

Thema: App-Entwicklung mit MIT App Inventor

Name der Autorin/ des Autors:	Jochen Pogrzeba, StR Max-Weber-Schule, Freiburg
Fach:	Informatik, Wirtschaftsinformatik
Klasse/Jahrgangsstufe:	JG1
Schulart:	Berufliches Gymnasium
Lehrplanbezug:	LPE 9: Objektorientierte Systementwicklung (Fortführung) LPE 9: Objektorientierte Systemanalyse und -entwicklung
Zeitraum:	-
Betriebssystem/e:	Android
Apps:	-
Technische Settings:	Schüler-PC's mit Windows, Chrome oder Firefox-Browser und installierten MIT App Inventor Tools. Schüler-Tablets mit Android und der App MIT AI2 Companion.

Fachliche Ziele:

- Kenntnisse der Besonderheiten der App-Entwicklung.
- Umsetzung des ereignisorientierten Ansatzes der GUI-Entwicklung.

Sonstige Ziele:

- Erweiterung der Sozialkompetenz durch zusammenführende Gruppen- oder Projektarbeit.
- Schulung der gestalterischen Kreativität.
- Förderung der Medienkompetenz.
- Evtl. Schulung der Präsentationskompetenz.
- Evtl. individuelle Förderung.

Diese Unterrichtseinheit ist nicht als direkt umsetzbares Material konzipiert. Vielmehr soll dieses Beispiel der Lehrerin/dem Lehrer die Möglichkeit geben, sich ggf. schnell in das Thema einzuarbeiten. Die direkte Umsetzung des Beispiels mit Schülern ist möglich, aber nicht Hauptziel. Diese Unterrichtseinheit soll den Lehrer motivieren, ein eigenes, evtl. profilbezogenes Beispiel, mit den Schülern umzusetzen. Aus diesem Grunde wurde darauf verzichtet einen Verlaufsplan hinzuzufügen.

App-Entwicklung mit dem MIT AppInventor

- Entwicklung einer BMI-App – (Ein Beispiel für Einsteiger)

Autor: Jochen Pogrzeba, StR
Max-Weber-Schule, Freiburg

Inhaltsverzeichnis:

1	Vorbemerkungen	1
2	Vorbereitungen	2
3	Erstellung der Benutzeroberfläche	3
3.1	Der Designer-Modus.....	3
3.2	Erstellung der GUI für die BMI-App.....	3
4	Programmierung der Logik	4
4.1	Der Blocks-Modus.....	4
4.2	Erstellung der Logik für die BMI-App.....	5
5	Starten der App	6
5.1	Starten über den AI Companion	6
5.2	Starten über den Emulator	6
5.3	Starten über USB	7

