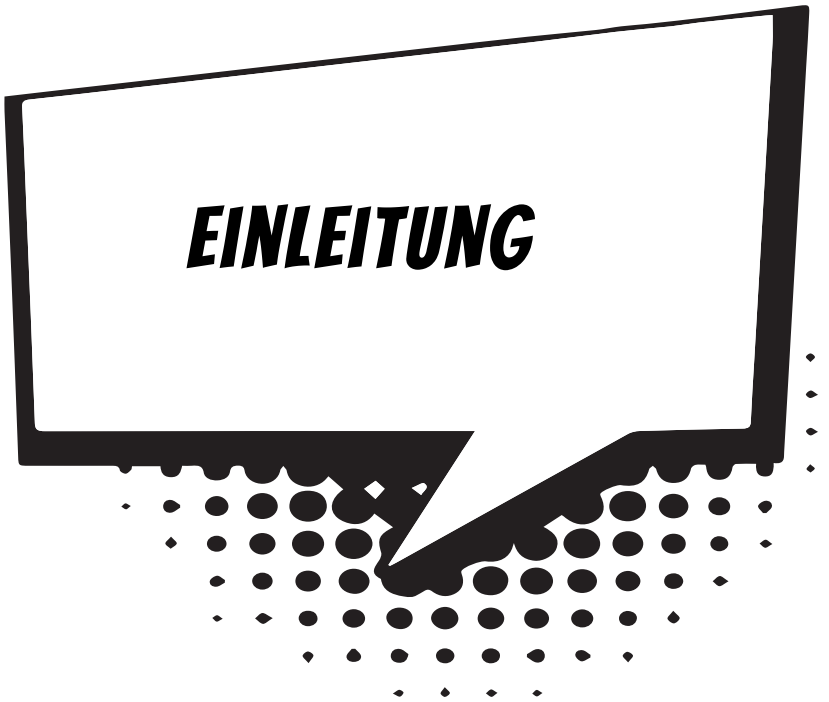
HANS-GEORG SCHUMANN

SMARTPHONE-APPS ENTWICKELN
OHNE VORKENNTNISSE

INHALT



EINLEITUNG

Android? Ist das nicht so etwas wie eine Mensch-Maschine? Ein Roboter, der einem Menschen (täuschend) ähnlich sieht? Ja, und gleichzeitig der Name eines Betriebssystems, das **Google** entwickelt hat und kostenfrei zur Verfügung stellt.

Während man Windows vorwiegend auf »großen« PCs findet, ist Android auf Smartphones und Tablets am meisten verbreitet. Rund Dreiviertel aller Handys (wie man sie auch immer noch nennt) laufen mit dem Android-System.

Computer sind sie eigentlich alle: Geräte, die man auf oder unter den Tisch stellt (Desktop-PCs), Geräte, die man mitnehmen kann (Notebooks und Tablets), und dann noch eine Nummer kleiner: die Smartphones.

Vorwiegend um Android für Smartphones geht es hier. Die meisten Projekte funktionieren aber auch auf Tablets. Programme für dieses System werden meisten kurz **Apps** genannt (als Abkürzung für Applikationen).

WAS HEIßT EIGENTLICH PROGRAMMIEREN?

Wenn du aufschreibst, was ein Computer tun soll, nennt man das **Programmieren**. Das Tolle daran ist, dass du selbst bestimmen kannst, was getan werden soll. Lässt du dein Programm laufen, macht der Computer die Sachen, die du ausgeheckt hast. Natürlich wird er dann nicht dein Zimmer aufräumen und dir auch keine Tasse Kakao ans Bett bringen. Aber kannst du erst mal programmieren, kannst du den Computer sozusagen nach deiner Pfeife tanzen lassen.

Allerdings passiert es gerade beim Programmieren, dass der Computer nicht so will, wie du es gerne hättest. Meistens ist das ein Fehler im Programm. Das Problem kann aber auch irgendwo anders im Computer oder im Betriebssystem liegen. Das Dumme bei Fehlern ist, dass sie sich gern so gut verstecken, dass die Suche danach schon manchen Programmierer zur Verzweiflung gebracht hat.

Vielleicht hast du nun trotzdem Lust bekommen, das Programmieren zu erlernen. Und ausgerechnet noch für die »ganz Kleinen«. Dann brauchst du ja nur noch eine passende **Entwicklungsumgebung**, und schon kann's losgehen.

WAS IST EINE ENTWICKLUNGSUMGEBUNG?

Um ein Programm zu erstellen, musst du erst etwas eintippen. Das ist wie bei einem Brief oder einer Geschichte, die man schreibt. Das Textprogramm dafür kann sehr einfach sein, weil es ja nicht auf eine besondere Schrift oder Darstellung ankommt wie bei einem Brief oder einem Referat. So etwas wird **Editor** genannt.

Ist das Programm eingetippt, kann es der Computer nicht einfach lesen und ausführen. Jetzt muss es so übersetzt werden, dass der PC versteht, was du von ihm willst. Weil er aber eine ganz andere Sprache spricht als du, muss ein Dolmetscher her.

Du programmierst in einer Sprache, die du verstehst, und der Dolmetscher übersetzt es so, dass es dem Computer verständlich wird. So etwas heißt dann **Compiler**.

Im Prinzip kann man für Android mehrere Programmiersprachen benutzen, am meisten verbreitet ist Java. Hier gibt es als Dolmetscher die **Java Virtual Machine** (kurz JVM). Diese Art »Zwischencomputer« lässt sich kostenlos installieren.

Eigentlich wird ein Java-Programm also an die JVM weitergereicht, die es dann für den jeweiligen Computer passend zubereitet: Das kann ein PC z.B. mit Windows sein. Oder ein Smartphone z.B. mit Android. Ein und dasselbe Java-Programm kann so im Prinzip auf jedem beliebigen Gerät funktionieren, das über eine JVM verfügt.

Schließlich müssen Programme getestet, überarbeitet, verbessert, wieder getestet und weiterentwickelt werden. Dazu gibt es noch einige zusätzliche Hilfen. Daraus wird dann ein ganzes System, die Entwicklungsumgebung.

WARUM GERADE JAVA?

Leider kannst du nicht einfach programmieren, wie dir der Mund gewachsen ist. Eine **Programmiersprache** muss so aufgebaut sein, dass möglichst viele Menschen in möglichst vielen Ländern einheitlich damit umgehen können.

Weil in der ganzen Welt Leute zu finden sind, die wenigstens ein paar Brocken Englisch können, besteht auch fast jede Programmiersprache aus englischen Wörtern. Es gab auch immer mal Versuche, z.B. in Deutsch zu programmieren, aber meistens klingen die Wörter dort so künstlich, dass man lieber wieder aufs Englische zurückgreift.

Eigentlich wäre es egal, welche Programmiersprache du benutzt. Am besten eine, die möglichst leicht zu erlernen ist. Doch in der Android-Welt ist die Nummer 1 die Programmiersprache **Java**, mit der du es auch in diesem Buch zu tun hast. Diese gehört inzwischen zu den am meisten verbreiteten Sprachen im Computer-Bereich. Sie ist nicht einfach, aber auch für Anfänger geeignet, die mit Java ihre erste Programmiersprache lernen wollen. Und: Es ist eine Sprache, die Smartphones und Tablets mit Android gut verstehen.

In letzter Zeit hat sich mit **Kotlin** eine neue Sprache verbreitet, die man auch unter Android Studio benutzen kann. Wir bleiben hier jedoch bei Java.

Der Weg zum guten Programmierer kann ganz schön steinig sein. Nicht selten kommt es vor, dass man die Lust verliert, weil einfach gar nichts klappen will. Das Programm tut etwas ganz anderes, man kann den Fehler nicht finden und man fragt sich: Wozu soll ich eigentlich programmieren lernen, wo es doch schon genug Apps gibt? Und dann noch ausgerechnet für Android. Aber du verspürst da einen Reiz, eigene Apps zu schreiben? Es ist also nicht nur einen Versuch wert, es kann sich durchaus lohnen, das Programmieren zu erlernen.

ANDROID STUDIO, DIE ENTWICKLUNGSUMGEBUNG ZUM BUCH

Um den Kauf einer Entwicklungsumgebung für Java und für Android musst du dich nicht kümmern, beides bekommst du kostenlos aus dem Internet.

Google stellt dir **Android Studio** zur Verfügung, eine komfortable und leistungsstarke Entwicklungsumgebung, mit der du unter allen Versionen von Windows programmieren kannst. Android Studio macht es auch möglich, Smartphones und Tablets zu simulieren und damit alle Apps erst mal auf dem PC unter Windows zu testen.

Wie man Android Studio und Java aus dem Internet holt und installiert, erfährst du in **Anhang A**.

UND WAS BIETET DIESES BUCH?

Über eine ganze Reihe von Kapiteln verteilt lernst du

- die Grundlagen der Programmierung kennen
- mit Android Studio unter Windows umzugehen
- mit Komponenten zu arbeiten (das sind Bausteine, mit denen du dir viel Programmierarbeit sparen kannst)
- auch komplexere Programm-Elemente einzusetzen
- eine ganze Reihe von Spiel-Projekten zu erstellen
- wie man Apps für den Play Store vorbereitet

Im **Anhang** gibt es dann noch zusätzliche Informationen und Hilfen, u.a. über Installationen und den Umgang mit Fehlern.

WIE ARBEITE ICH MIT DIESEM BUCH?

Grundsätzlich besteht dieses Buch aus einer Menge Text mit vielen Abbildungen dazwischen. Natürlich habe ich mich bemüht, alles so zuzubereiten, dass daraus lauter gut verdauliche Happen werden. Damit das Ganze noch genießbarer wird, gibt es zusätzlich noch einige Symbole, die ich dir hier gern erklären möchte:

ARBEITSSCHRITTE

≫ Wenn du dieses Zeichen siehst, heißt das: Es gibt etwas zu tun. Damit kommen wir beim Programmieren Schritt für Schritt einem neuen Ziel immer näher.

Grundsätzlich lernt man besser, wenn man einen Programmtext selbst eintippt oder ändert. Aber nicht immer hat man große Lust dazu. Deshalb gibt es alle Projekte im Buch auch als Download:

http://www.mitp.de/899

Und hinter einem Programmierschritt findest du auch den jeweiligen Namen des Projekts oder einer Datei (z.B. → *GameView.java*). Wenn du also das Projekt nicht selbst erstellen willst, kannst du stattdessen den passenden Ordner laden (zu finden im Download-Paket).

