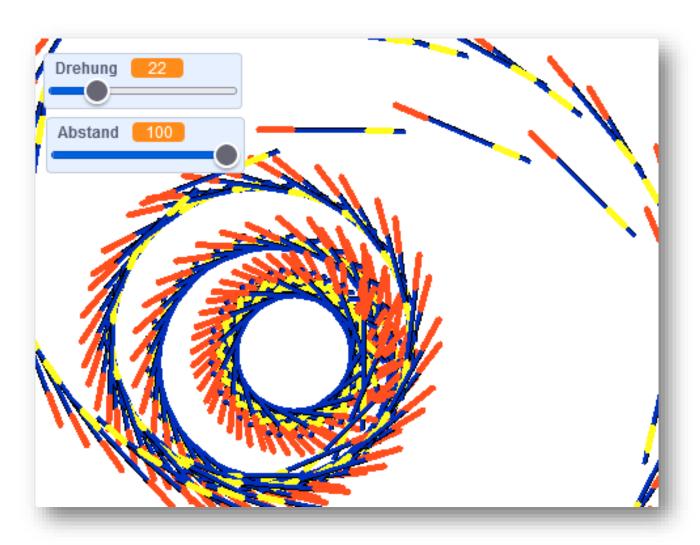


Gerhard Brandhofer

# **Coding und Making** mit Scratch und MakeyMakey

Mit Beiträgen von Bernhard Abfalter, Sabine Apfler, Moritz Brinnich, Elieser Grill, Duy Ho David Tran, Natalie Traxler und Andreas Weiner.





# **Inhalt**

Allgemeines zu Scratch
EIS Karten Scratch
Lösungen zu den EIS Karten Scratch
MakeyMakey Beispiele

# Allgemeines zu Scratch

Scratch ist eine visuelle Programmiersprache, sie wurde von der Lifelong Kindergarten Group am MIT entwickelt. Viele neuere erziehungsorientierte Programmiersprachen nehmen sich das Design von Scratch als Vorbild (Snap!, Ozoblockly, Touchdevelop).

Scratch ist so einfach in der Bedienung, dass bereits Volksschulkinder damit erste Erfahrungen im Programmieren machen können. Scratch läuft ab Version 2 im Browser, eine Installation ist nicht nötig. Die einzelnen Befehle werden per Drag & Drop zu Sequenzen verbunden, eine kindgerechte Bedienung steht im Vordergrund. Die Projekte werden online gespeichert und können veröffentlicht werden, so sind erste Ergebnisse schnell online verfügbar, aber auch ein lokales Speichern ist möglich.

```
Wenn angeklickt

gehe zu x: 0 y: 0

setze Richtung auf 0 wische Malspuren weg

schalte Stift ein

wiederhole 30 mal

schalte Stift aus

gehe 10 er-Schritt

drehe dich ( um 8 Grad

schalte Stift ein

wiederhole 180 mal

gehe 2 er-Schritt

drehe dich ( um 2 Grad
```

Der Editor beinhaltet die Bereiche Bühne, Objektkatalog, Detailansicht (Skripte, Klänge, Kostüme) und die Objektbibliothek. Sowohl Bühne als auch die einzelnen Objekte können über den Skriptbereich programmiert werden, die Befehle umfassen unter anderem Bewegung, Aussehen und Klänge aber auch Daten können verarbeitet werden, mit Variablen kann operiert werden und ab Version 2 ist auch die Erstellung eigener Blöcke möglich. Mit dieser Vielfalt an Zugangsmöglichkeiten und der Multimedialität lädt Scratch zum spielerischen Forschen ein.

Den Scratcheditor kann man ohne Anmeldung ausprobieren. Zusätzlich ist es möglich, einen Blick in den Code veröffentlichter Projekte zu werfen. Auch auf diese Weise kann man sich Ideen und Anregungen holen. Für große Projekte ist diese Plattform weniger geeignet, der primäre Zweck ist, den Einstieg ins Programmieren so einfach und kindegerecht wie möglich zu machen. Ein rechtzeitiger Umstieg auf andere – mächtigere – Programmierumgebungen sollte daher eingeplant werden, im Anschluss bietet sich etwa Python an. Der Scratcheditor ist für die Arbeit am Desktop optimiert, für Tablets eignet sich die weniger mächtige App ScratchJr.

Link: <a href="http://scratch.mit.edu/projects/editor/">http://scratch.mit.edu/projects/editor/</a>



eis.ph-noe.ac.at

Karte Scratch #1

# Schmetterling, flieg!



eis.ph-noe.ac.at/scratch1

### Aufgabe:

Lass einen Schmetterling über die Bühne fliegen! Wenn er den Rand berührt, soll er beim Weiterfliegen nicht auf dem Kopf stehen. Füge im Hintergrund eine Blumenwiese ein.

Erweitere das Programm mit eigenen Ideen!





## eis.ph-noe.ac.at

Karte Scratch #2

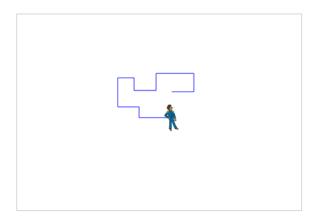
### Fritz zeichnet



eis.ph-noe.ac.at/scratch2

## Aufgabe:

Ein kleines Männchen soll mit den Pfeiltasten über die Bühne gesteuert werden und dabei eine Spur hinterlassen!



Erweitere das Programm mit eigenen Ideen!



### eis.ph-noe.ac.at

Karte Scratch #3

### **Mein Aquarium**



eis.ph-noe.ac.at/scratch3

### Aufgabe:

Baue dir dein virtuelles Aquarium! Wähle einen passenden Hintergrund. Füge dann einen Fisch als Objekt ein. Dieser soll im Aquarium hin und her schwimmen und dabei sein Maul öffnen und schließen.

Füge weitere Fische ein, den Code kannst du vom ersten Fisch kopieren und dann anpassen.



#### Erweiterungen:

- Die Fische sollen ihre Größe oder ihre Farbe ändern.
- Ein Hai zieht seine Runden und frisst die kleineren Fische.
- Ergänze eigene Ideen!





eis.ph-noe.ac.at

Karte Scratch #4

### Musizieren mit Scratch



eis.ph-noe.ac.at/scratch4

### Aufgabe:

In der Erweiterung Musik findest du alle Befehle, um ein Lied zusammen zu stellen. Versuche mit diesen Befehlen ein einfaches Lied zu schreiben! Lege Geschwindigkeit und Instrument gleich am Beginn des Skripts fest.

#### Zusätzliche Aufgaben:

- Die Geschwindigkeit soll sich während des Liedes erhöhen.
- Verwende Schleifen für Wiederholungen!
- Erfinde einen Tanz, der gleichzeitig zum Lied aufgeführt wird.
- Ergänze eigene Ideen!









eis.ph-noe.ac.at

Karte Scratch #5

### Der wandelnde Geist

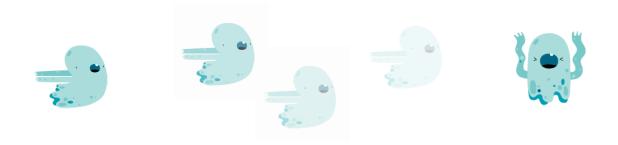
Lösung:



eis.ph-noe.ac.at/scratch5

### Aufgabe:

Füge aus der Bibliothek eine Geistfigur ein (oder zeichne selbst eine mit dem Maleditor). Programmiere die Figur so, dass sie jedes Mal, wenn sie angeklickt wird, verschwindet und nach kurzer Zeit an anderer Stelle wieder erscheint.



**Erweiterungen**: der Geist soll allmählich verschwinden und schrittweise an anderer Stelle wieder auftauchen. Er soll seine Größe und Form verändern. Gestalte den Hintergrund!





und ein Mikrofon

### eis.ph-noe.ac.at

Karte Scratch #6

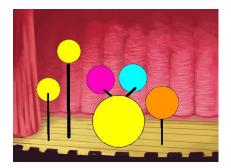
# **Mein Schlagzeug**



eis.ph-noe.ac.at/scratch6

### Aufgabe:

Mit Scratch kannst du wunderbar musizieren! Baue dir dein eigenes Schlagzeug in Scratch: zeichne zuerst mit dem Maleditor einige Formen. Jede dieser Formen soll einen bestimmten Klang abspielen, wenn du sie anklickst. Du kannst dafür auch selbst Geräusche aufnehmen, die dann abgespielt werden.



**Erweiterungen:** damit du nicht immer mit der Maus die einzelnen Formen anklicken musst, ändere das Programm so, dass du das Schlagzeug mit deiner Tastatur spielen kannst.





### eis.ph-noe.ac.at

Karte Scratch #7

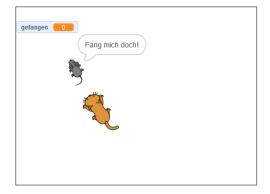
#### **Katz und Maus**



eis.ph-noe.ac.at/scratch7

### Aufgabe:

Die Katze soll eine Maus fangen! Hole dir aus der Bibliothek eine Katze und eine Maus. Die Maus neckt immerzu die Katze "Fang mich doch!" und läuft davon. Die Maus bewegt sich immer in die Richtung des Mauszeigers. Wenn die Katze die Maus erwischt, soll sie "Hab dich!" rufen.



#### Varianten:

Ändere dein Programm so, dass die Katze die Maus häufiger fängt.

Ändere dein Programm so, dass die Katze die Maus fast nie erwischt.

Baue einen Punktezähler ein.

Lass dir die Spur der beiden Tiere anzeigen.

etc. etc.





Sozialform:

Dauer: Medium:



0

Material: http://scratch.mit.edu/ ein eigenes Foto und ein Mikrofon.

eis.ph-noe.ac.at

Karte Scratch #8

### Mit dem Fischauge betrachtet





eis.ph-noe.ac.at/scratch8

### Aufgabe:

Verwende ein Foto von dir selbst. Je nachdem, wo die Maus steht, soll das Bild mehr oder weniger wie durch ein Fischauge betrachtet aussehen. Du kannst auch einen Wirbeleffekt verwenden.



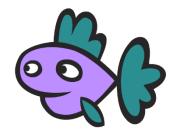




#### Variante:

Ändere dein Programm so, dass das Bild verwirbelt, wenn du laut sprichst und rot wird, wenn du schreist!

Versuche auch andere Effekte!





eis.ph-noe.ac.at

Karte Scratch #9

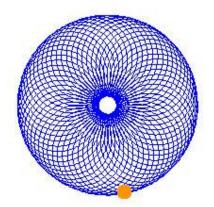
### Störe meine Kreise nicht!



eis.ph-noe.ac.at/scratch9

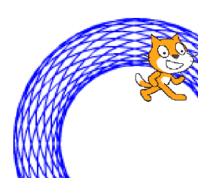
### Aufgabe:

Verwende den Malstift und versuche Muster aus Kreisen auf die Bühne zu bringen!



#### Varianten:

Du kannst mit Farben und Stiftstärken variieren! Du kannst anstelle von Kreisen auch andere geometrische Formen verwenden. Mit Variablen kannst du die Konstruktion jedes Mal anders gestalten.





