

Die ersten Schritte mit Scratch

Dieses Tutorial hilft dir bei den ersten Schritten mit Scratch. Nachdem du die einzelnen Übungen durchgearbeitet hast, kannst du selber kreativ werden.

Aber was ist Scratch?

Mit der Programmiersprache Scratch kann man eigene Computerprogramme schreiben. Hierfür werden farbige Programmier-Bausteine verwendet, die man wie Puzzleteile zusammensetzt. Mehrere zusammengesetzte Bausteine nennt man Skript. Sie werden vom Computer Schritt für Schritt abgearbeitet.

Aufgabe

Arbeite das Tutorial von Scratch sorgfältig durch. Dabei wirst du viele Möglichkeiten von Scratch entdecken und dich mit der Entwicklungsumgebung vertraut machen. Zum Schluss wirst du sogar dein erstes Spiel mit Scratch programmiert haben.

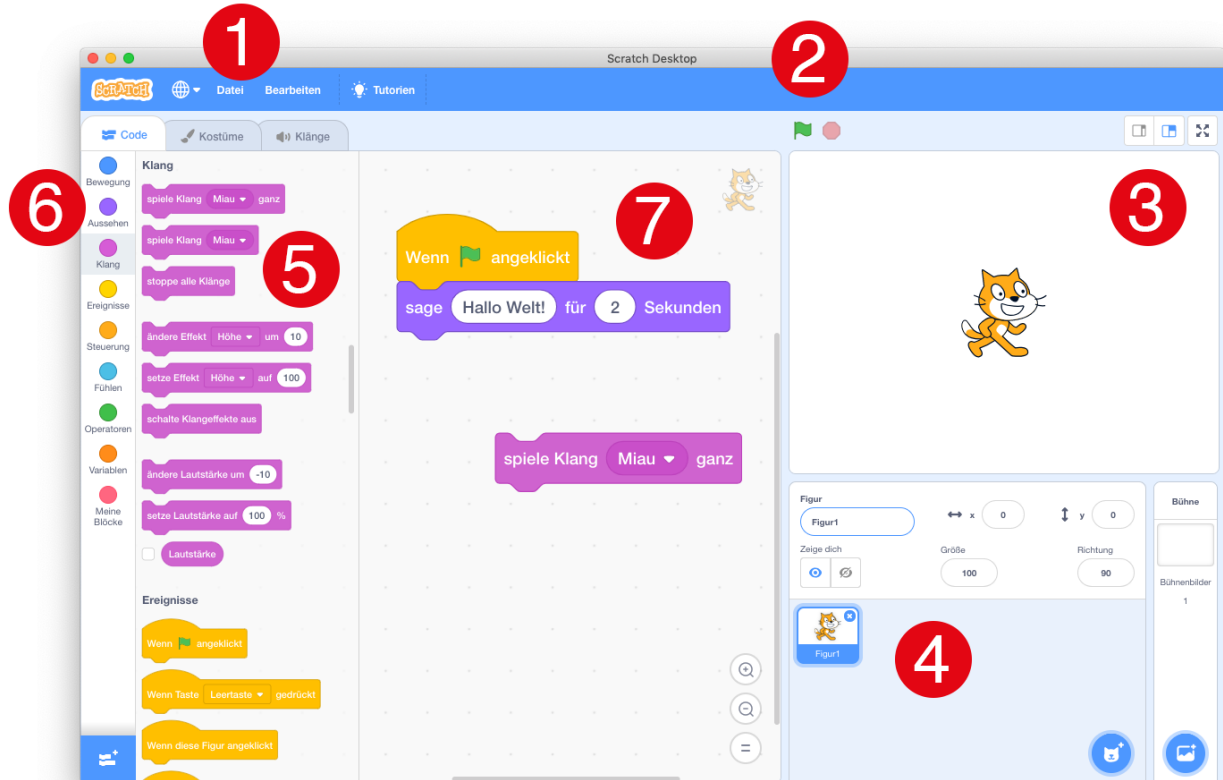
Wichtiger Hinweis

Achte darauf, dass du deinen Fortschritt ausreichend abspeicherst. Lasse deiner Kreativität genügend Platz: Es darf auch einmal eine andere Figur als die vorgeschlagene ausgewählt werden.



