

1 Vorbereitung Android-Handy:

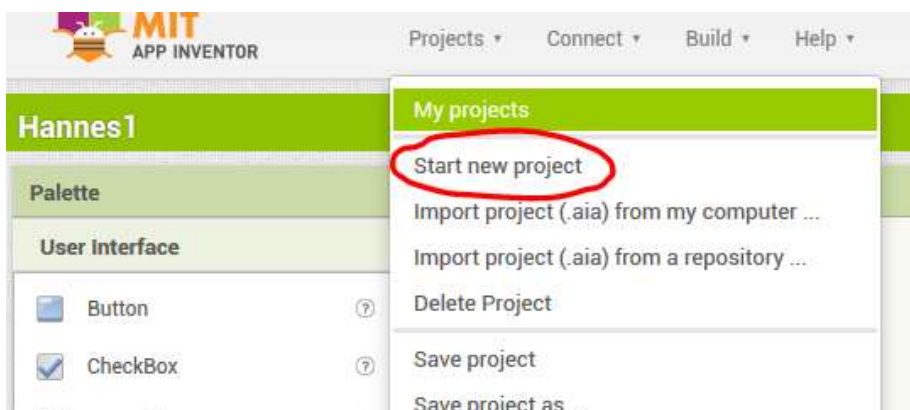
Android-APP: „MIT AI2 Companion“ installieren.

2 Computer:

Wichtig: Computer und Handy müssen im gleichen Netzwerk sein. Sonst gibt es einen „Internen Error“.

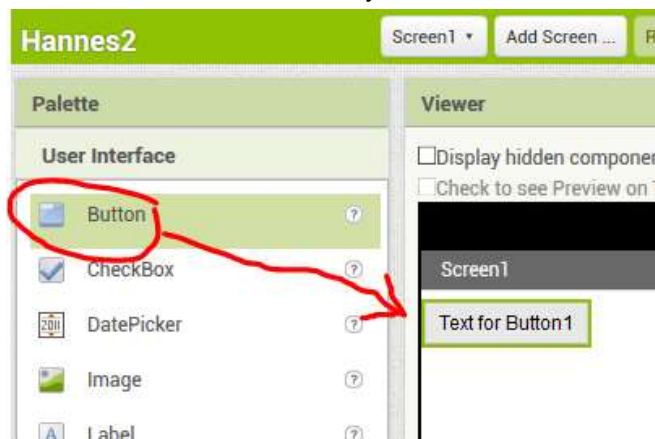
AppInventor starten (<http://ai2.appinventor.mit.edu/>) (am Besten mit **Firefox**) und mit dem Google-Konto anmelden.

