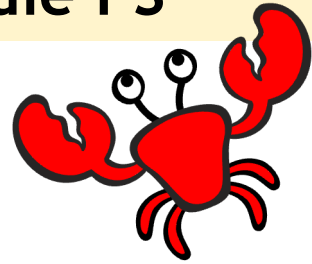


Scratch Projektideen

Kreative Algorithmen:
Für die Sek I und die PS



SCRATCH

Michael Hielscher
Beat Döbeli Honegger
3. Februar 2019

Überarbeitung: Giulia Paparo
25. April 2024



Worum geht es?

Scratch ist eine kostenlose Programmierumgebung für Kinder, die am MIT entwickelt wird. Scratch ist leicht zu erlernen und eignet sich sowohl für einfache Animationen als auch für komplexe Spiele und Simulationen. Du kannst mit wenigen Klicks eigene Bilder und Töne erstellen und damit deine ganz persönlichen Projekte entwickeln. Scratch lässt sich auch mit echten Robotern und Sensoren kombinieren. Kurz gesagt: Scratch macht einfach Spass und bietet unendlich viele Möglichkeiten! Alles was du brauchst ist ein Webbrowser.



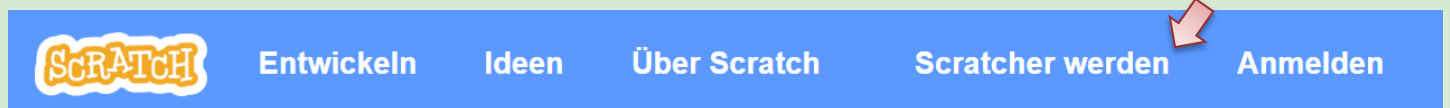
Wie funktioniert es?

Hast du bereits mit Scratch gearbeitet? Dann kannst du dieses Blatt überspringen.

1. Bei Scratch anmelden


Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox.


Gehe auf <https://scratch.mit.edu>. Es lohnt sich, ein Scratch-Konto anzulegen. Klicke dazu auf «Scratcher werden» oben rechts und erstelle dir einen eigenen Account, oder melde dich an, wenn du schon einen Account besitzt.

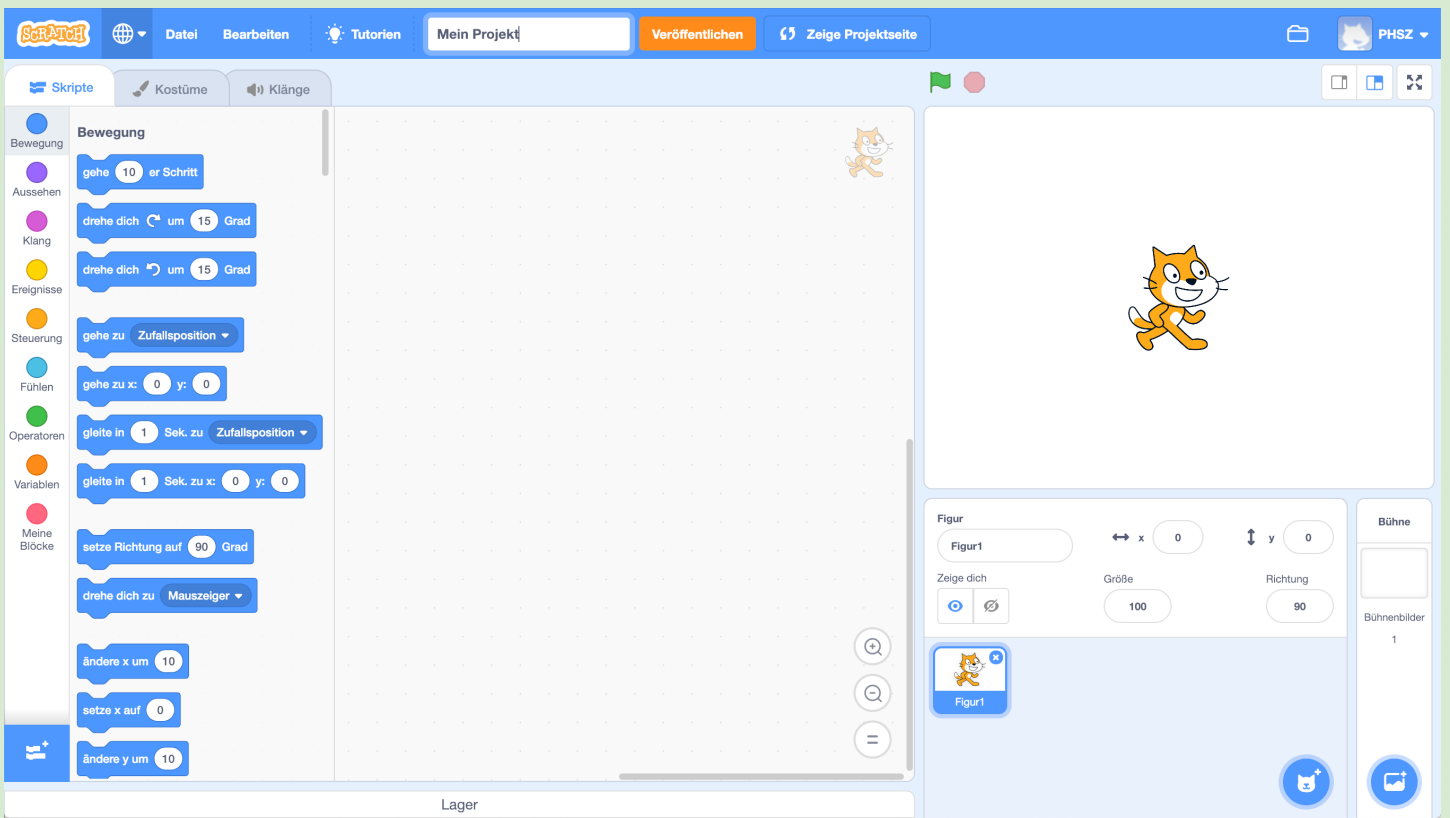


Du benötigst die folgenden Angaben: Einen Benutzernamen (wähle etwas Phantasievolles), ein Passwort, das Land deines Wohnsitzes, dein Geburtsdatum und eine E-Mail-Adresse.

