



Froschhüpfen – Teil II

Programmiere mit dem MIT App Inventor:

Wenn du Hilfe brauchst oder etwas kontrollieren möchtest, schaue dir die Zwischenergebnisse für die einzelnen Aufgaben an. Sie liegen im Klassenzimmer aus.

1. *Erzeuge eine neue Liste und gib ihr den Namen „sprungweiten“.*
2. *Die Eingabe der Sprungweiten soll über eine TextBox möglich sein. Durch Klick auf einen Button wird die eingegebene Weite als neues Element an die Liste angehängt und alle Elemente der Liste in einer ListView angezeigt.*

Programmiere zwei Buttonereignisse.

3. *Wenn auf den Button btMaximum gedrückt wird, soll der maximale Wert der Liste in einem Label ausgegeben werden.*
4. *Wenn auf den Button btMittelwert gedrückt wird, soll der Mittelwert der Sprungweiten in einem Label ausgegeben werden.*
5. *Erweitere dein Projekt um die Möglichkeit, auch zufällige Sprungweiten erzeugen zu lassen.*
6. ****** Suche dir eine Partnerin / einen Partner in deiner Gruppe, der auch schon fertig ist, und überlegt euch dann, wie man die Sprungweiten in der Liste sortieren könnte. Wie in einer Highscoreliste soll die größte Sprungweite am Anfang und die kleinste Sprungweite am Ende stehen. Schreibt eure Ideen auf ein Blatt.*



Zwischenergebnisse

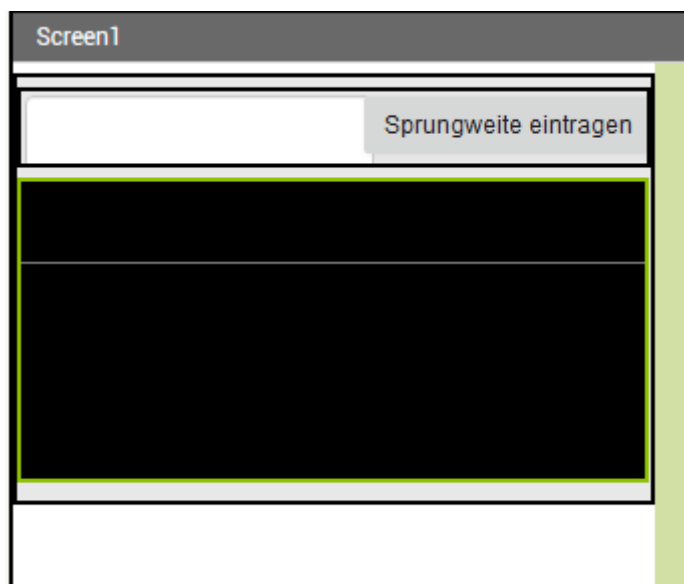
Aufgabe 1

Erzeugen und Initialisieren einer neuen, noch leeren Liste „sprungweiten“. (Den ersten Teil findest du unter „Variables“, den zweiten Teil unter „Lists“.)

initialize global `sprungweiten` to `create empty list`

Aufgabe 2

Der Screen (hier mit Nutzung zweier Layouts: einem vertikalen für die gesamte Aufgabe, mit Höhe 50 % und einem horizontalen für Textfeld und Button):



Möchtest du die Hintergrundfarbe der ListView verändern, denke daran, auch die Schriftfarbe zu verändern.

Setze einen Haken bei der Eigenschaft NumbersOnly des Textfeldes, dann können nur Zahlen eingegeben werden.

Code:

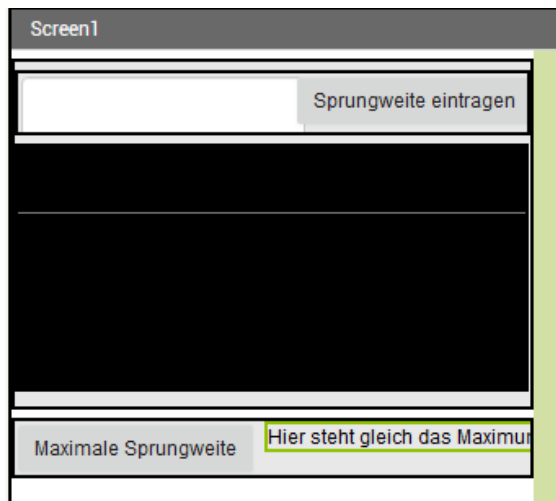
```

when btEintragen .Click
do
  add items to list list item
  get global sprungweiten
  tbEingabeWeite .Text
  set lvSprungweiten .Elements to get global sprungweiten
  
```



Aufgabe 3:

Auf dem Screen kannst du ein horizontales Layout mit Button und Label ergänzen:



Wie das Maximum bestimmt werden kann, habt ihr euch schon überlegt. Euer Vorgehen muss jetzt nur noch in den richtigen Code übersetzt werden.

