

pädagogische hochschule schwyz

# Scratch

## Gruppenpuzzle



Michael Hielscher  
Beat Döbeli Honegger

3. Oktober 2017



zur Verwendung im Unterricht  
hier schneiden

## Worum geht es?

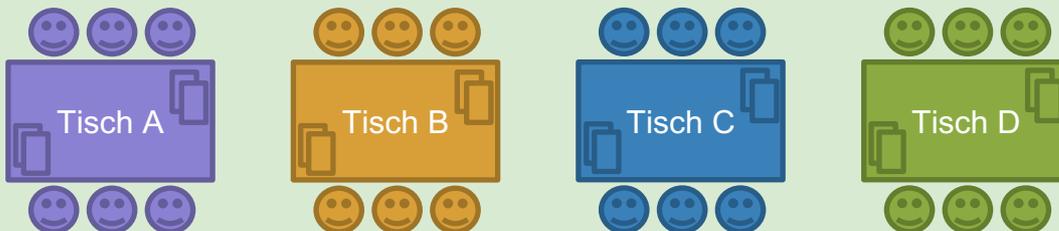
Das vorliegende Gruppenpuzzle ermöglicht Schülerinnen und Schülern, sich in der sog. Expertenrunde mit einem von vier Aspekten (Gestaltung, Bewegung, Sensoren sowie Rechnen & Variablen) der Programmierumgebung Scratch intensiver zu beschäftigen und zu echten Expertinnen und Experten zu werden. In der Unterrichtsrunde bringen alle ihr spezifisches Wissen und Können ein, um ein Spiel mit Scratch zu entwickeln. Die Durchführung als Gruppenpuzzles erfordert mindestens 2 x 90 Minuten. Alternativ zur Puzzlemethode lassen sich die einzelnen Posten auch gemeinsam nacheinander bearbeiten.



## Wie funktioniert das Gruppenpuzzle?

### 1. Expertengruppen bilden

Die Lehrperson verteilt die farblich codierten Arbeitsblätter der vier Themen auf vier Tischen. Jetzt verteilen sich die Schülerinnen und Schüler gleichmässig auf die vier Tische. Alle arbeiten ihre Arbeitsblätter für sich (idealerweise mit einem eigenen Computer) durch. Gibt es Fragen, wird immer zuerst versucht, diese am Tisch gemeinsam zu beantworten. Wichtig: Jedes Mitglied einer Expertengruppe ist später verantwortlich für ihr Thema. Wenn jemand etwas nicht ganz verstanden hat, sollte er sich unbedingt von seiner Gruppe helfen lassen oder die Lehrperson um Hilfe bitten.



### 2. Ergebnissicherung

Mit Hilfe von Kontrollfragen sollte in jeder Expertengruppe kurz überprüft werden, ob die Inhalte ihres Themas weitestgehend erfasst und verstanden wurden. Hier je ein Beispiel pro Gruppe:

- Gruppe A: Was ist der Unterschied zwischen Kostüm und Figur?
- Gruppe B: Welche Koordinaten hat die rechte untere Ecke der Bühne?
- Gruppe C: Könntest du mit „Pfeil nach unten“ die Maus auch rückwärts laufen lassen?
- Gruppe D: Was müsste man ändern, um eine Zahl bis 10'000 raten zu lassen?

### 3. Unterrichtsrunde: Gemeinsame Entwicklung eines Spiels in Scratch

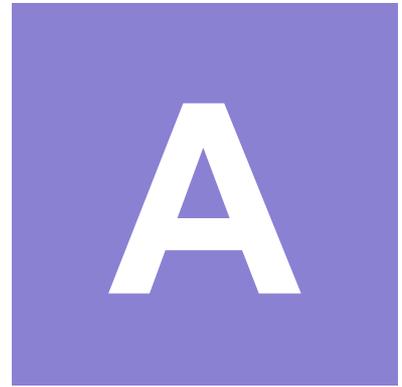
Alle sind nun Expertin bzw. Experte für einen Teil von Scratch. Jetzt werden 4er-Gruppen gebildet, sodass jedes Thema genau einmal vertreten ist. Sollte es nicht aufgehen, können einige 5er-Gruppen gebildet werden. Pro Tisch genügt ein Computer. In diesen Gruppen wird ein Scratch-Spiel mit Hilfe der grauen Arbeitsblätter entwickelt. Die Arbeitsblätter geben Hinweise, wo welche Expertise besonders gefragt ist (farbige Kästchen). Einige Teilaufgaben sind als Zusatz gekennzeichnet. In der Gruppe wechselt man sich am Computer ab, erklärt was man tut und unterstützt sich gegenseitig!



### 4. Präsentation

Zum Schluss stellen die Gruppen ihre Spielidee und den aktuellen „Prototypen“ vor. Was ist das Besondere am Spiel? Jede Gruppe sollte drei Dinge auswählen, auf die sie besonders stolz ist oder die sie als besonders schwierig empfand. Es spielt dabei keine Rolle, ob die Spiele schon komplett fertig sind oder nicht ganz wie gewünscht funktionieren.

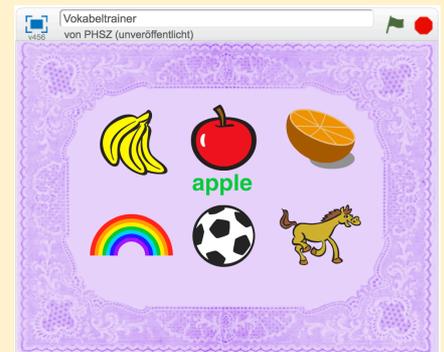
# Gestalten mit Scratch



## Worum geht es?

Mit Scratch kannst du sowohl fertige Bilder, Töne und Klänge verwenden, als auch selbst neue erstellen. Mit dem Bildeditor kannst du Figuren und Hintergründe bearbeiten und neu zeichnen. Jede Figur kann mehrere Bilder verwenden, um sie zu animieren (z. B. stehen, laufen, springen ...). Töne kannst du mit dem Toneditor direkt aufnehmen und schneiden.

Mit dieser Anleitung baust du einen sprechenden Vokabeltrainer. Starte jetzt deine Ausbildung zur Expertin / zum Experten für die Gestaltung von Bild und Ton mit Scratch!



## Bei Scratch anmelden

Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox mit aktuellem Flashplayer. Gehe auf <http://scratch.mit.edu> Es lohnt sich ein Scratch-Konto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts.



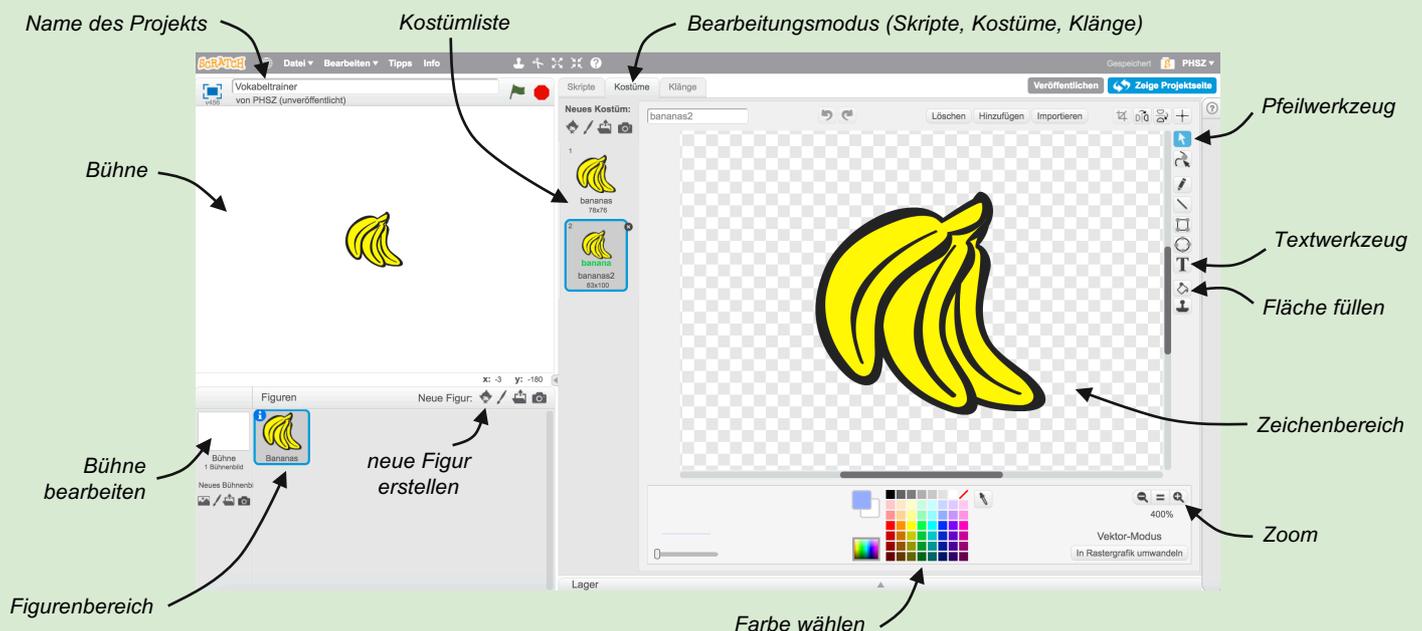
Du benötigst die folgenden Angaben: Einen Benutzernamen (wähle etwas phantasievolles), ein Passwort, dein Geburtsdatum und eine Emailadresse.

