

- Skript zum Scratch Workshop -

April 2010

## Wie Bilder laufen lernen

Tomma Radtke und Sönke Schulmeister



Institut für Informatik

Christian-Albrechts-Universität zu Kiel



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Zum Experimentieren einladen . . . . .	9
1.2	Realisierung unterschiedlicher Projekte . . . . .	11
1.3	Verwirklichung komplexer Projekte . . . . .	12
1.4	Austausch mit der Scratch-Community und Teamarbeit . . . . .	13
<b>2</b>	<b>Making of „Mondlandung mit Folgen“</b>	<b>15</b>
2.1	Ideenfindung und Storyboarderstellung . . . . .	15
2.2	Tabellarische Szenenübersicht . . . . .	16
2.3	Herstellen der Filmmaterialien . . . . .	22
2.4	Erstellung des Kurzfilms mit Scratch . . . . .	22
2.4.1	Akteur: Bühne . . . . .	22
2.4.2	Akteur: Astronaut . . . . .	24
2.4.3	Akteur: Shuttle . . . . .	24
2.4.4	Akteur: Abspann . . . . .	26
2.5	Veröffentlichung . . . . .	27
<b>3</b>	<b>Selbststudium mit dem Katzenprogramm</b>	<b>29</b>
3.1	Beispielprogramm 1 . . . . .	29
3.1.1	Die Katze geht . . . . .	29
3.1.2	grüne Flagge . . . . .	30
3.1.3	x-Position der Katze ändern . . . . .	31
3.2	Aufgabe 1 . . . . .	32
3.3	Beispielprogramm 2 . . . . .	32
3.3.1	Endlosschleife . . . . .	32
3.3.2	Endlosschleife mit grüner Flagge . . . . .	33
3.3.3	Abprallen - Bewegungsänderung am Rand . . . . .	34
3.4	Aufgabe 2 . . . . .	35
3.5	Beispielprogramm 3 . . . . .	35
3.5.1	Warten - langsamere Bewegungen der Katze . . . . .	35
3.5.2	Kostüme - Bewegungsbilder . . . . .	36
3.6	Aufgabe 3 . . . . .	37
3.7	Beispielprogramm 4 . . . . .	37

3.7.1	Farbeffekt . . . . .	37
3.8	Aufgabe 4 . . . . .	38
3.9	Beispielprogramm 5 . . . . .	38
3.9.1	Fallunterscheidung - Wenn-Dann-Abfrage . . . . .	38
3.10	Aufgabe 5 . . . . .	41
3.11	Beispielprogramm 6 . . . . .	41
3.11.1	Neues Sprite erstellen . . . . .	41
3.11.2	Senden einer Nachricht . . . . .	42
3.11.3	Empfangen einer Nachricht . . . . .	43
3.11.4	Sprechtexterweiterung . . . . .	44
3.12	Aufgabe 6 . . . . .	44
3.13	Lösungen . . . . .	45
3.13.1	Lösung zu Aufgabe 1 . . . . .	45
3.13.2	Lösung zu Aufgabe 2 . . . . .	45
3.13.3	Lösung zu Aufgabe 3 . . . . .	45
3.13.4	Lösung zu Aufgabe 4 . . . . .	46
3.13.5	Lösung zu Aufgabe 5 . . . . .	46
3.13.6	Lösung zu Aufgabe 6 . . . . .	47
<b>4</b>	<b>Handbuch</b>	<b>49</b>
4.1	Einführung . . . . .	49
4.1.1	Grundlagen eines Scratch Projekts . . . . .	49
4.2	Das Scratch Programmfenster . . . . .	50
4.2.1	Die Bühne . . . . .	50
4.2.2	Neue Sprites . . . . .	51
4.2.3	Spriteliste . . . . .	52
4.2.4	Blockpalette und Programmebereich . . . . .	52
4.2.5	Kostüme . . . . .	54
4.2.6	Klänge . . . . .	55
4.2.7	Informationen zum aktuellen Sprite . . . . .	55
4.2.8	Rotationsmodus . . . . .	55
4.2.9	Symbolleiste . . . . .	56
4.2.10	Menü . . . . .	56
4.2.11	Grüne Flagge . . . . .	57
4.2.12	Paint Editor . . . . .	57
4.3	Scratchblöcke . . . . .	60
4.3.1	Blockarten . . . . .	60
4.3.2	Listen . . . . .	61
4.3.3	Zeichenketten . . . . .	61
4.3.4	Tastatureingabe . . . . .	62
4.3.5	Bewegungen . . . . .	62
4.3.6	Motorblöcke . . . . .	63
4.3.7	Aussehen . . . . .	63

4.3.8 Klänge . . . . .	64
4.3.9 Malstift . . . . .	65
4.3.10 Steuerung . . . . .	66
4.3.11 Fühlen . . . . .	67
4.3.12 Operatoren . . . . .	68
4.3.13 Variablen . . . . .	68
4.4 Anhang . . . . .	69
4.4.1 SYSTEM-ANFORDERUNGEN FÜR DIE SCRATCH-VERSION 1.4	69
4.4.2 NETZWERK INSTALLATION . . . . .	70
<b>5 Hilfen</b>	<b>71</b>
5.1 Hilfsmaterialien und Linksammlung . . . . .	71
5.1.1 Uniwebseiten und E-Mailadressen . . . . .	71
5.1.2 Scratchwebseite . . . . .	71
5.1.3 Bilder und Soundeffekte . . . . .	71
5.1.4 Bildbearbeitungsprogramm . . . . .	72
5.1.5 Audibearbeitungsprogramm . . . . .	72
5.1.6 Videotutorials . . . . .	72



# Kapitel 1

## Einleitung

Ungefähr 60 Prozent der Kinder zwischen sechs und 13 Jahren spielen mehrmals in der Woche oder jeden Tag Computerspiele. 50 Prozent surfen regelmäßig im Internet. Viele nutzen den Computer für das Schreiben von Texten und E-Mails, für das Chatten und für Musik hören. Das belegen Zahlen der letzten statistischen Untersuchung des Medienpädagogischen Forschungsverbunds Südwest (mpfs) zur Mediennutzung 6- bis 13-Jähriger in Deutschland – die KIM-Studie 2008.

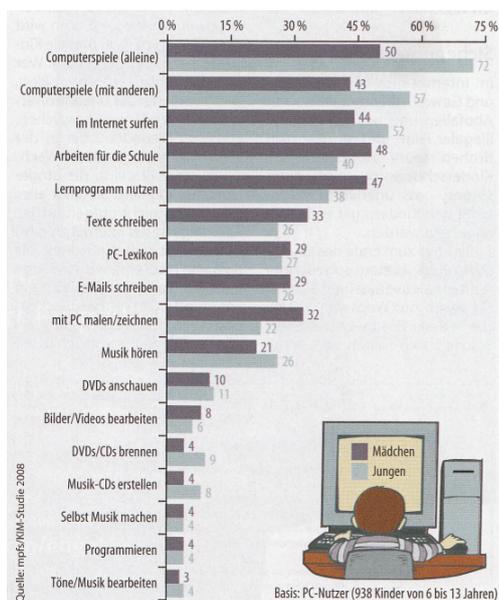


Abbildung 1.1: Computernutzung in der Gruppe der 6- bis 13-Jährigen

Die Zahlen machen deutlich, dass die Nutzung des Computers selbstverständlich für Kinder und Jugendliche geworden ist. Vor diesem Hintergrund prägte u.a. Marc Prensky für Jugendliche die Bezeichnung „digital natives“<sup>1</sup>.

<sup>1</sup>Marc Prensky: *Digital Natives, Digital Immigrants*, On the Horizon (MCB University Press, October 2001, Vol. 9, No. 5)

Interessant an der Studie ist nun aber auch die Betrachtung der Bereiche, die kaum Beachtung bei den Jugendlichen finden. Hierzu zählen:

- Bilder/Videos bearbeiten,
- selbst Musik machen,
- Töne/Musik bearbeiten und
- Programmieren.

Nur ungefähr fünf von 100 Jugendlichen nutzen den Computer in diesen Bereichen. Mitchel Resnick, der Entwickler von Scratch, schrieb in diesem Zusammenhang: „Although young people interact with digital media all of the time, few of them can create their own games, animations, or simulations. It's as if they can „read“ but not „write“.“<sup>2</sup>

Auch im Unterricht der Primar- und Sekundarstufe I spiegelt sich dieser Zusammenhang wider. Der Computer im Unterricht wird im wesentlichen verwendet

- für das Schreiben von Texten,
- für Präsentationen,
- für Informationsrecherchen,

aber kaum für kreative Tätigkeiten, wie

- die Erstellung und Bearbeitung von Musik und Videos,
- die Erstellung eigener Animationen,
- die Programmierung von eigenen Spielen.

Dabei besteht gerade für die letztgenannten Tätigkeiten ein großes Interesse bei den Kindern, wie jüngst eine empirische Untersuchung in zwei 6. Klassen der Gemeinschaftsschule Friedrichsort zeigt.

Es stellt sich also die Frage, warum der Computer im Unterricht weniger für kreative Tätigkeiten eingesetzt wird. Ein Grund ist sicherlich, dass viele Computerprogramme, die für die Erstellung eigener multimedialer Produkte benötigt werden, zu kompliziert in ihrer Benutzung sind. Dies gilt insbesondere für viele Programmierwerkzeuge. Zwar sind in den letzten Jahren Entwicklungsumgebungen entstanden, die einen spielerischen Ansatz verfolgen, z.B. „Kara“, „Karol“, „Java-Hamster“, aber auch diese Werkzeuge weisen zum Teil eine zu komplizierte Oberfläche und Spach-Syntax auf (Vgl. Abb. 1.2).

Vor diesem Hintergrund entwickelte Mitchel Resnik 2007, Leiter der Lifelong Kindergar-

---

<sup>2</sup>Mitchel Resnick: *Scratch: Programming for All*, Communications of the ACM, November 2009, Vol. 52, No. 11, Pages 60-67

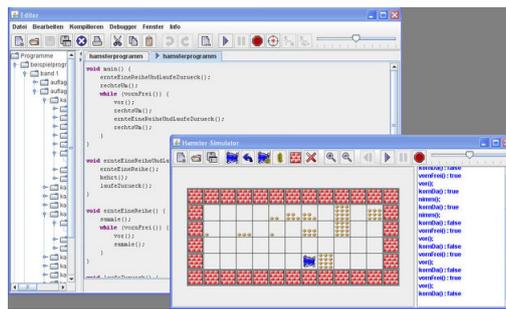


Abbildung 1.2: Programmoberfläche „Java Hamster“



Abbildung 1.3: Mitchel Resnick

ten Group am MIT Media Lab<sup>3</sup>, das Programmierwerkzeug „Scratch“. In den USA findet die Anwendung von Scratch in Computer Club Houses (Treffpunkte für (meist sozial benachteiligte) Jugendliche) schon großes Interesse, da Kinder hier spielerisch das Programmieren erlernen können. Moshe Vardi, ein berühmter Informatik-Professor in den USA, schrieb: „A couple of days ago, a colleague of mine told me how she tried to get her 10-year-old daughter interested in programming, and the only thing that appealed to her daughter (hugely) was Scratch.“

Bei der Entwicklung von des Programmierwerkzeugs Scratch verfolgte man folgende Philosophien:

1. Es soll einfach zu bedienen sein und zum Experimentieren einladen.
2. Es soll die Realisierung ganz unterschiedlicher Projekte ermöglichen (z.B. Animationen, virtuelle Touren, Spiele).
3. Es soll auch die Verwirklichung komplexer Projekte ermöglichen.
4. Es soll den Austausch mit der „Scratch-Community“ und die Teamarbeit fördern.

## 1.1 Zum Experimentieren einladen

Die Oberfläche von Scratch

<sup>3</sup><http://llk.media.mit.edu/>

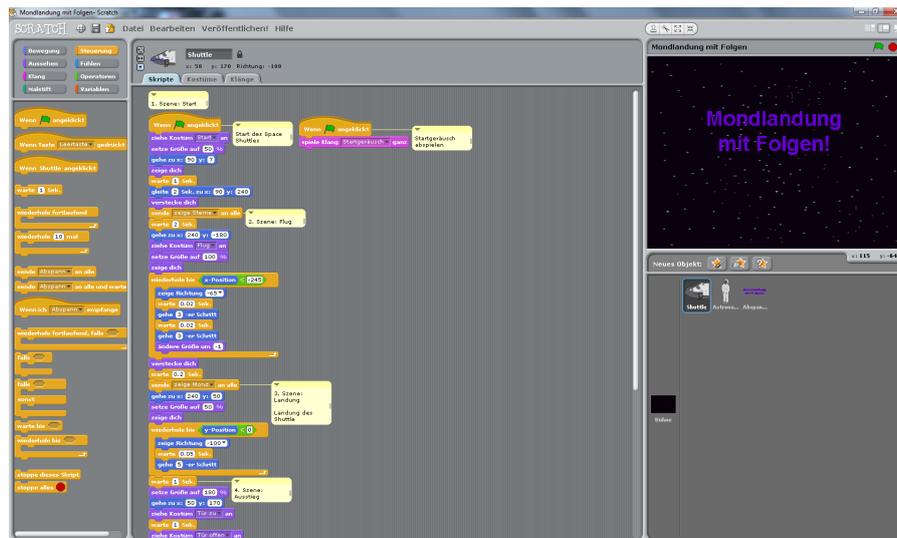


Abbildung 1.4: Benutzeroberfläche von Scratch

ist übersichtlich gestaltet. Auf der linken Seite der Benutzeroberfläche befindet sich eine Blockpalette mit verschiedenen Anweisungs- und Kontrollstrukturen. Diese Anweisungen werden in verschiedene Bereiche, die durch unterschiedliche Farben gekennzeichnet sind, eingeteilt. Beispielsweise ist der Bereich der Bewegung lila farbig, der Bereich der Steuerung gelb und der Bereich der Klänge rosa. Im mittleren Teil der Oberfläche befindet sich der Skriptbereich. Wie mit LEGO-Bausteinen werden verschiedene farbige Bausteine im Skripteditor zu einem Programm zusammengefügt:



Abbildung 1.5: Programmierbausteine von Scratch

Unterschiedliche Kombinationen können so mit direktem visuellen Feedback ausprobiert werden – ohne „Syntax-Qualen“. Mit diesen Skripten steuern die Kinder einzelne oder mehrere Objekte (Sprites) auf einer selbst erstellten Bühne (Stage). Diese befindet sich im rechten Teil der Oberfläche.

Wie Hip-Hop-DJs können SchülerInnen Programmierbausteine, Grafiken (selbst gezeichnete Bilder, Fotos) und Musik mixen oder scratchen. Hieraus resultiert der Name des Pro-

grammierwerkzeugs: „Scratch“.