2.1 Neues Projekt erstellen:

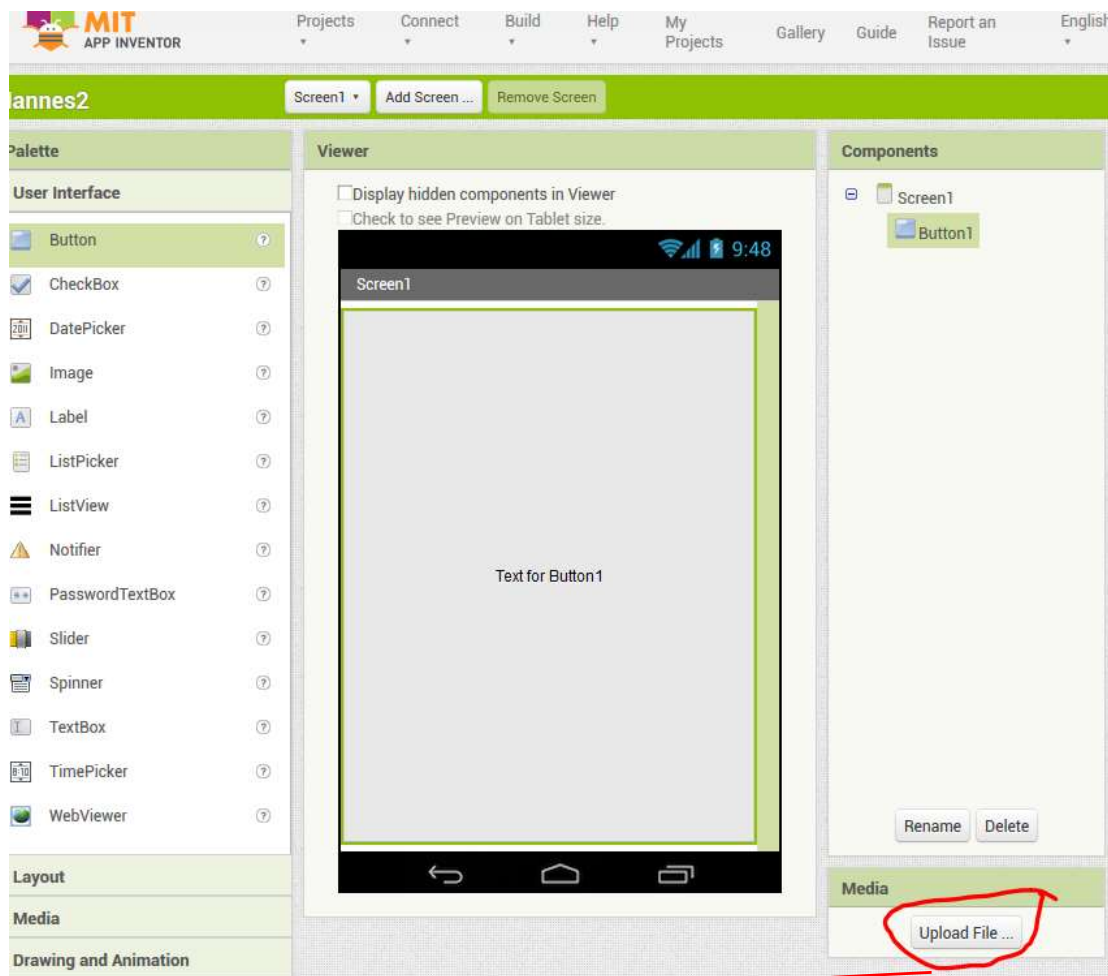


Projektname ist am Besten der Vorname des Schülers.

Button auf den virtuellen Handyschirm ziehen.



2.2 Grafiken in das Projekt hochladen.

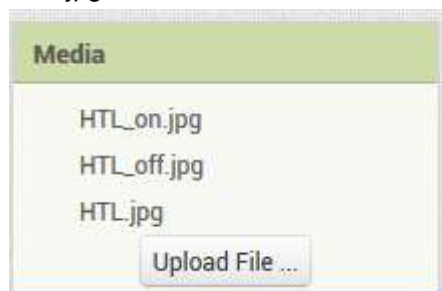


Die Dateien ins Projekt kopieren:

