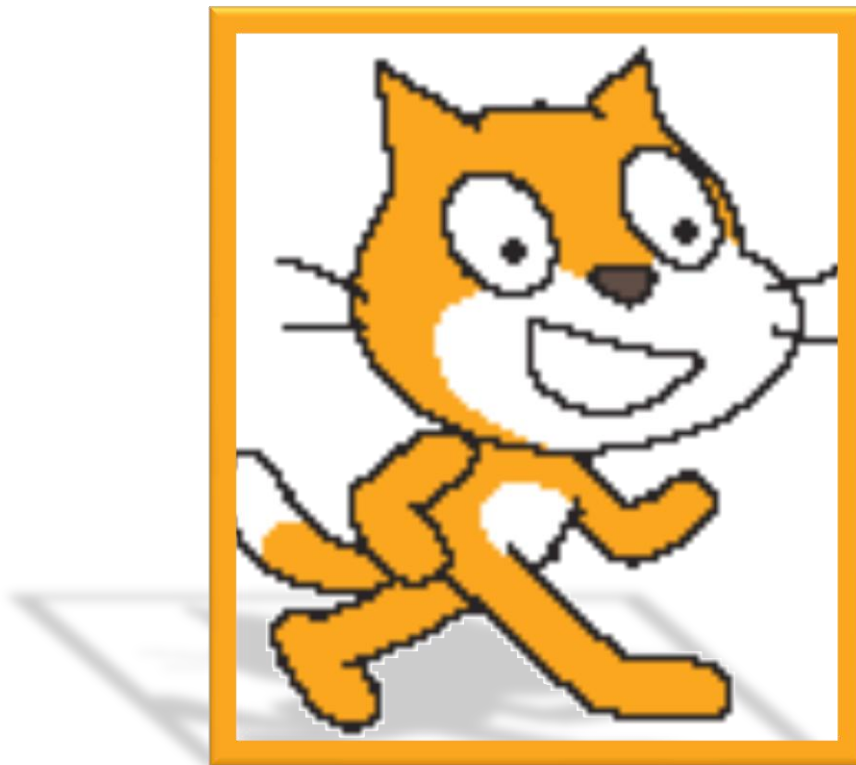


**EINFÜHRUNG IN DIE PROGRAMMIERUNG MIT SCRATCH –  
EINE UNTERRICHTSEINHEIT IM WAHLPFLICHTFACH  
INFORMATIK EINES ZEHNTEN JAHRGANGS AN DER  
WALTHER-RATHENAU-OBERSCHULE (GYMNASIUM)**

Darstellungsschwerpunkt:  
Förderung der Fähigkeit zum selbstständigen Arbeiten

Schriftliche Prüfungsarbeit zur zweiten Staatsprüfung  
für das Amt des Studienrats



vorgelegt von

**Armin Schaeper**  
**Studienreferendar**  
**2. Schulpraktisches Seminar Charlottenburg-Wilmersdorf (S)**

Berlin, 7. April 2008

## INHALT

1	Einleitung.....	4
2	Scratch und Unterricht nach kreativen Maßstäben.....	7
2.1	Die Programmiersprache Scratch .....	7
2.1.1	Hintergrund.....	7
2.1.2	Die Programmieroberfläche und Programmierung mit Scratch.....	8
2.2	Förderung der Fähigkeit zum selbstständigen Arbeiten durch Unterricht nach kreativen Maßstäben .....	10
2.3	Scratch im Unterricht.....	13
3	Unterrichtsvoraussetzungen .....	14
3.1	Allgemeine Unterrichtsvoraussetzungen.....	14
3.2	Spezielle Unterrichtsvoraussetzungen.....	15
3.2.1	Lerngruppe .....	15
3.2.2	Erfahrungen mit Partnerarbeit und mit Präsentationen .....	16
3.2.3	Bisherige Programmiererfahrungen im Wahlpflichtunterricht .....	16
3.2.4	Weitere Themen im bisherigen Unterricht.....	17
4	Sachanalyse .....	17
4.1	Auswahl des Produkts .....	17
4.2	Informatische Konzepte .....	18
4.2.1	Botschaften .....	18
4.2.2	Boolesche Variablen.....	19
4.2.3	Ganzzahlvariablen .....	19
5	Didaktische Analyse.....	20
5.1	Bezug zum Rahmenlehrplan .....	20
5.2	Didaktische Reduktion .....	22
6	Methodische Überlegungen.....	22
6.1	Planung der Unterrichtsreihe nach kreativen Maßstäben .....	22
6.2	Online-Medieneinsatz.....	26
6.2.1	Einsatz der Online-Community Scratch-Online.....	26
6.2.2	Einsatz des Online-Arbeitsbereiches BSCW .....	26
7	Verlaufskonzeption der Unterrichtsreihe .....	27
7.1	Grobverlauf .....	27

7.2	Synopse .....	29
8	Darstellung und Analyse einzelner Stunden .....	31
8.1	Stunde 1: „Einführung in Scratch“ .....	31
8.1.1	Darstellung und Begründung der Phasen .....	31
8.1.2	Analyse .....	33
8.2	Stunde 5/6: „Eine mehrdimensionale Kurzgeschichte“ .....	35
8.2.1	Darstellung und Begründung der Phasen .....	35
8.2.2	Analyse .....	36
8.3	Stunde 13: „Projektdokumentation für den Schaukasten“ .....	39
8.3.1	Darstellung und Begründung der Phasen .....	39
8.3.2	Analyse .....	40
9	Gesamtreflexion .....	42
9.1	Projekte und Dokumentationen .....	42
9.2	Abschließende Analyse .....	43
10	Quellen- und Literaturverzeichnis .....	47
11	Anhang .....	49

Anonymisierte Fassung

# 1 EINLEITUNG

„Profi-Programmierer schnallt Euch fest an, die Turbo-Wissenschaftler vom MIT könnten mit ihrem Projekt Scratch eine ganze Generation in eine Zombie-Armee williger Kinderprogrammier[er] verwandeln. Objektorientiertes Programmieren ist damit nämlich angeblich so einfach, wie mit Lego-Bausteinen zu spielen. Und sieht auch fast so aus. [...]“<sup>1</sup>

Bereits zu einem früheren Zeitpunkt im Verlauf meines Referendariats hatte ich mich schon einmal mit Scratch auseinandergesetzt: Ende Mai 2007 durfte ich die damals brandneue Programmiersprache im Fachseminar Informatik kurz im Rahmen eines Referats zu objektorientierten „Mini-Sprachen“ für den Informatikunterricht vorstellen. Mein Eindruck von Scratch entsprach dabei annähernd dem des voranstehenden Zitats von SPIEGEL ONLINE. Auch wenn ich zu diesem Zeitpunkt noch nicht das volle Potenzial dieser Programmiersprache absehen konnte, stand für mich fest, dass ich sie gern in meinem Informatikunterricht einsetzen würde. Im darauf folgenden Schuljahr bot sich nun die Gelegenheit dafür, als ich einen Wahlpflichtkurs Informatik der zehnten Jahrgangsstufe übernahm.

Bei der Erwähnung von Scratch wird häufig auf die motivierende Komponente verwiesen, die offenbar von dieser Programmiersprache ausgeht.<sup>2</sup> Die Annahme, dass Lernende durch stärkere Eigenmotivation besser zu selbstständigem Arbeiten animiert werden können, legt eine besondere Betrachtung der Entwicklung der Fähigkeit zum selbstständigen Arbeiten bei der Einführung von Scratch im Unterricht nahe. Sollte Scratch wirklich ein geeignetes Mittel zur Förderung dieser Fähigkeit sein, so müsste sich schrittweise vermehrt selbstständiges Arbeiten feststellen lassen.