Erstelle ein neues, leeres Projekt in Scratch, indem du auf „Entwickeln“ klickst. Nenne es „Vokabeltrainer“.

1. *Erstelle eine neue Figur und wähle die Bananen aus. Die Katze kannst du im Figurenbereich mit einem Rechtsklick löschen.*

2. *Wechsle in den Bearbeitungsmodus „Kostüme“ und stelle 400% Zoom ein.*

3. *Klicke in der Kostümliste mit der rechten Maustaste auf die Bananen und „dupliziere“ sie.*



## Das solltest du wissen

- Jede Figur besitzt ein oder mehrere Kostüme.
- Scratch hat einen eingebauten Bildeditor, um Kostüme zu bearbeiten.

## Weitere Aufgaben

- Experimentiere mit den Zeichenwerkzeugen auf der rechten Seite des Zeichenbereichs. Kannst du die Kopie der Bananen grün einfärben?
- Kannst du ein hellblaues, gefülltes Rechteck zeichnen und hinter die Bananen legen? Tipp: mit einem zweiten Rechteck darunter, kannst du einen Schatteneffekt erstellen.



4. Schreibe mit dem Textwerkzeug den englischen Namen „banana“ ins zweite Kostüm „banana2“

5. Verwende das Pfeilwerkzeug, um den Text „banana“ mittig unter den Bananen zu positionieren (siehe Bild auf Blatt A1).

6. Wechsle in den Bearbeitungsmodus „Skripte“ und erstelle ein Programm, um das Kostüm beim Anklicken zu wechseln.

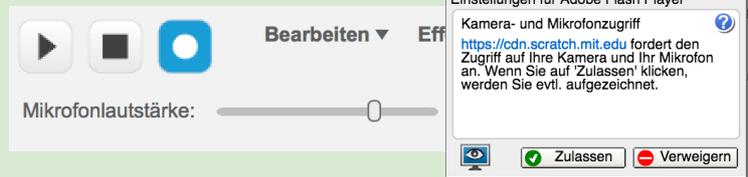
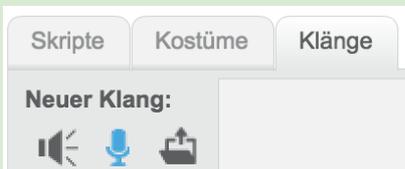
7. Ziehe dazu die Blöcke **Wenn ich angeklickt werde**, **wechsele zu Kostüm bananas** und **warte 1 Sek.** in den Skriptbereich und füge die Blöcke wie im Bild gezeigt zusammen (die Blockfarbe gibt an, wo du den Block in der Palette findest).



8. Klicke mit der Maus auf die Bananen auf der Bühne. Wenn du alles richtig gemacht hast, sollte für eine Sekunde der Text „banana“ erscheinen und danach wieder verschwinden.

9. Wechsle in den Bearbeitungsmodus „Klänge“ und klicke auf „Neuen Klang aufzeichnen“.

10. Klicke auf den Aufnahmeknopf und bestätige zunächst den Sicherheitsdialog mit „Zulassen“.

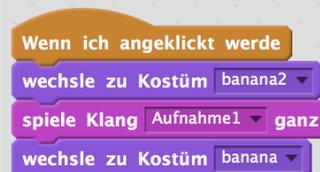


11. Erstelle die erste Aufnahme und sprich ein englisches „banana“ ins Mikrofon deines Computers.

12. Erstelle weitere Aufnahmen durch erneutes Klicken des Aufnahmeknopfes. Verändere auch den Mikrofonlautstärke-Regler. Höre dir deine Aufnahmen an und entscheide, welche dir am besten gefällt.

13. Markiere alle Teile in der Audiospur, die du nicht mehr benötigst und drücke die Entfernen-Taste oder Bearbeiten → Löschen, sodass nur noch deine Auswahl übrig bleibt.

14. Tausche den Block **warte 1 Sek.** durch den Block **spiele Klang Aufnahme1 ganz** im Skriptbereich aus und teste dein Programm auf der Bühne.



### Das solltest du wissen

- Bilder und Töne lassen sich direkt in Scratch erstellen und bearbeiten.
- Kostüme lassen sich über einen Skriptblock wechseln.

### Weitere Aufgaben

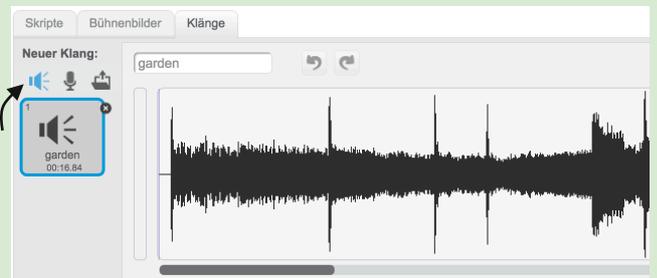
- Experimentiere mit den Skriptblöcken in der Kategorie „Aussehen“. Kannst du die Bananen beim Anklicken kurzzeitig vergrößern oder die Farbe verändern?
- Kannst du mit den Effekten im Audioeditor deine eigene Stimme rückwärts abspielen lassen?

15. Wähle unten links die Bühne aus, sodass diese blau umrandet wird. Wähle als neues Bühnenbild aus dem Katalog „doily“ aus.



neues Bühnenbild auswählen

16. Wechsle zurück auf Skripte und klicke mehrfach schnell hintereinander auf den Block **ändere Farbe** -Effekt um **25** in der Skriptpalette (Aussehen), bis du deine Lieblingsfarbe gefunden hast.



17. Wechsle auf „Klänge“ und klicke auf und füge den Klang „garden“ (aus der Kategorie Musikschleifen) zur Bühne hinzu. Nicht benötigte Klänge kannst du mit löschen.

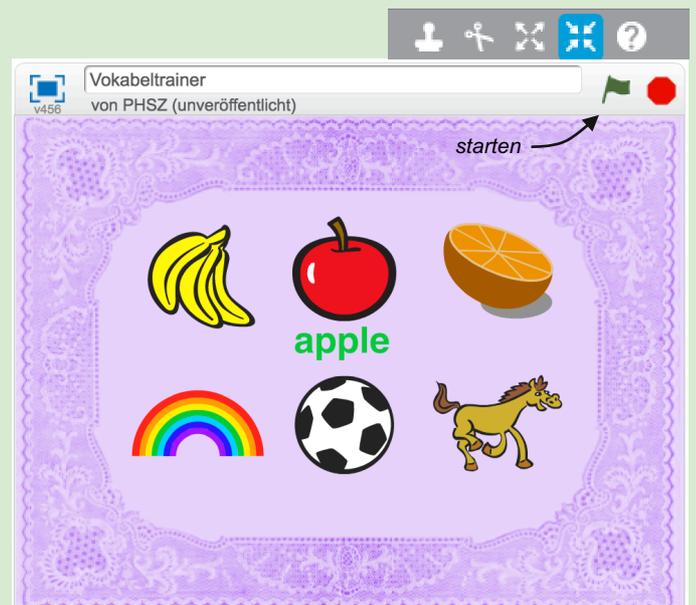
18. Erstelle für die Bühne das folgende Skript, um die Hintergrundmusik in einer Endlosschleife abspielen zu lassen. Teste dein Programm durch einen Klick auf das grüne Fähnchen.



