

Tutorial zum Umgang mit Scratch

In diesem Tutorial sollst Du die grundlegenden Funktionen von Scratch ausprobieren und kennenlernen. Danach solltest Du gut mit dem Programm umgehen können und eigene Programme in Scratch schreiben können.

Probiere auch während dem Tutorial immer wieder Funktionen aus, die Du interessant findest. Bitte stelle auch während dem Durcharbeiten dieses Tutorials immer Fragen, falls Du etwas nicht verstehen solltest.

1. Bewegung



In diesem Kapitel, soll der Umgang mit Bewegungsfunktionen in Scratch erlernt werden.

Wir bedienen uns dazu der direkt nach dem Programmstart von Scratch dargestellten Katze und wollen versuchen, ihr Bewegung beizubringen.

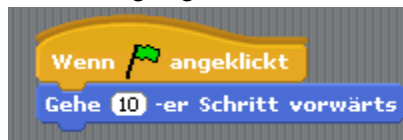


Versuche zunächst, die Katze von links nach rechts bewegen zu lassen.

Ziehe aus dem Bewegungsmenü die Funktion **Gehe 10 -er Schritt vorwärts** in das Scriptfenster. Die Katze sollte sich nun 10-er Schrittwise von links nach rechts bewegen. Um die Bewegung allerdings zu starten benötigst du eine Startbedingung. Diese findest Du im Steuerungsmenü.



Wähle hier eine geeignete Startbedingung und verbinde es durch Ziehen ins Scriptfeld mit der Vorwärtsbewegung.



Wenn Du jetzt auf die Grüne Flagge klickst, wirst du sehen, wie sich die Katze langsam vorwärts bewegt. Diesen Ablauf wollen wir allerdings noch deutlich verbessern.



Probiere auch die anderen „Startbedingungen“ mit der Vorwärtsbewegung zu verknüpfen, und teste, wie sie funktionieren.

Warum bewegt sich die Katze eigentlich von links nach rechts?

Im oberen mittleren Fenster siehst du etwa Folgendes:



Der „x“ und „y“ Wert gibt die Position der Katze in der Bühne an. Wenn Du mit der Maus über die Bühne fährst, siehst du direkt unterhalb der Bühne die x, y Koordinaten der Mausspitze auf der Bühne.

Welche Koordinaten hat die Mitte der Bühne?

Welche Koordinaten, das rechte obere Eck?

Neben dem „x“ und „y“ Wert siehst Du ein Feld „direction“. Das gibt die Richtung an, in der sich die Katze bewegt.

Versuche nun ein Programm zu schreiben, in dem die Katze sich 10 Schritte nach oben bewegt, wenn du auf die grüne Flagge drückst!

Dabei kann die Funktion  sehr hilfreich sein. Achte darauf, wie sich das Feld „direction“ dadurch verändert.



Durch Setzen von Haken bei  **x-Position**,  **y-Position** und  **Richtung** kannst Du Dir laufend Position und Richtung der Katze auf der Bühne anzeigen lassen.

Bis jetzt musst Du, um die Maus sich einmal bewegen zu lassen immer wieder auf die grüne Flagge drücken. Versuche jetzt die Katze ganz nach rechts bewegen zu lassen.

Verwende hierfür das Konzept der Wiederholungsanweisung. Alles was in der „Fortlaufend“ – Klammer eingeschachtelt ist, wird immer wieder ohne Ende ausgeführt.

Eine mögliche Lösung könnte so aussehen:



(Durch drücken der roten Flagge wird die „Fortlaufend“- Schleife abgebrochen.)



(Brett vorm Kopf?)

Was passiert, wenn Du nachdem die Katze ganz rechts angekommen ist, wieder auf die grüne Flagge drückst?

Du kannst die Katze mit der Maus wieder an eine andere Position in der Bühne ziehen, um sie erneut nach Drücken der grünen Flagge in Bewegung zu sehen.

Noch schöner ist es allerdings die Katze automatisch an eine Startposition gehen zu lassen, bevor sie mit der Bewegung beginnt.

