

# Scratch

Dieses Unterrichtspaket beinhaltet flexibel gestaltete Stundenbilder (samt zugehörigem Material), die Schüler\_innen mit ihrer ersten Programmiersprache (Scratch) samt einigen Programmierkonzepten vertraut machen soll. Insgesamt befinden sich auf den nachfolgenden Seiten fünf Stundenbilder (5x 100 Min.). Die gesamte geplante Unterrichtszeit beläuft sich also auf 500 Minuten oder 10 Unterrichtsstunden.

Es werden keine Informatik-Kenntnisse vorausgesetzt, jedoch sollten die Schüler\_innen bis zu einem gewissen Grad vertraut mit dem Umgang mit einem PC und dem Internet sein (USB-Stick, E-Mails versenden, etc.). Als Zielgruppe wurde eine 5. Klasse AHS oder höher gewählt (14 Jahre aufwärts).

## Allgemeine Ziele

- Schüler\_innen lernen grundlegende Programmierkonzepte (Wenn-Dann-Abfragen, Schleifen, Variablen, I/O kennen)
- Schüler\_innen haben mehrere Scratch-Projekte modifiziert oder erweitert und diese Neukreationen allesamt öffentlich zugänglich gemacht
- Schüler\_innen haben am Ende ein eigenes Spiel in Scratch entwickelt
- Schüler\_innen haben ihr eigenes Scratch-Spiel auf <https://scratch.mit.edu/> veröffentlicht

## Vormerkungen

Während jeder Unterrichtseinheit sollte die Lehrkraft regelmäßig die wichtigsten Begriffe, Konzepte, Erkenntnisse, etc. auf die Tafel schreiben. Schüler\_innen sollen diese in einem kollaborativ angelegten Online-Dokument (z.B. Etherpad, Google Drive, ...) zusammentragen und durch eigene Notizen ergänzen. Diese Mitschrift dient als Lernunterlage für einen Test und als ständige Wissens-Ressource während der vielen Projektarbeiten, auf die zu jeder Zeit zurückgegriffen werden kann (und soll).

Lehrkräfte sollen zudem sicherstellen, dass sie von ihren Schüler\_innen die Scratch-Account kennen und dass alle Schüler\_innen alle Scratch-Projekte, die im Rahmen dieser Unterrichtssequenz stattfinden, unter diesem Account anlegen und auch öffentlich zugänglich machen.

Dieses Material wurde von Andreas Schuch (schuch.andreas@gmail.com) erstellt und steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/4.0/>.

Es soll zudem angemerkt werden, dass die hier vorgelegten Stundenbilder nicht in Stein gemeißelt sind. Sie zeigen lediglich eine Variante von vielen auf, wie der Unterricht gestaltet werden kann. Die Stundenbilder müssen für jede Klasse leichter oder stärker angepasst werden. Wenn eine Klasse langsamer bzw. schneller arbeitet, als geplant, können natürlich gewisse Themenbereiche nach hinten oder vorne geschoben, zusätzlich hinzugefügt oder stattdessen entfernt werden.

## Stundenbilder

### Einführung in das Programmieren

<b>Stundenbild-ID</b>	Scratch/1
<b>Dauer</b>	100 Min.
<b>Thema</b>	Programmieren kennenlernen und algorithmische Denkweise üben
<b>Unterrichtsziele</b>	<ul style="list-style-type: none"><li>• Alle Schüler_innen haben die Definition von „Programmieren“ aufgeschrieben und in der Klasse diskutiert</li><li>• Alle Schüler_innen haben mindestens 20 Bereiche aufgeschrieben, in denen Informatik eine wichtige Rolle spielt</li><li>• Alle Schüler_innen haben alle Level des Blockly-Programms „Labyrinth“ gemeistert</li><li>• Alle Schüler_innen haben alle Level des Blockly-Programms „Vogel“ gemeistert</li></ul>
<b>Hilfsmittel</b>	Tafel, Beamer, Internet, Browser, USB-Stick

