pädagogische hochschule schwyz

Scratch Projektideen

Michael Hielscher
Beat Döbeli Honegger

3. Februar 2019

Scratch Projektideen

Scratcher werden

phsz

1

Worum geht es?

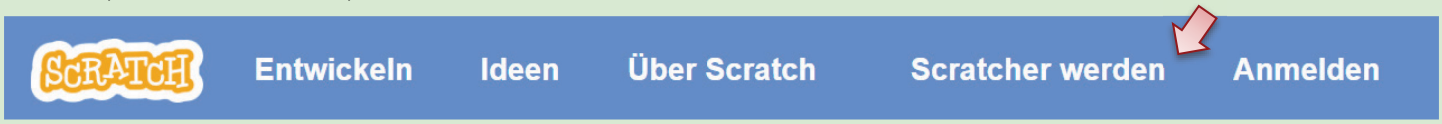
Scratch ist eine kostenlose Programmierumgebung für Kinder, die am MIT entwickelt wird. Scratch ist leicht zu erlernen und eignet sich sowohl für einfache Animationen als auch für komplexe Spiele und Simulationen. Du kannst mit wenigen Klicks eigene Bilder und Töne erstellen und damit deine ganz persönlichen 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

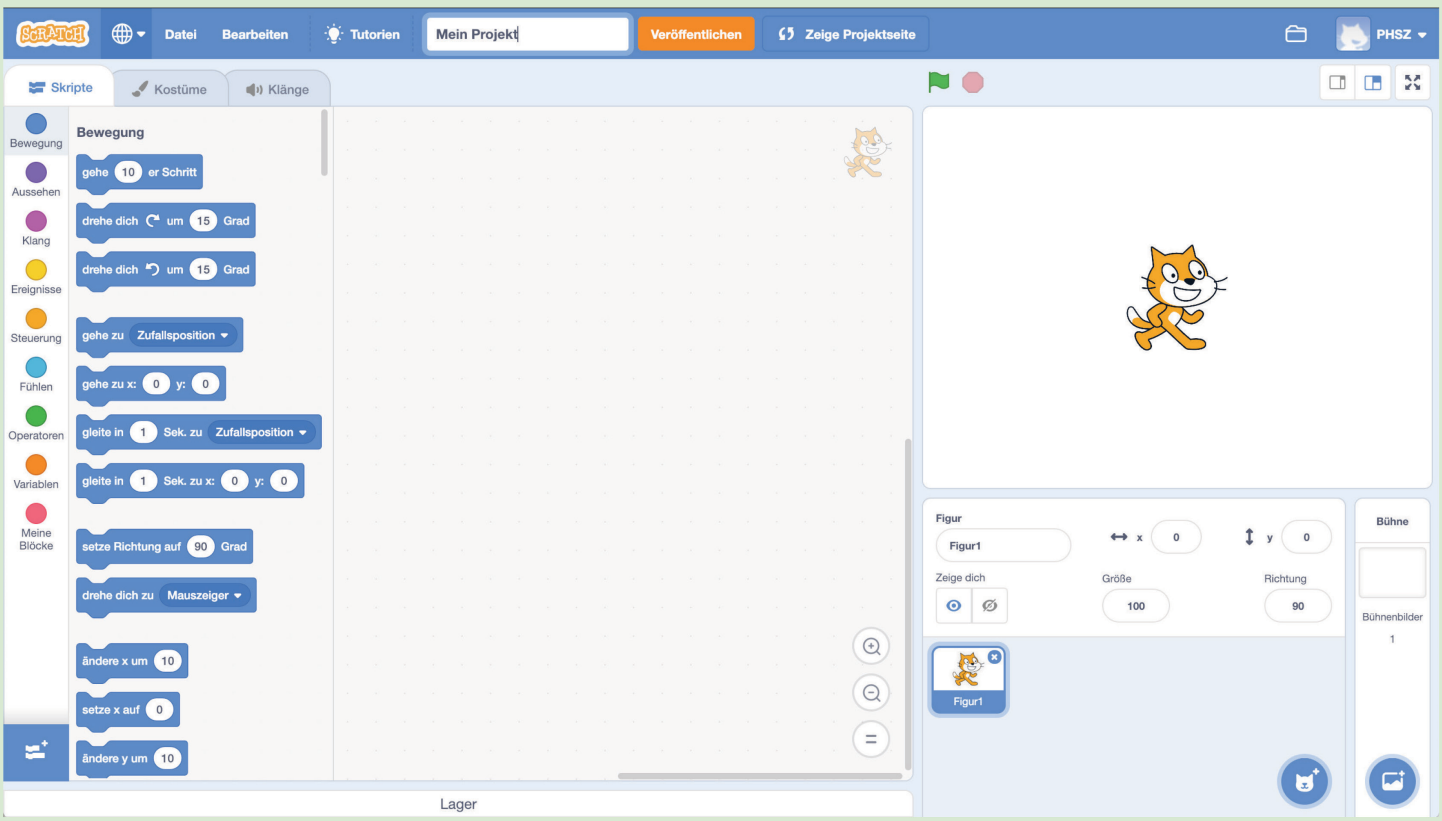
Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox. Gehe auf https://scratch.mit.edu
Es lohnt sich ein Scratch-Konto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts und erstelle dir einen eigenen Account, oder melde dich an, wenn du schon einen Account besitzt.



