

# Spielend Programmieren lernen mit Scratch



[1]

**Verfasser:** Kathleen Hartkopf, Laura Tiemann, Georg Mierau, Jan Schulz, überarbeitet von Evelyn Willam, Jan Nowak und Christina Schramm

**Kurz-Info:**

Informatischer Inhalt: Grafische Programmierung mit Scratch

Jahrgangsstufe: 6 bis 8

Vorwissen: keins

## KURZINFORMATION FÜR DIE LEHRKRAFT

**Titel:** *Spielend Programmieren lernen mit Scratch*

**Schulstufe:** *Unter- und Mittelstufe*

**Optimale Jahrgangsstufe:** *Klasse 6 bis 8*



[2]

**Themenbereich:** *Grafische Programmierung, Variablen, Kontrollstrukturen, Bildbearbeitung*

### EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

**Bildungsstandards der GI:** *Modellieren und Implementieren, Begründen und Bewerten, Strukturieren und Vernetzen, Kommunizieren und Kooperieren, Algorithmen, Informatiksysteme*

**Einbindung in den Unterricht:** *Das Modul bietet sich ideal an, um Schülerinnen und Schüler mit viel Freude an das Programmieren heranzuführen. Hierzu wird die grafische Programmierungsumgebung des MIT-Scratch-Editors (<https://scratch.mit.edu>) genutzt, die vorgefertigte Programmier-elemente in Form von Puzzleteilen/Blöcken bereitstellt, welche dann entsprechend zusammengesetzt werden können.*

**Vorkenntnisse:** *keine*

**Dauer:** *ca. 5 Stunden*

**Inhaltsbeschreibung:** *Zu Beginn werden den Schülerinnen und Schülern die Grundelemente und die Handhabung der Entwicklungsumgebung vorgestellt.*

*In Expertenteams sammeln die Schülerinnen und Schüler dann anhand von kleinschrittigen Anleitungen in Form von Arbeitsblättern „Expertenwissen“ zu einem von vier unterschiedlichen Grundlagenschwerpunkten wie z. B. Variablen. Dieses Expertenwissen müssen sie im Anschluss in ihren Stammgruppen anwenden, um gemeinsam ein kleines, fantasievolles Spiel zu kreieren.*

## INHALT

Kurzinformation für die Lehrkraft.....	2
Lernziele .....	4
Fachliche Analyse .....	4
Einordnung in gesetzliche Rahmenbedingungen – Bildungsstandards der GI .....	5
Einbettung in den Schulunterricht .....	7
Didaktische/Methodische Schwerpunktsetzung.....	7
Beschreibung des Modulablaufs .....	7
Begründung des Methoden- und Medieneinsatzes.....	8
Didaktische Prinzipien nach Baumann und Hubwieser .....	8
Verlaufsplan des Moduls.....	9
Quellenverzeichnis .....	10
Abbildungsverzeichnis .....	10
Anhang .....	11

## LERNZIELE

Die Schülerinnen und Schüler...

- entwerfen unter der Verwendung des Scratch-Editors Programme, in denen sie die grundlegenden Konzepte der Schleifen, Algorithmen und Kontrollstrukturen anwenden.
- analysieren und erklären entworfene Programme.
- beschreiben ihre Programme unter Verwendung der Fachsprache.
- arbeiten nach grundlegenden Prinzipien des Projektmanagements (in Form eines Gruppenpuzzles) und setzen diese um.
- bedienen sich simpler Methoden, um mit dem Computer zu interagieren.
- präsentieren ihre Ergebnisse und begründen dabei ihr Vorgehen bei der Programmierung.

## FACHLICHE ANALYSE

Scratch ist ein Editor mit einer dazugehörigen Online-Gemeinschaft, der von der *Lifelong-Kindergarten-Group* am Media-Lab des MIT entwickelt wurde und das Ziel verfolgt, Kinder an das Programmieren heranzuführen, welches ein wichtiger Bestandteil der modernen Gesellschaft geworden ist. Dabei soll vor allem das kreative und systematische Denken sowie kooperatives Arbeiten gefördert werden. Die Hauptzielgruppe des Editors sind 8- bis 16-Jährige, was mit der Altersgruppe, für die dieses Modul entwickelt wurde, weitestgehend korreliert. Tatsächlich belegen die vom MIT veröffentlichten Statistiken zur „Altersverteilung neuer Scratcher“<sup>1</sup>, dass der Großteil der Nutzerinnen und Nutzer zwischen 8 und 18 Jahre alt ist. Darüber hinaus erfreut der Editor sich jedoch auch jenseits dieser Altersgrenzen weltweit großer Beliebtheit.

