



ALGORITHMEN

HINTERGRUND ZUM UNTERRICHTSGANG

Dieses Werk ist unter einem **Creative Commons 3.0 Deutschland Lizenzvertrag** lizenziert:

- Namensnennung
- Keine kommerzielle Nutzung
- Weitergabe unter gleichen Bedingungen



Um die Lizenz anzusehen, gehen Sie bitte zu <https://creativecommons.org/licenses/by-nc-sa/3.0/de>.

Michael Schmitt – E-Mail: [schmitt \[at\] copernicus-gymnasium.de](mailto:schmitt[at]copernicus-gymnasium.de). – Januar 2017

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>.
Screenshots von Scratch und Scratch-Dateien sind lizenziert unter **CC BY-SA 2.0**
(<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).



Inhaltsverzeichnis

1 Algorithmus und Programm.....	3
2 Visuelle Programmierung.....	3
3 Programmieren mit Scratch.....	4
4 Scratch im Internet.....	5
5 Scratch-Versionen.....	5
6 Hinweise zum Unterrichtsgang.....	7
7 Struktogramme.....	11
8 Bundesweite Informatikwettbewerbe.....	13
Anhang: Scratch 1.4 vs. Scratch 2.0.....	14



1 Algorithmus und Programm

Der Algorithmus ist eine der Grundideen der Informatik. Bevor ein Computer zum Bearbeiten einer Problemstellung verwendet werden kann, muss er vom Menschen in die Lage versetzt werden, das Problem zu lösen, d.h., er muss programmiert werden. Ein Programm ist ein Text, der in eindeutiger Weise beschreibt, welche Verarbeitungsschritte der Computer in welcher Reihenfolge auszuführen hat. Das Programm ist in einer Programmiersprache geschrieben, die die Syntax, also die Zusammensetzung der verwendeten Zeichen und Wörter, und die Semantik, also die Bedeutung der Programmbestandteile, präzise festlegt.

Bevor ein Programmtext in einer programmiersprachlichen Formulierung entsteht, hat der Programmierer zumeist schon eine Idee von den wesentlichen Verarbeitungsschritten, die die Maschine später ausführen soll. Diese Idee vor und zu einem Programm ist der Algorithmus. Er beschreibt, wie ein gegebenes Problem gelöst werden kann, und zwar in Schritten, die so eindeutig und präzise sind, dass sie in jede denkbare hinreichend mächtige Programmiersprache übertragen und letztendlich von einer Maschine ausgeführt werden können.

Ein Algorithmus, als Bindeglied zwischen der vom Menschen ersonnenen Lösungsidee und der Maschine, wird in der Regel zunächst nicht in einer Programmiersprache formuliert. Zum Notieren von Algorithmen wurden in der Informatik eigene Darstellungsformen entwickelt. Sie verwenden Bausteine, die sich in den Programmiersprachen wiederfinden. Zu diesen Bausteinen gehören die elementare Anweisung, die Sequenz, die Schleife oder Wiederholung, die Verzweigung und die Bedingung. Gängige Notationsformen für Algorithmen sind beispielsweise Pseudocode, Flussdiagramm und Struktogramm.

2 Visuelle Programmierung

In der vorliegenden Unterrichtseinheit lernen die Schülerinnen und Schüler die algorithmischen Grundbausteine kennen und verwenden sie, indem sie Algorithmen entwerfen und in ausführbare Programme umsetzen. In der Klassenstufe 7 haben sie damit zum ersten Mal Kontakt mit dem Computer auf eine Weise, bei der sie die Geräte nicht nur verwenden, sondern gestalterisch auf sie Einfluss nehmen. Zieht man die oft verwendete Analogie des Algorithmus zum Kochrezept heran, so wurden bisher im Unterricht Speisen nur verzehrt, mit der Programmierung aber werden neue Rezepte entworfen und vielleicht sogar noch unbekannte Gerichte erschaffen.

Im Informatikunterricht hat sich im Lauf der Zeit gezeigt, dass die im praktischen Einsatz verwendeten höheren Programmiersprachen wie z.B. C++, Java oder Python für den Anfangsunterricht ungeeignet sind. Sie sind textorientiert, d.h., die Anweisungen und Kontrollstrukturen müssen Zeichen für Zeichen nach genau vorgegebenen Regeln in einen Editor eingegeben werden. Dies führt gewöhnlich dazu, dass die Schülerinnen und Schüler an Problemen mit der Syntax hängen bleiben und vor dem Erreichen des eigentlichen Ziels, der Fertigstellung und Ausführung des Programms, frustriert aufgeben.

Mit der Weiterentwicklung der Informatik ist eine Reihe von Programmiersprachen entstanden, die die Syntaxprobleme textueller Sprachen vermeiden. Man bezeichnet sie als visuelle Pro-



grammiersprachen.¹ In jüngster Zeit hat diese Entwicklung auch die Programmiersprachen für den Informatikunterricht erfasst. Es entstand eine Gruppe von visuellen Programmiersprachen für den Unterricht.² Zu diesen Sprachen gehört Scratch, ein Projekt der Lifelong-Kindergarten-Gruppe am MIT Media Lab des Massachusetts Institute of Technology. Es ist kostenlos unter <https://scratch.mit.edu> verfügbar.