### eis.ph-noe.ac.at

Karte Scratch #10

### Zufallsbilder

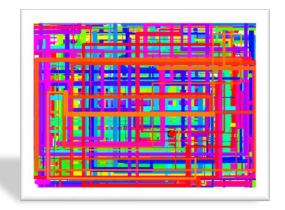




eis.ph-noe.ac.at/scratch10

### Aufgabe:

Verwende den Malstift und die Zufallszahlen um immer wieder neue Bilder auf die Bühne zu zaubern!

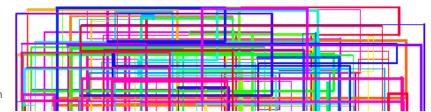


#### Varianten:

Du kannst mit Farben und Stiftstärken variieren!

Mit Variablen kannst du die Konstruktion jedes Mal anders gestalten.

Versuche Erweiterungen zu deinem Programm: das Zufallsbild ist von der Position des Mauszeigers abhängig, usw.







eis.ph-noe.ac.at

Karte Scratch #11

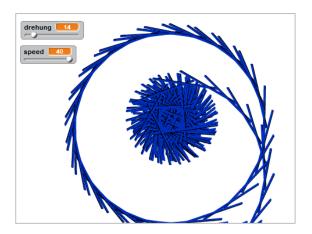
### Variable verwenden



eis.ph-noe.ac.at/scratch11

### Aufgabe:

In der Registerkarte Daten kannst du Variable definieren und verwenden. Erstelle zwei neue Variable. Zeichne einen einfachen Stab. Nutze die Funktion "Abdruck hinterlassen" und deine zwei Variablen um Muster zu erzeugen!



#### Variante:

Du kannst auch für die Farbe eine Variable verwenden!



eis.ph-noe.ac.at

Karte Scratch #12

# **Hier spricht dein Chatbot!**

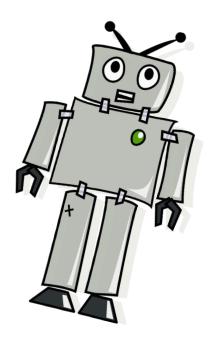


eis.ph-noe.ac.at/scratch12

### Aufgabe:

Programmiere einen einfachen Chatbot! Ein Chatbot stellt dir eine Frage. Je nachdem, was du auf die Frage antwortest, bekommst du eine bestimmte Rückmeldung und eine neue Frage.

Nutze dafür den Befehl Frage und die falls, dann – sonst Schleife.





### eis.ph-noe.ac.at

Karte Scratch #13

### **Ein Zufallstextgenerator**



eis.ph-noe.ac.at/scratch13

### Aufgabe:

Sicherlich kennst du das Knickspiel – oder? Das geht so: Der erste Mitspielende schreibt eine Zeitangabe, der zweite ein transitives Verb (essen, gehen, tragen, ...) der dritte eine Person, der vierte ein Ding im Akkusativ (4. Fall) auf einen Zettel in Spalten. Daraus entstehen seltsame Sätze.

#### Ein Beispiel:

Jeden Morgen -- zerreißt -- Herr Dimpflinger -- seinen Schlafrock.

Versuche in Scratch einen Zufallstextgenerator zu bauen. Verwende dafür Listen aus denen durch Zufall je ein Element ausgewählt wird. Die ausgewählten Elemente werden zu einem Satz zusammengefügt.



#### Varianten:

- Nutze mehr Listen.
- Der nächste Zufallssatz soll erscheinen, wenn geklatscht wird.



> http://fahrschule.wikia.com/wi ki/Anhalteweg

eis.ph-noe.ac.at

Karte Scratch #14

### In der Fahrschule

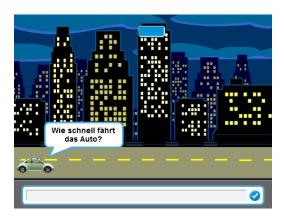




eis.ph-noe.ac.at/scratch14

### Aufgabe:

Dein Programm soll den Anhalteweg eines Fahrzeuges berechnen, wenn man die Geschwindigkeit eingibt. Besorge dir dafür die Formeln zur Berechnung von Reaktionsweg, Bremsweg und Anhalteweg.



#### Varianten:

- Simuliere den Anhalteweg.
- Lass den Anhalteweg zuvor schätzen und vergleiche die Schätzung mit dem Ergebnis.
- Berechne den Anhalteweg für trockene Fahrbahn, Nässe und Eis.













Sozialform: Dauer: Medium:







Material: http://scratch.mit.edu/

eis.ph-noe.ac.at

Karte Scratch #15

### **Animierte Geburtstagskarte**



eis.ph-noe.ac.at/scratch15

### Aufgabe:

Mach jemandem mit einer virtuellen Geburtstagskarte eine Freude! Alle Figuren sollen sich verändern, wenn du sie anklickst.

Wähle eine Figur aus, die dir gut gefällt und lass sie "Happy Birthday!" sagen und das Kostüm ändern. Wenn du auf die Torte klickst, sollen sich die Kerzen darauf entzünden und ein Geburtstagslied soll gespielt werden.

Suche dir eine weitere Figur aus, die dir gut gefällt und setze sie auf die Karte. Auch diese Figur darf etwas sagen oder ihr Kostüm ändern. Nun füge auf der Bühne einen Text ein.

Fertig ist deine virtuelle Geburtstagskarte!





### eis.ph-noe.ac.at

Karte Scratch #16

### **Dreiecke**



eis.ph-noe.ac.at/scratch16

### Aufgabe:

Schreibe ein Programm zur Dreieckskonstruktion. Es soll ausgewählt werden können, ob ein gleichseitiges Dreieck, ein rechtwinkliges Dreieck, ein gleichschenkliges Dreieck oder ein rechtwinkliges <u>und</u> gleichschenkliges Dreieck gezeichnet wird. Dabei sollen die Dreiecke zufällig sein, also alle Werte, bei denen es möglich ist, aus Zufallszahlen gezogen werden.

#### **Zusatzaufgabe 1:**

Stelle sicher, dass die rechtwinkligen Dreiecke nicht gleichschenklig werden können und die gleichschenkligen Dreiecke nicht rechtwinklig oder gleichseitig werden können.

#### Zusatzaufgabe 2:

Entwickle das Projekt mit eigenen Ideen selbstständig weiter. Bitte mache zuvor eine Kopie des Projekts, damit auch das Originalprojekt vorhanden bleibt.

Entwickle zum Beispiel ein Spiel oder Quiz. Es könnte ein Dreieck auftauchen und es muss gelöst werden um welches Dreieck es sich handelt.

Oder: mehrere Dreiecke tauchen auf und eines passt nicht dazu.

Oder: Seitenlängen und Farben der Dreiecke können ausgewählt werden.

Und so weiter.





### eis.ph-noe.ac.at

Karte Scratch #17

### **Blumengarten**



eis.ph-noe.ac.at/scratch17

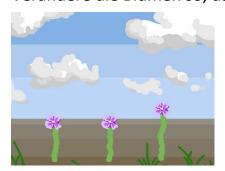
### Aufgabe:

Verwende **Zufallszahlen**, **Variablen** und **Rekursion** um eine Wachstumssimulation eines Blumengartens in Scratch zu programmieren. Die Blumen sollen eine zufällige Länge haben und unterschiedlich schnell wachsen.



#### Zusatzaufgabe:

Verändere die Blumen so, dass sich Zweige mit mehreren Blüten bilden können.



#### Tipp:

Beachte beim Erstellen von Variablen, dass "Nur für diese Figur" ausgewählt ist! Sonst haben alle Blumen dieselbe Länge.





### eis.ph-noe.ac.at

Karte Scratch #18

# **Apfelfangen**



eis.ph-noe.ac.at/scratch18

### Aufgabe:

Es ist so weit, die heutige Aufgabe wird sein, ein erstes Computerspiel zu programmieren. Ziel soll ist es, Gegenstände vom Himmel fallen zu lassen und diese aufzufangen. Eine Figur soll dazu bewegt werden können und Punkte sollen für jeden gefangenen Gegenstand gesammelt werden.



### Zusatzaufgabe:

Erweitere dein Spiel mit eigenen Ideen!





### eis.ph-noe.ac.at

Karte Scratch #19

### **Balloonhunt**



eis.ph-noe.ac.at/scratch19

### Aufgabe:

Lass Ballons über die Bühne gleiten. Sie sollen mit einem Zauberstab abgefangen werden. Wenn die Ballons den Rand berühren, sollen sie verschwinden. Füge einen passenden Hintergrund ein.



#### Zusatzaufgabe:

Füge eine Variable *Counter* ein, die die Zahl der abgefangenen Luftballons zählt.

Füge ein Variable *Countdown* (z.B. Countdown = 3) ein, die sich verringert, wenn ein

Luftballon die andere Seite berührt.

Erweitere das Programm gerne mit eigenen Ideen!





### eis.ph-noe.ac.at

Karte Scratch #20

# **Mein Labyrinth**





eis.ph-noe.ac.at/scratch20

### Aufgabe:

- Zeichne dein personalisiertes Labyrinth.
   Das Labyrinth hat einen Start und ein Ziel.
   Dies könnte in etwa so aussehen.
   (Start: oben links, Ziel: unten rechts)
- Zeichne dazu deine eigene Figur. Ihre Größe sollte so angepasst werden, dass die Figur durch die Gänge des Labyrinths passt.
- 3. Wenn die Figur die schwarze Linie ' ' berührt, dann macht sie ein beliebiges Geräusch und sie fängt wieder vom Start an.
- 4. Die Figur bewegt man mit den Pfeiltasten.
- 5. Erstelle eine Variable als Punktezähler. Beim Sammeln bekommt man entsprechende Punkte, ein Klang ertönt und der Punktegegenstand verschwindet.
- 6. Beim Ziel ertönt ein Klang und alles wird gestoppt.

#### Erweiterungen:

• Erstelle eine weitere Variable, welche die Zeit misst. Die Zeit stoppt, wenn das Ziel erreicht wird.





### eis.ph-noe.ac.at

Karte Scratch #21

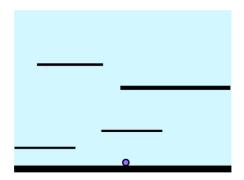
### **Der Tower Jumper**



eis.ph-noe.ac.at/scratch21

### Aufgabe:

Erstelle ein Programm in Scratch, bei dem es möglich ist, ein Objekt oder einen Avatar über die Pfeiltasten links und rechts zu steuern. Mit der Leertaste soll man springen können. Füge Objekte ein, auf die gesprungen werden kann und von denen man auch wieder auf tiefere Ebenen herunterfallen kann.



#### Variante:

Mache es möglich, dass dein Avatar durch längere gleichförmige Bewegung in eine Richtung schneller wird und dadurch auch höher springen kann.

### Zusatzaufgabe:

Die Treppenstufen sollen sich nach unten bewegen und der Spieler soll einen Punkt erhalten, wenn eine Treppenstufe den unteren Rand erreicht. Die Treppenstufen sollen immer schneller nach unten fallen und kleiner werden, wodurch es schwieriger wird vom unteren Rand weg zu bleiben. Falls der Avatar den unteren Rand berührt, ist das Spiel beendet.