Die Vorteile des Scratch-Editors sind, dass er browserbasiert, kostenlos zugänglich und einfach zu bedienen ist. Abgesehen von einer rudimentären Kompetenz im Umgang mit dem Computer (Tastatur, Maus, Browser) müssen die Nutzerinnen und Nutzer keinerlei Vorwissen mitbringen. Auch eine (textuelle) Programmiersprache muss im Vorfeld nicht erlernt werden, da der Text zur Programmierung in Scratch nicht selbst geschrieben werden muss. Die Lernenden werden vielmehr intuitiv an die Grundzüge der Programmierung herangeführt, indem sie bereits fertige Code-Puzzle-Blöcke zu einem Programm zusammenfügen. Diese Form der grafischen Programmierung bietet u. a. den Vorteil, dass es keiner syntaktischen Fehlerüberprüfung bedarf, weil syntaktische Fehler schlicht nicht möglich sind. Wie bei einem gewöhnlichen Puzzle können nämlich nur Blöcke zusammengesteckt werden, die zueinander passen. Zusätzlich sind die Blöcke im Menü im Bereich Skripte in mehrere Kategorien (Bewegung, Aussehen, Klang etc.) unterteilt, um eine bessere Orientierung zu ermöglichen.

Neben dem Menü und der Programmieroberfläche, in der die Blöcke zusammengesetzt werden, verfügt der Editor über eine Bühne, auf der das geschriebene Programm ausgeführt wird. Diese Bühne lässt sich durch eine Vielzahl von Bühnenbildern ebenso individuell gestalten wie die

---

<sup>1</sup>Scratch-MIT-Edu: *Statistiken*. Abgerufen von: <https://scratch.mit.edu/statistics/> (eingesehen: 23.04.2019).

zahlreichen Figuren, deren Aussehen über den Kostüm-Bereich des Menüs verändert werden kann. In diesem Bereich werden vor allem die Mitglieder der Expertengruppe *Grafik und Aussehen* tätig. Das Tätigkeitsfeld dieser Expertengruppe lässt sich außerdem heranziehen, um beispielhaft Verknüpfungen zum Fach Mathematik (fächerübergreifendes Lernen) zu demonstrieren. So orientiert die Bühne sich an einem Koordinatenkreuz und die Position der Figuren wird stets über eine x- und eine y-Koordinate angegeben. Auch die Verwendung der (Vergleichs-)Operatoren in diesem Kontext dürfte den Schülerinnen und Schülern aus ihrem Mathematikunterricht bekannt sein.

Neben den speziellen Themenbereichen der Expertengruppen begegnen den Schülerinnen und Schülern grundlegende informatische Phänomene: Das Prinzip der Schleifen müssen die Lernenden immer wieder anwenden, um ein Programm oder einzelne Teile zu wiederholen. Eine herausragende Bedeutung kommt dabei der Wiederhole-fortlaufend-Schleife zu, die als einer der Grundbausteine der meisten Programme fungiert. Die Lernenden werden schnell erfassen, dass Programme, die diesen Grundbaustein nicht enthalten, oft nicht wie geplant funktionieren. Falls-dann-Anweisungen, die den Kindern ebenfalls begegnen, würden ohne Verwendung des Wiederhole-fortlaufend-Blocks nur einmal ausgeführt und somit nicht die gewünschte Funktion erfüllen. Eine Funktion im informatischen Sinne lernen die Schülerinnen und Schüler in Form so genannter Eventlistener kennen, die Events und die dazugehörigen Aktionen verknüpfen. So sollen Programme beispielsweise erst starten, wenn die grüne Fahne angeklickt wurde. Oder die Katze soll erst loslaufen, wenn die Leertaste gedrückt wurde. Mit dem Thema Variablen ist sogar eine ganze Expertengruppe betraut, die u. a. einen „Merkspeicher“ anlegt, der erfasst, wie oft eine Figur angeklickt/gestreichelt wurde.

Wenn die Schülerinnen und Schüler schließlich in ihren Stammgruppen ein gemeinsames Fangspiel programmieren, wird ihre Fähigkeit zum systematischen Denken – eines der von der *Lifelong-Kindergarten-Group* formulierten Leitziele – gefordert, denn in dieser Phase soll das gesamte neue Wissen gebündelt und produktiv eingesetzt werden, indem zahlreiche Menükategorien und damit auch Programmierschritte kombiniert werden.

## EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN – BILDUNGSSTANDARDS DER GI

- Modellieren und Implementieren
  - Implementieren Modelle mit geeigneten Werkzeugen
  - Reflektieren Modelle und deren Implementierung, bzw. Modellkritik
    - Überprüfung der Angemessenheit der Lösung
    - Bewertung der erreichten Resultate/Implementierung
- Begründen und Bewerten
  - Begründen Entscheidungen bei der Nutzung von Informatiksystemen (Schülerinnen und Schüler begründen bei der Präsentation ihrer Ergebnisse ihr Vorgehen)
    - Nennen Vor- und Nachteile

- Können Argumente nachvollziehen
- Begründen die Darstellung
  
- Strukturieren und Vernetzen
  - Strukturieren Sachverhalte durch zweckdienliches Zerlegen und Anordnen (Schülerinnen und Schüler zerlegen die Spielprogrammierung in einzelne Schritte, welche teils durch die Arbeitsmaterialien vorgegeben sind)
    - Zerlegen Sachverhalte durch Erkennen und Abgrenzen von einzelnen Bestandteilen
    - Erkennen Reihenfolgen in Handlungsabläufen
  
- Kommunizieren und Kooperieren (in Teams und Gruppenarbeit)
  - Kommunizieren fachgerecht über informatische Sachverhalte
    - Tauschen sich untereinander verständlich über informatische Inhalte aus
    - Stellen informatische Sachverhalte unter Benutzung von Fachbegriffen mündlich und schriftlich sachgerecht dar (denn später müssen die Schülerinnen und Schüler ihr Vorgehen präsentieren und begründen)
  - Kooperieren bei der Lösung informatischer Probleme
    - Kooperieren in verschiedenen Formen der Zusammenarbeit bei der Bearbeitung einfacher informatischer Probleme
    - Kooperieren in arbeitsteiliger Gruppenarbeit
    - Beschreiben die Bearbeitung und Ergebnisse in einem gemeinsamen Dokument (durch Gruppenpräsentation am Ende)
  
- Algorithmen
  - Kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten und lesen und interpretieren gegebene Algorithmen (durch Arbeitsblätter)
    - Lesen und verstehen Handlungsvorschriften für das Arbeiten mit Informationssystemen
    - Interpretieren Handlungsvorschriften korrekt und führen sie schrittweise aus
  - Entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar (Befehlsblöcke von Scratch)
  - Benutzen die algorithmischen Grundbausteine zur Darstellung von Handlungsvorschriften (Befehlsblöcke von Scratch)
    - Entwerfen Handlungsvorschriften als Text oder mit formalen Darstellungsformen (formale Darstellung durch das Zusammensetzen der Bausteine von Scratch)
    - Entwerfen und testen einfache Algorithmen (Schülerinnen und Schüler erstellen zuerst kleinere Programme und schließlich das Spiel)

- Informatiksysteme
  - Wenden Informatiksysteme zielgerichtet an
    - Arbeiten mit grafischen Benutzungsoberflächen (Scratch)

## EINBETTUNG IN DEN SCHULUNTERRICHT

Dieses Modul kann als Einstieg in die (grafische) Programmierung genutzt werden. Im weiteren Verlauf kann auf andere grafische Programmierumgebungen wie z. B. den MIT-App-Inventor (<http://appinventor.mit.edu/>) zurückgegriffen werden, zu dem ebenfalls InfoSphere-Module angeboten werden.

## DIDAKTISCHE/METHODISCHE SCHWERPUNKTSETZUNG

### BESCHREIBUNG DES MODULABLAUFS

#### Phase 1

Zu Beginn des Moduls wird den Schülerinnen und Schülern der Scratch-Editor präsentiert, wobei die Programmieroberfläche mit ihren einzelnen Komponenten und Funktionen, inklusive ihrer visuellen Programmiersprache, durch Beispiele anschaulich gemacht wird. Des Weiteren werden die Schülerinnen und Schüler darüber in Kenntnis gesetzt, dass das Modul in Form eines Gruppenpuzzles stattfinden wird: Hierzu werden die Schülerinnen und Schüler zunächst in Stammgruppen eingeteilt. Innerhalb dieser Stammgruppen entscheiden sie im Rahmen einer kurzen Besprechung dann eigenständig, welches Stammgruppenmitglied zu welchem der vier möglichen Experten (BZ, GA, KK, FV)<sup>2</sup> „ausgebildet“ werden soll. Nach Abschluss der Ausbildung zum Experten kommen die Kursteilnehmerinnen und -teilnehmer dann erneut in ihren Stammgruppen zusammen, in denen sie ihr Expertenwissen austauschen und gemeinsam ein Fangspiel entwickeln.