HTL_on.jpg

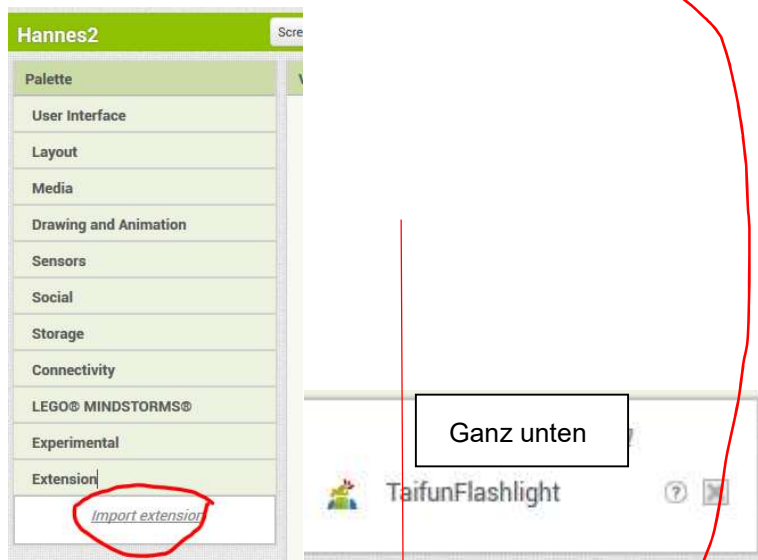
HTL_off.jpg

HTL.jpg

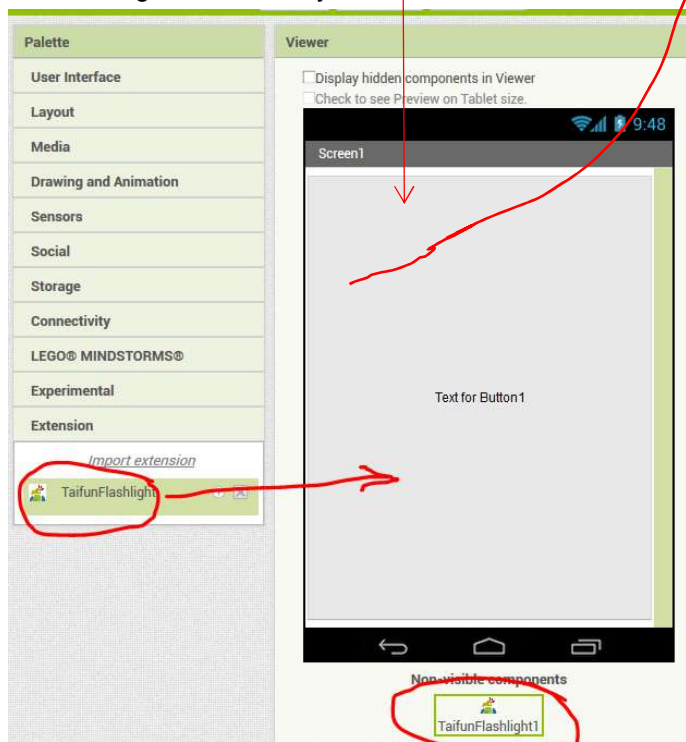


2.3 Erweiterung (Extension) für die Blitzled importieren

[com.puravidaapps.TaifunFlashlight.aix](https://www.puravidaapps.com/TaifunFlashlight.aix)

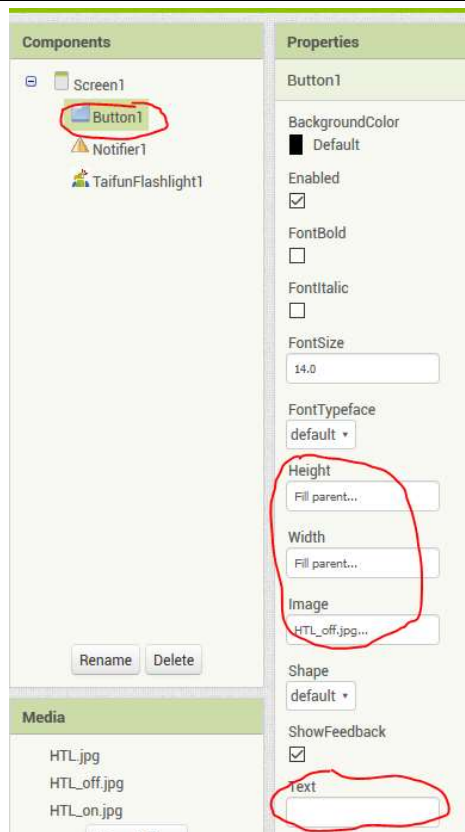


Erweiterung auf das Handy ziehen – es wird dann unter dem Handybildschirm abgelegt:



