



# **Einführung in die Programmiersprache**

Zusammengestellt von Axel Chobe

30.09.2019

# 1 Inhalt

1	Inhalt .....	2
2	Einführung .....	6
3	Installation .....	7
3.1	Installation .....	7
3.2	Browserbasiert.....	7
4	Entwicklungsumgebung .....	8
4.1	Die Navigationsleiste .....	8
4.1.1	Sprachauswahl.....	8
4.1.2	Datei.....	8
4.1.3	Bearbeiten.....	8
4.1.4	Tutorien.....	8
4.2	Programmelemente.....	9
4.2.1	Reiter Sripte.....	9
4.2.2	Reiter Kostüme .....	10
4.2.3	Reiter Klänge .....	11
4.3	Ausgabebereich.....	12
4.3.1	Grüne Flagge.....	12
4.3.2	Rotes Stoppschild .....	12
4.3.3	Vollbildmodus.....	12
4.4	Objektauswahl / Objektbereich .....	12
5	Begriffsbestimmung .....	13
5.1	Koordinaten .....	13
5.2	Grad .....	13
5.3	Blockformen.....	14
5.3.1	Kopf-Blöcke .....	14
5.3.2	Stapel-Blöcke .....	14
5.3.3	Wahrheits-Blöcke.....	14
5.3.4	Wert-Blöcke.....	14
5.3.5	Klammer-Blöcke.....	14
5.3.6	Abschluss Blöcke.....	14
6	Grundlegende Elemente der Programmierung .....	15
6.1	Bewegung .....	15
6.1.1	Gehe [10] -er Schritt .....	15
6.1.2	Drehe rechts [15] Grad.....	15
6.1.3	gehe zu .....	15
6.1.4	Setze Richtung auf [90] Grad.....	16
6.1.5	setze (ändere) x,y auf [x].....	16
6.1.6	Setzte Drehtyp auf [X].....	16
6.1.7	Position anzeigen.....	16
6.2	Aussehen.....	17
6.2.1	sage / denke .....	17
6.2.2	Wechsle zu Kostüm [X].....	17
6.2.3	Wechsle zu Bühnenbild [X].....	17
6.2.4	Größe ändern .....	17
6.2.5	Effekte .....	18

6.2.6	Zeigen / Verstecken .....	18
6.2.7	Ebene wechseln .....	18
6.2.8	Info über aktuelle Einstellung .....	18
6.3	Klang .....	19
6.3.1	Spiele Klang .....	19
6.3.2	Effekte .....	19
6.3.3	Lautstärke .....	19
6.3.4	Lautstärke anzeigen .....	19
6.4	Ereignisse .....	20
6.4.1	Startmöglichkeiten durch Benutzer .....	20
6.4.2	Startmöglichkeit durch Ereignisse .....	20
6.4.3	Startmöglichkeit durch Nachrichtenaustausch .....	20
6.5	Steuerung .....	21
6.5.1	Pause .....	21
6.5.2	Wiederholungen .....	21
6.5.3	Bedingte Ausführungen Kopfschleife .....	21
6.5.4	Bedingte Ausführung Fußschleife .....	22
6.5.5	Stoppe .....	22
6.5.6	Klonen .....	22
6.6	Fühlen .....	23
6.6.1	Wird [ ] berührt/gedrückt? .....	23
6.6.2	Entfernung .....	23
6.6.3	Frage [ ] und warte .....	23
6.6.4	Maus-Position .....	24
6.6.5	Lautstärke .....	24
6.6.6	Stoppuhr .....	24
6.6.7	Benutzername .....	24
6.7	Operationen .....	25
6.7.1	Operation mit Zahlenergebnis .....	25
6.7.2	Operation mit Wahrheitsergebnis .....	25
6.7.3	Logische Operatoren .....	25
6.7.4	Zufallszahlen .....	26
6.7.5	Arbeit mit Texten .....	26
6.7.6	Funktionen .....	26
6.8	Variablen .....	27
6.8.1	Variable anlegen .....	27
6.8.2	Variable benutzen .....	27
6.9	Listen .....	27
6.9.1	Liste erstellen .....	28
6.9.2	Liste füllen .....	28
6.9.3	Liste bearbeiten .....	28
6.9.4	Mit Listen arbeiten .....	28
6.9.5	Liste speichern .....	28
6.10	Meine Blöcke (Unterprogramme) .....	29
6.10.1	Block erstellen .....	29
7	Erweiterte Elemente .....	30
7.1	Musik .....	31
7.1.1	Schlaginstrument .....	31

7.1.2	Setze Instrument auf.....	31
7.1.3	Spiele Ton.....	31
7.1.4	Pausiere.....	31
7.1.5	Tempo.....	31
7.2	Malstift.....	32
7.2.1	Lösche alles.....	32
7.2.2	Hinterlasse Abdruck.....	32
7.2.3	Zeichnen.....	32
7.2.4	Optionen.....	32
7.3	Video.....	33
7.3.1	Einschalten / Transparenz.....	33
7.3.2	Auswertung der Bewegung.....	33
7.3.3	Einfache Bewegungsauswertung.....	33
7.4	Sprachausgabe.....	34
7.4.1	Text zu Sprache.....	34
7.4.2	Übersetzen.....	34
7.5	Makey Makey.....	35
7.5.1	Tastenabfrage.....	35
7.6	Micro:bit.....	36
7.6.1	Einrichtung.....	36
7.6.2	Abfrage der Knöpfe A und B.....	37
7.6.3	Bewegungsabfrage.....	37
7.6.4	LED ansteuern.....	37
7.6.5	Anzeigen.....	37
7.6.6	Lagesensor auswerten.....	37
7.6.7	Neigungswinkel anzeigen.....	38
7.6.8	Abfrage weiterer Pins.....	38
7.6.9	Hardwarerweiterungen.....	38
7.7	Lego Boost.....	39
7.7.1	Einrichtung.....	39
7.7.2	Motorsteuerung.....	39
7.7.3	Motorsteuerung variieren.....	40
7.7.4	Motorposition auslesen.....	40
7.7.5	Farberkennung.....	41
7.7.6	Neigungsauswertung.....	41
7.7.7	Farbausgabe.....	41
7.7.8	Abstandsmessung.....	41
8	Debugging.....	42
8.1	Skript am falschen Ort.....	42
8.2	Vergessene Blocks.....	42
8.3	Verwechselte Blocks.....	42
8.4	X und y verwechselt.....	42
8.5	Falsche Zahlen.....	42
8.6	Falsche Auswahl.....	42
8.7	Falsche Reihenfolge.....	43
8.8	Schleifen.....	43
8.9	Zeitliche Begrenzung.....	43
8.10	Falsches Timing.....	43

9	Veröffentlichen .....	44
9.1	Voraussetzung .....	44
9.2	Projekt hochladen .....	44
9.3	Eigene Projekte anschauen .....	44
10	Beispiele .....	45
10.1	Kurzbeispiele .....	45
10.2	Spiel Pong.....	46
10.3	Katzenrennen .....	47
10.4	Zeichnen von bunten Linien .....	48
10.5	Objektsteuerung mit micro:bit.....	48
10.6	Musik mit Gestik Steuerung.....	49
10.7	Linien Zeichnen .....	49
10.8	Zahlen raten .....	50
10.9	Kultspiele im Netz.....	50

## 2 Einführung

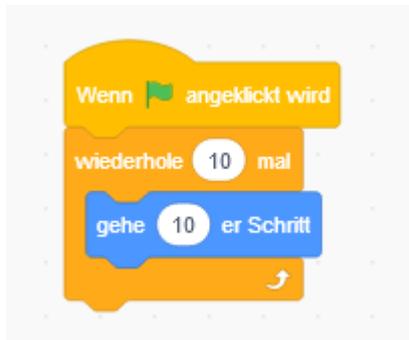
Ein Computer ist eine Maschine die nicht selber denken kann. Man muss ihm alles was er machen soll befehlen. Diese Befehle gehören zu einer Programmiersprache, d.h. eine Sprache die der Computer versteht. Eine Abfolge von Befehlen mit einem Anfang und einem Ende nennt man Programm. Ein Computer liest dieses Programm und führt jeden Befehl meist nacheinander aus. Das Prinzip ist ähnlich dem eines Kochrezepts. Beim Kochen müssen die einzelnen Schritte (Befehle) nacheinander in der richtigen Reihenfolge ausgeführt werden um ein wohlschmeckendes Ergebnis zu erzielen. Genauso muss auch der Computer die Befehle in einer sinnvollen Reihenfolge ausführen um ein nutzbares Ergebnis zu erzeugen.

In einer gängigen Programmiersprache wie bspw. „C“ würde ein kleines Programm zum Addieren von Zahlen wie folgt aussehen:

```
# include <stdio.h>

int main(int argc, char *argv[])
{
float a, b;
printf(„Bitte geben Sie den ersten Summanden ein: “);
scanf(a);
printf(„Bitte geben Sie den zweiten Summanden ein: “);
scanf(b);

printf(„%f + %f ist %f“, a, b, (a+b));
return 0;
}
```



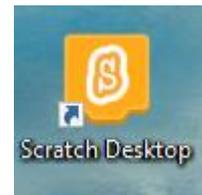
Scratch dagegen ist eine grafische Programmiersprache. Das bedeutet man benutzt keine schriftlichen Anweisungen wie z.B. in einem Kochrezept, sondern Symbole ähnlich als wenn man Bilder von den einzelnen Schritten des Kochrezepts hätte. Diese Symbole müssen in der auf das Problem zugeschnittenen richtigen Reihenfolge angeordnet werden. Dabei sind alle Symbole wie ein Puzzle aufgebaut, so dass ein Element mit spitzen Ecken nicht in ein Feld mit runder Form passen kann. Ein Problem, das im Folgenden gelöst wird, wäre z.B.: „Ich möchte gerne, dass auf meinem Bildschirm eine Katze von links nach rechts läuft!“

### 3 Installation

Die Programmierumgebung lässt sich in 2 Variationen benutzen.

#### 3.1 Installation

Die zurzeit aktuelle Version 3.4.0 steht z.B. auf der Seite von ComputerBild zum Download bereit. Anschließend wird die Datei ausgeführt und damit das Programm installiert.

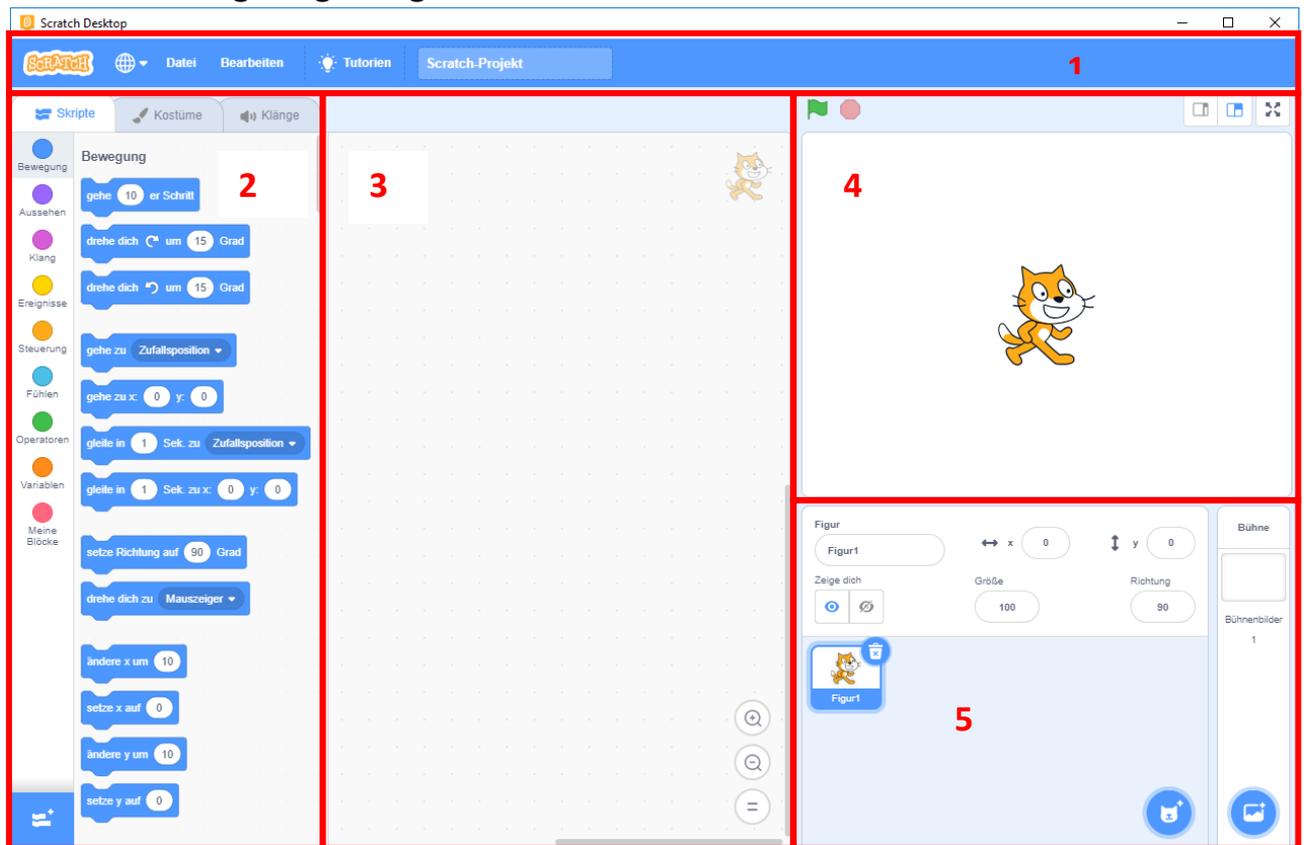


Durch Doppelklick auf das Icon startet dann die Entwicklungsumgebung.

#### 3.2 Browserbasiert

Zuerst wird ein Browser nach Wahl geöffnet. In die Adresszeile wird nun die Adresse „https://scratch.mit.edu/projects/editor“ eingegeben und es erscheint die Entwickleroberfläche.

## 4 Entwicklungsumgebung



Die Entwicklungsumgebung besteht aus 5 Teilen. Die obere Navigationsleiste (1), die linke Spalte mit den Programmelementen für die Programmierung (2), den mittleren Teil in dem die eigentliche Programmierung stattfindet (3) und den rechten Teil, in dem sich der Ausgabebildschirm (4) und die Objektauswahl (5) befindet.

### 4.1 Die Navigationsleiste



#### 4.1.1 Sprachauswahl

Über den Pfeil hinter der Weltkugel, lässt sich die Sprache einstellen

#### 4.1.2 Datei

Hier besteht die Möglichkeit eine neue Datei anzulegen, eine vorhandene Datei vom eigenen Computer zu öffnen sowie ein Programm auf dem eigenen Computer zu speichern.

#### 4.1.3 Bearbeiten

Der Turbo-Modus dient zur Beschleunigung von Scratch-Projekten. Hierbei werden alle Blöcke schneller ausgeführt, so dass Skripte insgesamt schneller laufen. Der Turbo-Modus hat jedoch keinen Einfluss auf den "warte () Sek."-Block.

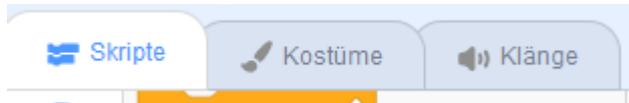
Der Punkt Wiederherstellen ist z.Z. noch nicht dokumentiert.

#### 4.1.4 Tutorien

Hier können kurze Tutorien zu verschiedenen Themen aufgerufen werden.

## 4.2 Programmelemente

### 4.2.1 Reiter Srippte



#### 4.2.1.1 Blockpalette

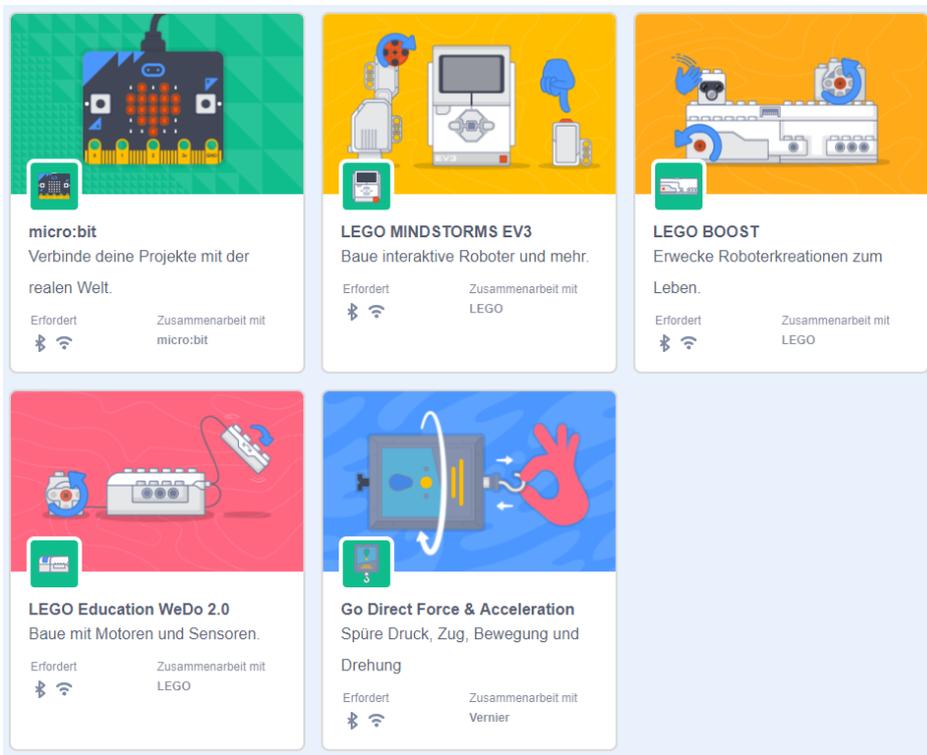
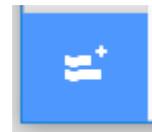
Die im linken Bereich angeordneten farbigen Kreise stellen die Blockpalette dar. Durch Anklicken einer Palette öffnen sich daneben, der dazugehörige Blockbereich.