Ergänze den Anfang des Programms nun um eine Startposition, die die Katze einnimmt, bevor sie beginnt, sich zu bewegen.

Verwende hierfür z.B.



Wie lässt sich die Geschwindigkeit der Bewegung der Katze beeinflussen?

Programmiere die nun Katze so, dass sie langsam dem Mauszeiger folgt.

Nun soll die Katze, wenn sie auf den Rand der Bühne trifft, in eine andere Richtung weiterlaufen, und so lange durch die Bühne laufen, bis die rote Flagge gedrückt wird.



Versuche z.B. die Funktion zu verwenden.

Mache Dich selbst, bevor Du mit der nächsten Lektion beginnst, mit den restlichen Bewegungsoptionen vertraut.

2. Klang

Nun wollen wir uns ein wenig der Musik zuwenden.

Du kannst mittels dieser Funktionen Klänge erzeugen.

Schreibe zunächst ein einfaches Programm, dass beim Drücken der grünen Flagge „pop“ macht.



Du kannst aber auch viel kompliziertere Lieder spielen lassen.



Wähle durch die Option und lasse es dann eine bestimmte Note



spielen.

Lasse doch mal ein Klavier folgende Noten spielen... **Wie heißt das Lied?**

CDEFGGAAAAGAAAAGFFFFEEDDDDC

Hinterlege das doch mal mit einem Schlagzeug!

3. Aussehen

Die Bewegung unserer Katze sieht leider immer noch nicht ganz gut aus. Das wird jetzt geändert!

Mit „costumes“ kannst Du der Katze ein anderes Aussehen verleihen. Es gibt von unserer Katze noch ein zweites Bild, das sie zeigt, wenn sie gerade einen Schritt ausführt. Wenn wir

die beiden Bilder während der Vorwärtsbewegung abwechseln, können wir den Anschein erzeugen, als ob die Katze geht.

Verwende hierzu das bereits geschriebene Vorwärtsprogramm:



und ergänze es um einen Kostümwechsel. Drücke auf den Registerreiter „Costumes“. Hier siehst du alle momentan verfügbaren „Kostüme“ der Katze. Du kannst sie natürlich jederzeit ergänzen, wenn Dir welche fehlen.

Nach dem Drücken der grünen Flagge siehst nun leider, dass die Kostümwechsel im Vergleich zur Bewegungsgeschwindigkeit viel zu schnell sind. Füge daher noch eine kleine Pause nach dem Kostümwechsel ein und betrachte das Ergebnis.



(Tipp: Du findest die verwendeten Symbole immer unter dem Menü, dass die entsprechende Farbe trägt. Die verwendete (gelbe) Pause findest Du also im Steuerungsmenü).

4. Senden und Empfangen

Noch ein wichtiges Konzept in Scratch ist „Senden und Empfangen“, das wir uns jetzt noch in einem größeren Programm anschauen.

Mit „Senden und Empfangen“ kann man eine Aktion in einer anderen „Figur“ auslösen. Es muss also um z.B. eine Bewegung in einer Figur auszulösen nicht die grüne Flagge gedrückt werden, sondern eine Figur muss einer anderen Figur eine geeignete Botschaft senden, um sie sich z.B. bewegen zu lassen.

Schreibe ein Programm, in dem eine forsche Maus von einer Katze davonläuft, und der Katze immer wieder „Fang mich doch!“ hinterher schreit.



Beginne mit der Katze:

Sie soll nur auf die Maus reagieren und ihr hinterher jagen.

Drücke auf „new“ und beginne ein neues Programm.

Schreibe ein Script, dass die Katze hinter der Maus hinterher jagen lässt, wenn die Maus ruft „Fang mich doch!“.

Benenne auch die Katze ausschlagkräftig um, weil „Sprite1“ wohl wenig aussagekräftig ist:



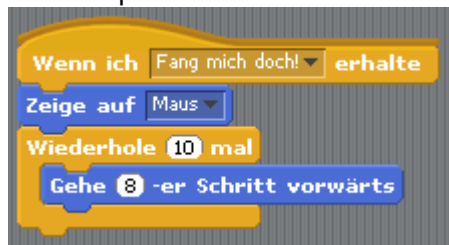
Verwende für das Script (Folge von Befehlen) der Katze aus dem Steuerungsmenü die



Funktion. Erstelle im Drop-Down Menü eine neue Botschaft, auf die die Katze reagiert.