#### Phase 2

Die Schülerinnen und Schüler finden sich in den Expertengruppen zusammen und beginnen mit der Bearbeitung der ersten Arbeitsblätter in Partnerarbeit. Durch die kleinschrittigen Anleitungen auf den Arbeitsblättern lernen sie alle Grundlagen, die sie als ausgebildete Experten benötigen, um in der späteren Gruppenphase sicher anhand eines weiteren Arbeitsblattes das bisher Gelernte anwenden und umsetzen zu können. In dieser Phase gilt es zu beachten, dass idealerweise alle Schülerinnen und Schüler einer Expertengruppe am selben Tisch sitzen, sodass sich die Zweier-Gruppen auch untereinander austauschen und unterstützen können. Eine darauffolgende Sicherungsphase, zu der sowohl die Sicherungsfragen des jeweiligen Arbeitsblattes als auch eine individuelle Besprechung des bisher erstellten Programmes mit einer betreuenden Person gehören, festigt den neuerlernten Stoff. Hierbei sollte darauf geachtet werden, dass sich alle Schülerinnen und Schüler auf einem soliden Wissensstand befinden, sodass sie sich in ihrer künftigen Gruppe gut als Experte einbringen können und ihren Wissensstand entsprechend teilen – denn nur jeweils ein Experte

<sup>2</sup> Arbeitsblätter Bewegung und Zeichnen (BZ), Grafik und Aussehen (GA), Klang und Kommunikation (KK) und Fühlen und Variablen (FV)

innerhalb der Stammgruppe verfügt über den Wissensstand eines Themenbereichs und trägt damit eine hohe Verantwortung.

### Phase 3

Nach einer Pause kommen die Kursteilnehmerinnen und -teilnehmer wieder in ihren Stammgruppen zusammen und erstellen mithilfe eines weiteren Arbeitsblattes nach dem *Prinzip der abnehmenden Hilfe* ein Fangspiel. Die hierbei geschaffenen Resultate werden am Ende des Workshops von den Entwicklerteams präsentiert, bevor eine Feedbackrunde den Workshop abschließt.

#### BEGRÜNDUNG DES METHODEN- UND MEDIENEINSATZES

Im InfoSphere – Schülerlabor Informatik ist eines der wichtigsten Prinzipien, dass die Schülerinnen und Schüler selbstständig arbeiten, um informatische Probleme zu lösen. Dabei sollen sie in Teams möglichst ohne direkte Anleitung durch eine Lehrperson arbeiten und sich während des Lernprozesses gegenseitig unterstützen. Um dies zu gewährleisten, wurden für dieses Modul Arbeitsblätter entwickelt, die die Schülerinnen und Schüler durch die einzelnen Phasen führen. Dabei ist die Rolle der Betreuerinnen und Betreuer eine rein beratende. Um trotzdem den zeitlichen Rahmen und vor allem das Erfolgserlebnis der Teilnehmerinnen und Teilnehmer gewährleisten zu können, wird auf die kurze Einstiegspräsentation der Programmieroberfläche zurückgegriffen. Bei der Programmieroberfläche Scratch handelt es sich um eine Drag-and-Drop-Oberfläche, die zudem ihre eigene Programmiersprache (Blöcke) mit selbigem Namen stellt.

#### DIDAKTISCHE PRINZIPIEN NACH BAUMANN UND HUBWIESER

Da sich dieses Modul nach dem Konzept des Schülerlabors InfoSphere richtet, liegt ein besonderer Fokus auf dem *Prinzip des aktiven Lernens*. Durch die Arbeitsmaterialien ist ein selbstständiges und aktives Lernen der Schülerinnen und Schüler gegeben. Sie müssen selber aktiv werden, um Probleme zu lösen. Während der Begrüßung der Schülerinnen und Schüler (Phase 1) wird auch auf den Ablauf des Moduls eingegangen. Hier werden ihnen die Ziele des Moduls nähergebracht, um einen möglichst hohen Grad an Transparenz auf Schülerseite zu garantieren. Auch auf den Arbeitsmaterialien (Phase 2) werden die jeweiligen Ziele und Teilziele bekanntgegeben. Hierdurch wird das Prinzip der Zielvorstellung erfüllt.