## 1.2 Realisierung unterschiedlicher Projekte

Scratch ermöglicht auf einfache Art und Weise die Erstellung ganz unterschiedlicher Projekte:

- Animationen (z.B. Day Dream<sup>4</sup>)
- Stop-Motion-Filme (z.B. Stop motion<sup>5</sup>)
- Virtuelle Touren (z.B. Virtuelle Tour durch ein Gebäude<sup>6</sup>)
- Spiele (z.B. Tetris<sup>7</sup>)



Abbildung 1.6: Bildausschnitt aus der Animation „Day Dream“



Abbildung 1.7: Bildausschnitt aus dem Film „Stop motion“

Somit ist es möglich, ganz unterschiedliche Interessen von SchülerInnen zu berücksichtigen.

<sup>4</sup><http://scratch.mit.edu/projects/creme/40150>

<sup>5</sup><http://scratch.mit.edu/projects/3X3/739385>

<sup>6</sup><http://scratch.mit.edu/projects/bmarcell/1882>

<sup>7</sup><http://scratch.mit.edu/projects/amyv/11292>



Abbildung 1.8: Bildausschnitt aus der „ELTE Virtual Tour“

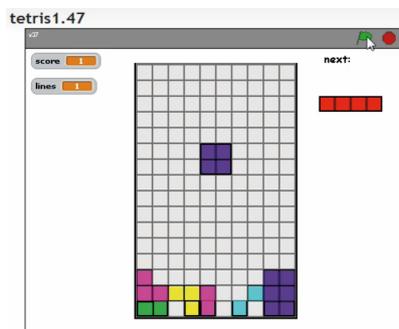


Abbildung 1.9: Bildausschnitt aus dem Spiel „Tetris“

### 1.3 Verwirklichung komplexer Projekte

Mit wachsender Kenntnis und Routine lassen sich auch komplexere Projekte umsetzen. Stellvertretend seien hier ein Schachspiel<sup>8</sup>



Abbildung 1.10: Bildausschnitt aus dem Spiel „Chess“

und ein Projekt zur visuellen Kryptographie<sup>9</sup> genannt.

<sup>8</sup><http://scratch.mit.edu/projects/qwertyupo/704041>

<sup>9</sup><http://scratch.mit.edu/projects/simplizialkomplex/831536>

## 1.4 Austausch mit der Scratch-Community und Teamarbeit

Zusätzlich zum Entwicklungswerkzeug Scratch gibt es auch eine Scratch-Webseite<sup>10</sup>.

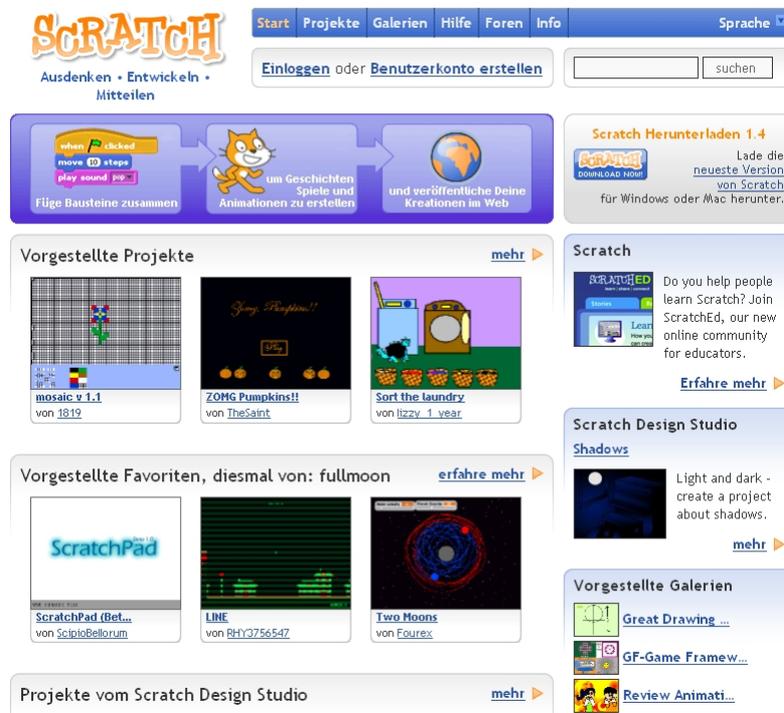


Abbildung 1.11: Scratch-Webseite

Sie ist eng verzahnt mit Scratch. Jedes Scratch-Projekt kann, wenn man einen eigenen kostenlosen Scratch-Account angelegt hat, von der Programmierumgebung aus mit einem Klick auf die Webseite veröffentlicht werden. Auf diese Weise kann jeder das Projekt betrachten, Kommentare und Anregungen hinterlassen. Hierfür ist nicht die Entwicklungsumgebung notwendig. Es reicht ein Browser. Anfang 2010 gab es bereits 863082 Projekte und 102306 Programmierer. Scratch wird deshalb gerne als das YouTube für interaktive Medien bezeichnet. Interessant ist, dass auch der umgekehrte Weg funktioniert. Es ist also möglich jedes veröffentlichte Projekt mit allen zum Projekt dazugehörigen Ressourcen (Skripte, Grafiken, Musik) in die Scratch-Entwicklungsumgebung herunterzuladen und dort nach eigenen Vorstellungen zu verändern oder zu erweitern. Der Open-Source Gedanke, die Weiterentwicklung bereits bestehender Projekte, wird hierbei aufgegriffen. Auch im Unterricht kann der Gedanke der Teamarbeit gut umgesetzt werden. Gruppen können sich arbeitsteilig um die Bereiche Ideen- und Storyboardentwicklung, Musik- und Soundbearbeitung, Grafikgestaltung und Programmierung kümmern. Deutlich wird hier der fächerverbindende und fachübergreifende Charakter von Scratch-Projekten, der im Unterricht die Chance bietet, ausgewählte Themen aus verschiedenen Fächer-Sichtweisen zu betrachten.

<sup>10</sup><http://scratch.mit.edu/>



# Kapitel 2

## Making of „Mondlandung mit Folgen“

### 2.1 Ideenfindung und Storyboarderstellung

Ziel war die Herstellung eines Kurzfilms, der eine witzige und überraschende Pointe bietet. Zudem sollte er nicht länger als eine Minute dauern. Da es bereits eine erste Geschichte gab, in der eine Mondlandung thematisiert war, wurde diese Idee aufgegriffen und weiterentwickelt. Hier erwies sich die Videoplattform youtube.de als geeignete Inspirationsplattform. Am Ende des Ideenfindungsprozesses stand eine Geschichte, in der ein Weltraumspaziergang eines Astronauten gezeigt werden sollte, der anders als geplant endet: Nach der Landung des Raumschiffs und dem Ausstieg des Astronauten fällt die Tür zur Astronautenkapsel unvermittelt zu. Natürlich hat der Astronaut den Haustürschlüssel vergessen, sodass er sich nun ausgeschlossen auf dem Mond befindet.

Nachdem die grundsätzliche Filmidee feststand, haben wir im anschließenden Schritt ein kleines Storyboard entwickelt, wo wir insbesondere die Kameraeinstellungen und die einzelnen Bildschnitte festgelegt haben:

- Szene 1 (Start): Das Raumschiff wird beim Startvorgang gezeigt. Musik wird eingeblendet.
- Szene 2 (Flug): Das Raumschiff fliegt zur Mondoberfläche. Das sich Nähern des Raumschiffs wird durch einen kleineren visuellen Effekt (Verkleinerung des Raumschiffs) dargestellt. Weiterhin wird die Animation durch eine entsprechende Soundkulisse untermalt.
- Szene 3 (Ausstieg): Es wird der Weltraumausstieg des Astronauten gezeigt. In der Frontalansicht wird nun gezeigt, wie sich der Astronaut von seiner Kapsel zur Mondoberfläche bewegt. Hinter seinem Rücken schließt sich Tür.
- Szene 4: Der Astronaut erschreckt sich beim Zuschlagen der Tür. Panisch überprüft er, ob er einen Türschlüssel in seinen Raumanzug eingesteckt hat—was nicht der Fall ist.

## 2.2 Tabellarische Szenenübersicht

Die folgenden Tabellen zeigen für die einzelnen Szenen die benötigten Akteure mit zugehörigen Bewegungsbildern (genannt Kostüme) und Klängen und die Regieanweisungen, welche die Akteure zu befolgen haben.

Die technischen Regieanweisungen sind so detailliert angegeben, dass sie leicht in Programmcodes übersetzt werden können.

**1. Szene: Shuttlestart**

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Bühne	Start1 Start2 Start3 Start4	Score	-Score abspielen -Hintergrundwechsel	-Score abspielen -Hintergrundwechsel von <i>Start1</i> bis <i>Start4</i>
Astronaut				-verstecken
Shuttle	Start	Startgeräusch	-Startgeräusch abspielen -von unten nach oben fliegen	-Startgeräusch abspielen -Kostümwechsel -Größe auf 50% setzen -zur Starposition {90;7} gehen -sich zeigen -für 2 Sekunden zur Position {90;240} gleiten -verstecken -Szenenwechsel senden
Abspann				-verstecken

**2. Szene: Flug**

Akteure	Kostüme	Klänge (Score)	Regieanweisung	technische Regieanweisung
Bühne	Sterne		Hintergrundwechsel	(-Score abspielen) -Hintergrundwechsel
Astronaut				(-verstecken)
Shuttle	Flug		-von rechts nach links fliegen -kleiner werden -Szenenwechsel senden	-zur Position {240;-180} gehen -Kostümwechsel -Größe auf 100% setzen -sich zeigen -von rechts nach links gehen, in 3-er Schritten, Richtung -65 -um 1 verkleinern, nach jedem zweiten Schritteblock -verstecken -Szenenwechsel senden
Abspann				(-verstecken)

**3. Szene: Landung**

Akteure	Kostüme	Klänge (Score)	Regieanweisung	technische Regieanweisung
Bühne	Mond		-Hintergrundwechsel	(-Score abspielen) -Hintergrundwechsel
Astronaut				(-verstecken)
Shuttle	Flug		von rechts zur Mitte fliegen	-zur Position {240;50} gehen -Größe auf 50% setzen -sich zeigen -von rechts bis nach links gehen, bis die y-Position < 0 ist, in 5-er Schritten, Richtung -100
Abspann				(-verstecken)

## 4. Szene: Ausstieg

Akteure	Kostüme (Mond)	Klänge (Score)	Regieanweisung	technische Regieanweisung (-Score abspielen)
Bühne Astronaut	stehend gehend drehend linke Hand rechte Hand kopfgreifend	abtasten	-aussteigen -umdrehen -die linke und rechte Tasche kontrollieren -an den Kopf greifen -Funkspruch senden	-Kostümwechsel zu <i>stehend</i> -zur Position {158;44} gehen -nach vorne kommen -sich zeigen -3 mal: Kostümwechsel zu <i>gehend</i> für 0.7 Sekunden zur Position {140;20}, ({128;-11} und {83;-43}) gleiten -Kostümwechsel zu <i>stehend</i> -Nachricht senden -nach Empfang von Nachricht: Kostümwechsel zu <i>drehend, stehend</i> und <i>linke Hand</i> - <i>abtasten</i> abspielen -Kostümwechsel zu <i>stehend</i> und <i>rechte Hand</i> - <i>abtasten</i> abspielen -Kostümwechsel zu <i>stehend</i> und <i>kopfgreifend</i> - <i>kopfklatzen</i> abspielen - <i>Houston wir haben ein Problem</i> abspielen -Kostümwechsel zu <i>stehend</i> -Nachricht senden -verstecken

Shuttle	Tür zu Tür offen Tür offen mit Treppe Tür zu mit Treppe	Tuerzu	-Tür öffnen -Treppe ausklappen -Nachricht an den Astronaut senden -Tür zumachen -score beenden -Nachricht an den Astronaut senden	-Größe auf 180% setzen -zur Position {50;170} gehen -Kostümwechsel von <i>Tür zu über Tür offen zu Tür offen mit Treppe</i> -Nachricht an den Astronaut senden -nach Empfang von Nachricht: Kostümwechsel zu <i>Tür zu mit Treppe</i> - <i>Tuerzu</i> abspielen -alle Klänge stoppen -Nachricht senden (-verstecken)
Abspann				

### 5. Szene: Abspann

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Bühne	Sterne		-Hintergrundwechsel	-Hintergrundwechsel
Astronaut				-verstecken
Shuttle				-verstecken
Abspann	Abspann1	Abspann	-von unten nach oben scrollen - <i>Abspann</i> abspielen -Farbe ändern	zur Position {0;180} gehen -sich zeigen -für 15 Sekunden zur Position {0;180} gleiten -wiederholt den Farbeffekt um 1 ändern - <i>Abspann</i> abspielen -alle Skripte stoppen

## 2.3 Herstellen der Filmmaterialien

Im nächsten Schritt haben wir für alle Akteure des Films (Bühne, Raumschiff, Astronaut) entsprechende Kostüme angefertigt. Für das Raumschiff und den Astronauten wurden mehrere Kostüme gezeichnet, um durch das An- und Ausziehen eine Bewegung darzustellen. Alle Zeichnungen wurden eingescannt, sodass sie als Grafikdateien in der Scratch-Entwicklungsumgebung genutzt werden konnten. Für die Bühnen-Kostüme haben wir auf Fotos zurückgegriffen. Unter flickr.com kann man nach Fotos suchen lassen, die frei für Projekte genutzt werden können (siehe Kapitel Hilfe). Für die Musikuntermalung haben wir auf ein Stück aus der Musiksammlung von Michael Donner<sup>1</sup> zurückgegriffen. Einige Musik-DVDs mit unterschiedlichen Geräuschen und Melodien stehen in Lizenz für die Scratch-Challenge zur Verfügung. Alternativ besteht aber in Scratch auch die Möglichkeit, eigene Musikstücke zu komponieren. Viele unterschiedliche Instrumente stehen hierfür zur Verfügung.