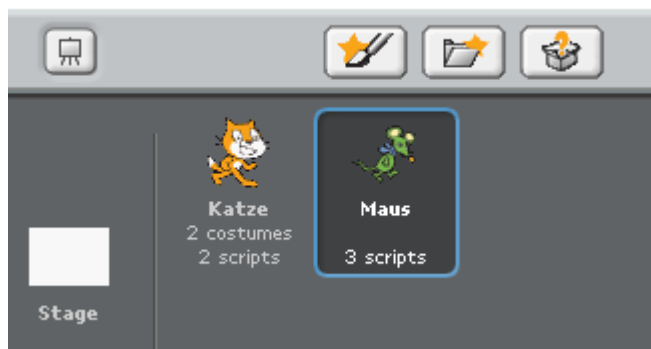
Die eigentliche Bewegung kannst Du so übernehmen, wie sie vorher schon mehrfach verwendet wurde.

Das Script der Katze könnte also so aussehen:



Falls Du auf die grüne Flagge drückst, passiert nun erst einmal gar nichts. Die Katze reagiert nur, wenn ihr die Maus eine Botschaft schickt!

Erstelle nun die Maus:



Drücke auf das Symbol für „Choose new Sprite from File“ und wähle das Bild der Maus aus.

Jetzt hast Du eine zweite Figur zur Verfügung für die Du eigene Scripte schreiben kannst. Indem Du einmal auf die Katze oder die Maus klickst, kannst Du Dir die zugehörigen Scripte der jeweiligen Figur anzeigen lassen oder ändern.



Mit diesen Buttons kannst Du die Maus verkleinern, so dass die Größenverhältnisse zwischen Katze und Maus stimmen.

Schreibe nun zunächst ein Script, dass die Maus auf der Bühne herumlaufen lässt.



Es könnte so aussehen:
(Die Drehung ist eingefügt, damit die Maus nicht immer nur von links nach rechts läuft).

Wenn du das Programm startest, siehst Du wie die Maus läuft, die Katze aber noch nicht reagiert.

Dafür schreiben wir der Maus ein weiteres Script, in dem die Maus die Katze anstachelt.

Verwende hierfür die Funktion  und wähle aus dem Drop-Down Menü die vorher erstellte Botschaft aus. Immer wenn die Maus diese Botschaft sendet, wird die Katze darauf reagieren.

Damit die Botschaft allerdings nicht nur einmal gesendet wird (und die Katze nur einmal hinter der Maus herjagt), solltest Du sie in eine „Fortlaufend- Schleife“ einpacken.

Das zweite Script der Maus könnte also so aussehen:



In einem dritten Script könntest Du nun die Maus noch rufen lassen „Fang mich doch!“.



Wenn Du nun die grüne Flagge drückst, werden alle 3 Scripte der Maus gleichzeitig ausgeführt. Zusätzlich reagiert die Katze auf die lebensmüde Maus und jagt.

5. Fühler & Zahlen

Bisher merkt die Maus noch nichts davon, wenn sie von der Katze gefangen wird. Das wollen wir jetzt ändern! Dazu brauchen wir eine neue Anweisung aus dem Menü "Steuerung":



nämlich die Anweisung.

Diese führt dazu, dass immer dann, wenn die Bedingung, die in dem Kästchen  angegeben ist, erfüllt wird, die nachfolgenden Anweisungen ausgeführt werden.

etwas unsinniges Beispiel:



in diesem Beispiel wird also im Falle, dass $3 = 3$ ist, und das ist ja immer der Fall "Du bist ein Matheass" gesagt. Wie du vielleicht schon bemerkt hast kann also die Bedingung nach dem wenn, die erfüllt werden muss, damit der nachfolgende Block ausgeführt wird, entweder aus dem Menü "Zahlen" (siehe Beispiel) oder aus dem Menü "Fühler" kommen. Damit die Katze merkt, dass sie die Maus gefangen hat, können wir eine Bedingung aus dem Menü "Fühler" verwenden. Versuche selbst herauszufinden, welche Bedingung das ist und programmiere anschließend die Katze so, dass immer wenn sie die Maus fängt sie "Hab dich!" für eine bestimmte Zeit sagt.

