Froschhüpfen – Teil I und II

Auftrag 1:

1. *Lasse deinen Frosch soweit wie möglich springen und notiere seine Sprungweite auf einen kleinen Zettel, den du in deine Streichholzschachtel mit der Aufschrift „Sprungweite“ legst.*
2. *Bei einem weiteren Sprung schreibst du die neue Sprungweite auf einen zweiten Zettel und vergleichst dann den Wert mit dem Wert auf dem Zettel in deiner Schachtel. Ist der neue Wert höher, legst du den zweiten Zettel statt des ersten in die Schachtel.*
3. *Wiederhole das so lange, bis der Frosch zehn Mal gesprungen ist.*
4. *Welchen Wert hast du am Ende in deiner Schachtel liegen?*

Froschhüpfen – Teil I

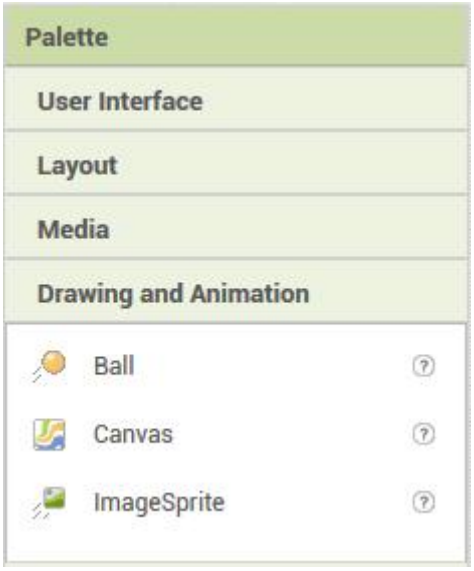
Auftrag 2 – Programmiere mit dem MIT App Inventor:

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

1. *Setze eine Leinwand (Canvas) auf deinen Screen und setze darauf zunächst einen grünen Ball (der unser Frosch sein soll).*
2. *Beim Klick auf einen Button soll der Frosch eine zufällige Pixelanzahl nach vorne springen. Diese Sprungweite wird einer Variable „sprungweite“ als Wert übergeben.*
3. *Beim Klick auf einen weiteren Button wird der Frosch in seine Ausgangsposition zurückgesetzt.*
4. *Bei jedem weiteren Klick springt der Frosch erneut. Die neue Sprungweite soll aber nur dann als Wert einer Variable max übernommen werden, wenn er weiter gesprungen ist, als bisher.*
5. *Lasse jeweils die aktuelle Sprungweite und die momentan in der Variable gespeicherte maximale Sprungweite angeben. Nutze als Ausgabekomponenten Labels.*
6. *Zur Übung kannst du die Sprünge des Frosches mitzählen und anzeigen lassen.*

Zwischenergebnisse

Aufgabe 1



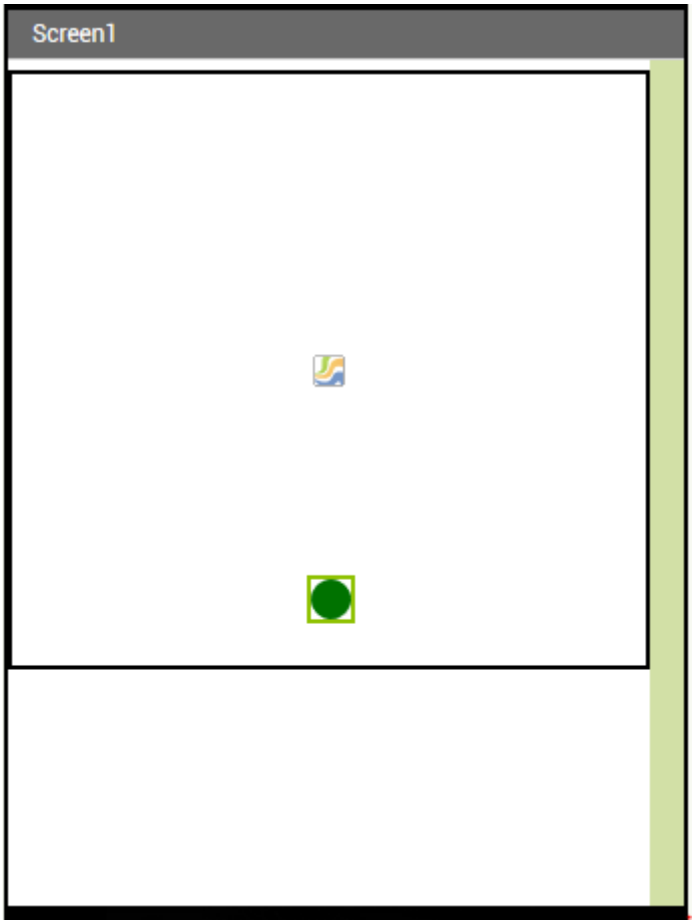
Sowohl die Leinwand (Canvas) als auch den Ball (für den Frosch) findest du in der Komponentenpalette „Drawing and Animation“.

