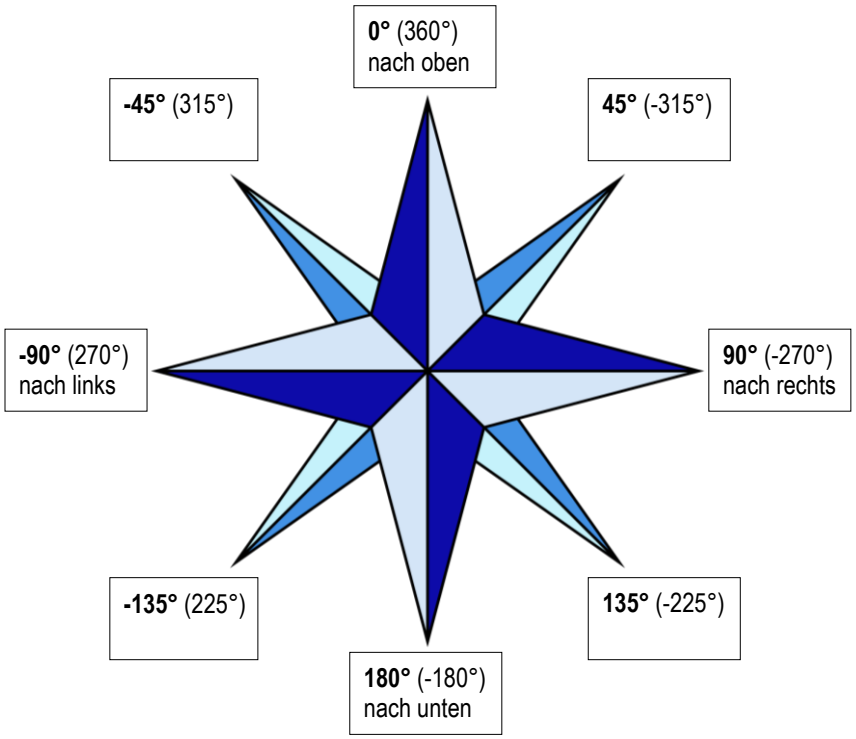
SCRATCH

Info Cards

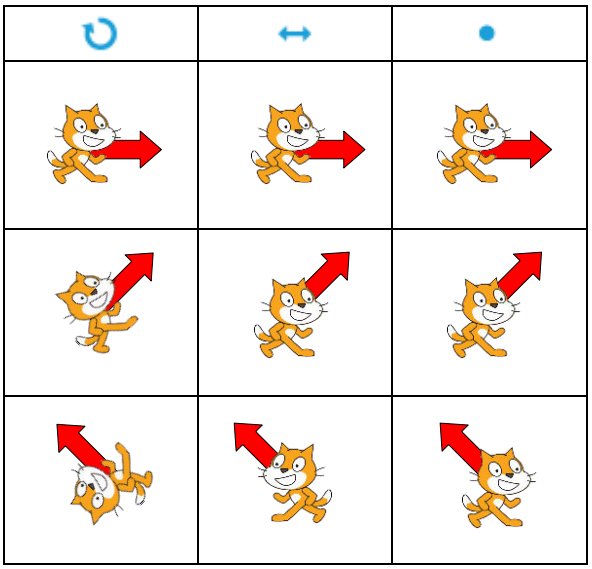
imagine / create / share

Richtung
Jede Figur kann sich um sich selber drehen. Die Drehrichtung wird in Grad (°) angegeben. Voreingestellt ist die Richtung 90° (nach rechts).