19. Eventuell ist die Hintergrundmusik zu laut im Vergleich zu deiner Aufnahme. Füge einen Baustein zur Lautstärkeinstellung ins Skript ein.

20. Erstelle mindestens drei weitere Figuren wie in Schritt 1 bis 13. Erstelle jeweils ein zweites Kostüm mit der englischen Bezeichnung als Text darunter und nimm ein passendes Audio dazu auf. Erstelle die passenden Skripte zum Abspielen und Kostümwechseln, sobald diese angeklickt werden.

21. Ordne die Figuren durch Ziehen mit der Maus auf der Bühne an. Einige Figuren sind vermutlich zu gross. Wähle in der Werkzeugliste (ganz oben) das Verkleinerungswerkzeug und klicke mehrmals auf die Figur, bis alle in etwa gleich gross sind.



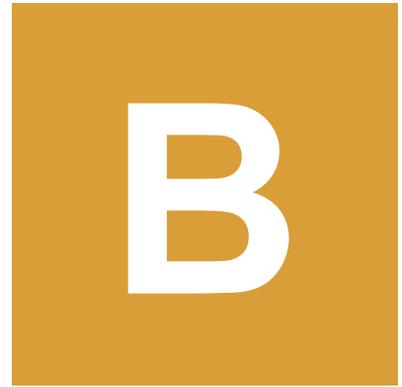
## Das solltest du wissen

- Auch die Bühne besitzt Klänge.
- Klänge lassen sich mit einer Schleife fortlaufend wiederholen.
- Figuren lassen sich vergrößern und verkleinern.

## Weitere Aufgaben

- Verändere die Skripte deiner Gegenstände so, dass beim Anklicken, zusätzlich zum Kostümwechsel der Effekt **setze Farbe** -Effekt auf **100** und **setze Farbe** -Effekt auf **0** genutzt wird, um die Farbe der Figuren zu verändern.
- Klicke auf **Neue Figur:** , um deine eigene Figur zu zeichnen. Wechsel in den Vektor-Modus und verwende die Zeichenwerkzeuge.

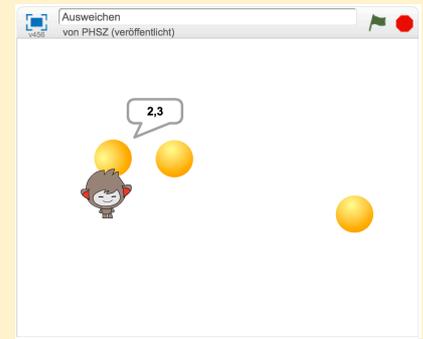
# Bewegung und Koordinaten



## Worum geht es?

In Scratch bist du der Regisseur und kannst allen Figuren Anweisungen geben, wohin sie sich auf der Bühne bewegen, wo sie auftauchen und verschwinden sollen. Für diese Anweisungen verwendest du in Scratch verschiedene Skript-Blöcke.

Mit dieser Anleitung baust du ein Spiel, bei dem es darum geht, herumfliegenden Bällen auszuweichen. Starte jetzt deine Ausbildung zur Expertin / zum Experten für Bewegung und Koordinaten in Scratch!



## Bei Scratch anmelden

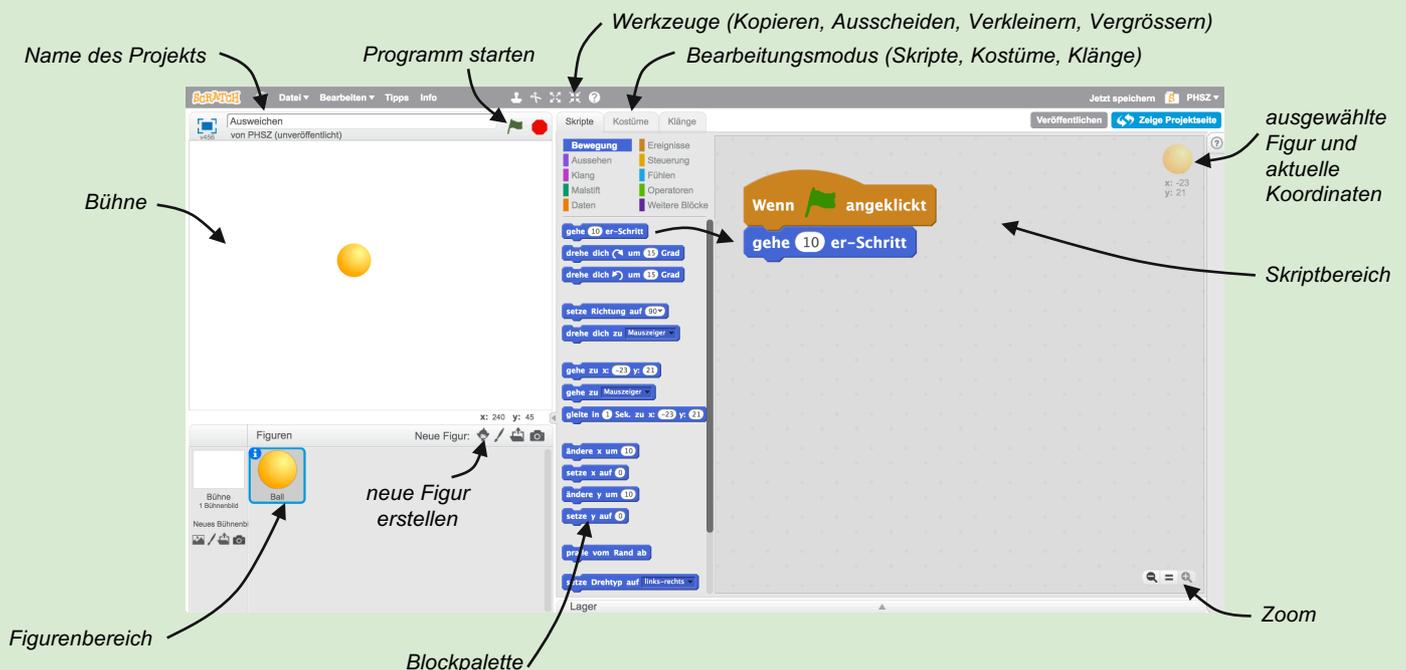
Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox mit aktuellem Flashplayer. Gehe auf <http://scratch.mit.edu> Es lohnt sich ein Scratch-Konto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts.



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

Erstelle ein neues, leeres Projekt in Scratch, indem du auf „Entwickeln“ klickst. Nenne es „Ausweichen“.

1. *Erstelle eine neue Figur und wähle „Ball“ aus. Die vorgegebene Katze kannst du im Figurenbereich mit einem Rechtsklick löschen.*
2. *Suche in der Blockpalette im Abschnitt „Ereignisse“ den Block **Wenn angeklickt** und ziehe ihn in den Skriptbereich.*
3. *Füge einen Block „**gehe 10er Schritt**“ (Bewegung) an den ersten Block an. Klicke auf , um das Programm zu starten.*



## Das solltest du wissen

- Ein Scratch-Projekt besitzt eine oder mehrere Figuren.
- Blöcke werden aus der Blockpalette gezogen und wie Puzzleteile zusammengefügt.

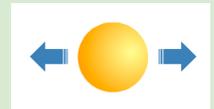
## Weitere Aufgaben

- Experimentiere mit dem **gehe 10er Schritt** Block und verändere die Zahl 10. Welche Auswirkungen hat das?
- Füge einen **drehe dich um 15 Grad** Block im Skriptbereich ein und starte dein Programm mehrmals hintereinander. Wie oft musst du klicken, bis deine Figur wieder am Anfang steht?

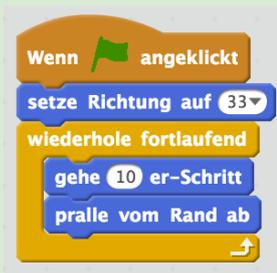
4. Der Ball soll herumfliegen und vom Rand abprallen. Damit er sich fortlaufend bewegt, benötigen wir eine Wiederholung (auch Schleife genannt). Verändere dein Skript wie in der Abbildung rechts gezeigt. Den Block **pralle vom Rand ab** findest du weit unten in der Blockpalette (Bewegung). Starte das Programm und beobachte die Bewegung des Balls.