### eis.ph-noe.ac.at

Karte Scratch und Python #01

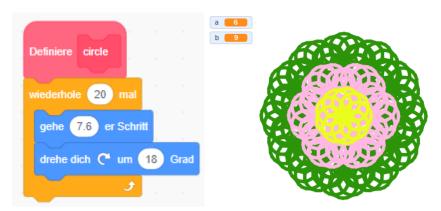
### Blume aus überlappenden Kreisen



eis.ph-noe.ac.at/scr pyth01

### Aufgabe Scratch:

Zeichne mit der Erweiterung "Malstift" eine Blume aus überlappenden Kreisformationen. Eine Kreisformation besteht auf 9 überlappenden Kreisen. Die Blume selbst besteht aus 19 überlappenden Kreisformationen. Für die Kreise kannst du die abgebildete Funktion definieren.



**Tipp**: Nutze den Turbomodus um das Zeichnen zu beschleunigen.

### Zusatzaufgabe:

Experimentiere mit unterschiedlichen Kreisgrößen, Stiftstärken und Farben, um die Blume nach deinem Geschmack anzupassen. Du kannst auch die Anzahl der Kreise und Kreisformationen variieren.





Sozialform:

Dauer:

Medium:

Material: https://docs.python.org/

### eis.ph-noe.ac.at

Karte Scratch und Python #01

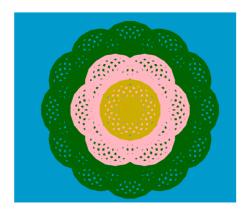
# Blume aus überlappenden Kreisen



eis.ph-noe.ac.at/scr pyth01

### Aufgabe Python:

Zeichne mit der Erweiterung "Malstift" eine Blume aus überlappenden Kreisformationen. Eine Kreisformation besteht auf 18 überlappenden Kreisen. Die Blume selbst besteht aus 19 überlappenden Kreisformationen.



### Zusatzaufgabe:

Experimentiere mit unterschiedlichen Kreisgrößen, Stiftstärken und Farben, um die Blume nach deinem Geschmack anzupassen. Du kannst auch die Anzahl der Kreise und Kreisformationen variieren.

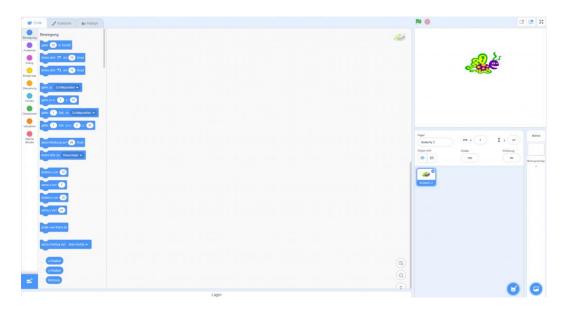


# #1 Schmetterling, flieg!

Bringe einen Schmetterling zum Fliegen!

Gehe auf scratch.mit.edu und wähle "Entwickeln"

Die Programmieroberfläche besteht aus der Bibliothek mit den Befehlen, dem Programmierbereich, der Bühne und den Figuren.



Klicke bei den Figuren mit der rechten Maustaste auf die Katze und wähle "Löschen".

Gehe anschließend auf das Katzensymbol unten – "Figur wählen". Hier kannst du dir eine Figur aus der Bibliothek auswählen. Bei den Tieren findest du mehrere Schmetterlinge, nimm einen davon.

Jetzt kannst du die Befehle aus der Bibliothek auf den Programmierbereich ziehen und zusammenstecken:



Mit dem ersten Befehl (**Drehtyp**) kann man verhindern, dass der Schmetterling beim Weiterfliegen am Kopf steht.

Fertig! Wenn du auf die grüne Flagge klickst, dann fliegt der Schmetterling über die Bühne und prallt vom Rand ab. Du kannst dich noch mit der Anzahl der Schritte spielen und mit den Gradzahlen der Drehung.

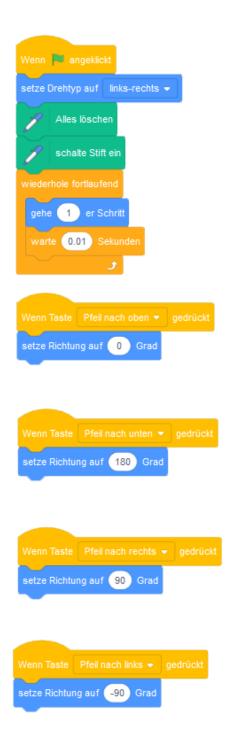
Ganz links neben den Figuren kannst du auch ein Bühnenbild auswählen ("Bühnenbild wählen") oder selbst zeichnen ("Malen").

# **#2 Fritz zeichnet**

#### Lösungsschritte:

- Lösche die vorgegebene Katze.
- Wähle eine Figur aus und verkleinere sie stark.
- Ändere gleich zu Beginn den Drehtyp der Figur auf "links-rechts".

#### **Code für Fritz:**



#### Varianten:

- Die Geschwindigkeit von Fritz wird immer höher.
- Der Stift lässt sich mit einer bestimmten Taste ein- und ausschalten.
- Eine zweite Figur wird hinzugefügt und mit d-f-e-r gesteuert.
- ...

So könnte das Ergebnis aussehen: <a href="http://scratch.mit.edu/projects/33818722/">http://scratch.mit.edu/projects/33818722/</a>

# **#3 Mein Aquarium**

Nachdem die grundlegenden Schritte in Scratch gemacht sind, können wir uns an eine etwas umfangreichere Aufgabe machen.

#### Ein Aquarium

- Den passenden Hintergrund findest du für die Bühne unter "Bühnenbild wählen". Du kannst ein Bild aus der Bibliothek wählen oder eines selbst mit dem Editor zeichnen.
- Lösche die Katze.
- Füge einen Fisch hinzu.
- Kostüme zuordnen In der Registerkarte "Kostüme" kannst du einem Objekt mehrere Kostüme zuordnen die dann wechselweise angezogen werden können. Manche Figuren haben bereits mehrere Kostüme (z.B. der Hai):



Der dazugehörige Code für einen Fisch:



Diesen Fisch kannst du jetzt mehrmals kopieren (klicke mit der rechten Maustaste auf das Objekt und wähle – "**Duplizieren"**).

Damit wird der Code gleich mitkopiert.

#### Zusatzaufgaben:

- Ändere die Kostüme der neuen Fische!
- Kostüme Bearbeiten: hier steht dir ein einfaches Malprogramm für deine Fische zur Verfügung gestalte deine Fische selbst.
- Verwende zur Belebung deines Aquariums Befehle aus dem Menü Aussehen: Farbeffekt, Größe, Helligkeit, Durchsichtigkeit.



# #4 Musizieren mit Scratch

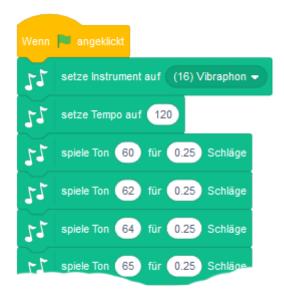
#### Mit Scratch komponieren:

In der Erweiterung "Musik" gibt es viele Werkzeuge zur Erstellung eigener Musikstücke.

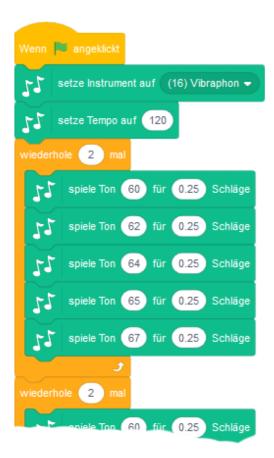
Lege zuerst Instrument und Tempo fest:



Dann kannst du dein Lied zusammen stellen:



#### Eleganter – mit Schleifen:



**Für Profis!** Musik machen mit Variablen: lege zuerst die Variable "Note" fest, ändere dann den Wert der Variable innerhalb einer Schleife und spiele den Ton "Note".



Mit Zufallszahlen kannst du dann ganz besondere Musikstücke komponieren.

Jetzt zum Tanz: Wähle eine Figur mit mehreren Kostümen aus der Bibliothek und platziere sie auf der Bühne.

Mit einem weiteren einfachen Skript kannst du sie zum Tanzen bringen:

Wenn grüne Fahne angeklickt – wiederhole fortlaufend – nächstes Kostüm – warte 0.5 Sek.



Zusätzlich kannst du noch die Lautstärke ändern, eine Schlagzeugbegleitung ergänzen u.v.m.

# **#5 Der wandelnde Geist**

Wie kann man in Scratch Figuren verstecken und wieder anzeigen lassen?

Hier die einfache Lösung:

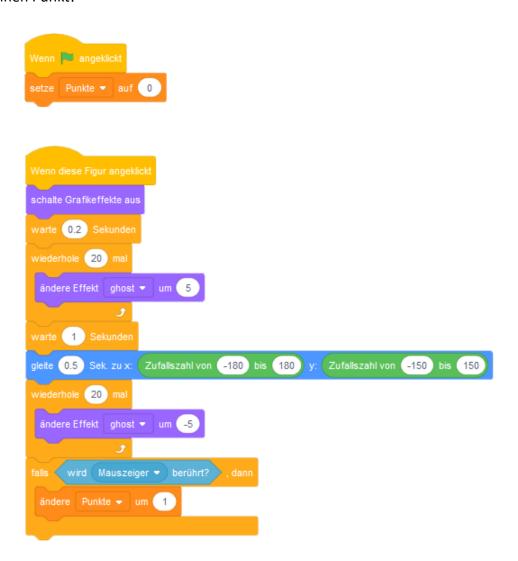
Wähle aus der Bibliothek einen Geist und einen Hintergrund (oder gestalte beides selbst!)

Der Geist:



Hier findest du das Beispiel: https://scratch.mit.edu/projects/146008610/

**Und das ist die umfangreichere Lösung**: du kannst aus dem Projekt ein kleines Spiel machen. Bewege die Maus dorthin, wo du glaubst, dass der Geist erscheint. Wenn du errätst, wo er auftaucht, dann erhältst du einen Punkt!



Mit einem Klick auf die grüne Flagge wird der Punktestand zurückgestellt. Mit jedem Klick auf den Geist bringst du ihn zum Verschwinden.

Der Link zum Beispiel: <a href="https://scratch.mit.edu/projects/145808741/">https://scratch.mit.edu/projects/145808741/</a>

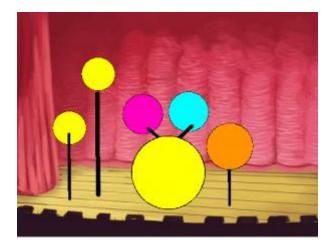
#### Varianten:

- Das Spiel soll auch ohne Klick auf den Geist einfach weitergehen, z.B. 20 Runden.
- Immer dort wo der Geist erscheint, soll eine Markierung gesetzt werden. Dann erkennt man am Ende die Zufallsverteilung.

Beispiel Nummer 3: <a href="https://scratch.mit.edu/projects/146009069/">https://scratch.mit.edu/projects/146009069/</a>

### #6 Mein Schlagzeug

Die Registerkarte **Klang** kann verwendet werden, um ein virtuelles Schlagzeug zu programmieren. Mit Hilfe des integrierten Zeicheneditors (**Figur wählen – Malen**) kannst du einige einfache Schlaginstrumente zeichnen.



Jedes dieser Elemente kannst du mit einem anderen Schlagzeugeffekt versehen.