Hier kannst du üben:
https://scratch.mit.edu/projects/160265753/#player

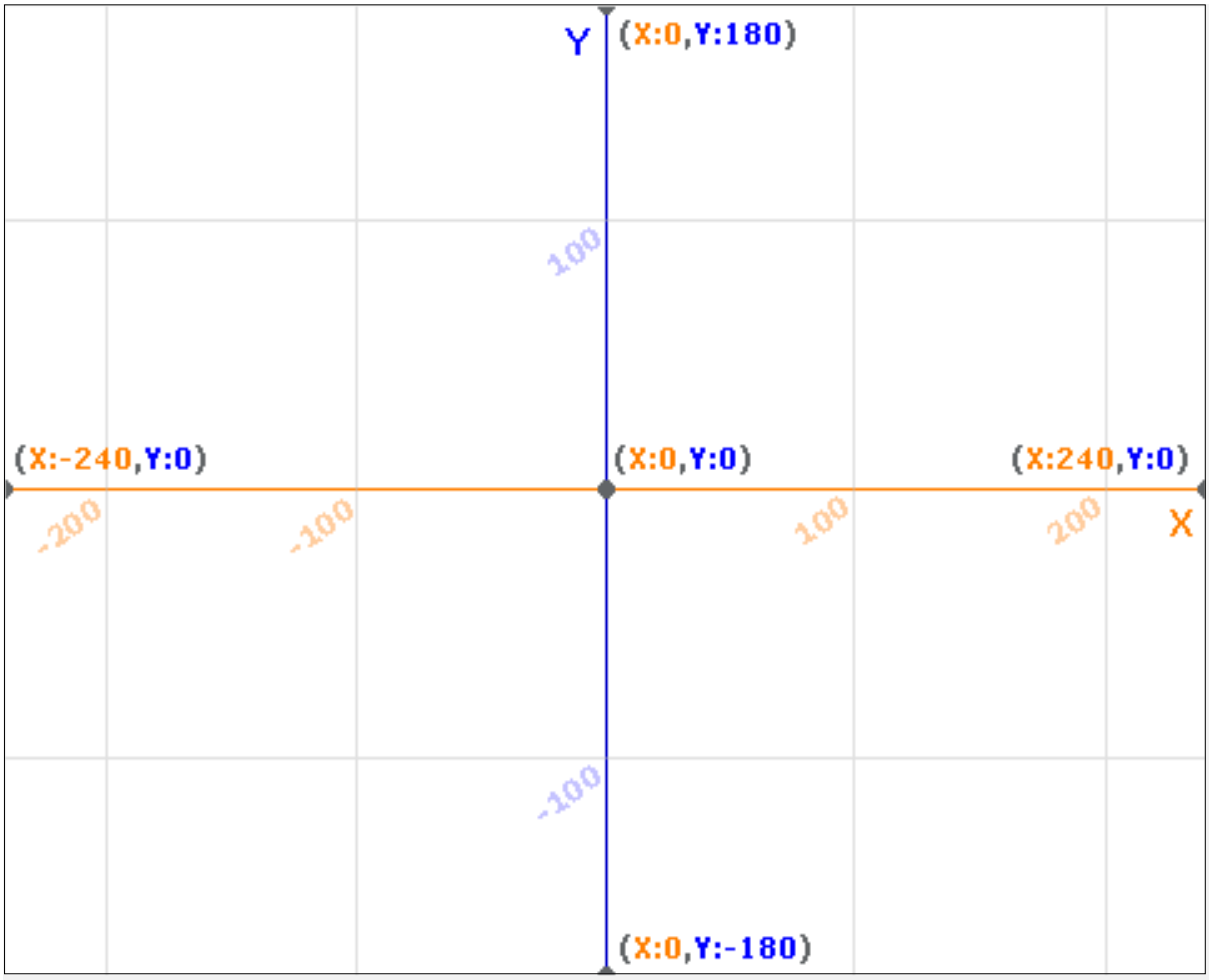
Drehmodus
Möchtest du, dass sich die Katze zwar in die angegebene Richtung bewegt, sich aber selber nicht dreht, kannst du dies im Informationsfenster der Figur beim Drehmodus angeben.



- Die Katze dreht sich immer in die Bewegungsrichtung, steht deshalb unter Umständen auf dem Kopf.
- Die Katze dreht sich nur nach links oder rechts.
- Die Katze dreht sich gar nicht.

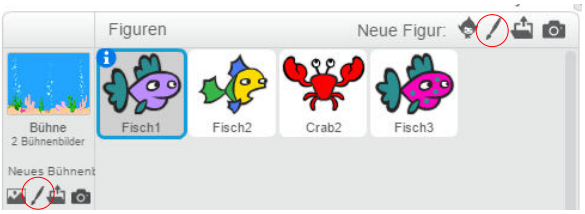
Koordinatensystem:
Willst du einer Figur die exakte Position auf der Bühne zuweisen, so verwende Koordinaten. Das sind Zahlenpaare, bestehend aus einem x- und einem y-Wert. Mit ihnen kann jeder Punkt auf der Bühne genau angegeben werden.
Die Bühne ist 480 Pixel breit und 360 Pixel hoch (Pixel sind die Bildpunkte aus denen die Bildschirmoberfläche aufgebaut ist).

Die Mitte der Bühne befindet sich beim Punkt x: 0/ y: 0 Der x-Wert gibt an, ob die Figur rechts oder links (negative Zahl) der Mitte steht und wird als erste der beiden Zahlen erwähnt. Der y-Wert gibt an, ob die Figur über oder unter (negative Zahl) der Mitte steht.

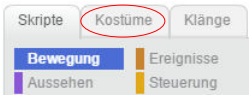


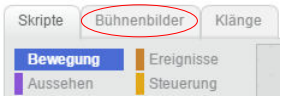
Hier kannst du üben:
https://scratch.mit.edu/studios/3502723/

Figuren selber zeichnen
Wenn du eine Figur oder einen Bühnenhintergrund selber zeichnen oder verändern willst, wählst du in der Figuren- oder Bühnenliste den Pinsel.