5. Du kannst die Geschwindigkeit der Bewegung anpassen, in dem du die Schrittlänge von 10 zum Beispiel auf 5 verringerst.



6. Für ein Ausweichspiel ist die Bewegung des Balls noch sehr vorhersehbar (nur von links nach rechts und umgekehrt). Füge einen **setze Richtung auf 33** Block vor der Schleife ins Skript ein. Gib z. B. 33 Grad als Richtung an.



7. Teste das Programm und beobachte nun die Bewegung des Balls. Verändere die Gradzahl des Richtungsblocks. Teste zum Beispiel auch 0, 90 oder 180 Grad.

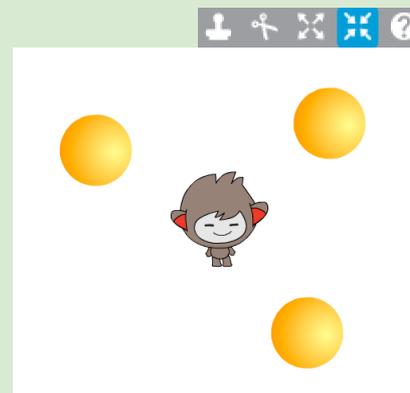
8. Dupliziere den Ball mit dem „Duplizieren Werkzeug“ (Stempel-Symbol) zweimal. Stelle im Skript für jeden Ball eine andere Anfangsrichtung ein und teste das Programm.



9. Für das Ausweichspiel benötigen wir eine Spielfigur, die vom Spieler gesteuert wird. Erstelle eine neue Figur mit der Vorlage „Nano“ und platziere sie in der Mitte der Bühne. Du kannst Figuren auch im Figurenbereich mit der Maus verschieben.



10. Um den Bällen ausweichen zu können, ist Nano noch viel zu gross. Wende das „Verkleinern Werkzeug“ solange auf Nano an, bis er etwa auf Ballgröße geschrumpft ist.



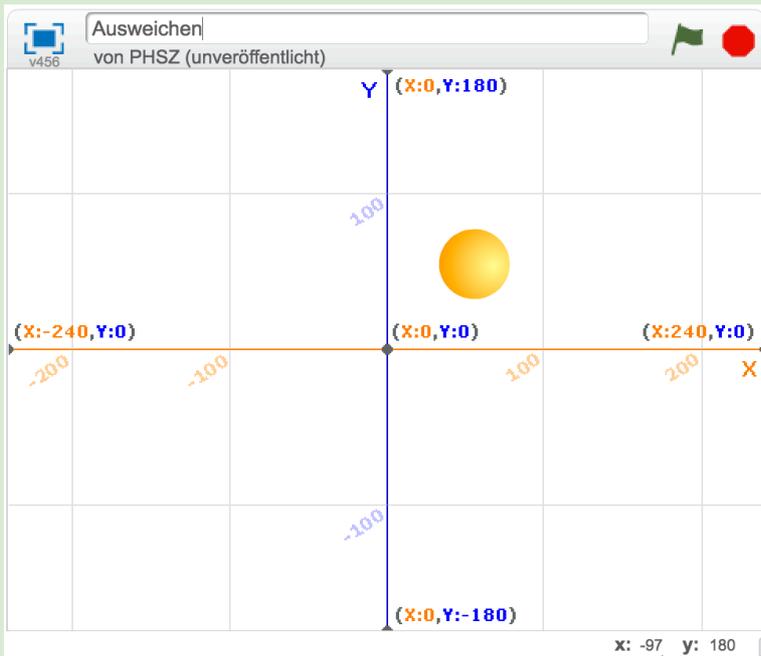
### Das solltest du wissen

- Jede Figur hat ihren eigenen Skriptbereich und alle Skripte laufen gleichzeitig ab.
- Mit einer fortlaufenden Schleife kannst du Animationen erstellen.

### Weitere Aufgaben

- Verwende das „Verkleinern Werkzeug“ bzw. „Vergrößern Werkzeug“ um die drei Bälle unterschiedlich gross zu machen.
- Stelle die Richtung und die Geschwindigkeit der Bälle so ein, dass sich die Bewegungen unterscheiden. Ein grosser Ball sollte sich langsamer bewegen als ein kleiner Ball.

11. In Scratch gibt es zwei verschiedene Arten von Bewegung. Bei der „relativen“ Bewegung lässt man eine Figur einfach einen Schritt nach vorn laufen, wobei sie immer von der aktuellen Position in die aktuelle Blickrichtung gehen wird. Bei der „absoluten“ Bewegung verwendet man genaue Koordinaten der Bühne, um einer Figur ganz exakt mitzuteilen, wo sie sich hinstellen soll.



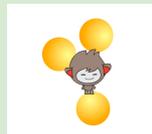
Tipp: Fahre mit der Maus über die Bühne. Unten rechts wird immer die aktuelle Position der Maus angezeigt.

aktuelle Mausposition

14. Auch die Bälle sollen bei jedem Programmstart an einer fixen Anfangsposition (eher am Rand des Spielfelds) starten. Zudem ändern wir die fortlaufende Schleife in eine bedingte Schleife. Die Befehle werden nur noch solange wiederholt, bis die Bedingung **wird Nano berührt?** eintritt. Vergiss nicht, dass du die Änderungen für jeden Ball einzeln vornehmen musst.



15. Starte mehrmals das Programm. Jedes Spiel sollte nun genau gleich ablaufen und die Bälle auf den genau gleichen Bahnen umherfliegen. Nano wird sie zudem einfangen, sobald sie ihn berühren.



16. Nano soll nicht nur in der Bühnenmitte herumstehen, sondern vom Spieler gesteuert werden. Erstelle eine fortlaufende Schleife für Nano. Füge einen Block **gehe zu Mauszeiger** darin ein.



17. Teste das Spiel und versuche mit der Maus den Bällen schnell auszuweichen. Wenn du möchtest, kannst du weitere Bälle duplizieren.

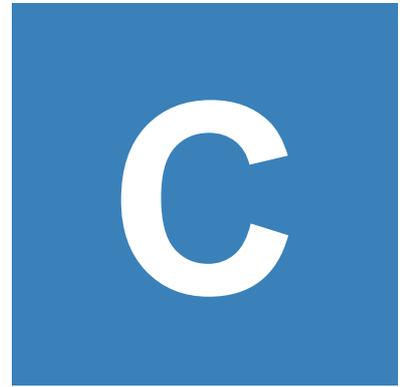
## Das solltest du wissen

- Die Bühne von Scratch hat ein Koordinatensystem.
- Es gibt zwei verschiedene Arten von Bewegung: relative und absolute.

## Weitere Aufgaben

- Das Spiel beginnt immer sofort, was für den Spieler schwierig sein kann. Abhilfe schafft ein **warte 1 Sek.** Block, den du jeweils vor die Schleife der Skripte von Nano und jedem Ball einfügen kannst.
- Wie lange kannst du den Bällen ausweichen? Füge am Ende jedes Ballskripts noch ein **sage Stoppuhr** ein (Blockpalette: Aussehen und Fühlen).

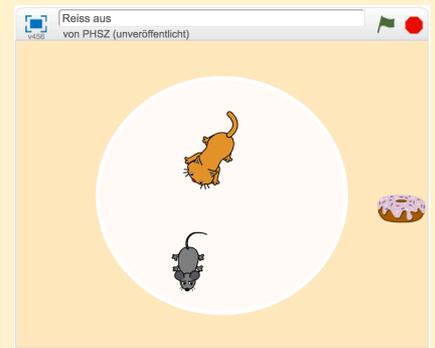
# Bedingungen und Sensoren



## Worum geht es?

Ist gerade eine Taste gedrückt? Wurde etwas angeklickt? Sind Geräusche zu hören? Bewegt sich etwas vor der Webcam? Berührt eine Figur eine andere? Die meisten Computerspiele stellen in einer Schleife fortlaufend solche Fragen, um je nach Antwort eine andere Reaktion auszulösen.

Mit dieser Anleitung baust du ein Spiel, bei dem eine Maus mit der Tastatur gesteuert wird, um einer Katze auszuweichen und gleichzeitig Futter zu sammeln. Starte jetzt deine Ausbildung zur Expertin / zum Experten für Bedingungen und Sensoren mit Scratch.



