

# Programmieren lernen mit Scratch® – So kreativ ist Informatik

Ein Unterrichtsbaustein zur Förderung informatischer  
Bildung im Sachunterricht der Jahrgangsstufen 3–4

Arbeitsbereich Didaktik der Informatik der WWU Münster  
Autor: Alexander Best

## Urheberrechtserklärung

© Arbeitsbereich Didaktik der Informatik der WWU Münster 2021

Dieser Unterrichtsbaustein ist eine Weiterentwicklung des in Kooperation mit Grundschullehrpersonen entwickelten, gleichnamigen Bausteins, der im Rahmen des Dissertationsprojekts von Alexander Best entwickelt wurde. Die Erstveröffentlichung kann unter der nachfolgenden Arbeit eingesehen werden:

Alexander Best (2020): Vorstellungen von Grundschullehrpersonen zur Informatik und zum Informatikunterricht. Dissertation, Westfälische Wilhelms-Universität Münster, S. 525–552. Online verfügbar unter [https://ddi.wwu.de/2020\\_best\\_diss](https://ddi.wwu.de/2020_best_diss), zuletzt geprüft am 02.02.21.

### Open Access

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ Namensnennung 4.0 International zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie <https://creativecommons.org/licenses/by/4.0/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.



Von dieser Lizenz ausgenommen sind Abbildungen, welche sich nicht im Besitz des Autors bzw. des Arbeitsbereichs befinden.

### Zitieren dieses Werkes

Arbeitsbereich Didaktik der Informatik (2021): Programmieren lernen mit Scratch® – So kreativ ist Informatik, Westfälische Wilhelms-Universität Münster

## Didaktisch-methodische Handreichung

**Hinweis:** Für diesen Unterrichtsbaustein werden stationäre oder mobile PCs benötigt. Die Schülerinnen und Schüler arbeiten in Partnergruppen, sodass die Anzahl der verfügbaren PCs der Hälfte der Lerngruppenstärke inklusive eines PCs für die Lehrperson entsprechen sollte. Um Scratch® zu installieren, müssen entsprechende Administratorenrechte zur Verfügung stehen. Bitte informieren Sie sich vorab bei dem/der entsprechenden Beauftragten an Ihrer Schule. Liegen diese Rechte nicht vor, kann Scratch® auch in einer online-basierten Variante genutzt werden. Hierfür benötigen die PCs Zugriff auf das Internet. Bitte informieren Sie sich vorab, ob Ihre Schule über einen entsprechenden Zugang verfügt und ob ggfs. Filter eingerichtet sind, die nur Zugriff auf spezifische Webseiten gestatten. Grundsätzlich können wir keine Unterstützung bei Fragen der Wartung oder des Betriebs zur digitalen Infrastruktur an Ihrer Schule bereitstellen. Die analogen Unterrichtsmaterialien, hierzu zählen bspw. die Arbeitsblätter, stehen am Arbeitsbereich Didaktik der Informatik der Westfälischen Wilhelms-Universität Münster (WWU) zur Verfügung und werden kostenlos bereitgestellt. Falls Sie dieses Angebot wahrnehmen möchten oder Fragen haben, erreichen Sie uns unter [grundschulinformatik@uni-muenster.de](mailto:grundschulinformatik@uni-muenster.de) oder alternativ über unser Sekretariat (Frau Andrea Lieske) unter +49 251 83-39397 (Tel.) bzw. +49 251 83-39369 (Fax). Sie finden unseren Arbeitsbereich am Institut für Didaktik der Mathematik und der Informatik, Corrensstr. 80, 48149 Münster.

Wir wünschen Ihnen viel Spaß und Erfolg bei der Durchführung.

---

**Kernanliegen:** Die Schülerinnen und Schüler verwenden spezifische Blöcke in Scratch® gezielt, um das Verhalten von Objekten zu steuern. Hierbei nutzen sie Elementaranweisungen, verbinden diese über die algorithmische Grundstruktur „Sequenz“ zu zusammengesetzten Anweisungen, gebrauchen die Kontrollstrukturen „Zählzyklus“ sowie „Verzweigung“, um Anweisungen zu wiederholen bzw. um sie in Abhängigkeit von Bedingungen auszuführen. Sie sind darüber hinaus in der Lage, ein selbstentwickeltes Programm zu erläutern sowie Abläufe auf verwendete algorithmische Grundbausteine und -strukturen zurückzuführen und umgekehrt von diesen auf zu erwartende Abläufe zu schließen.

**Paradigma:**

*Plugged*



*Unplugged*



**Informatische Vorkenntnisse:** Keine

**Fachübergreifende Vorkenntnisse/Fähigkeiten:** (a) Kooperatives Arbeiten zu zweit, (b) respektvoller Umgang miteinander und mit technischen Geräten, (c) psychomotorische Fähigkeiten zur Nutzung von Tastatur und Maus, (d) Erläutern von eigenen Arbeitsergebnissen im Plenum, (e) Interaktion mit dem Plenum, (f) Lesekompetenz zum Erfassen der Arbeitsblätter und (g) Annehmen von Hilfestellungen über Tippkarten bei Problemen mit Aufgabenstellungen

**Klasse:** 3–4

**Umfang:** 90 min

**Thema:** Einführung in die algorithmische Grundstruktur „Sequenz“ sowie die Kontrollstrukturen „Verzweigung“ und „Zählzyklus“ zur Modellierung und Implementierung eines Programms in der Programmiersprache Scratch®

**Themengebiet:** Algorithmik

**Inhalt:** Sequenz, Verzweigung und Zählzyklus

**Gegenstände:** Stationärer oder mobiler PC, Scratch®-Anweisungskarten, Scratch®-Tipp-Karten, Scratch®

**Benötigte Materialien:**

- (1) Scratch®-Benutzeroberfläche mit Bezeichnungen
- (2) Scratch®-Tipp-Karten
- (3) Arbeitsblatt eins „Mein erstes Programm“
- (4) Scratch®-Anweisungskarten
- (5) Große Variante der Scratch®-Anweisungskarten zur Verwendung an der Tafel
- (6) Magnete zur Fixierung von Scratch®-Anweisungskarten an der Tafel

**Benötigte Medien:** Tafel und Beamer

---

**Geförderte Kompetenzen – Informatik<sup>1</sup>**

**Inhaltsbereiche:** ALGORITHMEN

**Prozessbereich:** KOMMUNIZIEREN UND KOOPERIEREN

---

<sup>1</sup> Angelehnt an die Kompetenzen für informatische Bildung der Gesellschaft für Informatik (GI) 2019.