#### 4.2.1.2 Blockbereich

Hier werden die Blöcke der angewählten Kategorie angezeigt. Diese werden dann per Drag & Drop in den Programmierbereich gezogen, um das Programm zu erstellen. (Siehe Kapitel xx)

#### 4.2.1.3 Erweiterungen hinzufügen

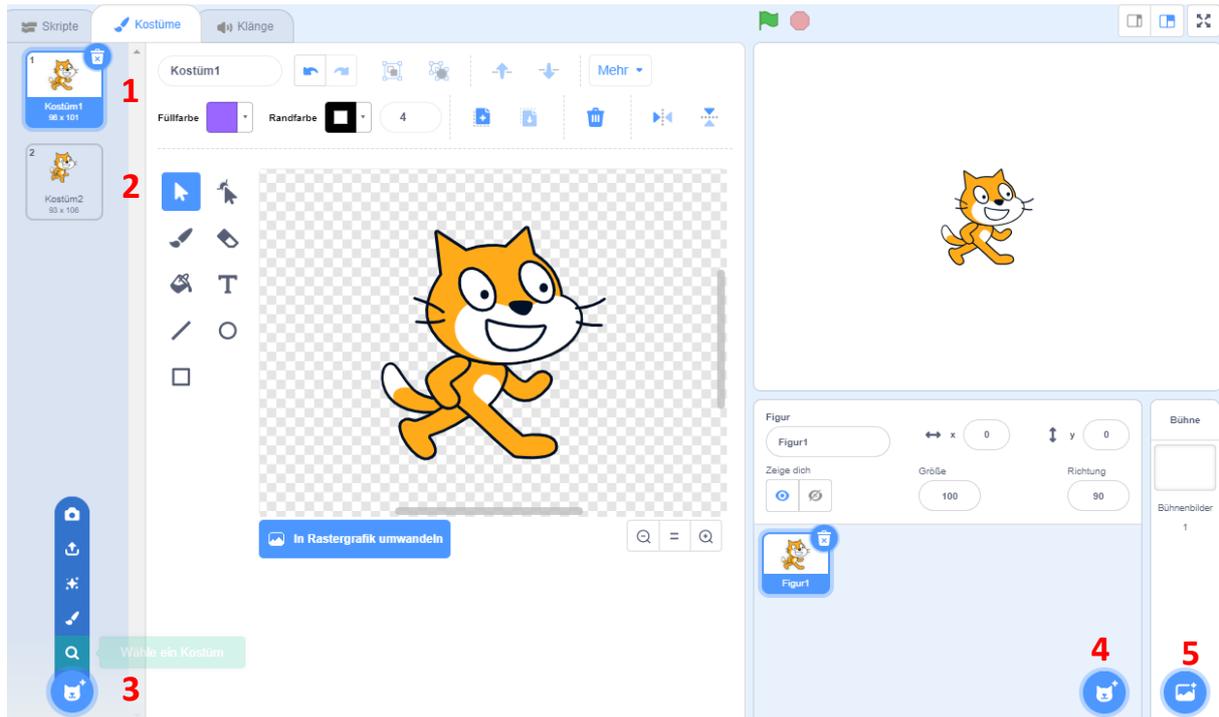
Mit diesem Button können Erweiterungen hinzugefügt werden. (siehe Kapitel9)



- \*Musik
- \*Malstift
- \*Video-Erfassung
- \*Text zu Sprache
- \*Übersetzen
- \*Makey Makey
- \*micro:bit
- \*LEGO MIND
- \*STORMS EV3
- \*LEGO BOOST
- \*LEGO WeDO 2.0
- \*GO Direct
- \*Fischertechnik

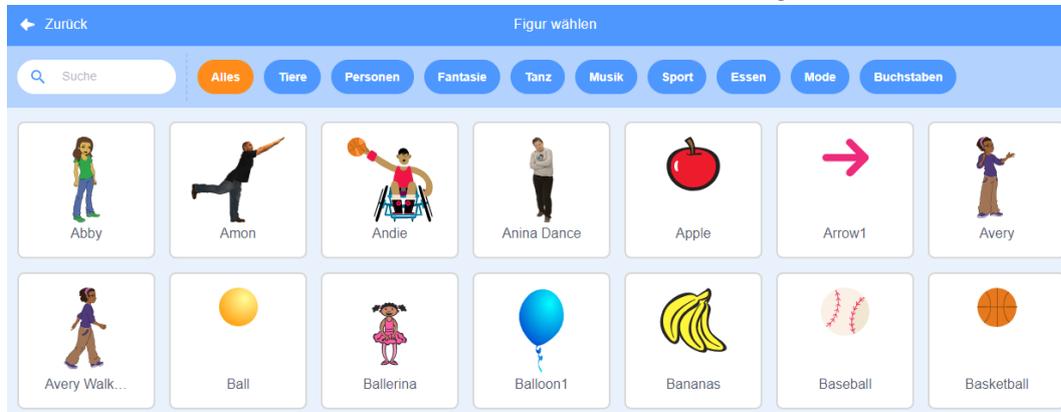
Mit Erweiterungen kann man seit Scratch 3.0 bestimmte zusätzliche Funktionen erhalten. Man kann Erweiterungen nicht selber erstellen, diese werden von Scratch bereitgestellt.

### 4.2.2 Reiter Kostüme

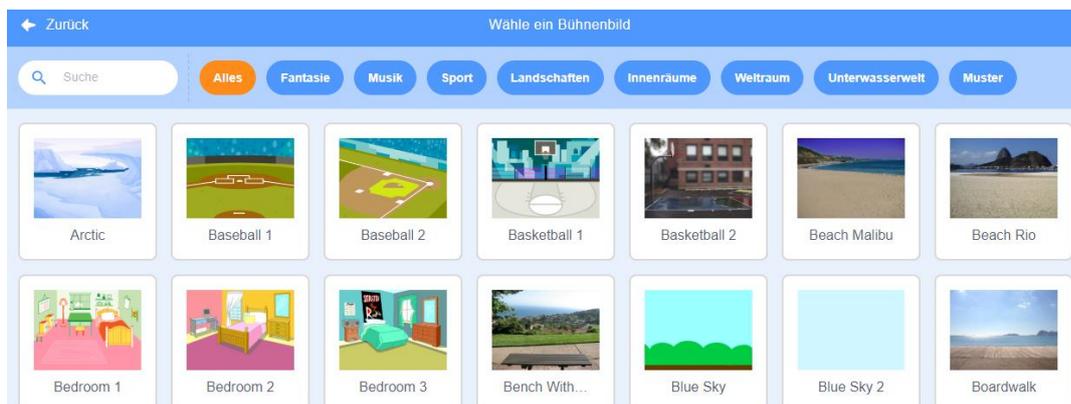


Jedes Objekt (hier die Katze) besitzt mindestens 1 Kostüm, d.h. ein Aussehen mit dem es später im Programm zu sehen ist. Ein Objekt kann beliebig viele Kostüme besitzen, um das Projekt dynamisch zu gestalten. Die Katze besitzt z.B. 2 Kostüme, um im Programm eine Laufbewegung darzustellen. Bild 1 – rechte Bein vorn, Bild 2 – linke Bein vorn.

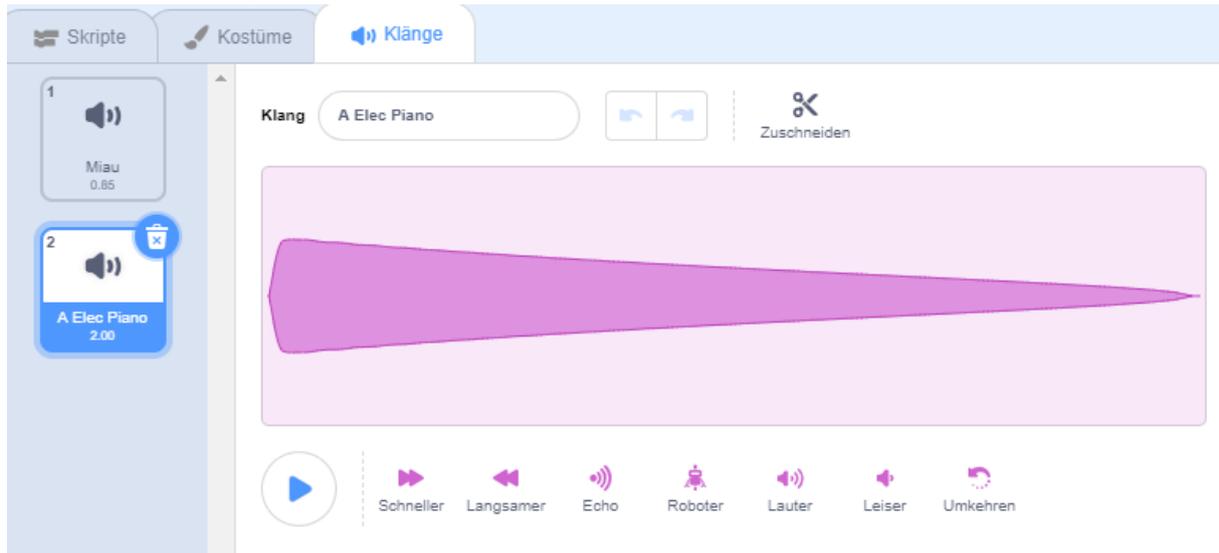
Mit dem Button 3 oder 4 können alle vorhandenen Kostüme aufgerufen werden.



Mit dem Button 5 können die Bühnenbilder (Hintergrundbild) ausgewählt werden.



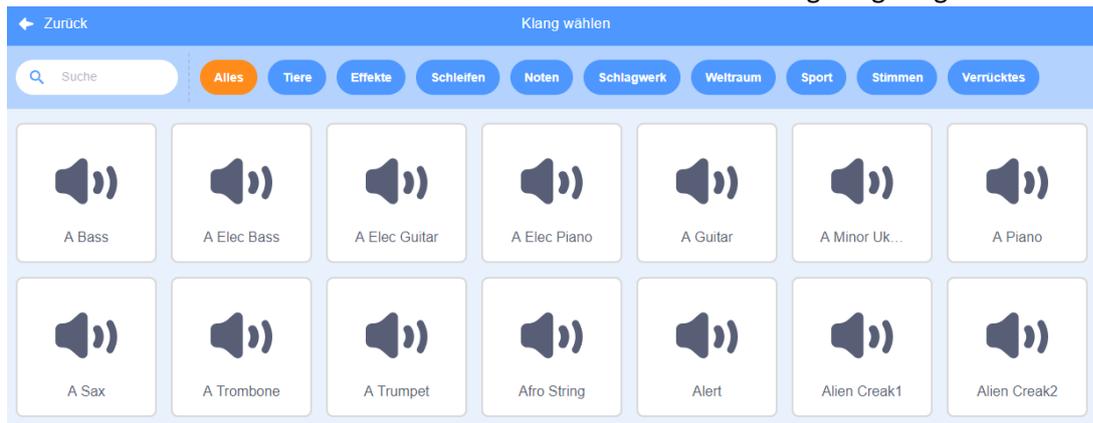
### 4.2.3 Reiter Klänge



Um in einem Programm verschiedene Klänge zu nutzen, muss man diese, genau wie bei Kostümen, erst einmal den Objekten zuweisen.



Mit dem Button unten links werden alle vorhandenen Klänge angezeigt.



Diese Klänge können entsprechend verändert werden. (Schneller, Langsamer, Echo ...)

### 4.3 Ausgabebereich



Auf dem Ausgabebereich erscheinen alle Objekte und Hintergründe. Er ist 480 Einheiten breit und 360 Einheiten hoch. Der Nullpunkt ( $x=0, y=0$ ) liegt genau in der Mitte des Bereiches.

#### 4.3.1 Grüne Flagge

Ein Klick auf die Grüne Flagge startet dein Programm. Doch Achtung – nur die Skripte (Befehlsfolgen), die am Anfang den Block »Wenn grüne Flagge angeklickt« (dazu später mehr) besitzen, werden durchgeführt.

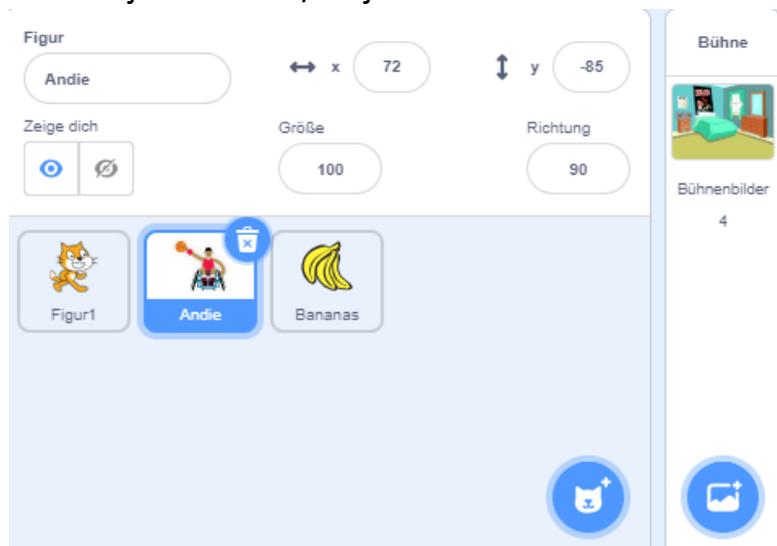
#### 4.3.2 Rotes Stoppschild

Das Stoppschild beendet das Programm. Keines der Skripte wird dann weiter durchgeführt.

#### 4.3.3 Vollbildmodus

Um ein Projekt auf einer größeren Bühne laufen zu lassen, drückt man einfach auf den Vollbildmodus und schon vergrößert sich die Bühne. Ein erneuter Klick auf das nun etwas veränderte Symbol, lässt die Bühne wieder zur Ausgangsgröße zurück schrumpfen.

### 4.4 Objektauswahl / Objektbereich

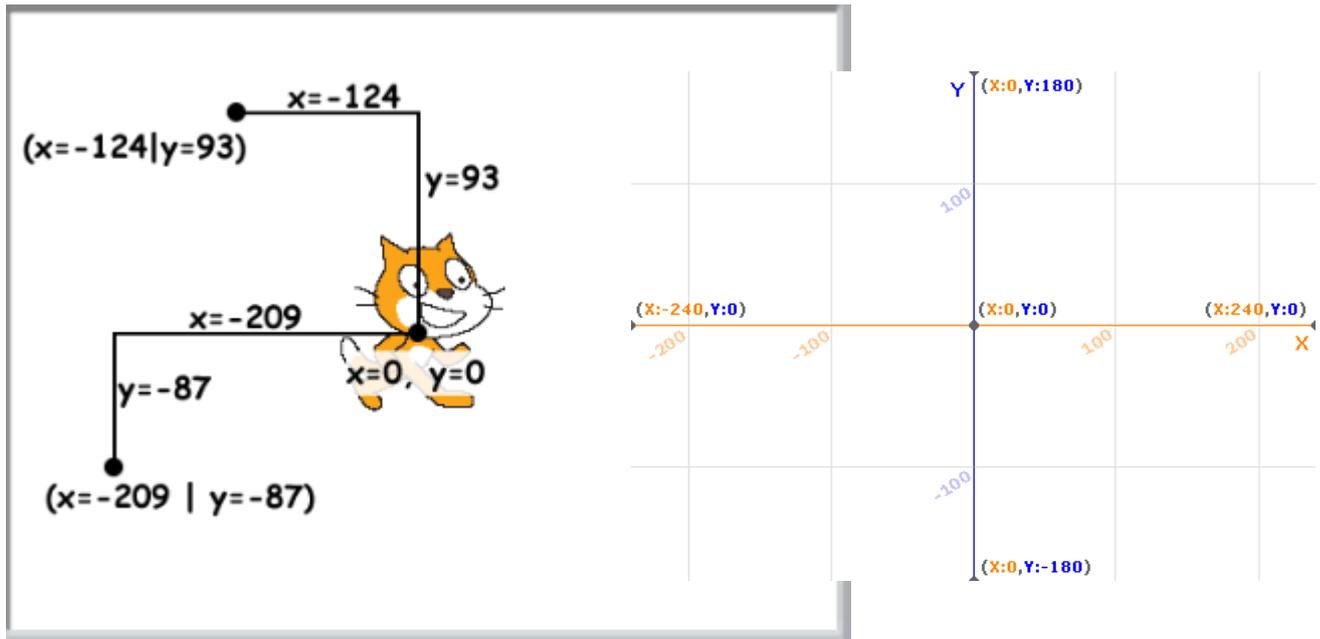


Dieser Bereich enthält alle Symbole zur Bearbeitung und Erstellung von Objekten. Mit den beiden Buttons lassen sich weitere Figuren bzw. Bühnenbilder wählen. Diese sind dann in der Größe und Position veränderbar.