3 Programmieren mit Scratch

Scratch ist insofern schülergerecht, als jedes Programm ausführbar ist, d.h., es gibt als Programmierfehler keine Syntaxfehler, sondern nur logische Fehler (z.B. eine Bedingung, die unbeabsichtigt zu einer Endlosschleife führt) oder Laufzeitfehler (z.B. eine Eingabe, die eine Division durch null zur Folge hat). Erreicht wird dies im Wesentlichen dadurch, dass bei der Programmierung lediglich vorgefertigte Blöcke aneinandergefügt oder ineinander verschachtelt werden müssen. Nur die vorgesehenen Felder für Parameterwerte sind ggf. durch Tastatureingaben abzuändern.



Ein Scratch-Programmabschnitt

Scratch ist nicht nur eine Programmiersprache, sondern auch eine Programmierumgebung (engl. *integrated development environment*, IDE), d.h., es stellt neben einem Editor für die Programme (die sogenannten Skripte) weitere Werkzeuge bereit, die über das bloße Erstellen und Ausführen des Programmcodes hinausgehen. Beispielsweise können Programmabschnitte zum Testen sofort ausgeführt werden. Die Kostüme von Objekten (Spielfiguren, engl. *sprites*) können aus einer Figurenbibliothek ausgewählt, mit einem integrierten Malprogramm neu gezeichnet oder mit einer angeschlossenen Webcam fotografiert werden. Klänge lassen sich aus einer Klangbibliothek abrufen oder mit dem Mikrofon neu aufnehmen. Fertige Projekte können als Video im flv-Format auf dem Rechner gespeichert werden. Auf der Scratch-Webseite angemeldete Benutzer können ihre Projekte dort veröffentlichen und so der weltweiten Scratch-Community zugänglich machen.

In Schulen wird Scratch auch häufig zum Erstellen von Bildsequenzen oder Spielszenen eingesetzt. Auf diese sogenannten Storyboards oder Skizzenbücher gehen auch die Dateiendungen gespeicherter Scratch-Projekte zurück (.sb bzw. .sb2). Diese Skripte weisen häufig einen linearen

¹ siehe z.B. Seite „Visuelle Programmiersprache“. URL:

https://de.wikipedia.org/wiki/Visuelle_Programmiersprache (Abgerufen: 30. Dezember 2016)

² siehe Seite „Erziehungsorientierte Programmiersprachen“. URL:

https://de.wikipedia.org/wiki/Erziehungsorientierte_Programmiersprachen. (Abgerufen: 30. Dezember 2016)



ren Programmablauf auf, der fast ohne Schleifen und Verzweigungen auskommt. Auch von der Möglichkeit, Kontrollstrukturen ineinander zu schachteln, muss dabei nicht unbedingt Gebrauch gemacht werden. Die ausschließliche „Programmierung“ von Storyboards eignet sich daher weniger für den Informatikunterricht.

4 Scratch im Internet

Die Internetseite von Scratch, <https://scratch.mit.edu>, hält das umfassende Angebot des MIT Media Lab zu Scratch in drei Themenbereichen bereit: *Create – Explore – Discuss* (Entwickeln – Entdecken – Diskutieren). Unter „Entwickeln“ öffnet sich der Online-Editor von Scratch, mit dem Programme direkt im Browser erstellt und ausgeführt werden können. (Mehr zu den Möglichkeiten der Programmierung und den Scratch-Versionen weiter unten.) Unter „Entdecken“ finden sich Tausende von Programmierprojekten, die von Scratchern weltweit entwickelt und hochgeladen wurden. Diese Programme lassen sich direkt im Browser ausführen. Ihr Quelltext ist einsehbar und kann sogar heruntergeladen oder verändert werden. Ein Grundgedanke von Scratch ist das sogenannte „Remixen“. Darunter versteht man das Teilen eines Programms mit anderen, damit es verändert und wieder geteilt werden kann. Unter „Diskutieren“ schließlich kann man sich mit anderen Scratchern in einem Online-Forum, auch in Deutsch und weiteren Sprachen, austauschen.

Ein Wiki in englischer Sprache bietet die Seite <https://wiki.scratch.mit.edu>. Das Wiki der deutschsprachigen DACH-Scratch-Gemeinschaft findet sich unter <http://scratch-dach.info/wiki>. Didaktische Hinweise für Lehrkräfte stellt ScratchEd bereit, die Webseite einer Online-Gemeinschaft von englischsprachigen Scratch-Pädagogen. Die Seite ist bei der Harvard University unter <http://scratched.gse.harvard.edu> erreichbar. Sie präsentiert zahlreiche Einsatzszenarien, Materialien und Veranstaltungen rund um das Thema „Programmieren Lernen mit Scratch“.

Für den Unterricht werden vom MIT Media Lab sogenannte Scratch-Karten zum Herunterladen angeboten. Dies sind kurze Anleitungen, die ausgedruckt, geklebt und an die Schülerinnen und Schüler ausgeteilt werden. Anhand dieser Karten lernen sie selbstständig zu programmieren, indem sie kleine Aufgabenstellungen bearbeiten, die sie später in eigene Projekte einbauen können. Die Vorderseite einer solchen Karte zeigt die Aufgabe, die Rückseite einen Lösungsvorschlag. Beispielsweise gibt es Scratch-Karten zum Animieren von Objekten, zum Programmieren eines Wettrennens oder zur Entwicklung eines Pong-Spiels. Die englischsprachigen Karten sind erhältlich unter <https://scratch.mit.edu/info/cards>. Scratch-Karten auf deutsch findet man beispielsweise auf der Seite <http://www.brandhofer.cc/scratchcard>. Diese Karten beinhalten allerdings nur elementare Bausteine wie etwa das Ändern der Farbe, die Tastensteuerung oder das Verwenden eines Zählers.