1 Vorbemerkungen

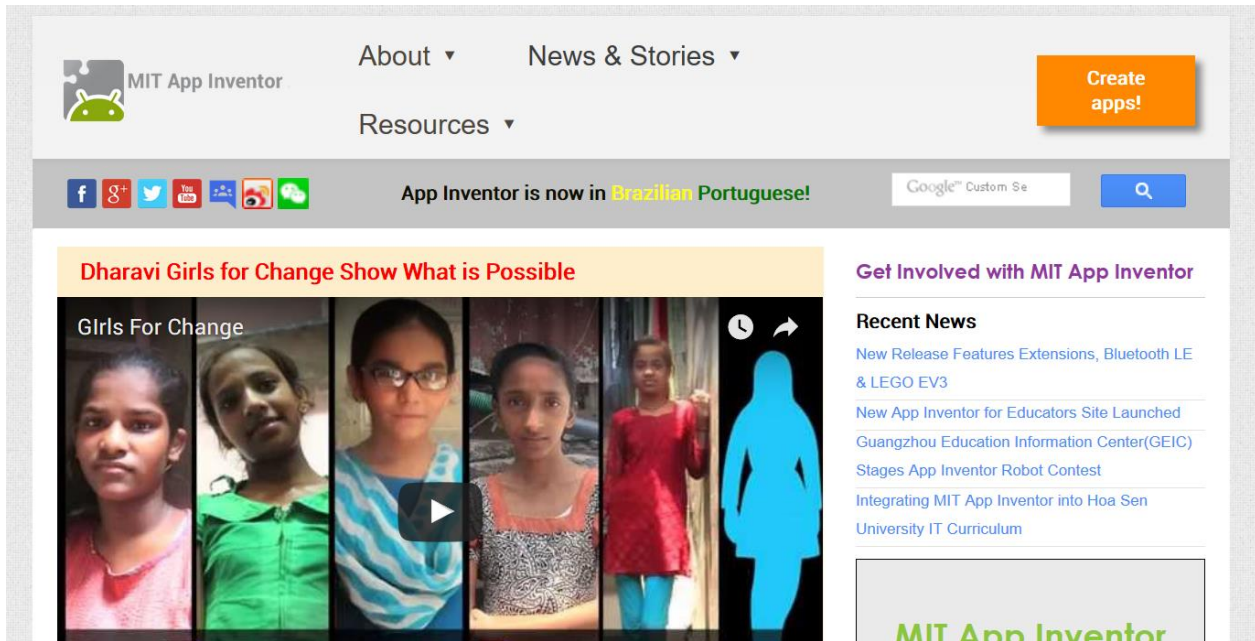
In dieser Unterrichtseinheit wird mit Hilfe der webbasierten Entwicklungsumgebung *MIT AppInventor* ein BMI-Rechner entwickelt, der auf Android-Tablets lauffähig ist. Dieses Projekt ist bewusst simpel gewählt, da es als Einstiegsbeispiel dienen soll. Es enthält keine typischen Tablet-Funktionalitäten wie Scrollen, keine Fachklasse und keine aufwändige GUI-Gestaltung.

Hauptziel dieser Unterrichtseinheit ist es, der Lehrerin/dem Lehrer eine Anleitung mitzugeben, mit der er sich leicht und schnell in die Thematik einarbeiten kann. Eine direkte Umsetzung des vorgestellten Projektes im Unterricht ist zwar möglich, aber nicht primäres Ziel. Der Leser ist also eingeladen, eine Unterrichtseinheit mit einer eigenen Softwareidee durchführen.

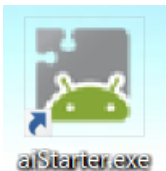
2 Vorbereitungen

Der MIT AppInventor ist ein webbasiertes Tool, mit dem sich Apps für Android entwickeln lassen. Dazu sind weder Kenntnisse des Android-Betriebssystems noch Java-Kenntnisse nötig. Die Gestaltung der Oberfläche erfolgt per Drag und Drop, die Entwicklung der Logik erfolgt mit einem Scratch-ähnlichem Tool.

Der Link <http://appinventor.mit.edu/explore> führt auf die MIT AI-Homepage:



Für den MIT App Inventor werden Chrome- oder Firefox-Browser vorausgesetzt, Internet Explorer wird offiziell nicht unterstützt.



Um die erstellte Anwendung später auf einem Tablet zu starten, benötigt man ein weiteres Tool, welches auf dem Rechner installiert sein muss. Diese **MIT App-Inventor-Tools** findet man unter <http://appinventor.mit.edu/explore/ai2/setup-emulator.html>

Außerdem benötigt man evtl. auf dem mobilen Gerät die **MIT AI2 Companion App** (kostenlos im Google Play Store), falls man die erstellten Apps auf dem mobilen Gerät ausführen will.