## Bei Scratch anmelden

Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox mit aktuellem Flashplayer. Gehe auf <http://scratch.mit.edu> Es lohnt sich ein Scratch-Konto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts.



Entwickeln

Entdecke

Diskutieren

Über Scratch

Hilfe

Suche

Scratcher werden

Anmelden

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

Erstelle ein neues, leeres Projekt in Scratch, indem du auf „Entwickeln“ klickst. Nenne es „Fang den Ball“.

1. *Erstelle eine neue Figur und wähle „Mouse1“ aus. Die vorgegebene Katze kannst du im Figurenbereich mit einem Rechtsklick löschen.*
2. *Suche in der Blockpalette im Abschnitt „Ereignisse“ den Block **Wenn Lautstärke > 10** und ziehe ihn in den Skriptbereich. Falls der Sicherheitsdialog erscheint, wähle „Zulassen“ aus.*
3. *Suche im Abschnitt „Bewegung“ den Block **gehe 10 er-Schritt** und füge diesen im Skriptbereich an den ersten Block an. Teste dein Programm, indem du in die Hände klatschst.*

Annotations in the screenshot:

- Name des Projekts
- Bühne
- Figurenbereich
- Bühne bearbeiten
- ausgewählte Figur
- Skriptbereich
- Sicherheitsdialog
- Zoom
- Blockpalette
- neue Figur erstellen
- Lager
- Bearbeitungsmodus (Skripte, Kostüme, Klänge)

## Das solltest du wissen

- Ein Scratch-Projekt besitzt eine oder mehrere Figuren.
- Blöcke werden aus der Blockpalette gezogen und wie Puzzleteile zusammengefügt.

## Weitere Aufgaben

- Lass dir die aktuelle Lautstärke von Scratch anzeigen, indem du unter „Fühlen“  **Lautstärke** anklickst. Wie laut ist es, wenn du leise sprichst?
- Verwende einen weiteren **Wenn Lautstärke > 10** Block mit einem höheren Vergleichswert. Füge einen **drehe dich um 15 Grad** Block (Bewegung) darunter ein. Kannst du die Maus nur durch lautes/leises Sprechen steuern?

4. Lösche das Skript von Mouse1, indem du die Blöcke wieder in die Palette zurück schiebst. Erstelle jetzt eine fortlaufende Schleife für die Figur Mouse1. Die Blöcke dafür findest du jeweils in der gleichfarbigen Kategorie der Blockpalette (Ereignisse, Steuerung).



6. Klicke auf und teste das Programm mit den Pfeiltasten. Passe die Geschwindigkeit und die Drehungen über die Zahlen 10 und 15 bei „gehe“ und „drehe“ solange an, bis dir die Steuerung gefällt.

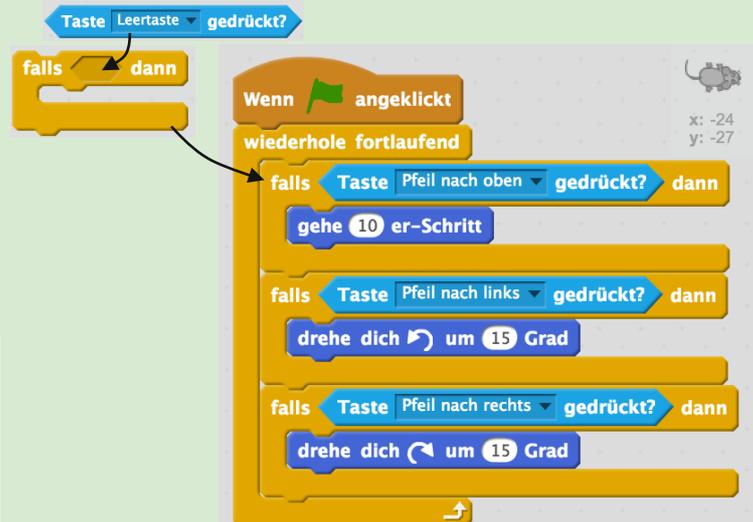
7. Füge eine neue Figur im Figurenbereich hinzu. Wähle aus dem Katalog die Vorlage „Cat2“ (Katze von oben). Verwende das „Verkleinern“ Werkzeug (Toolbar), um Katze und Maus auf der Bühne passend zum Spiel zu verkleinern.



9. Starte das Spiel. Steuere die Maus über die Bühne und versuche der Katze auszuweichen. Passe die Geschwindigkeiten so an, dass es nicht zu einfach oder zu schwierig wird.

10. Wenn die Katze die Maus berührt, ist das Spiel vorbei. Erneut verwenden wir einen „falls ... dann“ Block und prüfen, ob die Katze die Maus berührt. Wenn ja, beenden wir das Programm mit .

5. Die Maus soll nun mit der Tastatur gesteuert werden. Erstelle das folgende Skript mit drei „falls ... dann“ Blöcken und füge jeweils den Block in die Bedingungsücke ein. Wähle die Pfeiltasten aus und füge in jeden Block einen passenden Bewegungsblock ein:



8. Die Katze soll die Maus verfolgen. Erstelle eine fortlaufende Schleife und füge die beiden Bewegungsblöcke wie im Bild gezeigt ein.



11. Starte das Spiel erneut und weiche solange aus, wie du kannst.

Schiebe die Katze vor jedem Neustart von der Maus weg.

## Das solltest du wissen

- Mit einer fortlaufenden Schleife kannst du laufend Bedingungen mit „falls ... dann“ überprüfen.
- Bedingungen erlauben einem Programm, auf etwas zu reagieren.

## Weitere Aufgaben

Verändere das Verhalten der Katze so, dass sie bei Krach (z. B. Lautstärke grösser 10) einen zusätzlichen Schritt macht. Verwende dazu einen

Block und einen Block.

Passe die Zahl im Wenn-Block an deine Umgebungslautstärke an ☺

12. Die Maus würde in der Realität wohl einfach davon laufen. Es sei denn, es gäbe einen Anreiz sich in Gefahr zu begeben. Erstelle eine weitere Figur mit der Vorlage „Donut“.

13. Verwende erneut das „Verkleinern-Werkzeug“ aus der Toolbar oben und klicke damit mehrmals auf den Donut, damit auch dieser zum Spiel passt.

14. Wähle einen neuen Hintergrund für das Spiel (Abbildung rechts). Verwende die Vorlage „light“. In der Bühnenmitte soll es so aussehen, als ob dort eine Lampe brennt.

15. Wähle den Donut aus und erstellen ein Skript, welches fortlaufend prüft, ob der Donut von der Maus berührt wird. Wenn ja, soll er verschwinden und nach kurzer Wartezeit an einer zufälligen Stelle wieder erscheinen. Suche alle nötigen Blöcke in der Blockpalette zusammen und erstelle das folgende Skript:



16. Starte das Spiel. Steuere die Maus mehrmals mit den Pfeiltaste bis zum Donut, ohne dabei die Katze zu berühren.



17. Wähle die Katze aus und verändere ihr Skript so, dass sie sich bei Berührung der Fläche mit der Farbe Orange (ausserhalb des Lampe) langsamer bewegt als sonst. Eine mögliche Lösung ist rechts gezeigt.



**Hinweis:**  
Um eine Farbe auszuwählen, klicke zunächst auf das kleine farbige Quadrat und anschließend auf die Wunschfarbe auf der Bühne.

## Das solltest du wissen

- Die Reihenfolge der „falls ... dann“ Blöcke spielt in einer fortlaufenden Schleife meist keine Rolle.
- Du kannst prüfen, ob eine andere Figur oder eine Farbe berührt wird.

## Weitere Aufgaben

Verändere das Skript der Katze so, dass diese sich fortlaufend zum Mauszeiger statt zu Mouse1 dreht.  
Spielt das Spiel zu zweit. Spieler 1 spielt nun per Tastatur und Spieler 2 versucht mit der Computermouse die Katze zu steuern.

# Zahlen und Variablen



## Worum geht es?

Computer können sehr schnell rechnen und grosse Datenmengen speichern. In Scratch kannst du Variablen verwenden. Eine Variable kannst du dir wie eine beschriftete Schachtel vorstellen, in die eine Zahl, ein Buchstabe, ein Wort oder ein ganzer Satz hineingelegt werden kann.