Scratch-Tutorial I

Entwicklungsumgebung



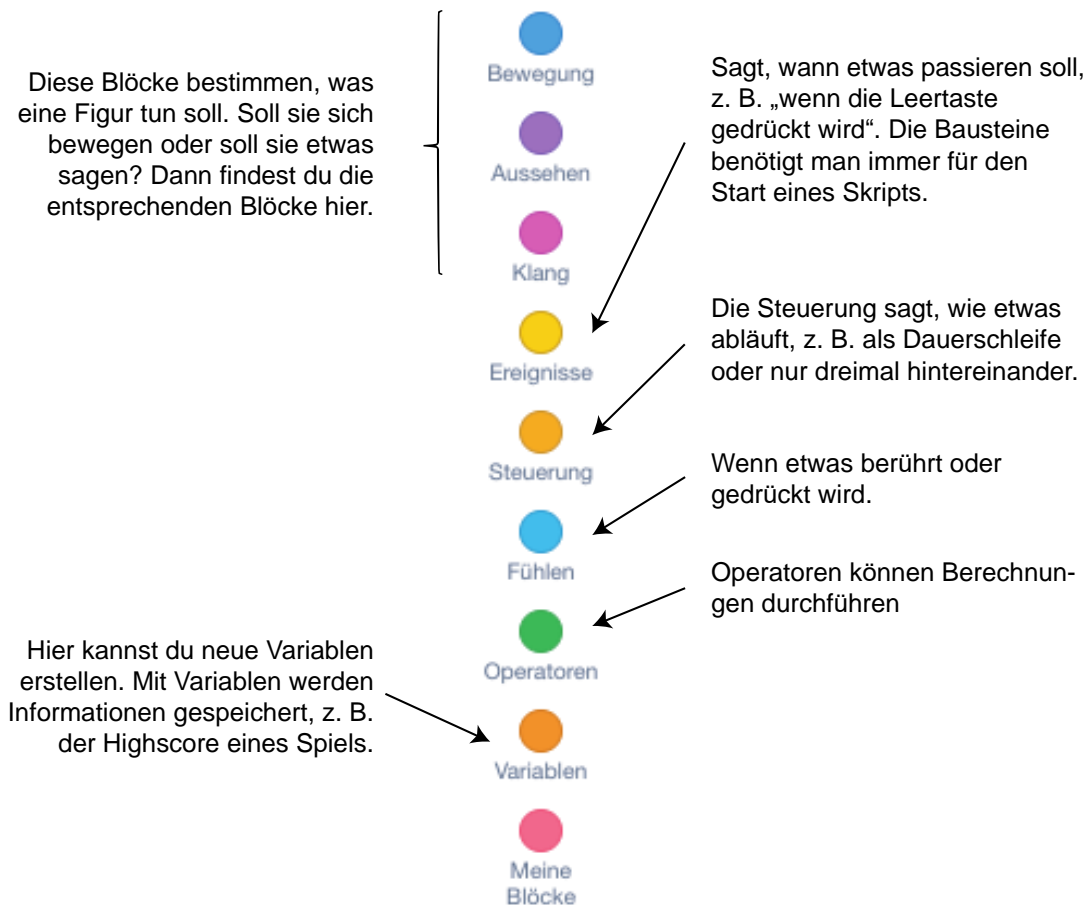
Die folgenden Erläuterungen beziehen sich auf die mit den Zahlen gekennzeichneten Bereiche:

1. In diesem Bereich findest du verschiedene Reiter wie „Datei“ oder „Bearbeiten“, das sind sogenannte Menüs. Man kann sie durch Anklicken „ausklappen“ (engl.: drop down = herunterfallen). Dadurch findest du zum Beispiel Möglichkeiten zum Speichern und Laden von Programmen.
2. Mit der grünen Fahne werden Programme in Scratch gestartet und mit dem roten Sechseck gestoppt.
3. Dieser Bereich stellt die Bühne (Stage) dar. Hier werden Figuren (Objekte oder auch Sprites genannt) positioniert und führen Aktionen aus. Alles was die Nutzer sehen und durch die Programme von Scratch ausgeführt wird, ist hier dargestellt.
4. Figuren müssen nicht immer auf der Bühne sichtbar sein (sie können sich auch verstecken). Damit man weiß, dass diese im Programm enthalten sind, werden alle Figuren in diesem Bereich aufgelistet. Links davon wird die Bühne in Klein angezeigt.
5. Anweisungen beziehungsweise Befehle, die der Computer ausführen soll, haben in Scratch die Form von Bausteinen. In diesem Bereich sind die Befehle/Bausteine aufgelistet.

6. Um nicht immer eine RIESIGE Liste an Bausteinen angezeigt zu bekommen, sind die Bausteine in Scratch in Kategorien zusammengefasst. Sie sind hier zu finden. Wenn man eine andere Kategorie auswählt, dann erscheinen andere Bausteine im Bereich 5.
7. Um mit Scratch zu programmieren, werden einzelne Bausteine in das Skriptfenster gezogen. Werden mehrere Bausteine miteinander verbunden, spricht man von einem Skript. In diesem Fenster werden immer nur die Skripte der ausgewählten Figur oder der Bühne angezeigt. Mehrere Figuren können also verschiedene Skripte haben. Du kannst dir die Skripte ähnlich vorstellen wie kleine Unterprogramme. Als Programm wird alles bezeichnet, was du verwendet hast, also die Bühne sowie alle Figuren und Skripte.

Wie funktioniert Scratch

Scratch besteht aus einer Vielzahl an farbigen Bausteinen, die zu Kategorien zusammengefasst wurden. Schau dir mal den Bereich 6 genauer an. Mit einigen Bausteinen kann man Bewegung und Aussehen ändern oder zeichnen.



Das erste Programm: Bewegungen

Zu Beginn wirst du den Umgang mit den Bewegungsfunktionen kennenlernen. Verwende dazu die beim Programmstart dargestellte Katze Scratchy und versuche, ihr einige Bewegungen beizubringen.

Aufgaben

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

1. Ziehe aus der „Bewegung“-Kategorie folgenden Baustein in das Skriptfenster:



2. Klicke mit der Maus auf den gerade hinzugefügten Baustein im Skriptfenster. Die Katze Scratchy sollte sich in 10er-Schritten von links nach rechts bewegen. Die Zahl gibt an, um wie viele Pixel sich die Katze vorwärtsbewegt. Je größer die Zahl, desto schneller ist die Geschwindigkeit.
3. Damit sich die Katze bewegt, sobald die grüne Fahne geklickt wird, brauchst du eine bestimmte Startbedingung. Suche den folgenden Baustein in der „Ereignisse“-Kategorie und verbinde ihn durch Ziehen ins Skriptfeld mit dem anderen Baustein.