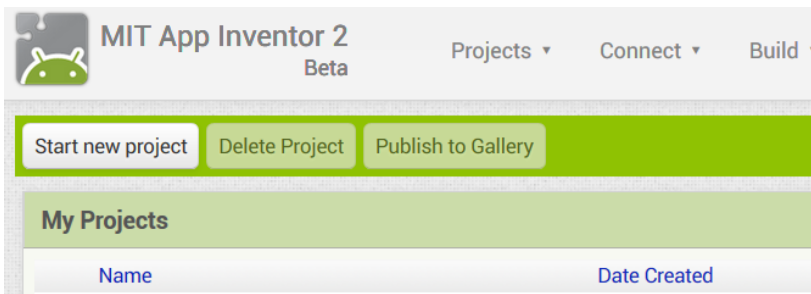


**Create
apps!**

Zur Benutzeroberfläche gelangt man nun über den Button **Create apps!** (rechts oben).

Der MIT AI benötigt eine Anmeldung (z.B. per Google-Konto).

3 Erstellung der Benutzeroberfläche

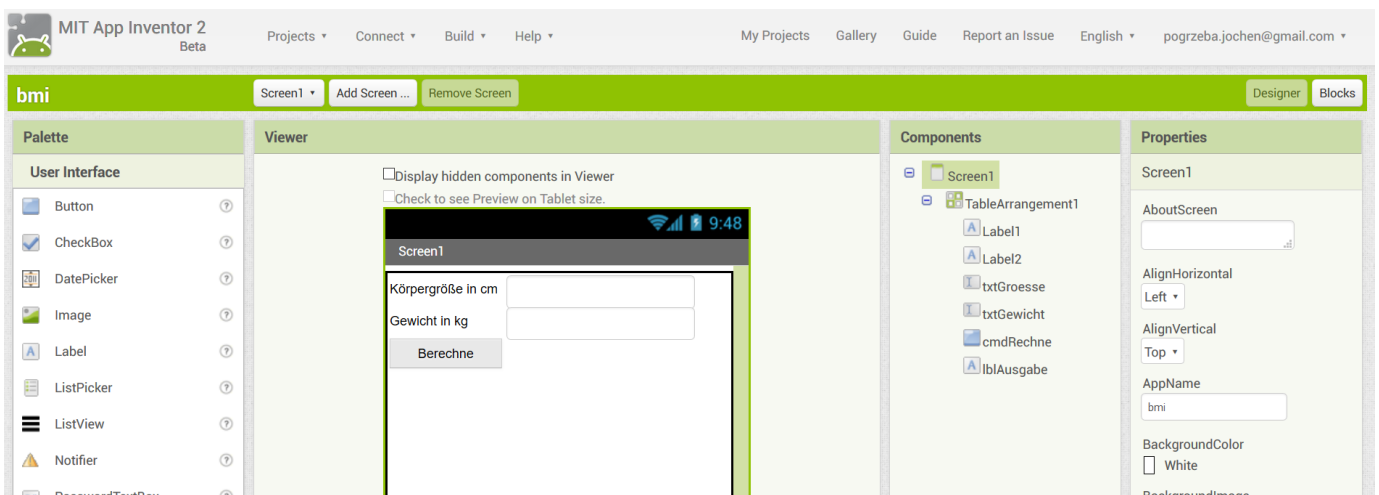


Die Oberfläche ist sehr einfach und übersichtlich gehalten.

Mit *Start new project* beginnt man ein neues Projekt.

3.1 Der Designer-Modus

Im Designer-Modus wird die GUI der App gestaltet.



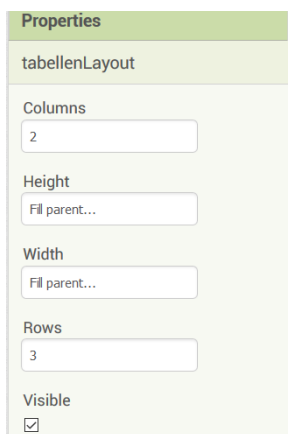
Die GUI-Elemente lassen sich per Drag & Drop auf die GUI ziehen

Im Viewer wird die erstellte GUI gezeigt.