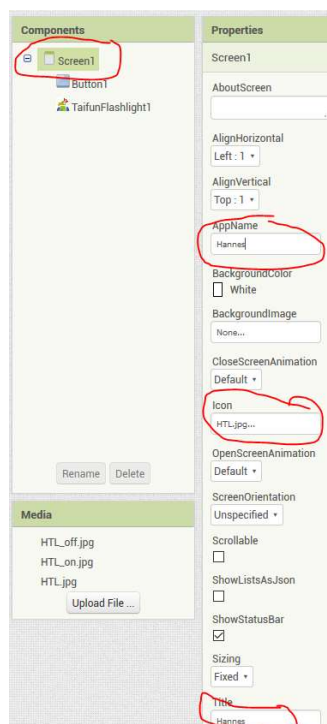
2.4 Eigenschaften des Buttons anpassen

dazu muss zuerst der Button1 markiert sein



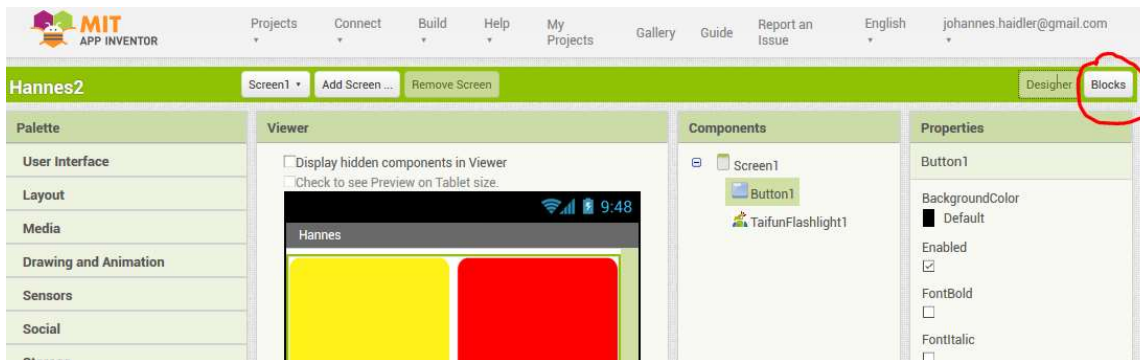
2.5 Eigenschaften des Screens anpassen

dazu muss zuerst der Screen1 markiert sein



2.6 Programmieren

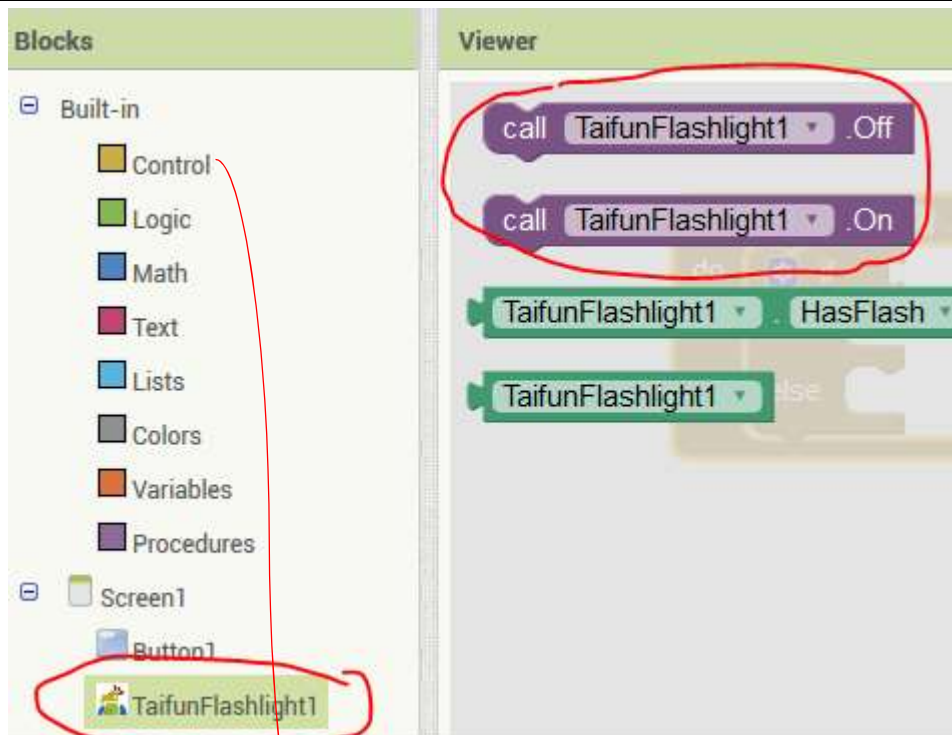
In den Programmierbereich wechseln (Blocks):