## 2.4 Erstellung des Kurzfilms mit Scratch

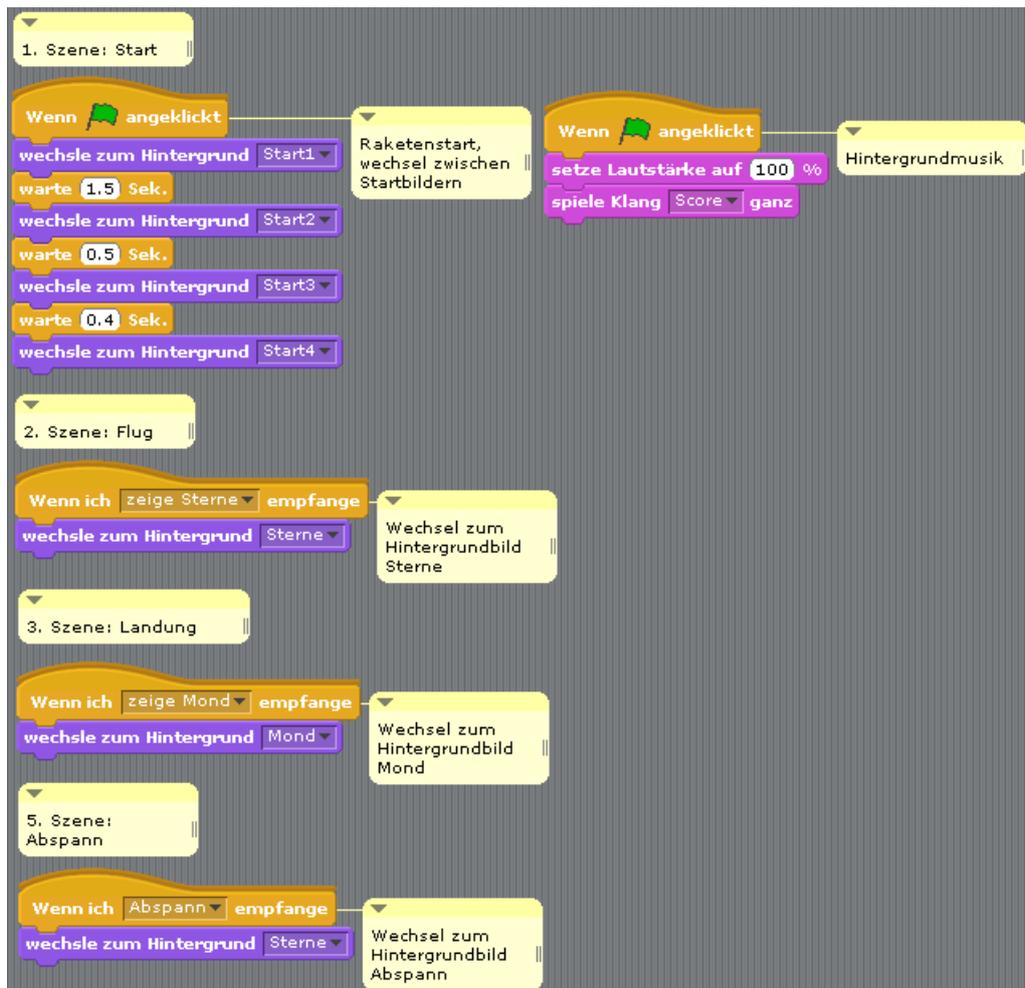
Im dritten Schritt haben wir für alle Akteure entsprechende Regieanweisungen geschrieben. Diese werden in Form von Skripten den einzelnen Akteuren zugeordnet:

### 2.4.1 Akteur: Bühne

Die Regieanweisungen aus der Tabelle für die Bühne werden wie folgt in die Programmcodes umgesetzt.

---

<sup>1</sup><http://www.michael-donner-music.de/>



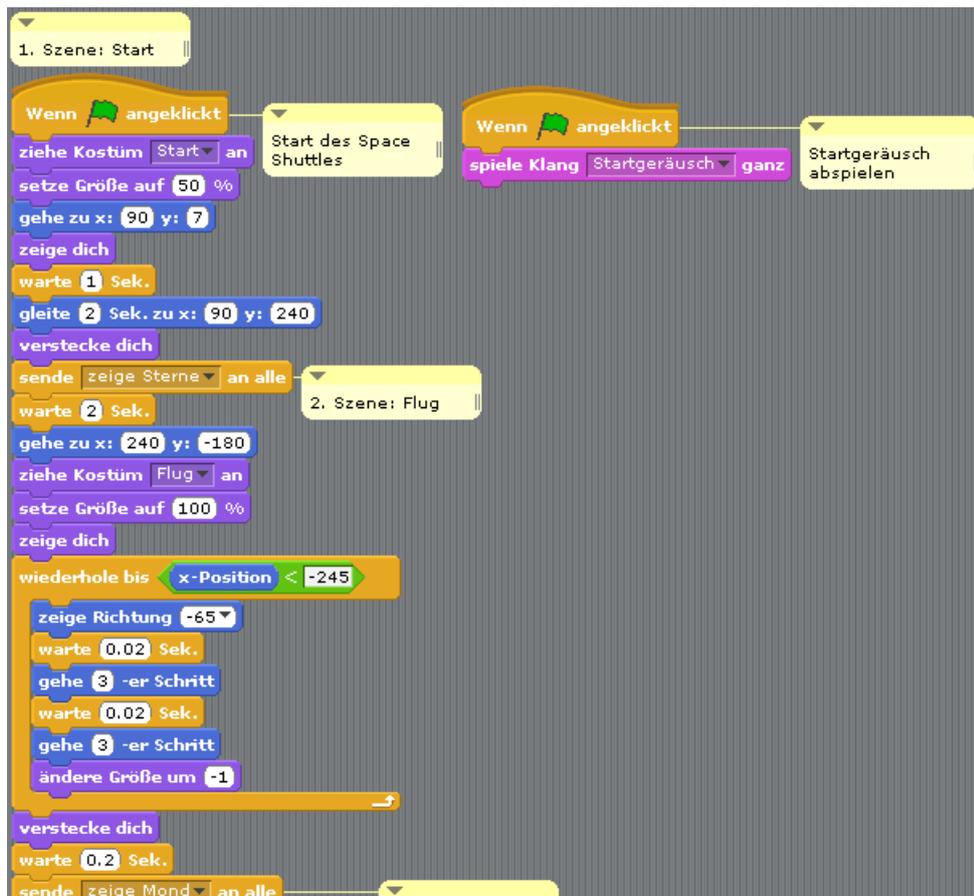
## 2.4.2 Akteur: Astronaut

Die Regieanweisungen aus der Tabelle für den Astronauten werden wie folgt umgesetzt.



## 2.4.3 Akteur: Shuttle

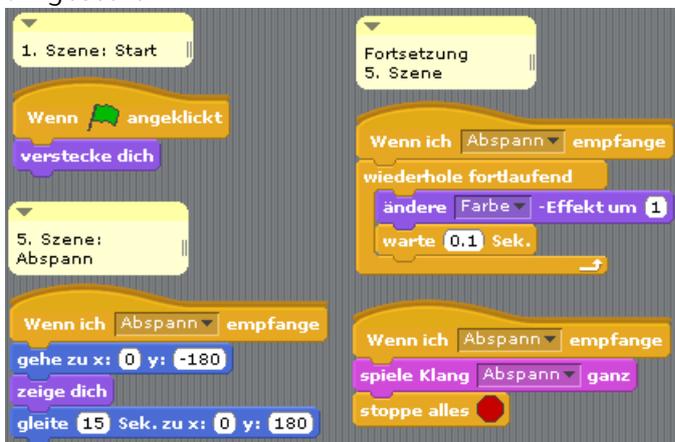
Die Regieanweisungen aus der Tabelle für das Shuttle werden in folgende Programmcodes umgesetzt.





## 2.4.4 Akteur: Abspann

Die Regieanweisungen aus der Tabelle für den Abspann werden in folgende Programmcodes umgesetzt:



## 2.5 Veröffentlichung

Der letzte Schritt sieht eine Veröffentlichung des Projekts vor. Per Mausklick ist es möglich, die Animation auf die Scratch-Webseite<sup>2</sup> hochzuladen. Auf diese Weise kann jeder das Projekt betrachten, Kommentare und Anregungen hinterlassen. Hierfür ist nicht die Entwicklungsumgebung notwendig. Es reicht ein Browser.

---

<sup>2</sup><http://scratch.mit.edu/>



# Kapitel 3

## Selbststudium mit dem Katzenprogramm

Ein einfaches Übungsprogramm mit genauen Anleitungen um mit den Möglichkeiten des Scratchprogramms vertraut zu werden.

Fachbegriffe sind fett gedruckt.

Kursiv gedruckte Anweisungen und Erklärungen erläutern detailliert die Oberfläche oder sie enthalten Zusatzinformationen.

Je vertrauter Sie mit dem Programm sind, desto weniger werden Sie die kursiven Texte brauchen.

### 3.1 Beispielprogramm 1

#### 3.1.1 Die Katze geht

Das **Sprite**, hier die Katze, soll animiert werden. Es soll einige Schritte gehen.

Die Tabellarische Szenenübersicht sieht also wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe ein Stück	-gehe 20 Schritte



Klicken Sie dafür in der **Palettenauswahl** auf die **Bewegungpalette** (dunkelblau). *Es werden nun dunkelblau unterlegte Palettblöcke*

angezeigt.

Ziehen Sie den Block  in den **Programmierbereich**. Klicken Sie den Block an und beobachten Sie was mit der Katze passiert.

*Die Katze geht nun 10 Schritte in die angegebene Richtung. Einzusehen ist dies in den **Informationen zum aktuellen Sprite**.*

Ändern Sie nun den Wert, der in diesem Block ist auf den Wert 20. *Sie können den Wert ändern, indem Sie in dem Block auf das weiße Textfeld klicken und den bestehenden Wert mittels der Tastatur löschen und den gewünschten Wert eingeben.*

Speichern Sie nun ihr Projekt, indem Sie das **Speichern**-Symbol  anklicken. Geben Sie ihrem Projekt einen Namen und klicken Sie dann auf speichern. *Es ist empfehlenswert von Zeit zu Zeit Ihr Projekt abzuspeichern.*

### 3.1.2 grüne Flagge

Die **grüne Flagge** (rechts oberhalb der Bühne) soll zum Starten des Programms benutzt werden.

Die Tabellarische Szenenübersicht sieht also wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe ein Stück	-grüne Flagge -gehe 20 Schritte



Sie brauchen den Block . Klicken Sie dafür in der **Palettenauswahl** auf die **Steuerungpalette** (gelb). *Es werden nun gelb unterlegte Palettenblöcke*



*angezeigt.* Ziehen Sie den Block  in den **Programmierbereich**. Setzen Sie ihn an den Anfang des Programms (oberhalb des Gehen-Blocks), indem Sie die beiden Blöcke zu einem **Stapel** zusammenfügen. *Es erscheint eine weiße Linie oberhalb des Bewegungsblocks um anzuzeigen, dass der Stapel hier fallen gelassen werden kann.*

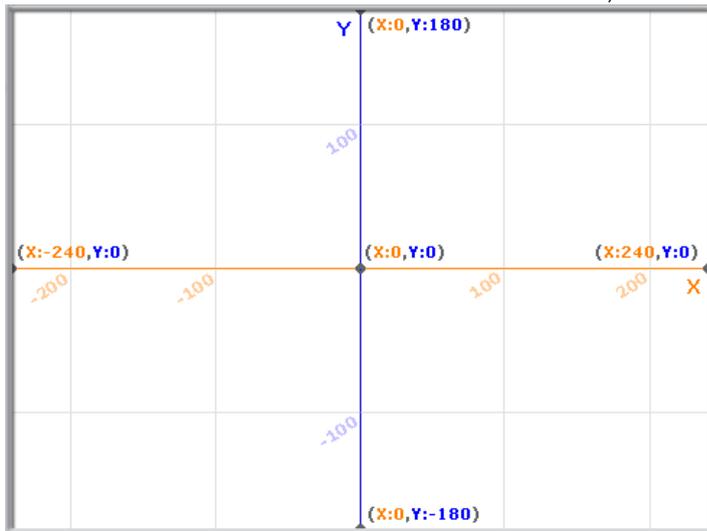


Klicken Sie nun auf die **grüne Flagge** (rechts oberhalb der Bühne), um die Blöcke unterhalb des Blocks mit der grünen Flagge zu starten und auf das **Stoppsymbol** um das Programm

zu beenden. Sie können weiterhin das Programm starten und beenden, indem Sie auf den auszuführenden/ausführenden Stapel klicken.

### 3.1.3 x-Position der Katze ändern

Nachdem die Katze einige Schritte gegangen ist, soll die Katze an eine andere Position gesetzt werden. Dazu sind die **x-Position** und die **y-Position** des **Sprites** zu ändern. Schon beim Gehen wurde die x-Position und/oder die y-Position der Katze geändert.



Die **x-Positionen** werden hier durch die orangefarbene Achse dargestellt, die **y-Positionen** durch die dunkelblaue Achse.

Die Tabellarische Szenenübersicht sieht also wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe ein Stück -sprige zu einer anderen Position	-grüne Flagge -gehe 20 Schritte -ändere die x-Position um 25

Klicken Sie in der **Palettenauswahl** auf die **Bewegungpalette** (dunkelblau) und ziehen



Sie den Block in den **Programmierbereich**. Ändern Sie den Wert des Blocks auf 25. Sie können den Wert ändern, indem Sie in dem Block auf das weiße Textfeld klicken und den bestehenden Wert mittels der Tastatur löschen und den gewünschten Wert eingeben.

Klicken Sie den Block an und beobachten Sie die Positionsangaben des Katzensprites, welche Sie in den **Informationen zum aktuellen Sprite** finden.

Dieser Block soll nun mit den anderen Blöcken verbunden werden. Dafür gibt es zwei Möglichkeiten: Die erste Möglichkeit ist es den Änderungs-Block zwischen den grüne-Flagge-Block und den Gehen-Block zu platzieren.

Die zweite Möglichkeit ist es den Änderungs-Block unter den Gehen-Block zu setzen. In diesem Fall soll die zweite Möglichkeit realisiert werden.

Wenn Sie die Blöcke zusammenfügen wollen erscheint wieder eine weiße Linie, je nachdem wo Sie den Änderungs-Block hinschieben oberhalb oder unterhalb des Gehen-Blocks, um anzuzeigen, dass der Stapel hier fallen gelassen werden kann. Lassen Sie den Block fallen, wenn die weiße Linie unterhalb des Gehen-Blocks erscheint.



Haben Sie versehentlich den Änderungs-Block oberhalb des Gehen-Blocks platziert, so können Sie diese wieder auseinanderziehen, indem Sie den unteren Block anklicken und vom anderen Block wegziehen. Wenn Sie auf den oberen Block klicken, so wird der gesamte Stapel unterhalb dieses Blocks bewegt.

## 3.2 Aufgabe 1

Erweitern Sie das Beispielprogramm 1, indem Sie die **y-Position** des Katzensprites auf den Wert -90 setzen.

Drehen Sie das **Sprite** um 22 Grad nach links und lassen Sie es dann 5 Sekunden zur Position {x:-150, y:100} gleiten.

## 3.3 Beispielprogramm 2

### 3.3.1 Endlosschleife

Die Katze soll nun über die Bühne laufen. Schreiben Sie hierfür ein neues Programm für das Katzensprite.

Die Tabellarische Szenenübersicht sieht also wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe über die Bühne	-wiederhole endlos: gehe 10 Schritte

Speichern Sie gegebenenfalls Ihr derzeitiges Programm ab und klicken Sie im **Menü** auf **Datei** und wählen Sie im Kontextmenü den Unterpunkt **Neu**.

Der Block  aus der **Bewegungspalette** (dunkelblau) soll also (endlos) wiederholt ausgeführt werden. Ziehen Sie den Block in den **Programmierbereich**. Sie brauchen

nun eine **Endlosschleife** . Klicken Sie in der **Palettenauswahl** auf die **Steuerungpalette** (gelb). *Die Palettenblöcke sind nun gelb unterlegt.*



Ziehen Sie den Block **(Endlosschleife)** in den **Programmierbereich**.