Du benötigst die folgenden Angaben: Einen Benutzernamen (wähle etwas Phantasievolles), ein Passwort, dein Geburtsdatum und eine Emailadresse.

2. Ein erstes Scratch-Projekt erstellen

Klicke auf der Startseite von Scratch auf „Entwickeln“ um ein neues Scratch-Projekt zu beginnen. Wähle 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.



Pair-Programming

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 ProgrammiererInnen 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 das 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 ins Keyboard/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

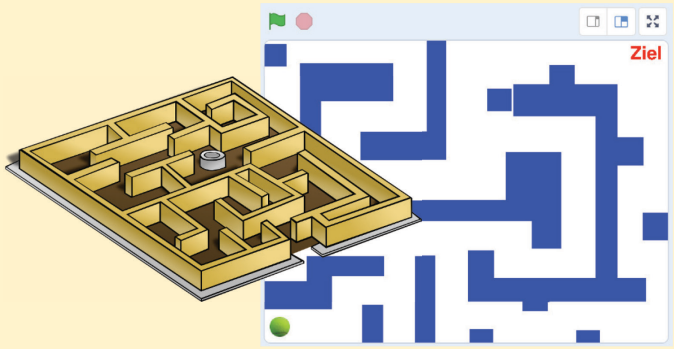
Hinweis

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

Labyrinth

Worum geht es?

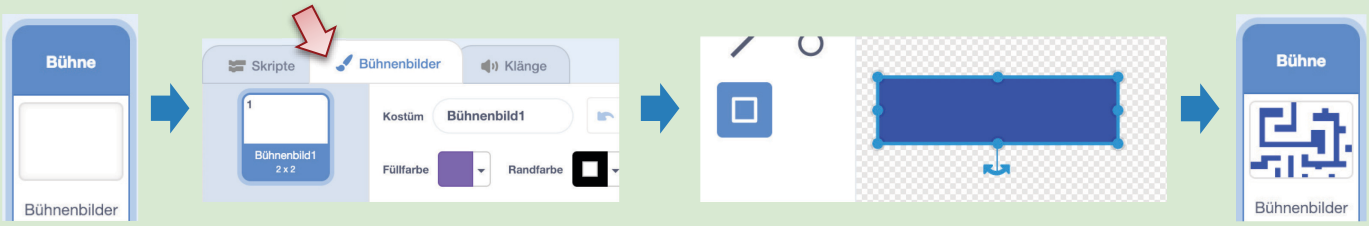
Mit Scratch lassen sich ganz einfach Geschicklichkeitsspiele erstellen, bei denen ein Objekt mit der Maus bewegt werden muss. Bei diesem Spiel geht es darum einen Tennisball ohne anzustossen durch ein Labyrinth zu navigieren. Du entscheidest, wie einfach oder schwierig das wird ☺
Probiert gegenseitig eure erstellten Labyrinthe aus.



Wie funktioniert es?

1. Ein Labyrinth mit Scratch zeichnen

Klicke rechts unten auf „Bühne“ und anschliessend auf „Bühnenbilder“ um das das Labyrinth zu zeichnen.
Wähle eine Füllfarbe aus und verwende das Rechteck-Werkzeug, um die Teile deines Labyrinths zu zeichnen.



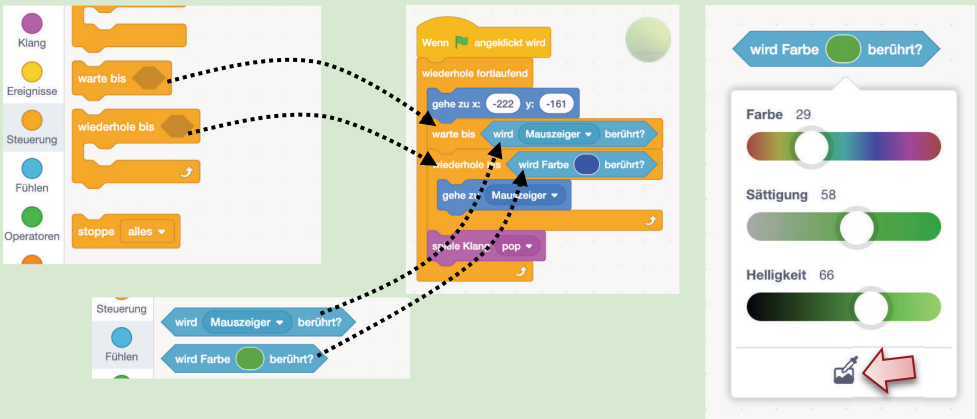
Du kannst dein Labyrinth jederzeit überarbeiten und mehr oder weniger viele Hindernisse einbauen. Verwende jedoch immer die gleiche Farbe für alle Wände.