### Einleitung

1. Schüler\_innen bilden 2er-Gruppen. Lehrkraft schreibt folgende Fragen auf die Tafel: „Was ist Programmieren? Erkläre den Begriff in 2-3 Sätzen.“, „Was weißt du schon über das Programmieren?“ und „Wofür braucht man es? Finde mindestens 20 Bereiche und Beispiele, in denen Informatik eine wichtige Rolle spielt.“ Nach ca. 10-15 Minuten Zeit wird diese Aufgabe beendet oder abgebrochen und die bisherigen Ergebnisse werden im Plenum diskutiert. Auf der Tafel werden wichtige Begriffe und Beispiele festgehalten. Schüler\_innen sollen ein Gefühl dafür bekommen, dass Informatik in (fast) allen Bereichen des modernen, westlichen Lebensstils eine wichtige Rolle spielt

und dass in sehr vielen Berufszweigen Informatik-Kenntnisse wichtig, wenn nicht sogar notwendig, sind.

2. Alternativ oder auch zusätzlich zum oberen Punkt kann die folgende Präsentation verwendet werden, um verschiedene Anwendungsbereiche der Informatik zu erforschen: „What is Programming?“ [1]

### Stundenkern / Ertragssicherung

3. Schüler\_innen sollen in dieser Einheit mit der (zur menschlichen Denkweise sehr unterschiedlichen) linearen Denkweise des Computers und seinen wortwörtlichen Interpretationen und seiner exakten Arbeitsweise umgehen können. Dieses Ziel soll über das Web-App *Blockly* erreicht werden. Hier haben Schüler\_innen eine anfangs stark limitierte Anzahl an Befehlen zur Verfügung, die sie in der richtigen Sequenz anordnen müssen, um ein konkretes Problem zu lösen. *Blockly* präsentiert grundlegende Konzepte der Informatik (z.B. Schleifen und if-Abfragen) und Probleme der Informatik (z.B. Navigation) innerhalb eines Spielerahmens. Schüler\_innen können einen Level lösen, indem sie beispielsweise die Verhaltensweise einer Figur, die den richtigen Weg zum Ziel finden muss, richtig programmieren. Jeder Level ist komplexer als der vorige, indem Probleme langsam, aber stetig komplexer werden und die Anzahl der Befehle, die verwendet werden können, zunimmt.
4. Um Schüler\_innen mit dem Steckkastensystem von *Blockly* vertraut zu machen (später auch relevant für *Scratch*), werden Schüler\_innen zuerst angehalten, die folgende Webseite zu öffnen: <https://blockly-games.appspot.com/puzzle?lang=de>. Schüler\_innen haben 2-3 Minuten Zeit, um die Bilder und Worte richtig zuzuordnen. Danach sollen sie auf den roten Button rechts oben klicken und ihre Punkteanzahl herausrufen. Sofern jemand deutlich weniger Punkte hat als die anderen, kann hier bereits relativ früh interveniert werden.
5. Die Lehrkraft zeigt am Beamer vor, wie man bereits „verschachtelte“ Boxen wieder voneinander lösen kann. Zusätzlich wird darauf hingewiesen, dass es verschiedene Stecksysteme gibt und Blöcke in einen ganz bestimmten Steckplatz gehören. Es ist wichtig, dass alle Schüler\_innen mit *Blockly* umgehen können, bevor weiter mit *Blockly* gearbeitet wird.
6. Nun sind die Schüler\_innen bereit, die erste große Aufgabenstellung zu meistern. Schüler\_innen sollen folgenden Link öffnen: <https://blockly-games.appspot.com/maze?lang=de>. Wenn alle so weit sind, werden die einzelnen Bestandteile des Interface erklärt. Danach werden die ersten drei Level zusammen mit

den Schüler\_innen gelöst. Die Lehrkraft lässt sich von **verschiedenen** Schüler\_innen eine Schritt-für-Schritt-Anleitung geben, in welcher Reihenfolge die Befehle verkettet werden sollen. Dabei soll nicht dafür zurückgeschreckt werden, auch falsche Programme auszuführen.