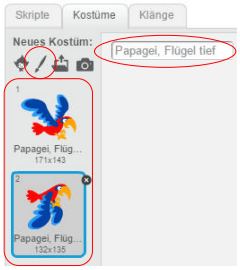


Danach klickst du auf den Reiter *Kostüme* od. *Bühnenbilder*:





Figuren können mehrere Kostüme und die Bühne mehrere Bühnenbilder haben. Diese sind unterhalb des Kostüm-Pinsels, bzw. des Bühnenbilder-Pinsels aufgeführt und sollten aussagekräftig benannt werden, damit du in deinen Skripts ohne Probleme die korrekten Kostüme anwählen kannst. Durch Klicken auf den Pinsel können weitere Kostüme und Bühnenbilder erstellt werden.



Jetzt muss eine grundsätzliche Entscheidung getroffen werden. In der unteren rechten Ecke kann zwischen den beiden Zeichnungs-Modi *Vektor Modus* und *Rastergrafik (Bitmap)-Modus* gewählt werden:

Bitmap oder Vektor?
Um korrekt entscheiden zu können, ist es wichtig die beiden Modi zu kennen:

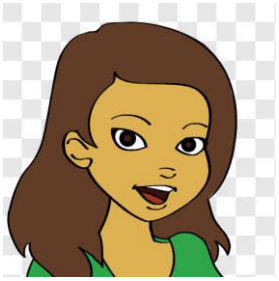
Ein **Bitmap-Bild** besteht aus Tausenden winziger Quadrate (Pixel). Diese Pixel sind alle gleich groß, können aber unterschiedlichste Farben haben.

Vorteil:
- Die Anwendung ist unkompliziert und mehr oder weniger selbsterklärend.

Nachteile:
- Es ist beim Zeichnen schwierig, mit der Maus ein befriedigendes Resultat zu erreichen.
- Nachträgliche Änderungen sind unter Umständen mit viel Aufwand verbunden.
- Bei Vergrösserung wirkt das Bild verpixelt.

Anwendung:
Eignet sich besonders gut für Fotos. Deshalb sind viele Hintergründe im Bitmap-Modus gehalten.

Eine **Vektorgrafik** besteht aus berechneten Linien. Diese werden bei Vergrösserung neu berechnet, so dass das Bild nie verpixelt wird.



Vorteile:
- Beim Zeichnen hast du die grössere Kontrolle über das Resultat.
- Nachträgliche Änderungen sind einfach zu machen.
- Bei Vergrösserung bleibt das Bild gestochen scharf.

Nachteile:
- Die Anwendung ist etwas schwieriger zu lernen.
- Wenn ein Projekt viele komplizierte Vektorgrafiken hat, kann dies das Projekt verlangsamen.

Anwendung:
Bei den meisten Zeichnungsprojekten in Scratch, sei es für Hintergründe oder Figuren, aber auch für Text, ist der Vektor-Modus die richtige Wahl.

Zeichnen im Bitmapmodus
Viele Werkzeuge sind mehr oder weniger selbsterklärend:


Pinsel: Ermöglicht das freihändige Zeichnen (soweit dies mit der Maus möglich ist…).
Wie bei vielen anderen der nachfolgenden Werkzeuge kann die Strickdicke unten links gewählt werden, wenn das entsprechende Werkzeug angeklickt ist.


Linie: Zeichnet gerade Linien. Drücken der Umschalttaste ergibt senkrechte, bzw. waagrechte Linien.


Rechteck: Zeichnet Rechtecke. Drücken der Umschalttaste ergibt ein Quadrat.


Ellipse: Zeichnet Ellipsen. Drücken der Umschalttaste ergibt einen Kreis.


Textwerkzeug: Ermöglicht das Schreiben von Text. Wenn es angewählt ist, kann unten links zwischen sechs Schriften gewählt werden.
Wenn der Text fertig geschrieben ist, kann er verschoben oder verzogen werden, solange die „Anfasser“ sichtbar sind.


Füllwerkzeug: Füllt abgegrenzte Flächen mit der gewählten Farbe. Möglich ist es auch, einen Farbverlauf herzustellen. Dazu muss, nachdem das Füllwerkzeug ausgewählt worden ist, unten links der gewünschte Verlauf angeklickt werden.


Radiergummi: Putzt ohne Rücksicht auf Verluste alles weg.


Auswahlwerkzeug: Wählt den angewählten Teil der Malfläche aus, so dass dieser verschoben, gedreht oder gelöscht werden kann.

Zauberstab: Soll helfen, eine Figur vom Hintergrund zu trennen, ist in der Anwendung aber ziemlich schwierig und oft nicht wirklich erfolgversprechend. Die Figur, welche behalten werden soll, muss mit dem grünen Stift inwendig umrundet werden. Danach erscheint der abzutrennende Teil schwarzweiss. Falls das Ergebnis ok ist, kann irgendwo ausserhalb der karierten Arbeitsfläche geklickt werden und die freizustellende Figur sollte übrigbleiben.

Stempel: Mit ihm kann ein Bereich ausgewählt werden, der anschliessend kopiert wird.