2. Spielfigur und Programmierung

Klicke auf Figur wählen unten rechts und suche den Tennisball Tennis Ball in der Bibliothek heraus. Die Katze benötigen wir in diesem Spiel nicht. Du kannst sie mit dem kleinen blauen „Löschen“. Bewege den Tennisball auf der Bühne an die Startposition deines Labyrinths (z. B. eine Ecke des Spielfelds).

Verkleinere den Ball eventuell ein wenig, damit er zu deinem Labyrinth passt. Trage dazu eine Zahl ein:

Erstelle für den Tennisball das rechte Skript. Die Blöcke findest du jeweils im gleichfarbigen Bereich der Blockplatte.
Wenn du fertig bist, klicke auf , um das Spiel zu starten. Der Ball wartet, bis er mit der Maus berührt wird (warte bis wird Mauszeiger berührt?) und wird anschliessend der Maus folgen (gehe zu Mauszeiger), bis ausversehen eine Wand berührt wird (wiederhole bis wird Farbe berührt?).
Bewege den Tennisball vorsichtig bis zur anderen Seite des Labyrinths.



Hinweis: Die Farbe kannst du wählen, indem du einmal den kleinen farbigen Bereich klickst. Um deine Labyrinth-Farbe auszuwählen klicke auf die Pipette ganz unten.

Beispiele

Labyrinth-Spiel
scratch.mit.edu/projects/154416386

Whac-A-Mole Spiel
scratch.mit.edu/projects/151083755

Wie weiter?

- Kannst du ein Zielfeld hinzufügen, welches bei Berührung mit dem Tennisball einen Soundeffekt abspielt?
- Kannst du eine Stoppuhr einbauen, wie lange man gebraucht hat?
- Kannst du zusätzliche Figuren (in Wandfarbe) als Hindernisse einbauen, die sich selbstständig hin und her bewegen?

Scratch Projektideen

Ritter, Löwen und Prinzessin

phsz

4

Worum geht es?

Du bist der Regisseur! Wer kommt in einer Szene auf die Bühne und wer sagt was? Mit Scratch kannst du kleine Filme oder Animationen erstellen. Die Figuren und Hintergründe kannst du programmieren, sodass sie sich nach deinen Idee bewegen und sprechen. Vom kurzen Witz bis zum kompletten Theaterstück ist alles möglich.

Wie funktioniert es?

1. Schreib deine eigene Geschichte

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

Szene 1: In der Wüste

Der Ritter erscheint von links auf der Bühne und läuft hin und her. Der Ritter sagt: „Ich bin müde und muss mich ausruhen." Der Ritter setzt sich unter einen Kaktus, schläft ein und schnarcht laut vor sich hin.

Der Löwe erscheint von rechts auf der Bühne und läuft zur Bühnenmitte. Der Löwe bemerkt den Ritter, seufzt und sagt „Mist, schon wieder Konservenfutter!". Der Löwe läuft schnell zum Ritter und frisst ihn auf (Ritter verschwindet).

Szene 2: Im Schloss

Die Prinzessin erscheint auf der Bühne und sagt: „Wo sind nur alle meine Ritter hin? Ich sollte nachschauen gehen."

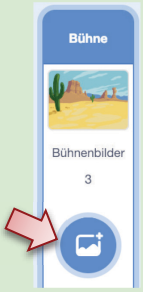
Die Prinzessin läuft rechts aus dem Bild und verschwindet von der Bühne.

2. Bilder und Töne zusammenstellen

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

Ist kein passendes Bild 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 Klänge kannst du passende Geräusche für jede Figur auswählen oder selbst aufnehmen ().

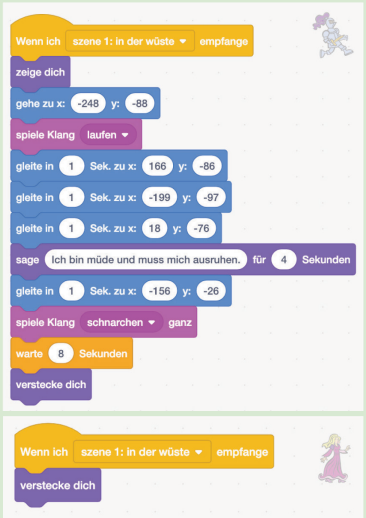


3. Übernimm die Regie

Wähle unten rechts die „Bühne" aus. Der Regisseur 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.

Regie-Skript (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 Prinzessin nicht mit (Regiebuch).



