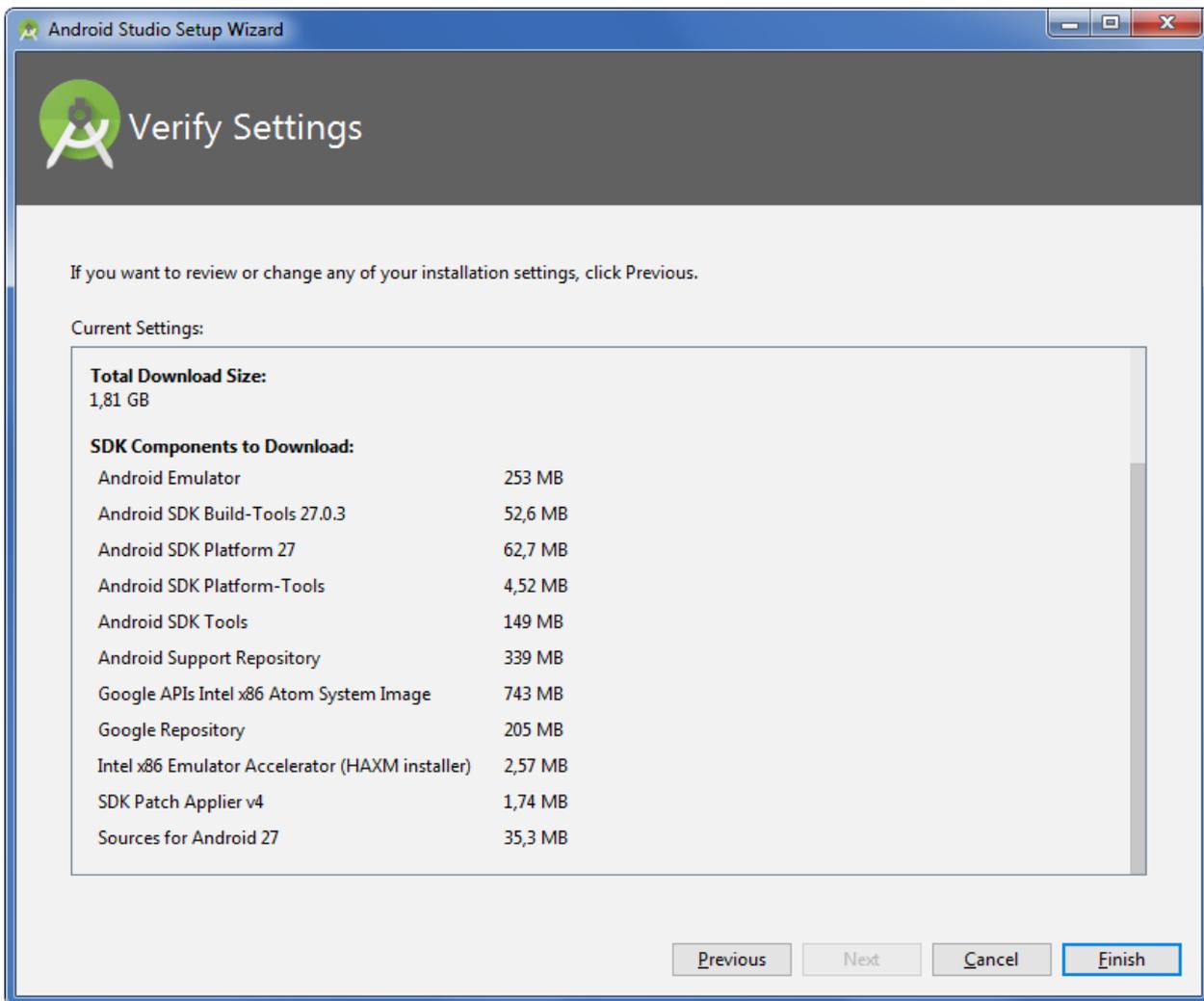
Необходимые инструменты

Первым делом установите программу Android Studio. Это официальная среда разработки (IDE) для Android, она работает на Windows, macOS и Linux. Хотя при разработке программ для Android можно использовать и другие среды, кроме Android Studio.

