

Scratch als motivierender Zugang zur Programmierung

„Mehr Kreativität im Informatikunterricht“ [3]

Severin Küpfer
Pädagogische Hochschule Bern, 2008

Worum geht es?

Die vorgestellte Unterrichtssequenz ermöglicht einen einfachen Zugang zur Programmierung mit Hilfe der Programmierumgebung Scratch. Scratch kann als Lernumgebung die fundamentalen Kenntnisse der Programmierung an ein Publikum im Alter von 8-18 Jahren weitergeben.

Die Unterrichtssequenz ist als Lernparcour für ca. 6-10 Lektionen aufgebaut und unterstützt die Lehrperson mit Unterrichtsmaterialien und weiterführenden Links.

Was ist Scratch?

Kurze Einführung findet man unter [6]

WORUM GEHT ES?	1
Was ist Scratch?	1
DIDAKTISCHE ÜBERLEGUNGEN	2
Zielpublikum	2
Fundamentale Ideen	2
Lernziele	3
Operationalisierte Ziele	3
Weshalb ein Lernparcour?	4
Quellen	5
UNTERRICHTSSEQUENZ	6
Verlaufsplanung	6
1. Lektion	6
2. Lektion	6
3. und 4. Lektion	6
5. und 6. Lektion	7
7. , 8. und 9. Lektion	8
10. Lektion	8
Material	8

Didaktische Überlegungen

Zielpublikum

Die Unterrichtssequenz eignet sich in erster Linie für Lernende in der Sekundarstufe I, wobei sie im Fach Informatik aber auch interdisziplinär mit den Fächern Gestalten oder Deutsch eingesetzt werden kann.

Als Voraussetzung sollten die Lernenden den alltäglichen Umgang mit dem Computer beherrschen, damit das Starten des Programms und die Bedienung der Menus keine Schwierigkeiten bieten.

Für die Sekundarstufe II gibt es einzelne Anwendungsmöglichkeiten. So kann zum Beispiel anhand der einzelnen Scripts der Aufbau eines Algorithmus bildhaft gezeigt oder die Interaktion mit dem Benutzer auf einfache Art analysiert werden. Als weiterführender Einstieg in die Programmierung reicht die Programmierumgebung Scratch jedoch nicht. Es fehlen wichtige Paradigmen der Programmierung, welche für umfangreiche Applikationen nötig sind. Als Beispiel fehlender Grundkonzepte könnte man die Typisierung oder den Zugriff auf externe Daten nennen und auch komplexe Algorithmen werden nur schwerfällig oder gar nicht realisiert.

Fundamentale Ideen

Das logische und abstrakte Planen von Abläufen liegt dieser Unterrichtseinheit als fundamentale Idee zugrunde. Durch das Erstellen eigener Programme und Produkte wird zudem die Kreativität und das projektbezogene Arbeiten geschult und gefördert.

Sowohl einfache Abläufe wie auch umfassende Algorithmen lassen sich in der Unterrichtssequenz planen und umsetzen. Es soll in verschiedenen Übungsposten die Handhabung und Planung der Abläufe geübt werden, bevor die Lernenden eigenständige Projekte in Angriff nehmen. Von einer einfachen Animation mit linearem Verlauf bis zu interaktiven und trickreichen Produkten ist somit im letzten Übungsposten alles möglich.

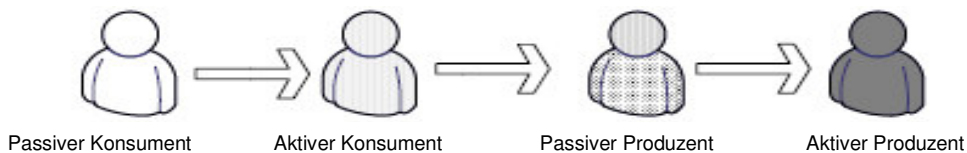
Es macht Sinn sich mit den Abläufen alltäglicher Arbeiten zu beschäftigen und sich neben der Analyse auch mit der Automatisierung der Vorgänge zu beschäftigen. Durch die Programmierung werden die nötigen Fähigkeiten zum besseren Verständnis automatischer Abläufe geschult. Die Kreativität bei der Erstellung eigener Projekte dient zusätzlich der Problemlöseorientierung der lernenden Person.

Mit Scratch hat man zudem einen repräsentativen Zugang zur Programmierung gefunden und Lernende sind motivierter an der Arbeit. In der kürzesten Zeit erstellen sie ein visuelles und interaktives Produkt, welches aus ihrer Hand erschaffen wurde.