Nun sollen die zwei Blöcke miteinander verbunden werden. Dafür klicken Sie den Gehen-Block an und ziehen ihn in die **Endlosschleife**. *Es erscheint eine weiße Linie innerhalb des Blocks, die anzeigt, dass der Gehen-Block hier fallen gelassen werden kann.*

*Achtung: Es erscheint auch eine weiße Linie oberhalb der Endlosschleife. In diesem Fall ist es jedoch wenig sinnvoll den Block hier fallen zu lassen, da die Katze wiederholt 10 Schritte gehen soll. Haben Sie versehentlich den Gehen-Block oberhalb der Endlosschleife plziert, so können Sie diese wieder auseinanderziehen, indem Sie den unteren Block anklicken und vom anderen Block wegziehen. Wenn Sie auf den oberen Block klicken, so wird der gesamte Stapel unterhalb dieses Blocks bewegt.*

Es entsteht ein Stapel der Form:



Klicken Sie diesen Stapel an und beobachten Sie was passiert.

Um die Katze wieder an den linken Rand der Bühne zu bekommen, klicken Sie sie an und ziehen Sie sie an die gewünschte Position. *Wenn die Katze erneut an den rechten Bühnenrand geht, muss vorher auf den Stapel oder auf das **Stoppszeichen** (rechts oberhalb der Bühne) geklickt werden. Dass das Programm läuft erkennt man daran, dass die Blöcke die ausgeführt werden weiß umrandet sind.*

### 3.3.2 Endlosschleife mit grüner Flagge

Auch für dieses Programm soll die **grüne Flagge** zum starten des Programms benutzt werden.

Die Tabellarische Szenenübersicht sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe über die Bühne	-grüne Flagge -wiederhole endlos: 1. -gehe 10 Schritte

Klicken Sie in der **Palettenauswahl** auf die **Steuerungpalette** (gelb). Ziehen Sie nun den

Block  in den **Programmierbereich** und setzen ihn an den An-



fang des Programms: . Auch hierbei erscheint wieder eine weiße Linie um anzuzeigen, dass der Stapel hier fallen gelassen werden kann.

Klicken Sie nun auf die **grüne Flagge**, um die Blöcke unterhalb des Blocks mit der grünen Flagge zu starten und auf das **Stoppsymbol** um das Programm zu beenden. Sie können weiterhin das Programm starten und beenden, indem Sie auf den auszuführenden/ausführenden Stapel klicken.

Beobachten Sie was passiert, wenn Sie die grüne Flagge anklicken.

### 3.3.3 Abprallen - Bewegungsänderung am Rand

Das Katzensprite soll nun nicht mehr aus dem Bild laufen, wenn es die Begrenzung der Bühne erreicht, sondern vom Rand abprallen und weiterlaufen.

Die Tabellarische Szenenübersicht sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe über die Bühne -pralle am Rand ab	-grüne Flagge -wiederhole endlos: 1. -pralle vom Rand ab 2. -gehe 10 Schritte



Benutzen Sie hierfür den Befehl . Sie finden diesen Block in der **Bewegungpalette**. Setzen Sie den Block mit in die **Endlosschleife** ein: beachten Sie dabei, dass der Block vor den Gehen-Block gesetzt werden muss, da die Katze nicht in den Rand laufen soll:



Nun dreht sich die Katze auf den Kopf, wenn sie das erste Mal (und dann jedes zweite Mal) den Rand berührt. *Dieses liegt daran, dass der Befehl vom Rand abzuprallen der Katze eigentlich sagt, dass sie sich um 180 Grad drehen soll, sobald sie den Rand berührt.*

Soll die Katze sich nicht auf den Kopf drehen können, so kann man mit dem Rotationsknopf in den **Informationen zum aktuellen Sprite** einstellen, dass sich die Katze nur nach links oder rechts drehen kann:  (oder auch frei  oder gar nicht .

## 3.4 Aufgabe 2

Schreiben Sie ein neues Programm. Benutzen Sie hierfür nur Blöcke aus der **Bewegungspalette** und der **Steuerungpalette**:

Wenn die **grüne Flagge** angeklickt wird, lassen Sie die Katze an die Position {x:-70, y:50} gehen. Dann soll sich das Katzensprite genau 100 mal um 5 Grad nach rechts drehen.

## 3.5 Beispielprogramm 3

### 3.5.1 Warten - langsamere Bewegungen der Katze

Die Katze soll sich jetzt langsamer über den Bildschirm bewegen.

Erweitern Sie hierfür das Beispielprogramm 2.

Die Tabellarische Szenenübersicht sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-gehe langsam über die Bühne -pralle am Rand ab	-grüne Flagge -wiederhole endlos: 1. -pralle vom Rand ab 2. -gehe 10 Schritte 3. -warte 0,25 Sekunden



In der **Steuerungpalette** befindet sich der Block , platzieren Sie diesen unter den Gehen-Block in der **Endlosschleife** und ändern Sie den Wert auf 0.25. *Sie können den Wert ändern, indem Sie in dem Block auf das weiße Textfeld klicken und den bestehenden Wert mittels der Tastatur löschen und den gewünschten Wert eingeben.:*



Beobachten Sie was passiert, wenn Sie die **grüne Flagge** anklicken.

### 3.5.2 Kostüme - Bewegungsbilder

Damit es nun aussieht als würde sich die Katze bewegen, soll die Katze ihr Erscheinungsbild bei jedem Bewegungsablauf ändern. Dieses realisiert man in Scratch indem man der Katze ein neues **Kostüm** anzieht.

Die Tabellarische Szenenübersicht sieht wie folgt aus:

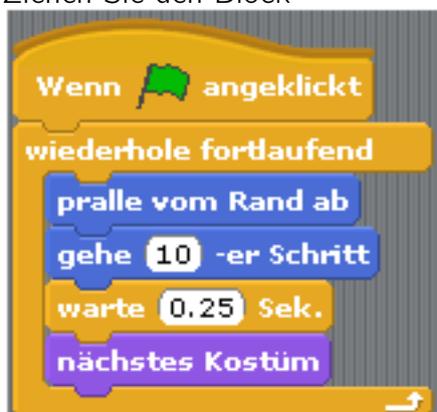
Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1 Kostüm2		-gehe langsam über die Bühne -pralle am Rand ab -Bewegungsbild	-grüne Flagge -wiederhole endlos: 1. -pralle vom Rand ab 2. -gehe 10 Schritte 3. -warte 0,25 Sekunden 4. -Kostümwechsel

Klicken Sie dafür in der **Palettenauswahl** auf die **Aussehenpalette** (lila). *Es werden nun lila unterlegte Palettenblöcke angezeigt.*

Ziehen Sie den Block

nächstes Kostüm

an das Ende der **Endlosschleife**:



## 3.6 Aufgabe 3

Klicken Sie auf den **Kostümereiter**



und kopieren Sie das Kostüm 1. Verändern Sie das neu entstandene Kostüm 3. Klicken Sie dafür auf den Button **Bearbeiten** und benutzen Sie das Malprogramm. Beobachten Sie was passiert, wenn Sie das Beispielprogramm 3 laufen lassen. Klicken Sie nun auf das **Vorschaubild** der Bühne und importieren Sie im **Kostümereiter** ein neues Hintergrundbild. Schreiben Sie für die Bühne ein Programm: Wenn die Taste b gedrückt wird, soll die Bühne die Hintergrundbilder alle 10 Sekunden wechseln.

## 3.7 Beispielprogramm 4

### 3.7.1 Farbeffekt

Mittels Grafikeffekten soll das Aussehen der Katze verändert werden. Erweitern Sie das Beispielprogramm 3. Beispielsweise lässt sich die Farbe der Katze ändern: Die Tabellarische Szenenübersicht sieht wie folgt aus:

Ziehen Sie dafür aus der **Aussehenpalette** (lila) den Block

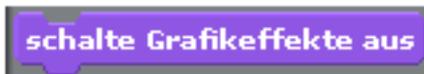


mit in die **Endlosschleife**: Ändern Sie den Wert in diesem Block (zum Beispiel in 3) und beobachten Sie was passiert, wenn Sie die grüne Flagge anklicken.

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1 Kostüm2		-gehe langsam über die Bühne -pralle am Rand ab -Bewegungsbild -Farbwechsel	-grüne Flagge -wiederhole endlos: 1. -pralle vom Rand ab 2. -gehe 10 Schritte 3. -warte 0,25 Sekunden 4. -Kostümwechsel 5. -ändere Farbeffekt um 3



Um alle Grafikeffekte auszuschalten, klicken Sie auf den Block



aus der **Aussehenpalette**.

## 3.8 Aufgabe 4

Schreiben Sie neben dem Beispielprogramm 4 ein weiteres Programm für das **Katzensprite**: Sobald die Katze mit der linken Maustaste angeklickt wird, soll sich das Aussehen der Katze verändern. Benutzen Sie, mit unterschiedlichen Werten, verschiedene Effekte aus der **Aussehenpalette**.

## 3.9 Beispielprogramm 5

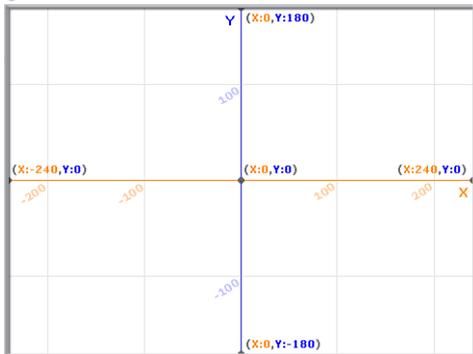
### 3.9.1 Fallunterscheidung - Wenn-Dann-Abfrage

In diesem Beispielprogramm soll eine **Fallunterscheidung** benutzt werden: Wenn das **Sprite** sich auf der linken Hälfte der Bühne befindet soll es sich vergrößern, ist es auf der rechten Hälfte der Bühne, soll es schrumpfen. Diese **Fallunterscheidung** soll in die **Endlosschleife** aus dem Beispielprogramm 3 mit eingebunden werden.

Die Tabellarische Szenenübersicht sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1 Kostüm2		-gehe langsam über die Bühne -pralle am Rand ab -Bewegungsbild -schrumpfen oder wachsen	-grüne Flagge -wiederhole endlos: 1. -pralle vom Rand ab 2. -gehe 10 Schritte 3. -warte 0,25 Sekunden 4. -Kostümwechsel 5. -Fallunterscheidung mit der Bedingung x-Position < 0 und vergrößern oder verkleinern

Es soll jedesmal, wenn die Katze 10 Schritte gegangen ist, nachgefragt werden, ob die Katze auf der linken Seite der Bühne ist. Ist dieses der Fall (**Wenn-Fall**), so soll die Katze größer werden. Ist dieses nicht der Fall (**Sonst-Fall**) so soll die Katze kleiner werden.



Die Katze ist auf der linken Seite der Bühne, wenn ihre **x-Position** zwischen -240 und -1 ist.

Um diese **Fallunterscheidung** zu realisieren, klicken Sie in der **Palettenauswahl** auf die **Steuerungspalette**.



Ziehen Sie den Block in den **Programmierbereich**.

In die Aussparung neben dem **falls** wird die **Bedingung** eingesetzt. In diesem Fall ist die **Bedingung**: Die **x-Position** des Sprites ist kleiner als Null.

Setzen Sie die **Bedingung** zusammen, indem Sie aus der **Operatorenpalette** den Block

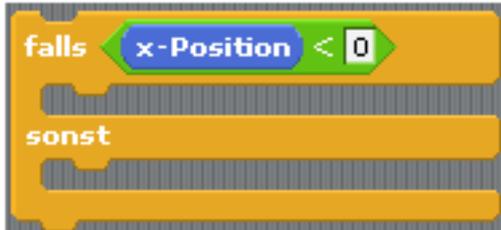


mit dem Block **x-Position** aus der **Bewegungspalette** verbinden. Der Block mit der **x-Position** der Katze muss auf die linke Seite des < Zeichens. Tragen Sie auf der rech-

ten Seite des < Zeichens eine 0 ein.



Setzen Sie nun die **Bedingung**  in den Block ein:



Im **Wenn**-Fall soll die Katze größer werden. Klicken Sie in der **Palettenauswahl** auf die **Aussehenpalette** und ziehen Sie den Block



 in die obere Aussparung der **Fallunterscheidung**. Setzen Sie den Wert auf 5.

Im **Sonst**-Fall soll die Katze kleiner werden, ziehen Sie dafür den Block



, den Sie auch in der **Aussehenpalette** finden, in die untere Aussparung der **Fallunterscheidung**. Setzen Sie den Wert auf -5.



Ziehen Sie die **Fallunterscheidung** an das Ende der **Endlosschleife** aus dem Beispielprogramm 3:



## 3.10 Aufgabe 5

Erweitern Sie das Beispielprogramm 3:

Jedesmal wenn die Katze den Rand berührt, soll sie den **Klang** Miau ganz abspielen. Lassen Sie außerdem die Katze für eine Sekunde sagen, dass sie den Rand berührt.

Die benötigten Blöcke finden Sie in den Paletten **Steuerung**, **Fühlen**, **Klänge** und **Aussehen**.

Den Klang finden Sie im **Klängereiter**.

## 3.11 Beispielprogramm 6

### 3.11.1 Neues Sprite erstellen

Zwei **Sprites**, das **Katzensprite** und sein Dialogpartner, sollen nun miteinander kommunizieren.

Schreiben Sie ein neues Programm.

Zuerst wird ein neues **Sprite**, der Dialogpartner, benötigt. Erstellen Sie ein neues **Sprite**, indem Sie eins importieren, selber malen oder ein Überraschungssprite importieren. Klicken Sie dafür jeweils auf einen der drei Buttons:



### 3.11.2 Senden einer Nachricht

Das Katzensprite soll nun den Dialogpartner ansprechen, sobald die **grüne Flagge** angeklickt wird. Dann soll es dem Partner signalisieren, dass es den Partner angesprochen hat. Die Tabellarische Szenenübersicht sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-sage Hallo! -sende Nachricht	-grüne Flagge -sage für 2 Sekunden Hallo! -sende Nachricht1 an alle
Partner	1.Kostüm			



Dafür ziehen Sie erst einmal den Block in den **Programmierbereich**.

Lassen Sie das Katzensprite für 2 Sekunden *Hallo!* sagen, indem Sie aus der **Aussehen-**

**palette** den Block unter den Block mit der **grünen Flagge**

setzen.

*Benutzen Sie den anderen Sagen-Block (ohne die Angabe wie lange etwas gesagt werden soll) um die Katze etwas sagen zu lassen, so bleibt die Sprechblase bestehen, bis die Katze etwas anderes sagt oder eine leere Sprechblase erzeugt wird.*

Um nun dem Dialogpartner zu signalisieren, dass das Katzensprite mit seiner Rede fertig ist, muss es eine **Nachricht** an den Dialogpartner schicken.

Ziehen Sie dafür den Block aus der **Steuerungpalette** in den **Programmierbereich**.