4. Wenn du alle Schritte richtig befolgt hast, dann sollte dein Skript so aussehen:



Wenn du jetzt auf die grüne Flagge klickst, wirst du sehen, wie sich die Katze einmal vorwärts bewegt.

Zusatzaufgabe

Probiere auch die rechts abgebildeten Startbedingungen mit der Vorwärtsbewegung zu verknüpfen, und teste, wie sie funktionieren.



Vielleicht fragst du dich ja: Warum bewegt sich die Katze eigentlich von links nach rechts? Unter der Bühne siehst du Folgendes:



Der „x“- und „y“-Wert gibt die Position auf der Bühne an. Diese Werte nennt man auch Koordinaten. Wenn du mit dem Mauszeiger über die Bühne fährst, siehst du direkt unterhalb der Bühne die x- und y-Koordinaten des Mauszeigers auf der Bühne.

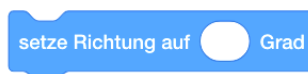
Im Bereich der „Bewegen“-Kategorie ist dir sicherlich schon Folgendes aufgefallen:



Wenn du dort die Haken setzt, dann kannst du dir die Position und Richtung der Katze auf der Bühne anzeigen lassen.

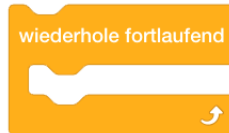
Aufgabe

1. Finde heraus, welche Koordinaten die Mitte der Bühne hat. Kannst du auch sagen, welche Koordinaten die rechte obere Ecke hat?
2. Versuche, das Skript so zu verändern, dass sich die Katze nach oben bewegt, wenn du auf die grüne Flagge drückst. Dieser Baustein sollte dir dabei helfen:



Schleifen

Bis jetzt musstest du, damit die Katze sich einmal bewegt, immer wieder auf die grüne Flagge drücken. Nun soll die Katze aber mit einmal Drücken immer weiter gehen, bis sie ganz am rechten Rand angekommen ist. Dafür benötigt man eine Wiederholungsanweisung, diese nennt man Schleife. Ein Beispiel dafür ist der „wiederhole fortlaufend“-Baustein:



Alles was sich innerhalb dieses Bausteines befindet, wird immer wieder ohne ein definiertes Ende ausgeführt. Damit die Katze nun also mit einmal Drücken der grünen Flagge ganz nach rechts läuft, muss das Skript so aussehen:



Übrigens kannst du durch Drücken des roten Sechsecks die Schleife abbrechen.

Aufgabe

1. Baue die „wiederhole fortlaufend“-Schleife in dein Programm ein. Ändere dazu das bisherige Skript und probiere aus, was passiert, nachdem die Katze ganz rechts angekommen ist.

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

2. Praktisch ist es, wenn die Katze automatisch an eine Startposition geht, bevor sie mit der Bewegung beginnt. Ergänze dafür dein Skript entsprechend. Verwende dazu folgenden Baustein:



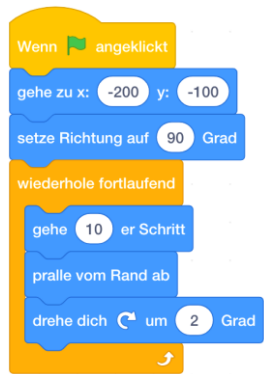
3. Nun soll die Katze, wenn sie auf den Rand der Bühne trifft, in eine andere Richtung weiterlaufen. Am besten so lange, bis das rote Sechseck gedrückt wird. Ändere dein Skript entsprechend. Der folgende Baustein könnte dir behilflich sein:



4. Damit die Katze nicht nur hin und her läuft, kannst du sie sich drehen lassen:



Alles zusammen, kann dann so aussehen:

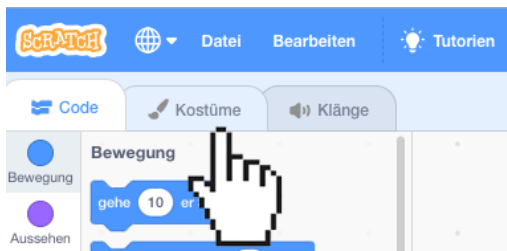


Zusatzaufgabe

Probiere aus, den einen oder anderen Baustein wegzulassen, und beobachte, was passiert. Welche Bausteine sind unbedingt notwendig und welche kann man auch weglassen?