Mit dieser Anleitung baust du ein einfaches Ratespiel, bei dem es darum geht, die Zahl in einer Variablen zu erraten. Starte jetzt deine Ausbildung zur Expertin / zum Experten für Zahlen und Variablen in Scratch!



## Bei Scratch anmelden

Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder Firefox mit aktuellem Flashplayer. Gehe auf <http://scratch.mit.edu> Es lohnt sich ein Scratch-Konto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts.



Entwickeln

Entdecke

Diskutieren

Über Scratch

Hilfe

Suche

Scratcher werden

Anmelden

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

Erstelle ein neues, leeres Projekt in Scratch, indem du auf „Entwickeln“ klickst. Nenne es „Errate die Zahl“.

1. *Erstelle eine neue Figur und wähle eine Vorlage aus, die dir gefällt. Die Katze kannst du im Figurenbereich mit einem Rechtsklick löschen.*

2. *Suche in der Blockpalette im Abschnitt „Ereignisse“ den Block **Wenn angeklickt** und ziehe ihn in den Skriptbereich.*

3. *Füge einen Block (Aussehen) **denke Hmm... für 2 Sek.** an den ersten Block an. Schreibe den Text wie unten gezeigt in das Textfeld. Klicke auf , um das Programm zu starten.*

Labels in the screenshot:

- Name des Projekts
- Bühne
- Figurenbereich
- Bühne bearbeiten
- neue Figur erstellen
- Blockpalette
- Bearbeitungsmodus (Skripte, Kostüme, Klänge)
- ausgewählte Figur
- Skriptbereich
- Zoom

## Das solltest du wissen

- Ein Scratch-Projekt besitzt eine oder mehrere Figuren.
- Blöcke werden aus der Blockpalette gezogen und wie Puzzleteile zusammengefügt.

## Weitere Aufgaben

- Ändere die Sekundenangabe im Block **denke Hmm... für 2 Sek.** und stelle den Wert so ein, dass du den Text gut lesen kannst, sobald du auf  klickst.
- Füge einen weiteren Block **sage Hello! für 2 Sek.** hinzu.  
Was ist der Unterschied zwischen „denke“ und „sage“?

4. Lass deine Figur eine zufällige Zahl zwischen 1 und 100 sagen. Verwende das folgende Skript:

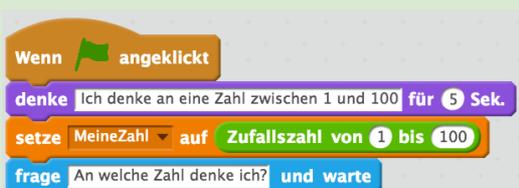


Hinweis: Den Block „Zufallszahl“ findest du im grünen Bereich Operatoren.

6. Im Bereich Daten stehen dir nun neue Blöcke zur Verfügung. Statt die Zufallszahl anzusagen, legen wir sie in die Variable „MeineZahl“ (wie in eine Schachtel). Verwende den Block **setze MeineZahl auf Zufallszahl von 1 bis 100**



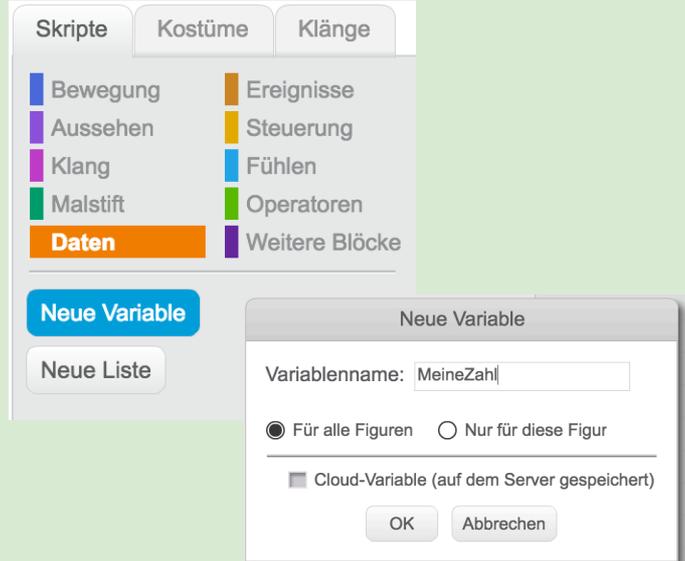
7. Der Spieler soll nun deine Zahl erraten. Lass die Figur eine Frage stellen. Du findest den passenden Block im Bereich Fühlen. Schreibe ins Textfeld. Klicke auf und teste dein Programm.



9. Mit einem „falls ... dann ... sonst“-Block können wir überprüfen, ob der Inhalt der Variable Antwort mit dem Inhalt der Variable MeineZahl übereinstimmt. Erweitere dein Skript wie rechts gezeigt:



5. Für das Ratespiel soll die Figur die Zahl natürlich nicht gleich verraten, sondern sich merken. Erstelle eine neue Variable „MeineZahl“ (Blockpalette - Daten).



8. Deine Figur stellt nun eine Frage, aber die Antwort wird noch nicht überprüft. Die Antwort ist in Scratch ebenfalls eine Variable. Da sie häufig verwendet wird und automatisch vorhanden ist, wird sie blau und im Bereich Fühlen dargestellt:



## Das solltest du wissen

- In Scratch können eigene Variablen angelegt werden.
- ist eine vordefinierte Variable, die durch den Block gesetzt wird.

## Weitere Aufgaben

- Teste das Spiel einige Male. Kannst du das Spiel so verändern, dass die Zahl nur zwischen 1 und 20 liegt? An welchen Stellen musst du dafür etwas verändern?
- Was passiert, wenn man als Spieler ein Wort wie „Maus“ als Antwort gibt, statt einer Zahl? Probiere es aus! Kannst du erklären was passiert?

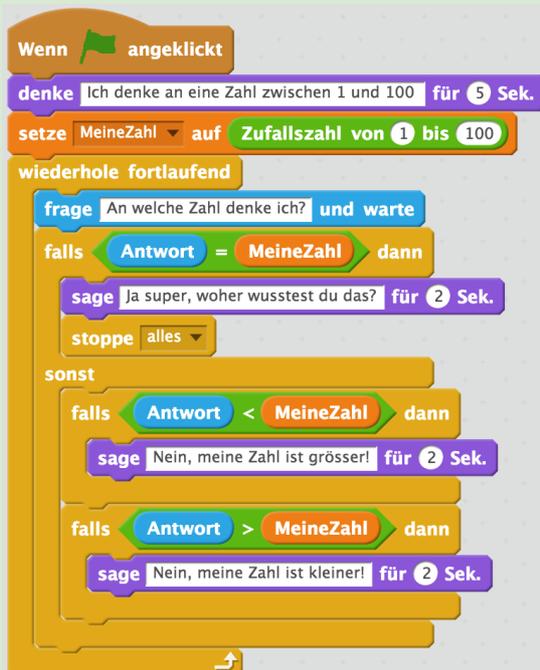
10. Variablen lassen sich auf der Bühne anzeigen. Das ist sehr praktisch, zum Beispiel um einen Punktestand oder die verbleibende Spielzeit anzuzeigen. Für das Ratespiel müssen wir die Variable aber verstecken. Entferne das Häkchen  **MeineZahl** in der Blockpalette.



11. Ohne die Anzeige benötigen wir sehr viel Glück, um die richtige Zahl zu erraten. Erlauben wir dem Spieler also mehrere Rateversuche. Verwende eine "wiederhole fortlaufend" Schleife. Teste das Spiel!



12. Ohne einen Anhaltspunkt, ob unsere Antwort zumindest in der Nähe der gedachten Zahl lag, ist das Ratespiel immer noch pures Glück. Geben wir dem Spieler deshalb einen Hinweis. Lass die Figur sagen, ob die gedachte Zahl grösser oder kleiner als die Antwort war. Füge noch einen **stoppe alles** Block im Erfolgsfall hinzu.



13. Bei den meisten Spielen gibt es eine Möglichkeit, sich mit anderen Spielern zu messen. Wer schafft es, die Zahl des Computers mit möglichst wenig Versuchen zu erraten?