Durch das Anklicken mit der Maus entsteht beim Schlagzeugern eine Verzögerung.

#### So spielt man mit der Tastatur:

Verwende anstelle des Kopfes "Wenn diese Figur angeklickt" diesen Kopf: "Wenn Taste *Leertaste* gedrückt" usw.

#### So kannst du die Elemente mit eigenen Aufnahmen versehen:

in der Registerkarte Klänge findest du ein Mikrofonsymbol. Wenn du ein Mikrofon hast, dann kannst du hier alle möglichen Geräusche aufnehmen und dann verwenden.

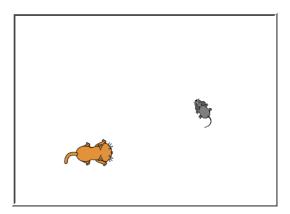
Ein Beispiel: <a href="https://scratch.mit.edu/projects/145631551/">https://scratch.mit.edu/projects/145631551/</a>

### **#7 Katz und Maus**

Aufgabenstellung: Die Katze soll eine Maus fangen!

#### Vorbereitung:

- Verkleinere die Katze oder suche eine andere Katze in der Bibiliothek.
- Füge eine Maus ein und verkleinere sie.



#### Der Code der Maus:

```
wiederhole fortlaufend

gehe 5 er Schritt

drehe dich C um 2 Grad

pralle vom Rand ab

drehe dich zu Mauszeiger
```

```
Wenn angeklickt

wiederhole fortlaufend

sende fang mich doch! ▼ an alle

warte 0.5 Sekunden

sage Fang mich doch! für 0.7 Sekunden

warte 0.3 Sekunden
```

#### Der Code der Katze:

```
Wenn angeklickt

setze gefangen v auf 0

wiederhole fortlaufend

falls wird Maus v berührt?, dann

sage Habdich für 2 Sekunden

ändere gefangen v um 1
```

```
Wenn ich fang mich doch! ▼ empfange

drehe dich zu Maus ▼

wiederhole 10 mal

gehe 7 er Schritt
```

#### Varianten:

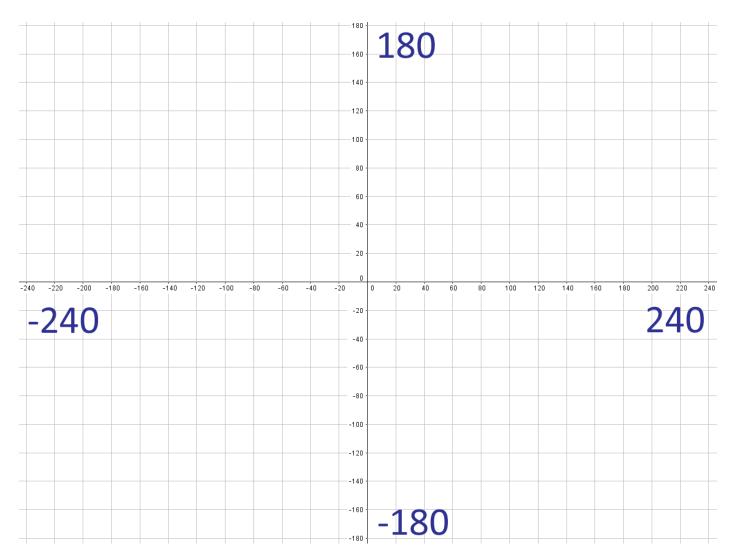
- Die Katze soll häufiger die Maus fangen.
- Die Katze soll die Maus fast nie fangen.
- Baue einen Zähler ein
- Lass die Spur der beiden Figuren anzeigen
- ...

So kann das Ergebnis aussehen: <a href="http://scratch.mit.edu/projects/62380/">http://scratch.mit.edu/projects/62380/</a>

# #8 Mit dem Fischauge betrachtet

Aufgabenstellung: Bilder verändern – je nachdem, wo die Maus steht!

Die Scratchbühne ist wie ein Koordinatensystem aufgebaut. Man kann die Koordinaten der Figuren und des Mauszeigers zum Programmieren nutzen. Die x-Achse reicht von -240 bis 240, die y-Achse von -180 bis 180.



#### Vorbereitung:

Füge ein Foto von dir selbst als Figur ein!

Der Code für das Foto ist dann ganz einfach:

```
Wenn angeklickt

wiederhole fortlaufend

setze Effekt Fischauge ▼ auf Maus y-Position

setze Effekt Wirbel ▼ auf Maus x-Position
```

Du kannst auch mit anderen Effekten experimentieren!

So könnte das Ergebnis aussehen: https://scratch.mit.edu/projects/138222999

Variante 1: Das Bild ändert sich je nach Lautstärke (dafür benötigst du ein Mikrofon).



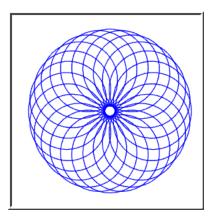
#### Variante 2:

Wenn du eine Webcam hast, kannst du den Effekt auch mit dem Befehl "Videobewegung" aus der Erweiterung "Video-Erfassung" koppeln!

### #9 Störe meine Kreise nicht!

Bei all jenen, denen es vergönnt war, in frühen Jahren (zu einer Zeit als es im Fernsehen noch das Testbild, die Bundeshymne gab und Telefone mit einem Kupferkabel mit der Welt draußen verbunden waren) Logo kennenzulernen, werden in Scratch mit der Erweiterung "Malstift" Erinnerungen an die wilden ersten Bekanntschaften mit dem PC wach werden. Mit dem Malstift können geometrische Muster gestaltet werden, mit einigen wenigen Befehlen ist dennoch eine Unzahl von Möglichkeiten verknüpft.

**Beispiel** Kreismuster – Aufgabe: Dein Avatar soll ein Kreismuster zeichnen – wie abgebildet:



Lösung: Wähle zuerst einen kleinen Kreis als Figur. Das ist der Code für die Figur:



Sollte das Zeichnen zu lange dauern: unter "Bearbeiten" gibt es den "Turbomodus".

#### Variationen – Erweiterungen

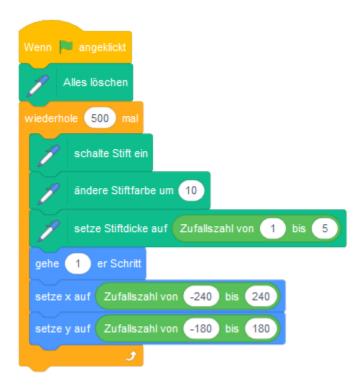
- Farbänderungen
- Schlangenlinien
- ähnliche Muster aber mit Quadraten / Sechsecken / Dreiecken etc.

### **#10 Zufallsbilder**

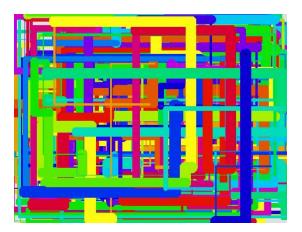
Mit Hilfe von **Zufallszahlen** lassen sich in Scratch bunte Bilder erzeugen.

Dafür benötigst du die Erweiterung "Malstift". Aktiviere diese links unten.

Nimm als Figur einen kleinen Punkt und probiere folgenden Code:



#### **Ergebnis:**



Auch hier sind die Variationsmöglichkeiten trotz der Einfachheit des Codes schier unbegrenzt. Ändere die einzelnen Werte für Schranken, Farben, Stiftdicke, Geschwindigkeit, etc., etc.

Beispiel: <a href="https://scratch.mit.edu/projects/146656/">https://scratch.mit.edu/projects/146656/</a>

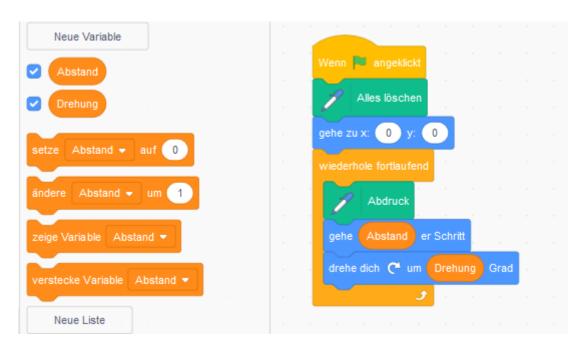
### #11 Variable verwenden

**Variable** definieren: -> Register **Daten** -> **Neue Variable** -> Namen festlegen -> fertig.

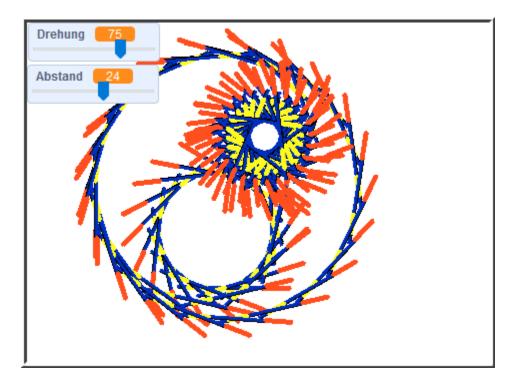
Für folgendes Projekt werden 2 Variable benötigt, benenne sie mit "Abstand" und "Drehung".

Verwende als Figur eine einfache Linie, diese kannst du selbst zeichnen.

#### 7 Zeilen Code reichen:



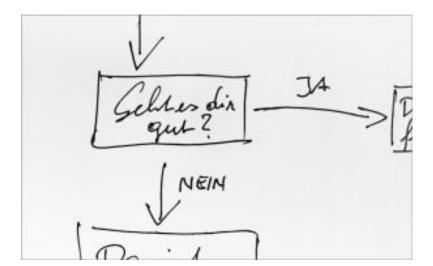
Durch Betätigen der Regler lassen sich die Variablen verändern. Das Ergebnis mit einer bunten Linie:



Beispiel: https://scratch.mit.edu/projects/33952310/#player

# **#12 Hier spricht dein Chatbot!**

Zeichne für deinen Chatbot zuerst ein Flussdiagramm.



So könnte der Code für den Chatbot aussehen:

```
Wenn 🏴 angeklickt
sage Klick mich an und sprich mit mir!
frage (Hallo! Wie heißt du?) und warte
sage verbinde Servus und name für 2 Sekunden
frage verbinde verbinde Geht es dir gut und name und ? und warte
falls Antwort = ja , danr
 wechsle zu Kostüm nano-c ▼
  sage Das freut mich! für 2 Sekunder
  sage Das ist schade! für 2 Sekunde
frage ch fliege zum Mond! Kommst du mit? und warte
      Antwort = ja , dann
  wechsle zu Kostüm nano-c 🕶
   ändere y um 10
    varte 0.1 Sekunder
   ändere y um -10
   warte 0.1 Sekunder
  sage Einen schönen Tag! für 2 Sekunder
  sage Einen schönen Tag! für 2 Sekunden
```

Hier das Beispiel: https://scratch.mit.edu/projects/146822938

Mit bestem Dank an CodeClubRik!