Ich gehe davon aus, dass der Lernerfolg nicht nur von der gewählten Programmiersprache abhängt, sondern es muss auch ein geeignetes Umfeld geschaffen werden, in dem die Wirkung des Mittels sich möglichst gut entfalten kann. Hier erschien mir eine Orientierung an ROMEIKES „Kriterien für einen kreativen Informatikunterricht“<sup>3</sup> als sinnvoll.

In der vorliegenden Arbeit soll betrachtet werden, ob und wie eine Einführung in die Programmierung mit Scratch in einem Unterricht nach kreativen Maßstäben die Fähigkeit der Schülerinnen und Schüler zum selbstständigen Arbeiten fördert. Diese Förderung ist dabei als Prozess zu sehen, der sich schrittweise entfaltet und anfangs eher in Ansätzen zu beobachten sein wird. Gelingt das Vorhaben, so sollten gegen Ende

---

<sup>1</sup> (Knoke, 2007)

<sup>2</sup> Vgl. beispielsweise (Romeike, [www.funlearning.de](http://www.funlearning.de) - Informatik macht Spass, 2007)

<sup>3</sup> Vgl. (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007) und Kapitel 2.2 dieser Arbeit

der Unterrichtsreihe die von ROMEIKE beschriebenen Kriterien wesentlich stärker erfüllt werden und es sollten Ergebnisse vorweisbar sein, die einen hohen Anteil selbstständigen Arbeitens beinhalten.

Wie bereits beschrieben wurde, sehe ich die Begriffe „Selbstständigkeit“, „Motivation“ und „Kreativität“ für dieses Unterrichtsvorhaben in einem engen Zusammenhang. Daraus lassen sich die folgenden Arbeitshypothesen ableiten.

**Arbeitshypothese 1 - Motivation:** Selbstständiges Arbeiten wird durch freiwilliges Auseinandersetzen mit dem Unterrichtsgegenstand und seinem Umfeld<sup>4</sup> gefördert.

**Arbeitshypothese 2 - Kreativität:** Selbstständiges Arbeiten wird durch die Möglichkeit zur kreativen Aneignung des Unterrichtsgegenstandes gefördert.

**Arbeitshypothese 3 - Progression:** Die Fähigkeit zum selbstständigen Arbeiten ist eine Größe, die im Unterricht prozessartig entwickelt werden kann.

Insbesondere der Aspekt der Progression deutet auf einen Sachverhalt hin, der nicht vernachlässigt werden darf: Zur progressiven Entwicklung von selbstständigem Arbeiten muss der zeitlich zur Verfügung stehende Rahmen angemessen gewählt werden. Steht zu wenig davon für das Unterrichtsgeschehen zur Verfügung, verhindert der entstehende Zeitdruck die angestrebten Fortschritte und hemmt die Entfaltung von Motivation und Kreativität.

**Arbeitshypothese 4 - Zeit:** Die Fähigkeit zum selbstständigen Arbeiten kann nur dann gefördert werden, wenn die Schülerinnen und Schüler ausreichend Zeit zur Bewältigung der gestellten Aufgaben zur Verfügung haben.

Die Punkte Motivation und Kreativität erfordern außerdem eine weniger starke Dominanz des Lehrers im Unterrichtsgeschehen. Intensives Einmischen in die Bearbeitungs- und Erkenntnisgewinnungsprozesse führt zu Blockadehaltungen in beiden Bereichen und verhindert damit den angestrebten Fortschritt. Umgekehrt muss der Lehrer für Beratungsgespräche und kreative Impulse verfügbar sein, um in Situationen, in denen das Vorankommen gefährdet ist, helfend zur Seite stehen zu können.

**Arbeitshypothese 5 - Lehrerrolle:** Für die Förderung der Fähigkeit zum selbstständigen Arbeiten ist eine Lehrerrolle erforderlich, bei der die Beratungsfunktion im Vordergrund steht.

Bei selbstständigem Arbeiten besteht die Gefahr der Vereinzelung. Um dieser entgegenzuwirken, können Phasen des Vergleichs und der Präsentation von Ergebnissen dazu beisteuern, dass Ideen nachvollzogen, Alternativen bekannt, Irrwege erkannt und Er-

---

<sup>4</sup> Mit dem Begriff „Umfeld“ ist in diesem Zusammenhang die Einbettung des Unterrichtsgegenstandes gemeint. Dazu gehören beispielsweise die verwendete Programmiersprache, die zur Verfügung stehenden oder zu erstellenden Medien usw.

gebnisse angemessen gewürdigt werden. Dies ermöglicht mit Blick auf Motivation und Verständnis ein effektives selbstständiges (Weiter-) Arbeiten.

Das selbstständige Arbeiten darf nicht Selbstzweck sein. Während des Arbeitsprozesses und danach ist dafür Sorge zu tragen, dass die gewonnenen Erkenntnisse festgehalten werden. Dies ermöglicht rückblickend eine Rekonstruktion der Arbeit und des hinzugewonnenen Wissens und vorwärtsblickend Anknüpfungspunkte für die weitere Arbeit.

Damit finden sich abschließend zwei weitere Arbeitshypothesen.

**Arbeitshypothese 6 - Präsentation:** Selbstständiges Arbeiten wird durch die Möglichkeit zum Vergleich der Ergebnisse gefördert. Die Möglichkeit zur Vorstellung eigener Produkte fördert die Motivation zur eigenständigen Auseinandersetzung mit dem Unterrichtsgegenstand.

**Arbeitshypothese 7 - Dokumentation:** Eine aussagekräftige Dokumentation der erzielten Resultate ist notwendig, um die Ergebnisse selbstständigen Arbeitens einschätzen und zukünftig darauf aufbauen zu können.

In dieser Arbeit soll zunächst die Programmiersprache Scratch vorgestellt werden. Dem schließt sich eine Betrachtung an, welchen Kriterien Unterricht nach kreativen Maßstäben genügen muss, und es wird aufgezeigt, weshalb sich Scratch und diese Unterrichtsform sinnvoll verbinden lassen.

Danach werden für die darzustellende Unterrichtsreihe die Unterrichtsvoraussetzungen analysiert und es wird in einer Sachanalyse dargestellt, wie den Schülerinnen und Schülern die informatischen Konzepte „Botschaften“, „boolesche Variablen“ und „Ganzzahlvariablen“ in dieser Unterrichtsreihe vermittelt werden sollen. Bei der folgenden didaktischen Analyse wird der Bezug zum Rahmenlehrplan hergestellt, wobei insbesondere auf die Kompetenzbereiche eingegangen wird. Es wird dann dargelegt, weshalb in der Unterrichtsreihe ein Schwerpunkt auf der Erstellung eines Adventure-Spiels<sup>5</sup> mit Scratch liegt und anschließend werden methodische Überlegungen unter Berücksichtigung eines Unterrichts nach kreativen Maßstäben vorgenommen.

Aus diesen Überlegungen heraus wird eine Verlaufskonzeption der Unterrichtsreihe entwickelt und es werden ausgewählte Stunden dieser Unterrichtsreihe dargestellt und entsprechend den Arbeitshypothesen analysiert. Abschließend erfolgt eine Gesamtreflexion, in der das Unterrichtsvorhaben und die entwickelten Arbeitshypothesen kritisch überprüft werden.

---

<sup>5</sup> Adventure-Spiele sind Computerspiele, in denen die Spielfigur üblicherweise Rätsel lösen muss, um das Ziel des Spiels zu erreichen. Dabei entwickelt sich der Spielverlauf meistens entlang einer Geschichte.