7. Die restlichen Levels (4-10) sollen Schüler\_innen selbständig lösen. Sie sollen ermutigt werden, oft das Programm auszuführen—auch wenn es vielleicht noch unvollständig ist—um ein „Gefühl“ für den linearen Denkprozess des Computers zu bekommen. Während die Schüler\_innen selbständig daran arbeiten, die verschiedenen Levels zu lösen, geht die Lehrkraft durch die Reihen und gibt Hilfestellungen. Vor allem lernschwächere Schüler\_innen müssen hier unterstützt werden, damit sie nicht zu weit zurückfallen.
8. Sobald einzelne Schüler\_innen Level 10 des Labyrinths erfolgreich abgeschlossen haben, werden sie weitergeleitet zum Vogel-Programm: <https://blockly-games.appspot.com/bird?lang=de>. Hier sollen sie wieder selbständig die 10 Level lösen, wobei die Lehrkraft Hilfestellung gibt.
9. Ziel ist es, dass alle Schüler\_innen die 10 Level des Labyrinth-Programms und des Vogel-Programms gemeistert haben.
10. Besonders schnelle Schüler\_innen können noch weitere Programme erforschen:
  - a. Schildkröte (Linien zeichnen) <https://blockly-games.appspot.com/turtle?lang=de>
  - b. Film (Formen zeichnen) <https://blockly-games.appspot.com/movie?lang=de>
  - c. Teich (simple Künstliche Intelligenz programmieren): <https://blockly-games.appspot.com/pond-basic?lang=de>

## Hausübung

11. keine

## Einführung in Scratch

Stundenbild-ID Scratch/2

Dauer 100 Min.

Thema Scratch-Projekte analysieren und ändern

Unterrichtsziele • Alle Schüler\_innen haben die Levels 8-10 in den Blockly-Spielen

Dieses Material wurde von Andreas Schuch (schuch.andreas@gmail.com) erstellt und steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/4.0/>.

„Labyrinth“ und „Vogel“ gemeistert

- Es wurden mindestens fünf verschiedene Scratch-Projekte analysiert und geändert oder neue Objekte/Funktionen hinzugefügt
- Das beim Analysieren und Ändern gesammelte Wissen wurde von der Lehrkraft und den Schüler\_innen in einer Mitschrift festgehalten
- Alle Schüler\_innen haben ein Scratch-Projekt selbständig gewählt, mindestens 3 Aspekte geändert und die Funktionsweise und die vorgenommenen Änderungen dem Rest der Klasse präsentiert

**Hilfsmittel** Beamer, Internet, Browser, USB-Stick

## **Einleitung**

1. Um das Wissen der letzten Einheit wieder zu aktivieren, sollen Schüler\_innen innerhalb eines strikten Zeitlimits (z.B. 10 Minuten) Levels 8-10 des Blockly-Spiels „Labyrinth“ (<https://blockly-games.appspot.com/maze?lang=de>) und Levels 8-10 des Spiels „Vogel“ (<https://blockly-games.appspot.com/bird?lang=de>) meistern. Da sie die Levels in der letzten Einheit bereits gelöst haben sollten, darf das Zeitfenster ruhig straff sein.

## **Stundenkern**

2. Die Energie und Motivation der letzten Einheit (durch die Blockly-Spiele) soll nun verwendet werden, um die Scratch-Programmiersprache einzuführen. Dabei ist es wichtig, dass der Fokus nicht auf theoretische und abstrakte Informatik-Konzepte gelegt wird, da jegliches aufgebautes Momentum so wieder vermutlich abflacht. Auch sollte darauf verzichtet werden, sofort ein neues Projekt anzulegen, da die anfängliche Leere des Projekts schnell motivationshemmend wirken kann. Häufige Iterationen und das Streben nach möglichst raschen Ergebnissen führen nicht nur in der Softwareentwicklung zu produktiveren Ergebnissen, sondern auch in der Didaktik. In dieser Einheit sollen Schüler\_innen zusammen mit der Lehrkraft verschiedene existierende Scratch-Projekte analysieren und daran experimentieren. Ziel ist zum einen, dass Schüler\_innen einen Überblick erhalten, was sie alles mit Scratch machen könn(t)en und in welchem Umfang. Zum anderen bietet diese Herangehensweise auch eine Art Orientierungsfunktion, da Schüler\_innen bereits funktionierende Programme von „innen“ betrachten.