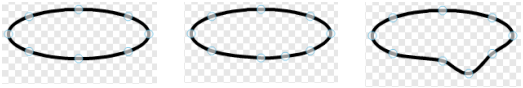
Zeichnen im Vektormodus

Werkzeuge, welche im Vektormodus ähnlich wie im Bitmapmodus aussehen, funktionieren anders:

Auswahlwerkzeug: Wird ein Objekt mit ihm angewählt, erhält es eine blaue Umrandung. Jetzt kann es verschoben, verzogen (an den quadratischen Anfassern), gedreht (am runden Anfasser) oder gelöscht werden (mit der Backspace/Rück-Taste oder der Delete-Taste).

Verformwerkzeug: Wird ein Objekt mit ihm angeklickt, erscheinen runde Punkte, an denen das Objekt verformt werden kann. Folgende Techniken gilt es zu kennen:

Zusätzlicher Verform-Punkt hinzufügen: Zwischen zwei Punkte auf die Linie klicken.

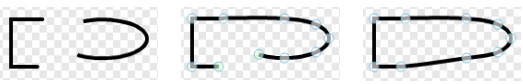


Kurven-Knotenpunkt auf einer geraden Linie einsetzen: bei gedrückter Umschalttaste zwischen zwei Punkte klicken.

Punkt verschwinden lassen: auf den Punkt klicken, ohne ihn zu bewegen.

Knotenpunkt auftrennen: bei gedrückter Umschalttaste auf den Punkt klicken

Zwei Knotenpunkte verbinden, um eine Form füllen zu können: die zwei Punkte aufeinander ziehen.



Stift: Funktioniert ähnlich wie der Pinsel. Allerdings kann die gezeichnete Figur mit dem Auswahlwerkzeug ausgewählt und dann verschoben, verzogen oder gedreht werden. Mit Hilfe des Verformwerkzeuges kann die Figur abgeändert werden.

Linien: Zeichnet gerade Linien. Drücken der Umschalttaste ergibt senkrechte, bzw. waagrechte Linien. Zusätzlich kann die gezeichnete Linie mit dem Auswahl- und dem Verformwerkzeug abgeändert werden.

Rechteck: Zeichnet Rechtecke. Drücken der Umschalttaste ergibt ein Quadrat. Zusätzlich kann das gezeichnete Rechteck mit dem Auswahl- und dem Verformwerkzeug abgeändert werden.

Ellipse: Zeichnet Ellipsen. Drücken der Umschalttaste ergibt einen Kreis. Zusätzlich kann die gezeichnete Ellipse mit dem Auswahl- und dem Verformwerkzeug abgeändert werden.

Textwerkzeug: Ermöglicht das Schreiben von Text. Wenn es angewählt ist, kann unten links zwischen sechs Schriften gewählt werden.
Der geschriebene Text kann mit dem Auswahlwerkzeug jederzeit verschoben, verzogen oder gedreht werden. Zusätzlich kann der Text auch jederzeit umformuliert werden, indem in ihn hineingeklickt wird.

Füllwerkzeug: Färbt den Inhalt von Objekten oder deren Umrandung, je nachdem, wohin geklickt wird. Auch das Färben von Text ist möglich. Schliesslich können auch Farbverläufe wie im Bitmapmodus erzeugt werden.

Stempel: Dupliziert ein Objekt, sobald es angeklickt wird.

Ebenenwerkzeug: Wird nur angezeigt, wenn mindestens ein Objekt ausgewählt ist. Bringt das gewählte Objekt eine Ebene nach vorne.

Ebenenwerkzeug: Wird nur angezeigt, wenn mindestens ein Objekt ausgewählt ist. Bringt das gewählte Objekt eine Ebene nach hinten.

Gruppieren: Wird nur angezeigt, wenn mindestens zwei Objekte ausgewählt sind. Gruppiert die ausgewählten Objekte auf einer Ebene zu einem Objekt.

Gruppierung aufheben: Wird nur angezeigt, wenn ein gruppiertes Objekt ausgewählt ist. Hebt die Gruppierung auf.

Weitere Werkzeuge

Rückgängig machen: Stellt den Zustand vor der letzten Aktion wieder her (Ctrl-Z ist die Alternative). Mit dem entgegengesetzten Pfeil kann auch diese Aktion wieder rückgängig gemacht werden.

Zuschneide-Werkzeug: Schneidet im Bitmapmodus alles weg, was sich ausserhalb des Auswahl-Bereichs befindet.

Spiegelung-Werkzeug: Dreht die Auswahl um eine Senkrechte.

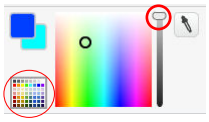
Spiegelung-Werkzeug: Dreht die Auswahl um eine Waagrechte.

Drehpunkt-Werkzeug: Mit ihm kann festgelegt werden, um welchen Punkt sich ein Kostüm drehen soll. Der Drehpunkt eines Kostüms bestimmt auch, wohin genau eine Figur bei einer Koordinatenangabe verschoben wird.

