

Apps-Entwicklung mit Eclipse

Version 1.1, 30. April 2013

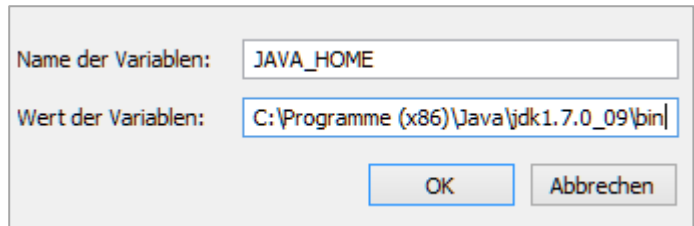
Vorbereitungen:

1. JDK installieren

JDK SE neuste Version (64 oder 32 Bit) herunterladen und installieren
(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

Environment variable: JAVA_HOME
auf das Java-Installations-
verzeichnis\bin definieren

Pfad auf <home>\bin anfügen



2. Android-SDK und Eclipse (ADT Bundle) installieren.

Download von <http://developer.android.com/sdk/index.html>

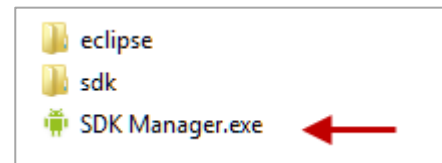
ADT Bundle (32- bzw. 64-bit wählen)

Inhalt mit Verzeichnissen eclipse und sdk sowie Datei SDK Manager in ein Verzeichnis auspacken, z.B. c:\Programme\adt (im Folgenden <adthome> genannt) und Pfad auf <adthome>\sdk\tools und <adthome>\sdk \platform-tools legen

3. SDK Manager starten (im <adthome>).

Zusätzliches Package installieren:

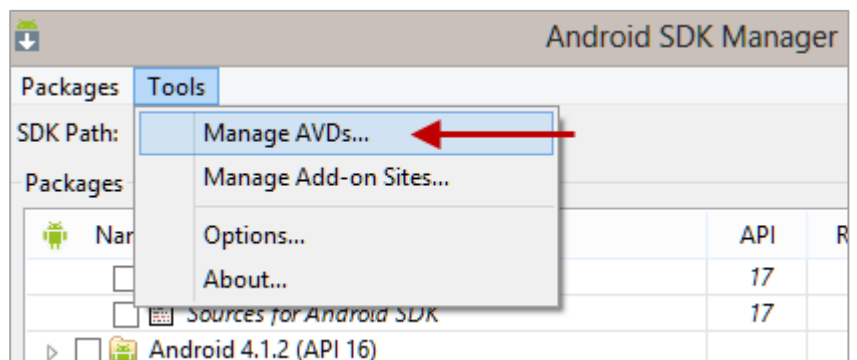
Android 2.2 (API 8) wählen und Install klicken.



<input type="checkbox"/>	Android 3.1 (API 12)		
<input type="checkbox"/>	Android 3.0 (API 11)		
<input type="checkbox"/>	Android 2.3.3 (API 10)		
<input checked="" type="checkbox"/>	Android 2.2 (API 8)		
<input checked="" type="checkbox"/>	SDK Platform		8
<input checked="" type="checkbox"/>	Samples for SDK		8
<input checked="" type="checkbox"/>	Google APIs		8
<input type="checkbox"/>	Android 2.1 (API 7)		

4. Smartphone Emulator

Im Menu Tools | Manage
AVDs wählen und New
klicken:

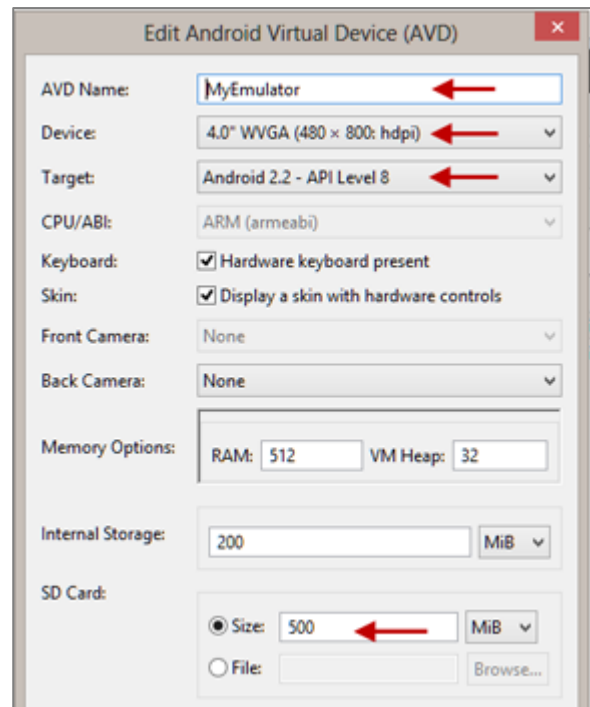


AVD Name: MyEmulator

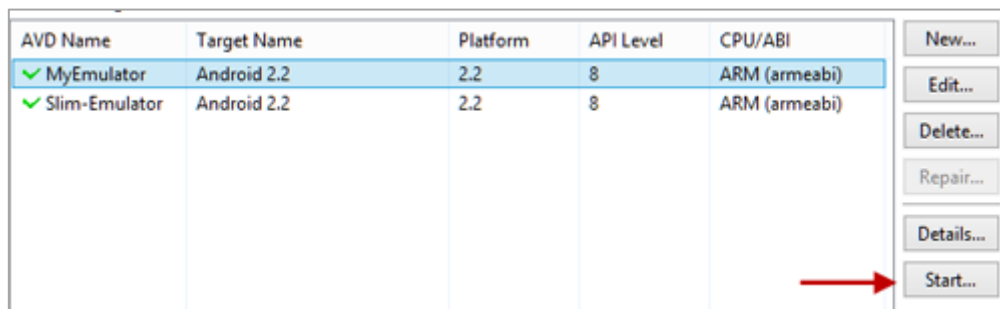
Device: 4.0" WVGA Target: Android 2.2

SD Card: Size: 500 MiB

OK drücken (warten)



Dann im SDK-Manager MyEmulator klicken und Button Start und dann Launch drücken. Der Emulator wird gestartet (sehr lange warten).



(Falls der Emulator ohne SDK-Manager gestartet werden soll, einen Batch geschrieben werden mit dem Inhalt: emulator @MyEmulator)

5. Eclipse starten

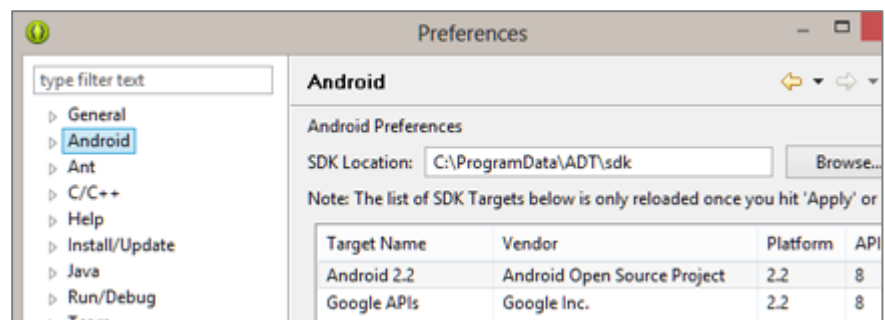
Einen Link auf die Ikone eclipse im Verzeichnis <adthome>/eclipse erstellen



6. ADT Plugin konfigurieren:

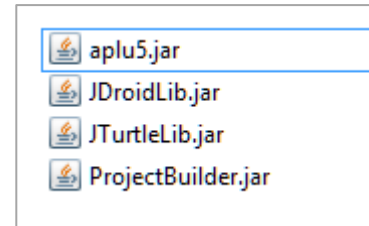
Menu | Window |
Preferences | Android:

SDK Location:
<adthome>/sdk



7. JDroidLib (Klassenbibliotheken und ProjectBuilder)

JDroidLib distribution von <http://www.aplu.ch/android> downloaden und auspacken. **JDroidLib.jar** in ein Verzeichnis mit anderen Jars kopieren, z.B. in c:\jars. **ProjectBuilder.jar** ebenfalls in irgendein Verzeichnis kopieren und einen Startlink darauf erstellen. Falls auch Turtle-Apps erstellt werden sollen, ebenfalls die Distribution von <http://www.aplu.ch/jturtlelib> downloaden und **JTurtleLib.jar** ins gleiche Verzeichnis kopieren



8. Neues Android Project erstellen

Menu *File | New | Android Application Project*

Application name: **MyApp1**
Project name: **MyApp1**
Package name: **unwichtig** (wird durch den ProjectBuilder bestimmt)

Die folgenden Felder sind **wichtig**:

Minimum Required SDK: **API 8: Android 2.2**
Target SDK: **API 8: Android 2.2**
Compile With: **API 8: Android 2.2**
Theme: **None**
Next klicken

Create custom launcher icon: **abwählen**

Create activity: **abwählen**

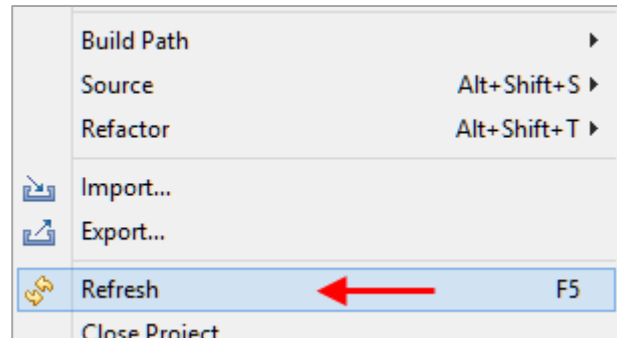
Finish klicken

9. ProjectBuilder starten. Ausfüllen (gemäss Beispiel, anpassen!)

Build drücken

Der Package Name wird jetzt festgelegt. Sprites Folder und Media Folder können leer bleiben, andere Felder müssen ausgefüllt sein. Achtung, im Feld *Library Folder* muss der qualifizierte Pfad zu den strichpunkt-getrennten Libraries angegeben werden. Für JDroid-Apps muss die Bibliothek **JDroidLib.jar** angegeben werden, für JTurtle-Apps die Bibliothek **JTurtleLib.jar** wählen. Zudem muss *Use Turtle* selektiert sein.