# **#13 Ein Zufallstextgenerator**

#### Erstelle vier Listen.

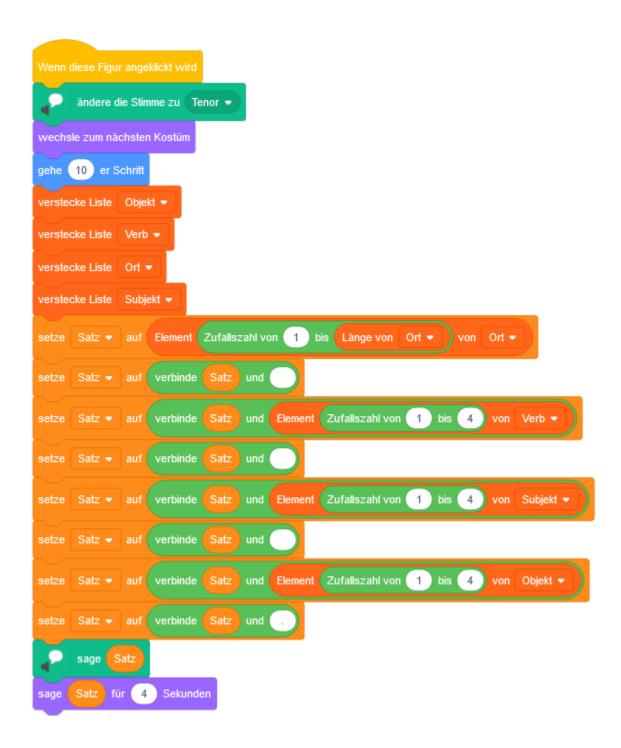
Ort, Verb, Subjekt und Objekt.

**Befülle die Listen** mit je mindestens vier Elementen – das kann dann so aussehen:



Eine Figur wählt zufällig je ein Element aus den Listen aus und setzt sie zu einem Satz zusammen. Dazwischen kommen noch Leerzeichen rein.



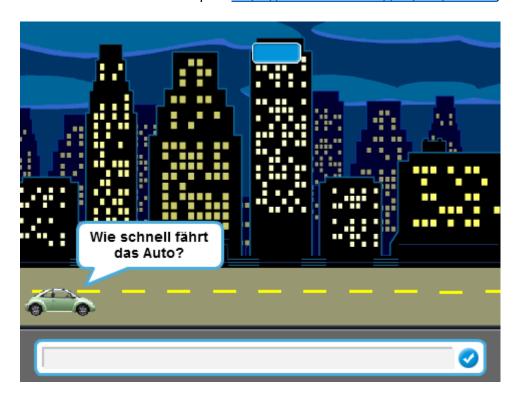


Hier das **Beispiel**: <a href="https://scratch.mit.edu/projects/146831650/">https://scratch.mit.edu/projects/146831650/</a>

### #14 In der Fahrschule

Die Formeln zur Berechnung von Reaktionsweg, Bremsweg und Anhalteweg findest du hier: <a href="http://fahrschule.wikia.com/wiki/Anhalteweg">http://fahrschule.wikia.com/wiki/Anhalteweg</a>

Und das ist das Scratch Beispiel: <a href="https://scratch.mit.edu/projects/645639/">https://scratch.mit.edu/projects/645639/</a>

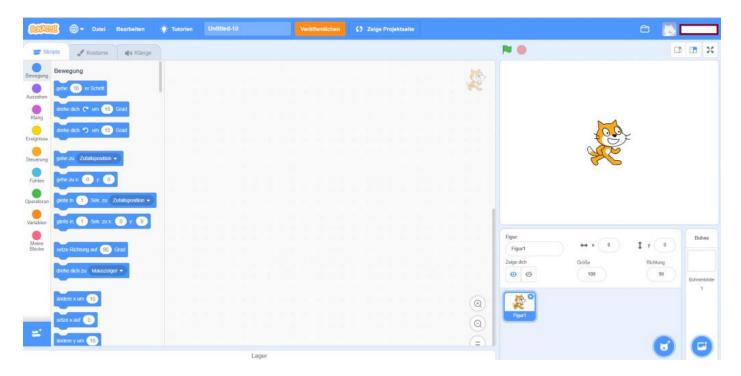


### **#15 Animierte Geburtstagskarte**

Erstelle eine virtuelle Geburtstagskarte!

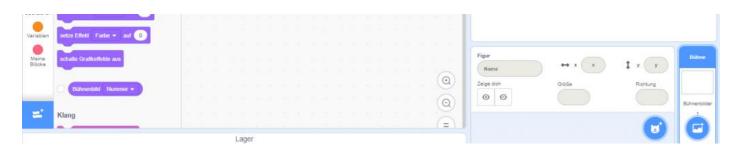
Gehe auf scratch.mit.edu und wähle "Entwickeln".

Die Programmieroberfläche besteht aus der Bibliothek mit den Befehlen, dem Programmierbereich, der Bühne und den Figuren.



Klicke bei den Figuren mit der rechten Maustaste auf die Katze und wähle "Löschen".

In der rechten unteren Ecke befindet sich das Feld "Bühnenbilder". Hier kannst du eine Bühne aus der Bibliothek auswählen.

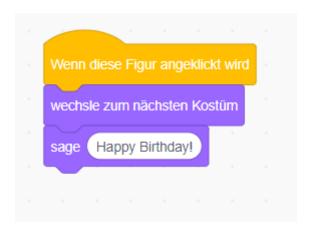


In der linken oberen Ecke findest du den Reiter Bühnenbilder. Füge mit Hilfe des Textwerkzeuges einen Text ein. Verändere ihn so wie er dir gut gefällt.

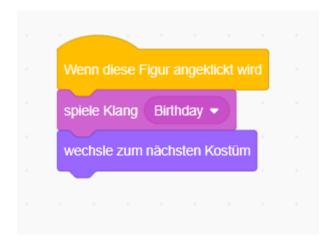


Gehe anschließend auf das Katzensymbol unten "Figur wählen". Hier kannst du dir deine Figur aussuchen.

Jetzt kannst du die Befehle aus der Bibliothek auf den Programmierbereich ziehen und zusammenstecken.



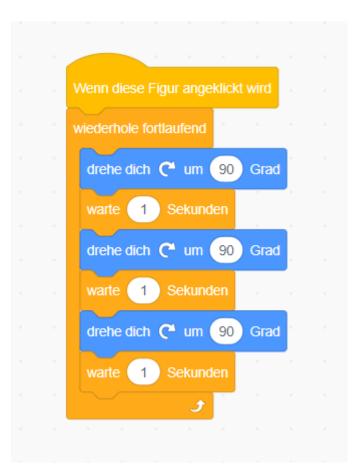
Danach füge eine weitere Figur ein, eine **Torte.** Programmiere sie, sodass sie ihr Kostüm ändert und "**Happy Birthday"** singt.



Im Anschluss füge eine weitere Figur auf deiner Bühne ein, welche sich ebenfalls bewegen soll.

#### FERTIG!

Wenn du nun auf die Figuren klickst, führen sie ihre programmierten Befehle aus.



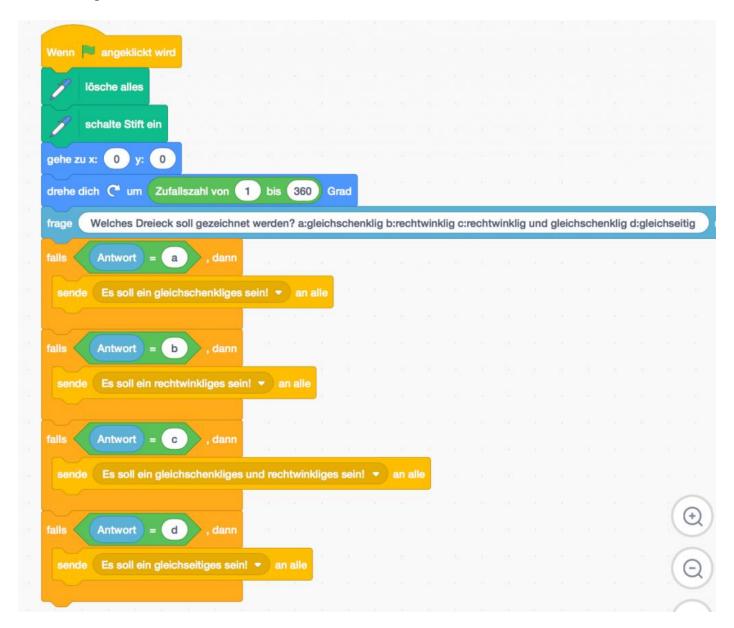
### **#16 Dreiecke**

Ein Projekt von Natalie Traxler

#### Lösung Grundaufgabe:

https://scratch.mit.edu/projects/492259970

Zum Zeichnen benötigst du die Erweiterung **Malstift**. Aktiviere diese links unten. Nimm als Figur einen kleinen Punkt.



Wähle einen Startpunkt und gegebenenfalls Startrichtung.

Frage ab, welches Dreieck gezeichnet werden soll. Je nach Antwort wird ein anderer Code für das jeweilige Dreieck gebraucht.

Dies kann gelöst werden durch die "falls (), dann" Steuerung und das Senden von Nachrichten, die jeweils als Startereignis für den jeweiligen Code genutzt werden. Durch das Nutzen der Nachrichten wird der gesamte Code auch überschaubarer.

#### **Gleichseitiges Dreieck**



Wichtig ist, dass alle Seiten gleich lang sind und die Winkel jeweils 60 Grad groß sind. Wie lang die Seiten sind, soll zufällig sein ("50 bis 200" wurde deshalb gewählt, weil das Dreieck dann eine angenehme Größe bekommt). Die drei Seiten sollen jedoch alle die gleiche Zufallszahl als Länge haben, weshalb die Variable x auf den Wert gesetzt wird. Die Drehung um 120 Grad ist nötig, damit die Figur (also der Punkt/Stiftspitze) nicht gerade geht (entspricht 0 Grad Drehung oder 360 Grad Drehung), nicht zurückgeht (entspricht 180 Grad Drehung), sondern so geht, dass ein Winkel von 60 Grad eingeschlossen wird, also 120(180-60) Grad. Da es, vor allem, wenn das Dreieck den Rand berührt, passieren kann, dass der Abschluss nicht exakt ist, wird (anstatt die Schleife einfach 3 Mal zu wiederholen) die dritte Seite gezeichnet, indem zum Startpunkt zurückgegangen wird.

#### **Rechtwinkliges Dreieck**

Wichtig ist, dass ein Winkel 90 Grad groß ist.



Wie lang die Seiten sind, soll zufällig sein ("50 bis 200" wurde deshalb gewählt, weil das Dreieck dann eine angenehme Größe bekommt).

Durch die Drehung um 90 Grad, entsteht ein 90 Grad Winkel zwischen den beiden Geraden. Da die Längen der Seiten diesmal nicht gleich sein müssen, wird ein zweites Mal eine Zufallszahl gewählt.

Damit der Abschluss exakt ist, wird die dritte Seite gezeichnet, indem zum Startpunkt zurückgegangen wird.

Da zweimal eine Zufallszahl generiert wird, kann diese Zahl jedoch auch gleich sein und ein gleichschenkliges Dreieck würde entstehen. (Siehe Zusatzaufgabe 1)

#### **Gleichschenkliges Dreieck**

Wichtig ist, dass 2 Seiten gleich lang sind.