## 5 Begriffsbestimmung

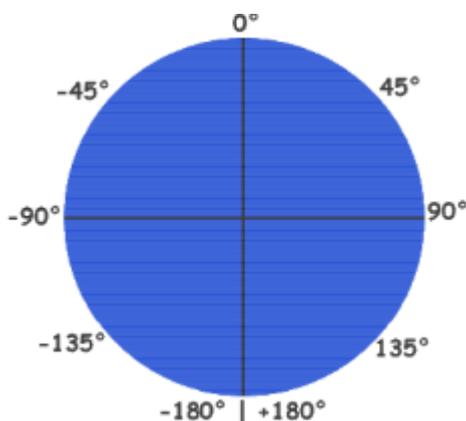
### 5.1 Koordinaten

Der Scratchbildschirm ist in 480 Spalten und 360 Zeilen aufgeteilt. So entstehen  $480 \cdot 360 = 172800$  einzelne kleine Quadrate. Jedes dieser Quadrate kann durch die Spalten- und die Zeilennummer angesprochen werden, z.B.  $(x=45, y=189)$  Der Nullpunkt dieses so genannten Scratchkoordinatensystems liegt genau in der Mitte. Demnach würden alle Punkte unterhalb des Mittelpunkts einen negativen y-Wert erhalten und alle Punkte über dem Mittelpunkt positive y-Werte. Genauso besitzen alle Punkte links vom Mittelpunkt einen negativen x-Wert und alle Punkte rechts vom Mittelpunkt einen positiven x-Wert.



### 5.2 Grad

Ein Kreis ist in 360 Teile untergliedert, d.h. er besitzt 360 Kuchenstücke die alle eine Nummer bekommen. Stellt man sich nun gedanklich in die Mitte des Kuchens und schaut in Richtung Kuchenstück 1 hat man die Richtung 1 Grad ( $1^\circ$ ). Dreht man sich nun um 15 Kuchenstücke ( $15^\circ$ ) im Uhrzeigersinn, schaut man auf das 16. Kuchenstück des Kreises und hat nun die Richtung  $16^\circ$ . In Scratch allerdings ist der Kuchen in 180 positive und in 180 negative Kuchenstücke unterteilt. Schaut man in die Richtung  $+178^\circ$  und dreht sich um  $11^\circ$  im Uhrzeigersinn schaut man in die Richtung  $-171^\circ$ .



## 5.3 Blockformen

### 5.3.1 Kopf-Blöcke



Kopf-Blöcke sind die Blöcke die jedes Skript starten. Sie sind oben abgerundet und haben oben und unten eine Einbuchtung um andere Blöcke einzuklinken. Es gibt sieben Kopf-Blöcke in den Kategorien Ereignisse und Steuerung.

### 5.3.2 Stapel-Blöcke



Stapel-Blöcke sind die Blöcke welche die wesentlichen Befehle ausführen. Sie haben eine Ausbuchtung oben und eine Einbuchtung unten, damit andere Blöcke oberhalb und unterhalb eingeklinkt werden können. Es gibt mehr als 70 Stapel-Blöcke in fast allen Farben, was sie zur am weitesten verbreiteten Blockform macht.

### 5.3.3 Wahrheits-Blöcke



Wahrheits-Blöcke antworten entweder mit Wahr oder Falsch. Es ist als ob Du einen Freund fragst: »Ist  $2 + 2 = 4$ ?«, und er antwortet entweder »Ja« oder »Nein«. Es gibt zurzeit elf Wahrheits-Blöcke in den Kategorien Fühlen, Operatoren und Daten.

### 5.3.4 Wert-Blöcke



Wert-Blöcke stehen für Werte. Diese Werte sind entweder Zahlen oder Zeichenketten. Es ist als ob Du einen Freund fragst, »Wie viel ist  $2 + 2$ ?«, und er würde antworten »4«. Es können auch Variablen sein, zum Beispiel, »Wie alt bist Du?« mit der Antwort: »15«.

### 5.3.5 Klammer-Blöcke



Klammer-Blöcke sind Blöcke, die manchmal aussehen wie ein C, deshalb auch der Name im Englischen. Diese Block Art kommt dann zum Einsatz, wenn etwas X mal passieren soll, oder wenn eine Bedingung gestellt ist, die erfüllt werden muss. Es gibt fünf dieser Klammer-Blöcke, die man in der Kategorie Steuerung findet.

### 5.3.6 Abschluss Blöcke



Abschluss Blöcke sind die Blöcke, welche die Skripte beenden. Diese Gruppe zählt nur zwei Blöcke, die man in der Steuerungskategorie findet.

## 6 Grundlegende Elemente der Programmierung

### 6.1 Bewegung

Um ein Objekt auf dem Bildschirm zu bewegen, gibt es den Menüpunkt „Bewegen“. Mittels der Programmelemente ist es möglich, ein Objekt auf verschiedene Weise über den Bildschirm zu bewegen. Diese Programmelemente sind blau.

#### 6.1.1 Gehe [10] -er Schritt



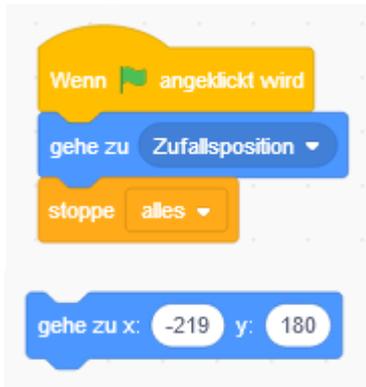
Die Richtung in die das Objekt sich bewegt entspricht hierbei der Blickrichtung des Objektes. Die Entfernung die dabei zurückgelegt werden soll, kann man selber in ein kleines Feld eintragen. Standardmäßig steht hier eine 10 drin. Dieses kleine Programm bewirkt, das sich beim Klicken auf das grüne Fähnchen das Objekt um 10 Koordinatenpunkte in Blickrichtung bewegt.

#### 6.1.2 Drehe rechts [15] Grad



Die Blickrichtung des Objektes ändert sich je nach der eingegeben Gradzahl. Standardmäßig ist eine 15 eingetragen. Es existiert ebenfalls ein Programmelement „drehe links [15] Grad“. Die beiden Programmelemente unterscheiden sich je nach Ausrichtung der Pfeile. Dieses kleine Beispielprogramm würde also das Objekt beim Klicken auf das grüne Fähnchen um 15 Grad nach rechts drehen.

#### 6.1.3 gehe zu



Um ein Objekt nicht nur in Blickrichtung entlang zu bewegen, gibt es dieses Programmelement. Dabei kann noch zwischen 2 Zielen unterschieden werden. Entweder ein Zufalls-  
punkt oder es wird dem Mauszeiger gefolgt.



Ein weiteres Programmelement bietet die Möglichkeit gezielt einen Punkt anzuspringen.



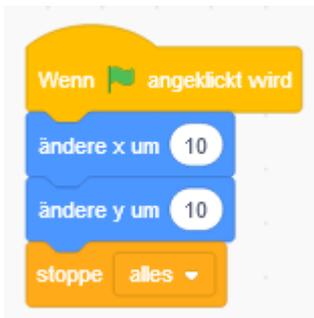
Beide Bewegungsmöglichkeiten lassen sich noch um den Wert Zeit erweitern. So wird die Zeit in Sekunden angegeben um den Zielwert zu erreichen.

### 6.1.4 Setze Richtung auf [90] Grad



Um die Richtung des Objektes nicht nur um einzelne Grad zu ändern, gibt es das Programmelement „zeige Richtung [90]“. Hierbei kann man die Blickrichtung direkt auf eine von vier verschiedenen Werten setzen. Es gibt die Möglichkeit die Richtung auf 90 (rechts), -90 (links), 0 (oben) und 180 (unten) zu setzen. In diesem kleinen Programm würde sich also die Blickrichtung beim Klicken auf das Fähnchen immer nach rechts richten.

### 6.1.5 setze (ändere) x,y auf [x]



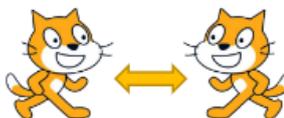
Um eine dynamische Bewegung zu erzeugen gibt es das Programmelement „ändere x um“ beziehungsweise „ändere y um“. Hierbei wird die x-beziehungsweise die y- Koordinate des Objektes jeweils um den eingetragenen Wert geändert. Somit würde in unserem Beispiel das Objekt beim Klicken auf das grüne Fähnchen, immer seine Position um 10 Punkte in x- und 10 Punkte in y-Richtung ändern. Das Objekt würde also langsam in die obere rechte Ecke wandern. Um Objektposition statisch anzugeben kann man die Programmelemente „setze x auf“ beziehungsweise „setze y auf“ verwenden.

### 6.1.6 Setzte Drehtyp auf [X]

Objekt der es einem erlaubt, den Drehtyp einer Figur während der Ausführung eines Projekts zu verändern, anstatt dies auf dem üblichen Weg manuell in den Figureigenschaften zu tun.

Rundherum bedeutet, dass die Figur sich um ganze 360° drehen kann.

Dies ist die Standardeinstellung. „links-rechts“ bedeutet, die Figur rotiert um seine eigene Y-Achse.



### 6.1.7 Position anzeigen



Durch Anklicken des Kästchens an dem entsprechenden Punkt, wird dieser auf der Bühne rechts oben angezeigt. Dies kann entweder für den Programmablauf wichtig sein, oder zur Fehlersuche benutzt werden.

Figur1: x-Position -61

Figur1: y-Position 70



## 6.2 Aussehen

Diese Blöcke kontrollieren das Aussehen von Figuren, ohne das Malprogramm von Scratch zu verwenden. Hier kann man zum Beispiel verschiedene Grafikeffekte anwenden, Figuren zeigen und verstecken oder Kostüme wechseln.

### 6.2.1 sage / denke



Die Comicartigen Sprechblasen können als sage oder denke-Element eingesetzt werden. Anders als beim Sage () für () Sek.-Block wird die Sprechblase so lange angezeigt, bis das Stoppschild gedrückt wird.

### 6.2.2 Wechsle zu Kostüm [X]



Mittels diesem Programmelement ist es möglich, dem Objekt ein anderes Aussehen zu geben. Die Kostüme müssen allerdings vorher im Objekt geladen werden. Alternativ kann auch das Element „wechsle zum nächsten Kostüm“ verwendet werden.

### 6.2.3 Wechsle zu Bühnenbild [X]



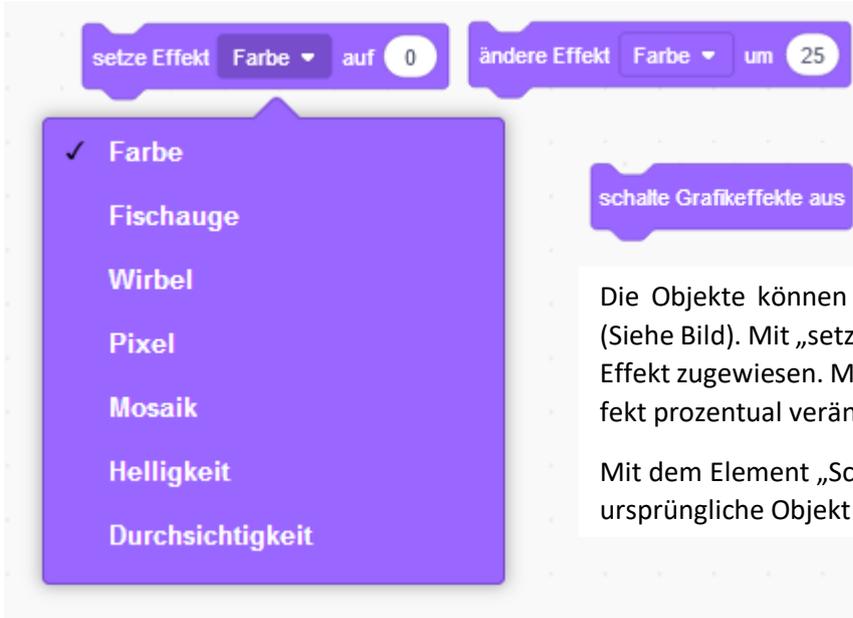
Mit diesem Programmelementen ist es möglich, das Bühnenbild zu wechseln. Auch hierbei müssen die Bühnenbilder zuvor geladen werden.

### 6.2.4 Größe ändern



Um die Größe eines Objektes zu ändern gibt es 2 Elemente. Bei „ändere Größe um“ wird das Objekt um den angegebenen Wert prozentual vergrößert. Beim Element „setze Größe“ wird das Objekt die den angegebenen Wert verändert.

### 6.2.5 Effekte



Die Objekte können verschiedene Effekte erhalten. (Siehe Bild). Mit „setze Effekt ...“ wird dem Objekt ein Effekt zugewiesen. Mit „ändere Effekt ...“ wird der Effekt prozentual verändert.

Mit dem Element „Schalte Grafikeffekt aus“ wird das ursprüngliche Objekt wiedergegeben.

### 6.2.6 Zeigen / Verstecken



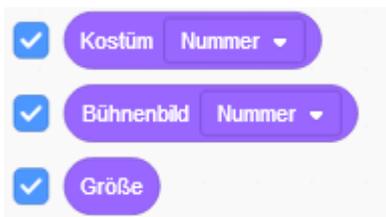
Unter bestimmten Bedingungen kann es notwendig sein ein Objekt zu verstecken. Im Beispiel erscheint das versteckte Objekt nach 1 Sekunde wieder auf der Bühne.

### 6.2.7 Ebene wechseln

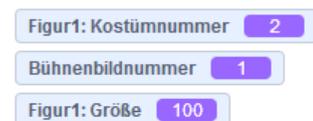


Sind mehrere Objekte auf der Bühne geht es hierbei darum, welches Objekt im Vordergrund bzw. im Hintergrund sichtbar ist.

### 6.2.8 Info über aktuelle Einstellung



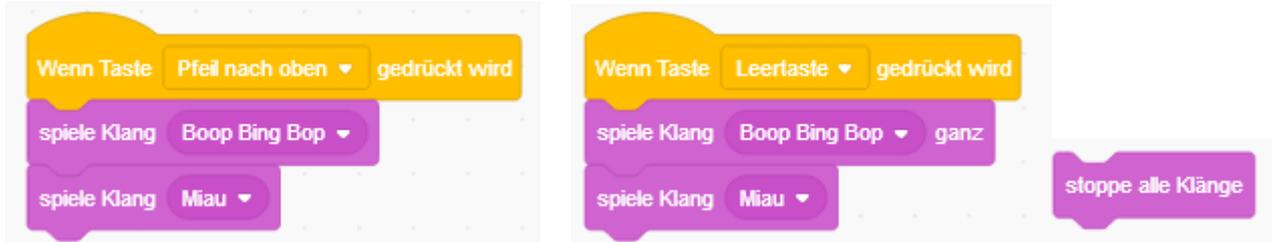
Durch Anklicken des Kästchens an dem entsprechenden Punkt, wird dieser auf der Bühne rechts oben angezeigt. Dies kann entweder für den Programmablauf wichtig sein, oder zur Fehlersuche benutzt werden.



## 6.3 Klang

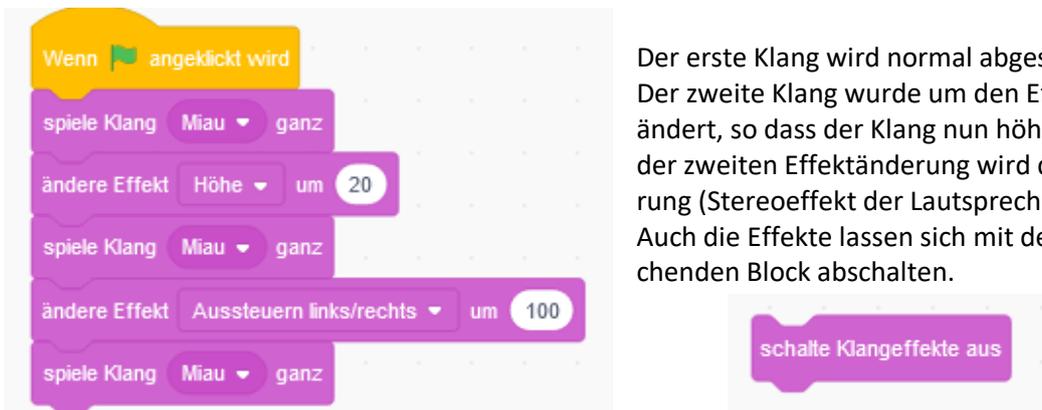
Jedem Objekt, ist ein typischer Klang zugeordnet. Weitere Klänge, die man für ein Objekt benutzen möchte, müssen allerdings erst geladen werden. Dazu wird der Reiter Klänge geöffnet. Siehe Punkt 4.2.3. Unter den Erweiterungen (Siehe Punkt X.X.X) können auch Instrumente eingesetzt werden.

### 6.3.1 Spiele Klang



Der „spiele Klang ()“ Block beginnt, den ausgewählten Klang abzuspielen und geht dann, Millisekunden später, zum nächsten Block weiter. Der „spiele Klang () ganz“ Block spielt den ausgewählten Klang von Anfang bis Ende ab und geht erst dann zum nächsten Block weiter. Der „stoppe alle Klänge“ Block stoppt global alle Klänge, die sich in Wiedergabe befinden.

### 6.3.2 Effekte



Der erste Klang wird normal abgespielt. Der zweite Klang wurde um den Effekt Höhe verändert, so dass der Klang nun höher klingt. In der zweiten Effektänderung wird die Aussteuerung (Stereoeffekt der Lautsprecher) verändert. Auch die Effekte lassen sich mit dem entsprechenden Block abschalten.

### 6.3.3 Lautstärke



Die Lautstärke kann um einen Wert oder um einen Prozentwert verändert werden.

### 6.3.4 Lautstärke anzeigen



Wenn der Haken gesetzt wird, kann die Lautstärke angezeigt werden.