Übrigens könntest du statt des Balls ein ImageSprite nutzen. In diese Komponente kannst du tatsächlich ein Bild laden, das z.B. einen Frosch zeigt.

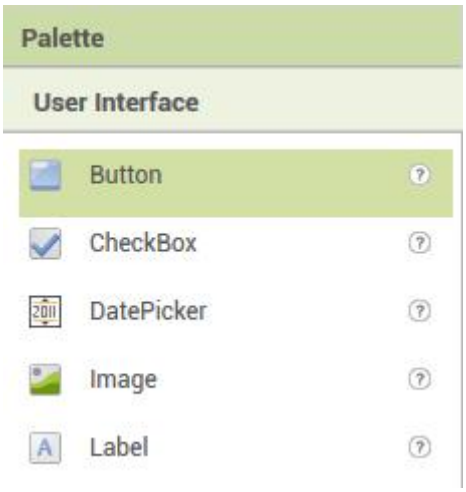
Achte beim Laden von Bildern aus dem Internet auf Urheberrechte!

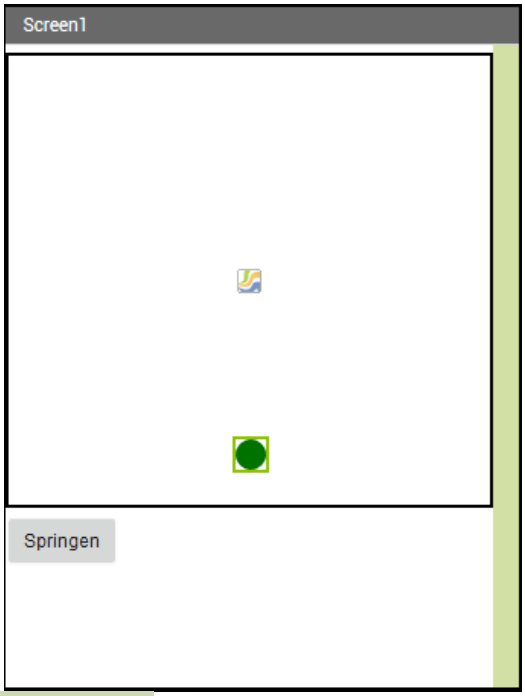
Setze eine Canvas auf den Screen und verändere zunächst die Breite so, dass der gesamte Bildschirm ausgefüllt wird (fill parent). Setze die Höhe auf 70 %.

Setze dann einen Ball auf die Canvas, färbe ihn grün, vergrößere den Radius und nenne ihn frosch.