Wie lang die Seiten sind, soll zufällig sein ("50 bis 200" wurde deshalb gewählt, weil das Dreieck dann eine angenehme Größe bekommt). Zwei Seiten sollen jedoch die gleiche Zufallszahl als Länge haben, weshalb die Variable x auf den Wert gesetzt wird. Die Größe des Winkels zwischen den zwei Schenkeln ist egal, solange es möglich ist, dass ein Dreieck entsteht (also größer als 0 Grad und kleiner als 180 Grad). ("10 bis 170" wurde deshalb gewählt, weil das Dreieck dann ansehnlicher wird als bei "0.1 bis 179.9) Damit der Abschluss exakt ist, wird die dritte Seite gezeichnet, indem zum Startpunkt zurückgegangen wird.

Da eine Drehung um 90 Grad und 120 Grad nicht ausgeschlossen wird, kann immer noch ein rechtwinkliges oder gleichseitiges Dreieck entstehen. (Siehe Zusatzaufgabe 1)

#### Rechtwinkliges und gleichschenkliges Dreieck

Wichtig ist, dass 2 Seiten gleich lang sind und ein Winkel 90 Grad groß ist.

```
Wenn ich Es soll ein gleichschenkliges und rechtwinkliges sein! ▼ empfange

setze x ▼ auf Zufallszahl von 50 bis 200

gehe x er Schritt

drehe dich C um 90 Grad

gehe x er Schritt

gehe zu x: 0 y: 0
```

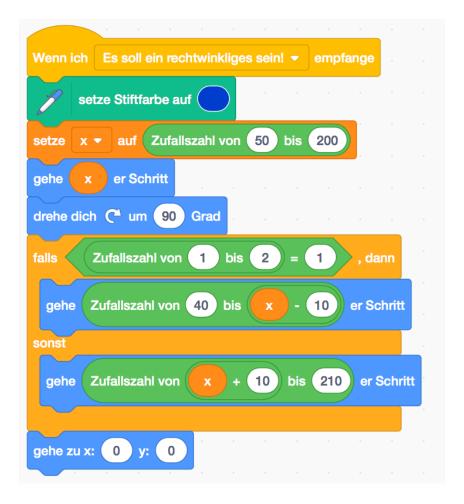
Wie lang die Seiten sind, soll zufällig sein ("50 bis 200" wurde deshalb gewählt, weil das Dreieck dann eine angenehme Größe bekommt). Zwei Seiten sollen jedoch die gleiche Zufallszahl als Länge haben, weshalb die Variable x auf den Wert gesetzt wird. Durch die Drehung um 90 Grad, entsteht ein 90 Grad Winkel zwischen den beiden Geraden. Damit der Abschluss exakt ist, wird die dritte Seite gezeichnet, indem zum Startpunkt zurückgegangen wird.

#### Lösung Zusatzaufgabe 1:

https://scratch.mit.edu/projects/491671540

#### Rechtwinkliges und nicht gleichschenkliges Dreieck

Wichtig ist, dass ein Winkel 90 Grad groß ist und die zwei Katheten nicht gleich lang sind.

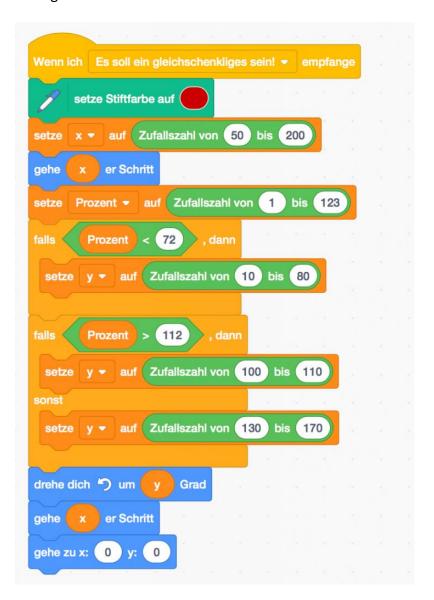


Der Code des rechtwinkligen Dreiecks wird ein wenig abgeändert.

Durch die Zufallszahl 1 bis 2 und die "falls () sonst" Steuerung wird zu 50% eine Zufallszahl größer als die Seite x und zu 50% eine Zufallszahl kleiner als die Seite x als Seitenlänge festgelegt. (Das ist natürlich nicht ganz fair, da x nicht der Median ist.) Zur besseren Erkennbarkeit wurden auch Werte im Bereich +/- 10 um den Wert x ausgeschlossen. Die Zufallszahlen wurden mit einem Minimum von 40 und einem Maximum von 210 versehen, falls der Fall eintritt, dass x genau 50 oder 200 ist.

#### Gleichschenkliges, nicht rechtwinkliges und nicht gleichseitiges Dreieck

Wichtig ist, dass 2 Seiten gleich lang sind, kein Winkel 90 Grad groß ist und kein Winkel 60 Grad groß ist.



Der Code des gleichschenkligen Dreiecks wird ein wenig abgeändert. Folgende Werte soll der Winkel zwischen den Schenkeln nicht annehmen: 0, 60, 90, 180. Also darf die Drehung nicht um folgende Werte erfolgen: 180, 90 120, 0 Zur besseren Erkennbarkeit wurden auch Werte im Bereich +/- 10 um diese Werte ausgeschlossen.

Für jeden möglichen natürlichen Wert für die Grad soll die gleiche Wahrscheinlichkeit

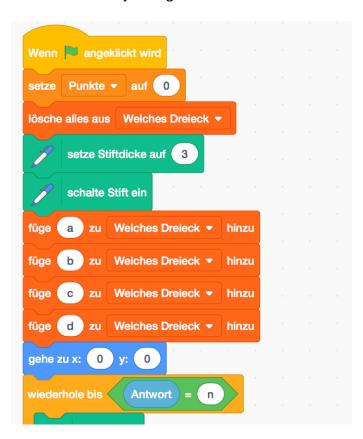
zugeschrieben sein. Es gibt 123 mögliche Werte. Durch das Ziehen der Zufallszahl von 10 bis 80, wird aus 71 Werten entschieden. Somit soll in 71 (<72) der 123 Fälle aus diesem Intervall gewählt werden. 11 mögliche Zahlen gibt es bei der Zufallszahl von 100 bis 110, somit soll bei 11 (>112) der 123 möglichen Werte aus jenem Intervall gewählt werden. In

den restlichen Fällen, also wenn die Variable "Prozent" einen der 41 Werte 72, 73, ...... 111, 112 annimmt, soll einer der 41 möglichen Werte der Zufallszahl von 130 bis 170 gewählt werden.

#### Lösung Zusatzaufgabe 2:

https://scratch.mit.edu/projects/492232937

Für das Quiz, in dem ein Dreieck erscheint, dem zugeordnet werden soll welches der 4 Dreiecke es ist und Punkte gesammelt werden können, wurde der erste Block verändert, die Codes für die jeweiligen Dreiecke sind unverändert.



#### Eine Liste wird angelegt.

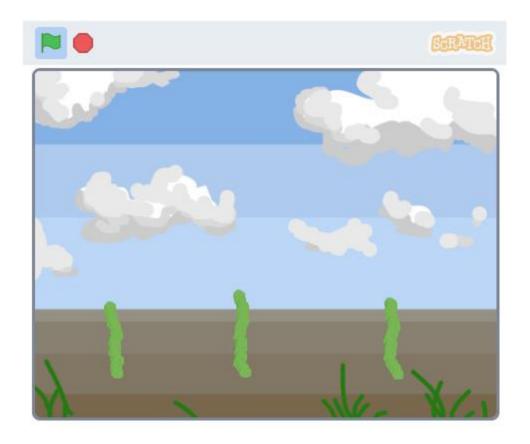
Aus dieser wird zufällig ein Element (Dreiecksart) gewählt, der Variablen "Dreieck" zugeordnet und durch das Senden der Nachricht der Code-Teil für das Zeichnen des entsprechenden Dreiecks aktiviert.

Dann wird der/die Benutzer\*in gefragt, welches Dreieck es ist. Stimmt die Antwort mit der Variable überein, wird der Punktestand um 1 erhöht. Außerdem wird abgefragt, ob man noch einmal spielen möchte und die Schleife wird wiederholt bis die Antwort n(ein) lautet.

# **#17 Im Blumengarten**

Ein Projekt von Bernhard Abfalter.

Der Link zur Lösung: <a href="https://scratch.mit.edu/projects/505248229/editor">https://scratch.mit.edu/projects/505248229/editor</a>

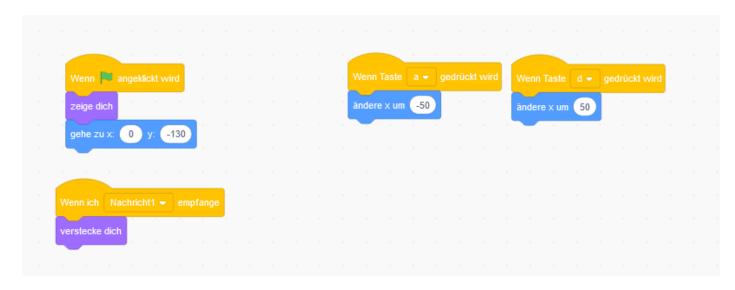


## **#18 Apfelfangen**

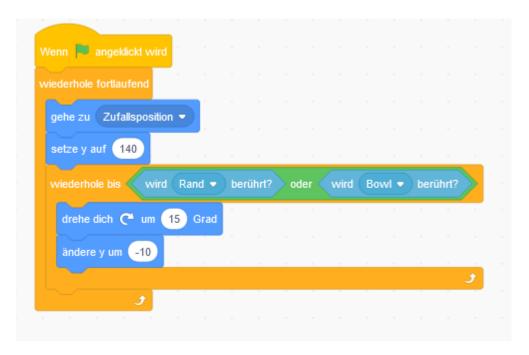
Ein Projekt von Moritz Brinnich.

Es werden mindestens 2 Figuren benötigt. Eine Figur ist zum Beispiel ein Korb, die andere ein zu fangender Gegenstand.

In der einfachsten Variante passiert mit dem Korb nur folgendes:



Kommen wir zum Gegenstand, den wir fangen möchten:



#### Punkte zählen:

Um die Punkte zählen zu können, müssen wir eine Variable anlegen und diese dann in der oberen Schleife verändern, wenn der Korb berührt wird.

```
Wenn angeklickt wird

setze Punkte → auf 0

wiederhole fortlaufend

gehe zu Zufallsposition →

setze y auf 140

wiederhole bis wird Rand → berührt? oder wird Bowl → berührt?

drehe dich C um 15 Grad

ändere y um -10

falls wird Bowl → berührt? , dann

ändere Punkte → um 1
```

#### **Erweiterung**:

```
Wenn angeklickt wird

setze Punkte auf 0

wiederhole fortlaufend

gehe zu Zufallsposition

setze y auf 140

wiederhole bis wird Rand berührt? oder wird Bowl berührt?

drehe dich C um 15 Grad

ändere y um -10

falls wird Bowl berührt? , dann

ändere Punkte um 1

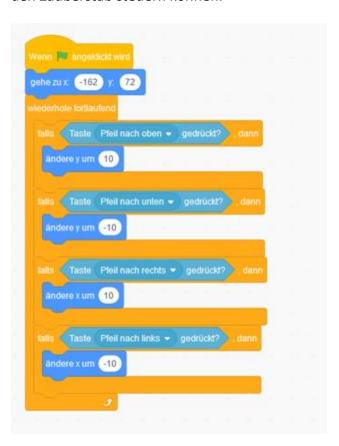
falls wird Rand berührt? , dann

stoppe alles v
```

### **#19 Balloonhunt**

Ein Projekt von Elieser Grill.