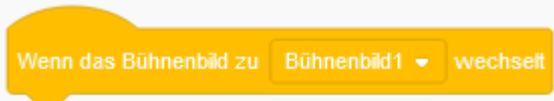
## 6.4 Ereignisse

Ereignisse-Blöcke werden verwendet, um Skripte zu starten. Sie kontrollieren auch das Senden von Signalen einer Figur and andere Figuren oder die Bühne mit den Sende-Blöcken.

### 6.4.1 Startmöglichkeiten durch Benutzer

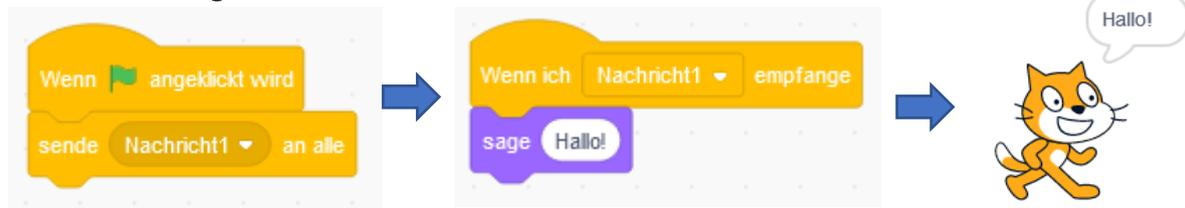
	Startet das Programm, wenn die Fahne angeklickt wird.
	Startet das Programm, wenn die ausgewählte Taste gedrückt wird.
	Startet das Programm, wenn eine Figur angeklickt wird.

### 6.4.2 Startmöglichkeit durch Ereignisse

	Startet den Programmteil, wenn im Programmablauf die Bühne gewechselt wird.
	Startet den Programmteil, wenn im Programm, die Lautstärke größer als 10 ist.

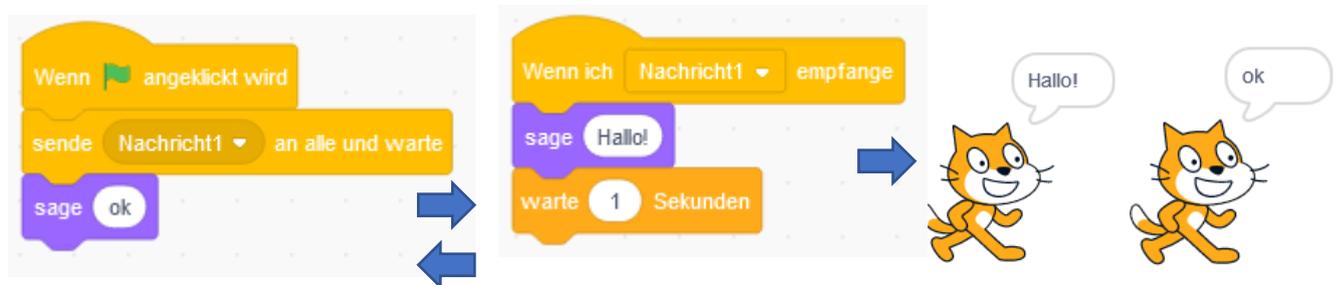
Zu Testzwecken starten die Programme auch durch das anklicken mit der Maus auf den Startblock!

### 6.4.3 Startmöglichkeit durch Nachrichtenaustausch



The diagram illustrates a message-passing mechanism. On the left, a Scratch script starts with a 'Wenn Flagge angeklickt wird' block, followed by a 'sende Nachricht1 an alle' block. A blue arrow points to a sub-program script that begins with a 'Wenn ich Nachricht1 empfangen' block, followed by a 'sage Hallo!' block. A final blue arrow points to a Scratch cat character with a speech bubble saying 'Hallo!'.

Wenn im Hauptprogramm der Block „sende [Nachricht1] an alle“ aufgerufen wird, startet das so genannte Unterprogramm und führt die weiteren Kommandos aus.



This diagram shows a sub-program that includes a 'warte 1 Sekunden' block. The main program's 'sende [Nachricht1] an alle' block is replaced by 'sende [Nachricht1] an alle und warte'. A blue arrow points to the sub-program script, which contains 'Wenn ich Nachricht1 empfangen', 'sage Hallo!', and 'warte 1 Sekunden'. A return arrow points back to the main program, which now has a 'sage ok' block. The Scratch cat character is shown saying 'Hallo!' and 'ok'.

Wenn im Hauptprogramm der Block „sende [Nachricht1] an alle und warte“ aufgerufen wird, startet das Unterprogramm und danach springt das System wieder zum Hauptprogramm und arbeitet dort weiter.

## 6.5 Steuerung

Steuerungen werden überall da eingesetzt, wo etwas beliebig oft, oder nur passieren soll, wenn eine bestimmte Bedingung erfüllt wird.

### 6.5.1 Pause



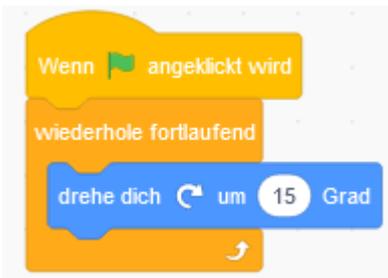
Das Programm stoppt für die angegebene Zeit

### 6.5.2 Wiederholungen



*wiederhole () mal*

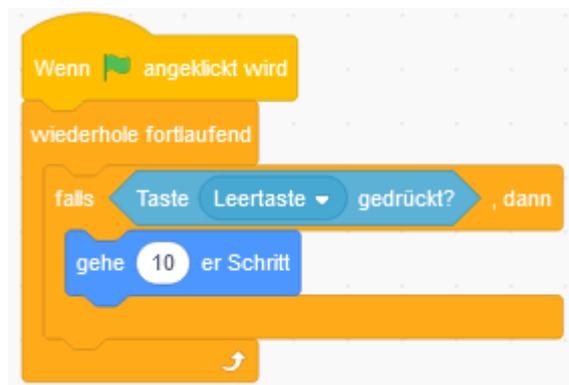
Er wiederholt die enthaltenen Befehle der eingetragenen Zahl entsprechend oft.



*wiederhole fortlaufend*

Dieser Block wiederholt die in ihm enthaltenen Befehle unendlich oft. Diese Schleife kann nur durch das Klicken des Stoppschildes oder durch einen "stoppe ()"-Block angehalten werden.

### 6.5.3 Bedingte Ausführungen Kopfschleife



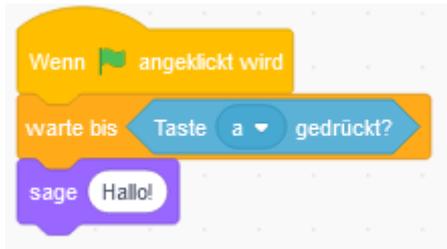
*falls () dann*

Das Skript wird nur bei angegebener Bedingung ausgeführt.

*falls () dann, sonst ()*

Das ausführende Skript hängt von der angegebenen Bedingung ab.





Der Block pausiert sein Skript bis die ausgewählte Bedingung erfüllt ist.

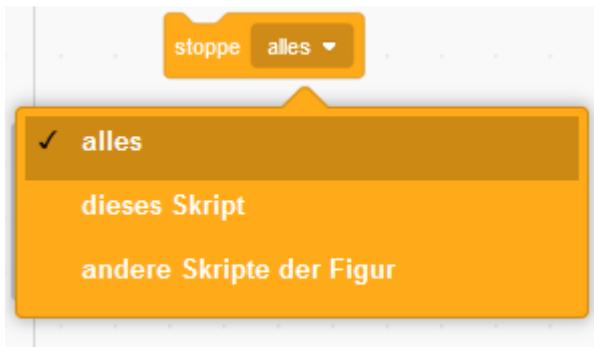
### 6.5.4 Bedingte Ausführung Fußschleife



*wiederhole bis ()*

Das Skript wird so lange ausgeführt, bis die angegebene Bedingung zutrifft.

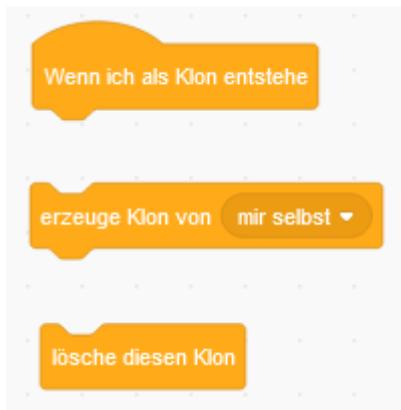
### 6.5.5 Stoppe



Der Stoppe [alles]-Block stoppt zwar das Projekt, hindert jedoch die Kopf-Blöcke nicht daran, erneut aktiviert zu werden.

Der Stoppe [dieses Skript]-Block stoppt das Skript, an dessen Ende er steht, verhindert jedoch nicht, dass der Ereignisblock (durch den das Skript ausgeführt wird) erneut aktiviert wird. Der Stoppe [andere Skripte der Figur]-Block stoppt das Skript, an dessen Ende er steht und alle weiteren Skripte des Objektes, das diesen Block ausführt.

### 6.5.6 Klonen



Es erspart viel Programmierarbeit und ist eleganter, eine einzige Figur während der Laufzeit als Kopie zu klonen, anstatt bereits beim Programmieren des Projekts eine bestimmte Anzahl Kopien zu erstellen. Man braucht dann nämlich nur diese eine Figur zu bearbeiten, statt alle programmierten Kopien, ist sich sicher, dass alle Kopien wirklich gleich sind und kann stets genau die beliebig hohe Anzahl von Kopien der Figur erzeugen, die benötigt wird. "Erzeuge Klon von ()" wird benutzt, um eine Figur zu klonen. Man kann auch Klone erneut klonen.

"Wenn ich als Klon entstehe" lässt das vom Klon auszuführende Skript starten, sobald dieser Klon erzeugt wurde. "Lösche diesen Klon" löscht den Klon am Ende seines Skriptes

## 6.6 Fühlen

Hier sind alle Programmelemente, die Informationen über einen Zustand eines Objektes oder eines Eingabegerätes enthalten, zusammengefasst.

### 6.6.1 Wird [ ] berührt/gedrückt?



Wird das Objekt vom Mauszeiger berührt, gibt es eine entsprechende Reaktion. (Hier Sprechblase)

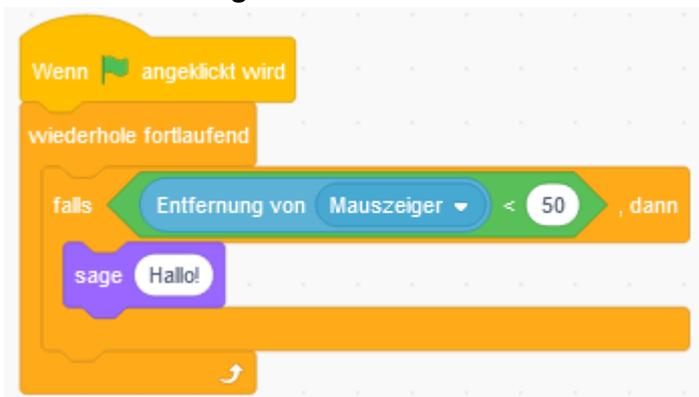
Es lässt sich auch austauschen gegen:

Wird Farbe [ ] berührt? (z.B. bei Labyrinth)

Wird eine Taste berührt? (z.B. Objektsteuerung)

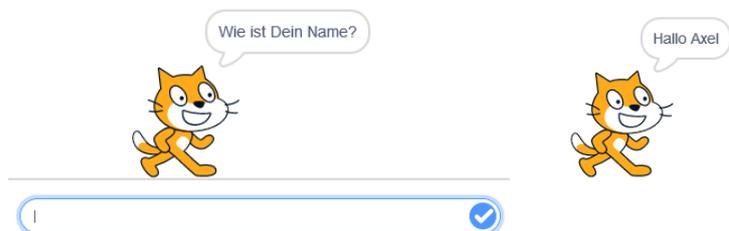
Wird Maustaste gedrückt? (z.B. als Feuertaste)

### 6.6.2 Entfernung



Er gibt die Entfernung einer Figur in Pixeln zu einer anderen Figur oder zum Mauszeiger an. Dabei wird immer vom Drehpunkt ausgegangen. Es kann z.B. die Kollisionen von Figuren vorhersagen.

### 6.6.3 Frage [ ] und warte



Dieser Block wird gebraucht, wenn ein Skript abhängig vom Input des Benutzers ausgeführt werden soll. Er erzeugt eine Box am unteren Rand der Bühne. Wird der Block von einer Figur ausgeführt, so wird die Frage in ihrer Sprechblase angezeigt.

### 6.6.4 Maus-Position



Der Block gibt die X-Position des Mauszeigers zurück. Im Beispiel wird er benutzt, um ein Objekt auf die Mausbewegung reagieren zu lassen. Wird die Maustaste berührt während das Objekt den schwarzen Punkt berührt, wird „Tor“ angezeigt.

### 6.6.5 Lautstärke



Er gibt zurück, wie laut ein vom Mikrofon empfangener Ton ist. Dieser Wert kann von 0 (kein Ton) bis 100 (lauter Ton) variieren.

Dieser Block funktioniert nur, wenn ein Mikrofon in deinen PC integriert oder an ihn angeschlossen und aktiviert ist. Ansonsten gibt der Block 0 oder -1 zurück.

### 6.6.6 Stoppuhr



Es gibt 2 Blöcke. Den runden "Stoppuhr"-Block, der den aktuellen Wert anzeigt und den "setze Stoppuhr zurück"-Block. Dieser sollte mit einem "Wenn grüne Flagge angeklickt"-Block verbunden werden, denn die Stoppuhr läuft immer weiter, auch wenn das Stoppschild gedrückt wurde.

werden, denn die Stoppuhr läuft immer weiter, auch wenn das Stoppschild gedrückt wurde.

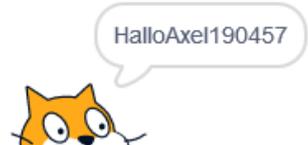


Wird der Haken vor dem Symbol gedrückt, wird die Zeit auf der Bühne angezeigt.

### 6.6.7 Benutzername



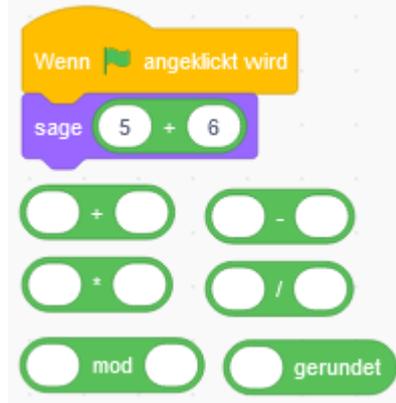
Um den Benutzernamen auszugeben, muss man als Scratchler angemeldet sein.



## 6.7 Operationen

Dies sind die Blöcke die Mathematische Funktionen und Textverarbeitung ausführen können.

### 6.7.1 Operation mit Zahlenergebnis



Zahlen können miteinander verbunden, und das Ergebnis ausgegeben werden. Wie in der Mathematik typisch können hier 2 Zahlen addiert, subtrahiert, multipliziert und dividiert werden. Dabei gibt es 2 Arten von Zahlen:

-Ganzzahl (Integer), z.B. [0, -4, 67, 200, ...]

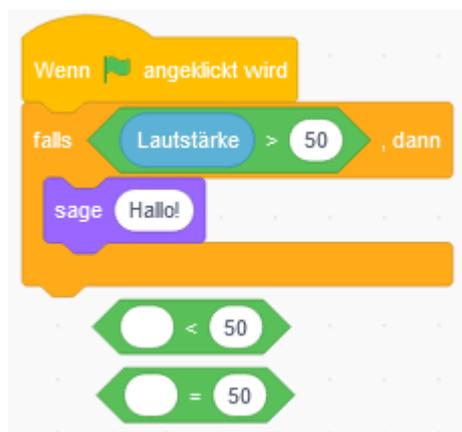
-Gleitkommazahl (Float), z.B. [0.1, -27.8, ...] mit einer Genauigkeit von 6 Stellen hinter dem Komma.

Die Eingabe von Kommazahlen erfolgt hierbei durch einen Punkt.

Mit „mod“ wird der ganzzahlige Rest einer Division ausgegeben.

Mit „gerundet“ wird der gerundete ganzzahlige Wert ausgegeben.

### 6.7.2 Operation mit Wahrheitsergebnis



Bedingungen sind wesentlicher Bestandteil von Fallunterscheidungen.

Ein Wert-Block gibt einen Wert zurück. Dieser Wert kann sowohl eine Zahl, als auch eine Zeichenkette (zum Beispiel ein Wort) sein. Wahrheitsblöcke sind eine spezielle Form von Wertblöcken, die entweder "wahr"(1) oder "falsch"(0) zurückgeben. Im Beispiel sagt das Objekt „Hallo“ wenn die Lautstärke größer als 50 ist.

Es besteht die Möglichkeit auch auf „größer als“ sowie „gleich“ zu prüfen

### 6.7.3 Logische Operatoren



Logische Operatoren werden zur Bildung komplexer Bedingungen benötigt. Die folgenden Tabellen zeigen, wie die drei logischen Grundoperatoren festgelegt sind.

und			oder			nicht	
a	b	x	a	b	x	a	x
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		



Nur wenn Taste „a“ und Taste „b“ gedrückt werden, wird der Text „Hallo“ ausgegeben.