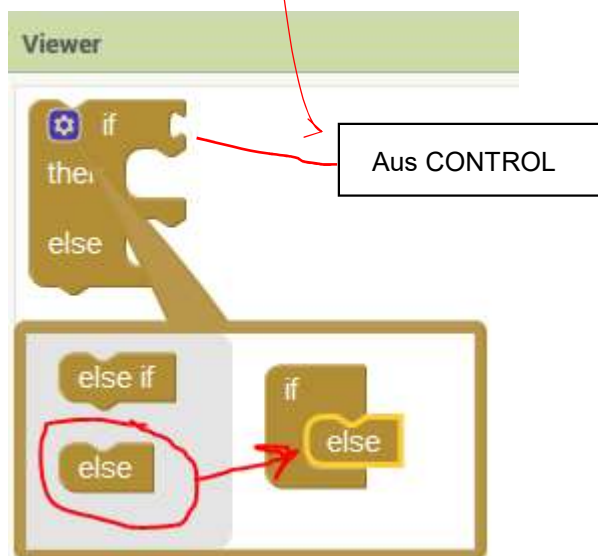
Programmieren lt. Vorlage auf der letzten Seite.

Die einzelnen Bausteine kommen von den entsprechenden Elementen:





Beim If-Then-Else: Blaues Zahnrad



Den Else-Block in das if einbauen - wieder blaues Zahnrad

Alles andere sollte mit der Vorlage fast selbsterklärend sein. Die Farben der Blöcke geben eine gute Orientierung.

Es reicht, wenn man den ersten Teil macht:

```

when Button1 .Click
do
  if Button1 .Image = "HTL_off.jpg"
  then
    set Button1 .Image to "HTL_on.jpg"
    call TaifunFlashlight1 .On
  else
    set Button1 .Image to "HTL_off.jpg"
    call TaifunFlashlight1 .Off
  
```

Abbildung 1: Programmteil 1 – eigentliche Bedienung

Der zweite ist nur dazu da um abzufragen ob das Handy eine Blitz-Led hat und wenn nein eine entsprechende Meldung auszugeben. Da fast alle Handys einen Blitz haben kann man darauf verzichten.

```

when Screen1 .Initialize
do
  if not TaifunFlashlight1 .HasFlash
  then
    call Notifier1 .ShowAlert
      notice "Leider kein Blitz vorhanden"
  else
    set Button1 .Image to "HTL_off.jpg"
  
```

Abbildung 2: Programmteil 2 – Initialisierung und Abfrage ob Blitzled vorhanden

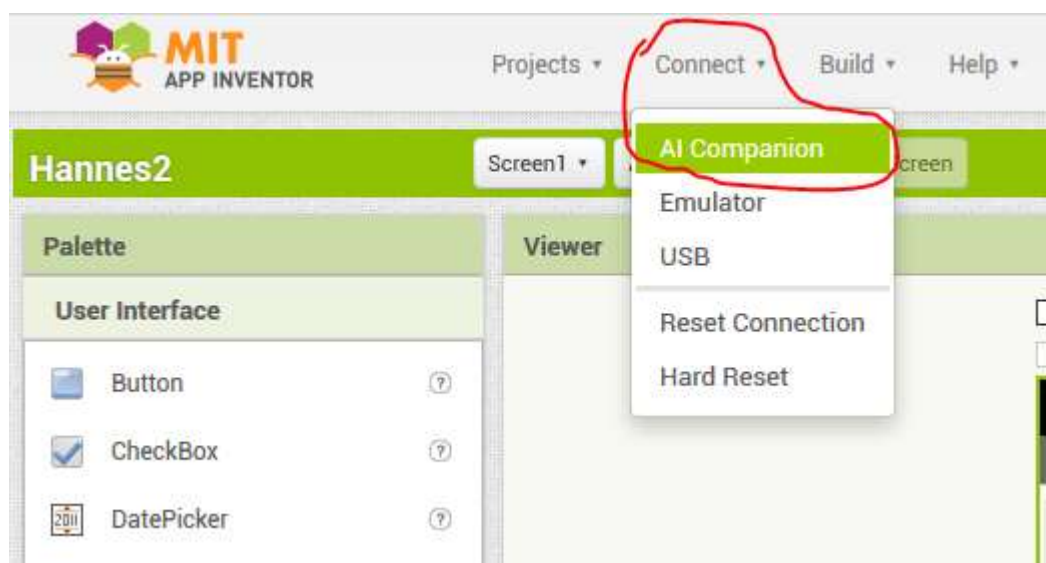
Der dritte Teil ist dazu da um das Programm richtig zu beenden und damit eine ev. eingeschaltete Taschenlampe beim Beenden auszuschalten. Dieser Teil ist auch nicht unbedingt nötig.

```

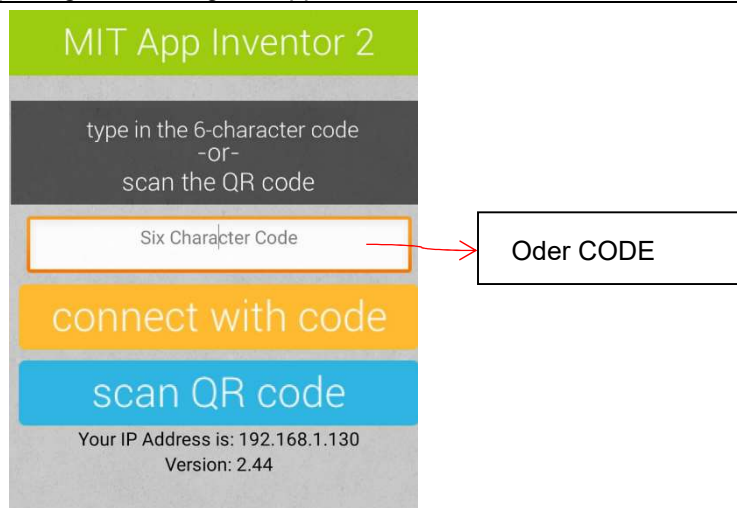
when Screen1 .BackPressed
do
  call TaifunFlashlight1 .Off
  close application
  