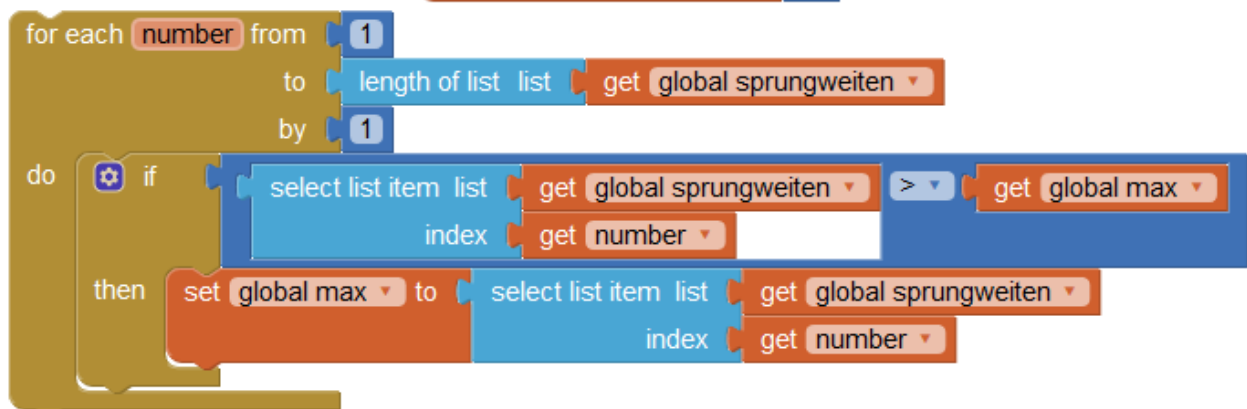
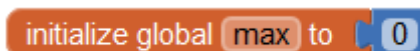
Zum Durchlaufen der gesamten Liste bietet sich eine sogenannte for-Schleife an. Du kennst diese Schleifenart aus Scratch: wiederhole x-mal...

Hier können wir sogar einen Start- und einen Endwert eingeben. Startwert ist 1 (für das erste Element der Liste) und Endwert die Länge der Liste (dort steht das letzte Element der Liste):



In jedem Schleifendurchlauf lassen wir das Element der Liste an dieser Stelle mit dem aktuell maximalen Wert vergleichen. Und wenn das Element größer ist, wird es in der Variable *max* gespeichert.

Initialisieren der Variable:





Jetzt fehlt nur noch die Ausgabe des Maximums, wenn die Schleife ganz durchlaufen ist.

```

set lbMaximum . Text to join ( " Maximum: "
                             get global max
    
```

Insgesamt könnte dein Ergebnis jetzt so aussehen:

```

initialize global max to 0

when btMaximum . Click
do
  for each number from 1
    to length of list list
    by 1
  do
    if select list item list
       index get global sprungweiten > get global max
    then
      set global max to select list item list
                       index get global sprungweiten
                       get number
  set lbMaximum . Text to join ( " Maximum: "
                               get global max
    
```

Der MIT App Inventor (<http://appinventor.mit.edu>) wurde ursprünglich von einem Entwicklerteam um Mark Friedman und Hal Abelson bei Google entwickelt und 2012 an das MIT übergeben.

Der MIT App Inventor wird unter der Creative Commons Attribution-ShareAlike 3.0 Unported License veröffentlicht: <https://creativecommons.org/licenses/by-sa/3.0>

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>. Scratch ist lizenziert unter CC BY-SA 2.0 (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).



Aufgabe 4:

Bei der Bestimmung des Mittelwertes geht ihr wieder so vor, wie ihr es euch schon überlegt habt.

Setzt zunächst wieder einen Button und ein Label auf den Screen.

Initialisieren einer Variable *summe*:

initialize global *summe* to 0

Die Liste kannst du wieder mit einer for-Schleife durchlaufen. Jedes Element wird zum aktuellen Wert der Variable *summe* addiert.

```

for each number from 1
  to length of list list get global sprungweiten
  by 1
do
  set global summe to
    get global summe +
    select list item list
      index get global sprungweiten
      get number

```

Am Ende muss noch der Mittelwert berechnet und ausgegeben werden.

```

set lbMittelwert . Text to
  get global summe /
  length of list list get global sprungweiten

```

Insgesamt könnte dein Ergebnis so aussehen:

```

initialize global summe to 0

when btMittelwert . Click
do
  for each number from 1
    to length of list list get global sprungweiten
    by 1
  do
    set global summe to
      get global summe +
      select list item list
        index get global sprungweiten
        get number
  set lbMittelwert . Text to
    get global summe /
    length of list list get global sprungweiten

```

Aufgabe 5:

```

when btZufall . Click
do
  add items to list list
    item get global sprungweiten
    random integer from 20 to 140
  set lvSprungweiten . Elements to
    get global sprungweiten

```