### **Kompetenzerwartungen:**

„Die Schülerinnen und Schüler

- entwerfen, realisieren und testen Algorithmen mit den algorithmischen Grundbausteinen *Anweisung, Sequenz, Wiederholung<sup>2</sup> und Verzweigung*
- programmieren ein Informatiksystem“ (Gesellschaft für Informatik (GI) 2019, S. 13).

### **Geförderte Kompetenzen – Sachunterricht<sup>3</sup>**

**Perspektive:** TECHNISCHE PERSPEKTIVE

**Perspektivbezogene Denk-, Arbeits- und Handlungsweisen:** TECHNIK KOMMUNIZIEREN

### **Kompetenzerwartungen:**

„Die Schülerinnen und Schüler können:

- Ideen für technische Lösungen, Konstruktionsergebnisse, Funktionszusammenhänge, Herstellungsprozesse sowie Arbeitsabläufe unter Nutzung von Sprache, Zeichnungen oder Demonstrationen verständlich vermitteln, diskutieren und dokumentieren (z.B. durch Skizzen, Sachzeichnungen, Beschreibungen, Abbildungen, Fotos)
  - Anleitungen lesen, verstehen und umsetzen sowie einfache Anleitungen selbst verfassen“ (Gesellschaft für Didaktik des Sachunterrichts (GDSU) 2013, S. 68).
- 

<sup>2</sup> In dieser Arbeit wird der Begriff „Zyklus“ gebraucht. „Wiederholung“ oder „Schleife“ werden als alltagssprachliche Synonyme aufgefasst.

<sup>3</sup> Angelehnt an den Perspektivrahmen Sachunterricht der Gesellschaft für Didaktik des Sachunterrichts (GDSU) 2013.

## Didaktischer Schwerpunkt

**Relevanz für die Schülerinnen und Schüler:** Die Gestaltung von Informatiksystemen über eine Modellierung und Implementierung erfolgt mittels standardisierter Verfahren, Bausteine und Strukturen. Deren Vermittlung muss an die jeweilige Zielgruppe und deren Voraussetzung angepasst sein. Um den Schülerinnen und Schülern diese Gestaltung von Informatiksystemen zu ermöglichen, benötigen sie Kompetenzen, um einerseits Abläufe nachvollziehen und sie andererseits so beschreiben zu können, dass diese auf Informatiksystemen umgesetzt werden können. Mit der Sequenz, der Verzweigung und dem Zyklus sollen ihnen hierfür drei Strukturen vermittelt werden. Diese stellen eine wesentliche Grundlage dar, sodass die Schülerinnen und Schüler Abläufe auf Informatiksystemen inner- und außerhalb der Schule nachvollziehen können. Sie bilden zudem ein Fundament für den Erwerb weiterführender informatischer Kompetenzen in der Sekundarstufe. Neben diesen kognitiven Aspekten erleben die Schülerinnen und Schüler, dass sie in der Lage sind, Abläufe auf Informatiksystemen nachzuvollziehen und diese zudem selbst zu beschreiben sowie anschließend umzusetzen. Informatiksysteme werden von einer *Blackbox* (nur funktionale Sichtweise) zu einer *Grey-Box* (funktional-strukturelle Sichtweise).

**Relevanz für die Grundschule und den Sachunterricht:** Die Nutzung von Informatiksystemen sowohl im Sachunterricht als auch in anderen Grundschulfächern wird in den kommenden Jahren weiter zunehmen. Dazu zählen neben Desktop-PCs auch mobile Endgeräte wie Notebooks, Tablets oder Smartphones. Darüber hinaus besteht die Möglichkeit, dass das in der Sekundarstufe erprobte Konzept *Bring Your Own Device (BYOD)* auch in der Grundschule Anwendung findet. Problematisch zeigt sich hierbei die mangelnde Kompetenz der Schülerinnen und Schüler beim Verständnis für zentrale Prinzipien, die diesen Systemen zugrunde liegen (*Blackbox*-Sichtweise). Anwenderschulungen haben sich bereits in der Sekundarstufe als nicht zielführend erwiesen und führen nicht zu nachhaltigen Kompetenzen, welche eine zeitlich längerfristige oder gar zeitinvariante Relevanz für die Alltagswelt der Schülerinnen und Schüler aufweisen. Mit der Einführung in die drei algorithmischen Grundstrukturen „Sequenz“, „Verzweigung“ und „Zyklus“ werden die Schülerinnen und Schüler in die Lage versetzt – vielleicht erstmalig – Informatiksysteme auch aus einer strukturellen Perspektive zu betrachten, die ihnen Einblick über die reine Interaktion dieser Systeme gibt. Diese Kompetenzen können die Grundlage für die Gestaltung von Informatiksystemen bilden sowie die Schülerinnen und Schüler dazu befähigen, Informatiksysteme anhand von Möglichkeiten und Grenzen der Gestaltung zu analysieren. Somit wird ein Grundstein gelegt, um den Umgang mit und die Gestaltung von Informatiksystemen fächerverbindend zu ermöglichen.