3. Auf dem Beamer vorzeigen, wie man auf der Scratch-Webseite <https://scratch.mit.edu/> anlegt. Schüler\_innen kopieren die Schritte und geben anschließend der Lehrkraft ihre Accounts bekannt.
4. Im Anschluss werden mindestens fünf Scratch-Projekte zusammen mit den Schüler\_innen untersucht. Das jeweilige Projekt kann am Beamer vorgezeigt werden, aber Schüler\_innen sollen dasselbe Projekt auch in ihrem Browser geöffnet haben.

Anmerkung: Es muss darauf geachtet werden, dass **verschiedene Arten oder „Genres“** von Scratch-Programmen betrachtet werden: **Spiele, Animationen, Geschichten...** Manche Schüler\_innen werden vermutlich bevorzugen, verschiedene Scratch-Spiele anzuschauen, während andere sich mehr für Animationen oder Geschichten begeistern können. Wichtig ist, Schüler\_innen eine große Bandbreite an unterschiedlichen Scratch-Programmen vorzuführen. Dies erhöht die Chance, dass alle Schüler\_innen mindestens einen Programmtyp finden, mit dem sie gerne arbeiten würden.

5. Am besten werden die Programme schon im Vorhinein (z.B. am Vortag) ausgesucht, um längere Suchprozesse während des Unterrichts zu vermeiden. Es sollte eine Mischung aus simplen und komplexeren Scratch-Projekte vorgezeigt werden, wobei auch Spiele, Animationen und Geschichten gemischt werden sollten. Die Scratch-Projekte können sowohl vollständig als auch unvollständig implementiert sein! Eine Auswahl an möglichen Projekten:
  - a. GHOSTS (Geschichte): <https://scratch.mit.edu/projects/41320614/> (einfach)
  - b. Kitty cat keep away (Spiel): <https://scratch.mit.edu/projects/47379772/> (einfach)
  - c. Girl on the Bench (Animation) <https://scratch.mit.edu/projects/89275010/> (einfach)
  - d. Help the wolf! (Spiel) <https://scratch.mit.edu/projects/91977460/> (einfach)
  - e. Labyrinth (Spiel): <https://scratch.mit.edu/projects/62020732/> (einfach)
  - f. Apple Catcher (Spiel) <https://scratch.mit.edu/projects/91953452/> (komplexer)
  - g. Flooder (Spiel) <https://scratch.mit.edu/projects/90690167/> (um zu zeigen, was mit Scratch alles möglich ist)
6. Wichtig ist zu erwähnen, dass man jeweils noch nicht das gesamte Programm verstehen muss. Das Scratch-Projekt kann im Gesamten zusammen analysiert werden oder auch nur Teile davon (besonders, wenn es sich um ein komplexeres Projekt

handelt). Schüler\_innen sollen ermutigt werden, sich nicht von vielen Codezeilen erschrecken zu lassen oder unbedingt 100% der Codezeilen jetzt verstehen zu müssen.

7. Es ist zudem wichtig, dass die Lehrkraft regelmäßig in einem Dokument die wichtigsten Erkenntnisse und Notizen mitschreibt. Schüler\_innen schreiben diese ab. Diese Mitschrift stellt für Schüler\_innen eine Lerngrundlage dar und zeigt der Lehrkraft, welche Konzepte der Informatik bereits behandelt wurden und in welchem Ausmaß.
8. Die vorgezeigten Beispielprojekte können auch relativ rasch abgeändert werden. Schüler\_innen sollen früh motiviert werden, kleine Änderungen und Experimente an existierendem Code durchzuführen. Betonen, dass oft und ausgiebig getestet werden soll, auch bei kleinsten Änderungen!
9. Nachdem mehrere Programme bereits analysiert und leichte Änderungen am Code durchgeführt wurden, beginnt die Lehrkraft nun, neue Objekte einzufügen und neuen (ganz simplen) Code zu schreiben. In diesem Zusammenhang wird auch der Scratch Mal-Editor vorgezeigt und erklärt sowie Schritt-für-Schritt gezeigt, wie man Bilder in Scratch importiert. Schüler\_innen machen nach

