**Test it on multiple operating systems**

**Test it on multiple devices with different screen sizes**

**Build and deploy to Android by:**

1. Plugging in the device set to developer mode
2. Pressing the run button in the lower left

**Use anchors to position elements**

<https://doc.qt.io/qt-6.2/qtquick-positioning-anchors.html>

**Fee of $25 to put an app on the Play Store**

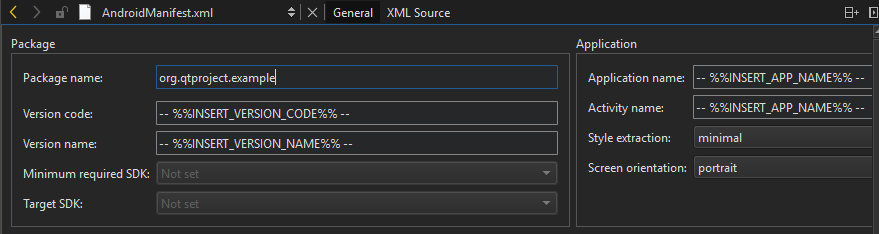
**Need to have a “Data Collection & Privacy” policy**

**Most smartphones are powered by ARM**

**Will have 4 versions of each program (INTEL/ARM & 32/64-bit)**

**Set name and version of the program as well as package name in AndroidManifest.xml file**

Projects -> Android builds... -> Build Steps -> Build Android APK -> Application -> Create Templates



Advised to leave version code and version name as it is

Advised to change application and package name

Enable the **USB Debugging** option under **Settings > Developer options**.

For Android 4.2 and newer, **Developer options** is hidden by default; use the following steps:

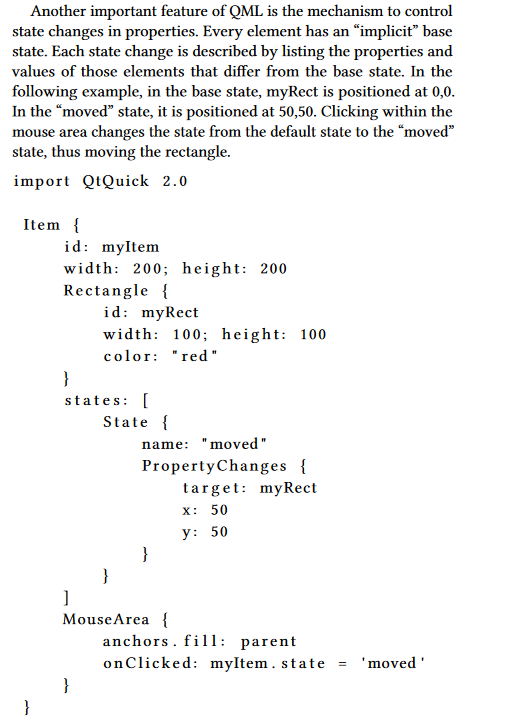
1. On the device, go to **Settings > About <device>**.
2. Tap the **Build number** seven times to make **Settings > Developer options** available.
3. Then enable the **USB Debugging** option.

<https://doc.qt.io/qt-5/qml-multimedia.html>

<https://doc.qt.io/qt-5/qml-qtmultimedia-videooutput.html>

<https://doc.qt.io/qt-6/qml-qtmultimedia-mediaplayer.html>

<https://doc.qt.io/qt-6/qjsengine.html>

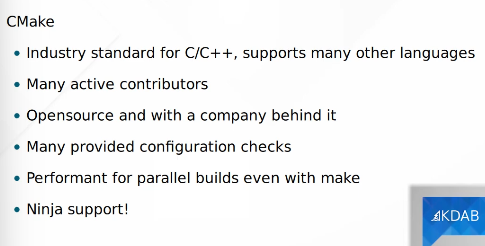


A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated



If you’re going

to exclusively target desktop operating systems like Windows, Linux and Mac, then Qt Widgets are

your best bet. If you think that you’ll need to build your application for mobile targets like Android

and IoS, or even for some embedded devices where people expect highly fluid and dynamic user

interfaces, then QML is the way to go