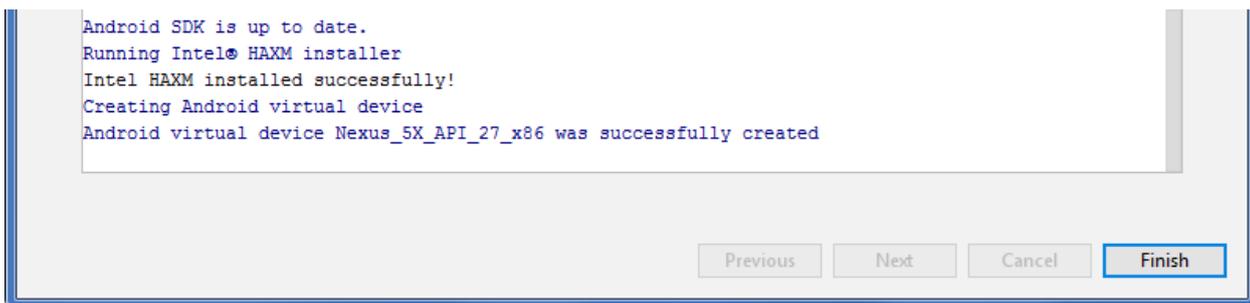


Если на компьютере не установлены **Android SDK** и другие компоненты, то **Android Studio** автоматически скачает их. Android SDK — это среда программирования, в которую входят библиотеки, исполняемые файлы, скрипты, документация и т.д.

Android SDK компилирует код вместе с любыми данными и ресурсами в файл с расширением **.apk**. Он содержит всё необходимое для установки приложения на Android-устройство.



Полезно установить и эмулятор *Android*, чтобы запускать и тестировать приложения. Эмулятор поставляется в комплекте с *Android Studio*.



Когда все инструменты установлены, можно создать первый проект. Но сначала нужно разобраться с основными понятиями.

Из чего состоит приложение на Android

Android-приложение состоит из четырёх компонентов. Каждый компонент — это точка входа, через которую система или пользователь может получить доступ.

- **Активность (activity)** — элементы интерактивного пользовательского интерфейса.
Одна активность задействует другую и передаёт информацию о том, что намерен делать пользователь, через класс Intent (намерения). Активности подобны веб-страницам, а намерения — ссылкам между ними. Запуск приложения — это активность Main.
- **Сервис (service)** — универсальная точка входа для поддержания работы приложения в фоновом режиме.
Этот компонент выполняет длительные операции или работу для удалённых процессов без визуального интерфейса.
- **Широковещательный приемник (broadcast receiver)** транслирует нескольким участникам намерения из приложения.
- **Поставщик содержимого (content provider)** управляет общим набором данных приложения из файловой системы, базы данных SQLite, интернета или другого хранилища.