```

Abbildung 3: Programmteil 3 – sauberer Programmausstieg

2.7 Verbinden der Browser-App mit dem Android-Handy



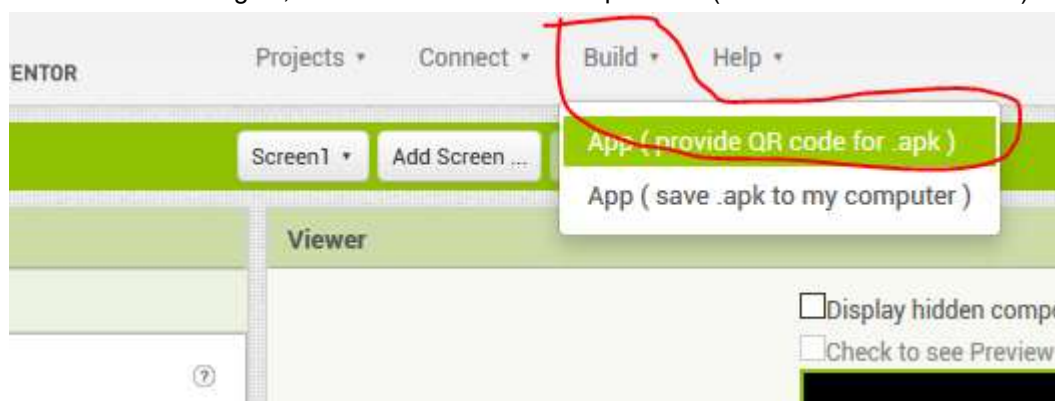
Über Connect->AI Companion mit dem Handy verbinden. Dazu wird am Bildschirm ein **QR-Code** angezeigt. Dieser kann mit der „**MIT AI2 Companion**“-App gescannt werden **oder** man kann auch den **sechsstelligen Code** abtippen und dann verbinden.



Ab sofort sollten alle Änderungen im Browser auch am Handy synchronisiert sein und man kann die Handy-App testen.

2.8 Installationsdatei erzeugen (Dauerhafte APP)

Wenn die APP fertig ist, kann man mit Build eine apk-Datei (Android-Installationsdatei) erzeugen.



Mit Build->App (provide QR code for .apk) wird ein QR-Code angezeigt, in dem ein Link zum Download der apk-Datei zur Verfügung gestellt wird. Dieser kann mit einem Browser geöffnet werden – es wird die apk-Datei heruntergeladen. Anschließend kann diese APP installiert werden. Dabei ist zu beachten, dass Android grundsätzlich keine **fremden Installationsquellen erlaubt**. Man kann das aber durch entsprechende Bestätigungen relativ leicht ausschalten. Dann ist die App selbständig auf dem Handy installiert und kann jederzeit verwendet werden.