Der Komponentenbaum zeigt alle verwendeten GUI-Elemente.

Hier werden alle Eigenschaften des gewählten GUI-Elementes gezeigt.

3.2 Erstellung der GUI für die BMI-App

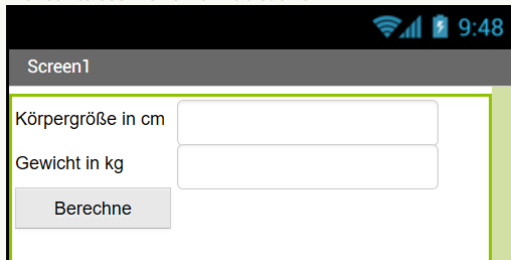


Für eine übersichtliche Anordnung der GUI-Elemente empfiehlt es sich, zuerst ein *TableArrangement*-Layout direkt auf die GUI zu legen.

In ihm kann man die GUI-Elemente wie in einen Setzkasten einordnen.

In den Properties lassen sich Anzahl der Spalten und Zeilen einstellen.

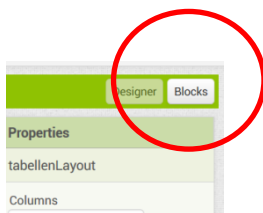
Die Höhe und Breite kann man durch die eingefügten Elemente bestimmen lassen (*Fill parent...*).



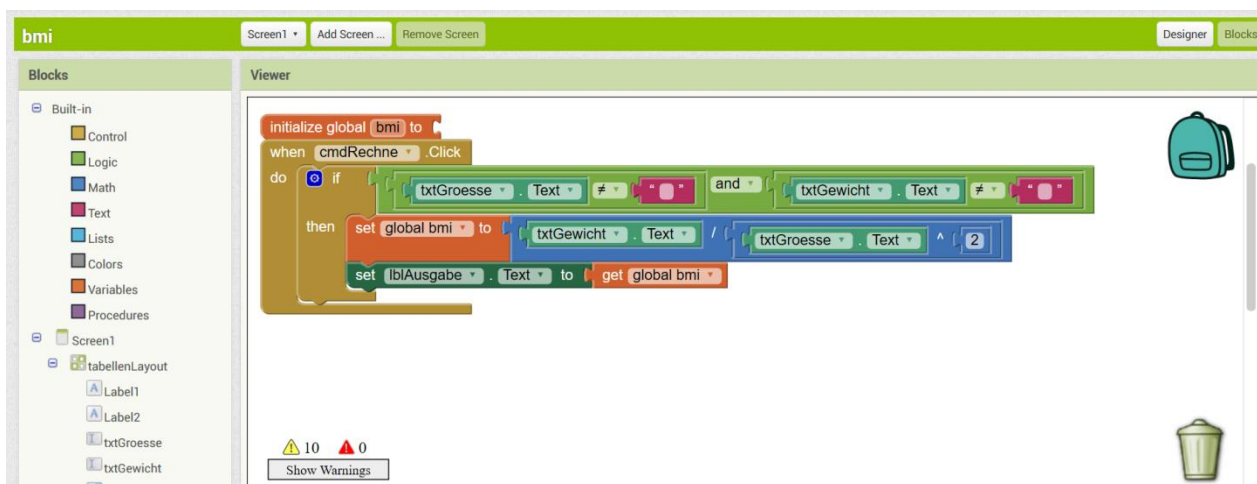
Dann kann man die gewünschten GUI-Elemente in das TableArrangement setzen (eines pro Zelle) und die gewünschten Properties einstellen. Tipp: Achten Sie auf eine sinnvolle Benennung (z.B. *cmdBerechne* oder *txtGroesse*).

4 Programmierung der Logik

4.1 Der Blocks-Modus



Über Blocks (oben rechts) wechselt man in den Blocks-Modus. Hier lässt sich dann die Logik in einer Scratch-ähnlichen Oberfläche realisieren.