Aussehen/Kostüme

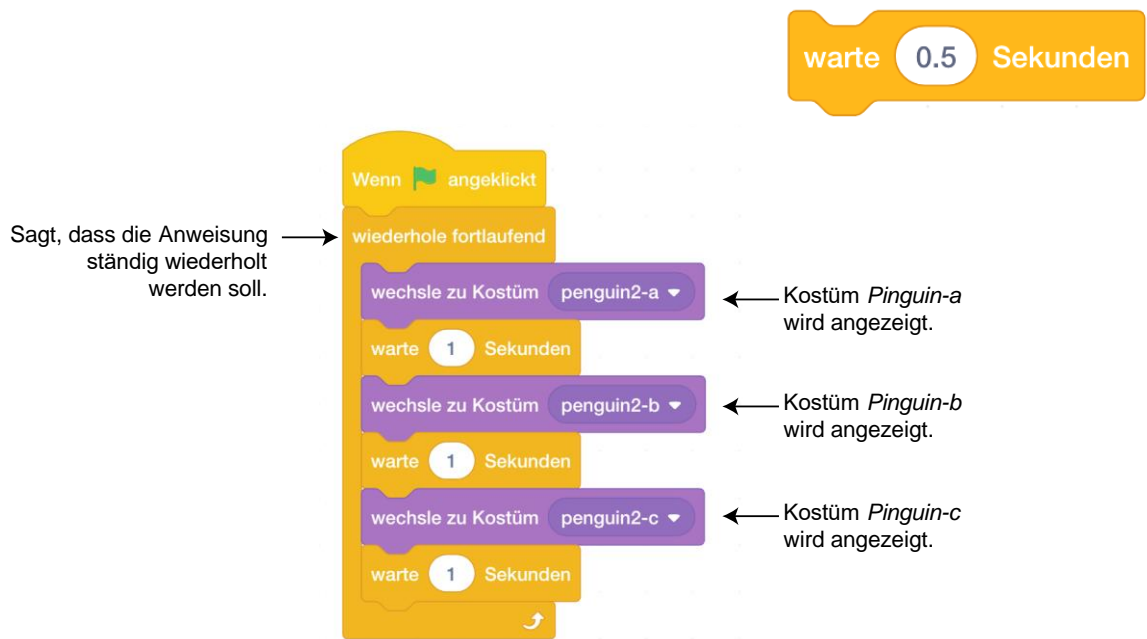
Versuch nun etwas Neues. Speichere dazu dein Programm ab und klicke im Menü auf „Datei“ und dann auf „Neu“ um ein neues Programm zu erstellen. Anschließend kannst du die Katze mit Rechtsklick auf die Figur



von der Bühne löschen. Danach musst du eine neue Figur hinzufügen, wähle hierfür den Pinguin aus. Wenn du nun auf den Karteireiter „Kostüme“ klickst, dann siehst du, dass es den Pinguin in mehreren Varianten gibt.



Man kann programmieren, dass der Pinguin sein „Kostüm“ wechselt. So bekommt man ziemlich schnell Bewegung rein. Baue Folgendes nach:

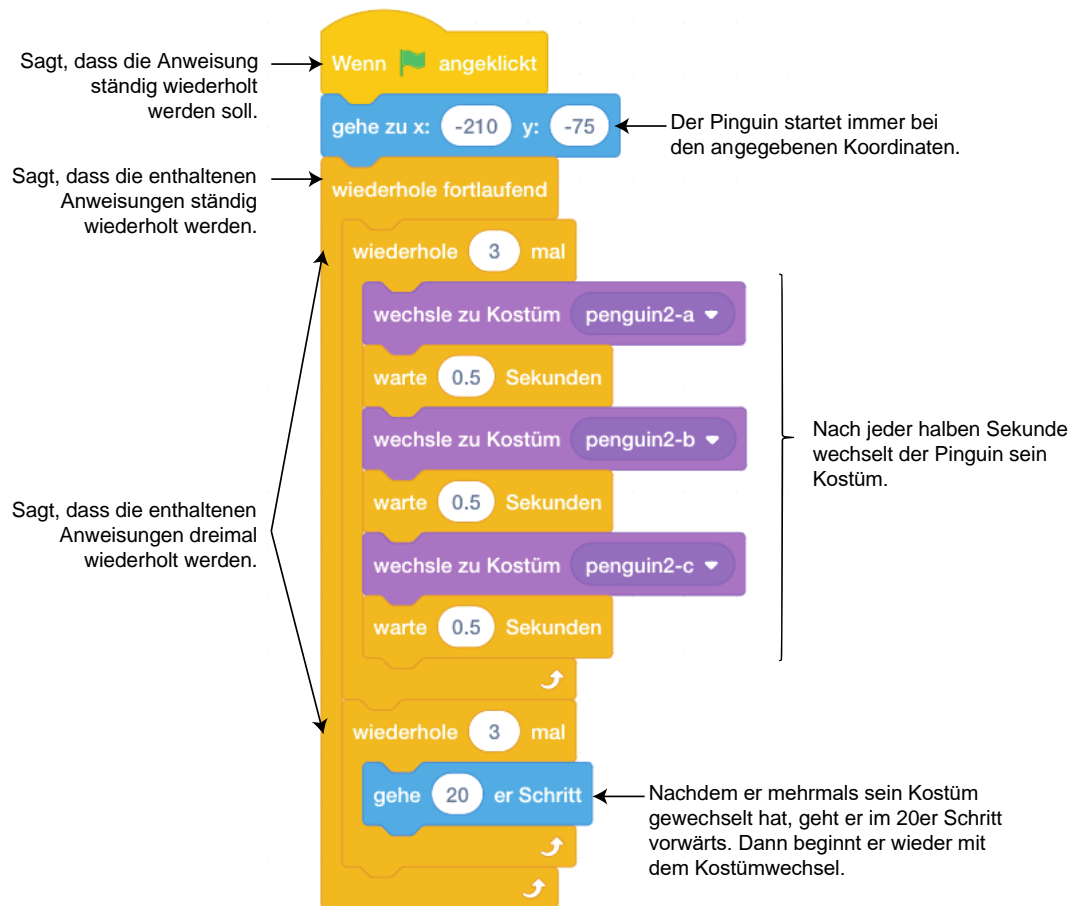


Aufgabe

Was passiert, wenn du keinen „Warte-Block“ einfügst oder du die Zeit im „Warte-Block“ veränderst?