Klicken Sie auf den Pfeil in diesem Block und wählen Sie dann den Punkt **Neu...** aus:



Es erscheint dann ein kleines Fenster, in dem Sie die **Nachricht** benennen können:



Setzen Sie diesen Block unter die anderen:



Benutzen Sie den anderen Senden-Block, welcher wartet, so wartet das Programm des sendenden Sprites, bis das Programm fertig ausgeführt ist, welches unter dem Empfangen-Block der empfangenden Sprites ist. In diesem Fall würde es keinen Unterschied machen, welcher Block ausgeführt wird, da nach den entsprechenden Blöcken keine weiteren Blöcke folgen.

### 3.11.3 Empfangen einer Nachricht

Klicken Sie nun das Vorschaublid des anderen Sprites an.

Der Dialogpartner muss nun die **Nachricht** empfangen.

Die Tabellarische Szenenübersicht (für die gesamten Anweisungen von 3.11.3) sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-sage <i>Hallo!</i> -sende Nachricht	-grüne Flagge -sage für 2 Sekunden <i>Hallo!</i> -sende Nachricht1 an alle
Partner	1.Kostüm		-empfangen Nachricht -sage <i>Guten Tag!</i> -sende Nachricht	-empfangen Nachricht1 -sage <i>Guten Tag!</i> für 2 Sekunden -sende Nachricht2



Ziehen Sie dafür aus der **Steuerungpalette** den Block

in den **Programmierbereich**. Ist nicht schon die **Nachricht** *Hallo!* eingegeben, so wählen Sie diese über den Pfeil aus.

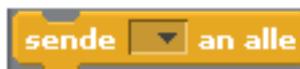
Der Dialogpartner soll jetzt etwas antworten. Lassen Sie ihn für 2 Sekunden *Guten Tag!*



sagen, indem Sie den Block

aus der **Aussehenpalette** mit verändertem Text unter den Empfangen-Block ziehen.

Dem Katzensprite soll jetzt signalisiert werden, dass der Dialogpartner mit seiner Rede fertig ist. Der Dialogpartner muss also seinerseits eine **Nachricht** senden.



Ziehen Sie also den Block

unter den Sagen-Block Block und erstellen

Sie eine neue **Nachricht**: *Begrüßungsantwort*.

Sie bekommen dann ein Programm der Form:

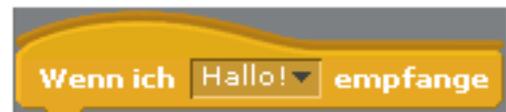


### 3.11.4 Sprechtexterweiterung

Das Katzensprite soll nun seinerseits antworten.

Die Tabellarische Szenenübersicht für die Sprachtexterweiterung sieht wie folgt aus:

Akteure	Kostüme	Klänge	Regieanweisung	technische Regieanweisung
Katze	Kostüm1		-sage <i>Ich freue mich dich zu sehen!</i>	-empfangen Nachricht2 -sage für 2 Sekunden <i>Ich freue mich dich zu sehen!</i>
Partner	1.Kostüm			



Ziehen Sie dafür aus der **Steuerungpalette** den Block

in den **Programmierbereich**. Ist nicht schon die **Nachricht** *Begrüßungsantwort* eingegeben, so wählen Sie diese über den Pfeil aus.

Lassen Sie die Katze für 2 Sekunden sagen, dass Sie sich freut, den Dialogpartner zu sehen.



Benutzen Sie dafür wieder den Block

aus der **Aussehenpalette** mit verändertem Text.

Die Katze bekommt also zwei Programme:



## 3.12 Aufgabe 6

Erweitern Sie das Beispielprogramm 6:

Setzen Sie die **Positionen** und die **Richtungen** der **Sprites** fest, sodass sie sich wenn die

**grüne Flagge** angeklickt wird gegenüberstehen.

Verlängern Sie den Dialog um mindestens 3 weitere Antworten.

Lassen Sie danach die Katze sich umdrehen, an den Rand laufen und sich dort verstecken.

Wird das Programm nochmals gestartet, soll die Katze wieder sichtbar sein.

## 3.13 Lösungen

### 3.13.1 Lösung zu Aufgabe 1



### 3.13.2 Lösung zu Aufgabe 2



### 3.13.3 Lösung zu Aufgabe 3

Programm für die Bühne:



### 3.13.4 Lösung zu Aufgabe 4

Beispiel:



### 3.13.5 Lösung zu Aufgabe 5



### 3.13.6 Lösung zu Aufgabe 6

Beispielprogramme für die Katze:



Beispielprogramme für den Dialogpartner:



# Kapitel 4

## Handbuch

Übersetzung des englischen Referenzhandbuchs für Scratch 1.4 ins Deutsche.

### 4.1 Einführung

Scratch ist eine neue Programmiersprache, die es leicht macht, interaktive Geschichten, Spiele und Animationen zu erstellen und deine Kreationen mit anderen im Netz zu teilen.

Diese Referenzanleitung liefert einen Überblick über die Scratch-Software.

Wenn du gerade erst mit Scratch angefangen hast, empfehlen wir, zuerst den *Getting Started Guide* anzuschauen (<http://scratch.mit.edu/files/ScratchGettingStarted.pdf>). Wenn du ausführlichere Information wünschst, hilft dir diese Referenzanleitung.

Die Scratch Webseite stellt viele weitere Hilfsmittel bereit, die dir helfen, Scratch zu erlernen: Videoanleitungen, Scratchkarten und häufig gestellte Fragen (FAQ).

Siehe <http://info.scratch.mit.edu/Support>.

Dieses Handbuch ist für die Scratchversion 1.4. Für die frühere Version dieser Referenzanleitung, siehe: <http://info.scratch.mit.edu/Support>.

Scratch wird von der Lifelong Kindergarten Group (Gruppe des lebenslangen Kindergartens) vom MIT Media Lab entwickelt, mit finanzieller Unterstützung von der National Science Foundation, Microsoft, Intel Foundation, Nokia und der MIT Media Lab Research Consortia.

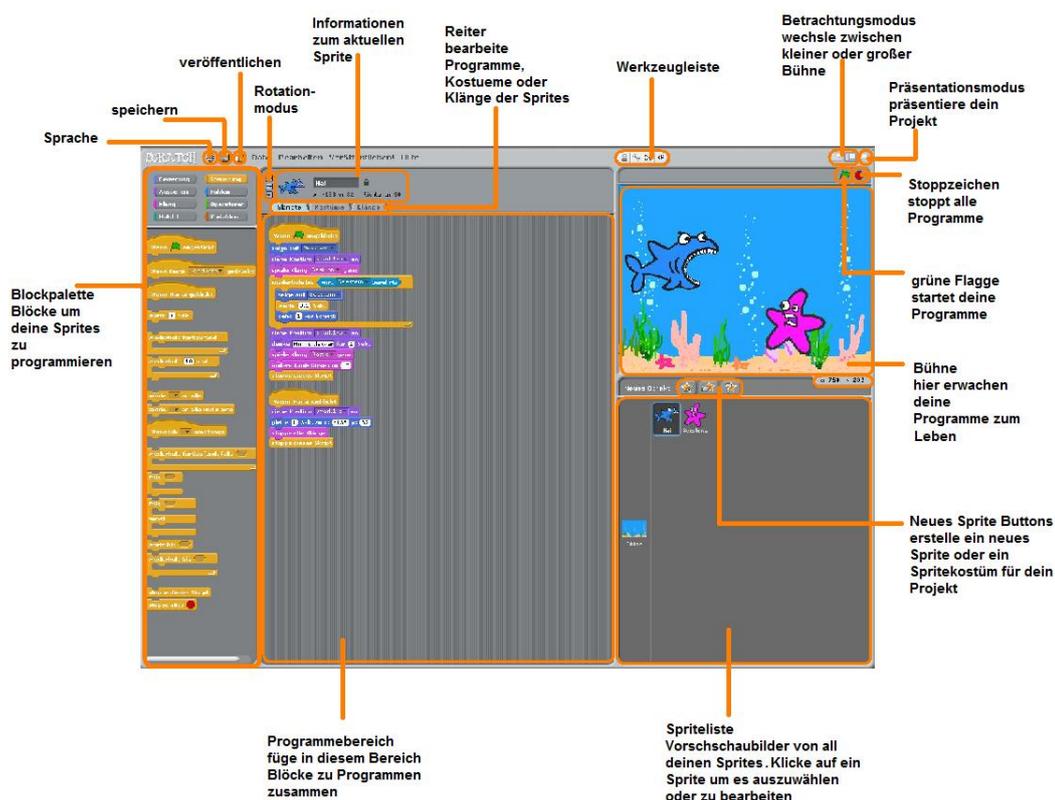
#### 4.1.1 Grundlagen eines Scratch Projekts

Ein Scratch Projekt besteht aus Objekten, die Sprites genannt werden. Du kannst das Aussehen eines Sprites verändern, indem du ihm ein anderes Kostüm gibst. Es ist möglich ein Sprite wie eine Person, einen Zug, einen Schmetterling oder etwas anderes aussehen zu lassen. Dabei kann jedes mögliche Bild als Kostüm verwendet werden: Du kannst ein Bild im Paint Editor erstellen, ein Bild von deiner Festplatte importieren oder ein Bild von einer Webseite ziehen.

Man kann ein Sprite Anweisungen ausführen lassen, ihm sagen sich zu bewegen oder Musik abzuspielen oder ihn mit anderen Sprites interagieren lassen. Um einem Sprite zu

erklären, was es tun soll, musst du grafische Blöcke zu Programmen, genannt Skripte, zusammenfügen. Wenn du ein Programm anklickst, führt Scratch die Blöcke von oben bis unten aus.

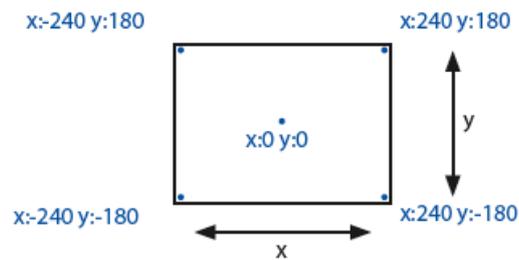
## 4.2 Das Scratch Programmfenster



### 4.2.1 Die Bühne

Die Bühne ist der Platz, wo deine Geschichten, Spiele und Animationen zum Leben erwachen. Sprites bewegen sich und interagieren mit anderen Sprites auf der Bühne.

Die Bühne ist 480 Einheiten breit und 360 Einheiten hoch. Sie wird in ein x-y-Koordinatensystem unterteilt. Der Mittelpunkt der Bühne hat die x-Koordinate 0 und die y-Koordinate 0.



Um x-y-Positionen auf der Bühne herauszufinden, bewege die Maus (Cursor) umher. Die jeweilige x-y-Position befindet sich direkt rechts unter der Bühne.

x: -109 y: -53

 Klicke den *Kleine Bühnengröße*-Button oder den *Volle Bühnengröße*-Button, um zwischen kleiner oder großer Bühnengröße hin und her zu wechseln. Du kannst die kleine Bühnenansicht verwenden, um Scratch auf kleinen Bildschirmen anzuzeigen oder den Skript-Bereich zu vergrößern.

 Klicke auf den *Präsentationsmodus*-Button, um dein Projekt in voller Größe zu betrachten. Um den Präsentationsmodus zu verlassen, drücke die ESC-Taste oder den Pfeil, der oberhalb der Bühne erscheint.

### 4.2.2 Neue Sprites

Wenn du ein neues Scratchprojekt beginnst, erscheint ein einzelnes Katzensprite. Um neue Sprites zu erschaffen, klicke einen dieser Buttons:



Zeichne mit dem Paint Editor dein eigenes Kostüm für ein neues Sprite.



Importiere ein neues Sprite.



Importiere ein Überraschungssprite.

Wenn du ein Sprite löschen möchtest, wähle die *Schere* aus der Werkzeugsymbolleiste und klicke das Sprite an. Oder du klickst mit der rechten Maustaste (Mac: Ctrl+klick) das Sprite an und wählst *Löschen* aus dem Kontextmenü.

Um ein Sprite zu erschaffen, das wie ein Teil der Bühne aussieht, klicke mit rechts (Mac: Ctrl+klick) auf die Bühne und wähle *Neues Objekt aus Bildschirmbereich erstellen* aus dem Kontextmenü aus. Markiere nun den Bereich der Bühne, der das neue Sprite darstellen soll.

### 4.2.3 Spriteliste

Die Spriteliste zeigt Vorschaubilder aller Sprites im Projekt. Der Name jedes Sprite erscheint unter seinem Vorschaubild.



Um das Skript, das Kostüm und die Klänge eines Sprites zu sehen und ändern, klicke das Vorschaubild des Sprites in der Spriteliste an oder klicke doppelt auf das Sprite auf der Bühne. (Das ausgewählte Sprite wird in der Spriteliste hervorgehoben und in blauer Farbe umrissen).

Um ein Sprite zu *zeigen*, *exportieren*, *kopieren* oder *löschen* klicke mit der rechten Maustaste (Mac: Ctrl+klick) auf das Vorschaubild des Sprites in der Spriteliste und wähle im Kontextmenü den entsprechenden Unterpunkt. Um ein Sprite zu zeigen, der nicht auf der Bühne ist oder der sich versteckt, drücke Shift und klicke auf das Vorschaubild des Sprites in der Spriteliste - dadurch wird das Sprite in die Mitte der Bühne geholt oder wieder gezeigt.

Du kannst die Anordnung der Sprites in der Spriteliste neu ordnen, indem du die Vorschaubilder an die gewünschte Position ziehst.

Ebenso wie ein Sprite sein Aussehen mit einem Kostümwechsel ändern kann, änderst du auch das Aussehen der Bühne, indem du die Hintergründe wechselst. Um die Programme, Hintergründe und Klänge der Bühne zu sehen und zu verändern, klicke auf das Bühnensymbol, welches sich links von der Spriteliste befindet.

### 4.2.4 Blockpalette und Programmebereich

Um ein Sprite zu programmieren, ziehe Blöcke von der Blockpalette zum Programmebereich. Um einen Block auszuführen, klicke ihn an.

Entwerfe Programme (Skripte) indem du Blöcke zu Stapeln zusammenfügst. Klicke auf irgendeinen Block im Stapel, um das vollständige Programm von oben bis unten laufen zu lassen.

Um herauszufinden was ein Block macht, klicke mit rechts (Mac: Ctrl+klick) auf ihn, wähle dann *Hilfe* aus dem Kontextmenü. Wenn du einen Block in den Programmebereich

ziehst, zeigt eine weiße Linie an, wo du den Block mit anderen Blöcken zusammenfügen kannst.

Um einen Stapel zu verschieben, hebe ihn am obersten Block an. Wenn du einen Block aus der Mitte eines Stapels ziehst, bewegen sich alle Blöcke unter ihm mit. Um ein Programm von einem Sprite zum anderen zu kopieren, ziehe den Stapel zum Vorschabild des anderen Sprites in der Spriteliste.