Arbeiten mit Farben

Werkzeuge wie Pinsel, Linie, Rechteck, Ellipse, aber auch das Textwerkzeug verwenden immer diejenige Farbe, welche gerade ausgewählt ist. Angezeigt wird diese Farbe mit dem oberen der beiden Quadrate neben der Farbpalette.

Die Farbe des unteren Quadrates wird für die Farbverläufe des Füllwerkzeuges gebraucht. Zwischen den beiden kann gewechselt werden, wenn auf eines von ihnen geklickt wird.
Ausgewählt werden kann eine Farbe entweder aus der Palette mit 55 vorgegebenen Farben (das Quadrat mit dem roten Schrägstrich steht für Durchsichtigkeit) oder mit der Palette, welche Tausende von Farben bereithält. Dort kann zusätzlich auch die Helligkeit der Farben mit einem Schieberegler beeinflusst werden. Um von einer Palette zur anderen zu wechseln, wird auf das kleine Abbild der entsprechenden Palette geklickt.



Soll eine bestimmte Farbe verwendet werden, die auf der Zeichnungsfläche schon vorkommt, kann auch die Pipette zur Farbauswahl genutzt werden. Nachdem die Pipette ausgewählt wurde, kann mit ihr auf die gewünschte Farbe geklickt werden und sofort wird diese zur ausgewählten Farbe.

Es gibt verschiedene Möglichkeiten, die Bewegungen einer Figur zu beeinflussen:

Maussteuerung

Die Figur folgt dem Mauszeiger:

Die Figur folgt dem Mauszeiger, aber nur in waagrechter Richtung:

Die Figur richtet sich nach dem Mauszeiger aus und schwebt laufend auf ihn zu:

Pfeiltastensteuerung

Die Figur bewegt sich in die Richtung des Pfeiltastendrucks, wartet aber bei länger gedrückten Taste nach der ersten Bewegung einen Moment, bevor weitere Bewegungen ausgeführt werden. →Eignet sich für Einzelklicks:

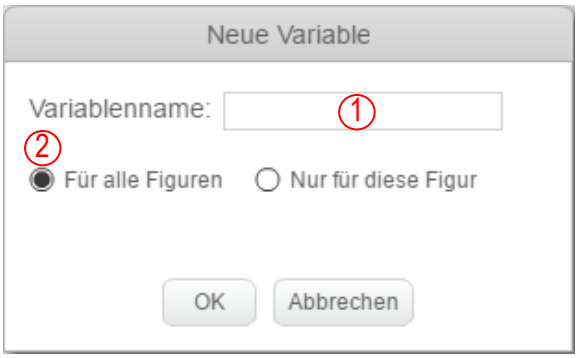
Die Figur bewegt sich ohne Verzögerung fortlaufend in die Richtung der gedrückten Pfeiltaste. → Eignet sich für länger gedrückte Pfeiltasten:

Die Figur bewegt sich nach den Tastaturbefehlen, wird aber durch farbige Objekte, beispielsweise (hier: blaue) Wände gestoppt.

Variablen können sich Daten merken. Mit ihnen werden zum Beispiel Punktestände, Programmeinstellungen oder Antworten auf Frage gespeichert.
Variablen sind vergleichbar mit Schachteln, in denen Informationen aufbewahrt werden: Die Schachteln bleiben, der Inhalt kann geändert werden.

Variable erstellen:
• Im Register *Skripte* auf *Neue Variable* klicken

• Variable benennen



1) Optimal ist ein aussagekräftiger, aber kurzer Namen
2) in den meisten Fällen sollen alle Figuren auf die Variable zugreifen können.

Variable benutzen:

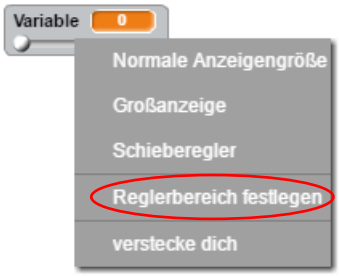
1) Die Variable, welche an ihrer abgerundeten Form erkannt wird, kann nun durch drag and drop in andere Blöcke eingesetzt werden. Möglich ist das in Blöcken mit einem runden (gehe 10 er-Schritt) oder viereckigen Platzhalter (sage).

Das Häklein vor der Variable bedeutet, dass die Variable im Projekt sichtbar ist. Soll sie weg, genügt ein Klick auf das Häklein. Gelöscht werden kann eine Variable durch einen Rechtsklick auf sie.
2) In einem Script wird damit der Wert der Variable festgelegt.
3) Damit wird die Variable verändert. Mit negativen Zahlen kann auch rückwärts gezählt werden.
4) Soll eine Variable nur zeitweilig in einem Projekt zu sehen sein, kann dies mit den beiden Blocks gesteuert werden.

Aussehen der Variablen verändern:
Das Aussehen der Variable kann durch Doppelklick im Bearbeitungsmodus auf folgende drei Arten geändert werden:

Normalanzeige | Grossanzeige | Schieberegler

Der Schieberegler ermöglicht es, eine Variable zu ändern, während ein Skript am Laufen ist. Sein Bereich kann durch einen Rechtsklick festgelegt werden.