Im Blocks-Fenster erscheinen nun alle Elemente, die man zur Programmierung verwenden kann (Kontrollstrukturen, logische Ausdrücke, Funktionen, die eben gesetzten GUI-Elemente)

Im Viewer wird die Programmlogik grafisch dargestellt.

Die Benutzung erschließt sich sofort beim Ausprobieren. Die Blocks werden wie Puzzle-Teile in- und untereinander geschoben.



Im Rucksack können Code-Blöcke zur späteren Verwendung gesichert werden.



Mit dem Papierkorb (südbadisch: Dreggeima) lassen sich Blöcke löschen.

4.2 Erstellung der Logik für die BMI-App

Schritt 1:

```

when cmdRechne .Click
do
  set lblAusgabe .Text to
    txtGewicht .Text /
    txtGroesse .Text ^ 2
  
```

Wenn der Button *cmdRechne* angeklickt wird, dann setze in das Label *lblAusgabe* folgenden Text: Das Ergebnis der Berechnung „Eingabe in *txtGewicht*“ geteilt durch „Eingabe in *txtGroesse hoch zwei*“

Schritt 2:

```

initialize global bmi to
when cmdRechne .Click
do
  set global bmi to
    txtGewicht .Text /
    txtGroesse .Text ^ 2
  set lblAusgabe .Text to
    get global bmi
  
```

Es wird eine Variable *bmi* initialisiert, in der das Rechenergebnis gespeichert (*set to*) wird. Im Ausgabe-Label wird dann der Inhalt dieser Variable gezeigt.

Schritt 3:

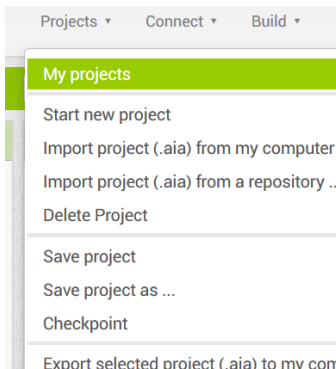
```

when cmdRechne .Click
do
  if
    txtGroesse .Text ≠ "" and
    txtGewicht .Text ≠ ""
  then
    set global bmi to
      txtGewicht .Text /
      txtGroesse .Text ^ 2
    set lblAusgabe .Text to
      get global bmi
  
```

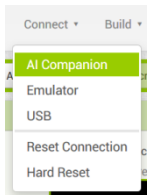
Im letzten Schritt erhält die Logik eine Überprüfung, ob Werte eingegeben wurden. Wenn der Text in *txtGrosse* oder *txtGewicht* ungleich leer ist, dann kann die Berechnung erfolgen.

5 Starten der App

Speichern Sie das Projekt im Menü *Projects* mit *Save project*



5.1 Starten über den AI Companion



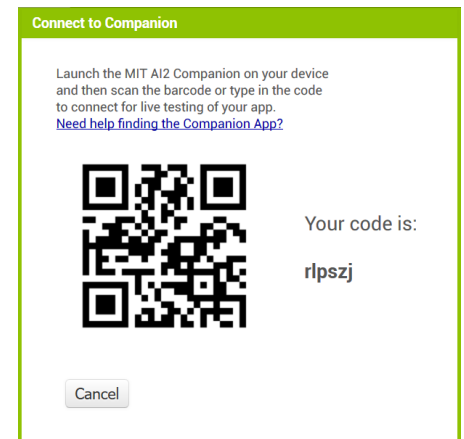
Über *Connect* → *AI Companion* kann man die erstellte App auf dem Tablet ausführen.