AUFGABEN

Am Ende eines Kapitels findest du jeweils eine Reihe von Fragen und Aufgaben. Diese Übungen sind nicht immer ganz einfach, aber sie helfen dir, noch besser zu programmieren. Lösungen zu den Aufgaben findest du in verschiedenen Formaten ebenfalls im Verzeichnis *Projekte*. Du kannst sie dir alle im Editor von Windows oder auch in deinem Textverarbeitungsprogramm anschauen. Oder du lässt sie dir ausdrucken und hast sie dann schwarz auf weiß, um sie neben deinen Computer zu legen. (Auch die Programme zu den Aufgaben findest du im Download-Paket.)

NOTFÄLLE

Vielleicht hast du irgendetwas falsch gemacht oder etwas vergessen. Oder es wird gerade knifflig. Dann fragst du dich, was du nun tun sollst. Bei diesem Symbol findest du eine Lösungsmöglichkeit. Notfalls kannst du aber auch ganz hinten im **Anhang B** nachschauen, wo einige Hinweise zur Pannenhilfe aufgeführt sind.

WICHTIGE STELLEN IM BUCH

Hin und wieder findest du ein solch dickes Ausrufezeichen im Buch. Dann ist das eine Stelle, an der etwas besonders Wichtiges steht.

EXPERTENWISSEN

Wenn du ein solches »Wow« siehst, geht es um ausführlichere Informationen zu einem Thema.

WAS BRAUCHST DU FÜR DIESES BUCH?

Installiert wird Java in ein Verzeichnis deiner Wahl, z.B. *C:\Programme\Java*. Auch Android Studio muss installiert werden und auch hier kannst du das Verzeichnis selbst bestimmen, z.B. C:*Programme**Android*. Zusätzlich empfiehlt es sich, auf einem anderen Laufwerk einen Arbeitsordner für die Projekte einzurichten, z.B. *D:\Projekte*.

Die Beispielprogramme in diesem Buch gibt es alle als Download von der Homepage des Verlages, falls du keine Lust zum Abtippen hast:

http://www.mitp.de/899

Und auch die Lösungen zu den Fragen und Aufgaben sind dort untergebracht.

BETRIEBSSYSTEM

Die meisten Computer arbeiten heute mit dem Betriebssystem Windows. Davon brauchst du eine Version zwischen 7 und 10. Außerdem wäre es gut, ein Smartphone oder Tablet mit Android zur Hand zu haben. Du kannst aber auch nur einen **Emulator** von Android Studio benutzen. Er reicht für die Mehrzahl der Projekte aus (es gibt aber einige wenige, die nur mit dem Smartphone oder Tablet funktionieren).

SPEICHERMEDIEN

Auf jeden Fall benötigst du etwas wie einen USB-Stick oder eine SD-Card, auch wenn du deine Programme auf die Festplatte speichern willst. Auf einem externen Speicher sind deine Arbeiten auf jeden Fall zusätzlich sicher aufgehoben.

Gegebenenfalls bitte deine Eltern oder Lehrer um Hilfe.

HINWEISE FÜR LEHRER

Dieses Buch lässt sich natürlich auch für den Informatik-Unterricht verwenden. Dort setzt natürlich jeder Lehrer seine eigenen Schwerpunkte. Benutzen Sie an Ihrer Schule bereits ein Werk aus einem Schulbuchverlag, so können Sie dieses Buch in Ergänzung zu dem vorhandenen Schulbuch einsetzen. Weil wir hier sozusagen »bei null« beginnen, ist auch ein direkter Einstieg in Java möglich – ohne irgendwelche anderen Programmierkenntnisse.

Der wesentliche Schwerpunkt in diesem Buch ist die Programmierung für das vor allem bei Smartphones weitverbreitete Betriebssystem Android. Dabei kommt auch die objektorientierte Programmierung nicht zu kurz.

In den Projekten werden die wesentlichen Elemente des Java-Wortschatzes wie auch die wichtigsten Komponenten für eine Android-Applikation (App) eingesetzt. In den Lösungen zu den Aufgaben finden Sie weitere Vorschläge zur Programmierung.

AUF DIE DATEIEN ZUM BUCH VERZICHTEN?

Vielleicht ist es Ihnen lieber, wenn Ihre Schüler die Projekte alle selbst erstellen. Dann lassen Sie die Download-Dateien einfach (erst einmal) weg.

ÜBUNGSMEDIEN

Für den Informatik-Unterricht sollte jeder Schüler ein anderes externes Speichermedium haben, um darauf seine Programmierversuche zu sichern. So wird verhindert, dass sich auf der Festplatte des Schulcomputers mit der Zeit allerlei »Datenmüll« ansammelt. Außerdem dient der eigene Datenträger dem Datenschutz: Nur der betreffende Schüler kann seine Daten manipulieren.

REGELMÄßIG SICHERN

Es kann nicht schaden, die Programmdateien, an denen gerade gearbeitet wird, etwa alle **zehn** Minuten zu speichern. Denn Computer pflegen gern gerade dann »abzustürzen«, wenn man seine Arbeit längere Zeit nicht gespeichert hat. In der Regel aber sorgt Android Studio dafür, dass bei Programmschluss alles gespeichert ist.

1 Willkommen in Android Studio

Du willst gleich loslegen? Dein Smartphone brauchst du dazu erst mal nicht, aber einen Windows-PC. Auf dem entwerfen wir dann gemeinsam dein erstes Programmprojekt. Wunder darfst du dabei nicht erwarten, es wird sicher nicht so bunt wie die fast schon unzählbaren Apps, die man aus dem Play Store herunterladen kann. Aber wir sind ja erst ganz am Anfang. Und für das allererste Mal gibt es viel zu tun.

In diesem Kapitel lernst du

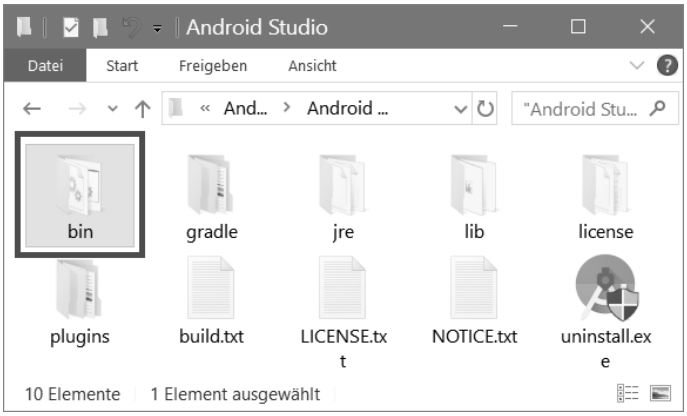
- wie man Android Studio startet
- etwas über den Einsatz des SDK Managers
- wie man eine neue App erzeugt
- etwas über den Einsatz des AVD Managers
- wie man den Android-Emulator startet
- wie man seine App ausführt
- wie man Android Studio beendet

ANDROID STUDIO STARTEN

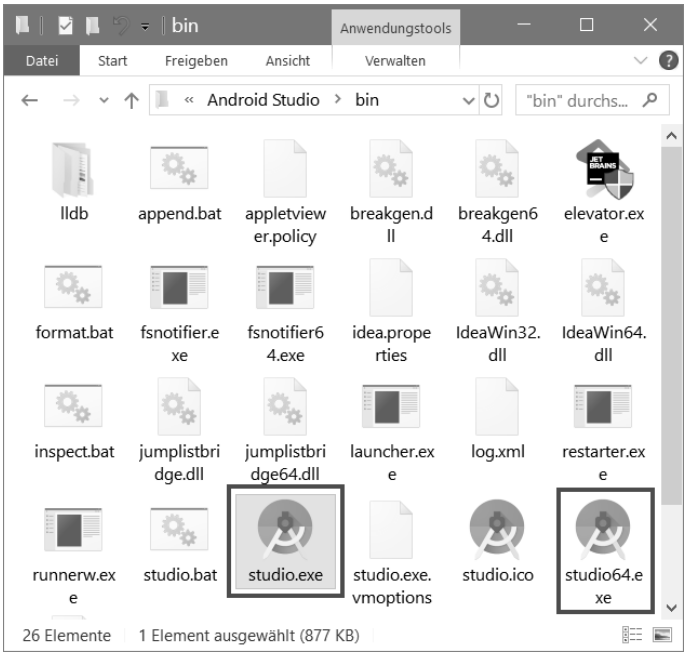
Bevor wir mit dem Programmieren anfangen können, muss das Entwicklungssystem **Android Studio** installiert werden. Genaues erfährst du im **Anhang A**. Hier solltest du dir von jemandem helfen lassen, wenn du dir das Einrichten nicht allein zutraust.

Eine Möglichkeit, Android Studio zu starten, ist diese:

≫ Öffne den Ordner, in den du Android Studio untergebracht hast (z.B. *C:\programme\Android*).



Dort musst du nun weiter in einen Unterordner mit dem Namen *BIN* wechseln:



Hier suchst du unter den vielen Symbolen eines der grünen heraus, und zwar das mit dem Namen *studio.exe*.

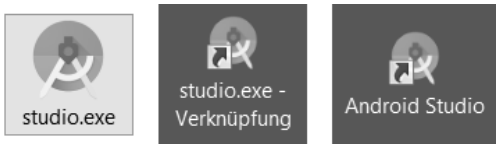
Falls es ein zweites mit dem Namen *studio64.exe* gibt, kannst du auch das ausprobieren. Doch das funktioniert nicht auf jedem PC, hier muss die sogenannte 64-Bit-Version von Windows installiert sein.

≫ Nun kannst du das Programm mit einem Doppelklick auf das Symbol starten.

Ich empfehle dir, eine **Verknüpfung** auf dem Desktop anzulegen:

- Dazu klickst du mit der rechten Maustaste auf das Symbol für Android Studio (STUDIO.EXE). Im Kontextmenü wählst du KOPIEREN.
- Dann klicke auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählst du VERKNÜPFUNG EINFÜGEN.
- Es ist sinnvoll, für das neue Symbol auf dem Desktop den Text *studio.exe – Verknüpfung* durch *Android Studio* zu ersetzen.