## 2 SCRATCH UND UNTERRICHT NACH KREATIVEN MAßSTÄBEN

### 2.1 DIE PROGRAMMIERSPRACHE SCRATCH

#### 2.1.1 HINTERGRUND

Die Programmiersprache Scratch wurde von der „Lifelong Kindergarten group“<sup>6</sup> am „MIT Media Lab“<sup>7</sup> in den USA entwickelt und im Frühjahr 2007 veröffentlicht.<sup>8</sup>

Die Entwickler beschreiben Scratch folgendermaßen:

„With Scratch, kids can create their own interactive stories, games, music, and animation - and share their creations with one another on the Web.

Scratch is designed to help young people (ages 8 and up) develop 21st century learning skills. As they create Scratch projects, young people learn important mathematical and computational ideas, while also gaining a deeper understanding of the process of design.“<sup>9</sup>

Diese Programmiersprache zielt nicht darauf ab, Standardanwendungen zu programmieren, sondern hat ihren Schwerpunkt in grafisch orientierter Programmierung. Sie wurde ursprünglich für den Einsatz in sogenannten „Computer Clubhouses“<sup>10</sup> und ähnlichen Einrichtungen der Nachmittagsbetreuung von acht- bis achtzehnjährigen Jugendlichen entwickelt, verbreitete sich jedoch rasch und wird inzwischen auch im schulischen Bereich und darüber hinaus zur Einführung in die Programmierung benutzt.<sup>11</sup>

Scratch wurde mit der Programmiersprache „Squeak“<sup>12</sup> geschrieben, die ihrerseits auf dem inzwischen nur noch selten gebrauchten „Smalltalk-80“ basiert. Bereits Squeak ist grafisch orientiert. Es verwirrt jedoch häufig wegen des durchgängig objektorientierten Aufbaus, da die deutliche Unterscheidung von Entwicklungsumgebung und Quelltext („Programm“) erschwert wird. Die Bedeutung von Squeak lässt sich gut erahnen, wenn

---

<sup>6</sup> Mehr Informationen unter <http://llk.media.mit.edu/>, Zugriff am 21. März 2008

<sup>7</sup> Mehr Informationen unter <http://www.media.mit.edu/>, Zugriff am 21. März 2008. „MIT“ ist die Abkürzung für „Massachusetts Institute of Technology“.

<sup>8</sup> Vgl. (Spotlight - Creating from Scratch, 2008)

<sup>9</sup> (Spotlight - Creating from Scratch, 2008), Übersetzung A. S.: „Mit Scratch können Kinder ihre eigenen interaktiven Geschichten, Spiele, Musik und Animationen erschaffen und diese miteinander im Internet teilen. Scratch hat den Zweck, jungen Leuten (ab 8 Jahren) zu helfen, die Lernfähigkeiten für das 21. Jahrhundert zu entwickeln. Während sie Scratch-Projekte erstellen, erlernen die Jugendlichen wichtige mathematische und informatische Konzepte und Ideen, erlangen aber auch ein tieferes Verständnis vom Entwicklungsprozess.“

<sup>10</sup> Mehr Informationen unter <http://www.computerclubhouse.org/>, Zugriff am 23. März 2008. Bei den „Computer Clubhouses“ handelt es sich um Freizeitangebote für Jugendliche, hauptsächlich in den USA.

<sup>11</sup> Vgl. (Romeike, www.funlearning.de - Informatik macht Spass, 2007)

<sup>12</sup> Mehr Informationen unter <http://www.squeak.org/>, Zugriff am 23. März 2008

man bedenkt, dass Squeak standardmäßig auf jedem Computer installiert sein soll, der mittels der „One-Laptop-Per-Child“-Initiative<sup>13</sup> verteilt wird.<sup>14</sup>

Die Programmierer von Scratch verfolgen im Wesentlichen drei Ansätze:

„Scratch offers a **low floor** (easy to get started), **high ceiling** (ability to create complex projects), and **wide walls** (support for a wide diversity of projects).“<sup>15</sup> – Besonders Programmieranfänger werden angesprochen, gleichzeitig soll aber sichergestellt sein, dass die Programmiersprache auch noch nach ersten Programmiererfolgen attraktiv bleibt.

Wegen des großen Erfolgs und der starken Beteiligung von Scratch-Nutzern an der Weiterentwicklung des Projekts ist Scratch inzwischen in einer deutschen Version erhältlich.

### 2.1.2 DIE PROGRAMMIEROBERFLÄCHE UND PROGRAMMIERUNG MIT SCRATCH

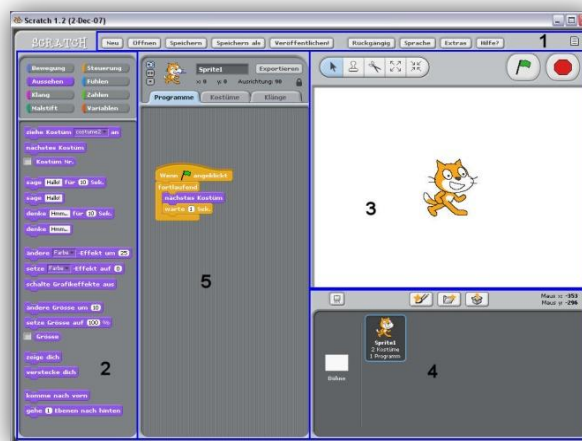


ABBILDUNG 1: PROGRAMMIEROBERFLÄCHE VON SCRATCH

Der Editor zum Erstellen von Scratch-Programmen ist in fünf Bereiche unterteilt:<sup>16</sup>

<sup>13</sup> Mehr Informationen unter <http://www.laptop.org/en/vision/mission/index.shtml>, Zugriff am 23. März 2008. Diese Initiative verfolgt die Absicht, jedem Kind in Entwicklungsländern einen Laptop bereitzustellen.

<sup>14</sup> (Squeak, 2008)

<sup>15</sup> (Rusk, Programming with Scratch, 2007), Übersetzung A. S.: „Scratch bietet einen niedrigen Boden (leichter Anfang), eine hohe Decke (Möglichkeit zur Erstellung von komplexen Projekten) und weit entfernte Wände (Unterstützung für eine Bandbreite gänzlich verschiedener Projekte).“

<sup>16</sup> Ich beziehe mich auf die aktuelle Version 1.2, Download unter (Scratch-Download, 2007). Alternative Bezeichnungen für die Bereiche der Programmieroberfläche, die aber meiner Meinung nach nicht strukturiert genug sind, finden sich unter <http://ilk.media.mit.edu/projects/scratch/materials/ScratchDiagram.pdf>, Zugriff am 24. März 2008