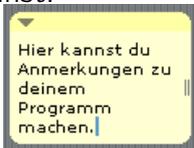
Einige Blöcke sind mit einem weißen Textfeld ausgestattet, wie beispielsweise:

. Um den Wert im Textfeld zu ändern, klicke innerhalb des weißen Textfelds, lösche den Wert und gebe einen neuen Wert ein. Du kannst auch einen gerundeten Block, wie beispielsweise:  in das Textfeld einfügen.

Einige Blöcke haben auch Ausklappmenüs, wie beispielsweise: . Klicke den Pfeil an, um das Menü zu sehen, dann klicke auf den ausgewählten Unterpunkt.

Um den Programmebereich aufzuräumen, klicke mit rechts (Mac: Ctrl+klick) auf den Hintergrund des Programmebereichs und wähle *Aufräumen* aus dem Kontextmenü. Um ein Bildschirmfoto des Programmebereichs zu machen, klicke mit rechts (Mac: Ctrl+click) auf den Hintergrund des Programmebereichs und wähle *Skripte als Bild speichern* aus dem Kontextmenü.

Um eine Anmerkung zum Programmebereich zu machen, klicke mit rechts (Mac: Ctrl+klick) auf den Hintergrund des Programmebereichs und wähle *Anmerkung hinzufügen* aus dem Kontextmenü. Ein gelber Anmerkungsbereich erscheint, in den du deinen Kommentar einfügen kannst.



Um die Breite des Anmerkungsbereichs zu ändern, nutze den Griff am rechten Rand. Klicke den Pfeil an der linken, oberen Seite, um den Anmerkungsbereich einzuklappen oder auszuklappen.

Anmerkungen können überall im Programmebereich hinzugefügt werden. Um sie zu bewegen, ziehe sie an den gewünschten Platz.

Um eine Anmerkung mit einem Block zu verbinden, ziehe die Anmerkung auf den Block. Um die Anmerkung vom Block zu trennen, ziehe die Anmerkung vom Block weg.

## 4.2.5 Kostüme

Klicke *Kostüme* auf dem Reiter an, um die Kostüme des Sprites zu sehen und zu ändern.



Dieses Sprite hat zwei Kostüme. Das gegenwärtige Kostüm (Hai 2) ist hervorgehoben.

Um zu einem anderen Kostüm zu wechseln, klicke einfach auf das gewünschte Vorschaubild des Kostüms.

Es gibt drei Möglichkeiten, neue Kostüme zu erstellen:

- \* Klicke **Malen** an, um ein neues Kostüm mit dem Paint Editor zu malen.
- \* Klicke **Importieren** an, um ein Kostüm von deiner Festplatte zu importieren. Du kannst auch Bilder aus dem Internet importieren.
- \* Klicke **Kamera** an, um ein Foto mit deiner (integrierten oder an den Computer angeschlossenen) Webcam zu machen. Jedes Mal wenn du  anklickst oder die Leertaste drückst, nimmt sie ein Foto auf.

- \* Kopiere Bilder aus dem Internet.

Scratch erkennt viele Bildformate: JPG, BMP, PNG, GIF (einschließlich animated GIF).

Jedes Kostüm hat eine Kostümmzahl (links vom Vorschaubild angezeigt). Du kannst die Anordnung der Kostüme ändern, indem du die Vorschaubilder an die gewünschte Stelle ziehst. Die Kostümnummer ändert sich entsprechend, wenn du die Anordnung veränderst.

Klicke mit rechts (Mac: Ctrl+klick) auf ein Kostümvorschaubild, um das Kostüm in ein neuen Sprite umzuwandeln oder eine Kopie des Kostüms zu exportieren.

### 4.2.6 Klänge

Klicke den *Klänge*-Reiter an, um die Klänge des Sprites zu sehen.



Du kannst neue Klänge aufnehmen oder importieren. Scratch kann MP3-Dateien lesen, sowie unkomprimierte WAV, AIF und AU-Dateien (mit 8- oder 16-Bits pro Sample, aber nicht 24-Bits pro Sample).

### 4.2.7 Informationen zum aktuellen Sprite



Die Informationen zum aktuellen Sprite zeigen den Spritennamen, die aktuelle x-Position, die aktuelle y-Position, die Richtung und die derzeitige Stiftfarbe des aktuellen Sprites an und ob das Sprite im Präsentationsmodus und auf der Webseite mit der Maus verschoben werden kann. Du kannst dem Sprite einen neuen Namen geben.

Die Richtung des Sprites zeigt an, in welche Richtung (0 = oben, 90 = rechts, 180 = unten, -90 = links) das Sprite sich bewegt, wenn es einen Bewegungsblock ausführt. Die blaue Linie auf dem Vorschaubild zeigt die Richtung des Sprites an. Ziehe an dieser Linie, um die Richtung des Sprites zu ändern. Klicke doppelt auf das Sprite, um die Richtung zurück zu seinem ursprünglichen Zustand (Richtung = 90) einzustellen.

Klicke das geschlossene Schloss an, um das Sprite im Präsentationsmodus und auf der Webseite bewegen zu können. Klicke auf das geöffnete Schloss, um dies rückgängig zu machen.

Die gegenwärtige Stiftfarbe des Sprites kannst du sehen (neben dem Schloss), wenn der Stift abgesenkt ist.

Um ein Sprite zu exportieren, klicke mit rechts (Mac: Ctrl+klick) das Sprite auf der Bühne oder das Vorschaubild in der Spriteliste an. Das Sprite wird als .sprite-Datei gesichert, diese kannst du dann in ein anderes Projekt importieren.

### 4.2.8 Rotationsmodus

Klicke die *Rotationsmodus*-Buttons an, um festzulegen, wie die Kostüme der Sprites ihre Richtung ändern können.



Rotation: Das Kostüm dreht sich, wenn das Sprite seine Richtung ändert.



Rechts-Links: Das Kostüm zeigt nur nach links oder rechts.



Keine Rotation: Das Kostüm dreht sich nie (selbst wenn das Sprite die Richtung ändert).

### 4.2.9 Symbolleiste



Klicke auf ein Werkzeug aus der Werkzeugsymbolleiste. Dann klicke auf das Objekt, welches du bearbeiten möchtest, um die Aktion auszuführen.



: dupliziert Sprites, Kostüme, Klänge, Blöcke und Programme. (Shift+klick für Mehrfachanwendungen)



: löscht Sprites, Kostüme, Klänge, Blöcke und Programme. (Shift+klick für Mehrfachanwendungen)



: vergrößert Sprites. (Shift+klick für größere Schritte.)



: verkleinert Sprites. (Shift+klick für größere Schritte.)

Um zum Pfeil-Cursor zurückzukehren, klicke auf einen unbelegten Bereich des Bildschirms.

### 4.2.10 Menü



Klicke das *Sprachensymbol*  an, um die Sprache zu ändern. Jeder kann Sprachübersetzungen zu Scratch hinzufügen oder ändern. Um eine Übersetzung hinzuzufügen oder zu ändern, wende dich bitte an die Betreuung der Scratchwebseite (<http://info.scratch.mit.edu/Translation/>).

Klickst du das *Speichersymbol*  an, speicherst du dein Projekt.

Klickst du das *Veröffentlichensymbol*  an, kannst du dein Projekt auf die Scratchwebseite laden.

Vom Menüpunkt *Datei* kannst du ein neues Projekt oder ein vorhandenes Projekt öffnen und du kannst dein Projekt speichern.

*Projekt importieren* importiert alle Sprites und Hintergründe eines anderen Projekts in dein gegenwärtiges Projekt. Diese Eigenschaft ist nützlich, wenn du Sprites aus verschiedenen Projekten in einem Projekt kombinieren möchtest.

*Objekt exportieren* speichert das gegenwärtigen Sprite in einer *.sprite*-Datei, welche in andere Projekte importiert werden kann.

*Projektanmerkungen* öffnet ein Textfeld, in dem du Anmerkungen zu deinem Programm machen kannst, beispielsweise eine Anleitung für dein Programm.

Mit *Beenden* verlässt du das Scratchprogramm.

Unter dem Menüpunkt *Bearbeiten* findet man folgende Befehle:

*Löschen rückgängig machen* holt den letzten Block, Sprite oder das letzte Programm zurück, das du gelöscht hast.

Mit *Ausführung in Einzelschritten* wird dein Programm Schritt für Schritt ausgeführt. Dabei wird der Block, der gerade ausgeführt wird farbig hervorgehoben. Diese Eigenschaft ist nützlich, um Fehler in deinem Programm zu finden und sie hilft anderen Programmierern den Ablauf deines Programms zu verstehen. Mit *Einzelschritte einstellen* kannst du die Geschwindigkeit (von normal bis turbo) einstellen, mit der die Blöcke im Einzelschrittmodus ausgeführt werden.

Mit *Klänge komprimieren* und *Bilder komprimieren* kannst du die Klänge und Bilder in deinem Projekt komprimieren, um die Gesamtgröße deines Projekts zu verringern. Dabei kann sich die Qualität der Klänge und Bilder reduzieren.

*Motorblöcke anzeigen* fügt Motorblöcke in die Bewegungsblockpalette hinzu. Du kannst diese Bewegungsblöcke benutzen, um einen Motor, der an deinem Computer angeschlossen ist, zu programmieren. Die Motorblöcke arbeiten mit LEGO Education WeDo?

(<http://www.legoeducation.com>).

Mit dem Menüpunkt *Veröffentlichen!* kannst du dein Projekt auf der Scratchwebseite veröffentlichen.

Im Menüpunkt *Hilfe* findest du Links zu Referenzmaterialien, Tutorials und FAQs. Weitere Hilfe gibt es auch auf der Scratchwebseite.

### 4.2.11 Grüne Flagge

Die *grüne Flagge* hilft dir mehrere Programme gleichzeitig zu starten. Klickst du auf die grüne Flagge (oben rechts über der Bühne) , so beginnen alle Programme, die mit dem

Block  anfangen.

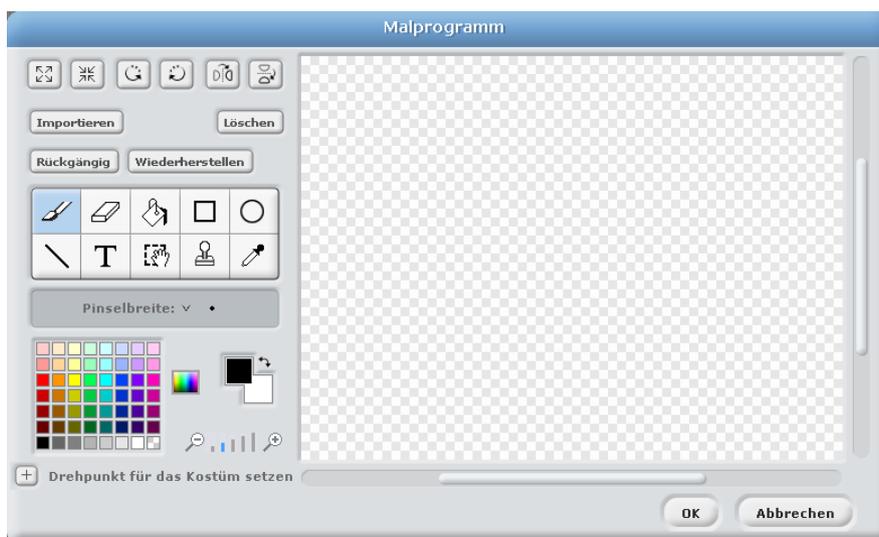
Solange dein Programm läuft, wird die grüne Flagge farbig hervorgehoben.

Im Präsentationsmodus hat das Drücken der Enter-taste den gleichen Effekt, wie das Klicken der grünen Flagge.

Sobald du auf der Scratchwebseite ein Projekt öffnest, wird dieses automatisch gestartet.

### 4.2.12 Paint Editor

Du kannst den *Paint Editor* benutzen, um Kostüme und Hintergründe zu malen oder zu ändern.



Die Paint Editor Werkzeugleiste stellt folgende Werkzeuge zur Verfügung:



: malt mit der derzeitigen Farbe. Klickst du auf dieses Symbol, so siehst du im Feld unter der Werkzeugleiste die derzeitige Pinselbreite. Klicke auf den Pfeil , um eine andere Pinselbreite auszuwählen.



: Löscht Bereiche. Gelöschte Bereiche werden transparent. Wenn du auf dieses Werkzeug klickst, siehst du im Feld unter der Werkzeugleiste die derzeitige Breite des Radiergummis. Klicke auf den Pfeil , um eine andere Breite auszuwählen.



: Füllt verbundene Bereiche einfarbig oder mit einem Farbverlauf. Wenn du dieses Werkzeug anklickst, zeigt das Feld unter der Werkzeugleiste die Füllart (einfarbig, horizontaler Farbverlauf, vertikaler Farbverlauf oder sternförmiger Farbverlauf). Farbverläufe mischen die gewählte Vordergrundfarbe mit der gewählten Hintergrundfarbe.



: Zeichnet ein gefülltes oder umrandetes Rechteck (halte Shift gedrückt für ein Quadrat) unter Verwendung der gegenwärtigen Vordergrundfarbe. Wenn du dieses Werkzeug anklickst, zeigt das Feld unter der Werkzeugleiste die Füllart (gefüllt oder umrandet).



: Zeichnet eine gefüllte oder umrandete Ellipse (halte Shift gedrückt für einen Kreis) unter Verwendung der gegenwärtigen Vordergrundfarbe. Wenn du dieses Werkzeug anklickst, zeigt das Feld unter der Werkzeugleiste die Füllart (gefüllt oder umrandet).



: Zeichnet eine gerade Linie (halte Shift gedrückt für eine horizontale oder vertikale Linie) unter Verwendung der gegenwärtigen Vordergrundfarbe. Wenn du dieses Werkzeug anklickst, zeigt das Feld unter der Werkzeugleiste die Pinselbreite. Klicke auf den Pfeil , um eine andere Breite auszuwählen.



: Fügt einer Zeichnung einen Text hinzu. Wenn du dieses Werkzeug anklickst, kannst du im Feld unter der Werkzeugleiste die Schriftart und die Schriftgröße ändern. Jedes Kostüm kann nur einen Text haben.



: Wählt einen rechteckigen Bereich aus, der auf eine neue Position verschoben werden kann (drücke löschen um die Auswahl zu entfernen, drücke Shift+Delete oder Shift+Backspace um nur den markierten Bereich zu bekommen).



: Wählt einen rechteckigen Bereich aus und kopiert ihn an eine neue Position (halte Shift gedrückt für wiederholtes stempeln).



: Verwendet die Spitze der Pipette, um eine Vordergrundfarbe zu wählen (klicke die Leinwand an, um eine Farbe von außerhalb der Farbpalette auszuwählen).

Die gegenwärtigen Farben (Vordergrund und Hintergrund) werden unten im Wahlbereich



gezeigt. Du kannst den Farbenvertauschungspfeil anklicken, um die Vordergrund- und Hintergrundfarben auszutauschen. Klicke in eine der Farbpaletten, um eine neue Vordergrundfarbe (Shift+klick zum Auswählen einer Hintergrundfarbe) auszuwählen.