2. Ein erstes Scratch-Projekt erstellen

Klicke auf der Startseite von Scratch auf «Entwickeln», um ein neues Scratch-Projekt zu beginnen. Wähle  oben rechts, wenn du an einem begonnenen Projekt weiterarbeiten möchtest.

Für den Einstieg gibt es mehrere kleine Projekte mit einer Schritt-für-Schritt Anleitung. Klicke dazu auf  Tutorien und wähle eine Anleitung aus. Folge den Anweisungen, um Scratch kennenzulernen.



Worum geht es?

Programmieren ist eine Tätigkeit, die vor allem im Kopf passiert und auch anstrengend und fehleranfällig sein kann. Schnell hat man etwas nicht bedacht oder verliert viel Zeit bei einem kleinen unwichtigen Detail.

Pair-Programming ist eine Arbeitstechnik, bei der zwei Programmierende zusammenarbeiten, um gemeinsam bessere Ergebnisse zu liefern als alleine. Eine Person erstellt das Programm und die andere behält das Gesamtprojekt im Blick. Gemeinsam spricht man fortlaufend darüber, was man tut und denkt.



Wie funktioniert es?

Rollen beim Pair-Programming

Jeder übernimmt eine Rolle – entweder bist du der *Driver* oder der *Navigator*. Das ist etwa so wie beim Rallye fahren, wo jemand fährt und der andere die Karte liest und ansagt, welche Kurve als nächstes kommt. Der *Driver* entwickelt das Programm und bedient den Computer. Der *Navigator* überlegt was als nächstes zu tun ist, denkt über Problemstellungen nach, kontrolliert das entstehende Programm und macht Vorschläge. Die beiden Rollen solltet ihr regelmässig tauschen.

Regeln beim Pair-Programming

Damit Pair-Programming funktioniert, müssen einige wichtige Regeln beachtet werden.

1. Miteinander sprechen

Sprecht über alles, was ihr gerade tut und denkt. Stille ist ein schlechtes Zeichen.

2. Rollen einhalten

Auch wenn du gerade besser weisst, wie etwas geht – nicht einfach ans Keyboard oder an die Maus greifen.

3. Bleibt gelassen

Auch wenn es gerade mal nicht so gut klappt – streitet euch nicht! Kein Beleidigen und Fluchen.

4. Driver lenkt

Erkläre immer, was du gerade tust. Begründe, warum du etwas tust und mit welchem Ziel.

5. Beim Thema bleiben

Sprecht über das Projekt und schweift nicht ab. Macht keine anderen Dinge nebenbei!

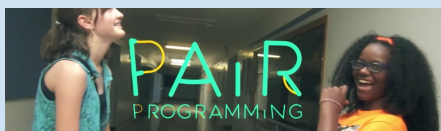
6. Navigator denkt

Kommandiere den *Driver* nicht herum und sag ihm nicht ständig, was er genau zu tun hat.

Ein Projekt mit Pair-Programming

Überlegt euch zusammen ein kreatives Scratch-Projekt, welches ihr gerne umsetzen möchtet. In dieser Broschüre findet ihr einige Ideen und Anregungen. Entscheidet, wer mit welcher Rolle startet und wechselt euch im Laufe der Entwicklung immer wieder ab.

Trefft alle Entscheidungen gemeinsam – welche Farbe soll eine Figur haben, welcher Text soll eingeblendet werden ...



Fiona und Samira erklären die Idee von Pair-Programming (englisch)
<https://youtu.be/vgkAhOzFH2Q>



Hinweis

Pair-Programming ist nicht etwa eine Erfindung von Pädagoginnen und Pädagogen oder gar nur ein anderer Begriff für klassische Partnerarbeit! Diese Arbeitstechnik wird tatsächlich in der Software-Entwicklung eingesetzt, um weniger Fehler zu machen und besseren Code zu schreiben.

Worum geht es?

Du bist der Regisseur oder die Regisseurin! Wer kommt in einer Szene auf die Bühne und wer sagt was? Mit Scratch kannst du kleine Filme oder Animationen erstellen, die natürlich auch kleine Kunstwerke sind. Du kannst die Figuren und Hintergründe so programmieren, dass sie sich nach deinen Vorstellungen bewegen und sprechen. Vom kleinen Witz bis zum ganzen Theaterstück ist alles möglich. Auch Geschichten sind Kunstwerke: Programmiere sie!



Wie funktioniert es?

1. Schreib deine eigene Geschichte


Um eine gute Geschichte zu erzählen, benötigst du zunächst ein gutes Drehbuch. Darin sind die Szenen und die verschiedenen Rollen (Figuren) und ihre Handlungen, Positionen auf der Bühne und Texte festgehalten. Beispiel: *Szene 1: Im Park*






Luca erscheint auf der Bühne und sagt: «Mir ist langweilig». Anina kommt von rechts auf die Bühne und will mit Luca sprechen. Luca ignoriert sie, er will eine Zukunftsreise machen und fliegt nach oben und verschwindet von der Bühne. Anina verschwindet auch von der Bühne.