### Ertragssicherung

10. In den letzten 30 Minuten der Unterrichtseinheit sollen Schüler\_innen selbständig ein eigenes Scratch-Projekt auswählen, welches sie analysieren und ändern (ca. 10-15 Minuten). Sie müssen mindestens 3 Aspekte entweder ändern oder neu hinzufügen und anschließend dem Rest der Klasse am PC vorführen (die Klasse versammelt sich um den PC herum). Es muss die Funktionsweise des Programms erklärt werden sowie die Änderungen, die durchgeführt wurden.

### Hausübung

11. Ähnlich zu Ertragssicherung. Aufgabenstellung: Suche ein **neues** Scratch-Projekt, das mindestens 40 Codezeilen hat. Versuche zu verstehen, wie es funktioniert. Führe mindestens drei Änderungen durch und präsentiere es in der nächsten Stunde. (Falls es zu kompliziert ist, suche dir ein anderes Projekt!)

### Text-Adventure

Stundenbild-ID Scratch/3

Dauer 100 Min.

Thema Spiel zusammen mit Lehrkraft machen

Dieses Material wurde von Andreas Schuch (schuch.andreas@gmail.com) erstellt und steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/4.0/>.

- Unterrichtsziele**
- Alle Schüler\_innen haben mindestens zwei praktische Anwendungsmöglichkeiten von Variablen genannt
  - Alle Schüler\_innen haben ein Flow Chart zum Text-Adventure beschrieben
  - Alle Schüler\_innen haben ein Text-Adventure, das auf der Struktur des Flow Charts basiert, erstellt und auf der Scratch-Webseite veröffentlicht
  - Alle Schüler\_innen haben der Lehrkraft den Link zur jeweiligen Projekt-Seite bekanntgegeben

**Hilfsmittel**      Tafel, Beamer, Internet, Browser, USB-Stick

## **Einleitung**

1. Schüler\_innen präsentieren der Klasse die Hausübung der letzten Stunde (Scratch-Projekt ändern und erklären).

## **Stundenkern / Ertragssicherung**

2. Schüler\_innen werden mit dem Konzept der Variablen vertraut gemacht. Folgende Aspekte sollten auf jeden Fall besprochen und am Beamer jeweils testweise demonstriert werden:
  - a. Konzept einer Variable erklären: Variablen sind wie „Container“, die einen bestimmten Inhalt (Wert) haben, der ausgetauscht (überschrieben) werden kann.
  - b. Variablendefinition in Scratch
  - c. Rechenarbeit mit Variablen (z.B. Demonstration, wie man  $x = 5$  und  $y = 2$  addiert)
  - d. Schüler\_innen nach praktischen Anwendungsmöglichkeiten von Variablen fragen. Sie sollen mindestens zwei Beispiele finden (z.B. in Zusammenhang mit Scratch). 2-3 Minuten Zeit geben und dann Antworten auf der Tafel zusammentragen. Antwortmöglichkeiten könnten z.B. sein: Punkte zählen, User-Input abspeichern, relativen Abstand zwischen Objekten ausdrücken (z.B. `Spielcharakter_pos + 10`), etc.
  - e. Zusätzlich können die Schüler\_innen auch die von KhanAcademy gut gemachte Einführung zum Thema Variablen in JavaScript durcharbeiten (siehe

<https://en.khanacademy.org/computing/computer-programming/programmings>).

Allerdings nehmen Variablen in Scratch keinen so großen Stellenwert ein als in text-basierten Programmiersprachen, weshalb sich diese Übung nur dann lohnt, wenn geplant ist, nach Scratch auf eine „traditionellere“ Programmiersprache wie JavaScript, Java, C#, etc. zu wechseln.