Klicke den Button  Drehpunkt für das Kostüm setzen um den Punkt des Sprites festzulegen um welchen sich das Sprite drehen soll, wenn es auf der Bühne ist.

Klicke die Symbole , um den Vergrößerungsmaßstab zu verkleinern oder zu vergrößern. Wenn der Zoom größer als 100% ist, werden Scrollbars benutzt, um den Leinwandbereich zu verschieben. Zoomen ändert nicht die Größe des Bildes.

Klicke auf den Button , um ein Bild aus einer Datei zu öffnen und es auf der Leinwand hinzuzufügen.

Klicke auf den Button , um alles von der Leinwand zu entfernen.

Um die Größe deines Bildes oder des gerade gewählten Bereichs zu ändern, klicke auf die

Buttons  oben links. Halte Shift gedrückt und klicke auf einen der Buttons. Es erscheint dann ein Fenster, in dem du einen exakten Wert eintragen kannst. *Schrumpfen* verringert die Größe und die Auflösung des Bildes.

Um dein Bild oder den gerade gewählten Bereich zu drehen, klicke auf die Buttons . Halte Shift gedrückt und klicke auf einen der Buttons, dann erscheint ein Fenster, in dem du einen exakten Wert eintragen kannst.

Um dein Bild oder den gerade gewählten Bereich zu spiegeln, klicke auf die Buttons



Hast du einen Fehler gemacht, kannst du mit dem Button  den letzten Schritt (oder durch wiederholtes drücken die letzten Schritte) rückgängig machen. Möchtest du

das Löschen rückgängig machen, so klicke den Button .

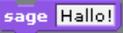
## 4.3 Scratchblöcke

### 4.3.1 Blockarten

Es gibt drei Hauptarten von Blöcke in der Blockpalette:

**Stapelbare Blöcke:** Diese Blöcke haben Beulen an der Unterseite und/oder Einbuchtungen an der Oberseite, wie beispielsweise . Du kannst diese Blöcke zu Stapeln zusammenfügen. Einige stapelbaren Blöcke haben einen Eingabebereich, wo du eine Zahl eingeben kannst (wie 10 im Block ). Bei einigen Blöcken kannst du auch ein Element aus einem Ausklappmenü wählen, (wie beispielsweise *Miau* im Block ). Einige Stapelblöcke, wie beispielsweise  sind C-förmig. In diese Blöcke kannst du andere Stapelblöcke einsetzen.

**Hüte:** Diese Blöcke haben gerundete Oberseiten, wie beispielsweise . Sie werden auf die Spitze der Stapel gesetzt. Sie warten auf ein Ereignis: zum Beispiel wenn eine bestimmte Taste gedrückt wird, dann werden die Blöcke unter ihm ausgeführt.

**Reporter:** Diese Blöcke, wie beispielsweise  und  wurden entworfen, um in den Eingabebereich anderer Blöcke zu passen. Reporter mit gerundeten Enden (wie beispielsweise  oder ) geben Zahlen oder Zeichenketten zurück und passen in Blöcke mit runden oder rechteckigen Eingabebereichen (wie beispielsweise  oder ). Reporter mit spitzen Enden (wie beispielsweise ) geben boolesche Werte (*wahr* oder *falsch*) zurück und passen in Blöcke mit spitzen oder rechteckigen Eingabebereichen (wie beispielsweise  oder ).  
Klickst du auf einen Reporterblock, so erscheint sein gegenwärtiger Wert.

Einige Reporterblöcke haben ein Auswahlkästchen neben sich, wie beispielsweise .

Wenn du in ein solches Auswahlkästchen anklickst, dann erscheint ein Anzeigefeld auf der Bühne - dieses zeigt den aktuellen Wert des Reporters an. Wenn sich der Wert des Reporters ändert, wird das Anzeigefeld automatisch angepasst. Das Anzeigefeld kann den Wert eines Reporters auf verschiedenen Arten darstellen:

 eine kleine Anzeige mit dem Namen des Reporters,

 eine große Anzeige ohne Namen und

 ein Schieber mit dem man den Wert des Reporters verändern kann (nur bei Variablen verfügbar).

Klicke doppelt oder mit rechts (Mac: Ctrl+klick) auf ein Anzeigefeld, um von einem Format zum anderen zu wechseln. Das Schieber-Format ist nur bei benutzerdefinierten Va-

riablen verfügbar. Klicke mit rechts (Mac: Ctrl+klick) auf den Schieber, um seinen minimalen und maximalen Wert anzupassen.

### 4.3.2 Listen

Du kannst in Scratch Listen erstellen und verändern. Listen können Zahlen, sowie Buchstabenfolgen und andere Zeichen speichern.

Um eine Liste zu erstellen, gehe in die Variablenblockpalette und klicke  an. Sobald du eine Liste erstellst, erscheinen einige Listenblöcke. Die Listenblöcke sind im Abschnitt 4.3.13 beschrieben.

Wenn du eine Liste erstellst, erscheint auf der Bühne eine Listenanzeige. Die Listenanzeige zeigt alle Elemente in einer gegebenen Liste. Du kannst neue Elemente direkt in die Listenanzeige schreiben.

Zuerst ist die Liste leer und hat die Länge 0. Um der Liste ein Element hinzuzufügen, klicke auf das Zeichen + in der Listenanzeige unten links. Es erscheint ein neuer Eintrag, den du benennen kannst, wenn du auf den Eintrag klickst. Die Länge der Liste nimmt um 1 zu. Alternativ kannst du ein Element der Liste hinzufügen, indem du einen Listenblock (beispielsweise ) verwendest.

Du kannst die Größe der Listenanzeige ändern, indem du an der unteren rechten Ecke ziehst.

Anmerkung: Klickst du mit rechts (Mac: Ctrl+klick) auf eine Listenanzeige, kannst du die Liste in eine .txt-Datei exportieren. Ebenso kannst du gespeicherte .txt-Dateien importieren, bei denen Werten in einzelnen Reihen stehen.

### 4.3.3 Zeichenketten

Zeichenketten (Strings genannt) bestehen aus Buchstaben, Wörtern oder anderen Zeichen (wie beispielsweise: Apfel; Oktober 2009; Sie gewinnen!).

Zeichenketten können in Variablen oder in Listen gespeichert werden (wie beispielsweise  oder ).

Du kannst Zeichenketten unter Verwendung von  zusammenfügen. Mit den folgenden Blöcken kannst du Zeichenketten miteinander vergleichen: ,  oder .

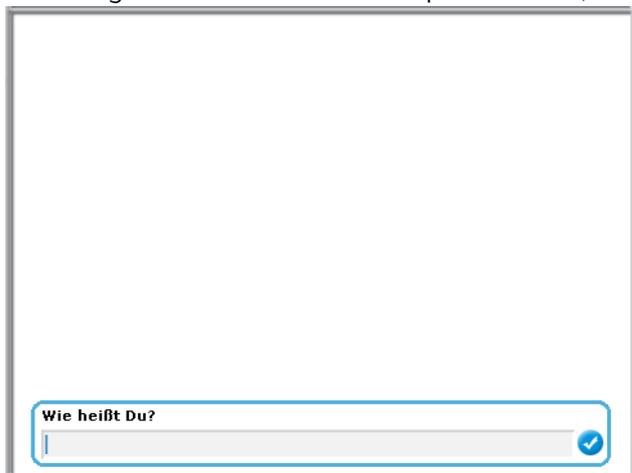
Zeichenketten werden als 0 ausgewertet, wenn sie in mathematischen Blöcken vorkommen (wie beispielsweise: ) und in den Blöcken, die eine Zahl erwarten (wie  und .

### 4.3.4 Tastatureingabe

Du kannst **Frage Wie heißt Du? und warte** verwenden, um einen Benutzer aufzufordern, eine Antwort mit der Tastatur einzutippen. Die Antwort wird in **Antwort** gespeichert. Die Frage erscheint auf dem Bildschirm und das Programm wartet, bis die Enter-Taste betätigt wird oder das Häkchen angeklickt wird.



Die Frage erscheint in einem Sprachballon, wenn sie von einem Sprite verwendet wird.



Die Frage erscheint am unteren Bildschirmrand, wenn sie von der Bühne verwendet wird.

**Antwort** ist für alle Sprites sichtbar (global) und wird jedes Mal ersetzt, wenn **Frage Wie heißt Du? und warte** ausgeführt wird. Wenn du den gegenwärtigen Wert von **Antwort** speichern möchtest, kannst du ihn in einer Variable oder in einer Liste sichern.

### 4.3.5 Bewegungen

**gehe 10 -er Schritt**: bewegt das Sprite vorwärts oder rückwärts

**drehe ↻ 15 Grad**: dreht das Sprite rechts herum

**drehe ↺ 15 Grad**: dreht das Sprite links herum

: lässt das Sprite in die eingegebenen Richtung (0=oben, 90=rechts, 180=unten, -90=links) zeigen

: lässt das Sprite in Richtung des Mauszeiger oder zu einem anderen Sprite zeigen

: bewegt das Sprite zur eingegebenen x-y-Koordinate auf der Bühne

: bewegt das Sprite zur Position des Mauszeigers oder zu einem anderen Sprite

: lässt das Sprite zur eingegebenen Position in der angegebenen Zeitspanne gleiten

: ändert die x-Position des Sprites um den eingegebenen Wert

: setzt die x-Position des Sprites auf den eingegebenen Wert

: ändert die y-Position des Sprites um den eingegebenen Wert

: setzt die y-Position des Sprites auf den eingegebenen Wert

: dreht das Sprite um 180 Grad, wenn das Sprite einen Rand der Bühne berührt

: gibt die x-Position des Sprites an (Wert kann zwischen -240 und 240 variieren)

: gibt die y-Position des Sprites an (Wert kann zwischen -180 und 180 variieren)

: gibt die Richtung des Sprites an (0=oben, 90=rechts, 180=unten, -90=links)

### 4.3.6 Motorblöcke

Die Motorblöcke erscheinen nur, wenn du in der Menüleiste unter dem Punkt *Bearbeiten* den Unterpunkt *Motor-Bausteine anzeigen* auswählst oder wenn du einen LEGO WeDo anschließt. (Siehe dafür <http://www.legoeducation.com>)

: schaltet für die eingegebene Zeit den Motor an

: schaltet den Motor an

: schaltet den Motor aus

: setzt die Motorkraft auf den eingegebenen Wert (zwischen 0 und 100)

: setzt oder ändert die Richtung des Motors, schaltet den Motor aber nicht an ( in diese Richtung=im Uhrzeigersinn, in jene Richtung=gegen den Uhrzeigersinn, umkehren=drehe um 180 Grad)

### 4.3.7 Aussehen

: ändert das Aussehen des Sprites, indem ein anderes Kostüm angezogen wird

: ändert das Kostüm des Sprites in das nächste Kostüm aus der Kostümliste (wenn das Ende der Kostümliste erreicht wird, wird zum ersten Kostüm der Kostümliste gewechselt)

 **Kostüm Nr.** : zeigt die aktuelle Kostümnummer

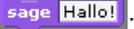
Ebenso ändern die 3 folgenden Buttons die Hintergründe der Bühne.

 **wechsle zum Hintergrund**  : ändert das Aussehen der Bühne, indem ein anderer Hintergrund angezogen wird

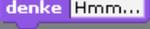
 **nächster Hintergrund** : ändert den Hintergrund der Bühne in den nächsten Hintergrund aus der Hintergrundliste

 **Hintergrund Nr.** : zeigt die aktuelle Hintergrundnummer

 **sage**  **für**  **Sek.** : eine Sprechblase mit dem angegebenen Text erscheint für die angegebene Dauer

 **sage**  : eine Sprechblase mit dem angegebenen Text erscheint (um die Sprechblase verschwinden zu lassen, kannst du eine Sprechblase ohne Text ausführen)

 **denke**  **für**  **Sek.** : eine Denkblase mit dem angegebenen Text erscheint für die angegebene Dauer

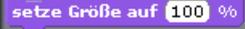
 **denke**  : eine Denkblase mit dem angegebenen Text erscheint

 **ändere**  **-Effekt um**  : ändere einen visuellen Effekt eines Sprites um den angegebenen Wert (benutze die Auswahlliste um den Effekt auszuwählen)

 **setze**  **-Effekt auf**  : setze einen grafischen Effekt eines Sprites auf den angegebenen Wert (die meisten visuellen Effekte können zwischen 0 und 100 variieren)

 **schalte Grafikeffekte aus** : löscht alle visuellen Effekte

 **ändere Größe um**  : ändert die Größe des Sprites um den angegebenen Wert

 **setze Größe auf**  : setzt die Größe des Sprites auf den angegebenen Wert

 **Größe** : zeigt die aktuelle Größe des Sprites in Prozent der Originalgröße

 **zeige dich** : lässt das Sprite auf der Bühne erscheinen

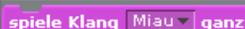
 **verstecke dich** : lässt das Sprite von der Bühne verschwinden (wenn ein Sprite versteckt ist, können andere Sprites es nicht mit dem *wird berührt?* -Block aufspüren)

 **komme nach vorn** : legt das Sprite vor alle anderen

 **gehe**  **Ebenen nach hinten** : lässt das Sprite um die angegebene Zahl an Ebenen nach hinten gehen (so kann sich das Sprite hinter anderen Sprites verstecken)

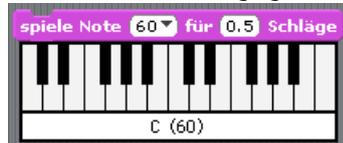
### 4.3.8 Klänge

 **spiele Klang**  : spielt den aus der Auswahlliste angegebenen Klang ab und geht unmittelbar zum nächsten Block über (sogar wenn der Klang noch abgespielt wird)

 **spiele Klang**  **ganz** : spielt den Klang ab und wartet bis der Klang abgespielt ist ehe der nächste Block ausgeführt wird

 **stoppe alle Klänge** : stoppt alle Klänge

**spiele Schlagzeug** 48 für 0.2 Schläge : der aus der Auswahlliste ausgewählte Schlagzeugton wird für die Dauer der angegebenen Zeit abgespielt



: spielt die angegebene Note (eine höhere Zahl bedeutet eine höhere Note) für die Dauer der angegebenen Zeit ab

**setze Instrument auf** 1 : ordnet das ausgewählte Instrument einem Sprite zu, damit Notenblöcke abgespielt werden können (Jedes Sprite besitzt ein eigenes Instrument.)

**ändere Lautstärke um** -10 : ändert die Lautstärke um den angegebenen Wert (Die Lautstärke kann zwischen 0 und 100 variieren.)