Tipp: Wenn der Pinguin sein Kostüm schneller wechseln soll, muss das Warten kürzer werden. Verwendet wird hier aber die englische Schreibweise, bei der man einen Dezimalpunkt verwendet und kein Dezimal**k**omma.

Du kannst nun auch programmieren, dass sich der Pinguin nicht nur auf der Stelle bewegt, sondern auch von links nach rechts:



Hex-Hex – vom Verschwinden und wieder Auftauchen

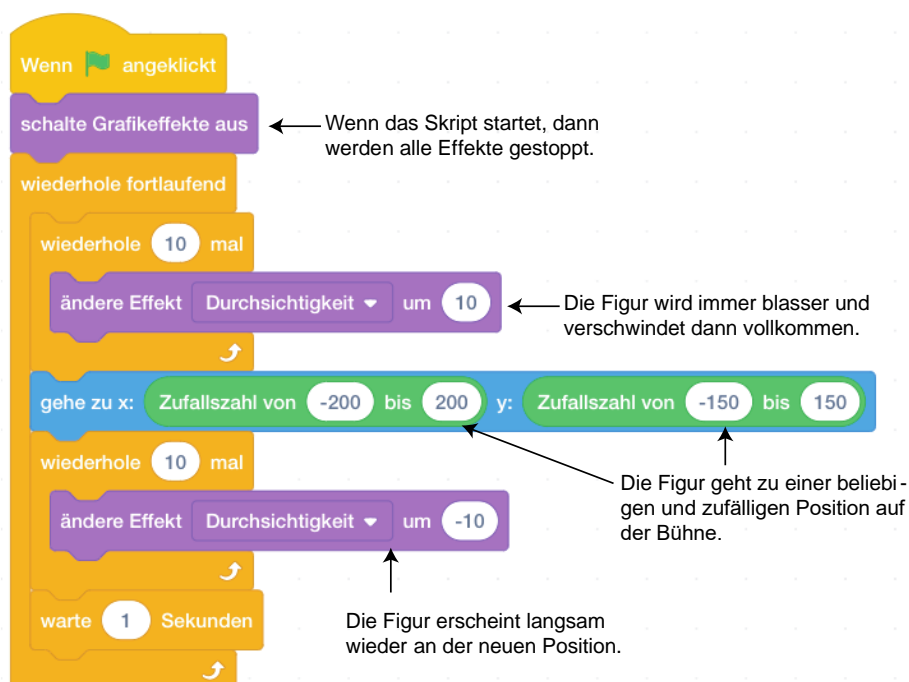
Es gibt viele verschiedene Möglichkeiten, Figuren unsichtbar, größer oder kleiner zu machen oder sogar die Farbe zu wechseln.

Suche dir wieder eine Figur aus, vielleicht eine kleine Hexe oder einen Geist. Die nötigen Bausteine findest du in der Kategorie „Aussehen“. Hinter den Bausteinen verstecken sich viele interessante Möglichkeiten.



Die Programmierung beginnt mit dem Baustein „Schalte Grafikeffekte aus“. Dies macht man, damit die Figur erst einmal ohne einen Effekt startet. Andernfalls kann es passieren, dass man vielleicht schon irgendwo einen Effekt programmiert hat und die Figur zu Beginn unsichtbar ist.

Versuche einmal Folgendes:



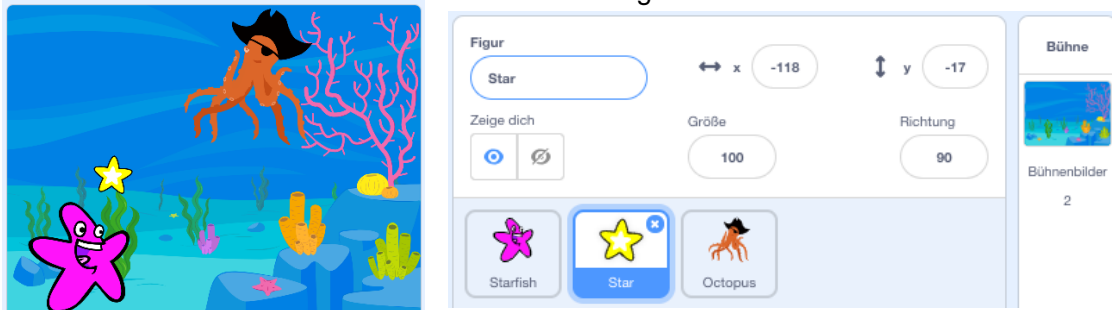
Aufgabe

1. Verändere die Zahlen: zum einem im „Wiederholen-Baustein“ und zum anderem beim „Durchsichtigkeit-Effekt“. Welche Auswirkungen hat das?
2. Probiere doch nun einmal die anderen Effekte aus: Farbe, Wirbel, Pixel und so weiter. Welcher gefällt dir am besten?

Tipp: Vielleicht ist dir aufgefallen, dass die grünen Bausteine nicht wie die anderen aneinandergereiht werden. Es sind sogenannte Operatoren, sie werden immer in andere Bausteine eingesetzt. Das gilt auch für einige andere Bausteine aus dem Bereich „Fühlen“.