### 6.7.4 Zufallszahlen



Er wählt eine Zufallszahl zwischen den beiden eingegebenen Werten. Werden keine Kommas in den Eingaben benutzt, so ist auch das Ergebnis ohne Nachkommastellen. Wird jedoch ein Komma verwendet, so werden immer drei Nachkommastellen angegeben, egal wie viele Nachkommastellen die beiden Eingaben haben.

Es können direkt Zahlen eingegeben werden, oder man setzt andere Wertblöcke ein (z.B. Variablen, Lärmpegel etc.).

### 6.7.5 Arbeit mit Texten



In der Ausgabe werden die beiden Texte verbunden ausgegeben. Hinter Apfel muss ein Leerzeichen mit eingegeben werden, damit die Zeichenketten nicht hintereinanderstehen.



Es ist auch möglich, die Länge eines Wortes auszugeben. Bei bedarf ist es auch möglich, den X-ten Buchstaben des Wortes anzuzeigen.

Weiterhin besteht die Möglichkeit abzufragen, ob ein bestimmter Buchstabe in dem Text enthalten ist.

### 6.7.6 Funktionen



Es stehen folgende Funktionen zur Auswahl:

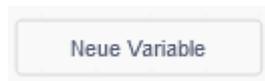
- Betrag (entfernt negative Vorzeichen)
- Abrunden
- Aufrunden
- Wurzel
- sin (Sinus) \*
- cos (Kosinus) \*
- tan (Tangens) (ist gleich Sinus geteilt durch Kosinus)
- asin "Arkussinus" (ist das Gegenteil von Sinus)
- acos (Arkuskosinus) (ist das Gegenteil von Kosinus)
- atan (Arkustangens) (ist das Gegenteil von Tangens)
- ln (Natürlicher Logarithmus)
- log (Logarithmus)
- e^ (e-Funktion) (ist das Ergebnis von e<sup>x</sup>)
- 10^ (Zehnerpotenz) (ist das Ergebnis von 10 hoch x).

\* Um die Koordinaten von einer Position im Kreis mit einem Pixel Radius von einem bestimmten Winkel vom Mittelpunkt aus zu berechnen.

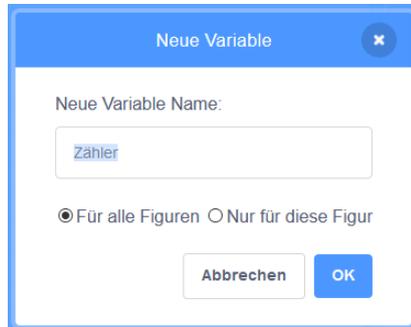
## 6.8 Variablen

Variablen sind Platzhalter mit zugewiesenen Werten, die während des Programmablaufs vom Benutzer oder von einem Skript verändert werden können. Sie werden häufig verwendet, um die Zeit einer Stoppuhr zu speichern, Verrechnungen durchzuführen oder einen Punktestand anzeigen zu lassen.

### 6.8.1 Variable anlegen



Den Button „Neue Variable“ drücken. Es erscheint ein neues Fenster. Name vergeben und OK drücken.



Die neue Variable erscheint nun im rechten Fenster. Durch Anklicken des Kästchens wird sie auf der Bühne sichtbar.

### 6.8.2 Variable benutzen



Die neu erstellte Variable wird als erstes auf „0“ gesetzt. Im folgenden Programm wird die Variable immer um „1“ erhöht, wenn die Leertaste gedrückt wird.

(Die Pause von 0,2 Sek. wurde eingefügt, damit der Zähler nicht bei zu langem Tastendruck arbeitet)



Die Variable wird z.B. zu Programmbeginn auf einen Anfangswert gesetzt.

Die Variable wird im Programm durch Aktivitäten im Wert verändert.

Die Variable wird im laufenden Programm auf der Bühne sichtbar gemacht.

Die Variable wird im laufenden Programm auf der Bühne unsichtbar gemacht.

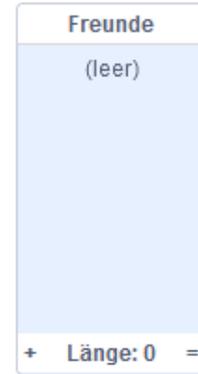
## 6.9 Listen

Listen sind eine Art kompakte Ansammlung von Variablen. Sie sind besonders nützlich für Programme, die eine große, unbestimmte Anzahl an Werten abspeichern und dann wiederverwenden müssen. In Adventure Spielen muss man oft irgendwelche Sachen aufsammeln. Perfekt geeignet für eine derartige Sammlung der Items stellt eine Liste an einem Bildschirmrand da.

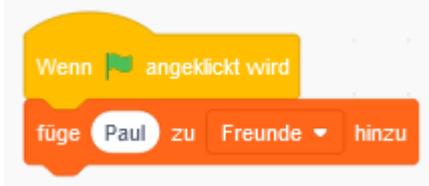
### 6.9.1 Liste erstellen



Nach dem Anklicken des Buttons öffnet sich ein Fenster, wo der Name eingegeben wird. Dann ist die Liste Links als Objekt und auf der Bühne als leere Liste sichtbar.

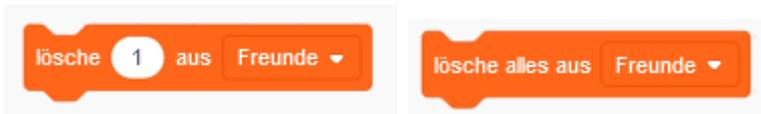


### 6.9.2 Liste füllen

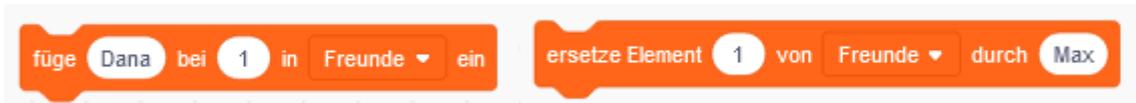


Das Objekt erwarten den Wert und den Namen der Liste, wo dieser Wert gespeichert werden soll.

### 6.9.3 Liste bearbeiten

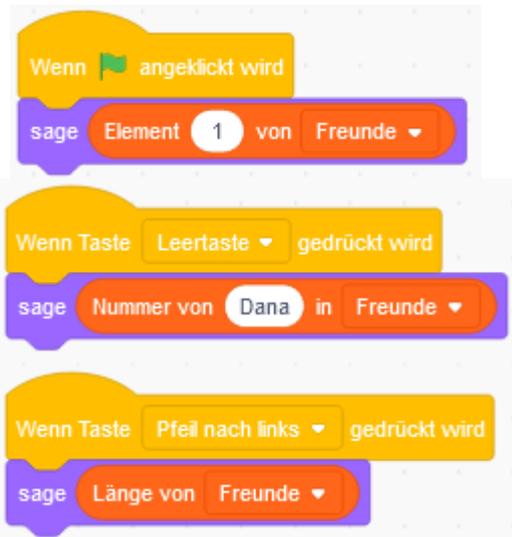


Inhalte können einzeln oder auch komplett gelöscht werden.



Inhalte können an bestimmten Stellen hinzugefügt oder ersetzt werden.

### 6.9.4 Mit Listen arbeiten



Über die Elemente Nummer, lassen sich die Inhalte gezielt ausgeben oder es werden Reaktionen in Abhängigkeit vom Inhalt der Liste programmiert.

### 6.9.5 Liste speichern

Mit einem Rechtsklick auf die Liste, kann diese als \*.txt Datei Exportieren bzw. Importieren. Exportierte Listen verwenden das UNICODE-Format (UTF-8). Zum weiterverarbeiten muss man diese Einstellung beim Import anwenden, sonst kommen z. B. statt der deutschen Umlaute irgendwelche Ersatzzeichen.

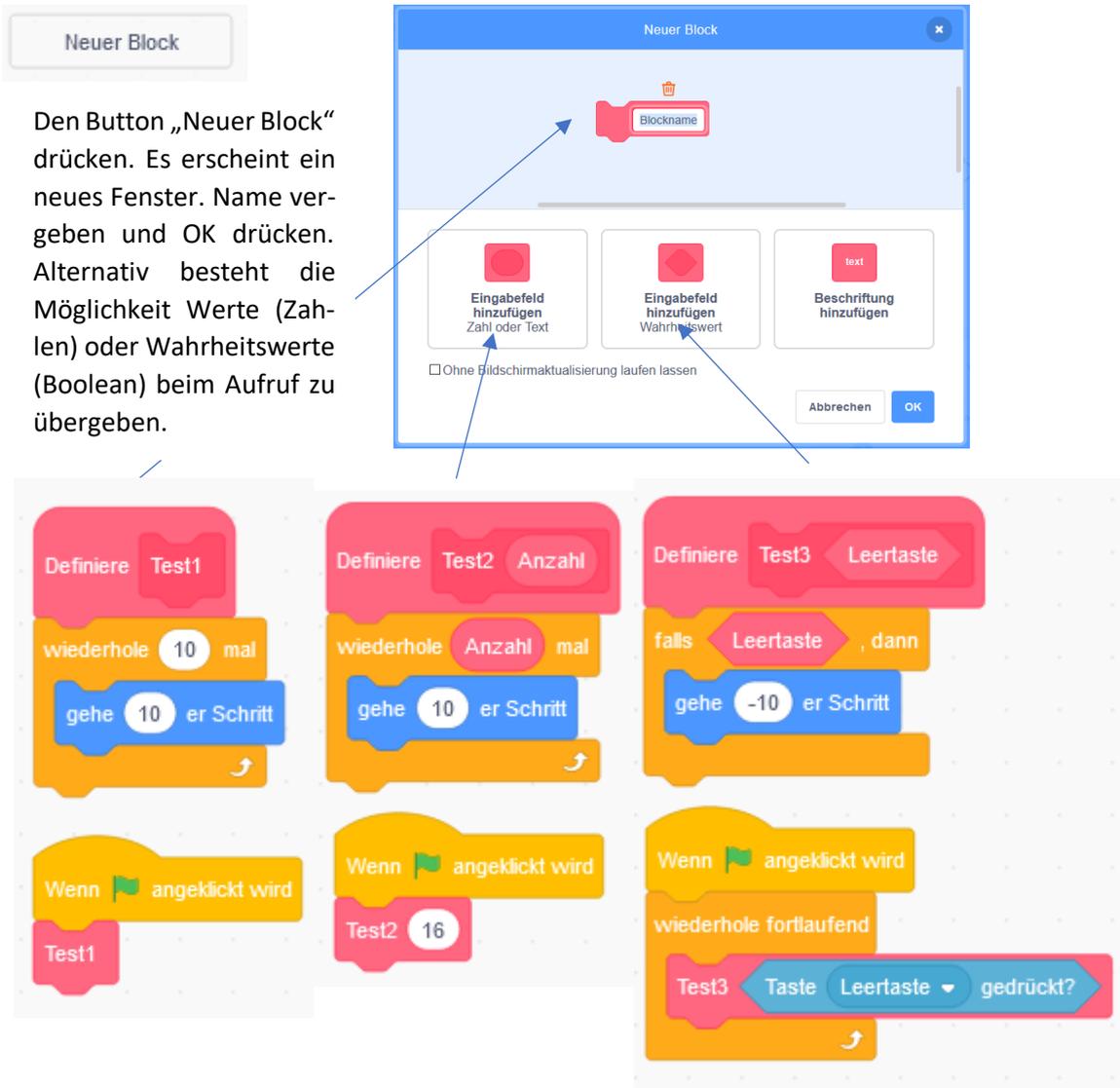


## 6.10 Meine Blöcke (Unterprogramme)

Ein Unterprogramm ist ein Teil eines Scratch-Projekts das von Skripten der Figur aufgerufen werden kann, um eine Aufgabe zu übernehmen. Nach der Ausführung des Unterprogramms wird das aufrufende Skript weiter ausgeführt. Mit Unterprogrammen kann man die Scratch-Blöcke somit um weitere Funktionen erweitern.

### 6.10.1 Block erstellen

Den Button „Neuer Block“ drücken. Es erscheint ein neues Fenster. Name vergeben und OK drücken. Alternativ besteht die Möglichkeit Werte (Zahlen) oder Wahrheitswerte (Boolean) beim Aufruf zu übergeben.

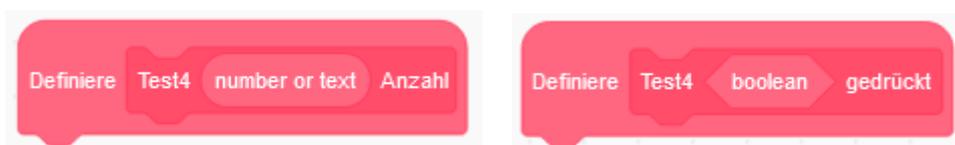


Test 1 Der obere Teil der Darstellungen ist die Erstellung des Unterprogrammes und darunter der entsprechende Aufruf. Hier ein einfacher Aufruf.

Test 2 Hier wird beim Aufruf ein Wert übergeben, der bewirkt, dass das Objekt 16 mal die Schleife durchläuft.

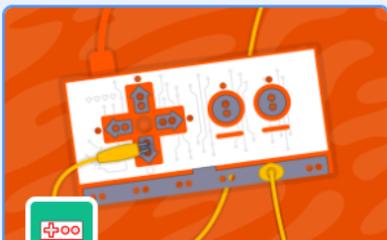
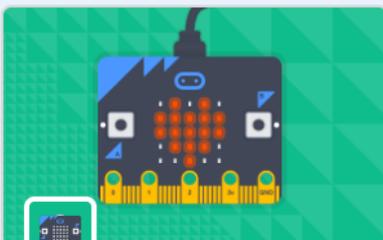
Test 3 Hier wird beim Aufruf ein Wahrheitswert übergeben, der bewirkt, dass die Schleife erst bei gedrückter Leertaste durchlaufen wird.

Test 4 Nach dem Übergabefeld Wert bzw. Wahrheitswert kann noch ein Text angehängt werden.



## 7 Erweiterte Elemente

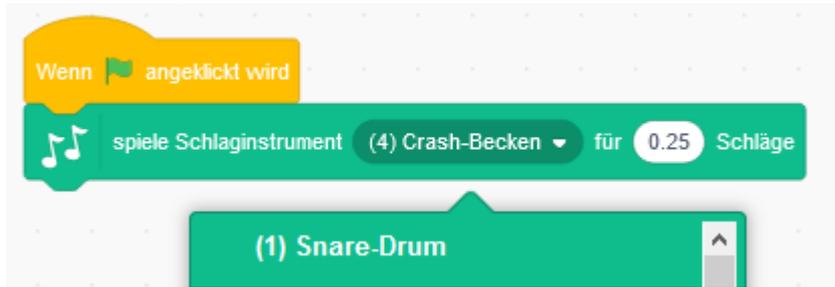
Unten links auf den Button  drücken, um das Auswahlfenster zu öffnen.

  <p><b>ftDuino</b> Control fischertechnik models.</p> <p>Zusammenarbeit mit Till Harbaum <a href="http://ftduino.de">http://ftduino.de</a></p>	  <p><b>Musik</b> Spiele Instrumente und Schlagzeug.</p>	  <p><b>Malstift</b> Zeichne mit deinen Figuren.</p>
  <p><b>Video-Erfassung</b> Erfasse Bewegung mit der Kamera.</p>	  <p><b>Text zu Sprache</b> Bring deinen Projekten das Sprechen bei.</p> <p>Erfordert  Zusammenarbeit mit Amazon Web Services</p>	  <p><b>Übersetzen</b> Übersetze den Text in viele Sprachen.</p> <p>Erfordert  Zusammenarbeit mit Google</p>
  <p><b>Makey Makey</b> Mach alles zu einer Taste.</p> <p>Zusammenarbeit mit JoyLabz</p>	  <p><b>micro:bit</b> Verbinde deine Projekte mit der realen Welt.</p> <p>Erfordert   Zusammenarbeit mit micro:bit</p>	  <p><b>LEGO MINDSTORMS EV3</b> Baue interaktive Roboter und mehr.</p> <p>Erfordert   Zusammenarbeit mit LEGO</p>

Es gibt Erweiterungen, für die keine zusätzliche Hardware erforderlich ist. Andererseits gibt es Erweiterungen, bei der zusätzliche Eingabelemente oder andere Kleinstrechner angeschafft werden müssen, um interagieren zu können.

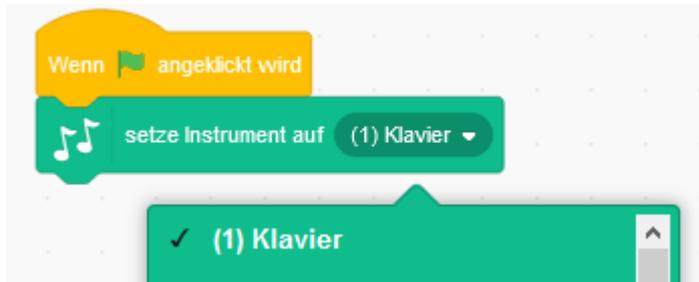
## 7.1 Musik

### 7.1.1 Schlaginstrument



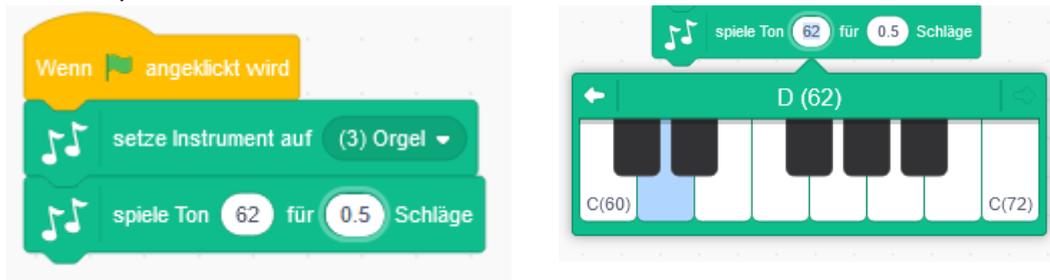
Das Element kann mit 18 verschiedenen Schlaginstrumenten ausgewählt werden. Der Tonlänge ergibt sich aus dem eingegeben Schlag-Wert und der aktuellen Geschwindigkeit. (Standard Wert 60)

### 7.1.2 Setze Instrument auf



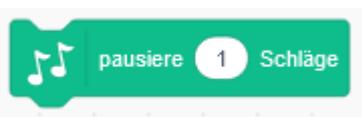
Bei der Instrumentenauswahl stehen 21 Instrumente zur Verfügung. Hier wird lediglich das Instrument vorausgewählt.