Szene 2: In Lucas Kopf

Luca erscheint auf der Bühne und fragt sich, wo er ist. Dann erscheint eine Fledermaus von rechts auf der Bühne und läuft zu Luca und sagt: «Luca, du bist in deinem Kopf! Die Zukunft existiert noch nicht, die musst du erfinden.»

2. Bilder und Töne zusammenstellen

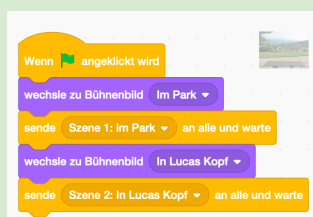
Nachdem du im Drehbuch die verschiedenen Figuren und Szenen beschrieben hast, kannst du diese in Scratch anlegen. Wähle aus der Bibliothek passende Figuren  und Bühnenbilder aus.

Ist kein passendes Bild oder Geräusch dabei, kannst du auch eigene Figuren und Hintergründe zeichnen () oder freie Bilder aus dem Internet (z. B. von www.pixabay.com) herunterladen und in Scratch hochladen (). Unter  kannst du passende Geräusche für jede Figur auswählen () oder selbst aufnehmen ().

3. Übernimm die Regie

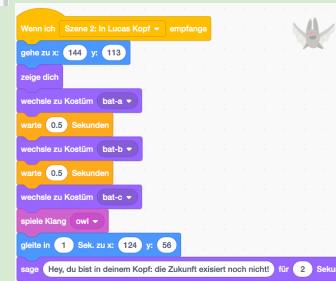
Wähle unten rechts die «Bühne» aus. Der Regisseur oder die Regisseurin des Theaterstücks dirigiert unsichtbar im Hintergrund und ruft allen Figuren zu, welche Szene als nächstes an der Reihe ist. Zudem wählt er das passende Bühnenbild aus. Die Figuren hören auf seine Anweisungen. Ausserdem können sich die Figuren untereinander auch Anweisungen schicken, z. B. dass Anina verschwindet nachdem Luca wegfliegt.

Regie-Skripte (Bühne):



Für jede Szene musst du für alle Figuren festlegen, ob und wann sie auf die Bühne kommen und gehen. In Szene 1 spielt die Fledermaus nicht mit (Drehbuch).

Beispiel-Skripte von Luca, Anina und der Fledermaus für Szene 1 und 2



Beispiele

Einen Witz erzählen
scratch.mit.edu/projects/517436688

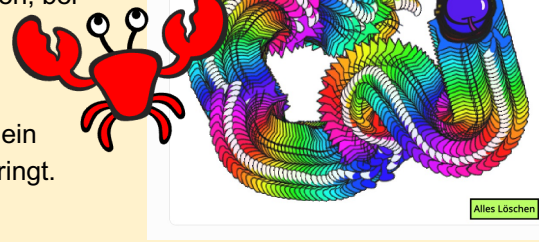
Reise in die Zukunft
scratch.mit.edu/projects/514212519

Wie weiter?

- Erstelle bei allen Figuren die Skripte für Szene 3 und 4.
- Füge in der Bühne passende Hintergrundmusik zu jeder Szene hinzu.
- Kannst du deine Geschichte «interaktiv» gestalten, sodass Zuschauer zum Beispiel Dinge anklicken oder sogar selbst steuern können?

Worum geht es?

In diesem Scratch-Projekt kannst du deine eigene digitale Malerei gestalten, bei der die Objekte Spuren oder Abdrücke hinterlassen und die Farbe wechseln, während du den Mauszeiger steuerst. Jede Bewegung deiner Maus erzeugt einzigartige Muster und Formen auf der Leinwand. Durch die Kombination von Farben und Bewegungen entsteht ein lebendiges Kunstwerk, das deine persönliche Kreativität zum Ausdruck bringt.



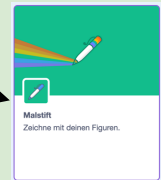
Inspiziert von: Colorful Trails von natalie.

Wie funktioniert es?


1. Malstift-Erweiterung auf Scratch hinzufügen

Als erstes füge unten links unter Erweiterungen  den Malstift hinzu.

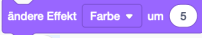
Jetzt kannst du in Scratch einen Malstift programmieren, der dir bei deinem Kunstprojekt helfen wird.

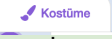


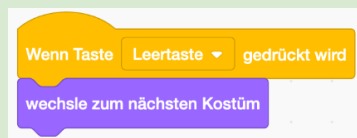
2. Figur auswählen und Abdruck hinterlassen

Wähle eine Figur aus und programmiere sie so, dass sie sich immer vorwärts bewegt, dem Mauszeiger folgt und einen Abdruck hinterlässt. Den Block  findest du in der Blockpalette unter Zeichenstift.



3. Bewegung, Farbe und andere Kostüme

Um deinem Kunstwerk mehr Dynamik zu verleihen, kannst du in der «wiederhole fortlaufend» Schleife den Block  einfügen.

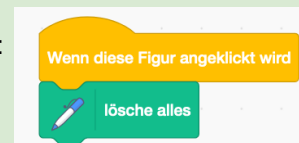
Unter  kannst du auch verschiedene Kostüme für deine Figur auswählen und so programmieren, dass immer, wenn du die Leertaste drückst, ein anderes Kostüm ausgewählt wird. Dann sieht das Skript für deine Figur so aus:



4. Löschtaste

Wenn man so malt, ist der Bildschirm schnell voll. Deshalb wollen wir eine Taste einbauen, die das Bild wieder leert. Unter Figuren kannst du mit dem Rechteck-Werkzeug  eine Taste zeichnen und mit dem Text  «Löschen» beschriften.