Lernziele

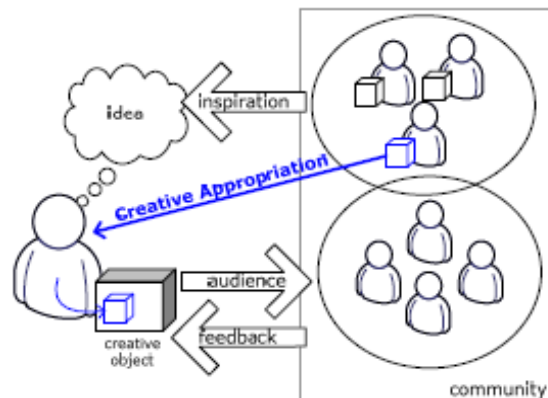
- Den Kontakt zu automatischen Abläufen ermöglichen.
- Die Interaktion zwischen Mensch und Maschine auf eine explorative Weise entdecken.
- Förderung des kreativen Potentials lernender Personen [3]
- Die Verarbeitung im Computer anhand von Sequenzen und einfachen Selektionen und Iterationen erkennen. [2]
- Vom passiven zum aktiven Lernenden (Konsument oder Produzent) wechseln. Erläuterung unter [4].

Der Lernende wechselt seine Rolle vom Konsumenten zum aktiven Produzenten. Aktiv bedeutet in diesem Zusammenhang keine Reproduktion sondern Eigenkreation eines Produktes.



- Gemeinsamer Austausch von Ideen und Wissen zum kreativen Anstoss für eigene Produkte. „Kreativer Wissenstank“. Unter [4] wird der Mehrwert der kooperativen und kreativen Unterstützung der Community dargestellt.

Die Community besteht selber aus aktiven Produzenten und kann durch ihre offen gelegten Produkte eigene Kreationen anregen.



Operationalisierte Ziele

- Die Lernenden verstehen Abläufe und können bestehende Programme analysieren und erkennen einfache implementierte Algorithmen.
- Die Lernenden verwenden die Grundbausteine von Algorithmen und kreieren eigene logische Abläufe.
- Die Lernenden bedienen sich einfacher Techniken, um die Interaktion zwischen Mensch und Computer umzusetzen.
- Die Lernenden planen ein eigenes Informatikprogramm, welches mit Hilfe von Bildern, Tönen und Texten animiert wird.
- Durch die Planung des eigenen Projektes wenden die Lernenden selber Prozesse zum Erstellen eines Produktes an. (Analyse, Entwurf, Implementierung, Test)
- Die Lernenden verschaffen sich eine Übersicht über den Umfang der Lernumgebung Scratch und setzen zum Beispiel die vorgegebenen Funktionen zur Steuerung des Programmablaufes gezielt ein.

Weshalb ein Lernparcour?

Die Informatik eignet sich optimal, um die Lernenden individuell zu fördern und zu unterstützen. Der Computer als Arbeitsgerät ermöglicht es in hohem Masse Aufgaben auf verschiedenen Niveaus zu erstellen, da jeder Computer bei Fehlern den Lernenden unbeirrt ein Feedback über Erfolg oder Misserfolg zurückgibt.

Mit einem Lernparcour soll diese Unterrichtssequenz sowohl versierten Anwendern wie auch Anfängern die Möglichkeit geben im eigenen Tempo genügend Kenntnisse zu erwerben, um zum Abschluss des Parcours eine angepasste und eigenständige Arbeit umzusetzen.

Die einzelnen Posten sind in obligatorische und freie Übungen aufgeteilt und auch in der Reihenfolge bleibt dem Lernenden eine genügend grosse Wahlfreiheit. Es macht jedoch Sinn den obligatorischen Posten 1 (Tutorial) als erstes zu lösen, damit die Basisfunktionen von Scratch bekannt sind. Von Lektion zu Lektion werden nun 3 oder mehr Posten zusätzlich zur Auswahl gegeben. Dies lässt den stärkeren Lernenden Freiraum offen und unterstützt schwächere bei der Auswahl des nächsten Postens.