3. Die Lehrkraft erstellt nun zusammen mit den Schüler\_innen ein Text-Adventure in Scratch.
  - a. Erklären, was ein Text-Adventure ist (Geschichte mit mehreren Verzweigungspunkten, die in gute und schlechte Enden überleiten)
  - b. Es ist empfehlenswert, Schüler\_innen darauf hinzuweisen, dass ihre Version des Text-Adventures am Ende auf Scratch veröffentlicht werden muss. Daher sollte es womöglich in englischer Sprache verfasst werden.
  - c. Als Denkanstoß für die Geschichte kann von Lehrerseite aus ein Startpunkt vorgegeben werden wie beispielsweise *Du bist in deinem Raum, wenn plötzlich...* Die Lehrkraft baut im Anschluss in Kooperation mit den Schüler\_innen eine Geschichte, die mehrere Verzweigungen hat und in mehreren Textdateien abgespeichert wird. Es sollte gleichzeitig ein Flow Chart auf der Tafel aufgezeichnet werden, das die grundlegende Struktur der Geschichte zeigt.
  - d. Um diesen Prozess zu beschleunigen, kann wahlweise aber auch den Schüler\_innen eine vorgefertigte Version des Text-Adventures vorgelegt werden (siehe Flow Chart unten und die Textdateien im Ordner „Text Adventure Beispiel“).



- e. Die Geschichte sollte auf jeden Fall in Textdateien ausgelagert werden, um das nachträgliche Editieren der Geschichte komfortabler zu gestalten und den Schüler\_innen gleichzeitig zu demonstrieren, wie einfache I/O-Operationen in Scratch funktionieren.
4. Die Lehrkraft zeigt den Schüler\_innen am Beamer vor, wie man Text-Dateien in Scratch einbindet. Die erste Verzweigung des Text-Adventures wird von der Lehrkraft am Beamer implementiert. Schüler\_innen werden danach aufgefordert, den Rest des Programms selbstständig fertigzustellen.
5. Während Schüler\_innen den Rest der Unterrichtsstunde damit verbringen, am Text-Adventure zu arbeiten, überprüft die Lehrkraft regelmäßig den Fortschritt der einzelnen Schüler\_innen und gibt individuelle Hilfestellungen.

## Hausübung

6. Schüler\_innen sollen, sofern das Text-Adventure nicht bis zum Ende der Einheit fertiggestellt wurde, dieses bis zu einer von der Lehrkraft vorgegebenen Frist selbst fertigzustellen und zu veröffentlichen. Die veröffentlichte Projekt-Seite soll der Lehrkraft per E-Mail gesendet werden.

## Eigenes Scratch-Spiel 1

Stundenbild-ID	Scratch/4
Dauer	100 Min.
Thema	Grundkonzepte der Programmierung
Unterrichtsziele	<ul style="list-style-type: none"><li>• Alle Schüler_innen haben der Lehrkraft eine Projektidee präsentiert und begonnen, diese zu implementieren</li></ul>
Hilfsmittel	Tafel, Beamer, Internet, Browser, USB-Stick

### Einleitung

1. Projektfindungsphase. Schüler\_innen entwerfen ein Scratch-Projekt. Entweder machen sie ein ganz neues Projekt erstellen oder ändern ein bestehendes Scratch-Projekt umfassend ab.
2. Die Schüler\_innen werden informieren, dass sie diese und nächste Einheit Zeit haben, ein Scratch-Projekt zu machen, das insgesamt aus mindestens 100 neu geschriebenen Codezeilen bestehen muss. Die endgültige Frist sollte jedoch so gesetzt sein, dass Schüler\_innen nach der zweiten Unterrichtseinheit noch Zeit haben, das Projekt zu Hause fertigzustellen, sofern sie in den beiden Unterrichtseinheiten nicht fertig geworden sind. Es kann natürlich noch eine dritte Einheit zur Verfügung gestellt werden und/oder das geforderte Mindestmaß an Codezeilen angepasst werden.
3. Schüler\_innen bekommen ein ca. 10 Minuten großes Zeitfenster, um sich ein Projekt für die nächsten zwei Unterrichtseinheiten auszusuchen. Es ist wichtig darauf hinzuweisen, dass sowohl Animationen als auch Geschichten und Spiele im Bereich der möglichen Projekte liegen. Sobald sie eine Idee haben, kommen Schüler\_innen zur Lehrkraft. Es wird dann abgeklärt, ob das Projekt auch tatsächlich realisierbar ist und ob etwaige Zielanpassungen vorgenommen werden müssen bzw. ob ein anderes Projekt gefunden werden sollte.