Dass die Taste bei jedem Klick alles löscht, kann wie folgt programmiert werden:



Beispiele

Hinterlasse eine Spur

<https://scratch.mit.edu/projects/997577889/>

Originalprojekt:

scratch.mit.edu/projects/10001052

Wie weiter?

- Kannst du es noch so umbauen, dass die Figur nach einer bestimmten Zeit von selbst das Kostüm wechselt?
- Kannst du noch andere Interaktionsmöglichkeiten einbauen, z.B. dass man die Geschwindigkeit, mit der sich die Figur bewegt, verändern kann?

Worum geht es?


Es gibt viele interaktive computerbasierte Kunstwerke, wie z.B. *Bubbles* von Wolfgang Münch und Kiyoshi Furukawa (<https://zkm.de/en/artwork/bubbles>). Wir haben es hier als Seifenblasenspiel mit Scratch nachgebaut. Wie viele Seifenblasen kannst du in 30 Sekunden zum Platzen bringen?

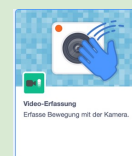
Für dieses Projekt brauchst du eine Webcam.







Wie funktioniert es?


1. Video-Erfassung als Erweiterung in Scratch laden


Für dieses Projekt benötigst du ein Notebook, PC oder Tablet mit Webcam. Scratch kann Bewegungen im Webcam-Bild erkennen. Erstelle ein neues Scratch-Projekt und klicke unten links auf . Im nächsten Menü kannst du verschiedene Erweiterungen auswählen. Für dieses Beispiel benötigst du die Erweiterung Video-Erfassung.



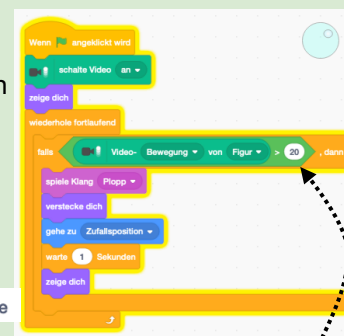
2. Seifenblasen platzen

Zeichne eine neue Figur mit  . Du kannst zum Beispiel mit dem Kreiswerkzeug  zwei Kreise ineinander zeichnen. Verkleinere die Seifenblase eventuell etwas, damit sie gut passt. Schreibe dazu eine kleinere Zahl in : 

Programmiere die Seifenblase so, dass sie immer dann, wenn eine Bewegung nah an der Seifenblase registriert wird (z. B. wenn du deine Hand bewegst, um die Seifenblase zum Platzen zu bringen), ein Geräusch macht, verschwindet und an einer anderen Stelle wieder auftaucht. Erstelle das Skript rechts für die Seifenblase. Dazu findest du die Blöcke jeweils im gleichfarbigen Bereich. Den Befehl  findest du unter den neu hinzugefügten Blöcken bei .

Starte das Programm mit  und erlaube Scratch, die Webcam zu benutzen.


Kannst du die Seifenblase mir deiner Hand zum Platzen bringen?



Hinweis: Mit dieser Zahl (0-100, hier 20) kannst du einstellen, wie sensibel die Bewegungserkennung reagiert. Experimentiere damit, bis die Einstellung passt.

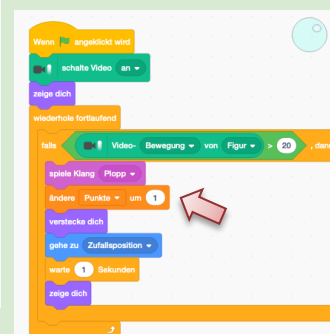
3. Timer und Punkte sammeln

Damit das Spiel interessant wird, kannst du noch *Punkte zählen* und einen *Timer* einbauen.

Dafür brauchst du zwei Variablen, diese erstellst du unter **Variablen** mit dem Block . Die Variablen kannst du dann so verwenden, wie unten im Bild gezeigt.



Erklärung: Wenn man die grüne Fahne anklickt, fängt das Spiel neu an. Am Anfang jedes Spiels ist es wichtig, dass die Variable *Punkte* wieder auf 0 gesetzt wird und die Variable *Timer* die Zeit angibt, die man zur Verfügung hat (in diesem Fall 30 Sekunden). Das nennt man «initialisieren». Ausserdem muss der Timer die Zeit herunterzählen, deshalb wird immer eine Sekunde gewartet und dann der Timer um -1 geändert. Das passiert bis der Timer bei 0 ist, denn dann ist das Spiel vorbei und das Skript wird gestoppt. Um die Punkte zu zählen, musst du ausserdem im Skript für die Seifenblase (hier ganz rechts) ergänzen, dass jedes Mal, wenn du diese berührst, die Punkte um 1 geändert werden.



4. Seifenblasen duplizieren

Wähle die Blase unter «Figuren» aus und klicke mit der rechten Maustaste darauf.

Wähle «Duplizieren», um eine Kopie zu erstellen. Du kannst sie auch anders einfärben.

Es ist wichtig, dass das Skript der ersten Seifenblase korrekt ist, teste es vorher.



Beispiele

Bubble

<https://scratch.mit.edu/projects/994256428/>

Fangspiel «Müll im Meer»

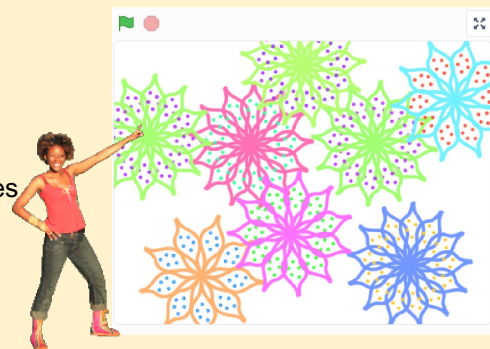
scratch.mit.edu/projects/514265508/

Wie weiter?