Von nun an kannst du auf das neue Symbol doppelklicken und damit Android Studio starten.

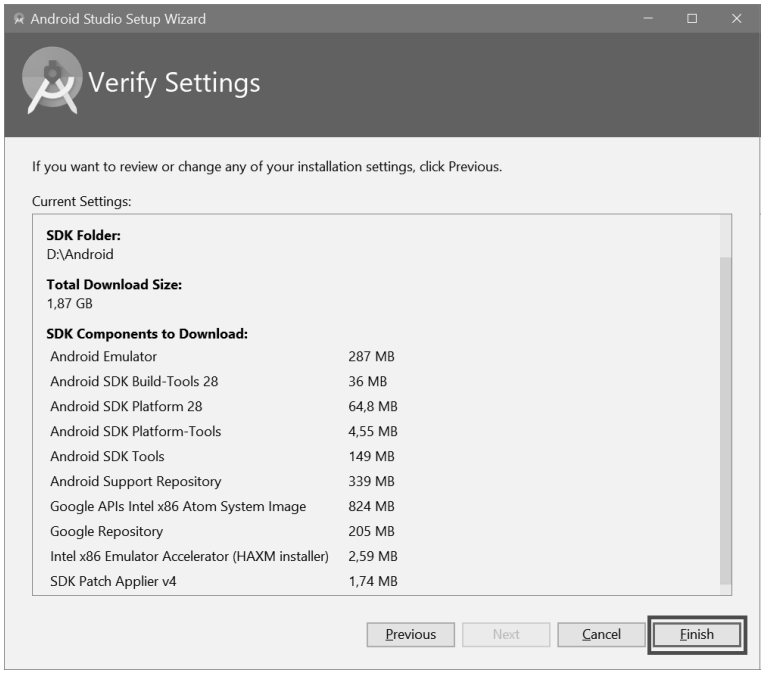


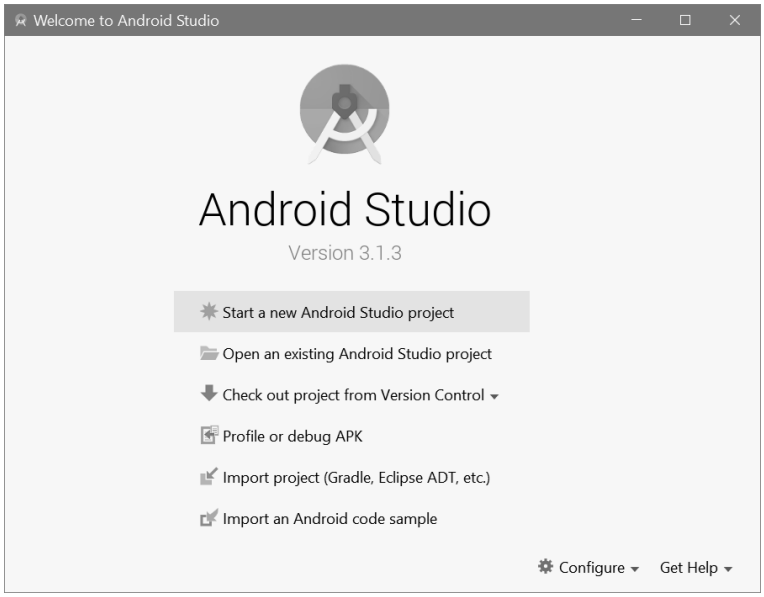
Beim ersten Start kann es sein, dass dich noch ein Überbleibsel von der Installation erwartet.

Ein **Setup Wizard** informiert dich über einige installierte Elemente (die du zum Teil später noch kennenlernen wirst).

≫ Schließe dieses Fenster mit einem Klick auf FINISH.

Je nach Computer kann es eine Weile dauern, bis Android Studio geladen ist. Einige Zeit später landest du in einem Willkommen-Fenster.





Du möchtest gern gleich mit einem neuen Projekt beginnen? Das könntest du mit einem Klick auf START A NEW ANDROID STUDIO PROJECT direkt in die Wege leiten.

Besser aber ist es, wir leisten noch ein bisschen Vorarbeit.

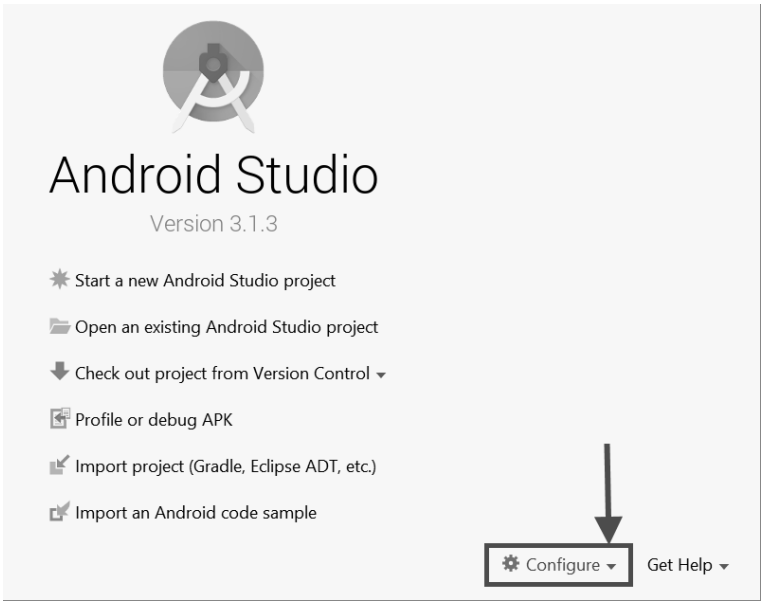
Um ein Android-Projekt zu erstellen und zum Laufen zu bringen, benötigen wir die Hilfe von zwei Managern:

- Der **Software Development Kit (SDK) Manager** stellt die nötigen Elemente des Betriebssystems Android zur Verfügung. Davon gibt es ja inzwischen einige Versionen. Und wenn wir für möglichst viele Smartphones programmieren wollen, brauchen wir nicht nur die neueste, sondern auch einige der älteren Versionen.
- Der **Android Virtual Device (AVD) Manager** bietet die geeigneten Smartphone-Emulationen (das heißt, er sorgt dafür, dass die jeweiligen Smartphones unter Windows künstlich nachgeahmt werden). Natürlich gibt es auch Emulationen für Tablets und andere Geräte.

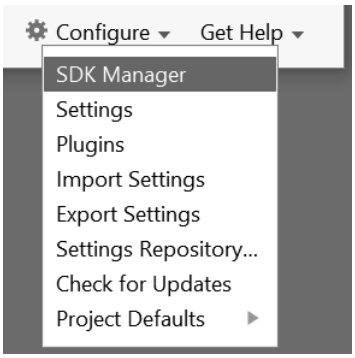
Android Studio hat bei der Installation schon dafür gesorgt, dass die wichtigsten Pakete mitinstalliert wurden. Aber das heißt nicht, dass ein Projekt sofort reibungslos läuft. Denn es gibt noch deutlich mehr Pakete im Angebot, und einige davon werden zusätzlich benötigt. Um selbst zu bestimmen, was letztendlich installiert ist, brauchen wir beide Manager. Dann können deine Apps störungsfrei laufen.

DER SDK MANAGER

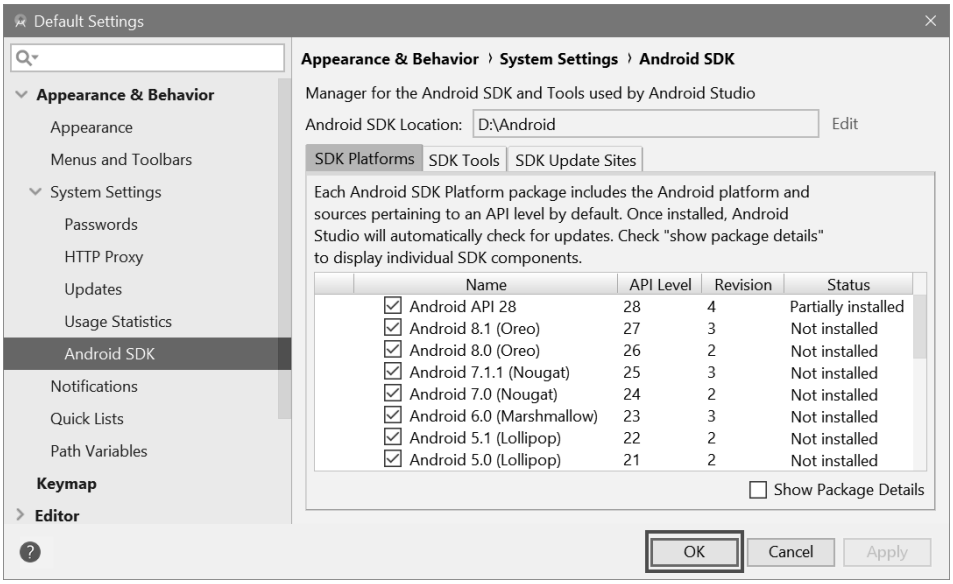
≫ Um das System auf den neuesten Stand zu bringen und das nötige »Drumherum« zu installieren, klicke unten rechts auf das kleine Dreieck hinter CONFIGURE.



≫ Es öffnet sich ein kleines Menü, wo du auf SDK MANAGER klickst.