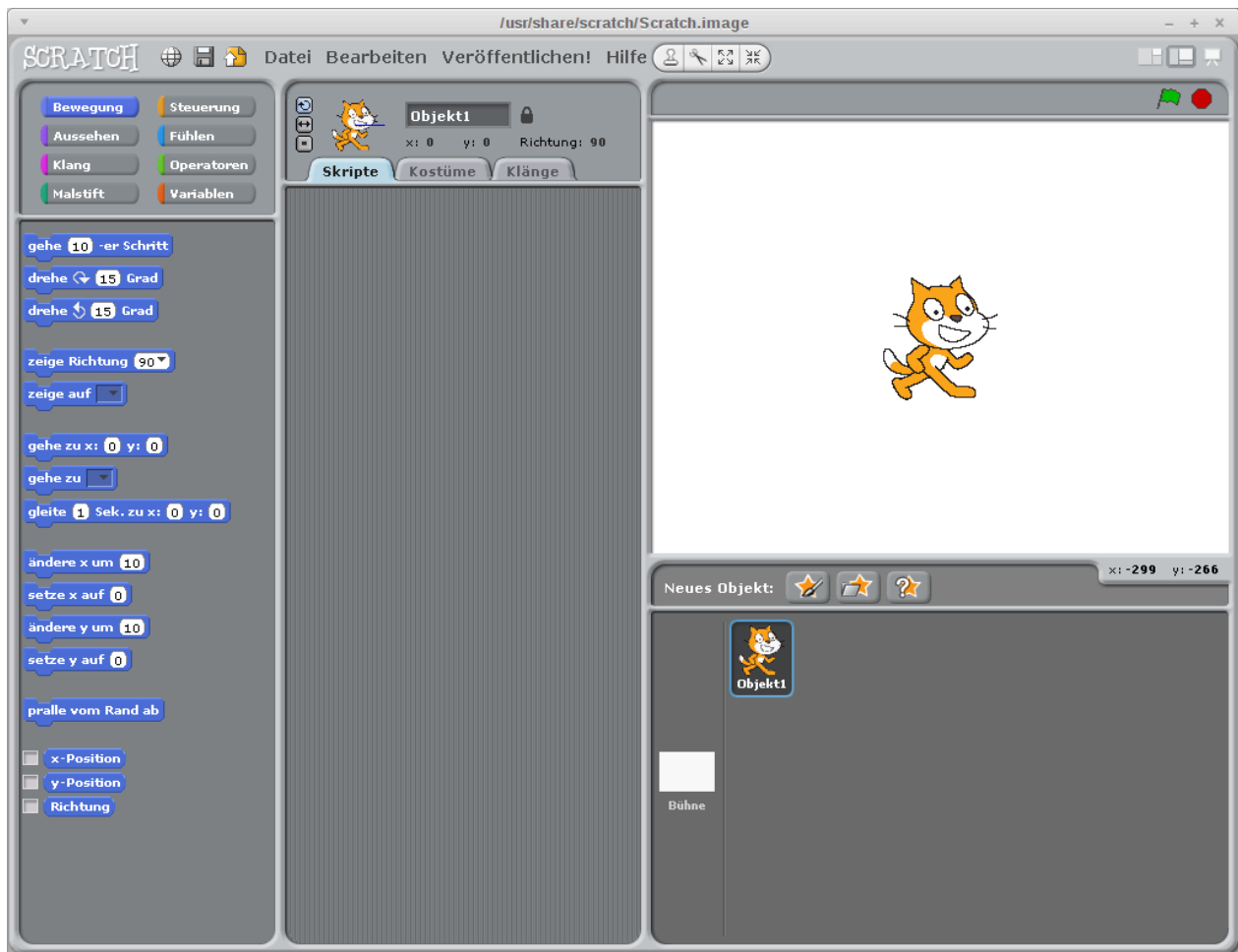
5 Scratch-Versionen

Wie bereits erwähnt, kann mit Scratch in jedem modernen Browser programmiert werden. Dies geht auch ohne Einrichtung eines Benutzerkontos. Ein Konto ist nur erforderlich, wenn Projekte auf der Scratch-Homepage veröffentlicht werden sollen. Allerdings benötigt man für den Browser das Adobe Flash Plugin und natürlich einen Internet-Zugang.