**Didaktischer Kommentar:** In diesem Baustein setzen die Schülerinnen und Schüler sich mit Hilfe von Scratch® einerseits mit Aspekten des Prozesses der Softwareentwicklung, insbesondere mit dessen kooperativen und partizipativen Anteilen, und andererseits mit den algorithmischen Grundstrukturen „Sequenz“, „Verzweigung“ und „Zyklus“ auseinander. Bei Scratch® handelt es sich um eine Programmiersprache, -umgebung und Onlinegemeinschaft. Scratch® zählt zu den sogenannten blockorientierten Programmiersprachen, welche speziell für

den niederschweligen Einstieg von Kindern in die Programmierung konzipiert wurde. Sie wird seit mehreren Jahren am *Massachusetts Institute of Technology (MIT)* in der sogenannten *Lifelong Kindergarten Group* unter Leitung von Mitchel Resnick entwickelt. Ein Kern des Projekts besteht in der Kreativitätsförderung. Für diesen Baustein wird Scratch® in der Version 2.0 genutzt, welche in einer offline- und online-Variante vorliegt. Mittlerweile wurde die online-Variante durch Scratch® 3.0 ersetzt. Die offline-Variante von Scratch® 2.0 steht weiterhin zum Download<sup>4</sup> bereit. Die Mehrheit der Nutzerinnen und Nutzer von Scratch® ist zwischen 8 und 19 Jahre alt.

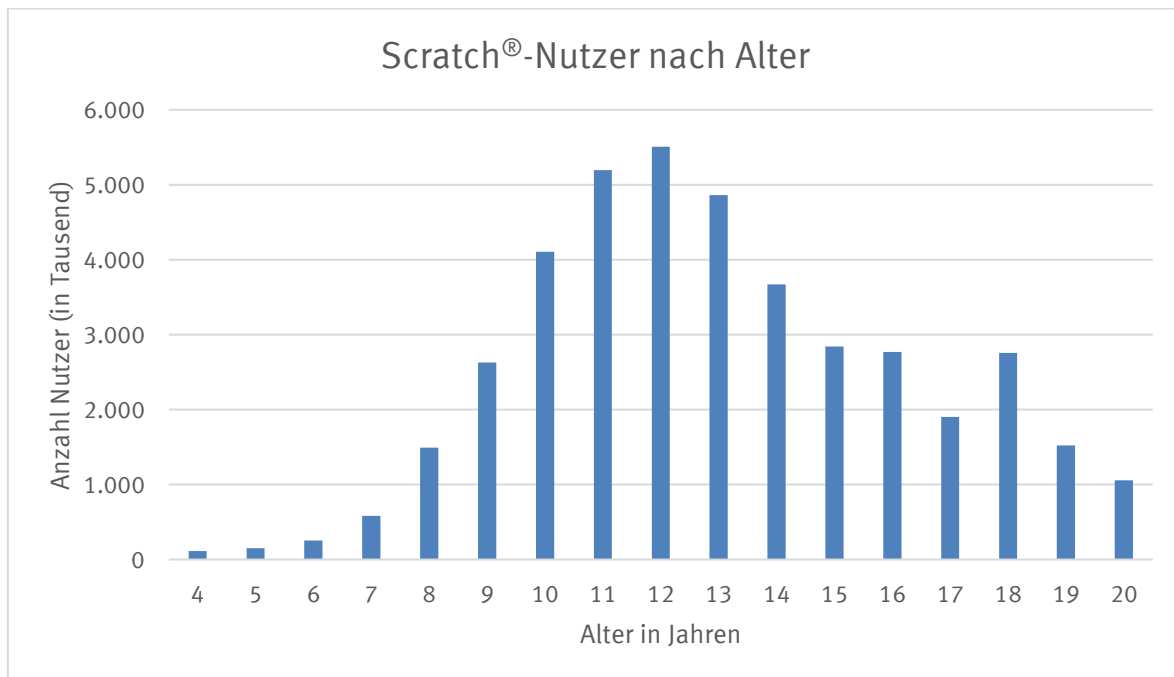


Abbildung 1: Verteilung von Scratch®-Nutzern nach ihrem Alter (Stand: Oktober 2020; Quelle: <https://scratch.mit.edu/statistics/> (Diagramm „Age Distribution of New Scratchers“); ab dem Sekundarbereich werden nur Altersgruppen mit einer Nutzerzahl von  $\geq 1$  Million berücksichtigt)

Für den Einstieg wird die Lebenswelt der Schülerinnen und Schüler einbezogen und beachtet. Dies dient einerseits der Steigerung der Motivation und soll andererseits eine Transparenz für die Schülerinnen und Schüler schaffen. Scratch® dient dabei lediglich als ein Werkzeug, mit dem die Einführung in die Programmierung erfolgt. Die Schülerinnen und Schüler sollen explizit nicht zu Programmierexperten geschult werden, sondern sich über Scratch® die oben genannten Prozesse und Inhalte aus der Softwareentwicklung aneignen sowie ein positives Informatik-Selbstkonzept entwickeln. Über den Begriff „Programmieren“ tragen die Schülerinnen und Schüler zunächst ihre eigenen Vorstellungen sowie ihr Wissen und ihre Kenntnisse zusammen und visualisieren diese anhand einer Mindmap. Die Mindmap bietet sich methodisch für die Lehrperson an, um eine kurze Evaluation der bestehenden Vorstellungen sowie des Vorwissens und der -kenntnisse bei den Schülerinnen und Schülern durchzuführen. Die Lehrperson kann nun gezielt Aspekte generalisieren oder akzentuieren. Im Plenumsgespräch können die Schülerinnen und Schüler die zusammengetragenen Aspekte auf ihren jeweiligen Mindmaps vorstellen. Zielführende Aspekte werden dabei von der Lehrperson

<sup>4</sup> <https://scratch.mit.edu/download/scratch2>, zuletzt geprüft am 13.10.2020.

aufgegriffen und in eine zentrale Mindmap überführt bzw. integriert. Dabei sollen im Plenumsgespräch einerseits konvergierende als auch divergierende Begriffe genannt werden. Hierbei sind insbesondere Begriffe für den weiteren Verlauf des Unterrichts erforderlich, die sich auf Informatiksysteme mittels des Programmierens beziehen. Dies können folgende, exemplarische Äußerungen sein:

„Programmieren bedeutet, dass man dem Computer sagt, was er tun soll.“

Inwieweit die Aussagen zu Fernsehprogrammen, die in bisherigen Durchführungen dieses Unterrichtsbausteins auftauchten, zielführend aufgegriffen werden können, ist bislang noch nicht geklärt. Möglicherweise könnte hieraus der Aspekt „Steuerung“ eines Geräts abgeleitet werden. Als Ergebnis des Einstiegs sollte in der zentralen Mindmap festgehalten werden, dass mittels Programmierung Computer gesteuert werden können, wobei hierfür nicht die natürliche Sprache, sondern eine formale Programmiersprache („Computersprache“ oder besser „Programmiersprache“) verwendet werden muss. Der Begriff „Programmieren“ wird von der Mindmap in einen Wortspeicher überführt. Somit fallen andere aufgenommene Aspekte aus der Mindmap, die sekundären Charakter haben, weg. Zudem kann der Begriff „Programmieren“ anschließend mit anderen Fachbegriffen gekoppelt werden.

Von der Einführung leitet ein kurzer Impulsvortrag in die Erarbeitungsphase über. In dieser soll von der Lehrperson die Benutzeroberfläche von Scratch® zunächst visualisiert und anschließend anhand spezifischer Bezeichnungen in unterschiedliche Bereiche aufgeteilt werden. Hierüber sollen die Schülerinnen und Schüler in die Lage versetzt werden, sich mittels der eingeführten Begriffe auszutauschen. Hierdurch sollen Mehrdeutigkeiten und Unsicherheiten verhindert werden. Anschließend findet die erste Interaktion der Schülerinnen und Schüler mit Scratch® statt. Hierfür erhalten sie zunächst Aufgabenstellungen mittels eines Arbeitsblattes. Dieses wird von der Lehrperson vorgelesen und es werden Unklarheiten erfragt sowie geklärt. Über das Vorlesen wird sowohl der auditive als auch der visuelle Inputkanal der Schülerinnen und Schüler angeregt.

In der Erarbeitungsphase eins arbeiten die Schülerinnen und Schüler in Partnerarbeit. Diese soll einerseits zu Anlehnung an kooperative Praktiken in der Softwareentwicklung führen und andererseits die Vorteile kooperativer Methoden im Unterricht nutzbar machen. Dazu zählen:

- Wechselwirkungen zwischen Implementieren und Modellieren
- Notwendigkeit der Verbalisierung mentaler Modelle
- Beachtung von *Debugging*, Testen, Dokumentation etc.
- Notwendigkeit der Erläuterung von Ideen und Ansätzen
- Möglichkeiten, um Verständnisschwierigkeiten zunächst im kleinen Rahmen zu thematisieren

Nach der Auseinandersetzung mit den gestellten Aufgaben in Scratch® gehen die Schülerinnen und Schüler dazu über, selbstständig Blöcke auszuwählen, einzusetzen und sich deren Funktion zu erarbeiten. Während in der Sekundarstufe I ein Einstieg in Scratch® auch ohne vorgegebene Aufgaben erfolgreich verlief, wurde für die Grundschule entschieden, dass die große Auswahlmöglichkeit an Blöcken die Schülerinnen und Schüler überfordern würde und sie zunächst einen klaren Rahmen bräuchten, um anschließend ihre Kreativität zu nutzen und die Funktion von Blöcken zu erarbeiten. Neben dem Ausprobieren dieser Blöcke sollen die



Partnergruppen bereits Überlegungen zu einem möglichen Softwareprojekt entwickeln. Dies wird i. d. R. ein kleines Spiel sein.



Abbildung 2: Beispiel für die Nutzung der Kontrollstruktur „Verzweigung“ zur Feststellung eines Tastendrucks (hier: Leertaste) in Scratch® 2.0 (erstellt mit scratchblocks 3.4, <https://scratchblocks.github.io/>, zuletzt geprüft am 13.10.2020)

Über eine Zwischensicherung wird einerseits festgehalten, welche Aufgaben zur Entwicklung des eigenen Projekts noch durchgeführt werden müssen und andererseits, ob das Projekt umsetzbar ist. Hier muss die Lehrperson beratend und lenkend eingreifen. In der Zwischensicherung werden darüber hinaus auch generelle Fragen, bspw. zu Blöcken, besprochen, welche in der ersten Erarbeitungsphase auftauchten. Neu ist der Hinweis auf die zwei bislang noch nicht genutzten Kontrollstrukturen „Verzweigung“ und „Zyklus“. Diese werden den Schülerinnen und Schülern erläutert, auf das Arbeitsblatt zwei aufgenommen und im Wortspeicher festgehalten.



Abbildung 3: Beispiel für die Nutzung der Kontrollstruktur „Zählzyklus“, um die Scratch®-Katze sich um 360° im Uhrzeigersinn drehen und dabei für jede Drehung um 90° ein „Miau“ ertönen zu lassen (erstellt mit scratchblocks 3.4, <https://scratchblocks.github.io/>, zuletzt geprüft am 13.10.2020)

In der Erarbeitungsphase zwei entwickeln die Schülerinnen und Schüler ihre eigenen Projektideen, müssen in diesen jedoch die zuvor eingeführten Kontrollstrukturen nutzen. Die Lehrperson sichtet die Zwischenstände der Partnergruppen, gibt Impulse und Rückmeldungen.

Die Vorstellung der Ergebnisse erfolgt zunächst in einer Präsentation im Plenum. Hierbei können, außer es handelt sich um eine sehr kleine Lerngruppe, nicht alle Partnergruppen ihr Projekt präsentieren. Die Lehrperson sollte bereits in der Erarbeitungsphase Projekte identifizieren, deren Präsentation im Plenum zielführend ist. Diese sind einerseits Projekte, in denen die Kontrollstrukturen sinnvoll eingesetzt wurden und welche auch gestalterisch eine hohe Qualität aufweisen. Doch auch unabgeschlossene Projekte oder gute Ansätze sollen vorgestellt werden. Hier können die Gruppen verbalisieren, welche Probleme sie hatten und ob bzw. wie sie diese lösen konnten.