Vorgegebene Variablen
Nebst den selbst definierbaren Variablen stellt Scratch eine Reihe vorgegebener Variablen zur Verfügung. Es lohnt sich, bei den Blöcken danach Ausschau zu halten. Beispiele:

x-Position
y-Position
Geben die Position einer Figur an.

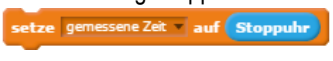
Kostümnummer — Gibt die Nummer des verwendeten Kostüms an.

Lautstärke — Gibt die Lautstärke des verwendeten Klanges an.

Antwort — Gibt die Antwort auf die mit dem Frageblock (frage und warte) gestellten Frage an.

Lautstärke — Gibt die Lautstärke des vom Mikrofon empfangenen Klanges an.

Stoppuhr — Gibt fortlaufend die seit dem Starten des Projektes oder seit dem letzten Zurücksetzen der Stoppuhr vergangene Zeit an. Mittels folgender Blockkombination kann die Zeit gestoppt werden:



Zufallszahl von bis — Zahl aus dem angegebenen Bereich an.

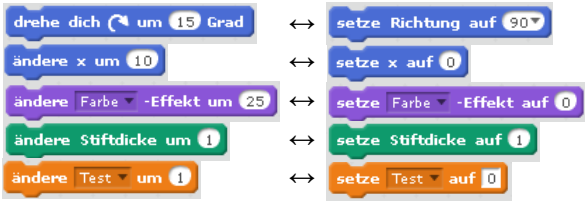
Recht häufig kommt es vor, dass ein Computer-Programm beim ersten Test alles andere macht, als was es sollte. In den seltensten Fällen ist dann der Computer schuld. Meistens hat der Scratcher einen Denkfehler oder eine Unaufmerksamkeit begangen und muss nun den Fehler suchen.
Programmierer nennen das *Debugging*. Das Wort kommt vom Wort *Bug*. In grauer Computer-Vorzeit, im Jahre 1947, wurde eine Motte als Ursache für eine Computerpanne ausgemacht und seither hält sich der Name Bug (Ungeziefer, Wanze) für Programmfehler.

Debuggen ist nicht besonders spannend, aber manchmal kommt man nicht darum herum. Die nachfolgende Checkliste kann dir beim Suchen helfen, wenn dein Programm nicht so läuft, wie du dir das vorgestellt hast:

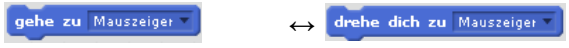
1. Skript am falschen Ort
Hast du das Skript bei der richtigen Figur geschrieben?

2. Vergessene Blocks
Gehe für dich in Gedanken die einzelnen Blocks durch. Machen sie Sinn oder fehlt etwas? Falls du einen Code abschreibst: Zähle die Blocks nach. Vielleicht kannst du deinen Code auch mit deiner Kollegin, deinem Kollegen vergleichen.

3. Verwechselte Blocks
Hast du die richtigen Blocks verwendet? Besonders häufig werden Blocks mit relativen und absoluten Anweisungen verwechselt:



Auch Blocks mit ähnlichen Anweisungen sind gefährdet:

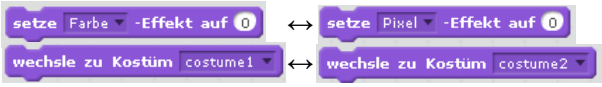


Vielleicht haben die Blocks aber auch nicht die richtige Farbe und damit eine falsche Funktion:

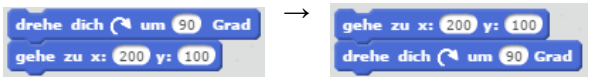
4. x und y verwechselt
Unter Umständen hast du einen X-Block statt einen Y-Block erwischt oder umgekehrt?

5. Falsche Zahlen
Gerade wenn du ein Skript oder ein Skript-Teil kopiert hast, ist es oft notwendig, noch die Zahlen oder das Vorzeichen zu ändern:

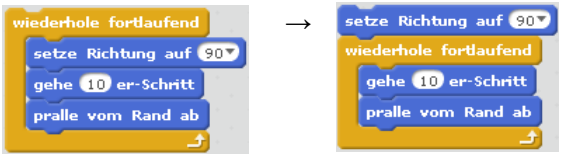
6. Falsche Auswahl
Ebenfalls geht das Ändern der Dropdown-Menüs schnell vergessen:



7. Falsche Reihenfolge
Sind die Blocks in der korrekten Reihenfolge?



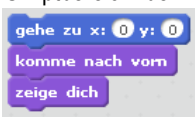
8. Schleifen
Sind Blocks innerhalb von Schleifen, wenn sie ausserhalb sein sollten oder umgekehrt?