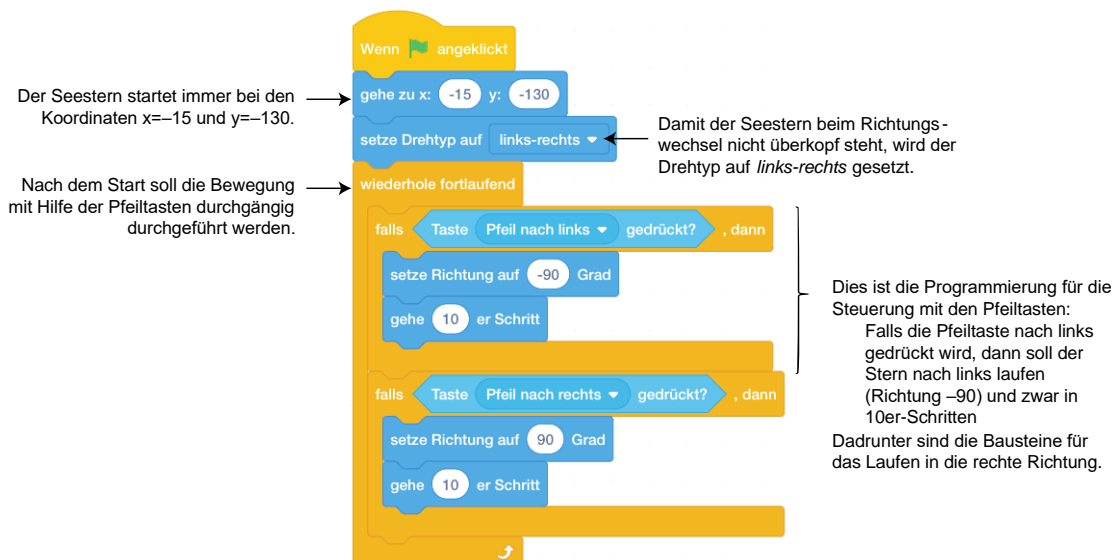
Scratch-Tutorial II

In diesem zweiten Teil wirst du ein eigenes Computerspiel programmieren. Dafür brauchst du drei Darsteller: einen, der etwas wirft, einen, der abgeworfen wird, und ein Objekt, das geworfen wird. Als Beispiel wurden in diesem Tutorial ein großer Seestern, ein kleiner Seestern und ein Tintenfisch ausgewählt.



Wähle für dein Computerspiel drei Figuren und einen passenden Hintergrund aus. Du kannst unsere Vorschläge übernehmen oder dir etwas Eigenes aussuchen.

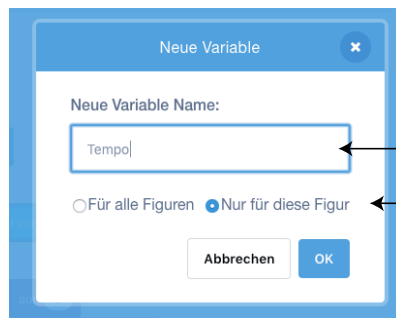
1. Zu Beginn wählst du die Figur aus, die werfen soll. In diesem Beispiel ist das der pinkfarbene Seestern. Der Seestern soll am Anfang immer an einem bestimmten Startpunkt im Koordinatensystem stehen und er soll mit den Pfeiltasten bedient werden können. Die Programmierung dazu sieht folgendermaßen aus:



Die Programmierung für unseren ersten Darsteller ist nun fertig. Wenn du auf das grüne Fähnchen klickst, musst du ihn mit Hilfe der Pfeiltasten nach links oder rechts laufen lassen können.

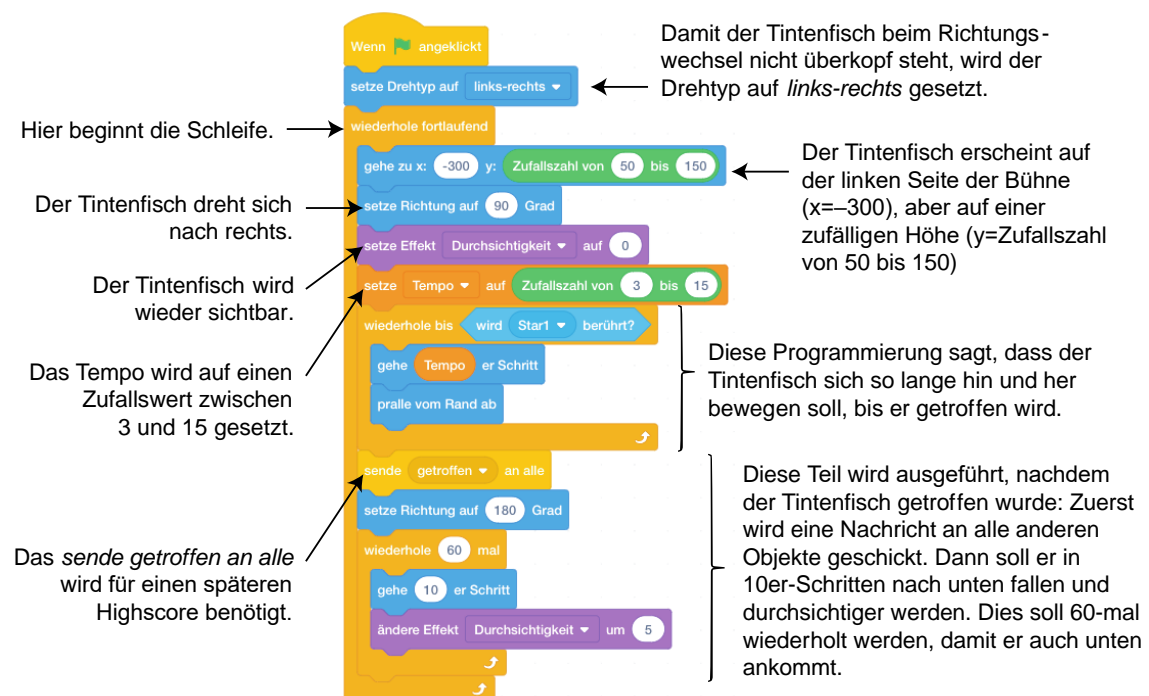
2. Wähle jetzt die Figur aus, die abgeworfen werden soll, in unserem Fall ist das der Tintenfisch. Dir fällt sicher auf, dass das Skriptfenster für die neue Figur leer ist, dies ist so, weil dort immer nur die Skripte der ausgewählten Figur angezeigt werden. Der Tintenfisch soll sich immer hin und her bewegen und das in unterschiedlichen Geschwindigkeiten. Das soll er so lange machen, bis er vom Wurfgeschoss getroffen wird – dann soll er herunterfallen. Bevor du damit starten kannst, musst du eine neue Variable erstellen.