In der abschließenden Sicherungsphase sollen einerseits die Begriffe aus dem Wortspeicher erneut thematisiert und von den Schülerinnen und Schülern erläutert werden. Andererseits sollen diese die Notwendigkeit der Modellierung und anschließend Programmierung zur

Gestaltung eines Informatiksystems mittels einer Programmiersprache betonen. Abschließend wird der Bezug zur Informatik als wissenschaftliche Disziplin sowie als Unterrichtsfach hergestellt, sodass die Schülerinnen und Schüler eine Vorstellung davon entwickeln können, welche Tätigkeiten eine Informatikerin bzw. ein Informatiker ausübt und welche Inhalte und Prozesse im Informatikunterricht in der Sekundarstufe vermittelt werden.

## Geplanter Verlauf des Unterrichts

### Artikulationsschema

Phase (Zeitangaben ggfs. anpassen)	Unterrichtsinhalte	Sozial-/ Arbeitsformen	Material/Medien	Didaktisch-methodischer Kommentar
Einstieg (~ 10 min)	<p>Mindmap: Die SuS werden aufgefordert, eine Mindmap zum Begriff „Programmieren“ zu entwickeln. Mindmaps werden bereits in der Grundschule genutzt, um Vorkenntnisse, -erfahrungen und -stellungen von SuS für den Unterricht sichtbar zu machen.</p> <p>Anschließend werden die Ergebnisse der SuS im Plenum gesammelt und von der Lehrperson in einer zentralen Mindmap festgehalten. Erfahrungsgemäß wird Programmieren einerseits mit Fernsehprogrammen assoziiert und andererseits mit der Steuerung von Computern („Einem Computer sagen, was er tun soll!“).</p> <p>Letzteres kann als</p>	<p>EA</p> <p>UG</p> <p>LV</p>	<ul style="list-style-type: none"> <li>• Mindmap</li> <li>• Zentrale Mindmap</li> <li>• Wortspeicher</li> </ul>	<p>Falls die SuS mit dem Begriff „Programmieren“ keine Vorstellungen assoziieren, dies trat jedoch in Unterrichtserprobungen nicht auf, kann evtl. über den Begriff „Programm“ operiert werden.</p> <p>Die Nutzung der Mindmap ermöglicht es den SuS einerseits, ihre bestehenden Vorstellungen bzw. ihr vorhandenes Vorwissen zu strukturieren und andererseits, dieses mit anderen SuS abzugleichen. Beide Aspekte sind auch für die Lehrperson von Bedeutung, um möglicherweise auf bereits bekannte Programmiersprachen (bspw. Java) einzugehen und zu verdeutlichen, dass ein Computer nicht ohne eine Programmiersprache programmiert und gesteuert werden kann.</p> <p>Nach Möglichkeit sollten in der zentralen Mindmap Erläuterungen aus den Mindmaps der SuS aufgegriffen und</p>

	<p>Ausgangspunkt genutzt werden, um zu Scratch® als einem Werkzeug überzuleiten, mit dem diese Steuerung möglich ist. Dabei sollte jedoch betont werden, dass Computer nicht über unsere natürliche Sprache gesteuert werden können, sondern andere, formale Sprachkonzepte („Computersprache“ oder besser „Programmiersprache“) hierfür genutzt werden müssen.</p> <p>Der Begriff „Programmieren“ wird in einem Wortspeicher festgehalten, da die Mindmap nicht zielführend ist, um in nachfolgenden Handlungssituationen, -mustern und Unterrichtsschritten gezielt auf eine Definition zugreifen zu können. Ein exemplarischer Wortspeicher befindet sich in den Unterrichtsmaterialien.</p> <p><b>Gelenkstelle:</b> Anschließend erfolgt die Überleitung zur Demonstration der grafischen Benutzeroberfläche von Scratch®.</p>			<p>adaptiert werden. Einerseits fällt es den anderen SuS leichter, diese Beschreibungen („Was bedeutet das?“) zu verstehen, da die Formulierung nahe an ihrer eignen Sprache ist, und andererseits erfolgt dadurch eine Wertschätzung der SuS und ihrer individuellen Mindmaps.</p> <p><b>Motivation</b> entsteht bei den SuS dadurch, dass ihr Vorwissen bzw. ihre existierenden Vorstellungen aufgegriffen werden und andererseits dadurch, dass sie die Möglichkeit der eigenen Gestaltung von Informatiksystemen erhalten. Dies soll über die schülerorientierte Programmiersprache/-umgebung Scratch® geschehen, welche ebenfalls zur Motivationssteigerung beiträgt.</p> <p><b>Zieltransparenz</b> wird für die SuS über die Gestaltung eines Informatiksystems hergestellt.</p> <p><b>Sinntransparenz</b> besteht darin, dass Informatiksysteme gestaltet werden müssen, um mit ihnen Aufgaben zu lösen. Ohne eine solche Gestaltung werden keine Operationen ausgeführt und die Hardware erfüllt keinerlei Funktion.</p> <p><b>Organisationstransparenz</b> wird noch</p>
--	--	--	--	---