- Kannst du noch einbauen, dass es andere Objekte (zum Beispiel spitze Objekte) gibt, die dir Punkte abziehen, wenn du sie fängst oder andere Objekte, die unterschiedlich viele Punkte wert sind?
- Wie könntest programmieren, dass nach Ablauf der Zeit «Game Over» erscheint und du deinen Punktestand ablesen kannst?

Worum geht es?


Jeder Klick erweckt eine neue Blume zum Leben, und du kannst deine künstlerische Vision durch die Kombination verschiedener Farben und Formen ausdrücken. Tauche ein in die Welt der Kreativität und entdecke, wie Scratch es dir ermöglicht, Kunst zu schaffen und gleichzeitig Grundlagen der Programmierung zu erlernen.



Inspiziert von: Blumen blühen von Michel Hauswirth.

Wie funktioniert es?





1. Malstift-Erweiterung auf Scratch hinzufügen

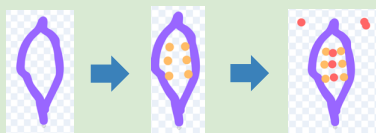
Als erstes solltest du unten links unter Erweiterungen  den Malstift hinzufügen.

Jetzt kannst du in Scratch einen Malstift programmieren, der dir bei deinem Kunstprojekt helfen wird.



2. Grundform / Blütenblatt zeichnen

Zeichne zuerst die Grundform (ein Blütenblatt) deiner Blumen.


Gehe auf *Figur malen*  und zeichne mit dem Pinsel  die Form deines Blütenblattes, wähle dann eine andere Farbe  aus und füge Punkte und andere Verzierungen hinzu. Achte darauf, dass der Blütenblattstiel im Mittelpunkt der Zeichenfläche  endet.




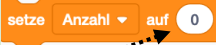
2. Variable erstellen

Jetzt musst du die Variable Anzahl Blütenblätter definieren. Dazu erstellst du in der Blockpalette unter Variablen  eine neue Variable  mit dem Namen «Anzahl». Damit definierst du, wie viele Blütenblätter deine Blumen haben werden.


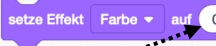
3. Skript erstellen: überall Blumen blühen lassen

Als nächstes wollen wir die Blumen überall dort blühen lassen, wo sich der Mauszeiger befindet. Dazu setzen wir die Skripte wie unten gezeigt zusammen. Als erstes nehmen wir den Befehl .

Wenn die Leertaste gedrückt wird, soll dort, wo der Mauszeiger gerade ist, eine Blume erscheinen. Damit die Blumen auch immer anders aussehen, stellen wir die «Anzahl» der Blütenblätter immer als Zufallszahl zwischen 6 und 15 ein, das machen wir mit den folgenden Blöcken:

Zufallszahl von 1 bis 10   **setze** Anzahl **auf** 0

Ausserdem sollen die Blumen immer unterschiedliche Farben haben:

Zufallszahl von 1 bis 10   **setze** Effekt **Farbe** **auf** 0

Und die Blütenblätter sollen überall im Kreis zu sehen sein, deshalb wird 360 Grad durch die Anzahl der Blätter geteilt.

```

Wenn Taste Leertaste gedrückt wird
  gehe zu Mauszeiger
  setze Anzahl auf Zufallszahl von 6 bis 15
  setze Effekt Farbe auf Zufallszahl von 1 bis 200
  wiederhole Anzahl mal
    hinterlasse Abdruck
    drehe dich um 360 / Anzahl Grad
  
```

```

Wenn angeklickt wird
  verstecke dich
  lösche alles
  
```

Letztendlich brauchen wir ein Skript, das die Wisen am Anfang leert.

Beispiele

Blumen blühen lassen

<https://scratch.mit.edu/projects/997748438/>

Originalprojekt: Blumen blühen

<https://scratch.mit.edu/projects/734403507/>

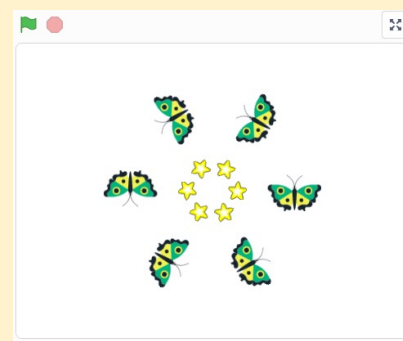
Wie weiter?

- Kannst du Blumen in verschiedenen Grössen wachsen lassen (Tipp: verwende dazu den Block «ändere Grösse um»)?
- Kannst du ganze Blumen mit Stiel wachsen lassen (beachte, dass es immer viele Blütenblätter, aber nur einen Stiel gibt)?

Worum geht es?


Du kannst Gegenstände so kombinieren, dass faszinierende Kunstwerke aus geometrischen Formen entstehen. Du programmierst deine eigenen Kunstwerke, indem du geometrische Muster entwirfst und sie zum Leben erweckst. Du kannst verschiedene Formen, Farben und Bewegungen verwenden, um einzigartige Kunstwerke zu schaffen, die deine kreative Vision widerspiegeln.

Das Projekt basiert auf dem Material *Entdecke mathematische Konzepte durch Informatik und Programmierung* von Michel Hauswirth.



Wie funktioniert es?

1 Figur und Kostüm wählen

Entscheide, aus welchen Elementen dein Muster bestehen soll. Erstelle ein neues Scratch-Projekt. Wähle und suche dir eine Figur unter **Figur wählen**  aus der Bibliothek aus. Oben links unter Kostüme kannst du dann verschiedene Kostüme auswählen (d.h. wie deine Figur aussehen soll).

2. Muster Programmieren

Nun kannst du dein schönes Muster programmieren.