Für den Unterricht bei nicht vorhandenem oder gesperrtem Internet können Offline-Editoren genutzt werden, die für jedes gängige Betriebssystem zur Verfügung stehen. Zurzeit sind zwei ver-



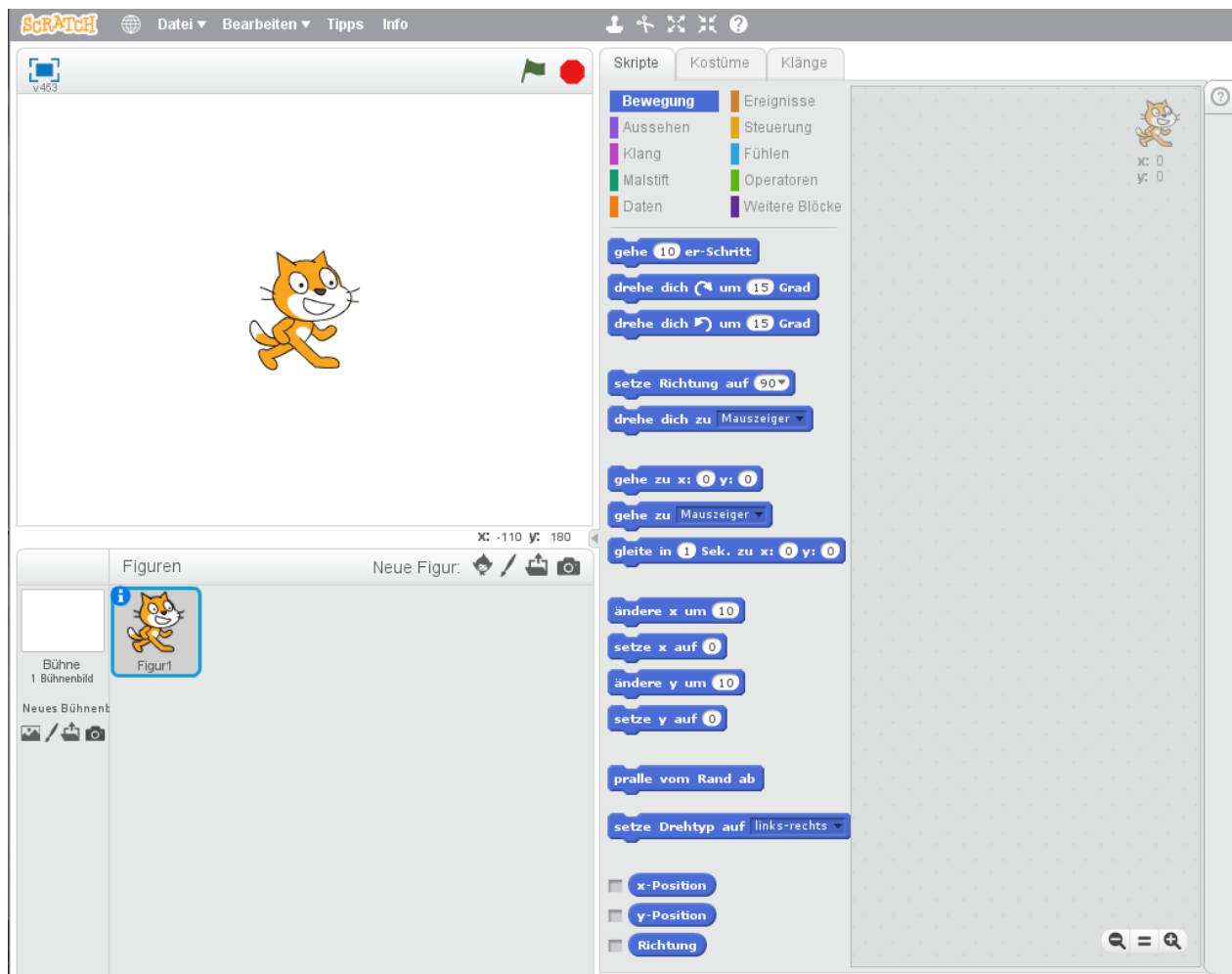
schiedene Versionen erhältlich: Version 1.4 (https://scratch.mit.edu/scratch_1.4), im Juli 2009 veröffentlicht, und die neuere Version 2.0 (<https://scratch.mit.edu/scratch2download>) vom Mai 2013. Das Erscheinen einer Version 3.0 ist nicht vor Ende 2017 angekündigt. Version 1.4 lässt sich einfacher installieren als Version 2.0, die zusätzlich die Installation von Adobe Integrated Runtime (AIR) voraussetzt.³



Die Programmierumgebung von Scratch 1.4

Die Verwendung der Version 1.4 ist für diesen Unterrichtsgang völlig ausreichend. Alle Programmieraufgaben lassen sich damit lösen. Sollte an der Schule Scratch 1.4 vorhanden sein, ist ein Umstieg auf Version 2.0 also nicht nötig. Schwierigkeiten können höchstens dann entstehen, wenn die Schülerinnen und Schüler Scratch auch zu Hause nutzen. Mit Scratch 1.4 erstellte Programme laufen zwar unter Scratch 2.0, umgekehrt geht dies jedoch nicht. Die Programmieroberflächen ähneln sich und bieten im Wesentlichen dieselben Funktionen. Ein Unterschied besteht in der Anordnung der einzelnen Fensterausschnitte.

³ In unregelmäßigen Abständen verlangt Scratch bzw. Adobe AIR eine Programmaktualisierung. Dies ist im Unterricht störend, es lässt sich aber derzeit nicht vermeiden, dass eine entsprechende Meldung erscheint. Sie kann ignoriert werden.