Zu Beginn machen wir die Steuerung für unseren Zauberstab. Mit dem ersten Befehl legst du die Startposition fest. Danach folgen Befehle, mit denen wir die Pfeiltasten so programmieren, dass wir damit den Zauberstab steuern können.



```
verstecke dich

wiederhote fortlaufend

falts Maustaste gedrückt? , dann

erzeuge Klon von mir selbst *

warte 0.3 Sekunden

Wenn ich als Klon entstehe

gehe zu x x-Position * von Magic Wand * y, y-Position * von Magic Wand *

zeige dich

wiederhote bis wird Rand * berührt?

gehe 10 er Schritt

losche diesen Klon
```

Als Nächstes befassen wir uns mit dem Stern, der unser Projektil darstellt. Der erste Befehl sorgt dafür, dass der Stern zu Beginn nicht sichtbar ist. Dann legen wir fest, dass der Stern, sobald die Maustaste gedrückt wird, einen Klon von sich selbst erzeugt.

Nun folgt der Teil, der festlegt, dass der Stern nicht auf der Stelle bleibt. Sobald ein Klon entsteht, legen wir dessen Startposition fest. Damit es so aussieht, als würden die Sterne aus dem Stab kommen, legen wir als Position des Sterns die des Zauberstabes fest. Danach sorgen wir mit dem Befehl "gehe x- er Schritte" dafür, dass sich der Stern bewegt. Um zu bewerkstelligen, dass diese Bewegung bis zum anderen Bildschirmrand gemacht wird, packen wir um den Schritte-Befehl eine Schleife herum, die als Bedingung "wird Rand berührt" hat. Sobald dies der Fall ist, soll der Stern verschwinden, daher auch der "Lösche diesen Klon"-Befehl.

```
wiederhole fortlaufend

gehe zu x (214) y Zufallszahl von (129) bis (-157)

wechsle zum nächsten Kostüm

erzeuge Klon von mir selbst (129)

warte (0.3) Sekunden

Wenn ich als Klon entstehe

zeige dich

wiederhole fortlaufend

andere y um Zufallszahl von (5) bis (-5)

andere x um Zufallszahl von (-3) bis (-10)

pralle vom Rand ab

warte (0.2) Sekunden

falls (129) bis (-157)

talls (129) bis (-157)

talls (129) bis (-157)

wechsle zum nächsten Klon

talls (129) bis (-157)

tall (129) bis (-157)

wechsle zum nächsten Klon

tall (129) bis (-157)

wechsle zum nächs
```

Der letzte Schritt sind die Ballons. Zu Beginn legen wir eine Startposition am rechten Rand fest. Damit der Ballon nicht immer die gleiche Farbe hat, wechseln wir im nächsten Schritt das Kostüm. Anschließend erzeugen wir einen Klon.

Sobald ein Ballon entsteht, müssen wir wie beim Stern dafür sorgen, dass er sich über den Bildschirm bewegt. Hier kommen ein weiteres Mal eine Schleife und Zufallszahlen zum Einsatz. Die Zufallszahlen sollten dabei in einem kleinen Bereich liegen, da dieser Bereich die Größe der Schritte, die ein Ballon macht, festlegt. Um die Geschwindigkeit, mit der sich die Ballons fortbewegen, etwas zu drosseln, fügen wir auch hier einen Wartebefehl ein. Abschließend fehlen nur noch die Bedingungen, die die Ballons zum Verschwinden bringen. Unsere Bedingungen sind in diesem Fall, die Berührung des Sterns oder die des Randes.

Besipielprojekt: <a href="https://scratch.mit.edu/projects/505300129/">https://scratch.mit.edu/projects/505300129/</a>

# **#20 Mein Labyrinth**

Ein Projekt von David Tran.

Link zur Lösung: https://scratch.mit.edu/projects/505719565/

Um dein persönliches Labyrinth zu zeichnen, kannst du unten rechts auf "Bühnenbilder" klicken. Oben links kannst du dann im Reiter "**Bühnenbilder**" dein eigenes Labyrinth zeichnen.

#### Code der Figur:



#### Der Code des Ziels:

```
wederhole felfassend

talls wild Fressmaschine berührt? dann

spiele klang sieg v ganz

stoppe sies v
```

#### Der Code einer Belohnung:

```
wind section of the s
```

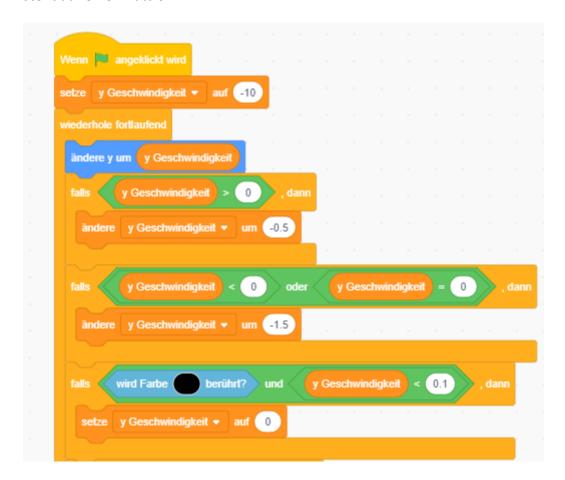
#### **Erweiterung:**

```
Wenn angeklickt wird
setze Zeit v auf 0
wiederhole fortlaufend
warte 1 Sekunden
lindere Zeit v um 1
```

# **#21 Der Tower Jumper**

Ein Projekt von Andreas Weiner.

Implementiere zuerst eine "Schwerkraft", die auf deinen Avatar wirkt, außer dieser steht auf einer Plattform:



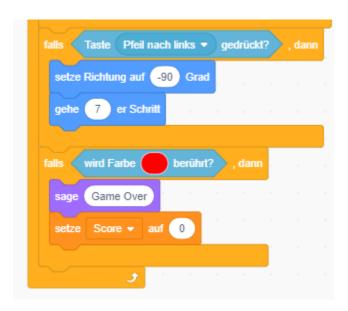
Dann kannst du die verschiedenen Bewegungsformen durch die Tastenbelegungen definieren:

#### Lösung für die Zusatzaufgabe:

Sorge zunächst dafür, dass sich die Blöcke nach unten bewegen und jeweils einen Punkt bringen, wenn sie den unteren Rand erreichen. Dann sollen sie am oberen Rand irgendwo links oder rechts erscheinen.



Nun kann man ein rotes Objekt am gesamten unteren Spielfeldrand platzieren, durch dessen Berührung das Spiel beendet wird:



## #01 Blume aus überlappenden Kreisen

Ein Projekt von Markus Kautz, für Scratch und Python.

#### Lösung Scratch:

Diese Aufgabe lässt sich mit der Erweiterung "Malstift" lösen. Für eine einfache Kreisformation kannst du den "Wiederholen bis"-Block aus dem Bereich Steuerung nutzen:



Zeichne die Blume am besten von außen nach innen. Überlege, welche Winkel du dazu benötigst und an welche Position du den Stift schiebst. Vergiss nicht, beim Bewegen den Stift zu deaktivieren.

In der Beispiellösung wird der Stift nach dem Zeichnen jeder Kreisformation in die Mitte zurückbewegt.







```
Definiere circle

wiederhole 20 mal

gehe 7.8 er Schritt

drehe dich C* um 18 Grad
```

#### Lösung Python:

Zeichne die Blume am besten von außen nach innen. In der Beispiellösung wurden Kreise mit Radius 50 verwendet. Achte darauf, die Abstände an deine gewählte Kreisgröße anzupassen. Die Farben wurden mit tk-color-strings angegeben. Du kannst aber auch RGB Werte verwenden und so zum Beispiel mit "randint" Zufallsfarben erzeugen.

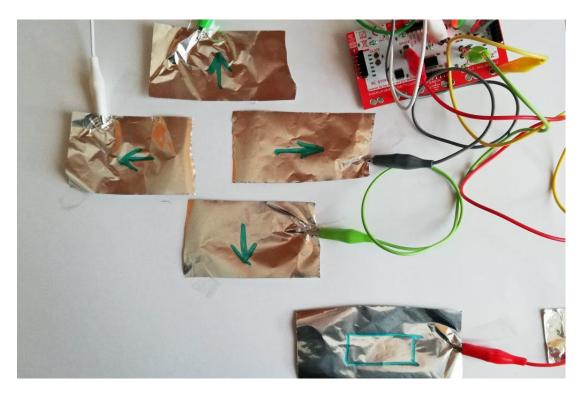
```
from turtle import *
window = Screen()
window .colormode (255)
window_.bgcolor ("DeepSkyBlue3")
window .title ("Turtle")
aaa = Turtle()
aaa.shape("turtle")
aaa.color("black")
aaa.pensize(10)
aaa.speed(0)
j = 1
aaa.color("DarkGreen")
while (j \le 12):
   aaa.penup()
    aaa.forward(200)
    aaa.pendown()
    i = 1
    while(i <= 18):
        aaa.circle(50)
        aaa.right(20)
        i=i+1
    aaa.penup()
    aaa.right(180)
    aaa.forward(200)
    aaa.right(210)
    j = j + 1
```

```
aaa.color("light pink")
while (j \le 6):
    aaa.penup()
    aaa. forward (100)
    aaa.pendown()
    i = 1
    while (i <= 18):
        aaa.circle(50)
        aaa.right(20)
        i=i+1
    aaa.penup()
    aaa.right(180)
    aaa. forward (100)
    aaa.right(240)
    j = j + 1
i = 1
aaa.pendown()
while(i <= 18):
    aaa.color("gold3")
    aaa.circle(50)
    aaa.right(20)
    i=i+1
```

### **#1 Jump and Run Spiele**

Wie der Name schon sagt, steht beim MakeyMakey das Selbermachen im Vordergrund.

Der MakeyMakey ist aus technischer Sicht ein Tastaturersatz an Stelle der Standardtastatur können die einzelnen Tasten dann Bananen, Äpfel, Plastilin, Alufolie oder Personen sein. Die Platine ist ein vereinfachter Arduino, Treiber sind nicht nötig. Auf der Vorderseite finden sich die sechs typischen Spielcontrollertasten, auf der Rückseite zusätzlich Anschlüsse mit der Belegung W, A, S, D, F, den Maustasten und Bewegungstasten. Der Einstieg erfolgt meist mit vorgegebenen Beispielen, die anschließend abgewandelt werden und schließlich Ideengeber für eigene Kreationen sein können.



Jump and Run: für diese Spiele genügt es, wenn für die Pfeilschaltflächen und für die Leertaste jeweils ein Stück Alufolie verwendet wird und mit dem MakeyMakey verbunden wird.