Zuerst verwendest du den Block **zeige dich**, um die Figur sichtbar zu machen.

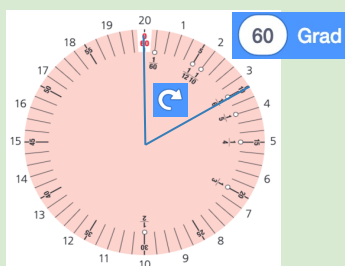
Mit dem Block **gehe zu x: 0 y: 0** bestimmst du, wo das Muster beginnt.

Jetzt programmierst du für jedes Kostüm eine Schleife mit dem Block "wiederholen () mal". In jeder Schleife bewegt sich deine Figur, klonst sich und dreht sich weiter.

Dann macht sie das Gleiche mit dem nächsten Kostüm, aber der Abstand zur Mitte ist größer.

Am Ende, wenn das Muster fertig ist, soll

sich die Figur verstecken bzw. nicht mehr sichtbar sein. Dies ist das Skript:



Erklärung:

Eine volle Umdrehung entspricht 360 Grad, d. h. damit sich die Figur einmal im Kreis dreht, muss die Zahl im Block «Drehe dich um () Grad» multipliziert mit der Anzahl der Wiederholungen der Schleife insgesamt 360 ergeben. In unserem Fall dreht sich die Figur 6 mal um je 60 Grad, also $6 \times 60 = 360$.

3. Erweiterung: Weitere Muster

Wenn du die Anzahl der Wiederholungen und den Drehwinkel anpasst, kannst du auch andere schöne Muster machen, wie das rechts. Das Muster rechts kannst du durch 12 Wiederholungen der Schleife mit einem Winkel von 30 Grad erzeugen.



Beispiele

Schöne Muster – Grundprogramm 1
scratch.mit.edu/projects/411947145

Schöne Muster- Grundprogramm 2
<https://scratch.mit.edu/projects/725198061/>

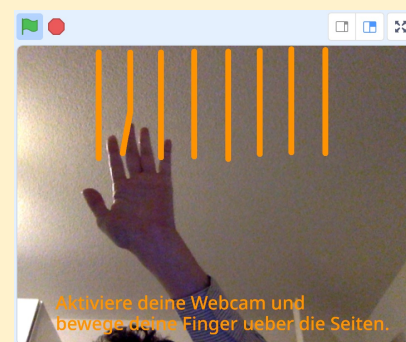
Wie weiter?

- Kannst du noch andere Muster machen? Zum Beispiel noch mehr Kostüme? Oder mit anderen Winkeln?
- Kannst du dein Programm so erweitern, dass es nach der Anzahl der Umdrehungen fragt und dann ein Muster erstellt?

Worum geht es?

Mit Scratch lassen sich sehr leicht eigene Musikinstrumente oder Soundeffekte erstellen. Scratch hat bereits viele Töne und Sounds eingebaut und über die Aufnahme-funktion lassen sich auch ganz einfach eigene Geräusche hinzufügen.

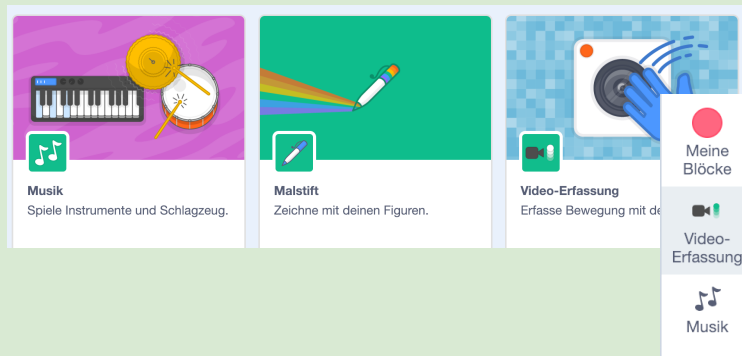
In diesem Beispiel baust du ein Klavier zum Anklicken und eine Harfe zum Spielen mit den Händen mit Hilfe einer Webcam.



Wie funktioniert es?

1. Erweiterungen in Scratch laden

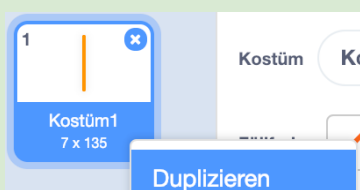
Für dieses Projekt benötigst du einen Notebook, PC oder Tablet mit Webcam. Scratch kann Bewegungen im Webcam-Bild erkennen. Erstelle ein neues Scratch-Projekt und klicke unten links auf . Im nächsten Menü kannst du verschiedene Erweiterungen auswählen. Jede Erweiterung stellt neue Befehle bereit. Für dieses Beispiel benötigst du die Erweiterung *Musik* und *Video-Erfassung*. Lade beide Erweiterungen nacheinander.



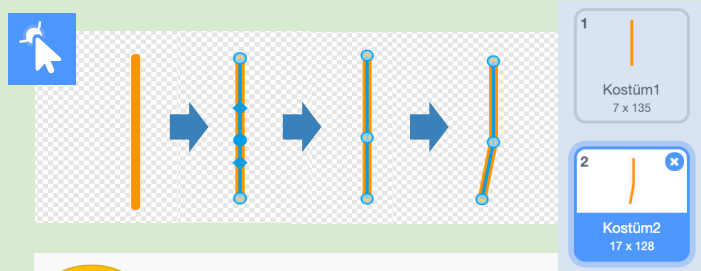
2. Ein Webcam-Harfe bauen

Erstelle eine neue Figur mit und zeichne mit eine gerade Linie in deiner Wunschfarbe.