				<p>nicht im Einstieg hergestellt, sondern im Anschluss an den Impulsvortrag der Lehrperson. Dann jedoch sollten die SuS verstanden haben, dass in Partnerarbeit ein eigenes Softwareprojekt mittels Scratch® entwickelt werden soll. Die Verwendung der Struktur „Sequenz“ erfolgt intuitiv und kann anschließend von der Lehrperson induktiv aufgegriffen werden. Die Einführung der Kontrollstrukturen „Verzweigung“ sowie „Zyklus“ erfolgt zunächst durch die Lehrperson; die Kontrollstrukturen sollen anschließend durch die SuS in ihren Projekten genutzt werden.</p> <p>Die Fachbegriffe in dieser Stunde sind: <b>Programmieren, Programmiersprache, Anweisung, Sequenz, Verzweigung</b> sowie <b>Zyklus</b>. Obwohl alltagsprachliche Begriffe existieren, denen ähnliche Konzepte zugrunde liegen (Reihe, Unterscheidung, WENN-DANN oder Schleife), wird auf diese nicht zurückgegriffen, da sie auch zu fehlerhaften Vorstellungen bezüglich der programmiersprachlichen Konstrukte führen können. So wird die Schleife häufig mit dem Binden von Schnürsenkeln assoziiert, welche jedoch eine Verzahnung zweier Schleifen voraussetzt. Auch die Reihe ist problembehaftet, da</p>
--	--	--	--	--

				sich die Anordnung der Elemente einer Reihe, bspw. durch Vordrängeln, verändern kann, was beim programmiersprachlichen Konzept der Sequenz vermieden werden sollte („ <i>Go To Statement Considered Harmful</i> “).
Impulsvortrag (~ 5 min)	<p>Die Lehrperson präsentiert am Beamer oder alternativ auf einem Poster oder per OHP die grafische Benutzeroberfläche von Scratch®. Diese kann anhand der folgenden Bereiche differenziert und benannt werden:</p> <ul style="list-style-type: none"> <li>• Bühnenbereich</li> <li>• Objektbereich</li> <li>• Blockbereich</li> <li>• Skriptbereich</li> </ul> <p>Eine Übersicht befindet sich in den Unterrichtsmaterialien. Die Lehrperson führt einige Beispiele in Scratch® vor, welche von den SuS anschließend in der Erarbeitungsphase nachvollzogen und erweitert werden sollen:</p> <ol style="list-style-type: none"> <li>1. Das Kostüm der Katze sowie der Bühne ändern (Objektbereich)</li> <li>2. „Wenn ,Grüne Flagge“</li> </ol>	LV	<ul style="list-style-type: none"> <li>• Beamer</li> </ul>	<p>Die SuS erhalten eine Übersicht über die grafische Benutzeroberfläche und entsprechenden Bezeichnungen in gedruckter Form. So haben sie die Möglichkeit, jederzeit auf die Übersicht zurückzugreifen und die Fachbegriffe zu verwenden. Dies ist wichtig, damit eine einheitliche Bezeichnung für die Elemente und Bereiche der Benutzeroberfläche in der Klasse gebraucht wird.</p> <p>Die Aufgabenstellungen werden von der Lehrperson sowohl mündlich erläutert als auch in schriftlicher Form an die SuS weitergegeben. Somit werden unterschiedliche Inputkanäle (auditiv, visuell) angesprochen.</p>

	<p>angeklickt“ + „gehe 10er-Schritt“ vom Blockbereich in den Skriptbereich ziehen</p> <p>3. Programm per grüner Flagge starten (Bühnenbereich)</p> <p>4. Zuschauen, was im Bühnenbereich passiert</p> <p>Die Beispiele und Arbeitsaufträge werden den SuS anschließend schriftlich präsentiert (siehe Arbeitsblatt eins).</p>			
<p>Erarbeitung I (~ 15 min)</p>	<p>Die SuS haben nun die Möglichkeit, die von der/vom LoL vorgeführten Funktionen auszuprobieren und natürlich weitere Blöcke zu verwenden. Die Lehrperson ist Ansprechpartner/in bei Problemen und Fragen.</p>	<p>PA</p>	<ul style="list-style-type: none"> <li>• Scratch®</li> <li>• Arbeitsblatt eins</li> </ul>	<p>Das Arbeiten in Partnerarbeit soll den SuS einerseits den hohen kooperativen, sozialen und partizipativen Anteil am Softwareentwicklungsprozess vermitteln und Fehlvorstellungen vorbeugen → Entwickler als isolierte Einzelgänger. Darüber hinaus sollen sich die SuS dabei unterstützen, Abläufe zunächst als Ziel festzusetzen und anschließend entsprechende Möglichkeiten über die Scratch®-Blöcke zu entdecken, um ihre Ziele umzusetzen.</p> <p>Vor dem Einsatz der Programmierumgebung Scratch® müssen die SuS dazu angeleitet werden, wie sie ihre Projekte sichern sollen. Dazu zählen Vorgaben zum Datenträger, zum Verzeichnis und zur Benennung der Dateien (Dateiname). Idealerweise</p>

				werden die Daten auf ein Netzlaufwerk oder ein Schul-Cloud gespeichert. Das Speichern auf lokalen Datenträgern, bspw. USB-Sticks, ist ebenfalls möglich, falls sichergestellt ist, dass die SuS über den gesamten Verlauf der Unterrichtseinheit Zugang zu diesen haben. Von der Speicherung bei kommerziellen Cloud-Anbietern sollte abgesehen werden.
Zwischensicherung (~ 5 min)	Die bisherigen Fragen der SuS sowie ihre Erkenntnisse bei der Nutzung von Blöcken werden im Plenum besprochen.	UG im Stuhlkreis	—	Mögliche Probleme und Lösungen, die in der Erarbeitungsphase aufgetaucht sind, können nun problematisiert werden. Darüber hinaus geben die SuS einen Ausblick darauf, welche Weiterentwicklungen ihrer Programme sie in der anschließenden Erarbeitungsphase II vornehmen wollen.
Erarbeitung II (~ 20 min)	Ihre Projekte sollen die SuS nun erweitern und hierbei die Kontrollstrukturen „Verzweigung“ und „Zyklus“ einsetzen. Die Nutzung innerhalb ihres Projekts kann dabei individuell erfolgen. Sie sollen jedoch festhalten, wieso der Einsatz der jeweiligen Kontrollstruktur sinnvoll war und ob sie diese Funktionalität auch ohne deren Einsatz hätten ermöglichen können.	PA	<ul style="list-style-type: none"> <li>• Scratch®</li> <li>• Arbeitsblatt zwei</li> </ul>	Die SuS sollen in dieser Phase einerseits den Einsatz der Kontrollstrukturen „Zyklus“ und „Verzweigung“ einüben und andererseits erkennen, dass diese Strukturen notwendig sind, um spezifische Funktionalitäten umsetzen zu können.
Präsentation	Es werden einige Ergebnisse von	SV	<ul style="list-style-type: none"> <li>• Beamer</li> </ul>	Zweck der Präsentation ist es nicht, eine