- (1) Über die **Menüleiste** werden die Basisaktionen ermöglicht (Erstellen eines neuen Projekts, Öffnen eines vorhandenen Projekts, Speichern eines Projekts, Veröffentlichen eines Projekts auf Scratch-Online<sup>17</sup>, Zurücknehmen einer Aktion, Sprach- und sonstige Einstellungen verändern, Hilfe aufrufen).
- (2) Im **Befehlsblöcke-Bereich** sind - in die acht Kategorien „Bewegung“, „Aussehen“, „Klang“, „Malstift“, „Steuerung“, „Fühlen“, „Zahlen“ und „Variablen“ unterteilt - die verfügbaren Befehle und Steuerungsstrukturen in Form von puzzleteilartigen Blöcken aufgelistet.
- (3) In der **Szenenbild-Anzeige** werden die im Projekt vorhandenen (und sichtbaren) Objekte<sup>18</sup> mit einigen ihrer Grundeigenschaften (beispielsweise Position, Orientierung, Größe, Aussehen) angezeigt. Ferner können hier Objekte dupliziert, gelöscht, verkleinert und vergrößert werden. Über eine stilisierte grüne Flagge und ein rotes Achteck („Stopp-Schild“) können Programme gestartet und angehalten werden.
- (4) Der **Objektauswahl-Bereich** gibt ebenfalls eine Übersicht, welche Objekte (sichtbare und nicht sichtbare) im Projekt vorhanden sind, und ermöglicht über die Auswahl der Objekte den Zugriff auf ihren Quelltext. Außerdem kann hiermit auf den Quelltext für den Hintergrund („Stage“ beziehungsweise „Bühne“) zugegriffen werden. Zusätzlich können hier fertige Objekte importiert und (unter Zuhilfenahme eines relativ primitiven Bildeditors) neue Objekte erstellt werden.
- (5) Der **Objektcode-Bereich** ist zweigeteilt: Im oberen Fünftel des Bereichs lassen sich Grundeigenschaften (wie der Name, die Position und das Orientierungsverhalten eines ausgewählten Objekts oder des Hintergrunds) festlegen. Außerdem können Objekte hier auch exportiert – also anderen Programmen zugänglich gemacht – werden. Die unteren vier Fünftel werden dafür genutzt, wahlweise den Quellcode des Objekts, sein Erscheinungsbild („Kostüme“) oder die zum Objekt gehörenden Klänge zu manipulieren.

Zur Erstellung eines neuen Scratch-Projekts wird zunächst im Objektauswahl-Bereich ein Standardobjekt, die „Scratch-Katze“, vorgegeben und angezeigt.<sup>19</sup> Dieses Objekt und gegebenenfalls jedes weitere hinzugefügte oder neu erstellte Objekt lässt sich in Aussehen und Verhalten manipulieren, indem sein Quellcode (per Doppelklick auf das

---

<sup>17</sup> Mit „Scratch-Online“ sind die verschiedenen Angebote auf der Scratch-Homepage (<http://scratch.mit.edu>) gemeint. Dort kann beispielsweise die aktuelle Version von Scratch heruntergeladen werden, es gibt Hilfe-Foren zur Programmier-Unterstützung, es können die eigenen Programme den anderen Scratch-Programmierern vorgestellt bzw. zur Verfügung gestellt werden usw.

<sup>18</sup> In früheren Versionen von Scratch wurde in der Regel der Begriff „Sprite“ (Übersetzung etwa: „Kobold“, „Elfe“, „Wicht“) für ein Objekt benutzt, wahrscheinlich, weil die Objekte häufig Wesen darstellten, die es zu steuern galt. Inzwischen ist man in der deutschen Version dazu übergegangen, meistens von „Objekten“ zu sprechen, auch um die Objektorientierung in Scratch zu unterstreichen.

<sup>19</sup> Dieses Objekt lässt sich löschen und verändern, ist aber bei jedem neuen Projekt erst einmal dabei.

Objekt) im Objektcode-Bereich angezeigt wird. Hier kann man nun, indem Bausteine aus dem Befehlsblöcke-Bereich in den Objektcode-Bereich gezogen und dort puzzleartig zusammengesetzt werden, das Verhalten des Objekts festlegen und auch sofort ausprobieren. Durch diese Grundstruktur ist das Auftreten von Syntaxfehlern<sup>20</sup> nahezu ausgeschlossen, was den „Low-Floor“-Ansatz unterstützt. Eine Interaktion zwischen Objekten ist durch Variablen, Botschaften zwischen den Objekten oder Sensoren<sup>21</sup> gewährleistet.

Fertige Projekte können lokal gespeichert und außerdem auf der Scratch-Homepage veröffentlicht werden. Hierzu ist eine kurze Registrierung auf der Homepage nötig, anschließend werden die vom gleichen Nutzer erstellten Projekte auf einer Seite zusammengefasst und sind für die restlichen Scratch-Nutzer sichtbar.

Einer der Grundgedanken der Programmierer von Scratch ist es, die Nutzer nicht nur eigene Projekte erstellen zu lassen. Sie sollen auch die Möglichkeit haben, Projekte anderer Nutzer nachzuvollziehen und zu ändern (und dann neu zu veröffentlichen). Diese als „Remix“ bezeichneten Projekte werden mit dem ursprünglichen Projekt verknüpft, sodass Eigenanteile an der Programmierung trotzdem deutlich sind.

Eine Vielzahl an mitgelieferten Beispielgrafiken und -projekten erleichtert, zusammen mit den unzähligen Beispielen auf Scratch-Online, den Einstieg und erhält auch nach längerer Arbeit mit Scratch die Motivation an der Verbesserung oder Veränderung bereits bestehender Projekte beziehungsweise der Erstellung neuer Produkte.

Nicht zuletzt gibt es seit einiger Zeit auch eine Hardware-Komponente, das sogenannte „Scratch-Board“<sup>22</sup>, welches über Sensoren und Steuerungselemente verfügt, die ein Scratch-Programm steuern können. Dadurch werden - ähnlich wie bei dem LEGO-Mindstorms-Projekt<sup>23</sup> - sensorgesteuerte Interaktionen mit der Umwelt ermöglicht.

## 2.2 FÖRDERUNG DER FÄHIGKEIT ZUM SELBSTSTÄNDIGEN ARBEITEN DURCH UNTERRICHT NACH KREATIVEN MAßSTÄBEN

Programmierender Informatikunterricht geht über die reine Vorstellung einer Programmiersprache hinaus. Es sind Grundsätze nötig, nach denen der Unterricht ausgerichtet wird und die zum jeweiligen Thema passen müssen.

„Erfolgreicher Informatikunterricht beachtet bekannte Rahmenbedingungen aus dem Lernprozess und aus dem Prozess des fachlichen Arbeitens. Hier wird ein Lernprozess angestrebt, der vom aktiven Schüler ausgeht, der sich Informatik handlungsorientiert,

---

<sup>20</sup> Syntaxfehler sind Fehler, die aus einer Verletzung des formalen Regelgerüsts zur Programmierung in der betreffenden Programmiersprache resultieren.

<sup>21</sup> Sensoren sind hier alle Schnittstellen zur Wahrnehmung der Umwelt.

<sup>22</sup> Mehr Informationen unter <http://scratch.mit.edu/pages/scratchboard>, Zugriff am 24. März 2008

<sup>23</sup> Mehr Informationen unter <http://mindstorms.lego.com/>, Zugriff am 24. März 2008

problemorientiert, anwendungsorientiert und ganzheitlich aneignet und dabei vom Lehrer begleitet wird. [...] Mit dieser Kombination von Kriterien für das Lernen gelingt nicht nur die Betonung der Schüleraktivität, sondern diese aktive Rolle wird für den Schüler auch sichtbar.“<sup>24</sup>

Um diese Form der Schüleraktivität auch zu erreichen, muss gewährleistet sein, dass die Schülerinnen und Schüler auch bereit sind, die ihnen zugedachte Rolle einzunehmen. Natürlich kommt es hierfür nicht in Frage, Zwang und Druck als Motivationsgrundlage zu wählen. Vielmehr sollten sie von sich aus bereit sein, sich aktiv mit dem Lerngegenstand auseinanderzusetzen.

Wie im Zitat schon unterstrichen wurde, wäre es erstrebenswert, wenn der Lerngegenstand den Schülerinnen und Schülern lediglich angeboten wird, die sich daraufhin aktiv und weitestgehend selbstgesteuert mit ihm auseinandersetzen. Dies wird jedoch nur dann gelingen, wenn die Rahmenbedingungen so gestaltet sind, dass die Motivation zur Auseinandersetzung mit dem Lerngegenstand vorhanden ist und bleibt.