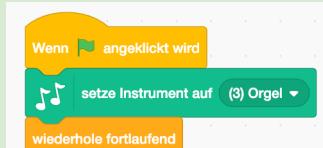
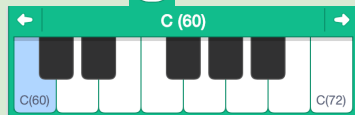
Klicke Kostüm1 mit der rechten Maustaste an, um es zu Duplizieren:



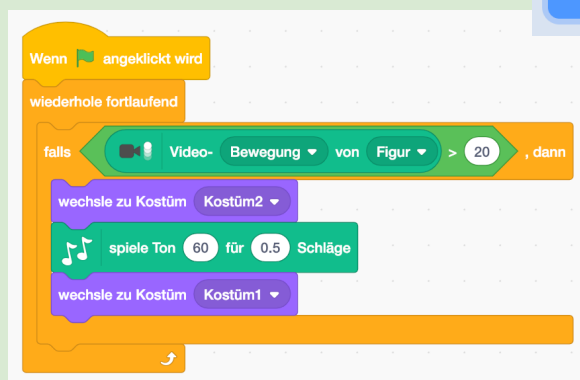
Kostüm2 soll nun eine leicht abgeknickte Linie zeigen. Verwende das Werkzeug . Klicke einmal in die untere Hälfte der Linie, um einen neuen Punkt hinzuzufügen. Ziehe nun am unteren Punkt wie rechts im Bild gezeigt:



Erstelle ein Script wie rechts gezeigt und dupliziere danach die Figur mehrmals. Ordne die Saiten auf der Bühne an. Stelle für jede Harfensaite den passenden Ton ein, indem du die kleine anklickst.



Solche Einstellungen an Blöcken nennt man auch **Parameter**. Du kannst auch jeder Figur ein anderes Instrument zuteilen. Starte mit und gestatte Scratch die Webcam zu verwenden.



Beispiele

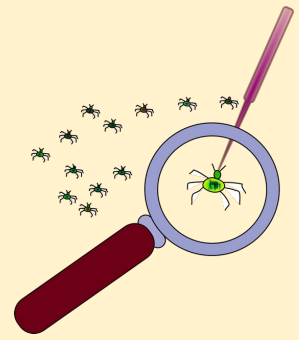
Klavier
scratch.mit.edu/projects/151957886
 Webcam Harfe
scratch.mit.edu/projects/151145571
 Webcam Schlagzeug
scratch.mit.edu/projects/151148401

Wie weiter?

- Kannst du statt der eingebauten Töne für jede Taste ein eigenes Geräusch oder Laut mit dem Mikrophon () aufnehmen?
- Auf www.auditorix.de findest du in der Geräusche-Box Soundeffekte zum Herunterladen, die du für dein Scratch Projekt verwenden kannst. Kannst du eine Geschichte mit Hilfe von Soundeffekten erzählen?

Worum geht es?

Recht häufig kommt es vor, dass ein Computer-Programm beim ersten Test nicht das macht, was es sollte. In den seltensten Fällen ist dann der Computer schuld. Meistens hat der Mensch einen Denkfehler oder eine Unaufmerksamkeit begangen und muss nun den Fehler suchen. Man nennt das Debugging. 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

Welche Figur ist gerade ausgewählt? Hast du das Skript bei der richtigen Figur erstellt?

2. Vergessene Blöcke

Gehe für dich in Gedanken die einzelnen Blöcke durch. Ergeben sie Sinn oder fehlt etwas? Falls du einen Code abschreibst: Zähle die Blöcke nach. Vielleicht kannst du deinen Code auch mit jemandem vergleichen.

3. Verwechselte Blöcke

Hast du die richtigen Blöcke verwendet? Besonders häufig werden Blöcke mit relativen und absoluten Anweisungen verwechselt:



Auch Blöcke mit ähnlichen Anweisungen können schnell verwechselt werden:



Vielleicht haben die Blöcke aber auch nicht die richtige Farbe und damit eine andere Funktion:



4. x und y verwechselt

Unter Umständen hast du einen X-Block statt einen Y-Block erwischt?



5. Falsche Zahlen

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



6. Falsche Auswahl

Manchmal geht das Ändern der Dropdown-Menüs vergessen:



7. Falsche Reihenfolge

Sind die Blöcke in der korrekten Reihenfolge?



8. Zeitliche Begrenzung

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



9. Schleifen

Sind Blöcke innerhalb von Schleifen, wenn sie ausserhalb sein sollten oder umgekehrt?

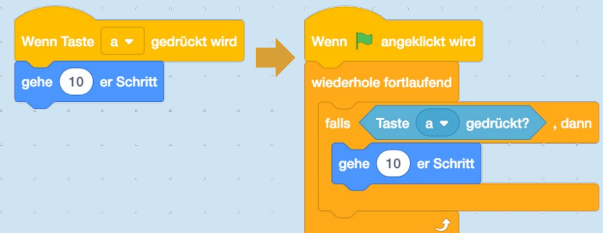


Fehlt eine Schleife, die etwas wiederholt prüft?



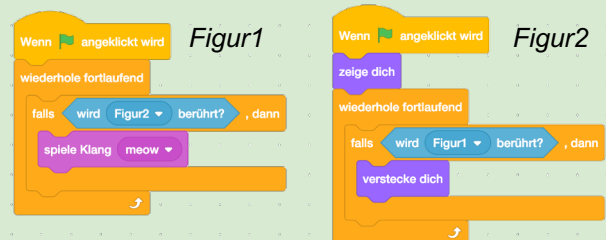
10. Tastatursteuerung einfach oder schnell

Beim Programmieren gibt es häufig verschiedene Lösungen für das gleiche Problem. In Scratch gibt es fertige Blöcke, die fortlaufend, aber langsam, auf Ereignisse prüfen. Diese kann man auch mit einer Schleife und Bedingung selbst nachbauen. Für die Steuerung einer Spielfigur sind diese meist besser geeignet, da sie häufiger prüfen und damit «flüssiger» wirken.