Wenn du dir im Umgang mit diesen Bedingungsanweisungen (Wenn, Fortlaufend wenn, ...) noch unsicher bist, findest du im Anhang (Seite 10) noch den Vergleich der Scratch – Befehle mit den entsprechenden Karol – Befehlen, die dir ja eventuell schon besser bekannt sind.

Im folgenden wollen wir jetzt die Maus nicht mehr selbständig über den Bildschirm laufen lassen, sondern wir wollen sie mit der Tastatur steuern.



Dafür können wir aus dem Menü "Fühler" die Bedingung

verwenden.

Versuche also nun die Maus so zu programmieren, dass immer dann wenn die Taste "oben" / "unten" / "links" / "rechts" gedrückt wird, die Maus ihre Bewegung genau in diese Richtung ändert. Es sollen also jetzt nur noch Bewegungen vertikal und horizontal zum Bildschirm möglich sein.

Um unser kleines Katz und Mausspiel noch etwas interessanter zu gestalten wollen wir der Maus jetzt noch ein Ziel geben: da die Maus am heutigen Tag noch nichts zu Mittag gegessen hat und ihr Lieblingssessen natürlich Käse ist, ist es nun deine Aufgabe ein neues "Käse-Sprite" zu entwerfen und auf dem Bildschirm zu platzieren. Wie ein solches Sprite entworfen wird, hast du ja bereits im Expertenpuzzle erfahren. Falls du dir dennoch unsicher

bist, frage noch mal bei den Experten zu diesem Thema nach.

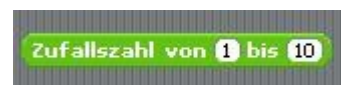
6. Variablen

Um ein richtiges Spiel daraus zu machen, wollen wir jetzt der Maus, immer wenn sie den Käse erreicht, Punkte geben. Dazu muss der aktuelle Punktestand in irgendeiner Form gespeichert werden. Um solche zusätzlichen Informationen zu speichern, eignet sich der Menüpunkt "Variablen":



Erzeuge also eine neue Variable mit dem Namen Punktestand, der zunächst nur für die Maus gelten soll (For this sprite only). Programmiere die Maus jetzt so, dass immer, wenn der Käse berührt wird, der Punktestand um eins hochgezählt wird und die Maus wieder auf eine Anfangsposition zurückgesetzt wird (wie das mit dem Hochzählen funktioniert solltest du herausfinden können, wenn du den Menüpunkt Variablen mal genauer ansiehst). Achte darauf, dass der Punktestand, beim Neustart des Programms wieder auf Null gesetzt wird. Damit die Katze auch ihr Erfolgserlebnis bekommt, soll jetzt immer, wenn sie die Maus gefangen hat, der Punktestand um Eins erniedrigt werden, und die Maus erneut auf ihre Anfangsposition gesetzt werden. Eventuell musst du dafür die Variable neu erstellen, und auf die Einstellung "For all sprites" setzen, damit auch die Katze direkten Zugriff darauf hat (vielleicht findest du aber auch eine andere Lösung).

Damit jetzt das Spiel nicht immer gleich abläuft, die Katze, die Maus und der Käse also nicht immer an derselben Position starten, sollst du jetzt noch das Konzept der Zufallszahlen kennenlernen.



Unter dem Menüpunkt "Zahlen" findest du einen Eintrag

Damit kannst du den Computer eine Zufallszahl in dem von dir angegebenen Bereich erzeugen lassen.