**setze Lautstärke auf** 100 % : setzt die Lautstärke auf den angegebenen Wert

**Lautstärke** : zeigt die aktuelle Lautstärke an

**ändere Tempo um** 20 : ändert das Tempo um den angegebenen Wert

**setze Tempo auf** 60 Schläge/Min. : setzt das Tempo auf den angegebenen Wert, gezählt wird in Schlägen pro Minute

**Tempo** : zeigt das aktuelle Tempo an

### 4.3.9 Malstift

**wische Malspuren weg** : wischt alle Malspuren und Stempel auf der Bühne weg

**senke Stift ab** : senkt den Stift des Sprites ab derart, dass das Sprite zeichnet, wenn es sich bewegt

**hebe Stift an** : hebt den Stift des Sprites an derart, dass es nicht mehr zeichnet, wenn es sich bewegt

**setze Stiftfarbe auf** [Farbe] : setzt die Stiftfarbe auf die, mit Hilfe der Farbpipette, gewählte Farbe

**ändere Stiftfarbe um** 10 : ändert die Stiftfarbe um den angegebenen Wert (Stiftfarbe 0=rotes Ende des Farbspektrums, Stiftfarbe 100=blaues Ende des Farbspektrums)

**setze Stiftfarbe auf** 0 : setzt die Stiftfarbe auf den angegebenen Wert

**ändere Farbstärke um** 10 : ändert die Farbstärke um den angegebenen Wert (0=sehr dunkel, 100=sehr hell)

**setze Farbstärke auf** 50 : setzt die Farbstärke auf den angegebenen Wert

**ändere Stiftdicke um** 1 : ändert die Stiftdicke um den angegebenen Wert

**setze Stiftdicke auf** 1 : setzt die Stiftdicke auf den angegebenen Wert

**hinterlasse Abdruck** : stempelt einen Abdruck des Sprites auf die Bühne

### 4.3.10 Steuerung



: das Programm unter dem Block wird ausgeführt, wenn die grüne Fahne angeklickt wird



: das Programm unter dem Block wird ausgeführt, wenn die angegebene Taste gedrückt wird



: das Programm unter dem Block wird ausgeführt, wenn das angegebene Sprite angeklickt wird



: wartet um den angegebenen Wert, bis der nächste Block ausgeführt wird



: wiederholt die Blöcke innerhalb des Blocks endlos oft



: wiederholt die Blöcke innerhalb des Blocks um den angegebenen Wert



: sendet eine Botschaft an alle Sprites und führt dann den nächsten Block aus, ohne auf die Beendigung der gestarteten Programme zu warten



: sendet eine Botschaft an alle Sprites und wartet dann auf die Beendigung aller gestarteten Programme, bevor der nächste Block ausgeführt wird



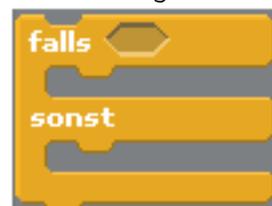
: das Programm unter dem Block wird ausgeführt, wenn die angegebene Botschaft empfangen wird



: kontrolliert fortlaufend die angegebene Bedingung. Ist diese *wahr*, so werden die Blöcke innerhalb des Blocks ausgeführt



: ist die angegebene Bedingung *wahr*, so werden die Blöcke innerhalb des Blocks ausgeführt



: ist die angegebene Bedingung *wahr*, so werden die Blöcke innerhalb des oberen Teils des Blocks ausgeführt, ist die Bedingung *falsch*, so werden die Blöcke innerhalb des unteren Teils des Blocks ausgeführt



: wartet bis die angegebene Bedingung *wahr* ist, dann werden die Blöcke danach

ausgeführt



: kontrolliert, ob die angegeben Bedingung *falsch* ist. Ist dies der Fall, werden die Blöcke innerhalb des Blocks ausgeführt und es wird wieder die Bedingung kontrolliert; ist die Bedingung *wahr*, so wird mit dem Block unterhalb des Blocks weitergemacht



: stoppt dieses Programm



: stoppt die Programme aller Sprites

### 4.3.11 Fühlen



: gibt *wahr* zurück, wenn das ausgewählte Sprite, den Bühnenrand oder den Mauszeiger berührt (unter dem Pfeil aus dem Kontextmenü auswählen)



: gibt *wahr* zurück, wenn das Sprite die ausgewählte Farbe berührt (die Farbe mit der Farbpipette auswählen)



: gibt *wahr* zurück, wenn die erste Farbe (eine Farbe des Sprites) die zweite Farbe (eine Farbe des Hintergrunds oder eines anderen Sprites) berührt (die Farben mit der Farbpipette auswählen)



: lässt das Sprite eine Frage stellen und speichert die Tastatureingabe in *Antwort* ; das Programm wird erst weiter ausgeführt, wenn die Enter-Taste oder das Häkchen gedrückt wird



: Speichert die Tastatureingabe der letzten Frage und steht allen Sprites zur Verfügung



: gibt die x-Position des Mauszeigers an



: gibt die y-Position des Mauszeigers an



: gibt *wahr* an, wenn die Maustaste gedrückt wird



: gibt *wahr* an, wenn die ausgewählte Taste gedrückt wird



: gibt die Entfernung zum ausgewählten Sprite oder zum Mauszeiger an



: setzt die Stoppuhr auf 0



: gibt den Wert der Stoppuhr zurück (die Stoppuhr läuft immer)



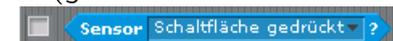
: gibt den ausgewählten Wert des ausgewählten Objekts an



: gibt die Lautstärke (von 1 bis 100) des angeschlossenen Mikrofons an



: gibt *wahr* zurück, wenn die Lautstärke des angeschlossenen Mikrofons größer als 30 (gemessen mit einer Skala von 1 bis 100) ist



: gibt den Wert des ausgewählten Sensors an; um diesen Block nutzen zu können, brauchst du einen an deinen Computer angeschlossenen Sensor; du kannst diese Blöcke mit dem Scratch-Sensor-Board

(<http://www.playfulinvention.com/picoboard.html>) oder mit LEGO WeDo

(<http://www.legoeducation.com>) benutzen



: gibt *wahr* an, wenn der ausgewählte Sensor gedrückt/ verbunden

ist; um diesen Block nutzen zu können, brauchst du ein an deinen Computer angeschlossenes Scratch-Sensor-Board (<http://www.playfulinvention.com/picoboard.html>)

### 4.3.12 Operatoren

- : addiert zwei Zahlen
- : subtrahiert die zweite Zahl von der ersten
- : multipliziert zwei Zahlen
- : dividiert die erste Zahl durch die zweite Zahl
- : wählt eine Zufallszahl aus dem angegebenen Bereich
- : ergibt *wahr*, wenn die erste Zahl kleiner ist als die zweite
- : ergibt *wahr*, wenn beide Zahlen identisch sind
- : ergibt *wahr*, wenn die erste Zahl größer ist als die zweite
- : ergibt *wahr*, wenn beide Bedingungen erfüllt sind
- : ergibt *wahr*, wenn mindestens eine der beiden Bedingungen erfüllt ist
- : ergibt *wahr*, wenn die Bedingung nicht erfüllt ist und *falsch*, wenn die Bedingung erfüllt ist
- : verbindet zwei Zeichenketten miteinander
- : gibt das angegebene Zeichen der ausgewählten Zeichenkette zurück
- : gibt die Anzahl der Buchstaben der ausgewählten Zeichenkette zurück
- : gibt den Rest der Division der ersten Zahl durch die zweite Zahl zurück
- : gibt den Wert der nächsten ganzen Zahl zurück
- : gibt das Resultat der ausgewählten Funktion (Betrag, Wurzel, sin, cos, tan, asin, acos, atan, ln, log, e<sup>^</sup>, 10<sup>^</sup>) mit der angegebenen Zahl zurück

### 4.3.13 Variablen

- : erstellt und benennt eine neue Variable; wenn du eine neue Variable erstellst, erscheinen die Blöcke für die Variable; du kannst eine Variable für alle Sprites erstellen (global) oder nur für ein Sprite (lokal)
- : löscht alle Blöcke, die mit der Variable verbunden sind
- : gibt den Wert der Variable zurück
- : setzt den Wert der ausgewählten Variable auf den angegebenen Wert (bei mehr als einer Variable kannst du die Variable aus dem Kontextmenü unter dem Pfeil wählen)
- : ändert den Wert der ausgewählten Variable um den angegebenen Wert
- : zeigt die Anzeige der ausgewählten Variable auf der Bühne
- : versteckt die Anzeige der ausgewählten Variable derart, dass sie

nicht auf der Bühne zu sehen ist

**Neue Liste**: erstellt und benennt eine neue Liste; wenn du eine neue Liste erstellst, erscheinen die Blöcke für die Liste; du kannst eine Liste für alle Sprites erstellen (global) oder nur für ein Sprite (lokal)

**Liste löschen**: löscht alle Blöcke in der Blockpalette, die mit der Liste verbunden sind

**Liste**: gibt alle Elemente der Liste zurück

**füge etwas zu Liste hinzu**: fügt das angegebene Element am Ende der ausgewählten Liste hinzu; das Element kann eine Zahl, eine Zeichenkette aus Buchstaben oder anderen Zeichen sein

**entferne 1 aus Liste**: löscht ein Element oder alle Elemente aus der Liste; du kannst ein Element aus dem Kontextmenü unter dem Pfeil oder die Zahl des Element wählen, welches dann gelöscht wird; wählst du *letztes*, so wird das letzte Element der Liste gelöscht; wählst du *alles*, so werden alle Elemente der Liste gelöscht; löschen verringert die Länge der Liste

**füge etwas als 1 in Liste ein**: fügt das angegebene Element an der ausgewählten Position der ausgewählten Liste hinzu; du kannst die Stelle, in dem das Element eingefügt werden soll aus dem Kontextmenü unter dem Pfeil oder als Zahl wählen; wählst du *letztes*, so wird das Element am Ende der Liste hinzugefügt; wählst du *beliebiges*, so wird das Element an einer zufälligen Stelle der Liste hinzugefügt; die Länge der Liste wird um 1 erhöht

**ersetze 1 Element von Liste durch etwas**: ersetzt das Element an der ausgewählten Stelle der ausgewählten Liste durch das angegebene Element; du kannst die Stelle, an dem das Element ersetzt werden soll, aus dem Kontextmenü unter dem Pfeil oder als Zahl wählen; wählst du *letztes*, so wird das Element am Ende der Liste ersetzt; wählst du *beliebiges*, so wird das Element an einer zufälligen Stelle der Liste ersetzt; die Länge der Liste wird nicht verändert

**1 Element von Liste**: gibt das ausgewählte Element der ausgewählten Liste zurück; du kannst die Stelle, an dem das Element zurück gegeben werden soll aus dem Kontextmenü unter dem Pfeil oder als Zahl wählen; wählst du *letztes*, so wird das Element am Ende der Liste zurück gegeben; wählst du *beliebiges*, so wird das Element an einer zufälligen Stelle der Liste zurück gegeben

**Länge von Liste**: gibt zurück wie viele Elemente in der ausgewählten Liste sind (Länge der Liste)

**Liste enthält etwas**: gibt *wahr* zurück, wenn die Liste das angegebene Element enthält (das Element muss genau angegeben werden, damit die Ausgabe *wahr* erfolgt)

## 4.4 Anhang

### 4.4.1 SYSTEM-ANFORDERUNGEN FÜR DIE SCRATCH-VERSION 1.4

Computer Betriebssystem: Windows XP, Windows 2000, Windows Vista

Mac Betriebssystem: X 10.4 oder später

Display: 800x480 oder größer, thousands or millions of colors (16-bit color oder größer).

Disk: mindestens 120 megabytes freier Speicher für die Installation von Scratch

Anmerkung: Scratch hat eine große Medienbibliothek und eine Sammlung an Beispielprojekten. Falls du nur begrenzten Speicherplatz zur Verfügung hast, kannst du die Medien und Projekte Ordner aus dem Scratchordner löschen.

Memory: die meisten Computer haben genug Arbeitsspeicher zur Verfügung um Scratch auszuführen. Ältere Computer können Scratch langsamer ausführen.

Sound: um die Klänge nutzen zu können, brauchst du einen Lautsprecher (oder Kopfhörer) und ein Mikrofon. In vielen Laptops sind Lautsprecher und Mikrofon schon integriert.

#### **4.4.2 NETZWERK INSTALLATION**

Für mehr Informationen und Updates zur Netzwerkinstallation, lese die readme.txt Datei oder siehe: [http://info.scratch.mit.edu/Network\\_Installation](http://info.scratch.mit.edu/Network_Installation).

# Kapitel 5

## Hilfen

### 5.1 Hilfsmaterialien und Linksammlung

#### 5.1.1 Uniwebseiten und E-Mailadressen

Arbeitsgruppe Theoretische Informatik: <http://www.ti.informatik.uni-kiel.de/>  
Skript, Forum, andere Hilfsmaterialien: unter <https://www.uni-kiel.de/lms/dmz/> einloggen, unter Schnellstartlinks *Kurse* anklicken, und den *Scratch-Workshop April 2010* suchen und anklicken (<http://www.uni-kiel.de:80/lms/auth/repo/go?rid=9404532>)

E-Mailadressen:

Tomma Radtke: [radtke@ti.informatik.uni-kiel.de](mailto:radtke@ti.informatik.uni-kiel.de)

Sönke Schulmeister: [schulmeister@ti.informatik.uni-kiel.de](mailto:schulmeister@ti.informatik.uni-kiel.de)

#### 5.1.2 Scratchwebseite

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

Hilfen auf der Scratchwebseite:

<http://info.scratch.mit.edu/Support> (Getting Started Guide, Scratch Reference Guide, Getting Started Page, Scratch Cards und FAQ's: alles auf Englisch)

und <http://info.scratch.mit.edu/Languages> (Getting Started Guide, Karten, Referenzhandbuch für eine ältere Version und Hilfebilder zum Einbinden in Scratch: alles auf Deutsch)

Um die deutschen Hilfebilder einzubinden, laden Sie diese runter und entpacken den Ordner. Den entpackten Ordner de müssen Sie nun kopieren und in den Help-Ordner der Scratch-Installation einbinden.

#### 5.1.3 Bilder und Soundeffekte

freie Soundeffekte: <http://www.freesfx.co.uk/>

Musikstücke: <http://free-loops.com/>

Bilder: <http://www.flickr.com/search/advanced/?> Häkchen setzen bei: Nur in Inhalten mit einer Creative Commons-Lizenz suchen.

### 5.1.4 Bildbearbeitungsprogramm

Bildbearbeitungsprogramm: <http://www.gimp.org/downloads/>

### 5.1.5 Audiotbearbeitungsprogramm

Audiotbearbeitungs- und Aufnahmeprogramm: <http://audacity.sourceforge.net/>

### 5.1.6 Videotutorials

Die Videotutorials kann man im deutschen Scratchforum unter dem Stichwort *MCS Juniorakademie-Thread* finden

(<http://scratch.mit.edu/forums/viewtopic.php?id=11836>).

Einstieg in Scratch: <http://screencast.com/t/qQCi80wu>

Installation und Oberfläche: <http://screencast.com/t/6dzTLzrJ>

Pacman: <http://screencast.com/t/vh7a7UbE>

Bewegungsblöcke: <http://screencast.com/t/kmtCwSp4Rpi>

Klangblöcke: <http://screencast.com/t/RmrYmKc2jzi>

Living creatures: <http://screencast.com/t/XypLzX5U6Q>