11. Koordination und Synchronisation

Wenn mehrere Skripte gleichzeitig laufen, kann es zu Konflikten kommen, da nicht klar ist, welches Skript zuerst abgearbeitet wird (Beispiel rechts). Es kann helfen, eine kleine Pausen einzufügen, bevor eine Figur versteckt wird. Besser noch ist es, mit den *sende-empfange*-Blöcken zu arbeiten.



12. Lokale oder globale Variable

Kannst du bei einer Figur eine Variable nicht einsetzen, hast du sie vermutlich als lokale Variable definiert („nur für diese Figur“). Ein Punktezähler sollte zum Beispiel „für alle Figuren“ angelegt werden.



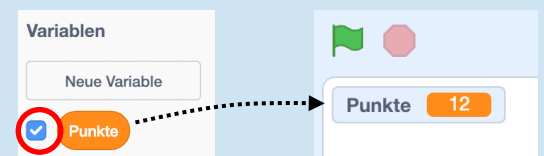
13. Initialisieren

Beim Starten deines Programms über das grüne Fähnchen müssen eventuell Einstellungen zurückgesetzt werden: Den Punktestand wieder auf 0 setzen, Figuren wieder anzeigen oder verstecken, Malspur wegwischen usw. Bei Geschichten müssen zum Beispiel häufig alle Figuren beim Start versteckt werden.



14. Werte/Zustände anzeigen

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



Tipp: Figur verschwunden?

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



Worum geht es?

Diese Broschüre bietet eine Reihe vielfältiger Projektideen für Scratch. Dabei werden jeweils eine Kernidee und einige Skript-Blöcke gezeigt, die ein erstes Erfolgserlebnis ermöglichen. Nach dieser Starthilfe bleibt jedoch sehr vieles offen und somit der individuellen Gestaltung überlassen. Die Broschüre eignet sich damit insbesondere für Projektarbeiten, wenn Scratch als Umgebung bereits bekannt ist und grundlegende Konzepte erarbeitet wurden.

Was lässt sich damit lernen?

Mit Scratch lassen sich Kompetenzen in den Bereichen Medien und Informatik, Technisches Gestalten, Natur und Technik, Musik, Bewegung und Sport sowie überfachliche Kompetenzen wie Teamfähigkeit und Sozialkompetenz erwerben.

Warum Scratch?

Scratch ist eine am MIT entwickelte, frei verfügbare Programmierumgebung für Kinder und Jugendliche nach dem «low floor – wide walls – high ceiling»-Prinzip. Es existiert bereits viel Material sowie unter scratch.mit.edu über 60 Millionen einseh- & änderbare Programme (Stand April 2021).

Was wird benötigt?

Scratch läuft seit der Version 3 auf allen Computern (Desktops, Notebooks, Tablets) mit einem HTML5-fähigen Browser und Internetverbindung.

<http://scratch.mit.edu/>

Möchtest du Scratch-Projekte lieber offline erstellen und speichern? Es gibt auch eine Scratch-App: <https://scratch.mit.edu/download>

Das «Low floor - wide walls - high ceiling»-Prinzip

Scratch versucht das «low floor - wide walls - high ceiling»-Prinzip umzusetzen:

- **Low floor / Leichter Einstieg:** Für erste Projekte sind keine Vorkenntnisse nötig. Erste Erfolge und Aha-Erlebnisse nach fünf Minuten!
- **Wide walls / Verschiedenste Zugangsweisen:** Mit Scratch lassen sich ganz unterschiedliche Projekte realisieren – animierte Geschichte, Simulationen, Spiele usw. Scratch-Projekte verbinden Kompetenzen unterschiedlichster Fachbereiche wie Mathematik, Deutsch, Musik, NMG und natürlich Medien & Informatik.
- **High ceiling / Nach oben offen:** Mit Scratch lassen sich auch komplexere Projekte umsetzen. Zudem lassen sich vielfältige Sensoren, Roboter und andere Elektronik mit Scratch verbinden. Damit lässt sich die Funktionspalette praktisch beliebig erweitern.

Snap!

Für fortgeschrittene Lernende eignet sich *Snap!*. Diese ebenfalls blockbasierte Programmierumgebung baut auf Scratch auf, erlaubt aber die Programmierung komplexerer und abstrakterer Projekte. Snap! läuft direkt im Browser unter:

<https://snap.berkeley.edu/>



Die Idee des Konstruktivismus

Scratch wurde mit der Idee des Konstruktivismus des Mathematikers und Piaget-Studenten Seymour Papert im Hinterkopf entwickelt. Die Idee des Konstruktivismus baut auf dem Konstruktivismus auf und geht davon aus, dass Menschen besonders dann etwas lernen, wenn sie etwas mit persönlicher Bedeutung selbst konstruieren und sich im Konstruktionsprozess das dafür nötige Wissen aneignen und anwenden. Das eigene aktive Handeln steht im Mittelpunkt um auch theoretische Überlegungen und Modelle konkret fassbar und begreifbar zu machen. Mehr zum Konstruktivismus erfährt man in Paperts Buch: *Mindstorms. Children, Computer and Powerful Ideas, Basic Books, New York, 1980*

Impressum / Kontakt

Download dieser Broschüre:

<http://link.phsz.ch/challenge>

Weitere Broschüren:

<http://iLearnIT.ch/broschueren>

Pädagogische Hochschule Schwyz
Giulia Paparo

giulia.paparo@phsz.ch