Dieses hohe Motivationspotential kann erreicht werden, wenn die damit verbundene Leistung eine hohe Bedeutsamkeit für die Lernenden hat. Diese Verbundenheit spiegelt sich darin wider, dass die Lernenden die Verantwortung für den Weg zum Ziel übernehmen dürfen, noch viel mehr: übernehmen **sollen**, und dass jedes Resultat, welches den Anforderungen entspricht - gerade auch in seiner individuellen Ausgestaltung - anerkannt wird.

Diese Etappe, der Weg zum Ziel in Eigenverantwortung, welcher von der Schülerin beziehungsweise dem Schüler beschritten wird, ist zwangsläufig schöpferisch, also kreativ.

„Kreativ ist eine Leistung im Allgemeinen dann, wenn sie neu ist, Seltenheitswert besitzt und nützlich ist. [...] Die Informatik wird von praktizierenden Informatikern als kreatives Gebiet angesehen[. ...] Auch für den Informatikunterricht sehen Fachdidaktiker kreatives Potential[.]“<sup>25</sup>

Dieses Potential gilt es auszuschöpfen, da es, wie bereits aufgeführt, für eine hinreichende Motivationslage notwendig ist, um eine hohe Schüleraktivität zu erreichen. Auf das in der Informatik bereits vorhandene kreative Potential kann im Informatikunterricht effektiv zugegriffen werden, wenn es von den Schülerinnen und Schülern direkt erfahren wird. Hierzu sind laut ROMEIKE die Relevanz des Unterrichtsgegenstandes für die Schülerinnen und Schüler sowie Problemlösungs- oder Produkterschaffungsprozesse notwendig, um überhaupt kreativ im Informatikunterricht tätig werden zu können.<sup>26</sup>

---

<sup>24</sup> (Schubert & Schwill, 2004), S. 34

<sup>25</sup> (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 2

<sup>26</sup> Vgl. (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 3f.

Die Aufgaben, die den Schülerinnen und Schülern gestellt werden, müssen ebenfalls bestimmten Kriterien genügen:<sup>27</sup>

- Die Aufgabe sollte für die Schülerin beziehungsweise den Schüler **subjektiv neu** sein, es darf noch kein konkretes Muster zur Lösung der Aufgabe vorliegen.
- Die Aufgabe sollte **offen** gestellt sein, darf also lediglich Rahmenbedingungen und eine ungefähre Zielerwartung zur Lösung der Aufgabe enthalten. Alternativ, aber nicht vorzuziehen, wären auch Abstufungen in der Formulierung der Zielerwartung möglich. Die Offenheit der Aufgabenstellung kann sich in einer offenen Erwartung gegenüber dem Ergebnis oder dem Bearbeitungsweg ausdrücken.
- Die Bearbeitungstiefe sollte **flexibel** sein, es sollten Erweiterungen, Veränderungen und Optimierungen möglich sein.
- Bei den Schülerinnen und Schülern sollte bereits **grundlegendes Konzeptwissen** vorhanden sein, um kreative Lösungsprozesse zu ermöglichen.
- Die Aufgabenstellung sollte **ideenanregend** gehalten sein.

ROMEIKE stellt weiterhin heraus, dass Schülerinnen und Schüler sich mit ihrer Beschäftigung identifizieren können müssen und dass Originalität im Sinne einer „eigenen Note“ in der Problembearbeitung und -lösung erlaubt sein sollte.<sup>28</sup>

Zur Realisierung eines kreativen Informatikunterrichts ist es notwendig, bestimmte Hemmschwellen zu vermeiden: „Schüler neigen in solchen [kreativitätshemmenden, Anm. A. S.] Situation[en] zu einem Problemvermeideverhalten, d. h. sie wählen sichere, ausgetretene Wege, die risikoarm und erfolgversprechend sind, aber für Kreativität nicht viel Raum lassen [...]. Frühe Evaluation und Perfektionsstreben sind in diesem Kontext als Kreativitätshemmer bekannt [...].“<sup>29</sup> Um solche Situationen zu vermeiden und stattdessen Kreativität im Informatikunterricht zu ermöglichen, sollten nach ROMEIKE vier weitere Aspekte der Unterrichtsumgebung beachtet werden:<sup>30</sup>

- **Experimentieren** – Die Schülerinnen und Schüler sollen die Möglichkeit haben, eigene Hypothesen zur Lösung eines Problems aufzustellen und sie experimentell – also am Produkt oder der Programmierumgebung – zu prüfen. Dazu zählt explizit auch das Konzept der Methode von „Versuch und Irrtum“.
- **Zeitlicher Raum** – Für den kreativen Unterrichtsverlauf ist es unabdingbar, Zeit für gedankliche Überlegungen und experimentelle Umsetzungen zur Verfügung zu stellen.
- **Unterrichtsklima der Vielfalt** – Kreative Leistungen sind naturgemäß individuell unterschiedliche Leistungen, das muss auch im Unterricht zum

---

<sup>27</sup> Vgl. (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 4ff.

<sup>28</sup> Vgl. (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 6

<sup>29</sup> (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 3

<sup>30</sup> Vgl. (Romeike, Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007, 2007), S. 6f.

Tragen kommen. Entsprechend dürfen Lösungsansätze nicht zu früh bewertet werden und es muss die Möglichkeit zur Vorstellung der unterschiedlichen Lösungen und ihrer Diskussion geben. Hierbei darf keinesfalls darauf abgezielt werden, eine „erwartete“ Leistung als Schablone zum Vergleich für die Schülerinnen- und Schülerleistungen zu benutzen.

- **Lehrerrolle** – Der Lehrer oder die Lehrerin hat die Aufgabe, den Schülerinnen und Schülern beratend, motivierend und orientierend zur Seite zu stehen. Eine starke Lenkung im Problemlöseprozess ist zu vermeiden, lediglich, wenn die Schülerinnen oder Schüler nicht von allein vorankommen, ist einzugreifen. Die Schülerinnen und Schüler sind von dem Lehrer beziehungsweise der Lehrerin zu vielfältigen Lösungen zu ermutigen.

## 2.3 SCRATCH IM UNTERRICHT

Die Programmiersprache Scratch eignet sich aus verschiedenen Gründen für den Einsatz im skizzierten kreativen Informatikunterricht.

Zunächst einmal ist Scratch, berücksichtigt man den Entstehungsprozess, eine „pädagogische“ Programmiersprache: Sie soll den Nutzer beim Erlernen der „21st Century Learning Skills“<sup>31</sup> unterstützen. Dabei handelt es sich um Fähigkeiten (und Wissen), welche von Bürgern nach Ansicht der gleichnamigen Initiative benötigt werden, um im einundzwanzigsten Jahrhundert im gesellschaftlichen Prozess bestehen zu können.<sup>32</sup> Scratch befähigt den Nutzer nach Ansicht der Autoren zu Fortschritten in drei Schlüsselbereichen mit insgesamt neun dieser geforderten Fähigkeiten:<sup>33</sup>

- „Information and Communication Skills“<sup>34</sup>: Hierzu zählen Fähigkeiten der Informationsbeschaffung und -verwaltung, Sicherheit im Umgang mit Medien und Fähigkeiten im überzeugenden Ausdrücken von Botschaften.
- „Thinking and Problem-Solving Skills“<sup>35</sup> umfasst die Fähigkeit zum kritischen Denken und zur Analyse von Systemen, zur Problemlösung (inklusive Problemidentifizierung, -formulierung, und -lösung), die Fähigkeit zum kreativen Handeln und zu intellektueller Neugierde.
- Mit „Interpersonal and Self-Directional Skills“<sup>36</sup> sind die Fähigkeiten gemeint, die eine zwischenmenschliche Zusammenarbeit, eine hohe Verbundenheit mit dem Produkt und die Rücksichtnahme auf die Bedürfnisse der anderen Menschen - in diesem Fall: der möglichen anderen Nutzer des Produkts - ermöglichen.