### 7.1.3 Spiele Ton



Der gespielte Ton wird über eine Zahl ausgewählt, welche die Nummer der Note in einem durchnummerierten Oktavsystem angibt. Der tiefste Notenwert ist "60" und der höchste ist "72".

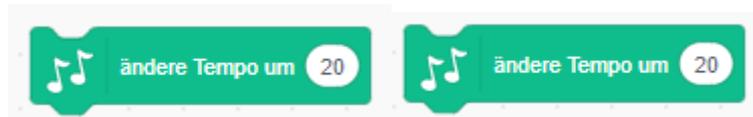
### 7.1.4 Pausiere



### 7.1.5 Tempo



Er gibt die Geschwindigkeit zurück, die man mit dem `ändere Geschwindigkeit um ()` und dem `setze Geschwindigkeit auf ()` Schläge/Minute verändern kann. Das Tempo wird in bpm, das heißt beats per minute, also Schläge pro Minute gemessen. Ist zum Beispiel das Tempo 60 bpm, so dauern 0.5 Schläge 0.5 Sekunden. Ist das Tempo jedoch 120 bpm, so dauern 0.5 Schläge 0.25 Sekunden.



## 7.2 Malstift

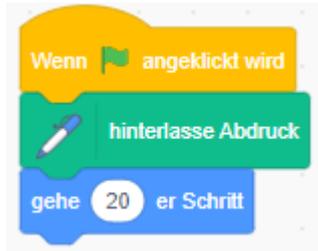
Der Malstift von Scratch kann zum Zeichnen von Punkten, Linien oder Flächen verwendet werden.

### 7.2.1 Lösche alles



Zuerst sollte immer das vorher gemalte von der Bühne gelöscht werden.

### 7.2.2 Hinterlasse Abdruck



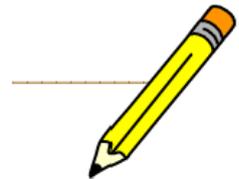
Bei jedem Start des Programms geht die Katze 20 Schritte vorwärts, aber der alte Ausgangspunkt bleibt weiterhin sichtbar.



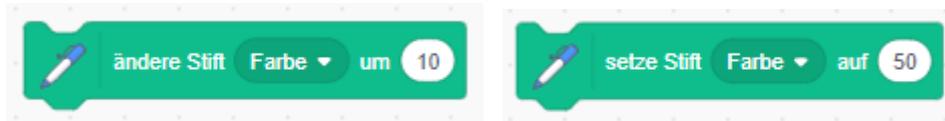
### 7.2.3 Zeichnen



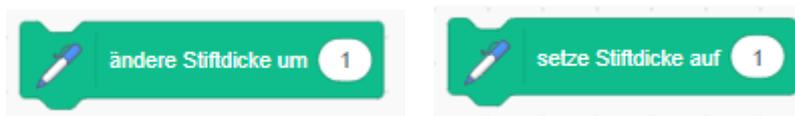
Um zu zeichnen wird als erstes der Stift eingeschaltet. Bei Bedarf kann er auch abgeschaltet werden. Als nächstes wird die Stiftfarbe festgelegt. Nun kann die eigentliche Darstellung erfolgen, indem das Objekt eine X- bzw. Y-Ausrichtung erfährt.



### 7.2.4 Optionen



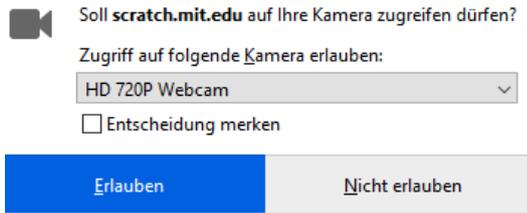
Die Stiftfarbe kann im Programm um einen Wert geändert bzw. auf einen Wert gesetzt werden.



Ebenso kann die Stiftdicke um einen Wert geändert bzw. auf einen Wert gesetzt werden.

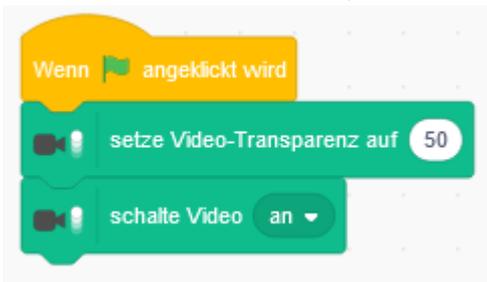
### 7.3 Video

Mit dem Video-Block ist es möglich, auf die Web-Cam zuzugreifen und das Bild in das Projekt einzubinden. Das von der Kamera eingebundene Bild kann aber nicht gespeichert werden, es wird nur einmalig im Projekt gezeigt.



Wenn der Block eingebunden wurde, kommt als erstes die Abfrage vom System, ob Scratch auf die Kamera zugreifen darf.

#### 7.3.1 Einschalten / Transparenz



Zuerst kann die Kamera ein bzw. ausgeschaltet werden. Auch die Transparenz des Videobildes kann vor dem Einsatz festgelegt werden. Dabei gilt: Transparenz 0 = starkes Bild; Transparenz 100 = ganz schwaches Bild.

#### 7.3.2 Auswertung der Bewegung



Es können folgende Auswertungen vorgenommen werden:

Wird das Objekt (Figur) durch eine „Bewegung“ berührt oder von einer „Richtung“ erreicht. Bild 1  
Wird auf der Bühne eine Bewegung festgestellt. (z.B. für Raumüberwachung). Bild 2

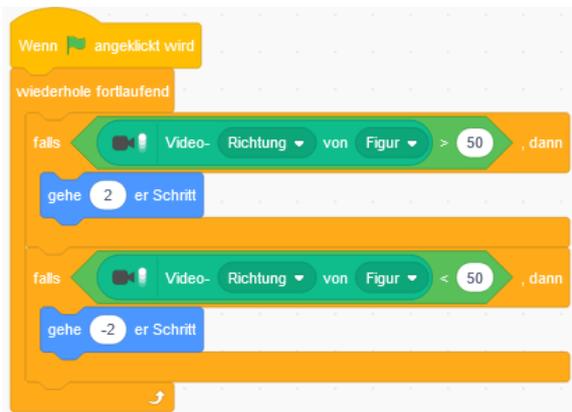
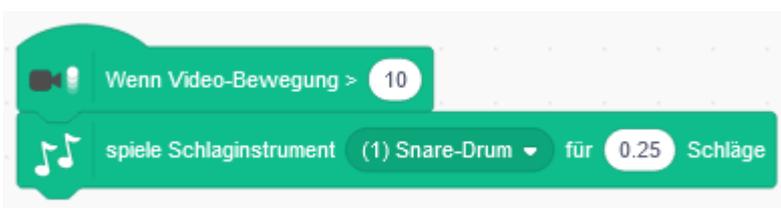


Bild 1



Bild 2

#### 7.3.3 Einfache Bewegungsauswertung



Die einfachste Auswertung erfolgt mit diesem Element. Es überwacht jede Bewegung auf der Bühne und es kann ein entsprechendes Ereignis programmiert werden.

## 7.4 Sprachausgabe

Mit den folgenden 2 Blöcken, ist es möglich geschriebenen Text als Sprachausgabe vorlesen zu lassen. Für die Ausgabe von deutschen Texten reicht der Block „Text zu Sprache“. Soll dieser Text in einer anderen Sprache ausgegeben werden, wird noch der Block übersetzen benötigt.

### 7.4.1 Text zu Sprache



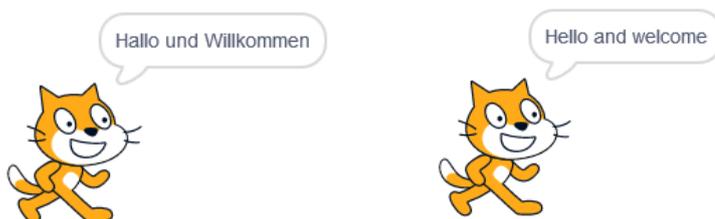
Als Voreinstellungen können noch die Stimme und die Sprache gewählt werden. Dabei erfolgt aber keine Übersetzung, sondern der Akzent wird definiert!

### 7.4.2 Übersetzen

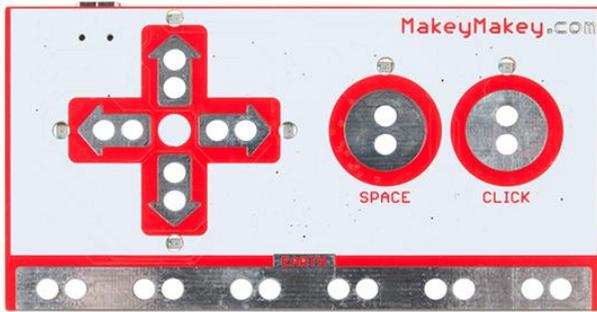


Wie die letzten beiden Zeilen zeigen, kann mit dem Block „Übersetze“ zum einen die Sprachausgabe, aber auch die Textausgabe übersetzt dargestellt werde.

Im nebenstehenden Beispiel wird zuerst die Stimme gesetzt. Dann wird eine Variable mit dem Namen Text angelegt, der der Satz „Hallo und Willkommen“ übergeben wird. Nun erfolgt die Text und Sprachausgabe in Deutsch. Nach einer Sekunde Pause, wird dieser Text jeweils übersetzt und als Text und Sprache in Englisch ausgegeben.

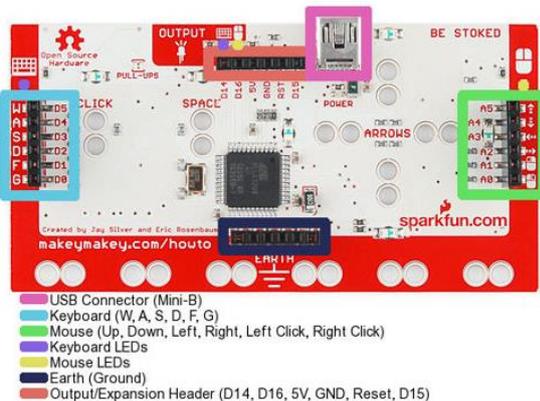
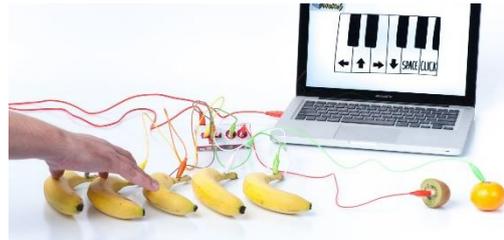


## 7.5 Makey Makey



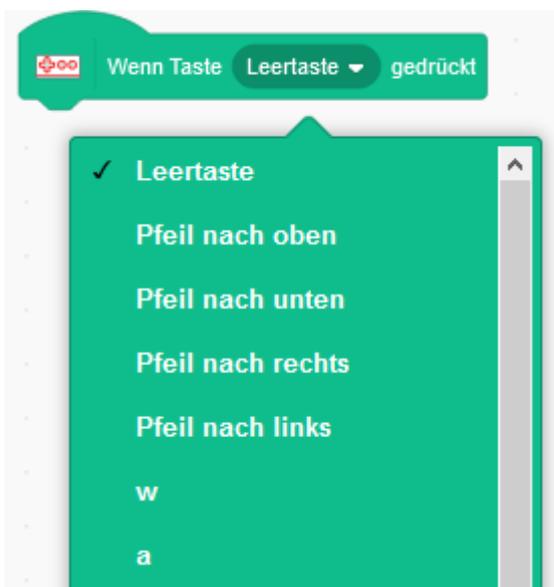
Makey Makey ermöglicht einem, Alltagsobjekte in Tasten umzuwandeln und sie mit seinem Computer auch als solche zu benutzen. Es ist für ca.15 Euro im Internet zu beziehen. Makey Makey schließt und öffnet Stromkreise zwischen einem Menschen und einem Objekt. Dabei arbeitet es mit sehr geringer Spannung, so dass das System ungefährlich ist und man es nicht spürt.

Man schließt ein Kabel an ein Objekt, beispielsweise eine Banane, an. Dasselbe tut man mit der eigenen Hand. Berührt man nun die Banane, wird der Stromkreis geschlossen und das Makey Makey erkennt, dass eine Taste gedrückt wurde. Dabei benutzt es USB-Anschlüsse, so dass der Computer diese Banane als normale Taste einer Tastatur oder einer Maus wahrnimmt.



Die Rückseite des Makey Makey bietet weitere 12 Anschlussmöglichkeiten für Drähte (6 für Tastatur, 6 für Maus) sowie weitere Anschlüsse, über die z.B. ein kleiner Servo-Motor oder LEDs betrieben oder Erdungskabel verbunden werden können.

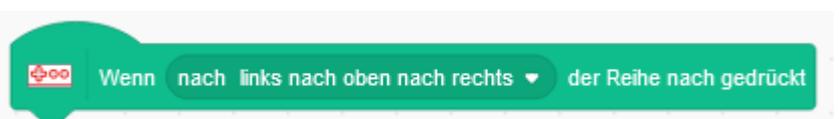
### 7.5.1 Tastenabfrage



Wie bei den Startmöglichkeiten durch den Benutzer können hier die Berührungspunkte des Makey Makey abgefragt werden. Zusätzlich ist es noch möglich, die Tasten w,a,s,d,f,g abzufragen. Die dafür erforderlichen Sensoren werden auf der Rückseite (blau eingekreist) abgegriffen.

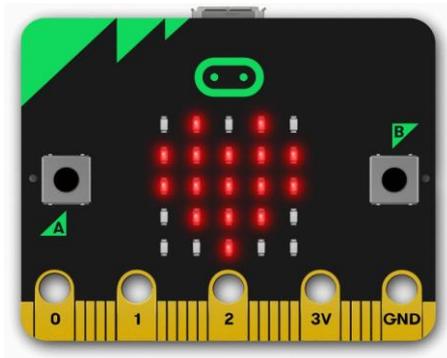


Da das Eingabegerät als USB-Objekt mit dem Rechner verbunden wird, können auch die bekannten Kopfblöcke verwendet werden.



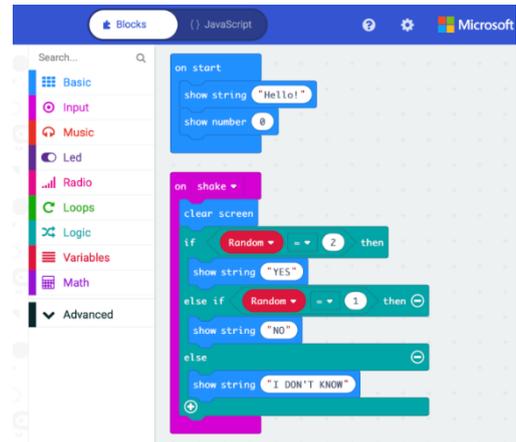
Zusätzlich gibt es die Abfrage für bestimmte Tastenfolgen.

## 7.6 Micro:bit



Zur Interaktion gibt es 2 Knöpfe, 25 LED's, Lagesensor und Kompass.

Der BBC micro:bit ist ein preiswerter Einplatinencomputer, der von der British Broadcasting Corporation im Juli 2015 vorgestellt wurde. Er ist im Internet inkl. Batteriehalter für ca. 25 Euro erhältlich. Vorrangiges Ziel des Projekts ist die Verbesserung der Schulbildung im Bereich der Informationstechnik. Das Gerät ist als Einplatinencomputer auf der Basis eines ARM-Mikrocontrollers realisiert und kann mittels verschiedener webbasierter Entwicklungsumgebungen programmiert werden.



Der Block Editor von Microsoft ist ein grafisches Programmier-Tool mit Ähnlichkeiten zu Scratch, bei dem man per drag&drop Programm-Konstrukte auf eine Arbeitsfläche schiebt und dann die Details wie variable Namen und Werte von Konstanten einträgt.

### 7.6.1 Einrichtung



Der micro:bit wird per USB angeschlossen



„scratch-microbit-1.1.0.hex“ aus dem Netz laden



micro:bit hat ein eigenen Laufwerksbuchstaben. Die geladene Datei dorthin kopieren.



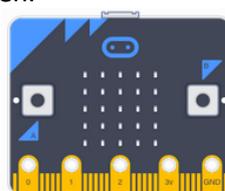
„ScratchLinkSetup.msi“ aus dem Netz laden und installieren.



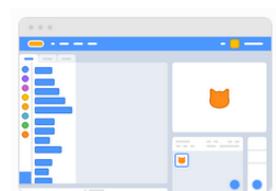
Icon ist dann in der Taskleiste sichtbar.



Für mehr Bewegungsfreiheit kann die Benutzung auch über Bluetooth erfolgen.