Die Programmierumgebung von Scratch 2.0 (offline und online)

Version 2.0 enthält mehr Sprachelemente als 1.4, so gibt es z.B. Anweisungen zum Klonen von Objekten oder die Möglichkeit, Blöcke neu zu definieren, die dann sogar Rekursion enthalten können. Diese Besonderheiten liegen außerhalb des Bildungsplans für Klassenstufe 7. Für die Dokumentationen und Arbeitsblätter dieses Unterrichtsgangs wurden ausschließlich Bildschirmfotos der Version 2.0 verwendet. Im Anhang dieses Dokuments findet sich eine Zusammenstellung der wichtigsten Unterschiede von Scratch 1.4 und 2.0.

6 Hinweise zum Unterrichtsgang

In der heutigen Zeit sind Schülerinnen und Schüler jeden Alters begeistert von Computerspielen. Dieser Unterrichtsgang baut auf diesem Interesse auf und möchte die Schülerinnen und Schüler in die Situation eines Spieleentwicklers versetzen. Die Wahl fiel dabei auf ein Spiel in Anlehnung zu dem Handyspiel *Crossy Road*⁴, das im Jahr 2014 erschien und der Kategorie „Jump 'n' Run“ zugeordnet wird. In den achtziger Jahren gab es eine frühere Variante mit Namen *Frogger*⁵.

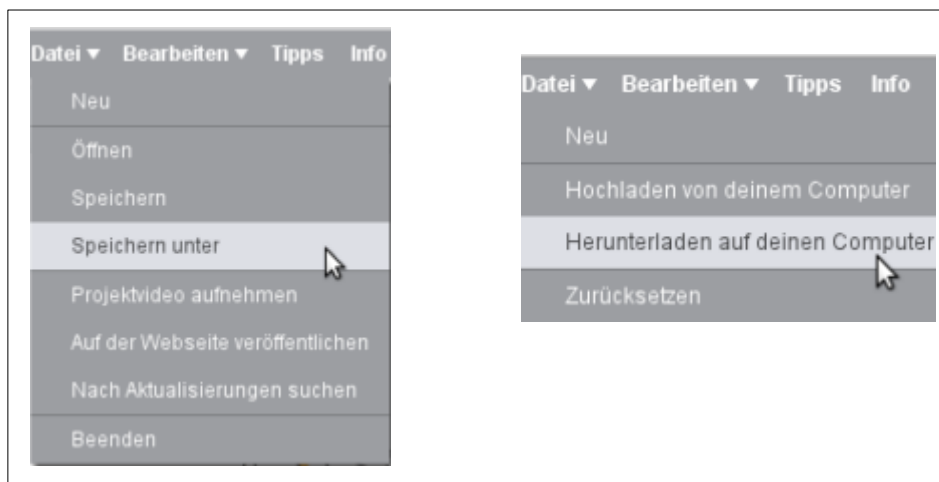
4 Seite „Crossy Road“. URL: https://de.wikipedia.org/wiki/Crossy_Road. (Abgerufen: 2. Januar 2017)

5 Seite „Frogger“. URL: <https://de.wikipedia.org/wiki/Frogger>. (Abgerufen: 2. Januar 2017)



Es geht darum, eine Spielfigur über eine stark befahrene Straße zu führen oder sie einen Fluss auf vorbeischwimmenden Baustämmen überqueren zu lassen. Sobald die Figur von einem Auto „angefahren“ wird oder im Fluss „ertrinkt“, endet das Spiel. In Scratch wird dazu die Katze mit den Pfeiltasten gesteuert. Fahrzeuge und Baumstämme sind jeweils eigene Objekte, die sich von selbst, auch mit veränderbarer Geschwindigkeit, bewegen. Für die Bewegung und Steuerung kommen so alle algorithmischen Grundbausteine zum Zuge; Variablen werden zur Speicherung von Fahrzeuggeschwindigkeiten und Punktstand eingesetzt. Das Spiel lässt genügend Raum, um es mit eigenen Ideen zu erweitern.

Die Schülerinnen und Schüler erstellen ihr Projekt anhand von Arbeitsblättern, die sie sich der Reihe nach abholen. Mithilfe eines Laufzettels behalten sie den Überblick über die bearbeiteten Aufträge. Zu Beginn sollte der Lehrer sicher stellen, dass die Schülerinnen und Schüler die Programme wiederauffindbar abspeichern können. Hierbei gibt es einen Unterschied im Datei-Menü zwischen dem Offline-Editor (in der Abbildung links) und der Online-Version.



Speichern von Programmen offline bzw. online

Der Unterrichtsgang erstreckt sich über 12 Unterrichtsstunden und ist gegliedert in die Teile A bis H. Teil A beginnt mit dem Kennenlernen der Programmierungsumgebung. Teil B stellt einfache Aufgaben zur Bewegung der Spielfigur mit einzelnen Anweisungen und Sequenzen. In Teil C soll die Figur mit den Pfeiltasten gesteuert werden. In Scratch gibt es dazu zwei prinzipiell verschiedene Möglichkeiten: Ob eine Taste gedrückt wurde, kann mithilfe eines Ereignisses oder einer Bedingung in einer Schleife festgestellt werden. Bei der Steuerung durch Ereignisse ist jeder Taste ein eigenes Skript zugeordnet. Im anderen Fall werden die Tasten in einer Endloschleife mithilfe einer Verzweigung und einer Bedingung aus dem Blockbereich „Fühlen“ abgefragt. Die Abbildung stellt die beiden Varianten einander gegenüber. Beide Lösungen bewirken dasselbe, es gibt jedoch einen spürbaren technischen Unterschied bei der Ausführung: Die Ereignissteuerung reagiert mit leichter Verzögerung im Vergleich zur Abfrageschleife. Letztere kann natürlich erst dann eingesetzt werden, wenn Schleife, Verzweigung und Bedingung bekannt sind. In diesem Unterrichtsgang ist dies in Teil G zum Ändern der Geschwindigkeit vorgesehen. Als freiwillige Zusatzaufgabe können danach die alternativen Methoden zur Steuerung der Figur bzw. der Geschwindigkeit ausprobiert werden.