Dazu müssen

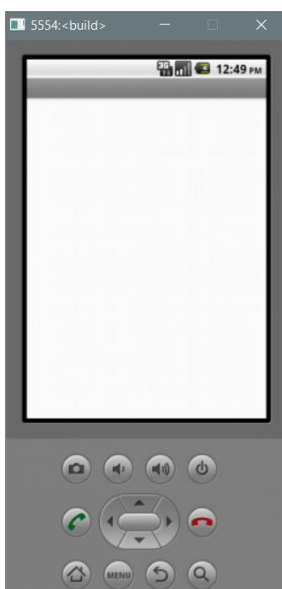
- sich PC und Mobilgerät im selben WLAN befinden,
- auf dem Tablet die App MIT AI2 Companion gestartet sein.

Der MIT AI zeigt auf dem PC einen Ziffern- und QR-Code. Dieser wird in die App auf dem Tablet eingegeben.

In der Praxis dürfte dieses Verfahren problematisch sein, da sich die Arbeits-PC's in den PC-Räumen selten in einem WLAN befinden.



5.2 Starten über den Emulator

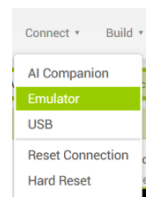


Die erstellte App kann in einem Emulator gestartet werden.

Dazu müssen auf dem PC die App Inventor Tools (*aiStarter.exe*) gestartet sein.

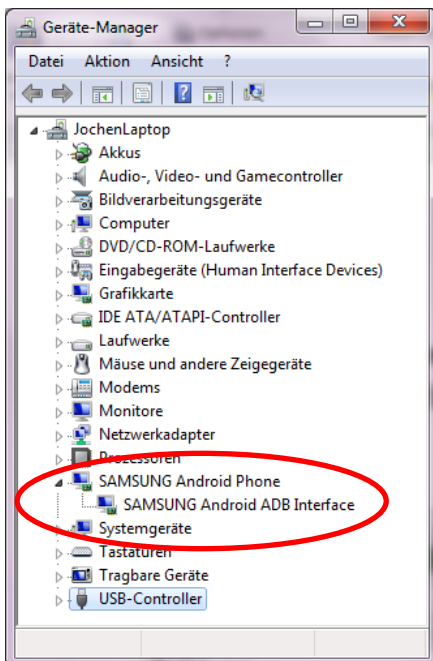
Der Emulator wird im Menü *Connect* gestartet.

Die App startet dann automatisch.



5.3 Starten über USB

Für die weiteren Schritte muss das Tablet über ein USB-Kabel am PC angeschlossen sein.

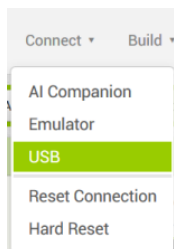


Die Kommunikation mit angeschlossenen Tablets wird von der *Android Debug Bridge (ADB)* erledigt. Über die ADB können wir unsere selbst erstellten Android Apps auf ein Android Gerät übertragen und installieren.

Der PC muss das Tablet als Android ADB Interface erkennen. Dies kann man im Geräte-Manager überprüfen (*System-Steuerung* → *Geräte-Manager*). Dort sollte das Mobilgerät mit der Kennzeichnung "ADB Device" (oder ähnlich je nach Gerät) eingetragen sein.

Sollte dies nicht der Fall sein, muss der ADB-Treiber installiert werden. Diesen installieren Sie entweder per Rechtsklick auf das Gerät oder über die Android-Developer-Website.

(<http://developer.android.com/tools/extras/oem-usb.html#Drivers>)



Über das Menü *Connect* wird die App an das angeschlossene Tablet geschickt und dort ausgeführt.

Dazu müssen auf dem PC die App Inventor Tools (*aiStarter.exe*) gestartet sein.

6 Weitere Quellen

Auf der Homepage <http://appinventor.mit.edu/explore> befinden sich weitere Tutorials und Anwendungsbeispiele zum Ausprobieren.

Get Started



Follow these simple steps to build your first app.

Start

Tutorials



Step-by-step guides show you how to build all kinds of apps.

Tutorials

Teach



Teachers, find out about curriculum and teaching resources.

Teach

Forums



Join community forums to get answers to your questions.

Forums