Erstelle eine **Neue Variable** unter Daten:  **Versuche**. Zu Beginn des Spiels setzen wir die Versuche immer auf 0 zurück. In jeder Spielrunde erhöhen wir sie um 1.



## Das solltest du wissen

- Der Wert einer Variable lässt sich setzen oder erhöhen/erniedrigen.
- Mit Operatoren (grün) lassen sich Variablen mit Werten oder mit anderen Variablen vergleichen.

## Weitere Aufgaben

- Wähle ein passendes Bühnenbild für das Spiel aus. Klicke unten links auf den ersten Knopf bei „Neues Bühnenbild“.
- Eine weitere, von Scratch bereits vorgegebene Variable ist  **Stoppuhr**. Probiere sie aus!



Ein eigenes Spiel  
entwickeln



## Worum geht es?

Ihr entwickelt gemeinsam ein umfangreicheres Spiel mit Scratch. Jede Expertin und jeder Experte bringt anderes Wissen und Können mit. Ihr benötigt nur einen Computer, an dem euer Spiel entstehen wird. Wechselt euch ab, je nachdem, welches Expertenwissen gerade gefragt ist. Helft euch gegenseitig!

Diese Anleitung soll euch helfen, zielgerichtet ein Spiel wie **Breakout** zu entwickeln. Das Spiel wurde 1976 von einem der Gründer von Apple für Atari entwickelt. Diese Anleitung ist nicht sehr detailliert. Ihr müsst selbst Lösungen erarbeiten, könnt dafür aber auch vieles selbst entscheiden. Testet das Spiel nach jedem Entwicklungsschritt. Viel Erfolg!



## Teamrollen bestimmen



5 min

Neben eurer Rolle als Expertin/Experte müssen die folgende Rollen zusätzlich in eurem Team verteilt werden. Lest euch die Rollenbeschreibungen durch. Legt gemeinsam fest, wer welche Zusatzrolle in eurer Gruppe übernehmen wird:

- **Rolle – Management:**  
Diese Person muss dafür sorgen, dass alle Teammitglieder bei der Sache bleiben. Ist das Team in einem Punkt uneinig, muss diese Person eine Entscheidung treffen, die dann alle akzeptieren.
- **Rolle – Entwicklungsleitung:**  
Diese Person ist verantwortlich für die Einhaltung der Zeitangaben für die einzelnen Schritte und schaut regelmässig auf die Uhr. Wird die Zeit in einem Abschnitt knapp, überspringt die Teilaufgaben, die mit ☀ gekennzeichnet sind und macht mit dem nächsten Abschnitt weiter.
- **Rolle – Beta-Testing:**  
Diese Person ist für die laufende Erprobung des Spiels zuständig und liefert Feedback, was noch nicht gut ist und was verbessert werden muss. Sie übernimmt die Rolle eines zukünftigen Spielers und soll auch kritisch sein, wenn etwas noch nicht richtig klappt.
- **Rolle – Marketing:**  
Diese Person wird euer Spiel am Schluss den anderen Gruppen vorstellen. Sie macht sich zwischendurch Notizen: Warum sollte man euer Spiel kaufen? Was ist besonders gelungen? Worauf seid ihr stolz?



## Spielidee kennenlernen



5 min

Schaut euch einen Ausschnitt auf YouTube zum Originalspiel von 1976 an: <http://tiny.cc/breakout>

Tragt kurz zusammen:

- Was ist das Ziel des Spiels?
- Worin liegt die Herausforderung?
- Wann gewinnt / verliert man das Spiel?
- Wer von euch könnte welchen Teil beitragen?



## Neues Projekt in Scratch anlegen, benennen und veröffentlichen



1 min



## Paddle / Schläger erstellen



10 min

- Wählt eine Figur aus dem Katalog als Paddle, oder zeichnet ein eigenes Paddle.
- Erstellt für das Paddle ein Skript zur Steuerung mit den Pfeiltasten (links / rechts) wie im Video.
- Begrenze die Bewegung am linken und rechten Bildrand mit einer Prüfung (z.B. `falls x-Position < -120 dann` )



### Ball erstellen und auf Aus prüfen

10 min

- Erstellt eine neue Figur „Ball“. Wählt dazu eine Figur aus dem Katalog, oder zeichnet selbst einen Ball. Verkleinert den Ball auf der Bühne, bis er gut zum Spiel passt (kleiner funktioniert meist besser).
- Erstellt ein Skript für den Ball, damit dieser *fortlaufend* herumfliegt und am Rand abprallt ( ).
- Prüft vor jeder Bewegung die Eigenschaft des Balls, ob er sich im Aus befindet ( ). Ist der Ball im Aus, soll er automatisch in die Bühnenmitte zurückgestellt werden und kurz warten ( ).



### Von Paddle abprallen

10 min

- Wenn der Ball das Paddle berührt, soll er einen Klang spielen und abprallen. Für das Abprallen könnt ihr die aktuelle Richtung des Balls mit umkehren. Fügt zusätzlich eine zufällige Drehung mit hinzu, damit der Ball nicht immer genau gleich fliegt.



### Steine erstellen

20 min

- Wählt eine Figur aus dem Katalog als Stein, oder zeichnet einen Stein mit dem Zeicheneditor (Vektormodus).
- Stein Skript: Nach Spielstart soll *fortlaufend* geprüft werden, ob der Stein den Ball berührt. Wird er berührt, soll nach kurzen Verzögerung ( ) der Stein verschwinden ( ) und ein Ton gespielt werden.
- Testet das Spiel mehrmals. Funktioniert der Stein wie er soll? Wird er beim Spielstart wieder angezeigt? Dupliziert ihn mehrmals und ordnet die Steine auf der Bühne an. Alle Steine verwenden das gleiche Skript.



### Von Steinen abprallen

20 min

- Der Ball soll beim Aufprall mit einem Stein abprallen. Verändert das Skript des Balls. Entweder ihr prüft auf Berührung mit jedem einzelnen Stein, oder ihr prüft auf Berührung einer Farbe (Steinfarbe). Das Abprallen sollte wie beim Paddle funktionieren. Testet das Spiel in diesem Schritt besonders intensiv.



### Punkte zählen und Spiel gewinnen

15 min

- Erstellt eine Variable „Punkte“ im Bereich „Daten“. Bei jedem Spielstart soll sie auf 0 gesetzt werden.
- Verändert das Skript von jedem Stein und fügt bei Berührung mit dem Ball noch ein.
- Fügt zwei Hintergrundbilder zur Bühne hinzu. Eins für „Gewonnen“ und eins für „Verloren“. Verwendet zum Beispiel ein Vorlagebild und schreibt grossen Text darauf.
- Fügt ein Skript zur Bühne hinzu, welches fortlaufend prüft, ob „Punkte“ der Anzahl Blöcke im Spiel entspricht. Wurden alle Blöcke getroffen, soll das Bühnenbild auf „Gewonnen“ wechseln und das Spiel gestoppt werden.



### Leben (Bälle) zählen und Spiel verloren

10 min

- Erstellt eine Variable „Leben“ im Bereich „Daten“. Bei jedem Spielstart soll diese auf 3 gesetzt werden.
- Wenn der Ball ins Aus gelangt, soll die Variable mit: um eins verringert werden.
- Ändert das Skript der Bühne so, dass bei keinen verbleibenden Leben ( ) das Bühnenbild auf „Verloren“ gewechselt wird und das Spiel gestoppt wird.

## Präsentation und Feedback

Jetzt müsst ihr euer Spiel kurz und bündig den anderen Gruppen präsentieren. Besprecht im Team die wichtigsten Eckpunkte, bevor das Marketing einen Kurzvortrag hält. Warum sollte man gerade euer Spiel spielen? Was ist euch besonders gut gelungen? Was war besonders schwierig und wie konntet ihr es lösen? Was würdet ihr als nächstes ins Spiel einbauen wollen, wenn ihr mehr Zeit hättet?

Stellt euch gemeinsam den kritischen Fragen des Publikums.



## Feedback

Ist gerade eine andere Gruppe am Präsentieren, notiere dir einige Stichpunkten für die Fragerunde. Was hat dir am Spiel besonders gefallen? Was passiert, wenn im Spiel ... passiert, wurde das von den Entwicklern bedacht? Warum wurde etwas auf die gezeigte Art gelöst und nicht anders? Wäre es möglich noch ... einzubauen?