---

<sup>31</sup> Mehr Informationen unter <http://www.21stcenturyskills.org/>, Zugriff am 24. März 2008

<sup>32</sup> Vgl. die derzeitige Kompetenzdebatte, beispielsweise bei (Deutsches PISA-Konsortium: Baumert, Jürgen (Hrsg.), 2001), (Klieme, et al., 2007), (Weinert, 2001)

<sup>33</sup> Vgl. (Rusk, Resnick, & Maloney, Scratch 21st Century Skills, 2007)

<sup>34</sup> Übersetzung A. S.: „Medien- und Kommunikationskompetenzen“

<sup>35</sup> Übersetzung A. S.: „Kognitive und Problemlösekompetenzen“

<sup>36</sup> Übersetzung A. S.: „Soziale und Selbstkompetenzen“

Scratch wurde im Hinblick darauf entwickelt, diese Fähigkeiten zu fördern. Hierbei wurde darauf geachtet, die Programmiersprache für Jugendliche möglichst motivierend zu gestalten. Auch wegen der Kombination der bereits beschriebenen „Low-Floor“- , „High-Ceiling“- und „Wide-Walls“-Grundsätze bietet sich Scratch für den Einsatz im kreativen Informatikunterricht an: Der „Low-Floor“-Ansatz, vor allem der (annähernde) Ausschluss von Syntaxfehlern, ermöglicht einen schnellen Einstieg, „High Ceiling“ differenziertes Arbeiten und „Wide Walls“ vielfältige Möglichkeiten der Aufgabenstellung.

Nicht unerwähnt bleiben soll auch die Tatsache, dass Scratch kostenfrei erhältlich ist. Dies erhöht sicherlich, zusammen mit der Vielzahl an mitgelieferten Beispielen und Grafiken sowie dem auf der Scratch-Homepage verfügbaren Material, die Attraktivität für den Einsatz im Unterricht.

### 3 UNTERRICHTSVORAUSSETZUNGEN

Die bisher allgemein betrachteten Sachverhalte sollen im Folgenden auf die geplante Unterrichtsreihe übertragen werden. Dazu muss zunächst dargestellt werden, welche Voraussetzungen den Unterricht bestimmen.

#### 3.1 ALLGEMEINE UNTERRICHTSVORAUSSETZUNGEN

Es handelt sich um einen Wahlpflichtkurs Informatik der zehnten Jahrgangsstufe, den die Schülerinnen und Schüler seit der neunten Klasse besuchen. Zum Schuljahreswechsel fand ein Lehrerwechsel statt, seitdem unterrichte ich die Lerngruppe.

Der Wahlpflichtunterricht erfolgt mit zwei Stunden in der Woche, die als Doppelstunde dienstags in der dritten und vierten Unterrichtsstunde im Stundenplan verankert sind. Schulorganisatorisch bedingt kam es in der Vergangenheit häufig zum Ausfall einer oder beider Stunden.<sup>37</sup>

Der Unterricht findet in einem der Informatikräume der Schule (Raum 7) statt. Dieser Raum bietet Platz für etwa 20 Schülerinnen bzw. Schüler und hat eine äußere Tischreihe für die Arbeit am Computer<sup>38</sup> und eine innere Tischreihe für Plenumsphasen. Dadurch ist sichergestellt, dass die Schülerinnen und Schüler nicht permanent mit dem Rücken zur Tafel sitzen, allerdings wird es dadurch im Raum auch schnell sehr eng.

Technisch ist der Raum modern ausgestattet. Es gibt einen Lehrercomputer und für jeweils ein bis zwei Schülerinnen bzw. Schüler einen Schülercomputer. Die Computer sind technisch auf einem neuen Stand, sie wurden zum Schuljahreswechsel beschafft und eingerichtet. Alle Computer sind durch ein Netzwerk und außerdem mittels eines

---

<sup>37</sup> Beispiele hierfür wären: mehrwöchiges Betriebspraktikum, Informationsveranstaltungen der Schule, Klassenfahrt, Umorganisation des Unterrichts durch Lehrerausfall in anderen, parallelen Wahlpflichtkursen.

<sup>38</sup> Synonym für den Begriff „Computer“ wird im Folgenden auch der Begriff „Rechner“ benutzt.

Hochgeschwindigkeitszugangs mit dem Internet verbunden. Im Raum gibt es eine Tafel (Whiteboard) hinter dem Lehrerarbeitsplatz und einen Beamer, der mit dem Lehrere-computer verbunden ist und an eine der Wände projiziert, an der Schülerinnen bzw. Schüler sitzen.

In der Vergangenheit führten Netzwerkausfälle und andere technische Störungen häufig zu Verschiebungen in der Unterrichtsplanung, was vor allem bei programmier-technischen Unterrichtsinhalten hinderlich ist, da hier „Offline-Varianten“ nur begrenzt sinnvoll sind<sup>39</sup> und im Störfall die Motivation der Schülerinnen und Schüler, sich auf diese Notfallvariante einzulassen, gering ausgeprägt ist.

## 3.2 SPEZIELLE UNTERRICHTSVORAUSSETZUNGEN

### 3.2.1 LERNGRUPPE

Die Lerngruppe besteht aus vier Schülerinnen und zwölf Schülern aus zwei zehnten Klassen der Schule.

Die Lerngruppe zeigte in der Vergangenheit während des Unterrichtsgeschehens ein mittleres Leistungsvermögen bezogen auf die Erwartungen und ein eher durchschnittliches Interesse - bezogen auf die Einbringungsbereitschaft von Unterrichtsbeiträgen - an den Themen des Wahlpflichtunterrichts, wobei die Ausprägungen bei den einzelnen Schülerinnen und Schülern stark differierten:

- Die Schüler S1, S2 und S3 zeigten durchgängig ein starkes Interesse an den fachlichen Inhalten und erzielten meist gute bis sehr gute Arbeitsergebnisse.
- Die Schülerinnen bzw. Schüler S4, S5, S6, S7, S8, S9, S10, S11 und S12 bemühten sich, Ergebnisse gemäß den Anforderungen zu erzielen, erbrachten jedoch häufig eher durchschnittliche Resultate.
- Die Schüler S13 und S14 zeigten einerseits häufig eine hohe Ablenkungsbereitschaft, andererseits brachten sie des Öfteren originelle und sinnvolle Unterrichtsbeiträge hervor. Ihre Arbeitsergebnisse beschränkten sich meist auf die reine Erfüllung der Minimalanforderungen.
- Die Schülerinnen S15<sup>40</sup> und S16 bemühten sich, am Unterrichtsgeschehen teilzunehmen. Sie besitzen jedoch große stofflich-inhaltliche Lücken, welche es ihnen nur selten ermöglichten, zielorientiert voranzukommen und anforderungsgemäße Ergebnisse zu erbringen. Bei S15 kommen sprachliche Hürden Verständnis erschwerend hinzu.

---

<sup>39</sup> Natürlich gibt es auch in der Programmierung Situationen, die ohne Rechnereinsatz auskommen (z. B. in der Modellierung). Aber es gibt eben auch genügend Situationen, in denen vor allem ein plötzlicher Netz- oder Rechnerausfall eigentlich nicht sinnvoll kompensiert werden kann (z. B. bei der direkten Implementierung und dem Testen von Quellcode).

<sup>40</sup> S15, eine Schülerin mit koreanischem Hintergrund, heißt eigentlich „### #####“, möchte aber mit „S15“ angesprochen werden und unterschreibt auch grundsätzlich mit diesem Namen. Aus diesem Grund wird auch hier und im Folgenden „S15“ statt „### #####“ benutzt.