Mit dem Menüpunkt Build->App (save .apk to my computer) wird die Datei erstellt und auf dem Computer gespeichert. Dann kann man diese Datei beliebig weiter versenden und alle können diese App installieren.

Achtung:

Wenn die Verbindung abbricht, muss das manchmal wiederholt werden. Hin und wieder sind die oberen drei Einträge grau. Dann muss mit „Reset Connection“ die Verbindung zurückgesetzt werden.

3 Programmiervorlage

ai2.appinventor.mit.edu/

Android-APP: MIT AI2 Companion

```

when Button1 .Click
do
  if Button1 .Image = "HTL_off.jpg"
  then
    set Button1 .Image to "HTL_on.jpg"
    call TaifunFlashlight1 .On
  else
    set Button1 .Image to "HTL_off.jpg"
    call TaifunFlashlight1 .Off

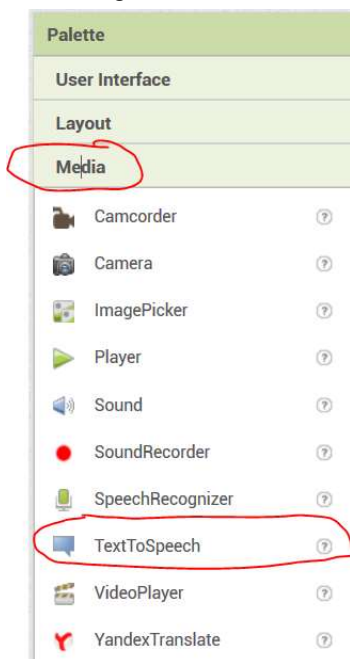
when Screen1 .Initialize
do
  if not TaifunFlashlight1 .HasFlash
  then
    call Notifier1 .ShowAlert
      notice "Leider kein Blitz vorhanden"
  else
    set Button1 .Image to "HTL_off.jpg"

when Screen1 .BackPressed
do
  call TaifunFlashlight1 .Off
  close application
  
```

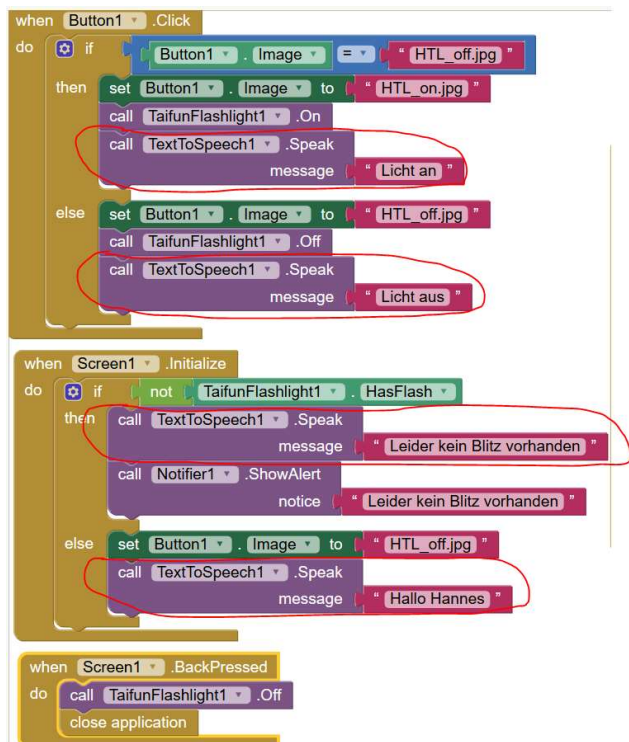
4 Erweiterungen

4.1 Textausgabe

Wenn noch Zeit ist, kann man die Taschenlampe noch um eine Sprachausgabe erweitern. Dazu muss im Designer noch die **TextToSpeech-Komponente auf das Handy gezogen** werden:



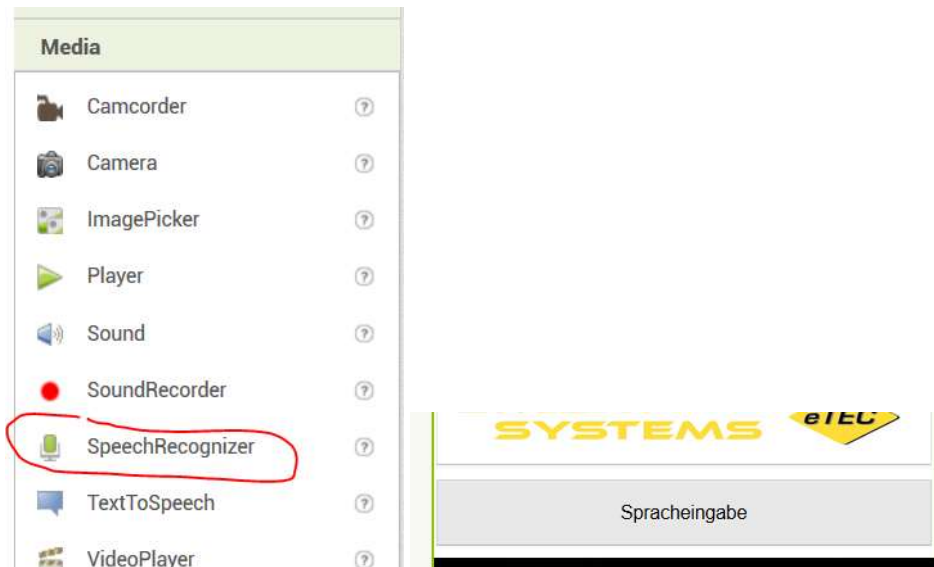
Und die entsprechenden Blöcke im Programm ergänzt werden.



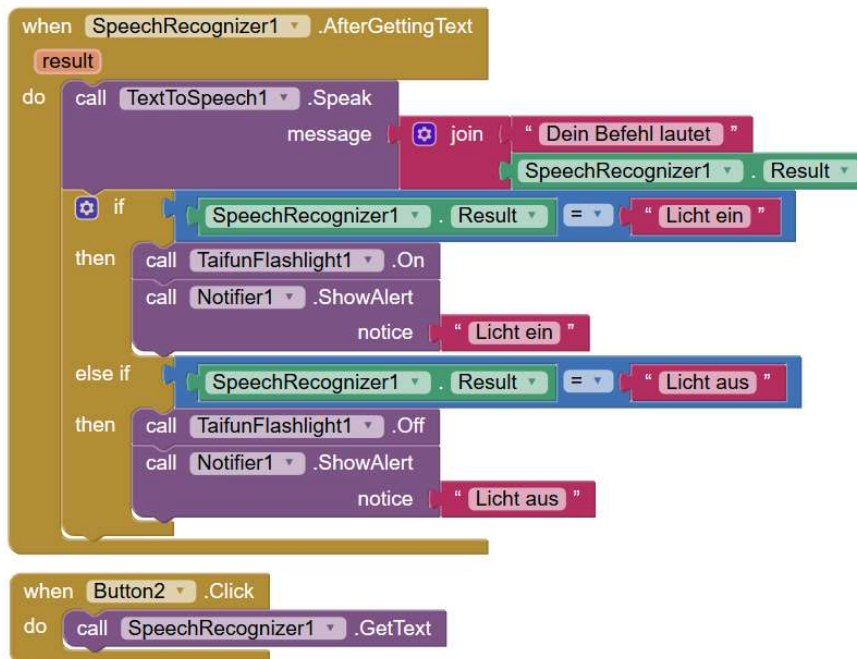
Selbstverständlich können die Texte beliebig angepasst werden.

4.2 Sprachbefehle

Wenn auch noch Sprachbefehle ergänzt werden sollen geht das folgendermaßen. Im Designer **Komponente SpeechRecognition und einen Button2** ergänzen:



Das Programm ist wie folgt zu erweitern:



Dann kann man auf den Button "Spracheingabe" drücken und den Befehl "Licht ein" oder "Licht aus" sprechen.

Viel Erfolg!