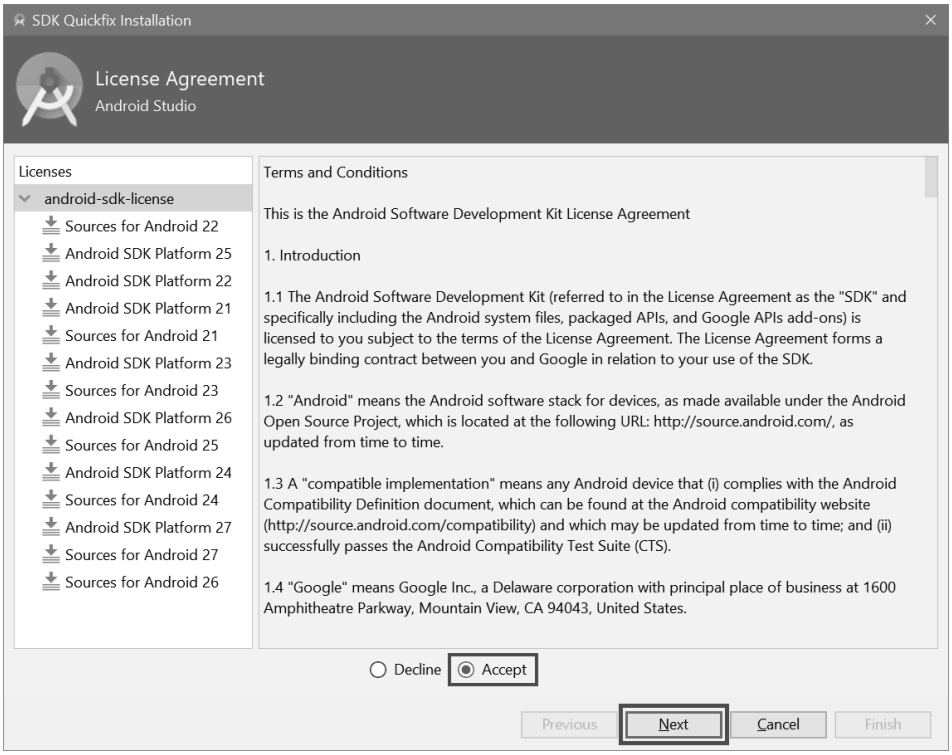
Es öffnet sich ein neues Zusatzfenster. (Möglicherweise wird noch einiges geladen, ehe es dir zur Verfügung steht.)



Unter der Option ANDROID SDK kannst du dir nun die Android-Versionen aussuchen, für die du deine Apps programmieren willst. Bedenke, dass es bis zu einer Stunde (und länger) dauern kann, bis alles heruntergeladen und installiert ist. Ich habe mich entschieden, alle älteren Versionen mindestens ab 5.0 (Level 21) zu bedienen, musste dafür jedoch teilweise sehr lange warten.

≫ Nach deiner Auswahl klickst du auf OK.

Nun musst du noch die Lizenz-Bestimmungen absegnen.

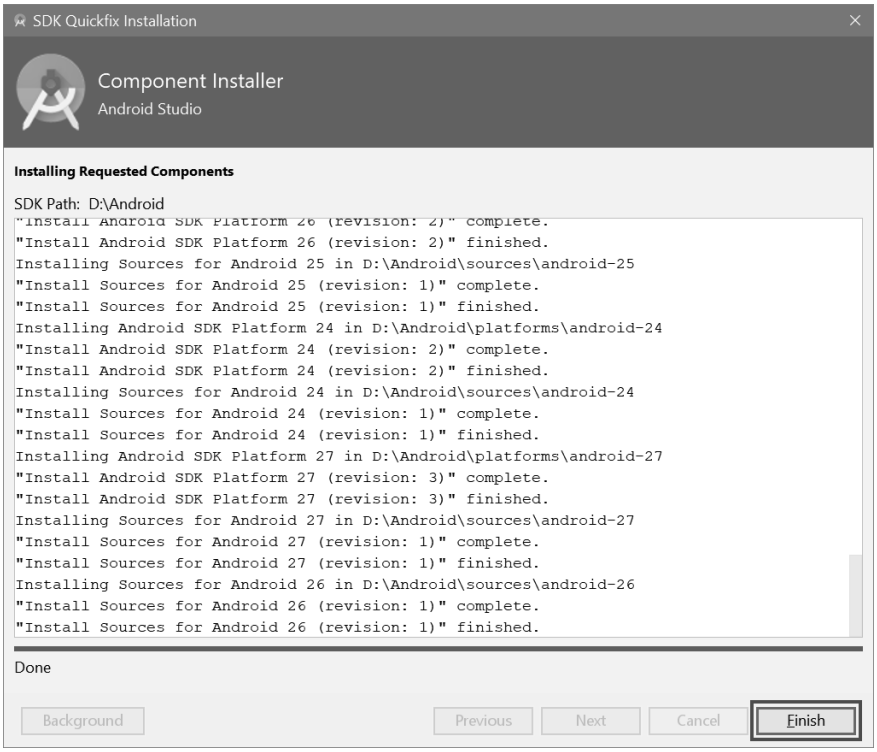


≫ Markiere unten den Eintrag ACCEPT und klicke dann auf NEXT. Damit startest du die Installationen.

Da die Pakete (Packages) für die SDK Tools aus dem Internet geladen werden, ist nun eine **Internet-Verbindung** nötig. Es wird immer das angezeigt, was gerade aktuell ist, deshalb kann das, was du siehst, von der Abbildung im Buch abweichen.

So wie Windows seine Versionsnummer hat, gibt es so etwas natürlich auch bei Android. Weil die meisten sich nicht jedes Jahr ein neues Smartphone oder Tablet kaufen, sind Versionen ab 5.0 noch sehr verbreitet. Du kannst natürlich auch alle Versionen auswählen, womit sich auch sehr viel ältere Smartphones bedienen lassen.

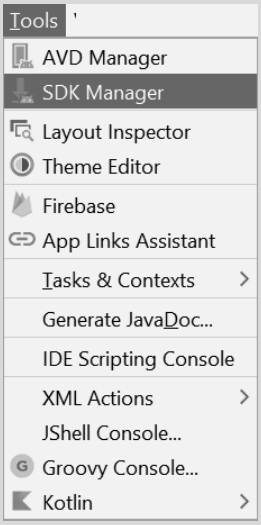
In einem neuen Fenster kannst du die langwierigen Downloads und Installationen mitverfolgen, wenn du willst.



≫ Am Schluss beendest du das Ganze mit einem Klick auf FINISH.

Und du landest wieder im Willkommen-Fenster von Android Studio.

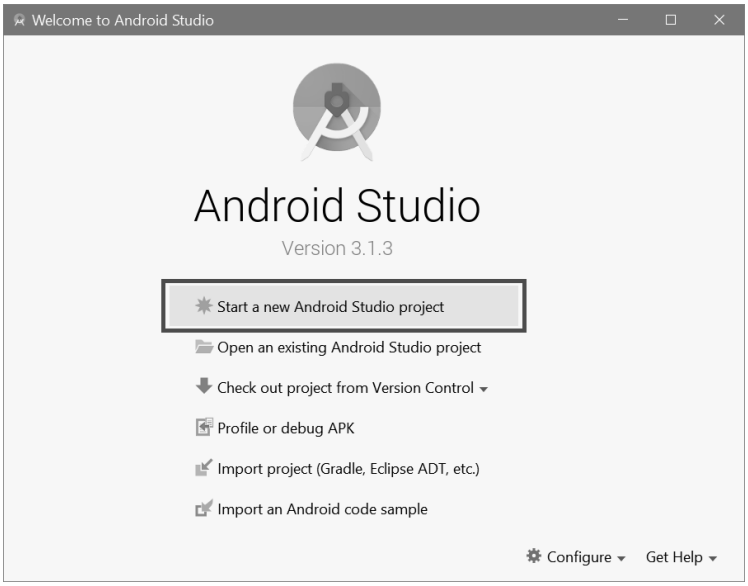
Der SDK Manager lässt sich auch über das TOOLS-Menü starten:



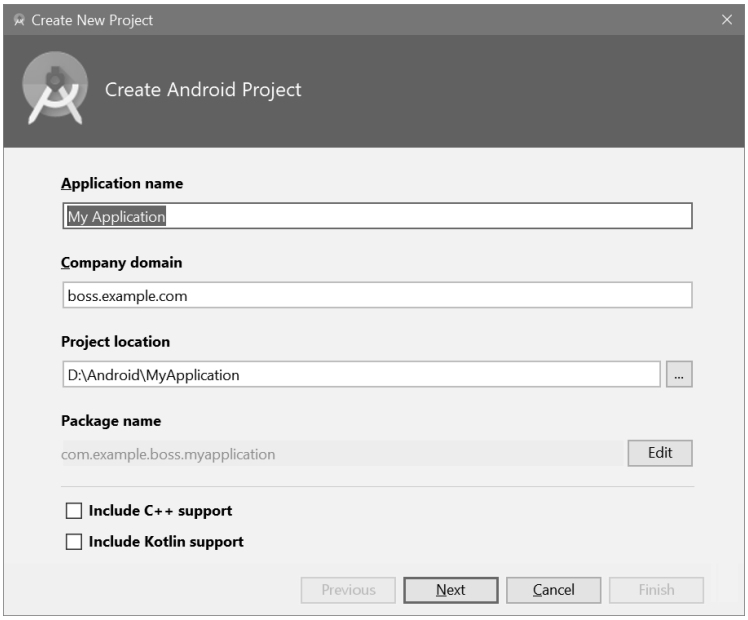
EIN NEUES PROJEKT ERZEUGEN

Und nun kannst du endlich in dein erstes Projekt einsteigen.

≫ Dazu klickst du auf die Schaltfläche START A NEW ANDROID STUDIO PROJECT.

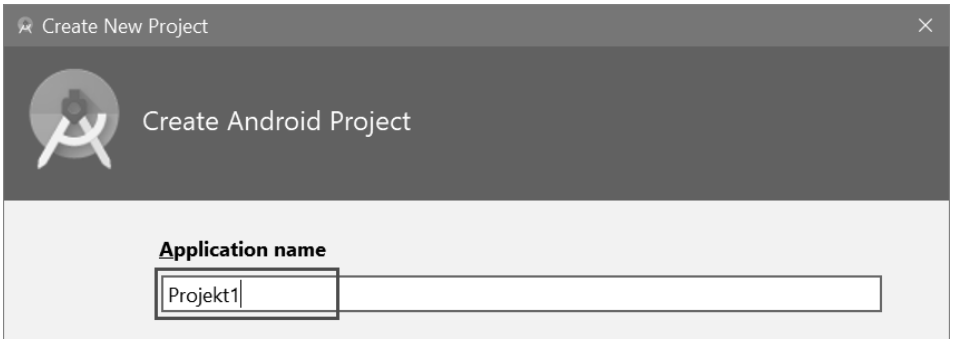


Ein neues Dialogfeld mit dem Titel CREATE NEW PROJECT erscheint und fordert dich heraus. Dann musst du etwas eingeben und gleich eine weitere Entscheidung treffen.