4. Mögliche Spielevorschläge für Schüler\_innen, die etwas Unterstützung bei der Suche nach einer Projektidee benötigen:

- a. Zum einen können Schüler\_innen ermutigt werden, ganz neue Projekte zu erstellen, wie z.B.
  - i. Ping Pong (für 2 Spieler)
  - ii. Hütchenspiel
  - iii. Finde die Fehler im Bild
  - iv. Alien Invader
  - v. Geschicklichkeitsspiel, das mit der Maus gesteuert wird (z.B. Hindernissen ausweichen)
- b. Zum anderen wäre es auch denkbar, existierende Scratch-Projekte zu suchen und sinnvoll zu erweitern, wie z.B.
  - i. Pong Starter <https://scratch.mit.edu/projects/10000036/>
  - ii. Labyrinth <https://scratch.mit.edu/projects/62020732/>
  - iii. Tipp: Auf der Scratch-Webseite kann man unter der Suche bei „Tags“ nach dem Stichwort „draft“ suchen. Dann werden unfertige Spiele und Entwürfe angezeigt, die Schüler\_innen vervollständigen könnten.

## Stundenkern

5. Schüler\_innen haben die ganze Unterrichtseinheit Zeit, um an ihrem Spiel zu arbeiten.

## Hausübung

6. keine

## Eigenes Scratch-Spiel 2

Stundenbild-ID Scratch/5

Dauer 100 Min.

Thema Grundkonzepte der Programmierung

Unterrichtsziele • Alle Schüler\_innen haben zumindest den Kern ihrer Projektidee fertig

Dieses Material wurde von Andreas Schuch (schuch.andreas@gmail.com) erstellt und steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/4.0/>.

implementiert

- Alle Schüler\_innen haben ihr (fast) fertiges Projekt der Klasse präsentiert
- Die Implementation aller Schüler\_innen hat jeweils mindestens 100 neu geschriebene Codezeilen
- Alle Schüler\_innen haben das Projekt auf der Scratch-Webseite veröffentlicht und der Lehrkraft den dazugehörigen Link geschickt

Hilfsmittel      Tafel, Beamer, Internet, Browser, USB-Stick

## Stundenkern

1. Schüler\_innen verbringen den Großteil der Unterrichtseinheit damit, an dem Scratch-Projekt, das sie in der letzten Stunde gestartet haben, weiterzuarbeiten.
2. In den letzten 20 Minuten der Einheit werden alle Schüler\_innen gebeten, ihr Scratch-Projekt der Klasse zu präsentieren. Jede Präsentation soll zwischen 1-2 Minuten dauern. Dabei ist nicht so wichtig, ob ihr Projekt nun tatsächlich fertig implementiert ist, sondern den momentanen Stand zu präsentieren. Sie sollen in diesem Zusammenhang kurz darauf eingehen, was ihr Projekt macht, wie man es bedient, was bereits implementiert ist und was eventuell noch gemacht gehört. Sie können auch erwähnen, auf welche Schwierigkeiten sie gestoßen sind und wie sie diese letztendlich gemeistert (oder nicht gemeistert) haben.

## Hausübung

3. Schüler\_innen stellen ihr Scratch-Projekt bis zu einer von der Lehrkraft in der Einheit angekündigten Frist fertig, sofern sie während der Unterrichtszeit damit nicht fertig geworden sind.

## Quellenangabe

[1] „What is Programming?“ [https://docs.google.com/presentation/d/1-xx4NnLVLJDVPrb2OSKQAPPMPoEmgOVXb\\_hhOJRVUaU/edit?pref=2&pli=1#slide=id.gf9aa6921\\_034](https://docs.google.com/presentation/d/1-xx4NnLVLJDVPrb2OSKQAPPMPoEmgOVXb_hhOJRVUaU/edit?pref=2&pli=1#slide=id.gf9aa6921_034) [29. Dezember 2016]