Danach unbedingt im Kontextmenü der rechten Maustaste **Refresh** drücken oder **F5**. Im Package Explorer ist die Source Datei MyApp1.java nun sichtbar.



Was macht der ProjektBuilder

Der ProjectBuilder ersetzt

AndroidManifest.xml <projectroot>\res\layout\main.xml

<projectroot>\res\values\strings.xml

<projectroot>\src<packagepath>\<ApplicationName>.

Library file in <projectroot>\libs kopiert (falls libs nicht existiert, wird der Folder erzeugt)

Alle Files aus <SpritesFolder> in <projectroot>\res\drawable kopiert (folder drawable erzeugt, falls nicht bereits vorhanden). Files jdroid__gglogo.png und turtle_default.png in <projectroot>\res\drawable kopiert.

In build.xml wird der App-name eingetragen

10. Installation/Starten der App auf Emulator

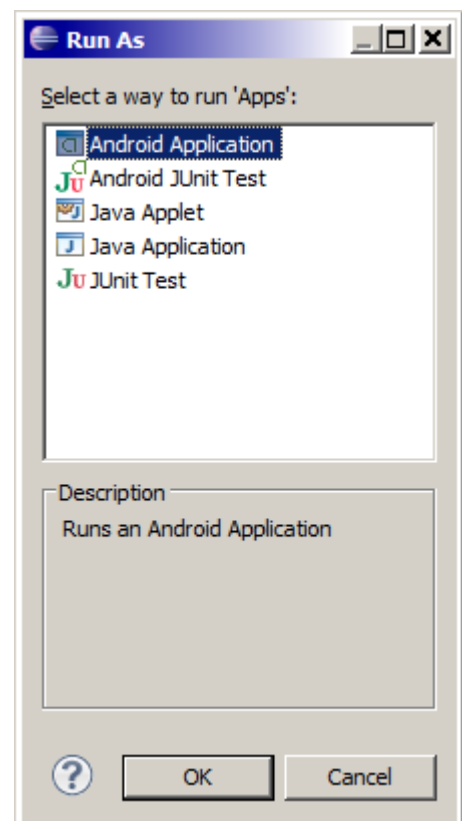
Im Package Explorer muss das Projekt oder das Package selektiert sein.

Den grünen **Run** Button in der Menuleiste klicken. und Android Application auswählen.

OK

Das Projekt wird kompiliert/gepackt. Ist kein Smartphone angeschlossen, wird die App nachfolgend auf den Emulator kopiert und dort gestartet.

Da es lange dauert, bis der Emulator gestartet wird, lässt man ihn am besten offen, solange man entwickelt. Um die Erstellung des APK zu verfolgen, sollte man mit Menü Window | Show View | Console die Ausgabekonsolle sichtbar machen. App erneut kompilieren/packen und installieren mit Ctrl-F11.



9. Installation/Starten der App auf Smartphones über USB:

Modellspezifische Synchronisationssoftware installieren, z.B. bei HTC: *HTC Synch*, bei Samsung *Kies*.

Einstellungen auf dem Smartphone:

- Anwendungen: Unbekannte Quellen OK wählen
- Entwicklung: USB Debugging OK
- Wach bleiben
- Smartphone NICHT als externe Disk ansprechen (auf HTC: Einstellungen | Mit PC verbinden | Standardtypen NUR LADEN !)

Smartphone über USB-Kabel anschliessen. Den grünen **Run** Button in der Menuleiste klicken. und Android Application auswählen.

Das Project wird kompiliert/gepackt und nachfolgend wird die App auf das Smartphone kopiert und dort gestartet.

In einer Command Shell kann mit **adb devices** überprüft werden, ob ein Device erkannt wird. Das Smartphone muss mit seiner ID erscheinen, sonst ist der Smartphone-USB-Treiber nicht installiert/nicht funktionsfähig.

10. Manuelle Installation einer App:

In einer Command Shell im Verzeichnis der Applikation

adb install -r Apps-debug.apk (Apps-debug.apk ist ein Beispiel)

ausführen. Eventuelle Fehlermeldung beachten.

Ergänzungen:

- Falls man die Applikationsicone ändern will, kann im Verzeichnis res/drawable die Datei `jdroid_gglogo.png` ersetzt werden
- Falls die App auf dem Smartphone nicht startet, kontrollieren, ob es nicht bereits eine Applikation mit demselben Package oder sogar mit demselben Namen gibt. Falls ja, mit Smartphone-Tool unter (Einstellungen/Anwendungen) **deinstallieren** (da es einen Konflikt wegen der Signatur geben kann, falls man sich auf einem anderen Rechner befindet.)
- Falls man im angegebenen Sprites-Folder ein Sprite ändert/hinzufügt, muss man den ProjectBuilder wieder ausführen, damit die Änderung übernommen wird.