<p>(3–5 min pro Gruppe)</p>	<p>Partnergruppen im Plenum präsentiert. Dabei sollte eine gewisse qualitative Bandbreite erreicht und auch unvollständige Projekte aufgegriffen werden.</p> <p>Die SuS sollen zunächst mündlich beschreiben, was sie mit ihrem Programm erreichen wollten. Anschließend lassen sie ihr Programm ablaufen und erläutern abschließend ihren Quelltext, wobei sie Bezüge zum Programmablauf herstellen.</p> <p>Es erfolgen Rückfragen der Lehrperson sowie der SuS aus dem Plenum.</p>	<p>UG</p>	<ul style="list-style-type: none"> <li>• Scratch®</li> <li>• Quelltexte einiger Partnergruppen</li> </ul>	<p>hohe Quantität an Programmen der SuS zu präsentieren. Es sollen stattdessen gelungene programmiersprachliche und auch Ideen aufgegriffen, aber auch Probleme angesprochen werden. Hierunter fällt bspw. eine Disparität zwischen der erwarteten Funktionsweise und der programmiersprachlichen Funktionsweise eines Scratch®-Blocks. Insbesondere solche Ergebnisse sind hier von Interesse, bei denen die SuS ein bestehendes Problem lösen konnten und diesen Prozess im Plenum rekapitulieren können.</p> <p>Die Lehrperson muss während der vorherigen Erarbeitungsphase II den Stand der Softwareprojekte sichten und hinsichtlich einer Thematisierung in der Präsentationsphase evaluieren.</p>
<p>Sicherung (10 min)</p>	<p>In der Besprechung erfolgt der Zirkelschluss auf den Einstieg. Es soll nochmals thematisiert werden, was unter „Programmierung“ zu verstehen ist und was das Arbeiten mit Scratch® gegenüber der reinen Anwendung von Software am Computer unterscheidet. Besonders die Strukturen „Sequenz“, „Verzweigung“ und „Zyklus“ sollen erneut</p>	<p>UG</p>	<p>–</p>	<p>Die Lehrperson weist darauf hin, dass die behandelten algorithmischen Grundstrukturen auch von Informatikern bei der Programmierung genutzt werden. Die Relevanz der Strukturen über den unterrichtlichen Einsatz hinaus in Wirtschaft und Wissenschaft kann die Motivation der SuS erhöhen und auch dazu führen, dass sie Interesse an Softwareentwicklung speziell und Informatik allgemein in ihrer Freizeit gewinnen.</p>

	<p>aufgegriffen und deren Funktion gesichert werden.</p> <p>Abschließend erfolgt der Hinweis auf die Disziplin der Informatik, welche sich mit derartigen Inhalten und Prozessen auseinandersetzt, sowie auf das Unterrichtsfach Informatik in der Sekundarstufe.</p>			
--	---	--	--	--

## Unterrichtsmaterialien

siehe nächste Seiten

# Folie 1 - Grafische Benutzeroberfläche

The image shows the Scratch graphical user interface with three red callouts identifying key areas:

- Bühnenbereich:** The stage area on the left, containing the Scratch cat character.
- Objektbereich:** The 'Figuren' (Sprites) panel at the bottom left, showing the 'Figur1' (Scratch cat) sprite.
- Skriptbereich:** The 'Skripte' (Scripts) panel in the center, containing a sequence of movement and control blocks.

The 'Skripte' panel contains the following blocks:

- gehe 10 er-Schritt
- drehe dich um 15 Grad
- drehe dich um 15 Grad
- setze Richtung auf 90
- drehe dich zu Mauszeiger
- gehe zu x: 0 y: 0
- gehe zu Mauszeiger
- gleite in 1 Sek. zu x: 0 y: 0
- ändere x um 10
- setze x auf 0
- ändere y um 10
- setze y auf 0
- pralle vom Rand ab
- setze Drehtyp auf links-rechts
- x-Position

The 'Figuren' panel shows 'Figur1' with a 'Neue Figur' button and icons for adding new sprites.

## Folie 2 - Exemplarischer Wortspeicher

Informatik-Wort	Was bedeutet das?	Ein Beispiel in Scratch
Programmiersprache	Besondere Sprache, die ein Computer versteht.	Alle Beispiele in dieser Spalte gehören zur Programmiersprache Scratch.
Anweisung	Eine kurze Aufgabe in einer Programmiersprache.	
Programmieren	Einem Computer Aufgaben in einer Programmiersprache vorgeben.	Wir programmieren den Computer später mit der Programmiersprache Scratch.
Sequenz	Das Aneinanderreihen von Anweisungen, die der Computer nacheinander befolgt.	
Verzweigung	Der Computer befolgt die Anweisungen nur dann, wenn eine bestimmte Bedingung eintritt. Sonst befolgt er eine andere Anweisung.	
Zählzyklus	Der Computer führt Anweisungen eine bestimmte Anzahl von Malen aus.	