Variablen sind Platzhalter, in denen Informationen und Werte gespeichert werden. Meistens verwendet man sie, um Punkte zu zählen oder um eine Stoppuhr laufen zu lassen. Um eine Variable zu erstellen, klickst du auf „Variablen“ und dann auf „Neue Variable“. Daraufhin sollte folgendes Fenster erscheinen:



Name der Variable:
Nenne sie Tempo oder Geschwindigkeit.
Hier kann man bestimmen, ob die Variable auf alle Figuren angewendet wird oder nur für diese Figur. In unserem Fall wird die Variable nur von dem Tintenfisch verwendet.

Weiter geht es mit der Programmierung:



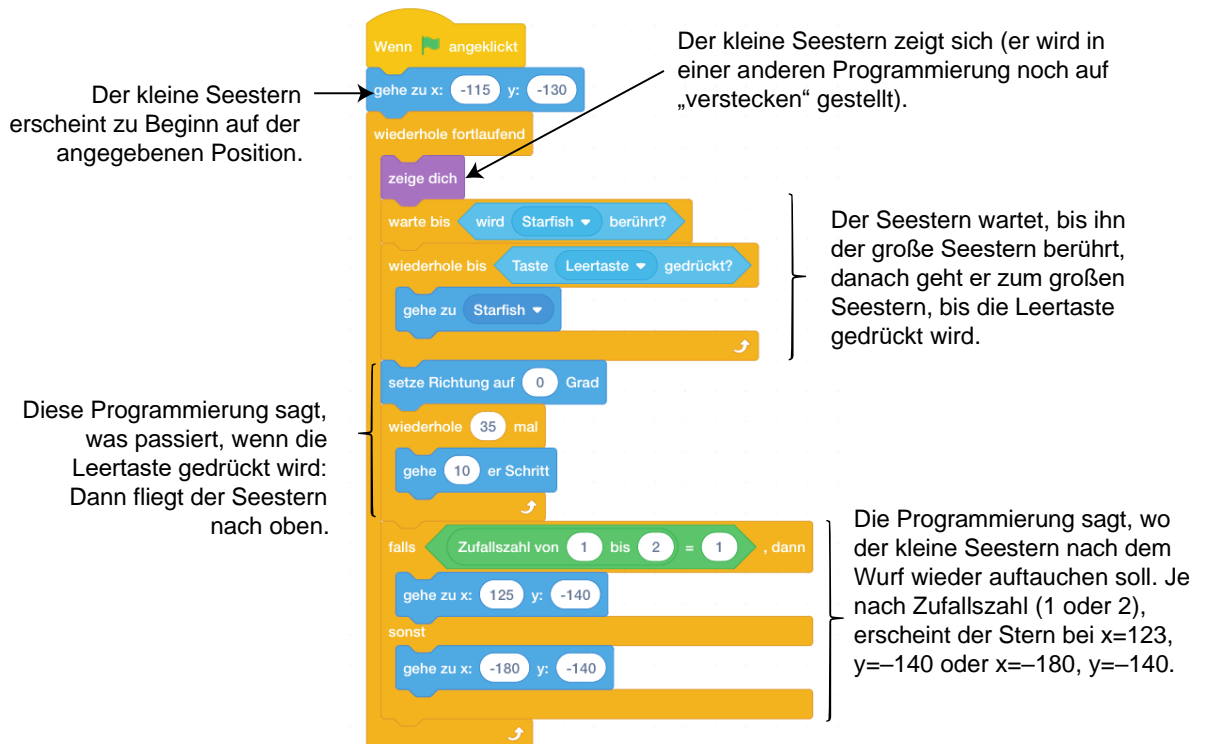
Übrigens kannst du beim „senden an alle“-Baustein auch den Text der Nachricht beispielsweise auf „getroffen“ ändern. Dazu musst du einfach im Dropdown-Menü „neue Nachricht...“ auswählen. Es sollte sich dann folgendes Fenster öffnen:



Wenn du dein Spiel jetzt testest, dann kannst du deinen Werfer mit den Pfeiltasten hin und her bewegen und deine Figur, die abgeworfen wird, bewegt sich selbstständig in unterschiedlicher Geschwindigkeit hin und her.