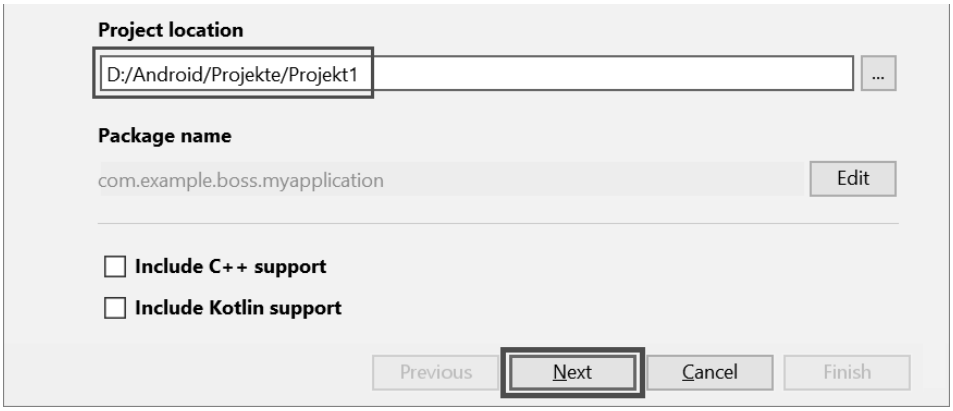
Zuerst braucht dein Projekt einen Namen. Du kannst natürlich *My Application* stehen lassen, doch ich empfehle dir, stets einen eigenen neuen Namen zu vergeben. Nennen wir das erste Projekt einfach *Projekt1*, auch weil wir noch nicht wissen, was genau die App, die nachher herauskommt, macht.

≫ Tippe hinter APPLICATION NAME einen neuen Namen ein.



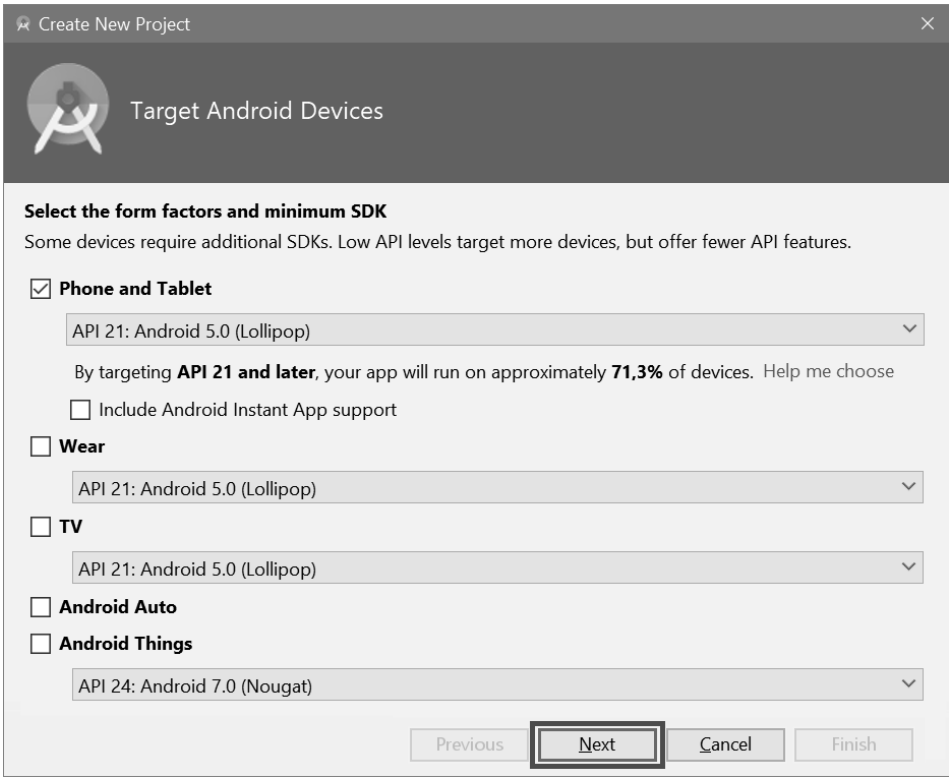
Weiter unten wird das Verzeichnis angezeigt, in dem dein Projekt untergebracht werden soll. Im Allgemeinen bietet Android Studio einen Platz im *Benutzer*-Ordner auf Laufwerk C: an. Ich schlage vor, die Projektdateien in einem anderen Laufwerk und Ordner unterzubringen.

Dazu könntest du einen weiteren Ordner in *D:\Android* mit dem Namen *Projekte* erstellen. Wenn dir das zu umständlich ist, lasse einfach den angebotenen Speicherort hinter PROJECT LOCATION stehen.



≫ Wenn Name und Speicherort feststehen, kannst du auf NEXT klicken.

Im folgenden Fenster genügt es, wenn ein Haken vor PHONE AND TABLET steht. Wir wollen eine App (= Applikation) für Smartphones entwickeln, aber wenn das Ding auch auf einem Tablet läuft, umso besser (an anderen Geräten sind wir derzeit nicht interessiert).



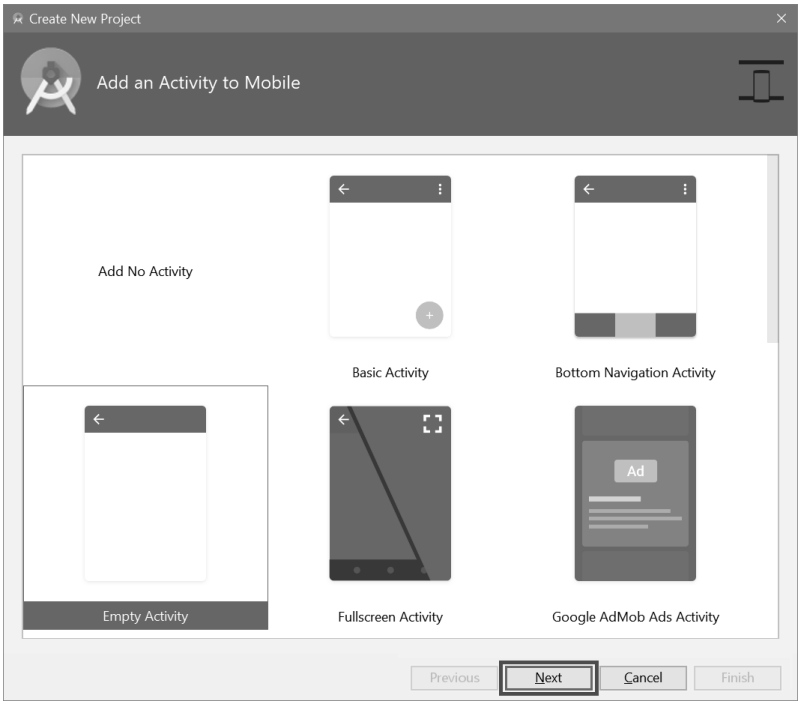
Da nicht alle Smartphones mit den neuesten Versionen von Android versorgt werden, ist es sinnvoll, ein Projekt für eine (etwas) frühere Android-Version zu entwickeln. So läuft deine App auf möglichst vielen (auch älteren) Smartphones.

Auf jeden Fall laufen sämtliche Apps, die für ältere Android-Versionen entwickelt wurden, auch auf allen neuen Smartphones.

Wichtig ist: Für die eingestellte Version muss das **exakte** SDK installiert sein. Notfalls musst du also noch mal den SDK Manager bemühen, um Fehlendes nachträglich zu installieren.

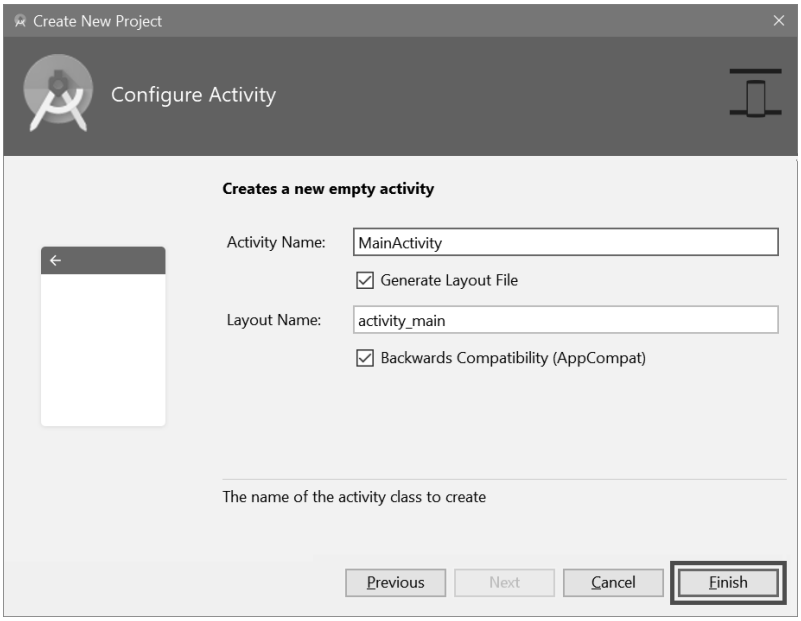
≫ Sorge dafür, dass die Android-Version eingestellt ist, mit der du arbeiten willst. Dann klicke auf NEXT.

Und nun geht es um die **Activity**. So nennt man ein Benutzerinterface. Das ist die Verbindungsstelle zwischen dem Benutzer (z.B. dir) und einem Smartphone. Vereinfacht kann man hier sagen: ein Fenster auf dem Display. Also quasi der Rahmen für die Aktivitäten einer App. Du solltest dich hier für eine »Empty Activity« entscheiden, also sozusagen ein »Fenster mit nix drin«.



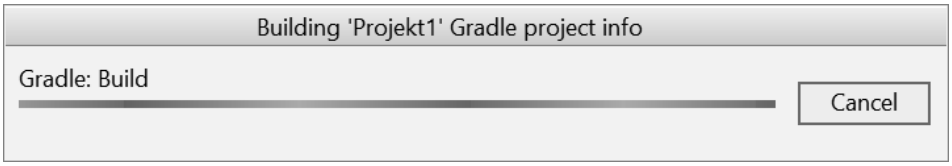
≫ Sorge dafür, dass EMPTY ACTIVITY markiert ist, und klicke dann auf NEXT.