## Arbeitsblatt 1: Die Scratch-Programmiersprache

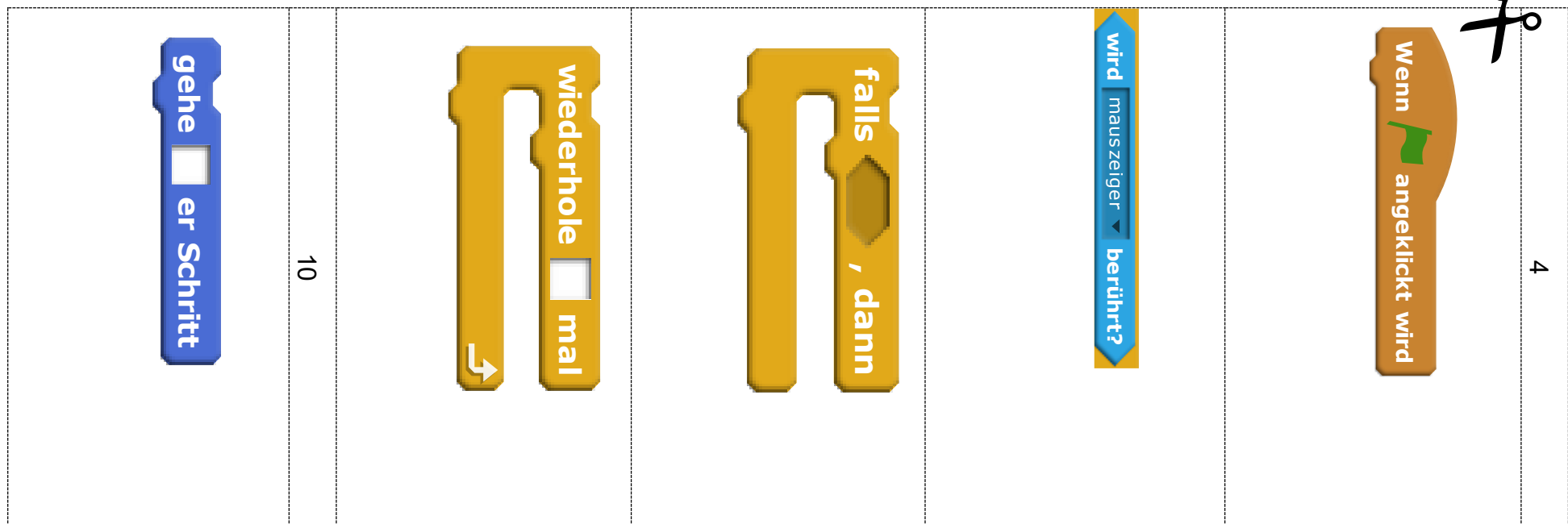
**Aufgabe 1:** Schneidet die Anweisungskarten an der gestrichelten Linie aus. Jede Karte steht für eine Anweisung in der Programmiersprache Scratch.

**Aufgabe 2:** Legt die Anweisungskarten so zu einer Sequenz zusammen, damit die Scratch-Katze um 10 Schritte nach vorne läuft.

**Aufgabe 3:** Erstellt jetzt ein Programm und nutzt dabei die Verzweigung. Die Scratch-Katze soll nur dann 10 Schritte nach vorne laufen, wenn die Maustaste geklickt wurde.

**Aufgabe 4:** Die Scratch-Katze soll jetzt 40 Schritte nach vorne laufen. Schafft ihr es ein Programm zu schreiben, das das kann?

**Tipp:** Schaut auf unseren Wortspeicher. Dort steht, was die Informatik-Wörter bedeuten.



## Arbeitsblatt 2: Unsere ersten Schritte mit Scratch



**Aufgabe 1:** Startet Scratch mit einem Doppelklick auf das Katzensymbol.

**Aufgabe 2:** Programmiert eure drei Programme, die ihr mit den Scratch-Karten programmiert habt, jetzt am Computer in Scratch.






**Aufgabe 3:** Ändert das Kostüm der Katze. Schafft ihr es auch, der Bühne ein anderes Aussehen zu geben?

**Aufgabe 4:** Stellt zunächst Vermutungen darüber an, was andere Blöcke bedeuten können. Probiert sie anschließend aus. Habt ihr das vermutet?

## Block-Karten für Erläuterungen im Plenum

4
Wenn  angeklickt wird
wird <input type="text" value="mauszeiger"/> berührt?
falls <input type="text" value=""/> , dann
wiederhole <input type="text" value=""/> mal
wiederhole <input type="text" value="3"/> mal
gehe <input type="text" value=""/> er Schritt
gehe <input type="text" value="7"/> er Schritt
10 

## Tipp-Karten

4	Die Zahl 4 könnt ihr in das freie, weiße Feld von orangenen Blöcken einsetzen.
	Wenn die grüne Flagge in Scratch angeklickt wird, dann wird euer Programm gestartet.
	Dies nennt man eine Bedingung. Hier wird geprüft, ob der Mauszeiger gedrückt wurde. Stimmt das, ist die Bedingung wahr. Stimmt es nicht, ist sie falsch. Es gibt noch viele andere Bedingungen.
	Alle Anweisungen, die zwischen der Verzweigung stehen, werden ausgeführt. Aber nur dann, falls die Bedingung eintritt. Die Bedingung steht im dunklen Feld.
	Alle Anweisungen, die vom Zählzyklus umschlossen werden, werden wiederholt. Die Zahl im weißen Feld gibt an, wie oft die Anweisungen wiederholt werden.
10	Die Zahl 10 könnt ihr in das freie, weiße Feld von blauen Blöcken einsetzen.
	Die Scratch-Katze bewegt sich 10 Schritte nach vorne.

  
 Bitte faltet hier!





## Literaturverzeichnis

Sämtliche Quellenangaben wurden zuletzt am 02.02.2021 geprüft.

Gesellschaft für Didaktik des Sachunterrichts (GDSU) (2013): Perspektivrahmen Sachunterricht. Zweite, vollständig überarbeitete und erweiterte Ausgabe. Kempten: Klinkhardt.

Gesellschaft für Informatik (GI) (2019): Kompetenzen für informatische Bildung im Primarbereich. Empfehlungen der Gesellschaft für Informatik e. V. erarbeitet vom Arbeitskreis »Bildungsstandards Informatik im Primarbereich«. Die Empfehlungen wurden am 31. Januar 2019 vom Präsidium der GI verabschiedet. In: LOG IN (Beilage) 39 (191/192), I-28. Online verfügbar unter <https://dl.gi.de/handle/20.500.12116/29621>.