Du solltest bei der Programmierung darauf achten, dass die Skripte für die verschiedenen Figuren auch im richtigen Skriptfenster sind. Sonst kann es zu Fehlern im Programm kommen.

3. Nun benötigst du noch etwas zum Werfen. In unserem Beispiel ist das der kleine Seestern. Die Programmierung für ihn sieht wie folgt aus:

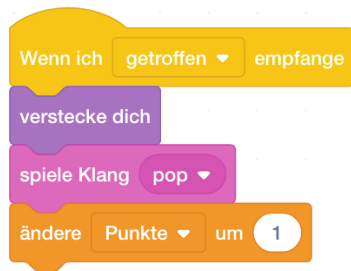


4. Deine Figur sollte nun eine andere Figur aufnehmen und eine dritte Figur damit bewerten können. Damit das Ganze noch interessanter wird, kannst du noch Punkte zählen. Das geht so:

Zuerst erstellst du eine neue Variable für alle Figuren, diese nennst du „**Punkte**“.

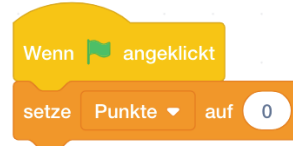
Zur Erinnerung: Dem Tintenfisch hast du gesagt, er soll „getroffen“ senden, wenn er vom kleinen Seestern berührt wird. Das kannst du nun nutzen und dem kleinen Seestern Folgendes hinzufügen:





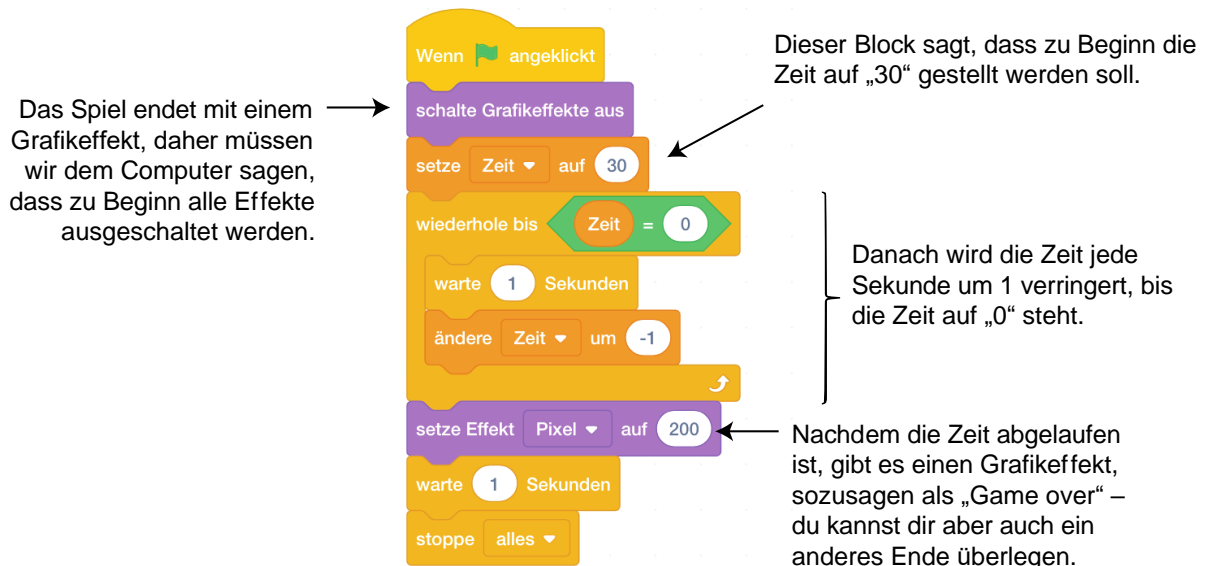
Wenn du „getroffen“ empfängst, dann verstecke dich, spiele einen Klang und ändere die Punkte um 1.
Da der Seestern sich hier versteckt, mussten wir in der vorherigen Programmierung das „zeige dich“

Damit der Punktestand bei mehrmaligem Spielen immer bei „0“ beginnt, müssen wir das dem Computer auch noch sagen.

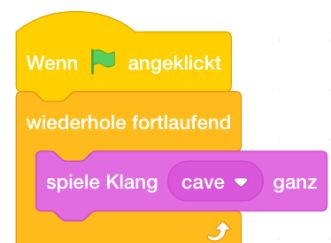


5. Zum Schluss legst du fest, dass das Computerspiel nach einer bestimmten Zeit zu Ende ist. Dafür wählst du diesmal keine Figur aus, sondern den Hintergrund/die Bühne. Darüber hinaus brauchst du wieder eine neue Variable, diesmal für die Zeit. Klicke dafür wie gewohnt auf „Daten“ und erstelle eine neue Variable.

Danach erstellst du folgende Programmierung:



Wenn du willst, kannst du nun deinem Computerspiel noch eine Hintergrundmusik geben. Das programmierst du auch auf dem Hintergrund.



Und nun bist du an der Reihe. Entwickle doch mal ein eigenes Spiel oder produziere einen Trickfilm!