Создаём Android-приложение в Android Studio

Шаг 1

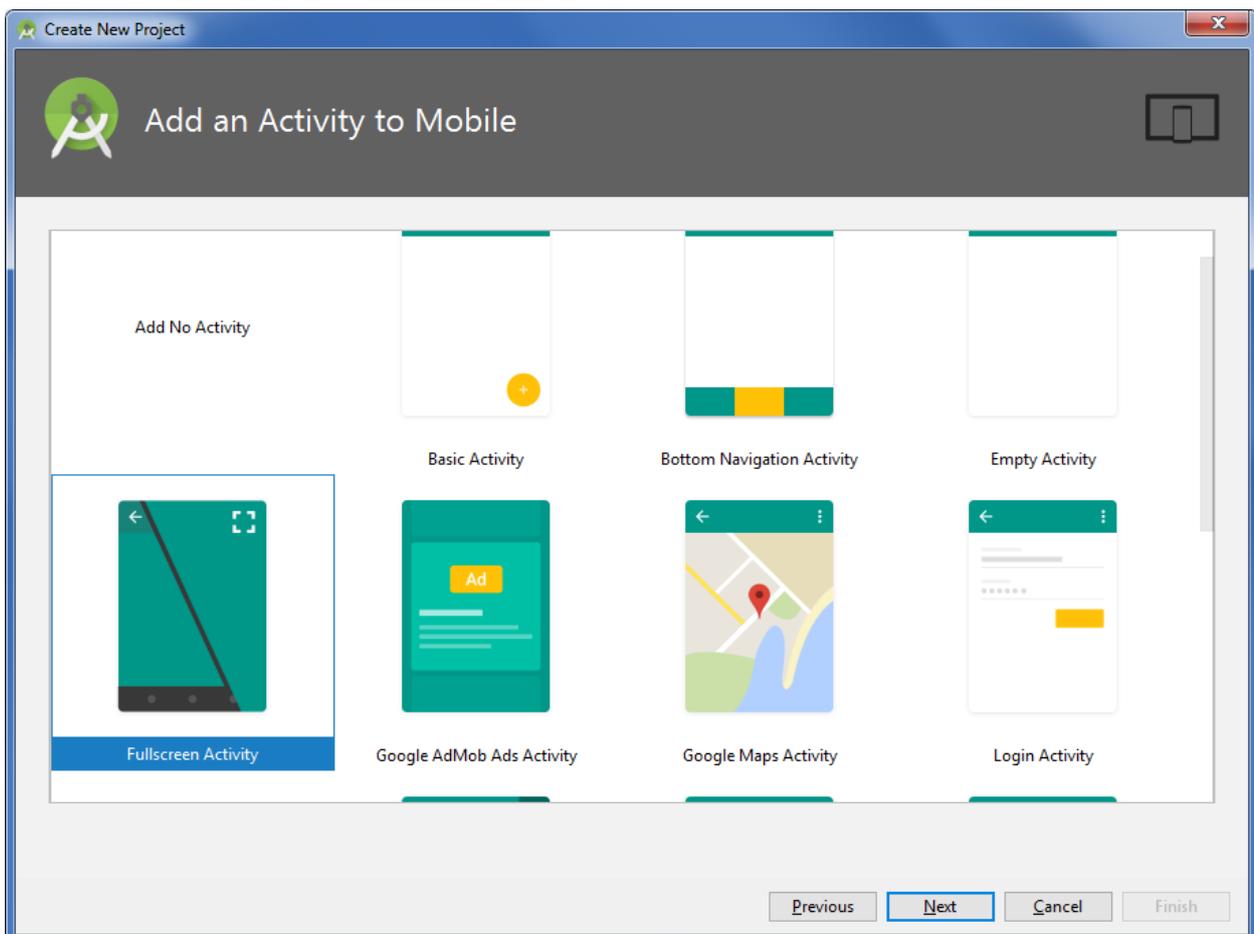
Выбираем название приложения, домен компании, путь к проекту и название пакета. Указываем, включить ли поддержку опциональных языков программирования C++ и Kotlin.

Шаг 2

Задаём одну или несколько целевых платформ для сборки. Для этого используется SDK и AVD, менеджер виртуальных устройств Android. Инструмент позволяет устанавливать в SDK пакеты, которые поддерживают несколько версий ОС Android и несколько уровней API (интерфейсов программирования приложений).

Шаг 3

Выбираем основную активность, которая будет запускаться при нажатии на иконку приложения, и даём ей имя.