Bewegungssteuerung durch Ereignisse bzw. Schleife

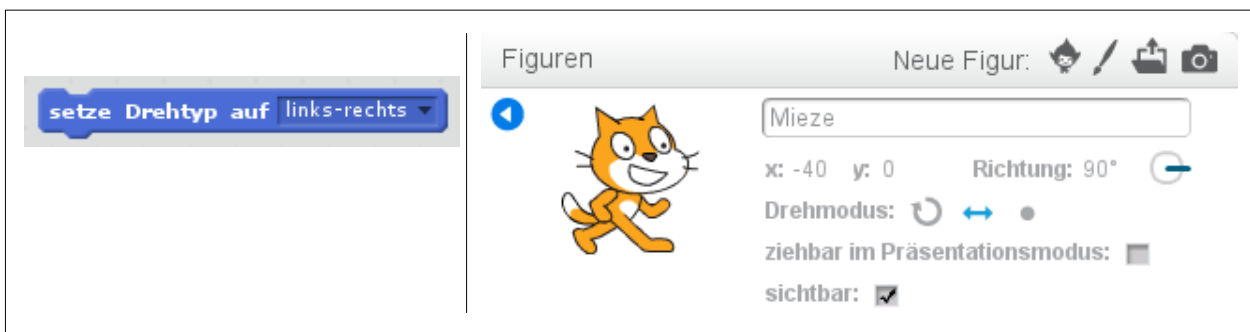
Bewegt man die Katze mit der Pfeiltaste nach links, läuft sie zunächst rückwärts. Damit die Figur immer in die Richtung blickt, in die sie sich bewegt, wird in Teil C ein neues Kostüm erstellt. Dies geschieht im Zeichenprogramm für Kostüme mit dem Knopf „Links und Rechts vertauschen“. In der Tastensteuerung muss dann der Kostümwechsel etwa wie folgt eingebaut werden.

Änderung der Blickrichtung mit Kostümwechsel

Alternativ kann man das Originalkostüm beibehalten, wenn man die Richtung der Figur mit den entsprechenden Anweisungen setzt.

Änderung der Blickrichtung ohne Kostümwechsel

Damit die Figur beim nach Links gehen dann nicht auf dem Kopf steht, muss man noch ihren Drehmodus einstellen. Dazu lässt man die Anweisung zum Setzen des Drehtyps im Skript der Figur einmal ausführen. Dies geht allerdings nicht in Scratch 1.4. Man kann stattdessen den Drehmodus auch manuell in den Eigenschaften der Figur ändern.



Methoden zum Setzen des Drehtyps

Teil D behandelt die Gestaltung der Bühne durch Zeichnen einer Straße und das Anlegen neuer Objekte in Form eines Autos. Dies wird in Teil E fortgeführt, wo das Auto so programmiert wird, dass es sich von selbst über die Bühne bewegt. Dazu werden Schleifen eingeführt.



Die Schleifen von Scratch

In Scratch gibt es drei Arten von Wiederholungen: Zählschleife, Endlosschleife und Schleife mit Bedingung.⁶ Die dritte Variante überprüft zuerst, ob die Bedingung zutrifft. Falls ja, wird das Innere der Schleife (der sogenannte Schleifenrumpf) nicht ausgeführt und die Schleife ist damit beendet. Damit unterscheidet sich diese Scratch-Schleife von der in vielen Programmiersprachen vorhandenen `while`-Schleife. Diese hat die Form „`while Bedingung do Schleifenrumpf`“, d.h., solange die Bedingung zutrifft, wird der Schleifenrumpf wiederholt. Die Scratch-Schleife mit Bedingung als `while`-Schleife formuliert lautet also `while not(Bedingung) do Schleifenrumpf`.

Zur weiteren Steuerung des selbstfahrenden Autos wird in Teil E auch die Verzweigung eingeführt, die es in Scratch, wie in den meisten Programmiersprachen, ohne und mit Alternative gibt. Mit der Verzweigung soll bewirkt werden, dass das Auto nach Erreichen des Randes verschwindet und auf der anderen Seite der Bühne wieder auftaucht. Teil E enthält als Einschub ein Arbeitsblatt (Teil F) mit Aufgaben, in denen die Anweisungsblöcke für den Malstift verwendet werden, um geometrische Figuren zeichnen zu lassen. Mit diesen Aufgaben wird auch die Fähigkeit gefördert, Programme zu lesen und gedanklich nachzuvollziehen.



Verzweigung ohne und mit Alternative

⁶ Die in Scratch 1.4 vorhandene Endlosschleife mit Bedingung „wiederhole fortlaufend, falls ...“ gibt es in Version 2.0 nicht mehr.