Super Mario: <a href="https://scratch.mit.edu/projects/31583772/">https://scratch.mit.edu/projects/31583772/</a>

Flappy Bird: <a href="https://scratch.mit.edu/projects/38729886/">https://scratch.mit.edu/projects/38729886/</a>

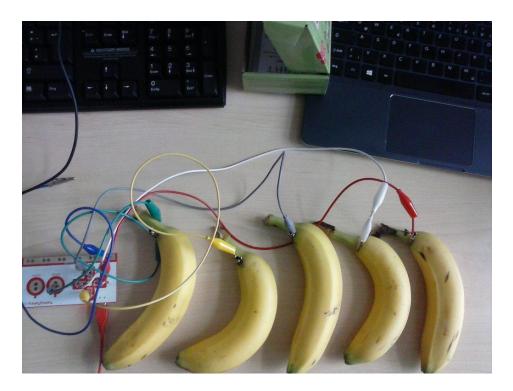
Pacman: https://scratch.mit.edu/projects/14608394/

Spannender wird es, wenn die Spiele mit mehreren Personen gespielt werden und jede/r für eine Taste zuständig ist. Die Felder können auch am Boden ausgelegt werden und man spielt mit den Füßen. Statt Alufolie geht beispielsweise auch Plastilin.



### **#2 Ein Obstklavier**

Wir kombinieren den MakeyMakey und Scratch. Für Tasten können alle stromleitenden Materialien verwendet werden, also auch Äpfel, Bananen usw. so kann man sich bspw. eine einfache Klaviatur aus Bananen bauen.



In Scratch genügt es, wenn die einzelnen Tasten mit jeweils einem Ton belegt werden. Dazu muss man die Erweiterung Musik in Scratch aktivieren.

Hier kann man dann auch mit Soundeffekten, selbst aufgenommenen Klängen und Schlagzeugeffekten experimentieren.



# #3 MakeyMakey Schlagzeug

Auch für das Schlagzeug kombinieren wir MakeyMakey und Scratch. Für die einzelnen Schlagzeugteile kann man bspw. Kreise aus Alufolie verwenden. Als Schlagzeugsticks kann man leitende Stäbe verwenden (Stricknadeln, Löffel, ...) und sie mit dem MakeyMakey verbinden.



In Scratch genügt es, wenn die einzelnen Tasten mit jeweils einem Schlagzeugeffekt belegt werden. Dazu aktiviert man zuvor die Erweiterung *Musik* in Scratch.

```
Wenn Taste Leertaste ▼ gedrückt wird

Wenn Taste Pfeil nach oben ▼ gedrückt wird

Wenn Taste Pfeil nach oben ▼ gedrückt wird

Wenn Taste Pfeil nach unten ▼ gedrückt wird

Wenn Taste Pfeil nach unten ▼ gedrückt wird

Spiele Schlaginstrument (2) Basstrommel ▼ für 0.25 Schläge

Wenn Taste Pfeil nach links ▼ gedrückt wird

Wenn Taste Pfeil nach links ▼ gedrückt wird

Spiele Schlaginstrument (3) Side-Stick ▼ für 0.25 Schläge

Wenn Taste Pfeil nach rechts ▼ gedrückt wird

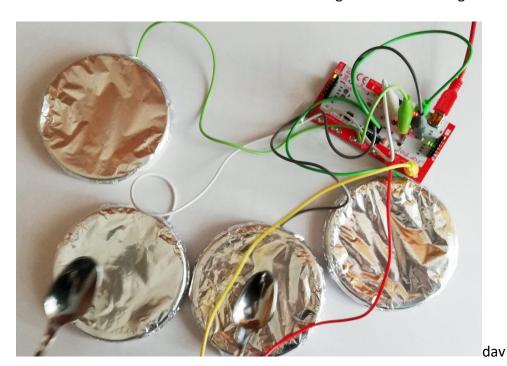
Wenn Taste Pfeil nach rechts ▼ gedrückt wird

Spiele Schlaginstrument (9) Klangstäbe ▼ für 0.25 Schläge

Wenn Taste W ▼ gedrückt wird

Spiele Klang pop ▼ ganz
```

Man kann auch mit Soundeffekten und selbst aufgenommenen Klängen das Schlagzeug belegen.



#### #4 Buzzer

Der Buzzer ist das ideale Hilfsmittel für ein Wissensquiz. Je nach Anzahl der Mitspieler kannst du dir Buzzerflächen basteln (bspw. mit Alufolie o.ä.). Für den Quizmaster gibt es eine Startschaltfläche. Alle Mitspieler/innen und der Quizmaster müssen mit der Erde verbunden sein.



Ablauf: der Quizmaster startet die Runde mit dem Berühren der Start Schaltfläche. Er stellt die Quizfrage. Wer glaubt, die Antwort zu wissen, drückt möglichst schnell seinen Buzzer und hat anschließend Zeit seine Antwort zu geben.

Das wäre ein möglicher Code für einen Buzzer:

```
Wenn angeklickt wird

zeige dich

gehe zu x: 5 y: -140

wechsle zu Kostüm white ▼

setze Punkte ▼ auf 0
```



```
Wenn Taste Leertaste ▼ gedrückt wird

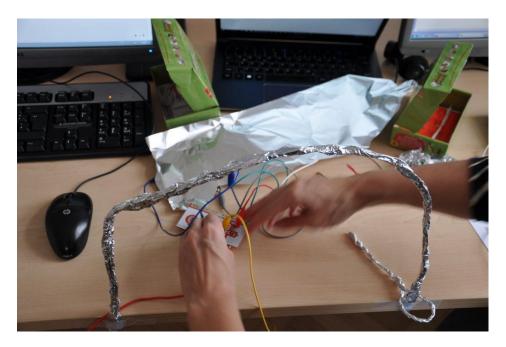
gehe zu x: 5 y: -140

wechsle zu Kostüm white ▼
```

Auch hier sind wieder viele Varianten möglich: mit Punktzähler, Zeitvorgabe, ...

### #5 Heißer Draht

Dieses Geschicklichkeitsspiel kennst du bestimmt! Dafür benötigst du einen Parcours, diesen kannst du aus Alufolie oder mit einem Draht bauen. Dann brauchst du eine Schlaufe, die durch den Parcours bewegt werden soll. Schließlich ist noch ein Start- und ein Zieltaster vonnöten. Die Schlaufe verkabelst du mit der Erde, die restlichen Teile mit jeweils einer Taste.





#### So könnte der Code in Scratch aussehen:

```
Wenn angeklickt wird

setze Fehler ▼ auf 0

wiederhole fortlaufend

falls Taste Leertaste ▼ gedrückt? , dann

spiele Klang pop ▼

ändere Fehler ▼ um 1

warte 1 Sekunden

falls Taste Pfeil nach oben ▼ gedrückt? , dann

spiele Klang gong ▼

warte 3 Sekunden

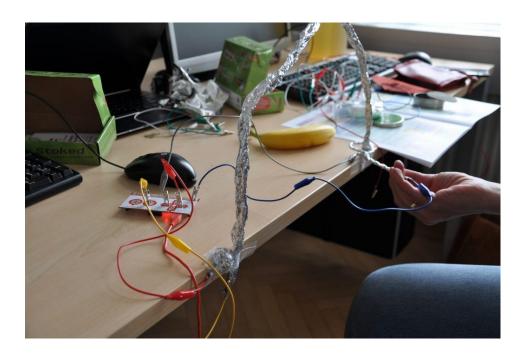
stoppe dieses Skript ▼
```

```
Wenn angeklickt wird

wiederhole bis Taste Pfeil nach oben ▼ gedrückt?

setze Zeit ▼ auf Stoppuhr

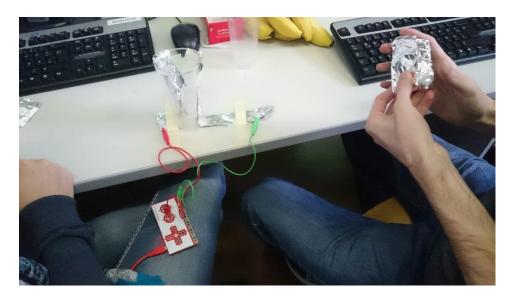
stoppe dieses Skript ▼
```



### #6 Korbball

Hierfür ist etwas Geschicklichkeit hilfreich: ein Ball soll in einen Korb geworfen werden. Bei jedem Treffer soll sich der Punktestand erhöhen.

Dafür musst du in einem Becher 2 Metallstreifen anbringen (Erdung, Taste). Der Ball muss genau so groß sein, dass er den Kontakt herstellt, wenn er im Korb landet.





#### Scratch Code:

```
wiederhole 1 mal

spiele Klang pop ▼

andere Score ▼ um 1

zeige Variable Score ▼

warte 2 Sekunden
```

## #7 Ein Bild, das Geräusche macht

Dafür benötigst du zwei Blätter, Alufolie, Schere, Kleber und Malfarben. Zeichne zuerst eine Skizze deines Bildes. Schneide dann die Teile aus, die später Geräusche machen sollen.

Anschließend legst du das zweite Blatt darunter. Markiere, wo die Löcher sind. Diese Felder beklebst du mit Alufolie. Lege auch eine Leitung mit Folie an das obere Bildende. Hier sollen später die Krokodilklemmen angeschlossen werden.



Anschließend kannst du dein Bild ausmalen.



Überprüfe, ob deine Verbindungen funktionieren und klebe danach die beiden Blätter zusammen.



Jetzt kannst du in Scratch die einzelnen Felder mit Klängen aus der Scratchbibliothek belegen.

Spannender wird es, wenn du für jedes Feld selbst eine Audioaufnahme in Scratch machst. So kannst du eine Geschichte zum Bild erzählen.

# #8 Fang den Maulwurf!

Beim Spiel MoleMash taucht ein Maulwurf an zufälligen Positionen auf einem Spielfeld auf, und der Spieler sammelt Punkte, indem er den Maulwurf trifft, bevor er wegspringt.

Du kannst damit beginnen, dass du 9 Felder bastelst, auf denen der Maulwurf auftauchen kann



Diese verbindest du dann mit dem MakeyMakey. Überlege dir dann, wie die Spielerin/der Spieler geerdet wird (ein metallener Armreifen ist eine Möglichkeit). Das Feld, auf dem der Maulwurf auftaucht soll man dann möglichst schnell berühren.

Dann brauchst du in Scratch 9 Maulwürfe mit dem jeweiligen Code:

```
werstecke dich

wiederhole fortlaufend

falls Taste Pfeil nach oben ▼ gedrückt? und MaulwurfNr = 1 , dann

setze MaulwurfNr ▼ auf 0

verstecke dich

ändere score ▼ um 1
```

```
Wenn ich 1 → empfange

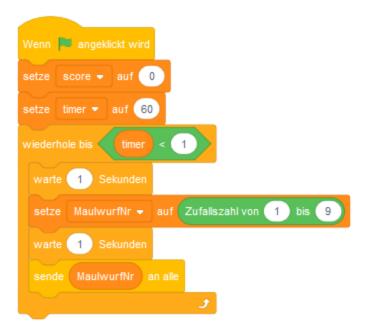
zeige dich

warte Zufallszahl von 1 bis 4 Sekunden

setze MaulwurfNr → auf 0

verstecke dich
```

Welcher Maulwurf auftaucht bestimmt der Code des Bühnenbildes. Hier ein Beispiel mit Timer:







# **#9 Laufparcours**

Du kannst am Boden einen Parcours aufkleben. Sobald der Parcours verlassen wird ertönt ein Signal. Ein Timer und ein Sekundenzähler können ergänzt werden.



Den Parcours soll jemand mit verbundenen Augen durchlaufen. Ein Mitspieler/eine Mitspielerin gibt Kommandos.