Oder fehlt gar eine Schleife? Statt so,

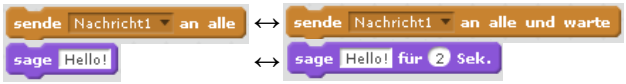
sollte das Skript so aussehen:

Tipp: Verschwundene Figur
Wenn plötzlich eine Figur verschwunden ist, hilft oft das folgende Skript. Schreibe es in den Skriptbereich der vermissten Figur und klicke darauf: →



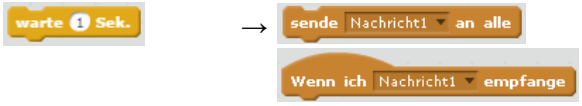
9. Zeitliche Begrenzung

Sind Aktionen zeitlich begrenzt, wo sie es nicht sein sollten oder umgekehrt?



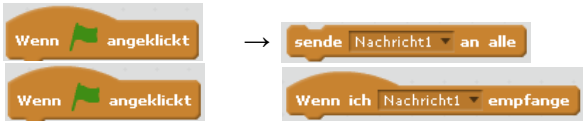
10 Falsches Timing

Um einen genauen zeitlichen Ablauf mit mehreren Figuren festzulegen, sind die *sende-empfange*-Blocks den *warte*-Blocks vorzuziehen.



11. Gleichzeitigkeit

Wenn mehrere Skripts mit dem Flaggenblock beginnen, ist unter Umständen nicht klar, welches Skript zuerst abgearbeitet werden soll. *Warte*-Blocks einschieben kann da helfen. Besser ist es aber, die *sende-empfange*-Blocks zu verwenden, um den genauen Ablauf festzulegen:

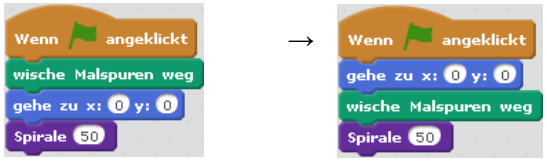


12. Lokale oder allgemeine Variable?

Kannst du bei einer Figur eine Variable nicht einsetzen, hast du sie vermutlich als lokale Variable definiert. Lokale Variablen erscheinen im Dropdown-Menü unterhalb eines schwarzen Querstriches. →

13. Initialisieren

Beim Initialisieren (bei den Startvorbereitungen) musst du darauf achten, dass Spuren des letzten Skript-Gebrauchs nicht den Neustart beeinflussen.

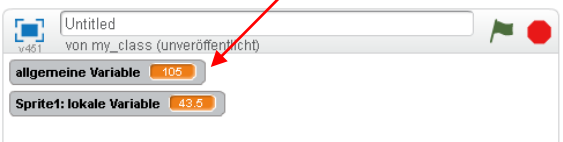


14. Zu viele Klone

Wenn dein Skript Klone erstellt, müssen einige davon unter Umständen zuerst wieder gelöscht werden, bevor neue erstellt werden. →

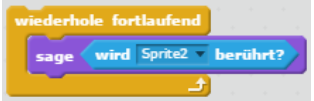
Tipp: Werte/Zustände testen

Um besser verstehen zu können, was das Programm gerade macht, ist es oft hilfreich, die Variablenwerte auf der Bühne anzeigen zu lassen: →



Möglich ist es aber auch, einen Wert laufend durch einen *sage*-Block anzeigen zu lassen:

Diese Methode hat den Vorteil, dass sie auch noch für andere Informationen herangezogen werden kann:



Allgemein
http://search.creativecommons.org/
http://www.newgrounds.com/

Grafiken
https://openclipart.org/Opensprites.org
Iconfinder.com („free“ als Preis wählen)
https://opengameart.org/
http://www.iconarchive.com/
https://publicdomainvectors.org/
http://www.clker.com/

Figuren, Hintergründe
http://www.openpixelproject.com/

Button erstellen
http://dabuttonfactory.com
http://www.freebuttons.com/index.php
http://www.buttongenerator.com/

Titel-/Textgenerator
https://de.cooltext.com/

Schriften
http://www.fontsplace.com/playhouse-free-font-download.html
http://www.dafont.com/

GIF herstellen
https://makeagif.com/
http://ezgif.com/

Animationen für Figuren
Opengameart.org

Fotos
Unsplash.com
https://commons.wikimedia.org/wiki/Main_Page
Pixabay.com
Wikipedia.org
https://www.flickr.com/
http://piqs.de/
http://pics4learning.com/

Upload-Website für DiskussionsForen-Pictures (white-listed host)
www.photobucket.com

Geräusche
Freesound.org
http://soundbible.com/
http://www.hoerspielbox.de/
http://www.auditorix.de/index.php?id=183
http://www.audiyou.de/