Die Einstellung "Zufallszahl von 1 bis 10" erzeugt also eine Zahl, die zwischen 1 und 10 liegt. Den Bereich, in dem diese Zahl liegen soll kannst du natürlich durch Angabe von neuen Zahlen in den Kästchen ändern.

Programmiere das Spiel nun so, dass Katze, Maus und Käse am Anfang des Spiels auf eine zufällige Position gesetzt werden. Ebenso sollen alle Figuren auf eine zufällige Position gesetzt werden, wenn die Maus den Käse erreicht, oder die Katze die Maus fängt. (Kleiner Tipp: du wirst dafür wahrscheinlich wieder Nachrichten senden und empfangen lassen müssen)

Was uns jetzt noch fehlt ist ein Ende des Spiels. Um das gesamte Programm zu stoppen gibt

es unter dem Menüpunkt "Steuerung" einen Eintrag



Erweitere das Programm nun so, dass wenn die Maus 10 Punkte bzw. -10 Punkte erreicht hat, das Spiel beendet wird. Du kannst natürlich die Maus der Katze bzw. die Katze der Maus am Ende des Spiels noch etwas sagen lassen, je nachdem wer gewonnen hat.

An sich sind wir jetzt am Ende unserer Einführung in Scratch und du solltest dich jetzt mit den wichtigsten Funktionen auskennen. Wenn du noch Zeit hast kannst du jetzt noch selbst Ideen in das Spiel einbringen, oder du bearbeitest noch den Punkt "Spielerweiterungsidee für besonders Schnelle".

7. Spielerweiterungsidee für besonders Schnelle oder zum Ausprobieren für zuhause

Um das ganze Spiel noch auszubauen, wollen wir jetzt einen richtigen Level entwerfen und zwar soll dieser Level die Form eines rechteckigen Labyrinths haben. Zeichne also für die Stage einen eigenen Background, der ein Labyrinth darstellt. Wie du eigene Backgrounds zeichnest, hast du ja bereits beim Expertenpuzzle gelernt, bei Fragen kannst du dich also wieder an die Experten wenden.

Im folgenden soll nun der Käse an einem bestimmten Ort, die Maus an einem bestimmten Ort und die Katze an einem bestimmten Ort im Labyrinth platziert werden. Ziel ist nun erneut, dass die Maus versucht, an den Käse zu kommen, ohne vorher von der Katze erwischt zu werden.

Wichtig ist jetzt, dass die Katze und die Maus, wenn sie gegen eine Wand des Labyrinths stoßen, natürlich nicht weiterlaufen können sollen, sondern genau in die Richtung aus der sie kamen zurücklaufen sollen.



Dafür kann für dich die Bedingung

von Nutzen sein (Immer wenn

also die Farbe der Labyrinthwand berührt wird, soll die Katze/Maus in die Richtung zurücklaufen, aus der sie ursprünglich kam.)

Außerdem macht es jetzt keinen Sinn mehr, die Katze immer nur in Richtung der Maus laufen zu lassen, da sie sonst ständig gegen irgendwelche Mauern stößt. Vielleicht hast du eine gute Idee, wie du eine "intelligente" Katze programmieren kannst, die trotz der vielen Hindernisse es noch schafft die vom Menschen gesteuerte Maus effektiv zu verfolgen. Beim Erreichen des Käses könnte dann zum Beispiel ein neuer Level mit einem neuen Labyrinth beginnen. Aber wie du diese Spielidee jetzt genau umsetzt sei ab dieser Stelle dir überlassen... Viel Spaß beim Programmieren!

Anhang

Vergleich der Bedingungsanweisungen Scratch – Karol

Scratch:



Karol:



```
wenn <bedingung> dann  
  <anweisung>  
  ...  
  <anweisung>  
*wenn
```



```
wenn <bedingung> dann  
  <anweisung>  
  ...  
  <anweisung>  
sonst  
  <anweisung>  
  ...  
  <anweisung>  
*wenn
```



ist eigentlich zusammengesetzt aus:



```
wiederhole immer  
  wenn <bedingung> dann  
    <anweisung>  
  ...  
  <anweisung>  
*wenn  
*wiederhole
```