Wird der micro:bit per Bluetooth benutzt jetzt die Batterie anschließen



Den Scratch-Editor öffnen (als Programm oder vom Browser)



Auf Erweiterung klicken und micro:bit auswählen



Warten, bis der micro:bit gefunden wurde

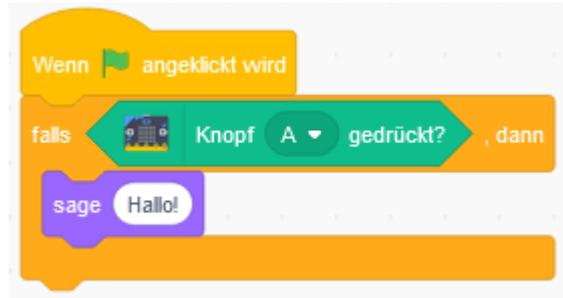
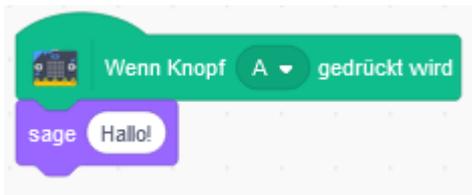


Nun den Button „Verbinde“ drücken



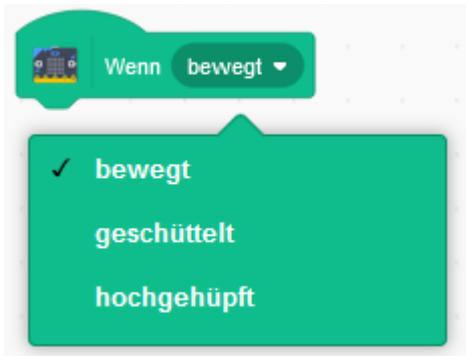
Bei Erfolg, sind die Steuerelemente des micro:bit zu sehen

### 7.6.2 Abfrage der Knöpfe A und B

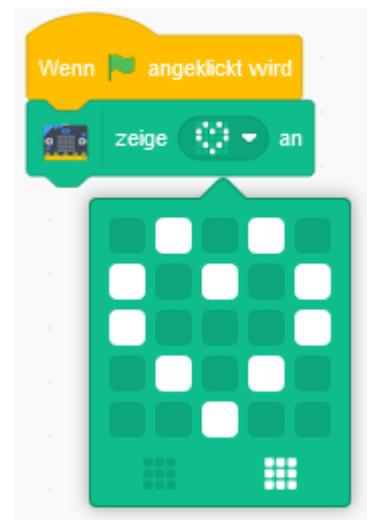


Die Knöpfe A, B bzw. A und B können als Startobjekt benutzt werden, oder für eine Bedingung genutzt werden.

### 7.6.3 Bewegungsabfrage



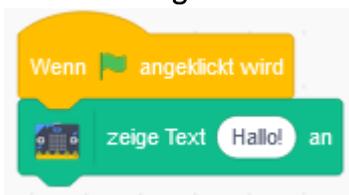
Mit diesem Startobjekt kann ein Programm beginnen, wenn der micro:bit bewegt wird, geschüttelt wird, oder man mit ihm hochhüpft.



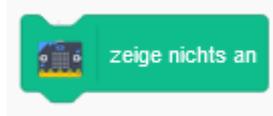
### 7.6.4 LED ansteuern

Die 25 LED's sind in einer 5 mal 5 Matrix angeordnet. Das Anzeigefenster kann geöffnet werden, um die einzelnen Pixel direkt anzusprechen. Ganz unten besteht die Möglichkeit, alle LED's an- oder auszuschalten.

### 7.6.5 Anzeigen

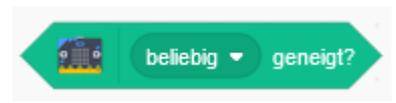


Wird ein Text zur Anzeige gebracht, scrollt dieser durch die LED-Matrix.



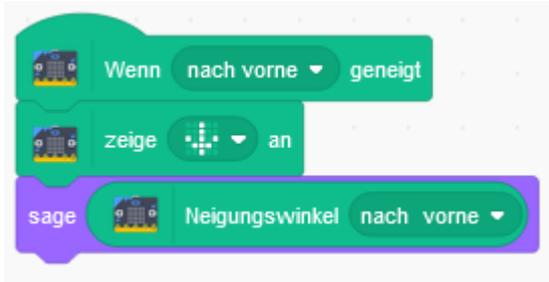
Hiermit kann die Anzeige ausgeschaltet werden.

### 7.6.6 Lagesensor auswerten



Die Auswertung der Lage kann entweder als Startobjekt oder als Bedingungsobjekt verwendet werden. Im Beispiel zeigt der Pfeil in die Richtung, nach der der micro:bit ausgerichtet wurde.

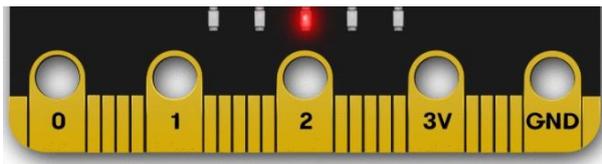
### 7.6.7 Neigungswinkel anzeigen



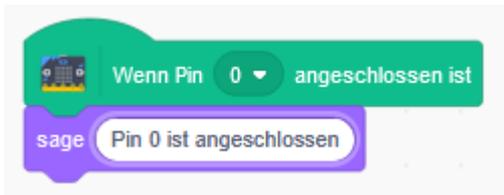
Zu Auswertungszwecken, kann es notwendig sein, den Neigungswinkel anzuzeigen.



### 7.6.8 Abfrage weiterer Pins



Am unteren Ende hat der micro:bit noch weitere Kontakte. Diese können über eine Adapterplatine oder über sogenannte Bananenstecker abgegriffen werden. Die Pins 0 bis 2 können mit dem folgenden Element geprüft werden.



### 7.6.9 Hardwarerweiterungen

Für den Micro:bit gibt es eine große Anzahl an Erweiterungen, um den Einplatinencomputer optimaler zu benutzen, bzw. um den Spaßfaktor zu erhöhen. (Hier nur eine Auswahl). Diese Adapter werden entweder über die 5 großen Löcher mit Schrauben verbunden, oder der micro:bit wird in den Adapter geschoben



Adapter zum besseren Abgriff der Ein- und Ausgänge



Klavier-Adapter



Lautsprecher-Adapter



Joystick-Adapter



Batterie-Adapter



Gehäuse

## 7.7 Lego Boost



Das Lego-Boost-Set kann als Komplettbaukasten für ca. 120 Euro erworben werden. So macht es nur Sinn diesen Block zu benutzen, wenn dieser Baukasten bereits erworben wurde.

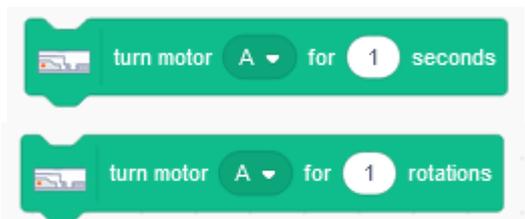


Für die Steuerung mit Scratch brauchen wir nur die verfügbare Hardware. Das ist zum einen die Bewegungsbasis „Move Hub“ mit zwei programmierbaren Motoren (A und B), dem Bluetooth-Modul, den 2 Ein- bzw. Ausgängen, sowie ein 6-achsiger Neigungssensor. An die Basis können ein weiterer Motor (C) und ein kombinierter Farb- und Abstandssensor angeschlossen werden.

### 7.7.1 Einrichtung

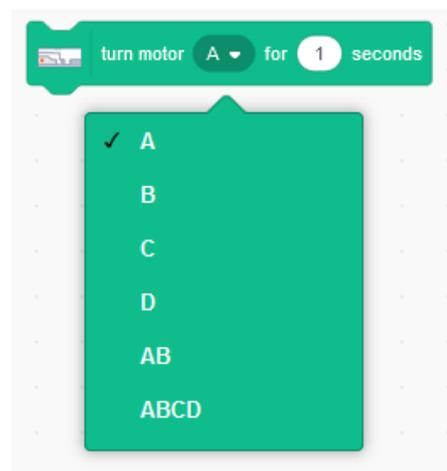
<p>„ScratchLinkSetup.msi“ aus dem Netz laden und installieren.</p>	<p>Icon ist dann in der Taskleiste sichtbar.</p>	<p>Der Move Hub funktioniert nur über Bluetooth.</p>	<p>Den Scratch-Editor öffnen (als Programm oder vom Browser)</p>
<p>Auf Erweiterung klicken und Lego Boost auswählen</p>	<p>Aufforderung, den grünen Knopf am Move Hub zu betätigen</p>	<p>Wenn Verbunden – gehe zum Editor</p>	<p>Bei Erfolg, sind die Steuerelemente des Move Hub zu sehen.</p>

### 7.7.2 Motorsteuerung



Die Motoren können für eine Zeit bzw. für eine Anzahl von Rotationen programmiert werden.

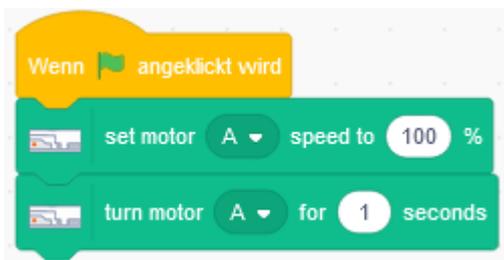
Motor A und B sind im Move Hub. Motor C und D sind als externe Motoren links und rechts ansteckbar.



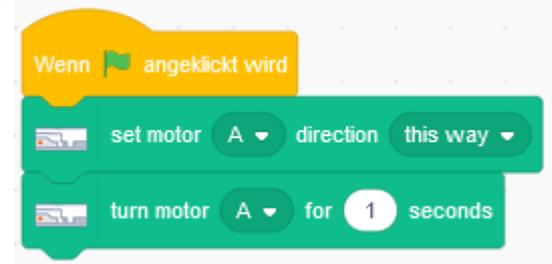


Mit „turn motor [] on“ bzw. “turn motor [] off” können die Motoren im laufenden Programm für eine bestimmte Bedingung ein und ausgeschaltet werden.

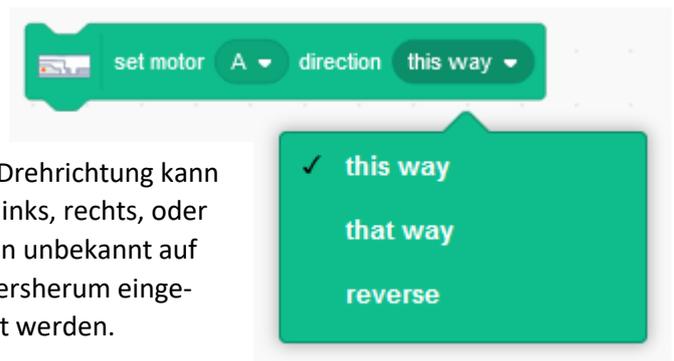
### 7.7.3 Motorsteuerung variieren



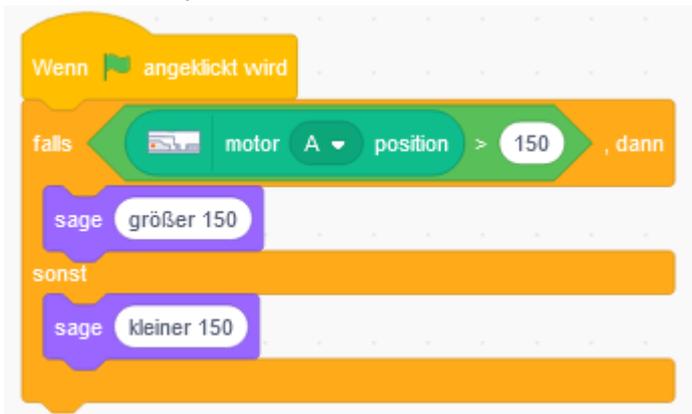
Hier kann die Geschwindigkeit verändert werden. Standardmäßig ist die Geschwindigkeit 50%:



Die Drehrichtung kann auf links, rechts, oder wenn unbekannt auf andersherum eingestellt werden.



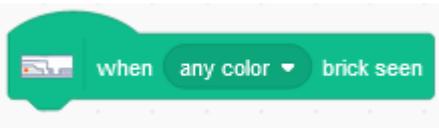
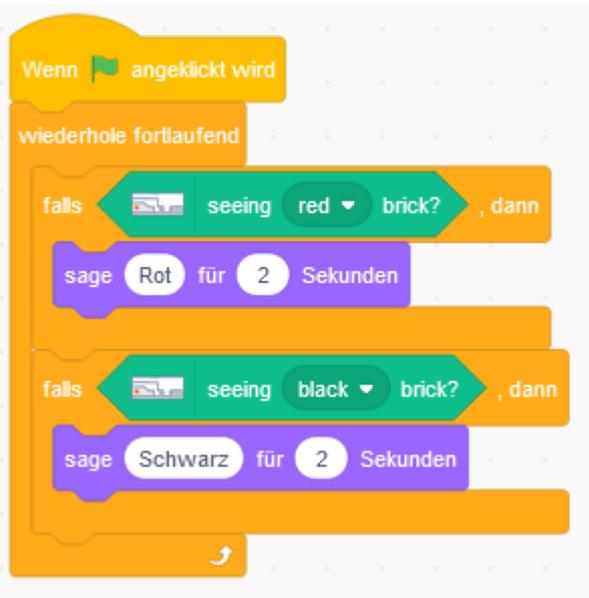
### 7.7.4 Motorposition auslesen



Hier wird ein Wert ermittelt und für Auswertungszwecke übergeben.

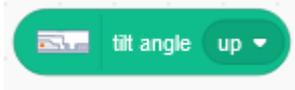
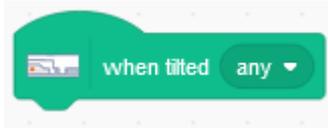
Der Wert kann entsprechend der Gradzahl eines Kreises dem Wert zwischen 0 und 360 annehmen.

### 7.7.5 Farberkennung



Es gibt einen Block für den Start und einen Block für die Auswertung im Programm. Die beste Erkennung erfolgt in einem Anstand von 1 cm.

### 7.7.6 Neigungsauswertung

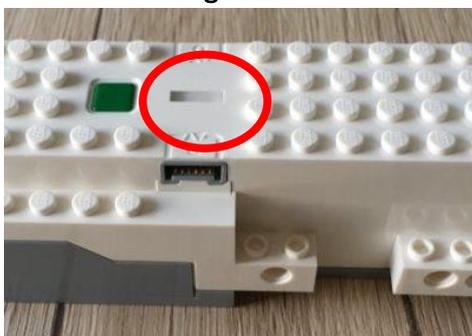


Es gibt je einen Block für den Start und einen Block für die Auswertung im Programm.



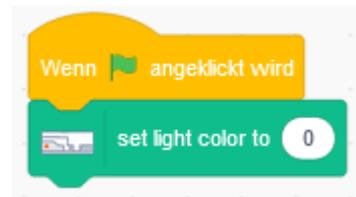
Im Beispiel wird die Bewegung links bzw. rechts ausgewertet. Dabei läuft die Katze 10 mal 2 Schritte in die entsprechende Richtung und die LED zeigt dann rot oder blau an.

### 7.7.7 Farbausgabe



Auf dem Move Hub befindet sich eine Kontroll-LED. Diese kann im Programm explizit angesprochen werden. Dabei gilt: (Zwischenfarben sind möglich)

- Gelb = 5
- Blau = 67
- Grün = 15
- Rot = 0
- Weiß = 44



### 7.7.8 Abstandsmessung

Für die Abstandsmessung des Sensors, sind derzeit keine Elemente implementiert.

## 8 Debugging

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

### 8.1 Skript am falschen Ort

Wurde das Skript bei der richtigen Figur geschrieben?

### 8.2 Vergessene Blocks

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

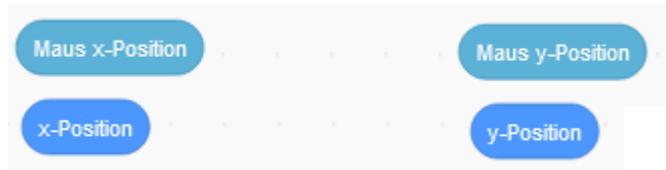
### 8.3 Verwechselte Blocks

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



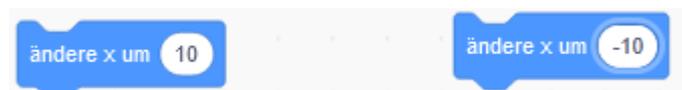
### 8.4 X und y verwechselt

Unter Umständen wurde ein X-Block statt ein Y-Block erwischt oder umgekehrt?



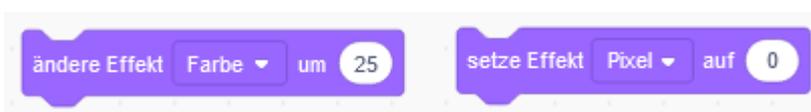
### 8.5 Falsche Zahlen

Gerade wenn ein Script kopiert wurde ist es notwendig, noch die Zahlen oder das Vorzeichen zu ändern.



### 8.6 Falsche Auswahl

Ebenfalls wird das Ändern der Dropdown-Menüs schnell vergessen.



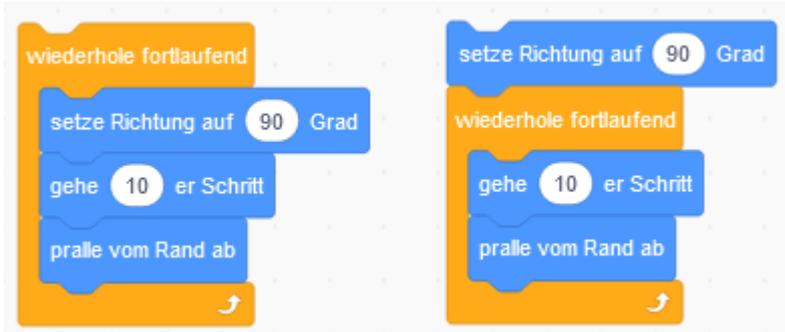
### 8.7 Falsche Reihenfolge

Sind die Blocks in der korrekten Reihenfolge?