### Gruppe 1:

---



---



---



---



---



---

### Gruppe 2:

---



---



---



---



---



---

### Gruppe 3:

---



---



---



---



---



---

### Gruppe 4:

---



---



---



---



---



---

### Gruppe 5:

---



---



---



---



---



---

### Gruppe 6:

---



---



---



---



---



---

Für die Vorbereitung oder ggf. als Hilfestellung am Ende der Arbeitsphase in den Expertengruppen finden sie unter <https://scratch.mit.edu/studios/4033910> Musterlösungen (Scratch-Studio mit allen 5 Teilprojekten).



<https://scratch.mit.edu/projects/157158235>

- Sollen Vokabeln zu einem konkreten Thema verwendet werden, können passende Bilder vorgängig zum Beispiel bei [pixabay.com](http://pixabay.com) herausgesucht werden (als Datei heruntergeladen und in Scratch wieder importieren).
- Auf Internetseiten wie [audiorix.de](http://audiorix.de) oder [www.freesound.org](http://www.freesound.org) können weitere Klänge als MP3 heruntergeladen und in Scratch importiert werden.



<https://scratch.mit.edu/projects/157655997>

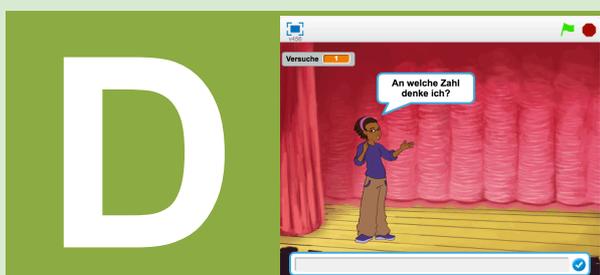
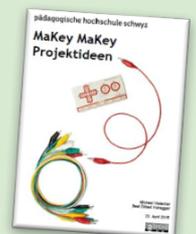
- eventuell möchte die Gruppe auch ein gegenseitiges Abprallen der Bälle einbauen. Hier sollte die LP empfehlen, auf das später folgende Spiel zu warten, da dieses Problem das Wissen mehrerer Expert/innen erfordert.
- Die Gruppe sollte unbedingt das Koordinatensystem von Scratch und seine Dimensionen kennen. Die LP kann kleine Kontrollfragen am Gruppentisch stellen: Welche Koordinate hat der linke Rand? Wird die Y-Koordinate nach oben hin grösser oder kleiner? usw. → Bühnenbild „xy-grid“ als Hilfe verwenden



<https://scratch.mit.edu/projects/157405689>

- Im ersten Teil wird das Mikrofon verwendet. Ist der Geräuschpegel im Raum allgemein sehr hoch, kann das die Aufgabe behindern.
- Wir empfehlen vorgängig (insbesondere für den Bereich Bedingungen und Sensoren) mit der Klasse ein oder mehrere Projekte mit MaKey-MaKey und Scratch zu realisieren. Mehr dazu in der separaten Broschüre:

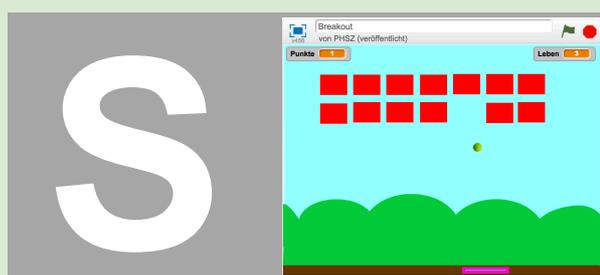
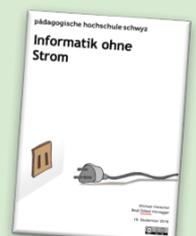
[www.iLearnIT.ch/makey](http://www.iLearnIT.ch/makey)



<https://scratch.mit.edu/projects/157870921>

- Das Hineinziehen von Blöcken als Parameter in andere Blöcke sollte ggf. vorgezeigt werden.
- Zahlenraten ist ein klassisches Beispiel für Binärsuche in der Informatik. Das Thema kann vor oder nach dem Gruppenpuzzle aufgegriffen werden. Mehr dazu in der separaten Broschüre:

[www.iLearnIT.ch/stromlos](http://www.iLearnIT.ch/stromlos)



<https://scratch.mit.edu/projects/157900135>

- Besonders schwierig ist das Abprallen des Balls von Paddle und Steinen. Je nach Lösung wird das Abprallen vom Stein und das Verschwinden des Steins gleichzeitig, jedoch in zwei Skripten passieren. Hier kann eine Pause von 0.01s helfen, um ein Ereignis garantiert vor dem anderen auszulösen.
- Besonders starke Gruppen können für die Steine das „Klon“-Konzept in Scratch ausprobieren.
- Wenn eine Gruppe sehr viele Steine dupliziert, sollte die LP unbedingt die Kollisionsprüfung über den Farbttest empfehlen, da sonst zu viele „falls Stein-n berührt?“ Blöcke nötig werden.

## Worum geht es?

Das Gruppenpuzzle ist ein möglicher Zugang zur Einführung ins Programmieren mit Scratch. Die Teilnehmenden lernen in 4 Gruppen:

- **Gestaltung:** Kostümwechsel, Sound, Musik
  - **Bewegung:** anderen Objekten folgen,
  - **Sensoren:** Tastatursteuerung, Kollisionsprüfung
  - **Variablen:** Inputs, Zähler, Operatoren
- kennen, um danach gemeinsam ein Spiel (Breakout) zu programmieren.

## Was wird benötigt?

Ein Computer mit Webbrowser und aktueller Flash-Version, sowie Lautsprecher/Kopfhörer und Mikrofon. Für Windows/MacOs/Linux gibt es eine Online- und eine Offline-Version. Wir empfehlen die Nutzung der Online-Variante: [scratch.mit.edu](http://scratch.mit.edu)

## Warum Scratch?

Scratch ist eine am MIT entwickelte, frei verfügbare Programmierumgebung für Kinder und Jugendliche nach dem «low floor – wide walls – high ceiling»-Prinzip (siehe unten). Zu Scratch gibt es viele Unterrichtsmaterialien, Bücher und auf [scratch.mit.edu](http://scratch.mit.edu) über 21 Millionen öffentliche Programmbeispiele.

Scratch ist mehr als eine Programmiersprache. Die Scratch-Community mit über 15 Millionen Teilnehmenden ist ein soziales Netzwerk, über das Lernende mit anderen Lernenden kommunizieren und ihre Entwicklungen austauschen, kommentieren und weiterentwickeln.

## 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 / Verschiedene Zugangsweisen:** Mit Scratch lassen sich ganz unterschiedliche Projekte realisieren – animierte Geschichten, Simulationen, Spiele usw. Scratch-Projekte verbinden Kompetenzen unterschiedlicher 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 Sensoren, Roboter und andere Elektronikgeräte mit Scratch verbinden. Damit lässt sich die Funktionspalette praktisch beliebig erweitern.

## Die Idee des Konstruktivismus

Scratch wurde nach der Idee des Konstruktivismus des Mathematikers und Piaget-Studenten Seymour Papert 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.*

## Broschüren

Diese und weitere Broschüren stehen kostenlos zum Download bereit:

[www.iLearnIT.ch/puzzle](http://www.iLearnIT.ch/puzzle)  
[www.iLearnIT.ch/makey](http://www.iLearnIT.ch/makey)  
[www.iLearnIT.ch/scratch](http://www.iLearnIT.ch/scratch)  
[www.iLearnIT.ch/stromlos](http://www.iLearnIT.ch/stromlos)  
[www.iLearnIT.ch/ozobot](http://www.iLearnIT.ch/ozobot)



## Links & Literatur

[doebe.li/w561](http://doebe.li/w561) Konstruktivismus  
[doebe.li/b5341](http://doebe.li/b5341) Buch “Invent to Learn”  
[doebe.li/b130](http://doebe.li/b130) Buch “Mindstorms”

## Impressum / Kontakt

Pädagogische Hochschule Schwyz  
 Michael Hielscher / Beat Döbeli  
[michael.hielscher@phsz.ch](mailto:michael.hielscher@phsz.ch)  
[beat.doebeli@phsz.ch](mailto:beat.doebeli@phsz.ch)