Auch im folgenden Fenster kannst du alles für dein aktuelles Projekt so stehen lassen.

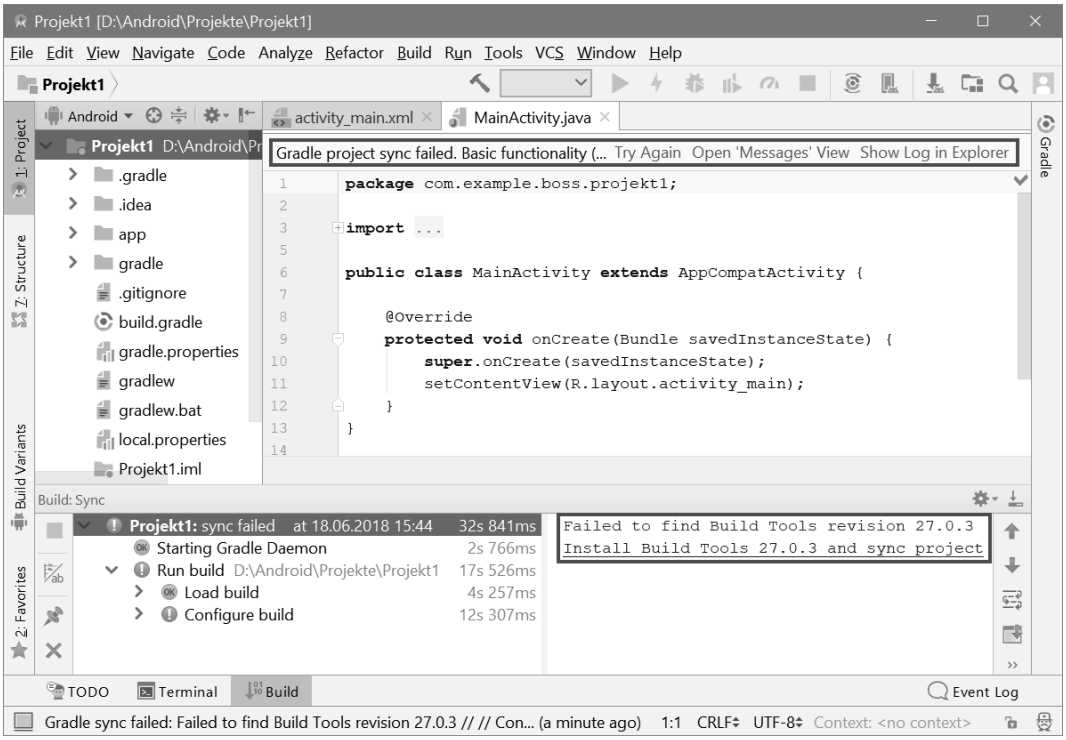


≫ Diesmal klickst du auf FINISH und schließt damit auch das Fenster.

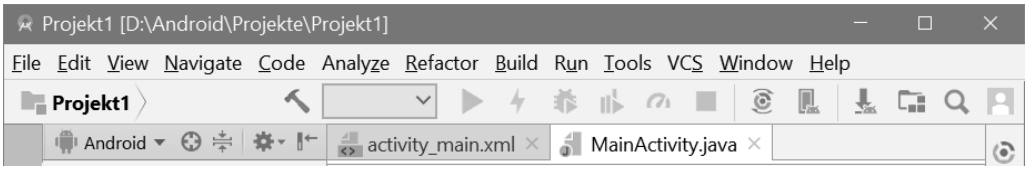
Nun beginnt Android Studio, deine Einstellungen zu verarbeiten. Das kann eine Weile dauern.



Was dich dann erwartet, könnte etwa so oder ähnlich aussehen:



Für den ersten Augenblick mag das ein bisschen sehr verwirrend sein. Doch fangen wir ganz oben an. Dort sitzt die Menüleiste. Darunter befinden sich mehrere Symbole, die man mit der Maus anklicken kann.



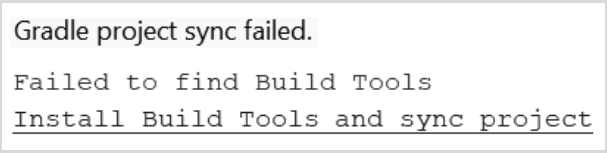
Von den vielen Menüs wirst du wahrscheinlich diese am meisten benutzen:

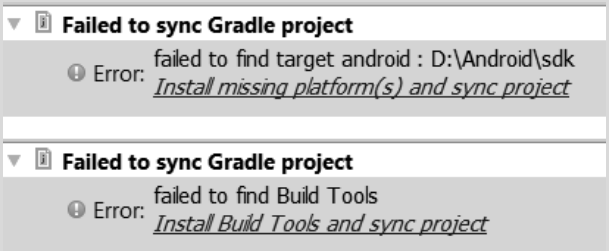
- Über das FILE-Menü kannst du Dateien speichern, laden (öffnen), ausdrucken, neu erstellen oder Android Studio beenden.
- Das Menü EDIT hilft dir bei der Bearbeitung deines Programmtextes, aber auch bei anderen Programmelementen. Außerdem kannst du dort bestimmte Arbeitsschritte rückgängig machen oder wiederherstellen.
- Über das RUN-Menü sorgst du dafür, dass dein Projekt ausgeführt wird.
- Und das HELP-Menü bietet dir vielfältige Hilfsinformationen (auf Englisch) an.

Unter dem Menübereich erwarten dich ein paar Fensterabschnitte:

- Links siehst du eine Spalte, die dem Explorer in einem Fenster unter Windows entspricht, hier werden die Bestandteile deines Projekts aufgelistet. (Möglicherweise sieht es bei dir anders aus, z.B. wenn oben nicht ANDROID eingestellt ist und die Listen eingeklappt sind.)
- Rechts steht der sogenannte Quelltext, mit dem du jetzt sicher noch nichts anfangen kannst.
- Ganz unten sind weitere Informationen. Dort werden unter anderem auch Fehler gemeldet.

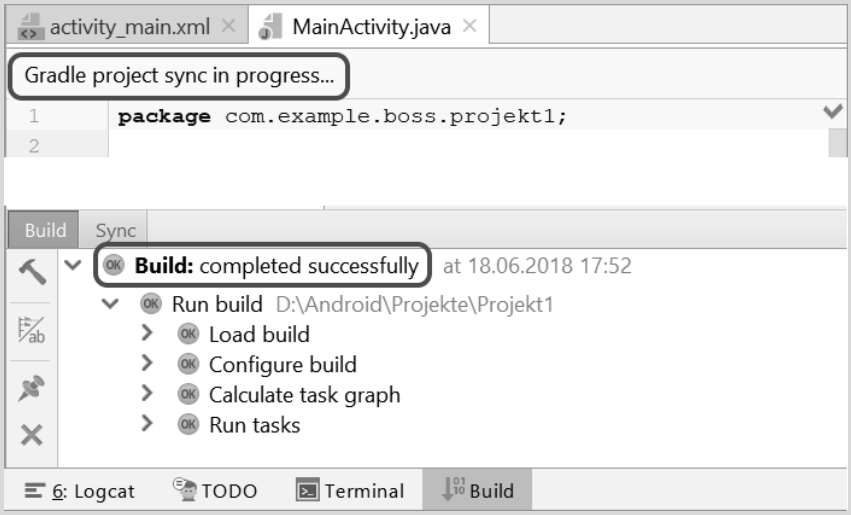
Möglicherweise gibt es bei dir ebenso wie bei mir gleich zu Anfang Fehlermeldungen. Von GRADLE und BUILD TOOLS kann da die Rede sein.





Im Falle einer solchen Meldung klickst du unten auf INSTALL BUILD TOOLS, danach werden einige Updates ausgeführt.

Anschließend kommt die Meldung, dass Android Studio erneut versucht, ein funktionierendes Programm zu erstellen.

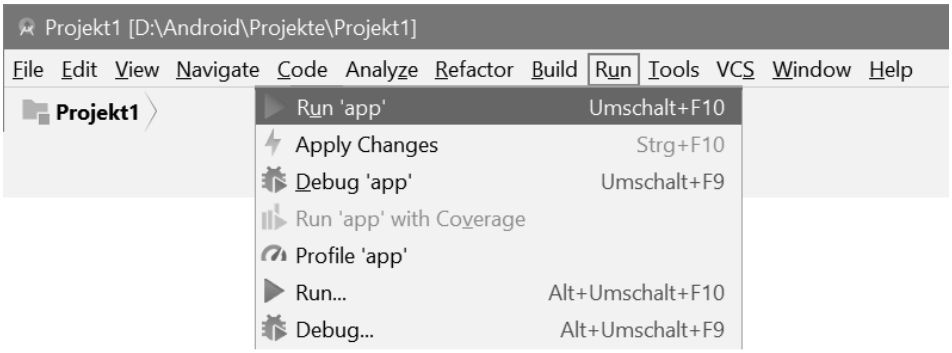


Und anschließend wird ein erfolgreicher Abschluss gemeldet.

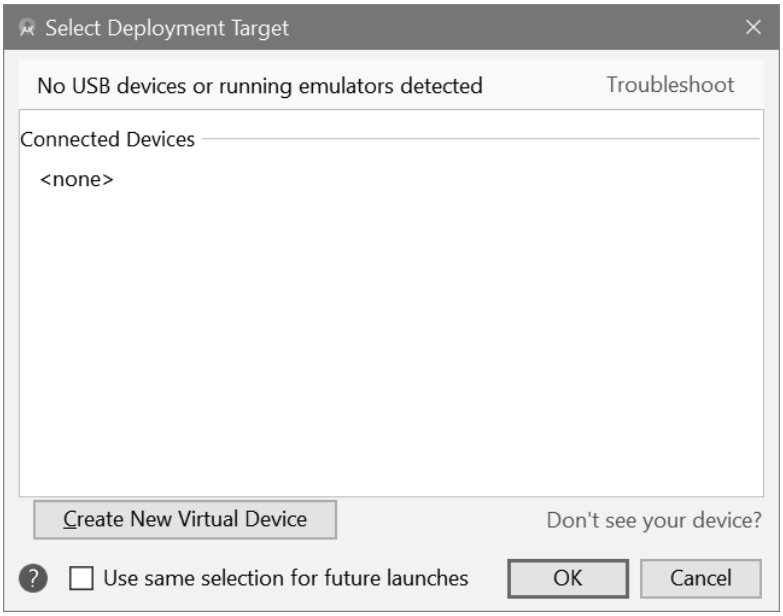
DER AVD MANAGER

Haben wir etwa schon ein Programm bzw. eine App?