Aufgabe 2

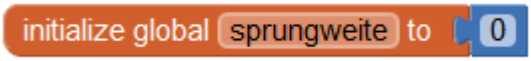




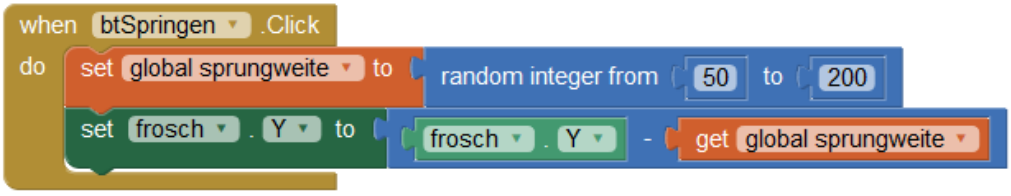
Für den Button brauchst du das Click-Ereignis.

Du findest alles, was du für eine Komponente (hier den Button) brauchst, bei den Blöcken, wenn du die jeweilige Komponente auswählst.

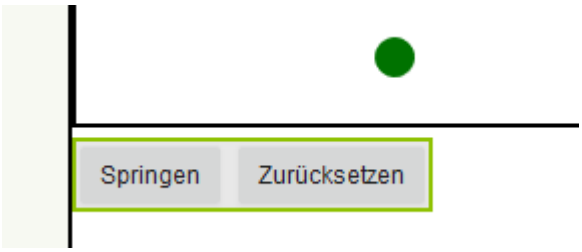
Initialisieren der Variable sprungweite:



Möglicher Code:



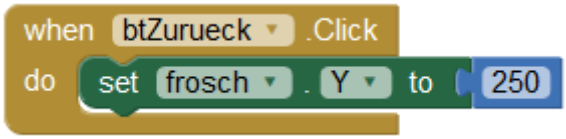
Aufgabe 3:



Um die Buttons nebeneinander setzen zu können, musst du aus der Palette „Layout“ ein horizontales Layout auf den Bildschirm setzen und die Buttons nacheinander hineinziehen.

In der Komponentenliste sind die beiden Buttons dem Layout untergeordnet:

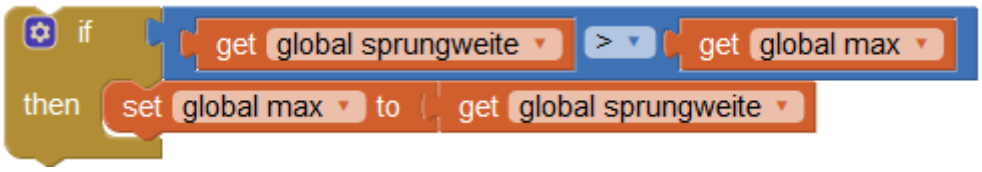
Die y-Koordinate der Ausgangsposition des „Frosches“ kannst du in den Eigenschaften im Designer ablesen, wenn du dort den Frosch an die entsprechende Stelle gesetzt hast.



Aufgabe 4:

Du brauchst eine neue Variable, die die maximale Sprungweite speichert.

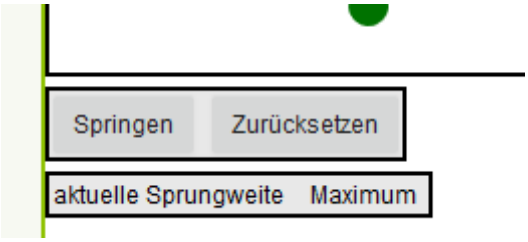
Baue dann in das ButtonClick-Ereignis von Aufgabe 2 eine Verzweigung ein.

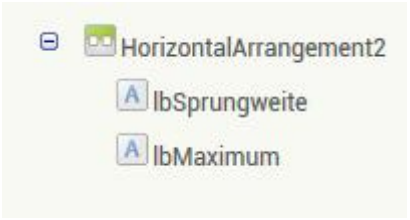


Das Ergebnis könnte folgendermaßen aussehen:

Aufgabe 5:

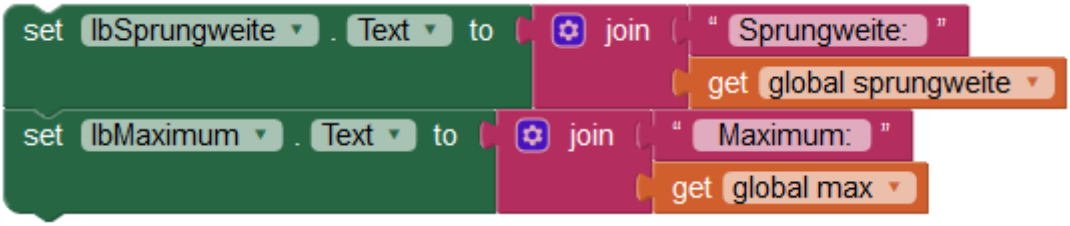
Nutze für die beiden Labels wieder ein horizontales Layout.



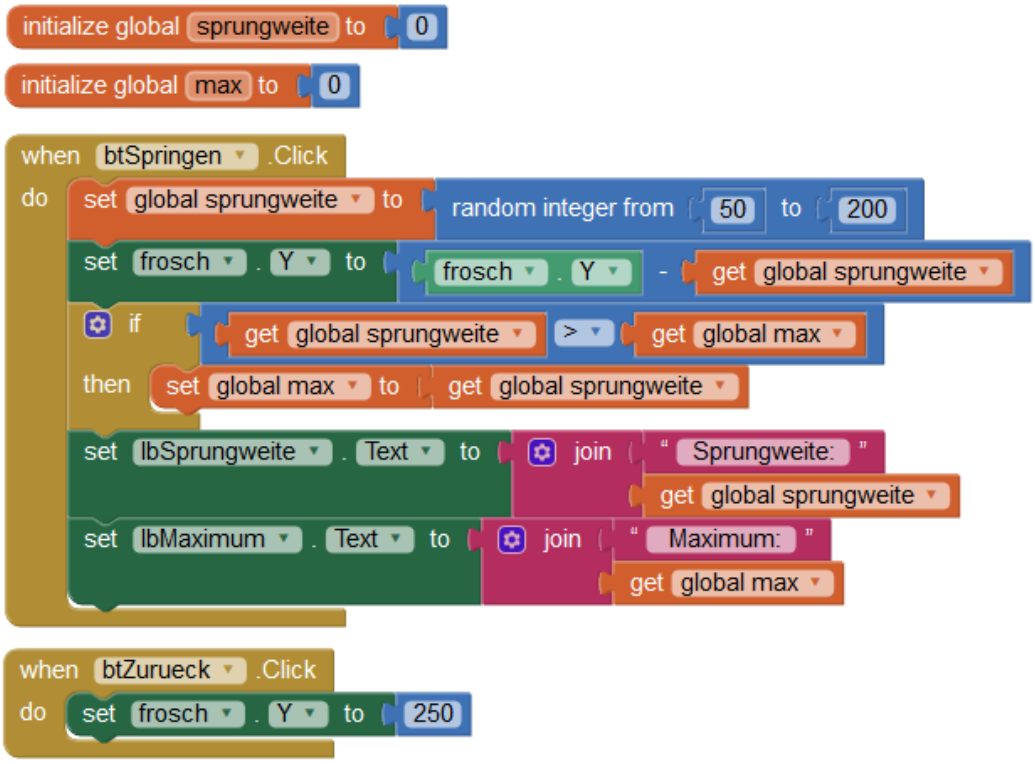


Füge dann die beiden Blöcke noch in das Ereignis aus Aufgabe 4 ein.

So werden nur die beiden Zahlen angezeigt und die Nutzerin / der Nutzer weiß nicht, wofür sie gut sind. Füge deshalb noch Text dazu ein.



Insgesamt sollte der Code jetzt etwa so aussehen:



Aufgabe 6:

Du bekommst diese Aufgabe sicher selbst hin.

Hier nur kleine Tipps:

Nutze eine neue Variable und ein weiteres Label zum Anzeigen.

Überlege dir, welchen Wert die Variable zu Beginn (Initialisierung) speichert und wie und wann sie verändert werden muss.