

JDroidLib mit Eclipse (Mac/Linux/Windows)

Version 1.3, 25. März 2013

(Unter Windows besser die ADT-Bundle Version installieren, siehe entsprechende Anleitung)

Vorbereitungen:

1. JDK SE neuste Version installieren, siehe
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
Environment variable: JAVA_HOME auf home Verzeichnis definieren
Pfad auf <home>\bin setzen
2. Android-SDK installieren <http://developer.android.com/sdk/index.html>
Wählen: DOWNLOAD FOR OTHER PLATFORMS, dann SDK Tools Only (kein Bundle) zip/tgz auspacken und in ein lokales Verzeichnis kopieren, z.B. unter Windows: c:\programs\android-sdk (im Folgenden <sdkhome> genannt
Pfad auf <sdkhome>\tools und auf <sdkhome>\platform-tools
3. SDK Manager starten (im <sdkhome>) Choose package to install: Minimale Version:
Alles wegwählen. Dann wieder dazuwählen:
 - Android SDK-Tools
 - Android SDK Platform-tools
 - Android 2.2 (API 8)

Nach Download und Installation im SDK Manager Menu Tools | Manage AVDs wählen. New

AVD Name: MyEmulator (oder sonst etwas)

Device: 4.0" WVGA (oder etwas anderes, kann nachträglich geändert werden)

Target: Android 2.2

SD Card: Size: 500 MiB

Dann OK drücken (warten)

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

(Unter Windows: Falls der Emulator ohne SDK-Manager gestartet werden soll, einen batch schreiben mit Inhalt:

```
emulator @MyEmulator
```

und einen Link darauf erstellen.)

4. Eclipse downloaden (<http://www.eclipse.org/downloads>) und installieren, z.B. Version Eclipse IDE for Java Developers.
 1. ADT Plugin installieren: Eclipse starten
 2. Help und dann auf Install Software
 3. Auf den Button Add klicken
 4. Name eingeben (frei wählbar)

5. Als Location muss <https://dl-ssl.google.com/android/eclipse> eingetragen werden
6. Developer Tools wählen und mit zweimal Next und Finish abschliessen
7. Nachdem die Plugins installiert wurden und Eclipse neu gestartet ist, muss man dem Plugin noch mitteilen, wo sich das Android SDK befindet:

1. Window und dann auf Preferences
2. Unter Android den Pfad zum <sdkhome> angeben

5. Distribution von <http://www.aplu.ch/android> downloaden und auspacken.
JDroidLib.jar in irgend ein Verzeichnis mit anderen Jars kopieren, z.B. unter Windows 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

Neues Android Project erstellen

Menu *File / New / Android Application Project*

Application name: **MyApp1** (Beispiel)

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

ProjectBuilder starten. Ausfüllen (gemäss Beispiel, anpassen!)



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** (Download von <http://www.aplu.ch/jturtlelib>). Zudem muss *Use Turtle* selektiert sein.

Build drücken. **Nachher im Eclipse Project Manager das Project mit Mausclick wählen und F5 (Refresh) drücken.** Im Package Explorer ist die Source Datei MyApp1.java nun im Package-Unterverzeichnis sichtbar und kann im Editor geöffnet werden.

Aktionen des ProjectBuilder:

AndroidManifest.xml ersetzt

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

<projectroot>\res\values\strings.xml ersetzt

<projectroot>\src<packagepath>\<ApplicationName>.java erzeugt (falls nicht bereits vorhanden)

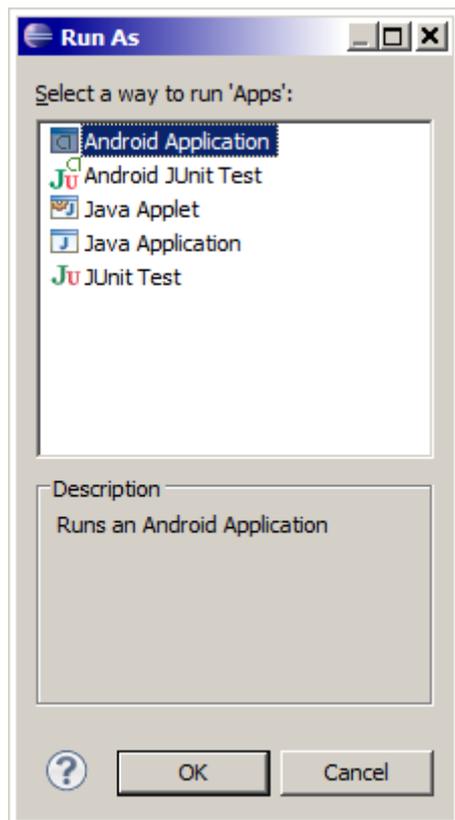
Library file in <projectroot>\libs kopiert (falls libs nicht existiert, wird der Folder erzeugt)

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

Installation/Starten der App auf Smartphone:

Smartphone gemäss untenstehender Anleitung installieren. Smartphone über USB-Kabel anschliessen. **Im Package Explorer muss das Projekt oder das Package selektiert sein.** Run Button klicken. Android Application auswählen.



OK

Das Project wird kompiliert/gepackt und nachfolgend auf den Emulator oder das Smartphone 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 Ausgabekonsole sichtbar machen. App erneut kompilieren/packen und installieren mit Ctrl-F11.

Installation des 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 !)
- 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.

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 sichtbare Applikationsikone ändern will, im Verzeichnis res/drawable die Icone `jdroid_gglogo.png` ersetzen
- 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