Beispiel Skripte von Ritter, Löwe und Prinzessin für Szene 1: In der Wüste

Beispiele

Einen Witz erzählen
scratch.mit.edu/projects/151205820

Der Ritter und der Löwe
scratch.mit.edu/projects/151409075

Wie weiter?

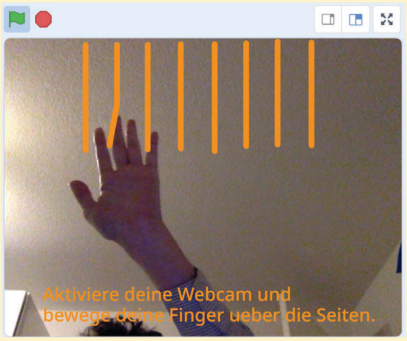
- Erstelle bei allen Figuren die Skripte für Szene 2 usw.
- Füge in der Bühne passende Hintergrundmusik zu jeder Szene hinzu.
- Kannst du die Geschichte „interaktiv" gestalten, sodass Zuschauer zum Beispiel Dinge anklicken oder sogar selbst steuern können?

Musikinstrumente mit Scratch

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 Aufnahmefunktion lassen sich auch ganz einfach eigene Geräusche hinzufügen.

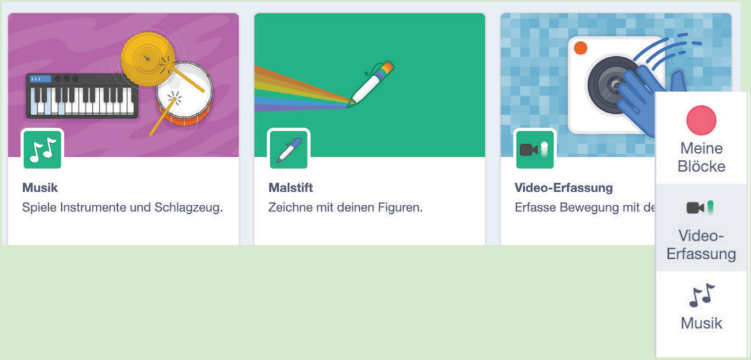
In diesem Beispiel baust du ein Klavier zum Anklicken und eine Harfe zum Spielen mit den Händen mit Hilfe einer Webcam.



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 klicken 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.

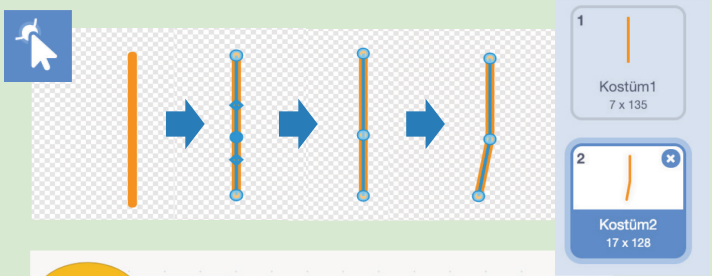


2. Ein Webcam-Harfe bauen

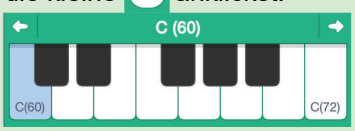
Erstelle eine neue Figur mit Malen und zeichne mit eine gerade Linie in deiner Wunschfarbe.

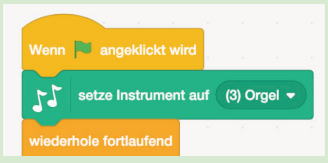
Klicke Kostüm1 mit der rechten Maustaste an, um es zu Duplizieren:

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:

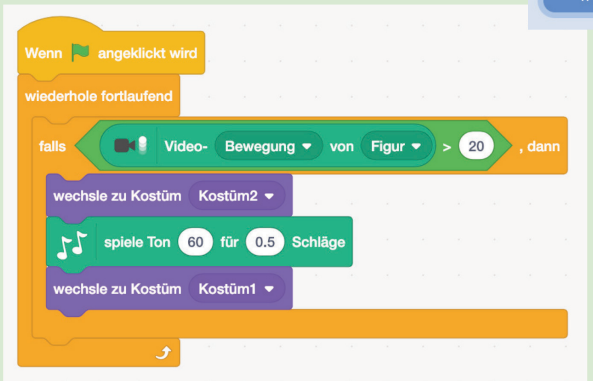


Erstelle ein Script wie rechts gezeigt und dupliziere danach die Figur mehrmals. Ordne die Saiten auf der Bühne an. Stelle für jede Harfensaite den passenden Ton ein, indem du die kleine 60 anklickst.





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

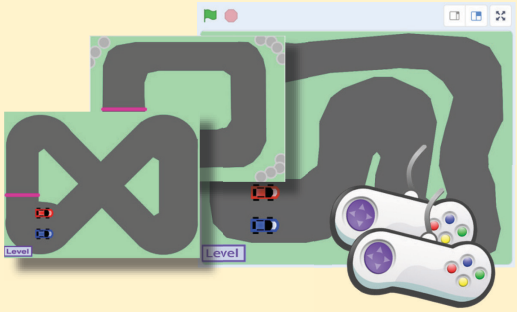
Wie weiter?