Durch die relativ freie Aufgabenstellung in der Erarbeitungsphase der Stammgruppen (Phase 3), der Gestaltung des Fangspiels und der Möglichkeit, bestimmte Teile des Fangspiels auszulassen bzw. als Bonus zusätzlich zu integrieren, wird auf das individuelle Lerntempo der Schülerinnen und Schüler eingegangen. Diese können neben dem verlangten Pflichtteil des Fangspiels individuell dessen Komplexität erhöhen oder sich bei Bedarf bei den betreuenden Personen Hilfe holen.



**VERLAUFSPLAN DES MODULS**

Zeit	Phase	Inhalt	Medium	Sozialform
0:00-0:05	Begrüßung	<i>Vorstellung InfoSphere und Betreuer team</i>		LV
0:05-0:25	Einführung	<i>Vorstellung Oberfläche, Stammgruppeneinteilung</i>	Smartboard/Beamer, PC, Leitfaden	LV
0:25-0:30	Gelenkstelle	<i>Aufteilung der Expertenrollen innerhalb der Stammgruppe</i>		GA
0:30-1:40	Erarbeitung I	<i>Expertengruppen: Erstellung eines ersten Programms (gemäß jeweiligem Arbeitsblatt)</i>	AB, Gruppentisch pro Expertenarbeitsblatt	PA, GA
1:40-1:50	Sicherung I	<i>Besprechung der Sicherungsfragen oder persönliche Besprechung des bisherigen Programms/Programmteils (jeweils tisch- bzw. teamweise)</i>	AB	D
1:50-2:05	Pause			
2:05-2:15	(Warmup-Spiel)			
2:15-3:55	Erarbeitung II	<i>Stammgruppen: Programmierung des Fangspiels</i>	AB	GA
3:55-4:05	Pause			
4:05-4:45	Sicherung II	<i>Vorstellung der Ergebnisse</i>	Beamer	SV
4:45-5:00	Verabschiedung	<i>(u. U. Urkundenverleihung) Feedbackrunde</i>		UG

Legende: Arbeitsblatt (AB), Lehrervortrag (LV), Unterrichtsgespräch (UG), Einzelarbeit (EA), Partnerarbeit (PA), Gruppenarbeit (GA), Schülervortrag (SV), Diskussion (D)

## QUELLENVERZEICHNIS

- Gesellschaft für Informatik e.V. (GI) (Hg.) (2008): *Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I*. Gesellschaft für Informatik e.V. (GI). Abgerufen von: [http://www.fa-ibs.gi-ev.de/fileadmin/gliederungen/fb-iad/fa-ibs/Empfehlungen/bildungsstandards\\_2008.pdf](http://www.fa-ibs.gi-ev.de/fileadmin/gliederungen/fb-iad/fa-ibs/Empfehlungen/bildungsstandards_2008.pdf) (eingesehen: 29.06.2015).
- Scratch-MIT-Editor. Abgerufen von: <http://scratch.mit.edu> (eingesehen: 09.04.2019).
- Scratch-MIT-Edu: *Statistiken*. Abgerufen von: <https://scratch.mit.edu/statistics/> (eingesehen: 23.04.2019).

## ABBILDUNGSVERZEICHNIS

**Abb. 1-2** – Quelle: Screenshot des MIT-Scratch-Editors (<https://scratch.mit.edu/>), CC BY-SA 2.0 Attribution-ShareAlike 2.0 Generic (<https://creativecommons.org/licenses/by-sa/2.0/>), erstellt am: 24.07.2023.

## ANHANG

- Anleitung für den Scratch-Editor: Scratch-online-Editor.pdf
- Scratch\_Expertenschilder.pdf
- Arbeitsblätter:
  - Gruppenpuzzle – Expertengruppe:
    - Scratch\_Gruppenpuzzle-Bewegung+Zeichnen.pdf
    - Scratch\_Gruppenpuzzle-Fuehlen+Variablen.pdf
    - Scratch\_Gruppenpuzzle-Grafik+Aussehen.pdf
    - Scratch\_Gruppenpuzzle-Klang+Kommunikation.pdf
  - Gruppenpuzzle-Stammgruppe: Fangen.pdf
- Musterlösungen für Betreuerinnen und Betreuer:
  - Musterloesungen-Gruppenpuzzle.pdf
  - Musterloesung-Fangen.pdf