≫ Probiere einfach aus, was passiert, wenn du im RUN-Menü auf den ersten Eintrag RUN 'APP' klickst (oder auf den grünen Pfeil in der Symbolleiste).



Nach einer Weile bekommst du dieses Dialogfeld zu sehen:



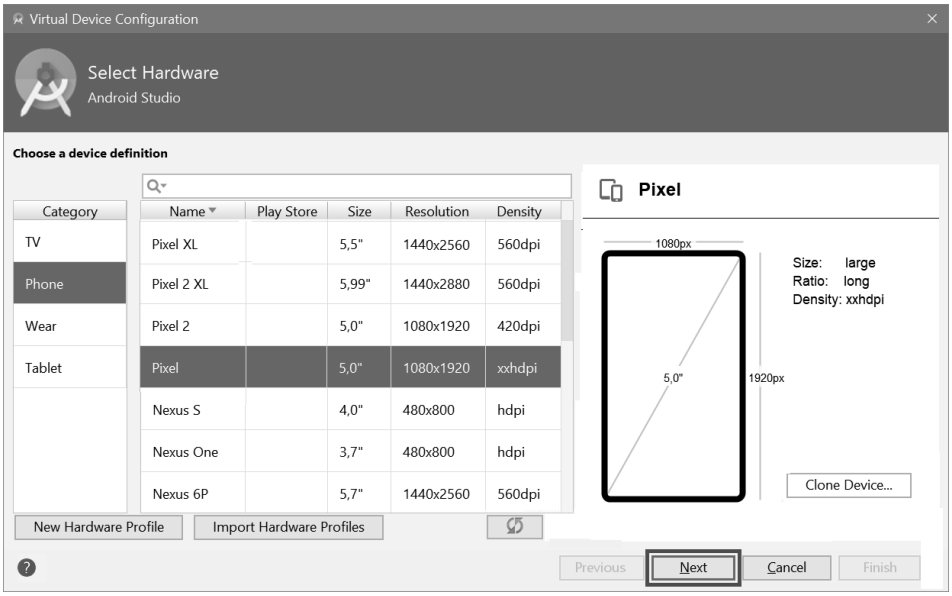
Mit »No USB devices« ist gemeint, dass offenbar kein Smartphone oder Tablet per USB am Computer angeschlossen ist. Und das »none« unter CONNECTED DEVICES weist darauf hin, dass auch kein Smartphone emuliert wird. Ansonsten ist darunter gähnende Leere.

Später werden wir direkt ein Smartphone an den PC anschließen, um unsere Apps auch dort zu testen. Jetzt aber brauchen wir ein virtuelles (also un-echtes) Gerät, das so tut, als sei es ein Smartphone (genau das tut ein **Emulator**).

Wir müssen also nur ein solches Gerät, genannt **Android Virtual Device** (kurz: AVD) finden und das aktivieren.

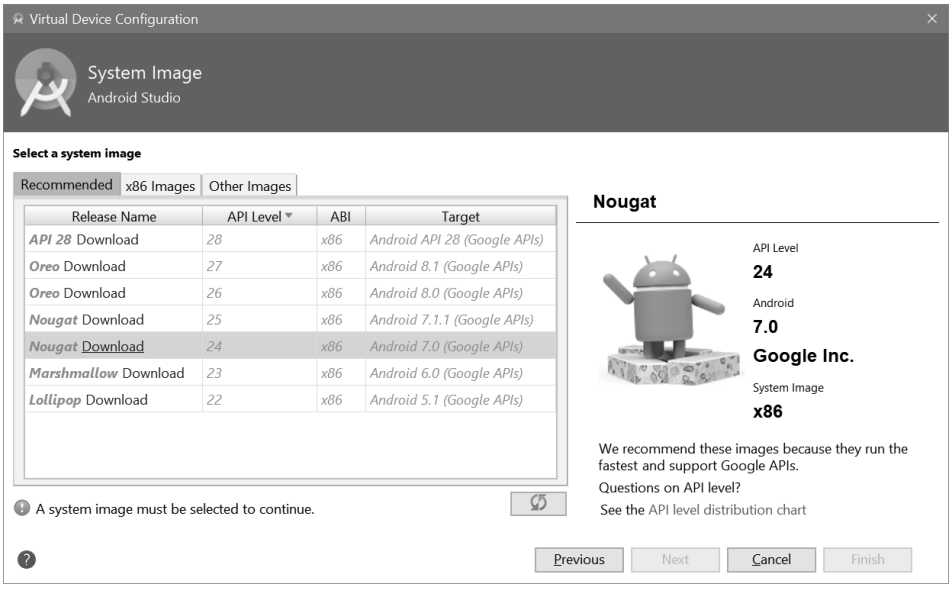
≫ Klicke auf CREATE NEW VIRTUAL DEVICE.

Und schon erscheint ein neues Dialogfeld mit dem Titel SELECT HARDWARE. Hier findest du unter PHONE eine ganze Reihe von Smartphones, die offenbar unter Android Studio emuliert werden können.



≫ Markiere das Gerät, das du haben willst, und klicke dann auf NEXT.

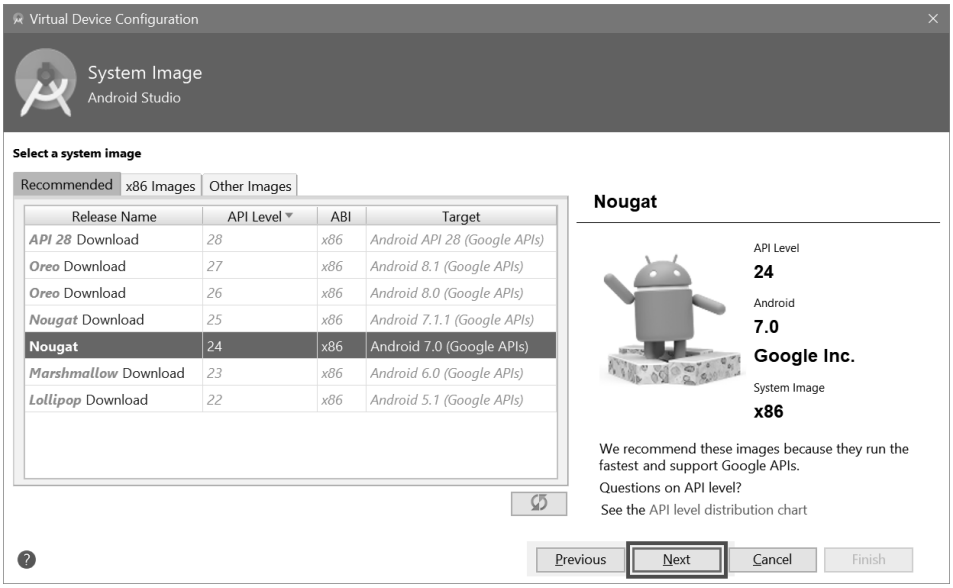
Und schon landest du im nächsten Dialogfeld mit dem Titel SYSTEM IMAGE.



Hier wählst du eines der Android-Systeme aus der Liste. Was darin steht, hängt auch davon ab, wie viel du mit dem SDK Manager installiert hast.

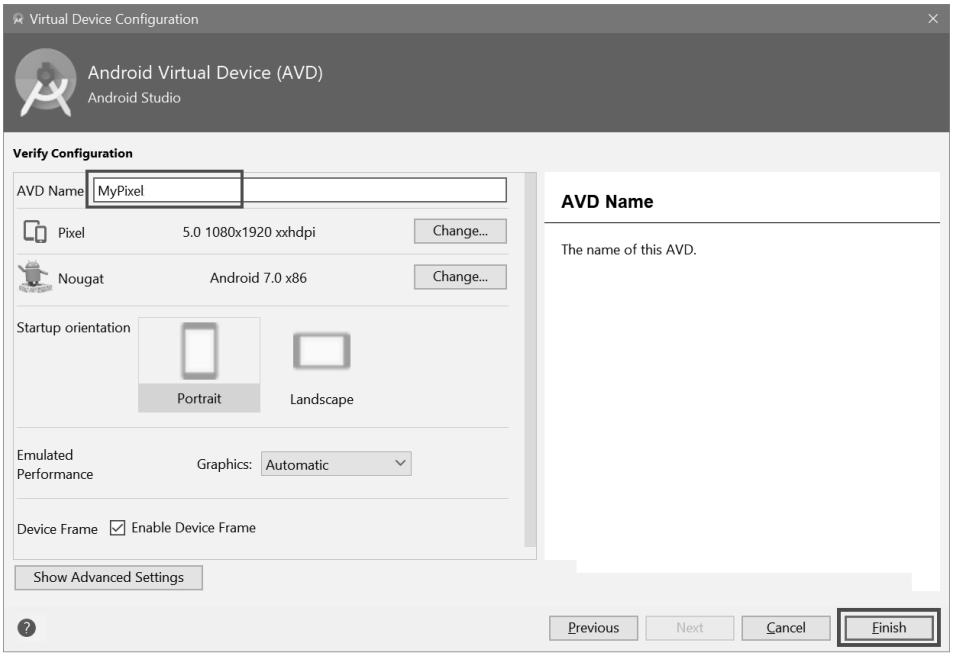
≫ Gegebenenfalls musst du noch Dateien für die betreffenden Versionen nachladen. Klicke also hinter dem Namen der Version auf DOWNLOAD.

Anschließend sollte das System deiner Wahl markiert sein.