Teil G schließlich ist der Verwendung von Variablen gewidmet. Die Geschwindigkeit des Autos soll in einer Variable gespeichert und mittels Tastatureingabe erhöht bzw. erniedrigt werden. Damit die Geschwindigkeit bestimmte Werte nicht über- bzw. unterschreitet, soll die Änderung nicht immer wirksam werden. Dazu werden die logischen Operatoren eingeführt. Sie sind in einer Tastensteuerung mit Endlosschleife zu verwenden.

Mit Teil H entsteht die erste Stufe des fertigen Spiels. Wenn die Katze das Auto berührt, soll es stehen bleiben. Es folgen weitere Aufgaben zur Erweiterung mit unterschiedlichem Schwierigkeitsgrad. Sie können unabhängig voneinander bearbeitet werden.

In diesem Unterrichtsgang nicht verwendet wurden die Anweisungen aus dem Blockbereich „Klang“. Er bietet natürlich weitere Mittel, das Spiel interessanter zu machen. Meistens kommen die Schülerinnen und Schüler von selbst auf die Idee, Klänge einzusetzen. Es empfiehlt sich allerdings, eine intensive Nutzung von Klängen nur zu gestatten, wenn alle Computer mit Kopfhörern ausgestattet sind.


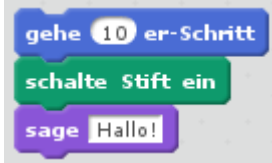

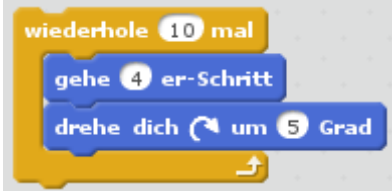

7 Struktogramme

Struktogramme sind eine Möglichkeit, Algorithmen unabhängig von einer Programmiersprache aufzuschreiben. Sie werden nach ihren Entwicklern auch Nassi-Shneiderman-Diagramme genannt. In Deutschland sind sie genormt in der DIN 66261.

Struktogramme sind Veranschaulichungen von Algorithmen mittels einfacher geometrischer Formen, deren Grundbaustein das Rechteck darstellt. Jedes Rechteck ist mit einer elementaren Anweisung beschriftet oder es stellt eine Kontrollstruktur wie z.B. eine Schleife oder eine Verzweigung dar. Rechtecke können aufeinander gestapelt und ineinander geschachtelt werden. Diese Konstruktionsprinzipien findet man in vielen Programmiersprachen wieder, sodass die Notation eines Algorithmus als Struktogramm einen hilfreichen Zwischenschritt auf dem Weg vom Problem zum Programm darstellt.

Die folgende Tabelle enthält Beispiele für Struktogrammbausteine, die sich für die Programmierung in Scratch anbieten. Daneben ist ihre Entsprechung in Scratch-Blöcken dargestellt.



Algorithmischer Grundbaustein	Struktogramm	Scratch-Programm
Anweisung	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> gehe 10er-Schritt </div>	
Sequenz	<div style="border: 1px solid black; padding: 5px;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">gehe 10er-Schritt</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">schalte Stift ein</div> <div style="border: 1px solid black; padding: 2px;">sage „Hallo!“</div> </div>	
Schleife mit Bedingung	<div style="border: 1px solid black; padding: 5px;"> wiederhole bis Rand berührt <div style="border: 1px solid black; padding: 5px; margin-top: 5px; width: fit-content; margin-left: 20px;"> ändere x um 10 </div> </div>	
Schleife mit Zähler	<div style="border: 1px solid black; padding: 5px;"> wiederhole 10 mal <div style="border: 1px solid black; padding: 5px; margin-top: 5px; width: fit-content; margin-left: 20px;"> gehe 4er-Schritt </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; width: fit-content; margin-left: 20px;"> drehe dich nach rechts um 5 Grad </div> </div>	
Endlosschleife	<div style="border: 1px solid black; padding: 5px;"> wiederhole fortlaufend <div style="border: 1px solid black; padding: 5px; margin-top: 5px; width: fit-content; margin-left: 20px;"> gehe 8er-Schritt </div> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; width: fit-content; margin-left: 20px;"> pralle vom Rand ab </div> </div>	



Verzweigung mit Alternative	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2">wird Rand berührt?</td> </tr> <tr> <td>ja</td> <td>nein</td> </tr> <tr> <td>drehe dich nach rechts um 180 Grad</td> <td>gehe 7er-Schritt</td> </tr> </table>	wird Rand berührt?		ja	nein	drehe dich nach rechts um 180 Grad	gehe 7er-Schritt	
wird Rand berührt?								
ja	nein							
drehe dich nach rechts um 180 Grad	gehe 7er-Schritt							
Verzweigung ohne Alternative	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2">wird Ball berührt?</td> </tr> <tr> <td>ja</td> <td>nein</td> </tr> <tr> <td>stoppe alles</td> <td></td> </tr> </table>	wird Ball berührt?		ja	nein	stoppe alles		
wird Ball berührt?								
ja	nein							
stoppe alles								

Im Unterrichtsgang müssen die Schülerinnen und Schüler Struktogramme nicht selbst erstellen. Sie sollen aber lernen, einfache Struktogramme zu verstehen und sie in Codeabschnitte zu übertragen.

Es gibt im Internet einige frei verfügbare Software-Werkzeuge, mit denen Struktogramme erstellt werden können, z.B. Structorizer (<http://structorizer.fisch.lu>), StruktEd (<https://www.robert-scheck.de/tools/strukted>) oder den Struktogrammeditor (<http://www.whiledo.de/programm.php?p=struktoqrammeditor>).