Die Lernenden erhalten zu Beginn auch einen Lernpass. Dieser dient dem Lehrer als Arbeitskontrolle und dem Lernenden als Erfolgsausweis, da jeder erledigte Posten durch die Lehrperson bestätigt wird. [5]

Ein Lernparcour vereint somit wichtige didaktische Konzepte: [8]

- Es ist eine Sammlung abwechslungsreicher Unterrichts- oder Übungsformen
- Die Lernenden können ein individuelles Tempo anschlagen
- Der Lernparcour kann durch schrittweise Erweiterung gewisse methodische Unterstützung bieten
- Die Lernenden behalten eine Wahlfreiheit
- Durch einen Lernpass behält die Lehrperson wie auch die Lernenden die Übersicht
- Der Lernparcour kann als Übungsanlage durch theoretische Inputs ergänzt werden
- Grosse Wahlfreiheit bei den Lernenden. Sowohl Inhalte, Sozialform und die Lernzeit kann in einem gegebenen Rahmen selber bestimmt werden.
- Komplexe Lerneinheiten können auf mehrere Posten aufgeteilt werden.
- Durch eine gute Kategorisierung der Posten kann sich der Lernende besser orientieren und einschätzen.

Quellen

Die Scratch-Community wächst von Tag zu Tag und man findet immer häufiger auch deutsche Beiträge zur Umsetzung im Schul- oder Freifachunterricht:

- [1] <http://scratch.mit.edu> : Heimplattform der Scratch Community. Alle wichtigen Informationen und Downloads sind hier zu finden. MIT, Boston
- [2] <http://signalbscw.tcs.informatik.uni-muenchen.de/pub/bscw.cgi/d194654/ScratchInf7-Schlager.pdf> : Unterrichtsbeispiel für Mittelstufe als Folienpräsentation mit Einführung Algorithmus. Universität München
- [3] http://www.die.informatik.uni-siegen.de/infos2007/vortragsfolien/p3/romeike_infos2007.pdf : Kreativität im Informatikunterricht. Wissenschaftliche Erkenntnisse. Universität Siegen
- [4] <http://web.media.mit.edu/~andresmh/scratchr/idc-paper.pdf> : Scratch Community als kreativer Wissenstank. MIT, Boston
- [5] http://www.studienseminar-koblenz.de/medien/fachseminare/ER/03_Didaktik_und_Methodik/Stationenlernen-Skript.ppt Script: Möglichkeiten und Grenzen von Lernzirkeln. Studienseminar Koblenz
- [6] <http://www.kuepfis.ch/ScratchIfYouCan.doc> Kurze Vorstellung der Programmierumgebung Scratch. Pädagogische Hochschule, Bern
- [7] <http://www.funlearning.de/> Eine didaktisch aufgebaute Unterrichtseinheit mit weitergehenden Überlegungen zum Thema Scratch und Algorithmen.
- [8] **Neue Wege der Wissensvermittlung:** Umfassende Erläuterung der erkenntnisorientierten Lehrform. Norbert Landwehr. Sauerländer 2006

Unterrichtssequenz

Verlaufsplanung

1. Lektion

Einstieg in den Lernparcour

Diese Lektion soll neben den Erläuterungen zum Lernparcour die Gelegenheit geben, einzelne Scratch-Beispiele kennen zu lernen. Die Beobachtungen der Beispiele können von einer einfachen Erklärung bis hin zur Analyse einzelner Algorithmen gehen. (Eventuell passende Beispiele bereitstellen)

Tätigkeit	Beschreibung	Material
Einführung	Weshalb Programmieren	mdl. ev. Folien
Einführung Lernparcour	Vorstellen des Lernpasses und des Lernparours. Ablauf und Vorgehen anhand Posten eines zeigen.	Lernpass Ein Posten des Lernparcours
Scratch vorstellen	Scratch gemeinsam Installieren	Scratchinstaller
Scratchprodukte kennen lernen	Posten 0: Beispiele aus Programmierumgebung laden und individuell ausprobieren lassen	Beispiele aus Scratchumgebung, ev. eigene Beispiele
Gemeinsamer Austausch	Welche Beispiele wurden betrachtet. Details hervorheben. Gewisse Algorithmen gemeinsam erkennen.	

2. Lektion

Kennen lernen der Programmierumgebung Scratch

Die Programmierumgebung Scratch kann anhand eines Tutorials am einfachsten kennen gelernt werden. Es werden alle wichtigen Funktionen und Vorgehen mit Hilfe eines erweiterbaren Beispiels erläutert.

Tätigkeit	Beschreibung	Material
Auftragserteilung	Posten 1: Durcharbeiten des Tutorials	Posten 1, Tutorial [1]
Arbeiten an Posten 1	Erste Schritte in der Programmierumgebung machen lassen. Unterstützen bei Verständnisproblemen	Eventuell Kopfhörer für Tonaufgabe des Tutorials
Feedback: Blitzlicht	Top oder Flop durch Handzeichen bei Lernenden abfragen.	