- Kannst du statt den eingebauten Töne für jede Taste ein eigenes Geräusch oder Laut mit dem Mikrophone ( ) 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?

Autorennen für Zwei

Worum geht es?

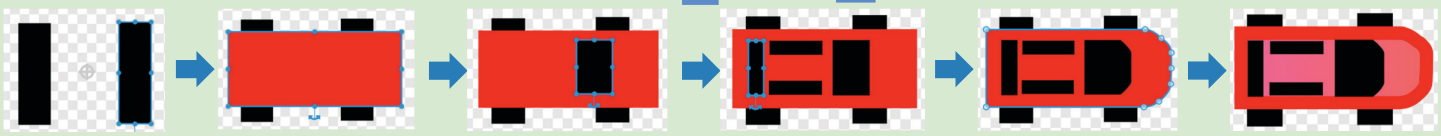
In vielen Computerspielen kann man mit zwei oder mehr Spielern zusammen spielen. In der einfachsten Variante sitzen alle Spieler gemeinsam vor einem Bildschirm und jede Spielfigur lässt sich einzeln steuern (z.B. durch bestimmte Tasten auf der Tastatur).



Wie funktioniert es?

1. Rennstrecke und erstes Auto erstellen

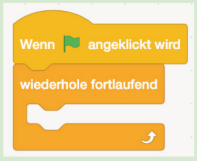
Zeichne ein Bühnenbild mit einer Rennstrecke. Verwende eine Farbe als Begrenzung der Fahrbahn (z.B. Grasgrün). Erstelle eine neue Figur mit Malen für das erste Auto. Zeichne das Auto aus der Vogelperspektive. Zoome 2-3 Mal ins Bild hinein. Zeichne das Auto mit dem ▢ und dem Werkzeug genau auf den Mittelpunkt ⊕.



Um uns Arbeit zu sparen, erstellen wir das zweite Auto erst am Schluss durch Duplizieren / Umfärben des ersten Autos. Die meisten Spiele verwenden eine sogenannte *Spielschleife*. Solange das Spiel läuft werden die Blöcke innerhalb der Schleife immer wieder abgearbeitet. In Scratch verwendet man dafür den „wiederhole fortlaufend"-Block und prüft auf bestimmte Bedingungen („falls ... dann"). Baue das folgende Programm für Auto1 nach:

2. Auf der Strecke bleiben

Klick auf ⚑ für eine Probefahrt mit den Pfeiltasten. Ein Spiel braucht eine Herausforderung und beim Autorennen geht es darum, auf der Strecke zu bleiben. Sobald das Auto die grüne Rasenfläche berührt, hat der Spieler verloren und das Auto soll wieder an den Start gestellt werden. Füge einen „falls ... dann" Block wie im Bild unten rechts hinzu. Wähle die Farbe der Rasenfläche von der Bühne aus.

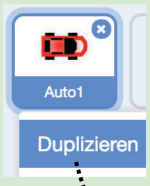


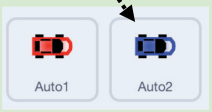
Die Zahlen 5 und -2 geben an, wie schnell sich das Auto bewegt und dreht. Experimentiere mit diesen Werten.

Beachte: Beim Rückwärtsfahren sind die Drehrichtungen vertauscht.

3. Spiel für Zwei

Wähle das erste Auto unter „Figuren" aus und klicke mit der rechten Maustaste darauf. Wähle „duplizieren" um eine Kopie zu erstellen. Bearbeite das Kostüm und lackiere das zweite Auto in einer anderen Farbe. Wähle andere Tasten in den hellblauen Blöcken aus (z.B. W,A,S und D). Starte das Spiel mit ⚑ neu und teste die Bewegung des zweiten Autos.





Beispiele

Autorennen für zwei Personen
scratch.mit.edu/projects/150489552

Stern ausweichen
scratch.mit.edu/projects/135827798

Wie weiter?

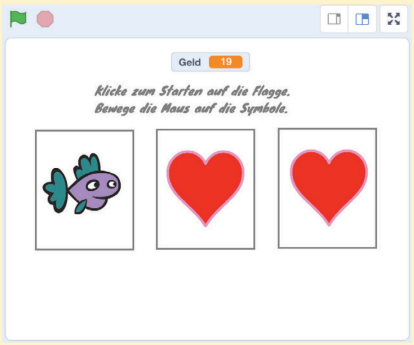
- Kannst du eine Ziellinie einfügen die beim Überfahren einen passenden Soundeffekt auslöst?
- Kannst du einen Rundenzähler für jedes Auto einbauen?
- Kannst du verschiedene, auswählbare Rennstrecken einbauen?
- Kannst du einen Computergegner erstellen, der selbstständig fährt?