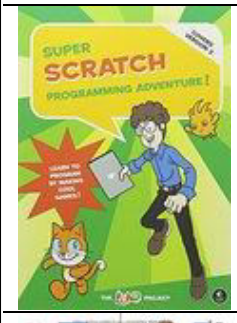
Musik
https://www.jamendo.com/?language=de
https://incompetech.com/
http://audionautix.com/
http://ericskiff.com/music/
www.ibiblio.org/mutopia
www.bensound.com
bit.do/beaterie
Opengameart.org
http://incompetech.com/
https://www.youtube.com/channel/UC_aEa8K-EOJ3D6gOs7HcyNg

Pseudocode erstellen
http://scratchblocks.github.io/
http://scratchblocks.github.io/generator/

Projekte beurteilen lassen
http://www.drscratch.org/

Scratch Tools
https://scratchtools.tk/

Scratch Compiler (Scratch → JavaScript)
https://phosphorus.github.io/

	POLLOCK, WILLIAM *Super Scratch Programming Adventure!* San Francisco no starch press 2014	leichtes Englisch, farbig, für fortgeschrittene Anfänger, 158 Seiten, rund sfr 20.- ★★★★	• Learning by doing! • Motivierende Projekte (mehrheitlich Computerspiele) • Gut aufgebaut mit Lernzielangaben zu Beginn jedes Kapitels • Knappe, verständliche Anweisungen, welche von guten Englisch-Schülern mit etwas Unterstützung mehr oder weniger selbständig durchgearbeitet werden könnten • Sehr übersichtliches Layout • Für die Programme benötigte Sprites/Bühnen, wie auch die Lösungen, können downgeloadet werden. • Bestes Spiel des Buches: Escape the Maze
	WOODCOCK, JON: *Programmieren supereasy* London DK 2015	deutsch, farbig, für Anfänger, 224 Seiten, rund sfr. 20.- ★★★½	• Leicht verständliche, grundlegende Infos zum Programmieren und zum Computer • Reich bebilderte Einführung in Scratch • Allerdings enthält das Buch nur wenige, einfache Projekte um mit Scratch zu trainieren. • Der zweite Teil des Buches führt in die Programmiersprache Python ein. Python ist keine grafische Programmiersprache und deshalb weniger attraktiv zum Lernen.
	IMMLER, CHRISTIAN *Der kleine Hacker: Programmieren für Einsteiger* FRANZIS 2015	deutsch, farbig, für fortgeschrittener Anfänger, 192 Seiten rund sfr.30.- ★★★½	• Grösstenteils leicht verständliche Einführung in Scratch • Arbeitet von Beginn an mit interessanten Projekten. Höhepunkte: - Guillochen-Programm - Flappy Bird (Geschicklichkeits-Game) - Senso (Gedächtnisspiel, anspruchsvoll zum Programmieren) • Skripte sind zum Teil recht unübersichtlich und nicht unbedingt ein Vorbild. • Referenz zu allen Scratchblöcken • Für die Projekte benötigte Dateien liegen auf einer DVD bei.
	BREEN, DEREK: *Erste Schritte mit Scratch* Hoboken John Wiley & Sons. Inc. 2016	deutsch, farbig, für Anfänger, 128 Seiten, rund sfr. 12.- ★★★	• Der Schwerpunkt liegt auf dem Gestalten mit Scratch (Comics zeichnen, Tiere zeichnen, Vektorbilder, digitale Collagen). • Programmiert wird nur wenig. • Aus dem Inhalt: - Spiel: Flappy bats - eigener Comic zeichnen - Tiere zeichnen und zum Bewegen bringen - Roboter mit Vektorgrafik zeichnen - Collage anfertigen • Viele der vorgestellten Programme können im Internet eingesehen werden: https://scratch.mit.edu/studios/489897/
	WOODCOCK, JON: *Spiele programmieren – supereasy* London DK 2016	deutsch, farbig, für Fortgeschrittene, 222 Seiten rund sfr. 20.- ★★★★	• Kurze Einführung (Game-Typen, wie ein gutes Game schreiben, Scratch-Oberfläche) • Danach werden Spiele (u. a. Jump'n'run, Ballergame, Autorennen, Denkspiel, etc.) in mehr oder weniger aufsteigendem Schwierigkeitsgrad programmiert und die Features von Scratch nach und nach eingeführt. • Im Verlauf der Lektüre erfährt man auch den einen oder anderen Trick zur Spieleprogrammierung. • Für zukünftige Game-Designer dürfte das Spiel Dog's dinner am interessantesten sein. Nach dessen Durcharbeitung ist man in der Lage, selber Platformer zu schreiben. • Das anspuchsvollste, der hier vorgestellten Bücher.
	JONATHAN MELMOTH & ANDERE, *Ganz easy programmieren lernen: Scratch* USBORNE 2017	deutsch, farbig, für Anfänger, 96 Seiten rund sfr. 15.- ★★★★	• Alle wesentlichen Features von Scratch werden anhand von verschiedenen Projekten und Spielen erklärt. • Die Beispiele sind kindgerecht, aber folglich auch eher einfach gehalten. • Mittels der im Buch gelieferten Links kommt man an einige vorbereitete und fertige Buchbeispiele heran.