8 Bundesweite Informatikwettbewerbe

Anreize für Jugendliche zur Programmierung und zum Thema Algorithmen bieten die bundesweiten Informatikwettbewerbe (www.bwinf.de), insbesondere der Informatik-Biber. Dieser Wettbewerb ist für Jugendliche aller Altersstufen gedacht (Klasse 3 bis 13). Die Aufgaben setzen einen Schwerpunkt auf das strukturierte und logische Denken und sind so angelegt, dass sie keine Vorkenntnisse in Informatik erwarten. Der Informatik-Biber ist ein Online-Wettbewerb, der jährlich im November stattfindet. Aufgaben früherer Wettbewerbe sind zum Herunterladen verfügbar und können so auch im Unterricht eingesetzt werden (informatik-biber.de).

Für die Zukunft ist ein weiteres Wettbewerbsformat in Entwicklung, das vom Anspruch her zwischen dem Informatik-Biber und dem Bundeswettbewerb Informatik angesiedelt ist. Es soll noch mehr Wert auf das eigenständige Entwickeln und Implementieren informatischer Problemlösungen gelegt werden. Damit wird insbesondere Schülerinnen und Schülern der Sekundarstufe I der Einstieg in die Programmierung und die Entwicklung algorithmischer Lösungen erleichtert. Näheres wird dann ggf. unter www.bwinf.de zu finden sein.



Anhang: Scratch 1.4 vs. Scratch 2.0

Diese Tabelle stellt die beiden erhältlichen Scratch-Versionen einander gegenüber. Sie verdeutlicht einige markante Unterschiede, ist aber keinesfalls vollständig. Scratch bietet auch die Möglichkeit, Programme über Sensoren und Antriebssteuerungen mit der Umwelt interagieren zu lassen. Dies geschieht in den meisten Fällen über die Installation von Erweiterungen. Es würde den Rahmen dieses Textes sprengen, diese auch noch mit einzubeziehen, zumal einige von ihnen nicht für alle Versionen bzw. Betriebssysteme verfügbar sind. Auch ist Vieles noch im Entstehen begriffen. Weitere Informationen bietet z.B. die Wikipedia-Seite von Scratch⁷.

Scratch 1.4	Scratch 2.0
Veröffentlichung	
Juli 2009	Mai 2013
Scratch 3.0 soll voraussichtlich Ende 2017 erscheinen.	
Download und Installation	
https://scratch.mit.edu/scratch_1.4	https://scratch.mit.edu/scratch2download
Betriebssysteme <ul style="list-style-type: none"> • Windows 2000, XP, Vista, 7, 8 (Installation mit exe- oder msi-Datei) • Linux Debian / Ubuntu 12.04+ • Mac OS X 10.4+ 	Offline Editor für <ul style="list-style-type: none"> • Windows • Mac OS X • Linux (einige 32-Bit Versionen) Der Offline Editor benötigt Adobe AIR.
Mit Adobe Integrated Runtime (AIR) lassen sich Webanwendungen als lokal installierte Programme nutzen. Für Linux wird AIR seit einiger Zeit nicht mehr unterstützt. Ältere Linux-Versionen können vom Adobe AIR Archive (https://helpx.adobe.com/air/kb/archived-air-sdk-version.html) heruntergeladen werden. Neuere Versionen sind bei externen Anbietern erhältlich (z.B. von http://www.noobslab.com/2015/05/adobeair-is-now-available-for-ubuntu.html für Ubuntu / Linux Mint).	
7 Seite „Scratch (Programmiersprache)“. URL: https://de.wikipedia.org/wiki/Scratch_(Programmiersprache) . (Abgerufen: 13. Januar 2017)	



Online Version

nicht verfügbar

scratch.mit.edu/projects/editor

auch ohne Benutzerkonto nutzbar

benötigt eine neuere Version eines Browsers, z.B.

- Firefox
- Google Chrome
- Internet Explorer
- Microsoft Edge
- Safari

Adobe Flash Plugin erforderlich

Bei der Online-Version ermöglicht ein Benutzerkonto, sogenannte Cloud-Variablen zu verwenden. Damit können Daten dauerhaft gespeichert und mit anderen geteilt werden.

Merkmale und Kompatibilität

Dateiendung .sb

Dateiendung .sb2

Programme der Version 2.0 nicht ausführbar

Programme der Version 1.4 ausführbar

Skripte können als gif-Bild gespeichert werden (Rechtsklick auf Skript-Hintergrund)

keine Funktion zur Speicherung von Skripten als Bild

Programmierung

Blockbereich Steuerung

Blockbereiche Ereignisse und Steuerung

Definition von Blöcken (Methoden) mit Parametern und Rekursion

Klonen von Objekten, Löschen von geklonten Objekten

Blockbereich Variablen

Blockbereich Daten; Variablen können umbenannt werden

Operatoren: aufrunden, abrunden



Malprogramm mit Pixelgrafik	Malprogramm mit Pixel- und Vektorgrafik, Formate sind ineinander konvertierbar
Einzelstschritt-Modus	Turbo-Modus
	Kamera-Steuerung
	Aufnahme von Projektvideos (60 s)
	weitere Anweisungen, z.B. zum Setzen des Drehtyps