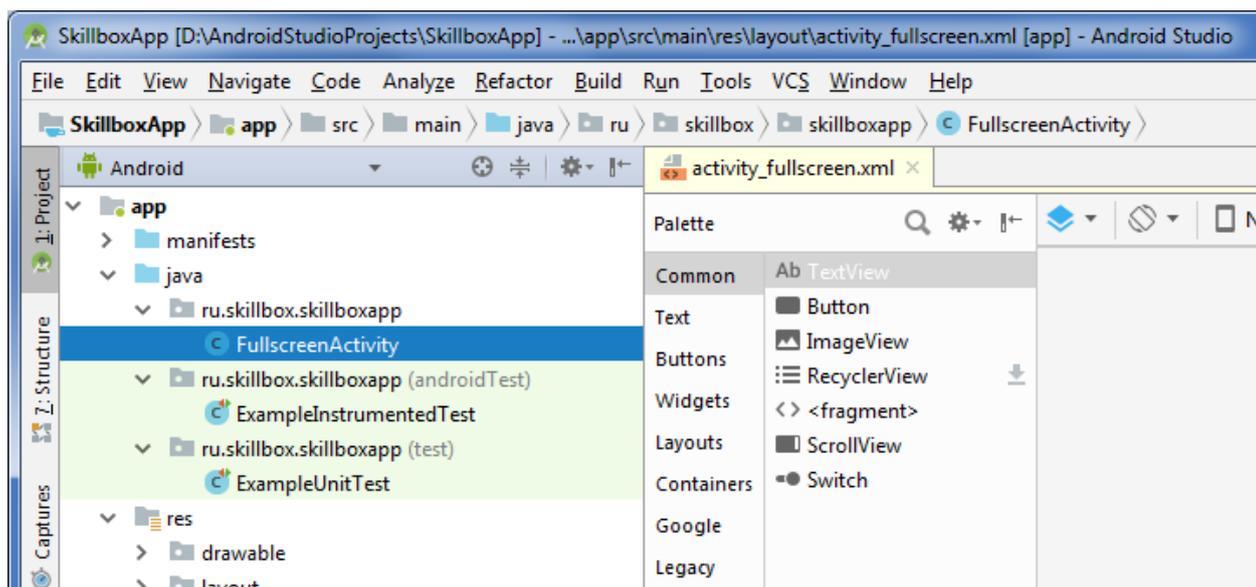


Шаг 4

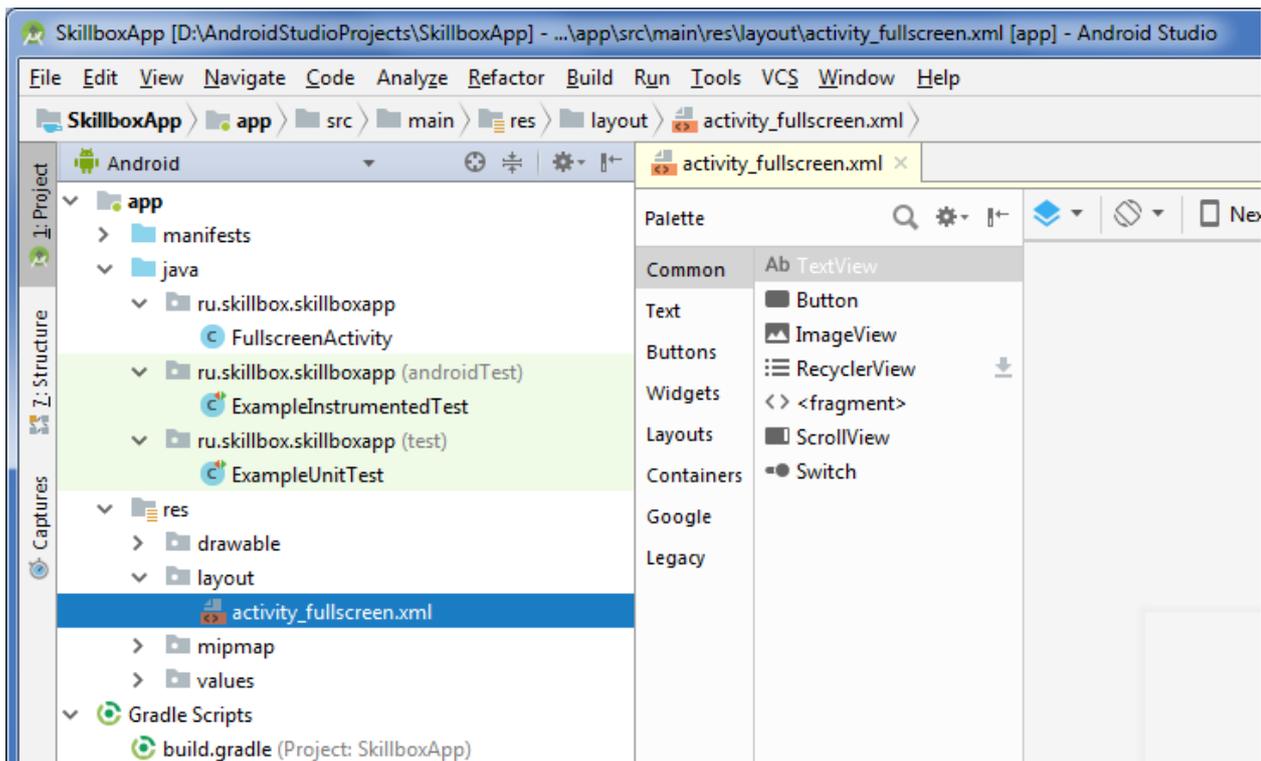
После нескольких минут сборки Android Studio открывает интерфейс IDE. Здесь три основных момента.