3. und 4. Lektion

Algorithmen analysieren

Zu den Grundlagen der Programmierung gehört die Analyse des Aufbaus einfacher Algorithmen. Die folgenden zwei Lektionen dienen der Betrachtung von Sequenz, Selektion und Iteration. Der Lernparcour wird mit weiteren ausgewählten Posten ergänzt und die Lernenden zur Wahl der für sie geeigneten Posten motiviert.

Tätigkeit	Beschreibung	Material
Theorieinput	Woraus besteht ein Algorithmus	Lektionenpräp.doc Faltvorlagen Faltpapier
Vorstellen der Posten 3 - 5	Abschliessen des Posten 1. (Auf Lernpass bestätigen) Auswahl des Folgepostens anhand eigener Präferenz.	Posten 1-5, Tutorial
Erste Resultate veröffentlichen	Zum Ende der Lektionen erste Resultate an Mitschüler veröffentlichen. (Rundgang, Vorstellen vor Klasse oder Auswahl durch Lehrperson)	

5. und 6. Lektion

Arbeiten an Lernparcour

Vertiefung in den Lernparcour. Weitere Posten werden hinzugefügt und vorgestellt. Hierbei hat die Lehrperson die Gelegenheit Schwergewichte auf einzelne Posten zu setzen, damit die Lernenden die geeignete Wahl treffen.
Nach Ende dieser Lektionen sollten die Lernenden den Umfang und Möglichkeiten der Programmierumgebung kennen. Regelmässig sollen die Lernenden auf ihr Abschlussprojekt hingewiesen und Ideen gesammelt werden.

Tätigkeit	Beschreibung	Material
Ev. Theorieinput	Planen eines Algorithmus	[7]
Vorstellen der Posten 4-9	Motivierender Einstieg zu den neuen Posten. Möglichkeiten, Tipps und Tricks vorzeigen. Gegenseitige Unterstützung der Lernenden.	Posten 1-9 Tipps und Tricks anhand des Lernfortschritts der Lernenden vorbereiten.
Weiterarbeit an Lernparcour	Individuelles und unterstütztes Arbeiten am Lernparcour	
Auftragserteilung	Grundidee für Posten 10 erarbeiten. Beispiele für eigenständiges Projekt je nach Wissensstand. Auftrag: Zur nächsten Informatikpräsenz Idee für eigenes Projekt mitbringen. Bsp.: <ul style="list-style-type: none"> • Lineare Animation (Geschichte) • Interaktives Spiel • Ausbau eines bestehenden Posten • Weiterentwicklung eines Projektes der ScratchCommunity 	

7. , 8. und 9. Lektion

Eigenes Projekt verwirklichen

Erst bei der Entwicklung eines eigenen Produktes, kann die gesamte Kreativität der Lernenden genutzt werden. Oftmals nutzt ein kleiner Anstoss und es entstehen ungeahnte Lösungen.

Unbedingt die Lernenden bei ihrer Idee abholen und durch Steuerung des Umfangs auf die vorgegebene Zeit anpassen.

Tätigkeit	Beschreibung	Material
Einführung / Planung	Präsentieren der Ideen	mdl.
Design	Sammeln der Materialien (Bilder, Töne ...) Design des Programms (Ablauf, Interaktion)	
Erstellen des Programms /Entwicklung	Selbständiges Erstellen des Programms	
Testen und Verbessern	Gegenseitige Unterstützung bei der Korrektur von Fehlern.	

10. Lektion

Präsentation der Resultate

Die Wertschätzung der erbrachten Leistung der Lernenden ist von grosser Bedeutung. Dies fördert die Selbstkompetenz und unterstützt die Kreativität. Spezielle Algorithmen können dabei hervorgehoben werden. Eventuell für eine nächste Durchführung speichern.

Tätigkeit	Beschreibung	Material
Vorbereitung	Veröffentlichen der Resultate (on- oder offline)	Ev. Scratch Account
Gegenseitiges Testen und Bewerten	Ausprobieren der Programme. Analysieren und Bewerten der einzelnen Programme	Punktesystem (Vor Weihnachten mit Schokoladetalern ;-)
Gratulation	Wertschätzung der Resultate	

Material

Inhalte	Datei
Lernpass	ScratchPass 6x.doc
Posten 0 – 10	Posten zu Lernparcour.zip
Tutorial	Tutorial_dt.pdf
Lektionenpräp Algorithmus	Lektionenpräp_Algorithmen.doc
Faltanweisungen	Faltanweisungen.zip
EVA Vorlage	EVA.doc