



Froschhüpfen – Teil I

Programmiere in Scratch:

1. Ein Frosch, der ganz links auf der Bühne sitzt, „springt“ beim Anklicken mit der Maus eine zufällige Weite nach rechts.
2. Das erste Ergebnis wird einer Variable „Sprungweite“ übergeben.
(Hilfekarten 1 – 11)
3. Ab dem zweiten Sprung wird überprüft, ob die neue Weite größer ist. Falls ja, wird der Wert der Variable ersetzt, sonst bleibt er gleich.
(Hilfekarten 12 – 18)
4. Zur Übung kann eine weitere Variable als Zähler genutzt werden, um die Sprünge des Frosches mitzuzählen.



Froschhüpfen – Hilfekarten – der Frosch springt eine zufällige Weite

Überlege dir zunächst, welches Ereignis du brauchst.

1



Überlege dir jetzt, was passieren soll, wenn auf den Frosch geklickt wird.

2

Der Frosch „springt“ nach rechts.

Welche Eigenschaft musst du verändern?

3

Die x-Position des Frosches.



Die Sprungweite soll zufällig sein. Wenn wir z.B. eine Sprungweite zwischen 30 und 200 Pixeln (Bildpunkten) haben wollen, brauchen wir eine Zufallszahl.

4





Die Zufallszahl können wir jetzt direkt einsetzen.

5

ändere x um Zufallszahl von 30 bis 200

Der Wert soll in einer Variable „sprungweite“ gespeichert werden.

6

Erzeuge unter „Daten“ eine neue Variable „sprungweite“.

Weise der Variable die Zufallszahl zu.

7

setze sprungweite auf Zufallszahl von 30 bis 200

Ändere die x-Position jetzt um den Wert der Variable.

8

ändere x um sprungweite



Dein Code sollte jetzt so aussehen

9



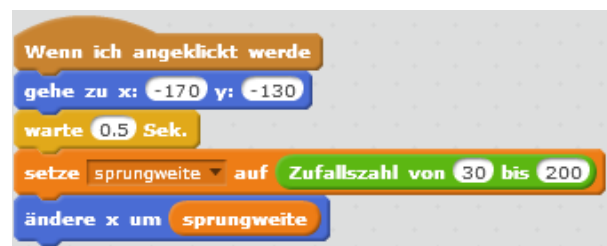
Teste den Code, in dem du den Frosch anklickst. Was würdest du noch verändern?

10

Der Frosch sollte vor jedem Sprung wieder an die Ausgangsposition zurück und dort kurz warten.

Zwischenergebnis

11



Was musst du jetzt noch ändern?

12

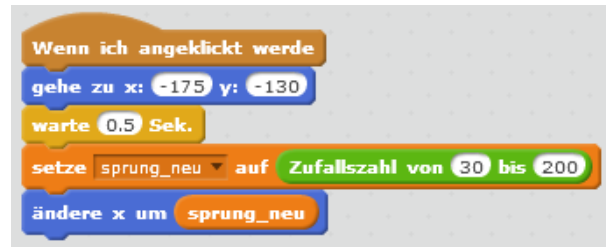
Im Moment überschreibt jede neue Sprungweite den Wert, der in der Variable „sprungweite“ gespeichert ist.

Es soll aber immer nur dann der neue Wert übergeben werden, wenn er größer ist als der aktuelle Wert der Variable.



Führe eine zweite Variable „sprung_neu“ ein, der die aktuelle Sprungweite als Wert übergeben wird.

13



Welche Kontrollstruktur brauchst du für die Abfrage, ob die neue Sprungweite größer ist als die bisher gespeicherte?

14

Eine Verzweigung (falls ... dann ...)



Formuliere die Verzweigung mit eigenen Worten.

15

z.B. „Falls die neue Sprungweite größer ist als die bisher gespeicherte Sprungweite, dann überschreibe den Wert mit der neuen Sprungweite.“

Wie lautet die Bedingung?

16





Welche Anweisung(en) brauchst du?

17

setze sprungweite auf sprung_neu

Ergebnis

18

falls sprung_neu > sprungweite dann
setze sprungweite auf sprung_neu

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) (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).



Mögliche Lösung:

Möglicher Code:



Beim Programmstart werden die beiden Variablen initialisiert und der Frosch auf seine Ausgangsposition gesetzt.

Beim Klick auf den Frosch, springt er auf die Ausgangsposition, wartet eine halbe Sekunde und springt dann eine zufällige Weite nach vorne. Dieser zufällige Wert wird der Variable „sprung_neu“ übergeben. Ist dieser Wert größer als der Wert der Variable „sprungweite“, so wird dieser verändert, sonst nicht.

Man erhält so den maximalen Wert der bisherigen Sprungweiten des Frosches.



Mögliche Erweiterungen:

sprungweite 145 zaehler 8
sprung_neu 130



Code:

```

Wenn [flagge] angeklickt
  setze zaehler auf 0
  setze sprungweite auf 0
  setze sprung_neu auf 0
  gehe zu x: -175 y: -130

Wenn ich angeklickt werde
  gehe zu x: -175 y: -130
  warte 0.5 Sek.
  setze sprung_neu auf Zufallszahl von 30 bis 200
  ändere x um sprung_neu
  falls sprung_neu > sprungweite dann
    setze sprungweite auf sprung_neu
  sonst
    sage Mist! Zu kurz! für 5 Sek.
  ändere zaehler um 1
    
```