Если выбрать в выпадающем меню вид Android, то вы увидите файлы проекта. Например, наша основная активность называется **app > java > ru.skillbox.skillboxapp > FullscreenActivity**.

При создании проекта мы указали вместо активности *Main* полноэкранную активность.



Далее можно посмотреть файл **app > res > layout > activity_fullscreen.xml**. Это XML-файл с макетом для UI нашей основной активности.



Наконец, третий важный файл *app > manifests > AndroidManifest.xml* описывает фундаментальные характеристики приложения и определяет все его компоненты.

Содержимое манифеста

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ru.skillbox.skillboxapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity
            android:name=".FullscreenActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:label="@string/app_name"
            android:theme="@style/FullscreenTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>

```

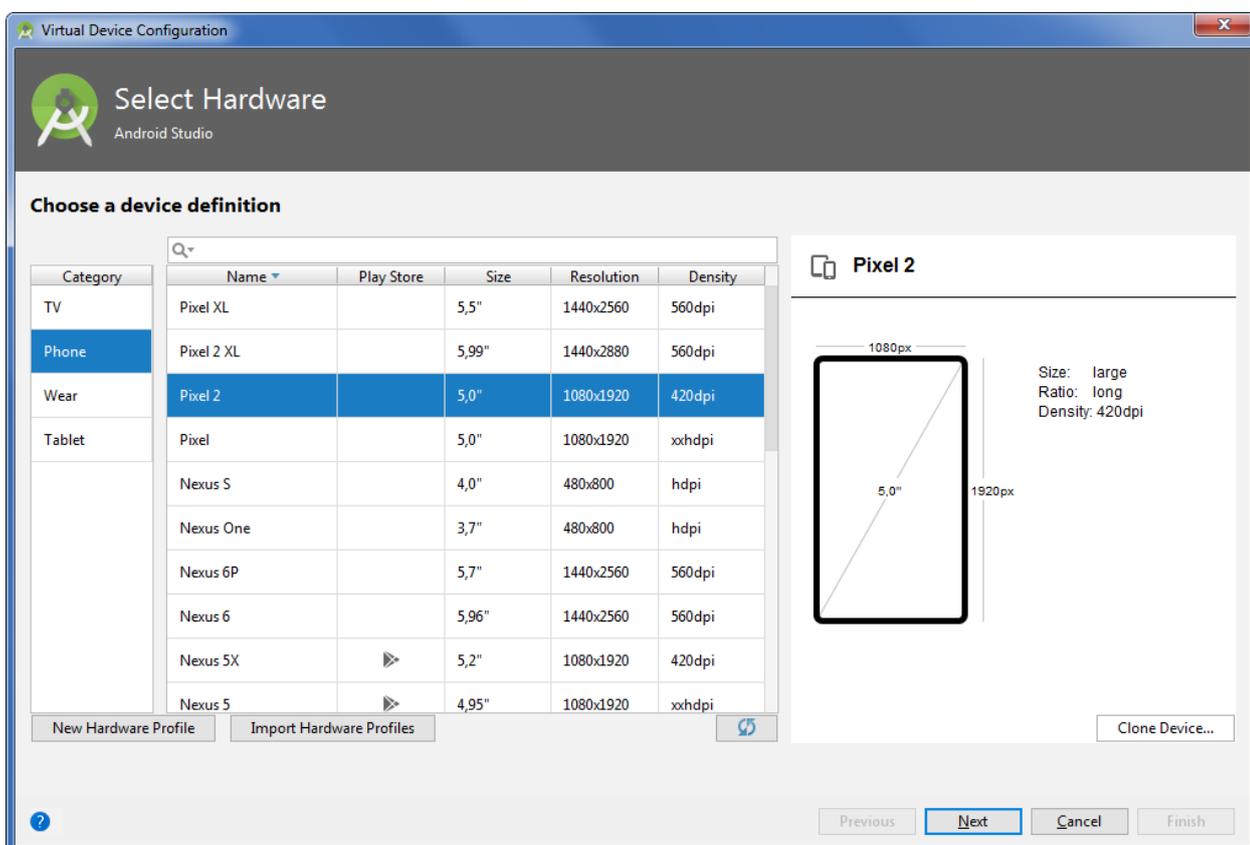
Созданное нами приложение — это одна активность, которая запускается в полноэкранном режиме и не имеет графических элементов.

Запускаем на Android-устройстве или в эмуляторе.

Смартфон или планшет для этого подключаем в режиме USB-отладки, которая активируется в **Настройках разработчика** в меню *Настройки*.

Для запуска в эмуляторе нажимаем в Android Studio кнопку **Run** в меню *Run* (Shift+F10).

Выбираем подходящее устройство и версию ОС, портретную или ландшафтную (альбомную) ориентацию.