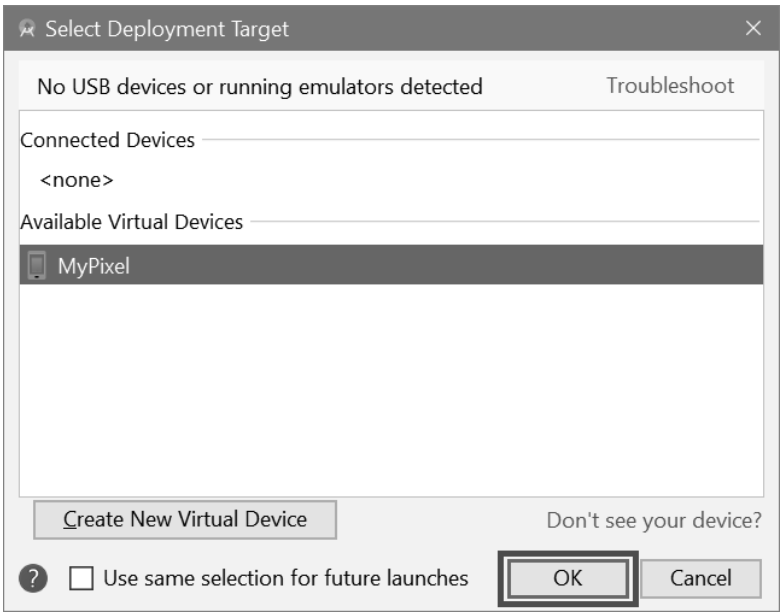
≫ Klicke nun auf NEXT.

Und damit bist du fast fertig.



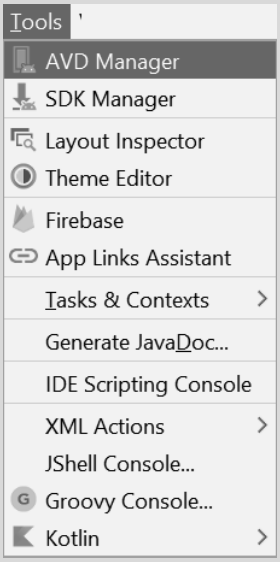
≫ Im letzten Dialogfeld kannst du für den Emulator (AVD) noch einen eigenen Namen eingeben, wenn du willst. Klicke abschließend auf FINISH.

Und damit landest du wieder am Anfang. Nun hast du eine Liste mit einem neuen Emulator. (Diese Liste kannst du natürlich später beliebig erweitern.)



≫ Um das Ganze abzuschließen, klicke auf OK.

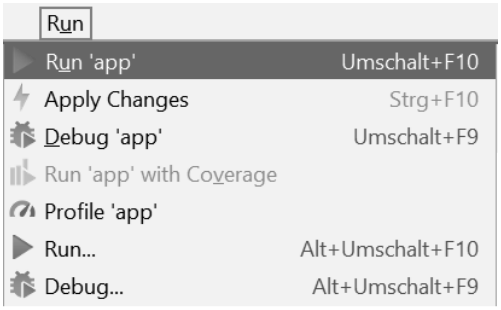
Der AVD Manager lässt sich auch über das TOOLS-Menü starten:



DIE EMULATION STARTEN

Und nun wollen wir schauen, ob wir unser erstes Programm zum Laufen kriegen.

≫ Dazu klickst du wieder im RUN-Menü auf RUN 'APP' (oder auf den grünen Pfeil in der Symbolleiste).



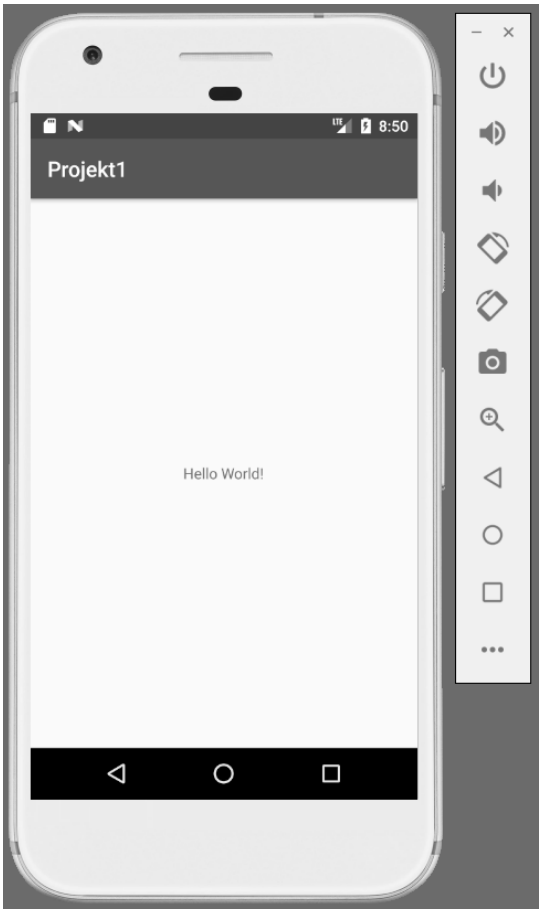
Das letzte Dialogfeld öffnet sich erneut.

≫ Klicke hier einfach (noch mal) auf OK. Und warte.

Kurz darauf tut sich ein weiteres, diesmal eindrucksvoll großes bzw. hohes Fenster auf, das wie ein Smartphone aussieht.

Und nun kann es viele Minuten dauern, bis das Android-System sich »aufgerappelt« hat. Das ist so ähnlich, als wenn du dein Smartphone komplett ausgeschaltet hast und neu startest. Also ist Geduld gefragt. (Aber die hast du ja bis jetzt ohnehin bewiesen.)

Warten wir jetzt erst einmal gelassen, bis das Emulations-Fenster so oder ähnlich aussieht:



Obwohl wir in Sachen Programmierung noch keinen Finger gerührt haben, erscheint hier schon ein Gruß an alle Welt (auf Englisch).

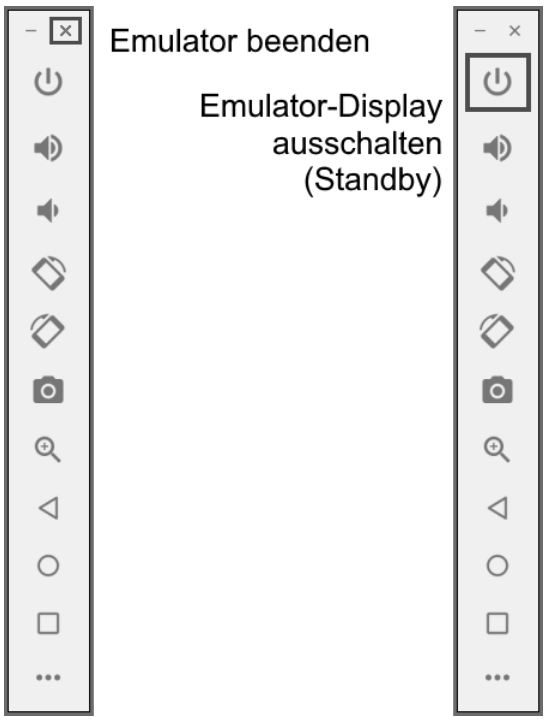
Früher oder später willst du natürlich deine Apps auf dem Smartphone testen. Wie das funktioniert, steht in **Anhang B**.

ANDROID STUDIO BEENDEN

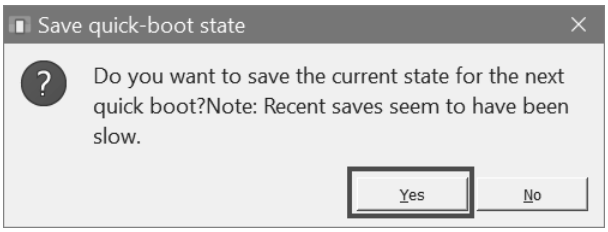
Das erste Projekt ist erstellt und gelaufen. Was will man bei diesem Aufwand mehr? Natürlich willst du mehr, doch jetzt solltest du für eine Pause erst mal Android Studio verlassen.

Das geht in zwei Schritten. Neben dem Smartphone siehst du eine Steuer-Leiste. Mit Klick auf das Symbol für POWER schaltest du das emulierte Smartphone aus.

≫ Klicke auf das kleine X ganz oben, um den Emulator zu schließen.



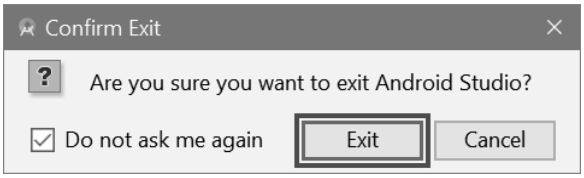
≫ Wenn dir dieses Meldefenster begegnet, solltest du auf YES klicken.



Damit musst du beim nächsten Mal nicht mehr so lange warten, bis der Emulator gestartet ist.

≫ Zurück im Hauptbereich von Android Studio klickst du auf FILE und dann auf EXIT. (Oder du klickst auch hier ganz oben rechts auf das kleine **X**.)

In einem weiteren Meldefenster wird noch einmal nachgefragt:



≫ Klicke auf EXIT. (Wenn du dieses Fenster nicht mehr sehen willst, musst du vorher DO NOT ASK ME AGAIN markieren.)

ZUSAMMENFASSUNG

Eine eigene App. Dieses Ziel haben wir hier nicht erreicht. Noch nicht. Aber du hast schon mal einen ersten Eindruck über die Arbeit mit Android Studio gewonnen. Aller Anfang ist schwer? Hier passt es, doch der erste Hürdenlauf ist geschafft.

Du weißt jetzt etwas über

- den SDK Manager, der für die Android-Entwicklungspakete zuständig ist.
- den AVD Manager, der sich um die Smartphone-Emulatoren kümmert.

Du weißt, dass du eine `Activity`, ein Benutzerinterface als Aktionsrahmen für dein Projekt brauchst. Und du kennst schon ein paar Operationen im Umgang mit Android Studio:

Android Studio starten	Doppelklicke auf das Symbol für Android Studio. Oder klicke auf START/AUSFÜHREN und tippe den kompletten Pfad für *STUDIO.exe* ein.
SDK Manager starten	Klicke auf TOOLS/SDK MANAGER.
AVD Manager starten	Klicke auf TOOLS/AVD MANAGER.
App-Projekt starten	Klicke auf RUN/RUN 'APP'.
Hilfesystem aufrufen	Klicke auf HELP.
Android Studio beenden	Klicke auf FILE/EXIT.

ZWEI FRAGEN ...

1. Was bedeuten SDK und AVD?
2. Was ist eine `Activity`?

... ABER NOCH KEINE AUFGABEN

STICHWORTVERZEICHNIS

D

E

F

G

H

I

P

Q

R

S

T

U

V

W

X

Z