Slot-Machine

Worum geht es?

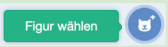
Scratch eignet sich hervorragend, um kleine Spiele zu entwickeln. Viele Spiele verwenden den „Zufall“, zum Beispiel in Form eines Würfels, damit das Spiel nicht vorhersehbar wird und spannend bleibt. Scratch bietet einen Befehlsblock für Zufallszahlen an, um zum Beispiel eine Zahl zwischen 1 und 6 zu würfeln.

Wir bauen in diesem Beispiel das Spiel Slot-Machine, bei dem der „Zufall“ eine wichtige Rolle spielt.

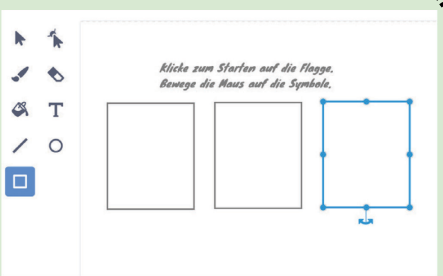


Wie funktioniert es?

1. Slots und Hintergrundbild erstellen

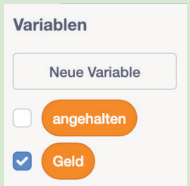
Erstelle ein neues Scratch-Projekt. Füge eine Figur mit 

hinzu und suche aus der Bibliothek ein erstes Bild heraus - zum Beispiel den Apfel. Benenne die Figur „Slot1“. Wechsel auf Kostüme und lade einige weitere Bilder dazu. Passe die Grösse der einzelnen Bilder so an, dass sie alle etwa gleich gross sind.
Erstelle ein neues Bühnenbild und zeichne drei gleich grosse Rechtecke:



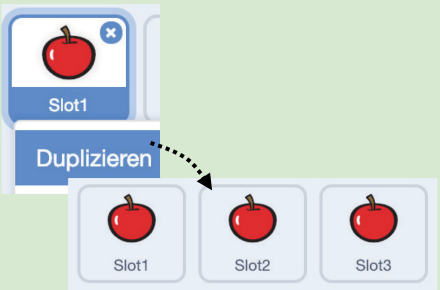
2. Slots programmieren

Erstelle zwei neue Variablen „Geld“ und „angehalten“ unter Variablen. Erstelle das folgende Skript für die Figur Slot1 :



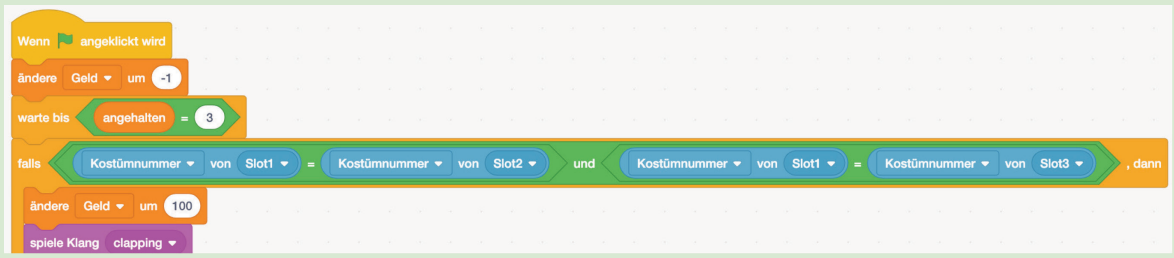
Bei „Zufallszahl“ musst du die Anzahl der Kostüme eintragen, die deine Figur verwendet. Damit wird ein zufälliges Startkostüm gewählt.

Klicke rechts auf Slot1 und wähle zwei Mal „Duplizieren“ aus, um Slot2 und Slot3 zu erstellen. Klick auf und fahre mit der Maus über die Slots.



3. Geld ausgeben und gewinnen

Jedes Spiel soll etwas Geld kosten – mit Scratch geht das mit „ändere Geld um -1“. Immer wenn ein Slot gestoppt wird, erhöht sich die Variable „angehalten“ um 1. Wir warten bis angehalten = 3 ist und prüfen dann, ob die Symbole der Slots gleich sind. Erstelle das folgende Skript für die Bühne:



Bau das Skript so aus, dass die Spieler auch etwas Geld für zwei gleiche Symbole bekommen.

Beispiele

Slot Machine
scratch.mit.edu/projects/167744761

Wie weiter?