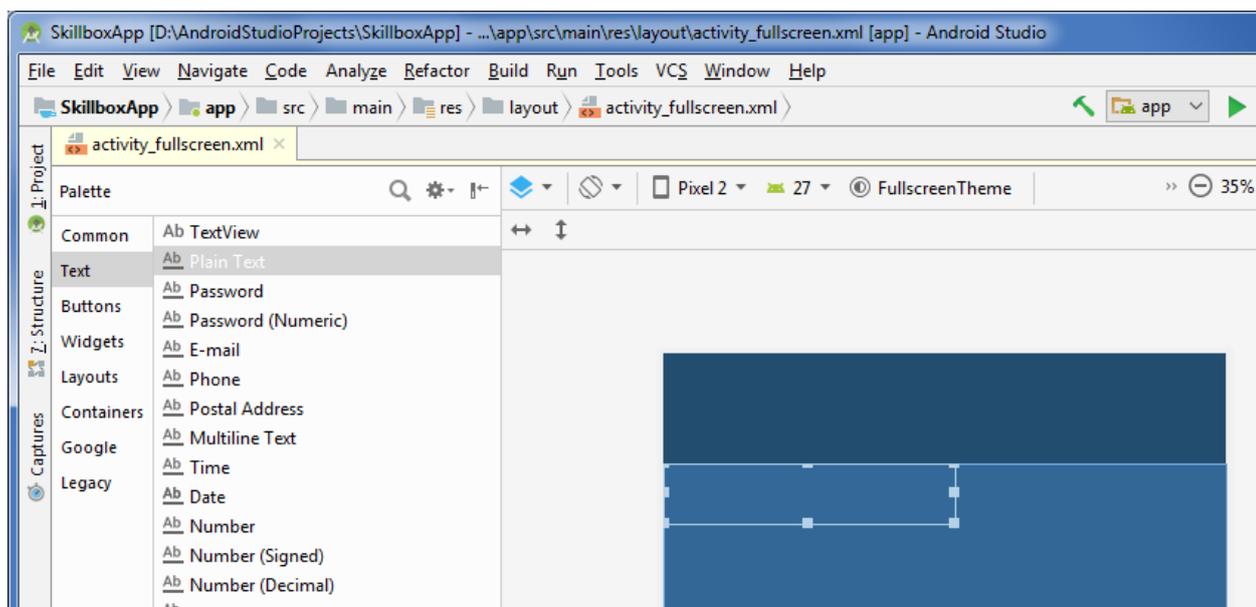
Android Studio установит эмулятор и запустит его.

Создаём простой пользовательский интерфейс

Пользовательский интерфейс Android-приложения создаётся через иерархию **макетов** (layouts, объекты ViewGroup) и **виджетов** (объекты View). Макеты управляют расположением дочерних виджетов на экране. Сами виджеты — это непосредственно компоненты UI: кнопки, текстовые поля на экране и т.п.

Интерфейс активностей создаётся в Android Studio в редакторе макетов (Layout Editor) и хранится по большей части в XML-файлах.

- Открываем файл **app > res > layout > activity_fullscreen.xml**.
- Добавляем на экран из палитры (Palette) виджеты перетаскиванием мышью.
- Например, берём текстовое поле (PlainText). Это виджет EditText, куда пользователь может вводить текст.
- Добавляем кнопки и другие нужные элементы.



Также на экран можно перетащить кнопки и другие элементы.

Добавляем действия, активности и навигацию

Предположим, мы создали активность с текстовым полем и кнопкой «Отправить». После этого нужно написать, что конкретно будет происходить при нажатии кнопки «Отправить».

- Заходим в код **app > java > FullscreenActivity**.
- Добавляем метод *SendMessage()* в класс **FullscreenActivity**, чтобы при нажатии на кнопку вызывался этот метод.
- Создаём намерения (класс Intent) для перехода от одной активности к другой, новые активности, навигацию и всё остальное, что необходимо для приложения.