Die Anstrengungsbereitschaft einiger Schülerinnen und Schüler (hier sind vor allem S13, S14 und S16 betroffen) erscheint eher gering ausgeprägt. Hier scheinen die Annahmen, die zur Kurswahl geführt haben, nicht zu den tatsächlichen Inhalten des Wahlpflichtunterrichts zu passen, wobei dies von den betreffenden Schülerinnen und Schülern auch nicht näher spezifiziert werden konnte.

Insgesamt erscheint es günstig, die Themenwahl stärker auf die kognitiven Fähigkeiten der Lerngruppe abzustimmen, um motivationale Verstärkung hervorzurufen.

### 3.2.2 ERFahrungen mit PARTNERARBEIT UND MIT PRÄSENTATIONEN

Im Wahlpflichtfach Informatik findet die Arbeit an den Computern meistens in Partnerarbeit statt. Ursprünglicher Grund hierfür ist, dass nur so ausreichend Computer für die Arbeit zur Verfügung stehen. Die Schülerinnen und Schüler wählen sich selbst ihren Partner aus und gestalten dann gemeinsam das Produkt am Rechner. Hierbei gibt es normalerweise nur wenige Konflikte und es ist zu beobachten, dass sich die Partner im Bearbeitungsprozess gut ergänzen. Kommunikation zwischen den Partnergruppen findet dabei ebenfalls statt, meistens gestaltet sich das zum Nutzen für die Qualität der Produkte. Diese bewährte Unterrichtsform soll auch in der vorliegenden Unterrichtsreihe die Produkterstellungsphasen dominieren.

Bei den bisher erfolgten Präsentationen von Ergebnissen zeigte sich ein Großteil der Lerngruppe unsicher und ungeübt. Dies äußerte sich häufig darin, dass Kommentare zu den Überlegungen zum Produkt fehlten oder nur auf intensive Nachfrage geäußert wurden. Aufgrund des sonst angenehmen Lernklimas entstand bei mir der Eindruck, dass der Grund hierfür nicht mangelnder Mut zur Präsentation, sondern Unsicherheit darin ist, welche Informationen für die Lerngruppe relevant sind. Zur Verbesserung dieser Situation passt es demnach in dieser Unterrichtsreihe gut, dass Präsentationen von Arbeitsergebnissen ein wichtiger Bestandteil des Konzepts sind.

### 3.2.3 BISHERIGE PROGRAMMIERERFAHRUNGEN IM WAHLPFLICHTUNTERRICHT

Im bisherigen Unterrichtsverlauf konnten die Schülerinnen und Schüler bereits Erfahrungen mit verschiedenen anderen Programmiersprachen sammeln: Zur Programmierung von LEGO-Robotern wurde in der neunten Jahrgangsstufe die Mini-Sprache „NQC“ verwendet, eine auf die Steuerung solcher Roboter zugeschnittene, C-artige Sprache. In der zehnten Jahrgangsstufe wurde die Programmiersprache „LOGO“ unmittelbar vor der darzustellenden Unterrichtsreihe eingeführt. Dabei handelt es sich um eine Programmiersprache, die befehlsgesteuert grafische Ausgaben vornimmt.

Im Verlauf dieser Unterrichtsreihen erschien es so, als würde ein größerer Teil der Lerngruppe lediglich versuchen, den Anschluss nicht zu verlieren. Ein Grundverständnis der informatisch relevanten Programm- und Programmierstrukturen schien sich jedoch nicht aufzubauen. Es ergab sich - bestätigt durch Schüleräußerungen - der Eindruck, dass die grundlegenden Konzepte nicht verinnerlicht wurden, sondern eher „ratend“ programmiert wurde.

Bei nicht wenigen Schülerinnen und Schülern konnte, ebenfalls belegt durch Schüleräußerungen, festgestellt werden, dass die innere Motivation, sich mit den jeweiligen Thematiken auseinanderzusetzen, nur sehr begrenzt vorhanden war.

Weitere, noch zu behandelnde Programmiersprachen sollten als Konsequenz stärker an den Bedürfnissen der Lerngruppe ausgerichtet sein.

#### 3.2.4 WEITERE THEMEN IM BISHERIGEN UNTERRICHT

In der neunten Jahrgangsstufe wurden außerdem die Themen „Künstliche Intelligenz/Chatbots“ und „Schadsoftware: Viren, Würmer und Trojaner“ behandelt.<sup>41</sup>

In der zehnten Jahrgangsstufe bildete das Thema „Datensammeln und Datenschutz“ einen Schwerpunkt. Hier zeigten die Schülerinnen und Schüler ein hohes Interesse am Thema und arbeiteten entsprechend mit. Allerdings waren auch hier wieder Unterschiede zu bemerken: Während vor allem einige Schülerinnen eine weitere Vertiefung des Themas wünschten, zeigten andere Schüler (vor allem S1 und S3) kein Interesse daran, sondern wollten sich wieder „traditionellen“<sup>42</sup> Themen der Informatik zuwenden.

Es gilt also für den weiteren Unterricht, eine Balance zwischen darstellendem und analytisch-programmierendem Unterricht zu finden.

## 4 SACHANALYSE

### 4.1 AUSWAHL DES PRODUKTS

Die Programmiersprache Scratch ist darauf ausgerichtet, verschiedene Arten von Programmen zu erstellen: z. B. Animationen, Spiele und Filme. Ihnen gemeinsam ist die Betonung von Grafik (und Sound) und die algorithmisierte Ablaufsteuerung.

Ich habe mich dafür entschieden, in dieser Unterrichtsreihe den Schwerpunkt auf die Programmierung eines Adventure-Spiels (ausgehend von einer Kurzanimation) zu legen, da dieser Schwerpunkt zwei Bereiche, nämlich „Animation“ und „Spiel“, berührt und damit eine größtmögliche Vielfalt entlang der Aufgabenstellung zulässt. Dies lässt erwarten, dass die Schülerinnen und Schüler stärker motiviert werden, da es für sie subjektiv relevant erscheint.

---

<sup>41</sup> Den damaligen Unterrichtsverlauf konnte ich größtenteils hospitierend verfolgen.

<sup>42</sup> Damit ist nicht gemeint, dass das Gebiet „Informatik und Gesellschaft“, zu dem „Datensammeln und Datenschutz“ gehört, kein Themenfeld der Informatik ist. Im Gegenteil, es scheint an Bedeutung in der letzten Zeit eher gewonnen zu haben. Aus Sicht der Schülerinnen und Schüler scheint es jedoch eher untergeordnet zur Informatik zu gehören, unter „traditionell“ werden hier „Programmierung“ und verwandte Themen verstanden.

## 4.2 INFORMATISCHE KONZEPTE

Neben vielen informatischen Konzepten, die bei der Programmierung mit Scratch quasi zwangsläufig einfließen,<sup>43</sup> habe ich den Schwerpunkt für die geplante Unterrichtsreihe auf drei Grundkonzepte („Botschaften“, „boolesche Variablen“, „Ganzzahlvariablen“) gelegt. Scratch eignet sich gut für die Auseinandersetzung mit diesen Konzepten, da die entsprechenden Bausteine in vielen Programmierkontexten von Scratch verwendet werden. Umgekehrt sind die Konzepte in nahezu jeder Programmiersprache von Bedeutung, was eine fokussierte Auseinandersetzung mit ihnen rechtfertigt.