### 8.8 Schleifen

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



Oder fehlt gar eine Schleife? Statt so (rechts), sollte das Skript so aussehen (links):



### 8.9 Zeitliche Begrenzung

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



### 8.10 Falsches Timing

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

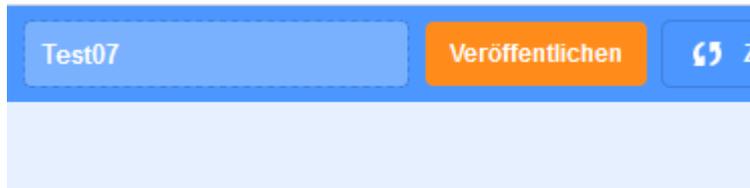


## 9 Veröffentlichen

### 9.1 Voraussetzung

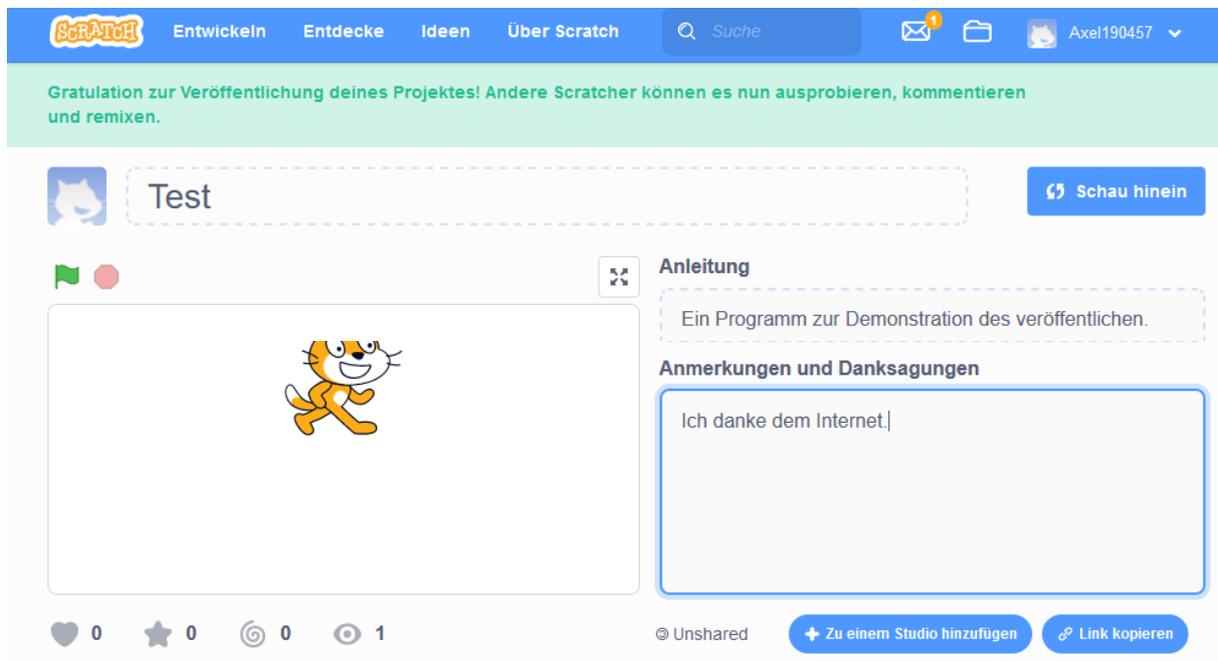
- 1 Man muss mit dem Internet verbunden sein
- 2 Man benötigt ein Projekt, das man hochladen will
- 3 Man muss auf der Scratch-Website registriert sein

### 9.2 Projekt hochladen

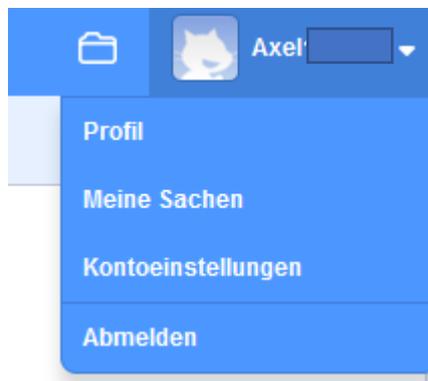


Ist das Programm geschrieben und auf dem PC gespeichert, drückt man auf den Button „Veröffentlichen“

Im folgenden Fenster kann noch einmal der Name eingeben werde, Außerdem kann eine Anleitung sowie eine Anmerkung hinzugefügt werden.



### 9.3 Eigene Projekte anschauen



Wird der Pfeil gedrückt öffnet sich ein Auswahlmnü. Unter „Meine Sachen“. Hier können die veröffentliche und unveröffentliche Projekte eingesehen werden.

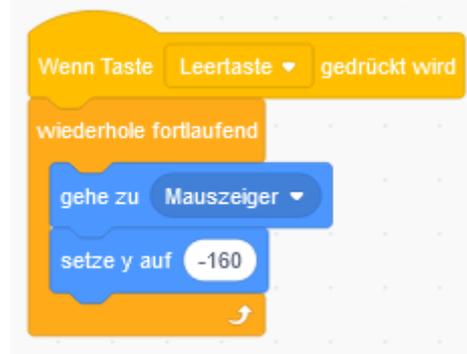
## 10 Beispiele

### 10.1 Kurzbeispiele

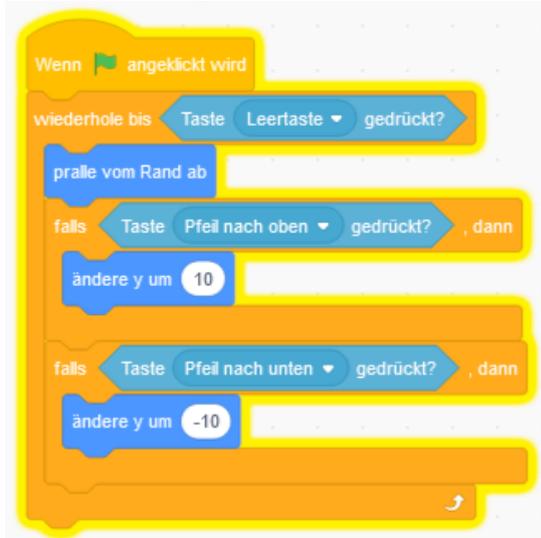
#### Spiel schneller machen



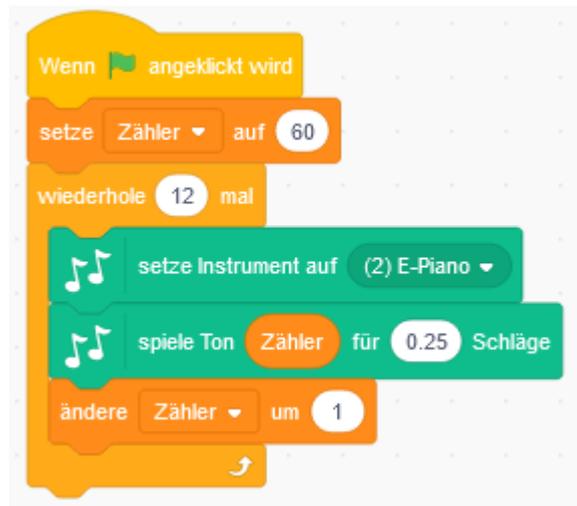
#### Der Maus nur in einer Achse folgen



#### Cursorsteuerung



#### Spiele alle Halbtonschritte einer Oktave



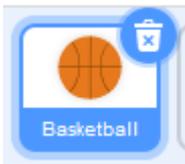
#### Bei Berührung am Rand umdrehen



#### Laufbewegung



## 10.2 Spiel Pong



Programm für Ball

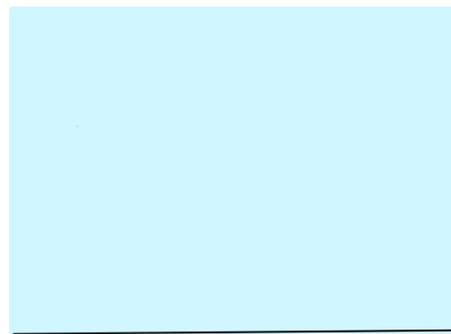


Programm für Schläger

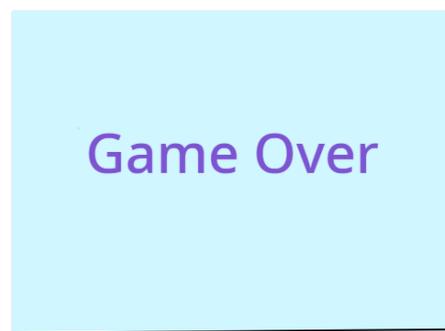
```
Wenn Taste Leertaste gedrückt wird
  wechsele zu Bühnenbild Blue Sky 2
  gehe zu x: 0 y: 100
  warte 1 Sekunden
  setze Punkte auf 0
  wiederhole fortlaufend
    gehe 10 er Schritt
    pralle vom Rand ab
    falls Farbe berührt ? , dann
      drehe dich um 180 Grad
      ändere Punkte um 1
      spiele Klang pop
    falls Punkte > 10 , dann
      gehe 13 er Schritt
    falls Punkte > 15 , dann
      gehe 16 er Schritt
    falls Farbe berührt ? , dann
      wechsele zu Bühnenbild Blue Sky 2 2
      spiele Klang basketball bounce
      stoppe alles
```

```
Wenn Taste Leertaste gedrückt wird
  wiederhole fortlaufend
    gehe zu Mauszeiger
    setze y auf -160
```

Bühne Blue Sky 2



Bühne Blue Sky 2 2



## 10.3 Katzenrennen



Programm für Katze 2 (bei 25%)



```
Wenn Taste s gedrückt wird
  setze Richtung auf -90 Grad
  gehe zu x: -35 y: -113
  wiederhole fortlaufend
    gehe 1 er Schritt
    falls wird Farbe orange berührt? , dann
      sage Game Over! für 2 Sekunden
      stoppe dieses Skript
    falls wird Farbe yellow berührt? , dann
      spiele Klang meow2 ganz
      sage Gewonnen für 2 Sekunden
      stoppe dieses Skript
  
```

```
Wenn Taste Pfeil nach links gedrückt wird
  drehe dich um 15 Grad

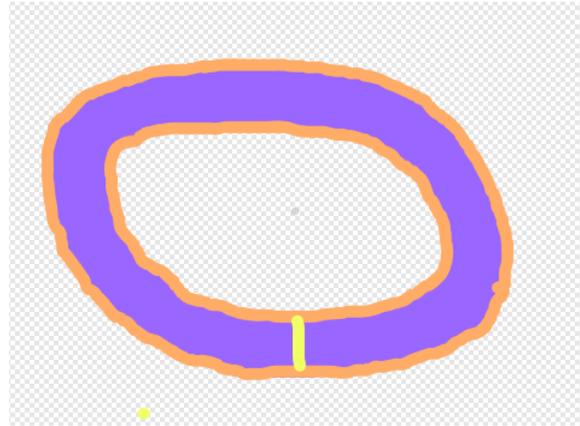
```

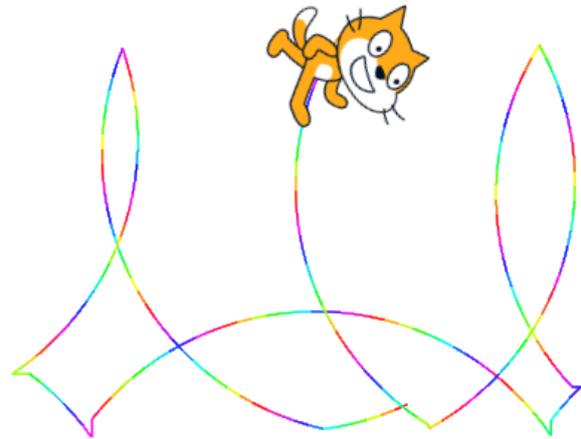
```
Wenn Taste Pfeil nach rechts gedrückt wird
  drehe dich um 15 Grad

```

Bühne

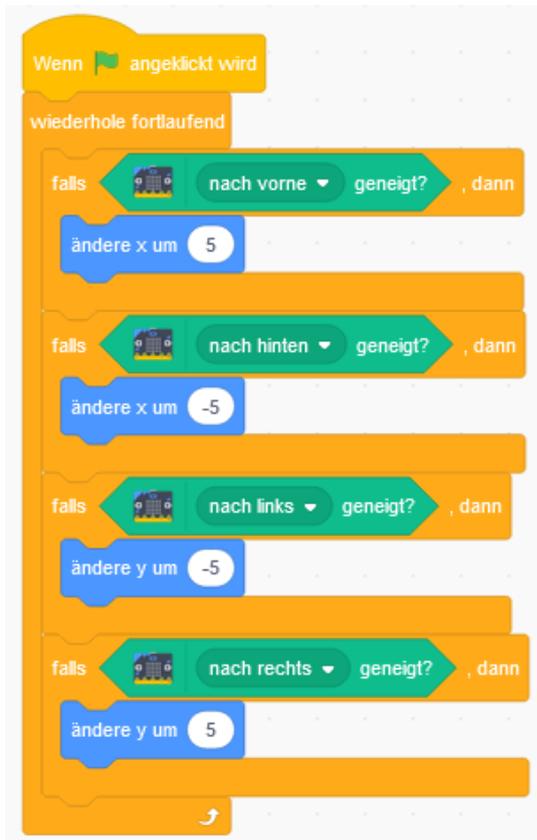


### 10.4 Zeichnen von bunten Linien



Mit dem Start wird eine bunte Linie gezeichnet, die am Rand abprallt.

### 10.5 Objektsteuerung mit micro:bit



### 10.6 Musik mit Gestik Steuerung

```

    Wenn  angeklickt wird
    schalte Video an
    setze Video-Transparenz auf 80
    
```

Für die Bühne wird der Programmcode zum Starten der Videokamera gesetzt.

Für die 3 Musikobjekte, wird der jeweilige Code wie abgebildet geschrieben. Nach dem Start geht es in die Endlosschleife. Dann wird die Kamera auf Bewegung des Objektes getestet und das entsprechende Klangbild abgespielt.



```

    Wenn  angeklickt wird
    wiederhole fortlaufend
    falls  Video- Bewegung von Figur > 50 , dann
    spiele Klang High Tom ganz
    
```

```

    Wenn  angeklickt wird
    wiederhole fortlaufend
    falls  Video- Bewegung von Figur > 50 , dann
    spiele Klang tap snare ganz
    
```

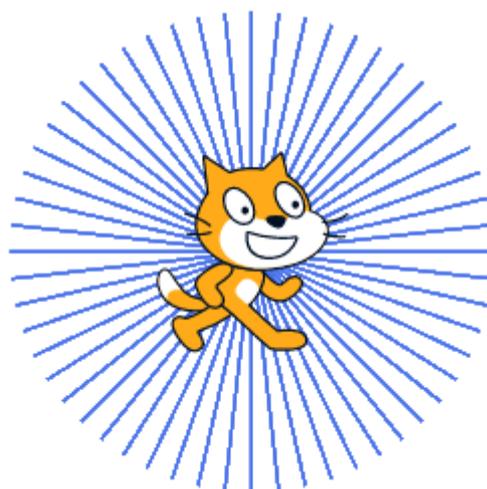
```

    Wenn  angeklickt wird
    wiederhole fortlaufend
    falls  Video- Bewegung von Figur > 50 , dann
    spiele Klang crash cymbal ganz
    
```

### 10.7 Linien Zeichnen

```

    Wenn Taste  Leeraste gedrückt wird
    lösche alles
    schalte Stift ein
    setze Stifffarbe auf 
    frage  und warte
    wiederhole Antwort mal
    drehe dich um  /  Grad
    gehe  er Schritt
    gehe zu x:  y: 
    
```



## 10.8 Zahlen raten

The image shows a Scratch script for a number guessing game. The script starts with a 'Wenn angeklickt wird' (When clicked) event. It then sets a 'Zufallszahl' (random number) from 1 to 10, sets 'versuch übrig' (attempts left) to 9, and sets 'erraten' (guessed) to 0. A speech bubble says 'Rate eine Zahl zwischen 0 und 100'. A 'wiederhole bis' (repeat until) loop runs while 'erraten' is not equal to 1. Inside the loop, it asks 'Wie lautet Dein Tipp' (What is your guess) and waits. It then checks if the 'Antwort' (answer) is equal to the 'Zufallszahl'. If yes, it sets 'erraten' to 1. If no, it checks if the answer is greater than the random number. If yes, it says 'kleiner' (smaller) for 2 seconds. If no, it says 'größer' (larger) for 2 seconds. Finally, it changes 'versuch übrig' by -1. After the loop, it says 'erraten'.

The visual interface on the right shows a variable 'erraten' set to 0. A speech bubble from the Scratch cat says 'Wie lautet Dein Tipp'. Below it is an input field with a checkmark button. Another Scratch cat says 'erraten'.

## 10.9 Kultspiele im Netz

[https://de.scratch-wiki.info/wiki/Kultspiele\\_in\\_Scratch](https://de.scratch-wiki.info/wiki/Kultspiele_in_Scratch)