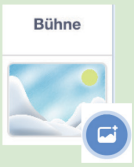
- Kannst du die Slots so anpassen, dass sie beim Berühren mit der Maus noch 2-3 Bilder weiterlaufen und dabei langsamer werden?
- Kannst du die Auswertung so umbauen, dass unterschiedliche Symbole unterschiedlich grosse Gewinne ergeben?
  (z.B. 2 Herzen sind besser als 2 Äpfel)

Pinguin Quiz

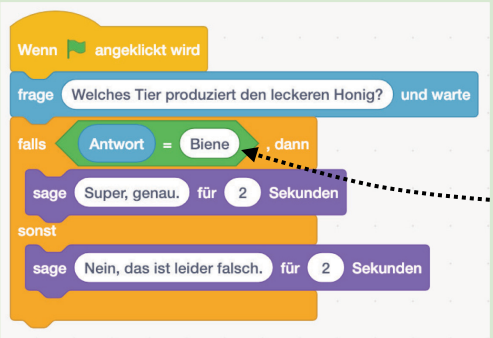
Worum geht es?

In Scratch-Programmen kann eine Figur eine Frage stellen, auf die man eine Antwort eingeben muss. Mit dieser Funktion können zum Beispiel Ratespiele erstellt werden. Kannst du ein spannendes Quiz zu deinem Lieblingsthema entwickeln?

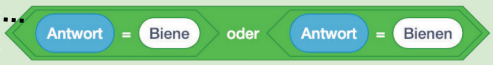
Wie funktioniert es?



1. Hintergrund und Figuren wählen und Quiz programmieren.
Erstelle ein neues Scratch-Projekt. Wähle Figur wählen und suche dir aus der Bibliothek eine Figur heraus. Wähle zu deiner Figur ein passendes Bühnenbild aus. Zum Beispiel den Pinguin und die Eislandschaft. Wähle deine Figur aus (blauer Rahmen) und erstelle ein Script für eine Quizfrage wie in folgendem Beispiel gezeigt:



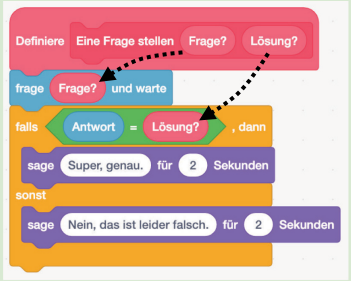
Erklärung: Wenn in Scratch ein hellblauer frage-Block verwendet wird, legt Scratch den eingegebenen Text in die Antwort Variable (Fühlen). Mit = 50 lassen sich zwei Werte miteinander vergleichen. Es wird also geprüft, ob „Biene“ als Antwort eingegeben wurde: Antwort = Biene. Du kannst mit dem ODER-Block auch mehrere Antworten als richtig werten:



Hinweis: Schiebe immer die linke obere Ecke eines Bausteins in das freie Feld. Nur wenn der Baustein einrastet, wird das Script richtig funktionieren.

Um eine weitere Spielfrage hinzuzufügen, klicke mit der rechten Maustaste auf den hellblauen frage-Block und „Dupliziere“ ihn. Alle Folgeblöcke werden mit kopiert und du musst nur noch die Frage- und Antwort-Felder neu ausfüllen.

3. Ein eigener Scratch-Block
Wenn du in Scratch mehrere Blöcke immer wieder verwendest, kannst du unter Meine Blöcke auch einen eigenen Block mit Neuer Block erstellen. Du kannst selbst festlegen, welchen Namen und welche Einstellungen dein neuer Block haben soll. Für das Quiz möchten wir einen Block Eine Frage stellen mit zwei Textfeldern für *Frage?* und *Lösung?* anlegen (2x). Schau dir das folgende Beispiel genau an und baue es nach. Beim Programmieren spricht man bei eigenen Blöcken auch von „Unterprogrammen“ und die Einstellungen bezeichnet man als „Parameter“, die an das Unterprogramm übergeben werden. Programmierer teilen ihre Programme häufig in kleine Unterprogramme auf, da sich diese einfacher wiederverwenden lassen und bei Änderungen nur jeweils an einer Stelle angepasst werden muss.



Hinweis: Die Variablen Frage? und Lösung? kannst du aus dem Definiere-Block ziehen.

Beispiele

Pinguin Quiz
scratch.mit.edu/projects/151391118

Mathe Quiz
scratch.mit.edu/projects/151955640

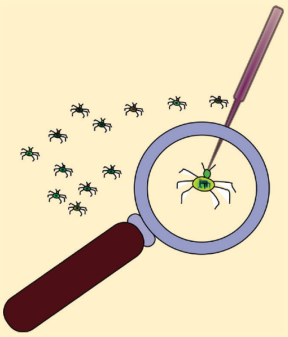
Wie weiter?

- Kannst du einen Punktezähler einbauen, der immer um 1 geändert wird, wenn eine Frage richtig beantwortet wurde?
- Kannst du im „Unterprogramm“ einbauen, dass bei einer falschen Antwort die richtige Lösung angezeigt wird?

Debugging - Fehler finden

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 hingeschrieben?

2. Vergessene Blocks

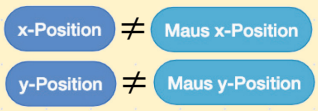
Gehe für dich in Gedanken die einzelnen Blocks durch. Machen sie Sinn oder fehlt etwas? Falls du einen Code abschreibst: Zähle die Blocks nach. Vielleicht kannst du deinen Code auch mit jemanden vergleichen.

3. Verwechselte Blocks

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

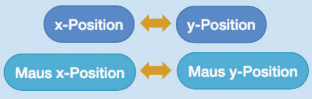
Auch Blocks mit ähnlichen Anweisungen können schnell verwechselt werden:

Vielleicht haben die Blocks 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 ein Skript-Teil kopiert hast, ist es oft notwendig, noch die Zahlen oder das Vorzeichen zu ändern:

6. Falsche Auswahl

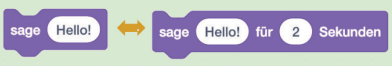
Manchmal geht das Ändern der Dropdown-Menüs vergessen:

7. Falsche Reihenfolge

Sind die Blocks in der korrekten Reihenfolge?

8. Zeitliche Begrenzung

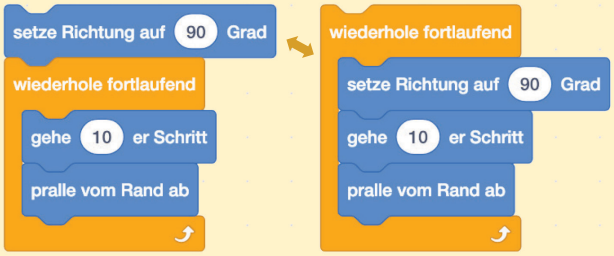
Sind Aktionen zeitlich begrenzt, wo sie es nicht sein sollten oder umgekehrt?



Debugging - Fehler finden

9. Schleifen

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



Fehlt eine Schleife, die etwas wiederholt prüft?

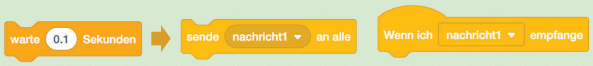
Ohne die Schleife findet diese Prüfung nur einmal ganz zu Beginn des Programms statt. Ein sehr häufiger Fehler!

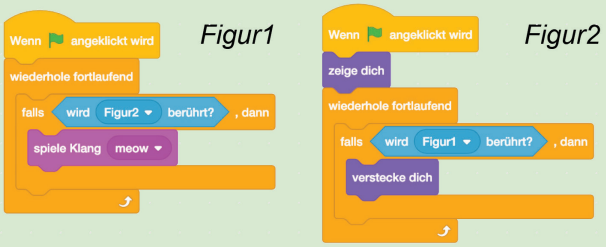
10. Tastatursteuerung einfach oder schnell

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 Skripts 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 dem *sende-empfange*-Blöcken zu arbeiten.





12. Lokale oder globale Variable

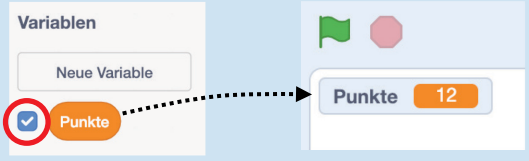
Kannst du bei einer Figur eine Variable nicht einsetzen, hast du sie vermutlich als lokale Variable definiert („nur für diese Figur“). Ein Punktezähler sollte zum Beispiel „für alle Figuren“ angelegt werden.

13. Initialisieren

Beim Starten deines Programms über das grüne Fänchen 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.

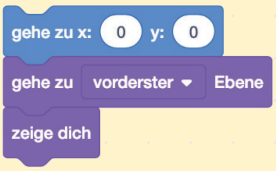
14. Werte/Zustände anzeigen

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:



Scratch Projektideen

Hintergrundinfos für Erwachsene

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 der individuellen Gestaltung überlassen. Die Broschüre eignet sich damit insbesondere für Projektarbeiten, wenn Scratch als Umgebung bereits bekannt 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 erlernen.

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 37 Millionen einseh- & änderbare Programme (Stand Februar 2019).

Was wird benötigt?

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

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.

ScratchJr

Für jüngere Kinder eignet sich ScratchJr, welches deutlich weniger Funktionen bietet und nicht mit Scratch kompatibel ist. ScratchJr ist kostenlos als App für Android und iOS verfügbar.

Die Idee des Konstruktionismus

Scratch wurde mit der Idee des Konstruktionismus des Mathematikers und Piaget-Studenten Seymour Papert im Hinterkopf entwickelt. Die Idee des Konstruktionismus 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 Konstuktionismus erfährt man in Paperts Buch: *Mindstorms. Children, Computer and Powerful Ideas, Basic Books, New York, 1980*

Impressum / Kontakt

Download dieser Broschüre:
http://iLearnIT.ch/scratch

Weitere Broschüren:
http://iLearnIT.ch/broschueren

Pädagogische Hochschule Schwyz
Michael Hielscher / Beat Döbeli
michael.hielscher@phsz.ch
beat.doebeli@phsz.ch