### 4.2.1 BOTSCHAFTEN

Die „Botschaft“ - oder synonym „Nachricht“ - ist ein „[z]entraler Begriff in der objektorientierten Programmierung. [...] Ein Objekt schickt einem anderen Objekt eine Nachricht, um dieses Objekt zu einer Aktivität [...] zu veranlassen.“<sup>44</sup> Außerdem erledigen Objekte „[...] Aufgaben prinzipiell in eigener Verantwortung, d. h. sie lassen sich nicht vom fragenden Objekt aufzwingen, wie sie eine Aufgabe zu erledigen haben, sondern man kann ihnen nur sagen, was sie erledigen sollen.“<sup>45</sup>

In Scratch - als vom Ansatz her objektorientierter Sprache - lässt sich Objektverhalten ebenfalls über Botschaften steuern. Dazu existieren zwei Bausteine zum Senden von Nachrichten („*Sende <BOTSCHAFT> an alle*“ und „*Sende <BOTSCHAFT> an alle und warte*“) und ein Baustein zum Empfangen von Nachrichten („*Wenn ich <BOTSCHAFT> empfangen*“). Das Vorhandensein zweier Bausteine zum Senden kann damit erklärt werden, dass dadurch die Möglichkeit zu unterschiedlicher Ablaufsteuerung gegeben ist: Es werden parallele („*Sende ...*“) oder lineare („*Sende ... und warte*“) Abläufe möglich. Ungünstig ist die Beschaffenheit des Bausteins zum Empfang von Nachrichten. Dieser lässt sich nur zum Start eines Anweisungsblocks verwenden. Dies entspricht dem Konzept, dass Nachrichten ereignisgesteuert verbreitet werden und dass durch Nachrichten Aktionen ausgelöst werden, ermöglicht aber nicht auf trivialem und intuitivem Weg eine nachrichtenabhängige Steuerung.<sup>46</sup>

Im vorgesehenen Adventure ist das Konzept des „Regisseurs“ von zentraler Bedeutung. Das zugrunde liegende Problem ist, dass gleiche Steuerungseingaben - beispielsweise über die Tastatur - in Abhängigkeit vom Zustand, in dem das Programm sich gerade befindet (wie weit die Geschichte fortgeschritten ist und welchen Verlauf sie genommen hat), unterschiedliche Reaktionen auslösen müssen. Insofern lassen sich die in Scratch vorgegebenen Bausteine „*Wenn Taste <TASTE> gedrückt*“ zur Reaktionssteuerung nicht

---

<sup>43</sup> Beispiele wären hier: Schleifen, Anfangs- und Terminierungspunkte und -bedingungen, Objekte mit Attributen und Methoden

<sup>44</sup> (Claus & Schwill, Duden Informatik, 2003), S. 431, Stichwort „Nachricht“

<sup>45</sup> (Claus & Schwill, Schülerduden Informatik, 2003), S. 341, Stichwort: „objektorientierte Programmierung“, Hervorhebung im Original

<sup>46</sup> Ein Beispiel für eine nachrichtenabhängige Steuerung wäre das Konstrukt „*Wiederhole <ANWEISUNG1> bis <BOTSCHAFT> eintrifft. Danach führe <ANWEISUNG 2> aus.*“

angemessen verwenden, da sie situationsunabhängig stets identisch reagieren.<sup>47</sup> Mit Botschaften lassen sich dagegen solche Abläufe einfach und intuitiv steuern, dafür muss jedoch ein Handelnder, der „Regisseur“ geschaffen werden, der entlang des Programmverlaufs und abhängig von Tastatur- oder anderen Steuerungseingaben unterschiedliche Botschaften an die Objekte sendet, welche dann den Anforderungen angemessen reagieren. Dieser „Regisseur“ kann im Quellcode-Bereich der Bühne angesiedelt werden, es lässt sich hier eine Analogie zum Theater erkennen: Der „Regisseur“ steuert hinter der Bühne, wie sich die einzelnen Akteure verhalten.

#### 4.2.2 BOOLESCHE VARIABLEN

Boolean ist ein „vordefinierter Datentyp in vielen Programmiersprachen. Variablen vom Typ boolean können die Werte false (falsch) und true (richtig) annehmen. Zum Datentyp boolean gehören stets die üblichen booleschen Funktionen[. ...] Da die Werte von Bedingungen vom Typ boolean sind, existiert dieser Datentyp in allen Programmiersprachen[. ...]“<sup>48</sup>

In Scratch existieren Variablen formal ohne ausgewiesenen Datentyp, allerdings gibt es die Bedingung, dass sie lediglich mit Zahlenwerten belegt werden dürfen.<sup>49</sup> Man kann diese Variablen unter anderem als boolesche Variablen einsetzen, indem man - wie auch sonst in der Informatik üblich - den Wert „false“ mit der Zahl „0“ und den Wert „true“ mit der Zahl „1“ identifiziert.<sup>50</sup>

Im Adventure lässt sich zum Beispiel der Besitz eines Gegenstands somit dadurch anzeigen, dass der Variablenwert bei Inbesitznahme um eins erhöht wird und bei Verlust um eins verringert wird. Der Variablenwert „0“ zeigt also an, dass die Figur nicht im Besitz des Gegenstandes ist und der Wert „1“ bedeutet, dass die Figur den Gegenstand aufgenommen hat.

#### 4.2.3 GANZZAHLVARIABLEN

„Variablen vom Typ integer können als Werte grundsätzlich ganze Zahlen annehmen.“<sup>51</sup> In der Informatik werden Ganzzahlvariablen häufig auch als Integer-Variablen bezeichnet. In der Vergangenheit waren - bedingt durch nur wenig vorhandenen Speicherplatz - nur stark eingegrenzte Wertebereiche für Ganzzahlvariablen möglich, weshalb man den Begriff „Ganzzahlvariable“ vermied. Natürlich ist es auch heute nicht möglich, die (mathematisch definierte) Menge der ganzen Zahlen vollständig im Computer innerhalb eines Datentyps abzubilden, da die Ressourcen nach wie vor endlich sind. Allerdings

---

<sup>47</sup> Es gäbe die Möglichkeit, beispielsweise über variabelengesteuerte Fallunterscheidungen unterschiedliche Reaktionsabläufe zu gestalten. Dies würde jedoch zu einem weniger intuitiv verständlichen Quellcode führen und würde einer Modellierung des Sachverhalts „Adventure-Spiel“ bzw. „Kurzgeschichte“ nicht gerecht werden.

<sup>48</sup> (Claus & Schwill, Duden Informatik, 2003), S. 106, Stichwort: „boolean“

<sup>49</sup> Zeichenketten als Variablenwerte existieren in Scratch also beispielsweise nicht.

<sup>50</sup> Theoretisch wären auch andere Zuordnungen denkbar, jedoch ist die „0-1“-Zuordnung sehr praktisch in der Handhabung.

<sup>51</sup> (Claus & Schwill, Duden Informatik, 2003), S. 308, Stichwort: „integer“

ist es inzwischen möglich, dass derart große (und kleine) ganze Zahlen als Datentypausprägung zur Verfügung stehen, dass im praktischen Alltag von Ganzzahlvariablen gesprochen werden kann.

Bezogen auf die Programmierung mit Scratch lassen sich Ganzzahlvariablen als eine Erweiterung der booleschen Variablen auffassen: Nach wie vor gibt es die Möglichkeit, Variablenwerte zu erhöhen und zu verringern, allerdings sind die Start- bzw. Zielwerte nicht mehr auf „1“ und „0“ beschränkt und es können auch nicht mehr nur Erhöhungen oder Verringerungen um eins - sondern auch darüber hinaus - vorgenommen werden.

Mit dem Konzept der Ganzzahlvariablen lassen sich im Adventure etwa „mehrere Leben“ beim „Helden“ realisieren. Dies entspricht mehreren Versuchen zum Durchlaufen der Geschichte im Versuch, ein „Ziel“ zu erreichen. Dabei werden im Falle des Scheiterns „Leben“ abgezogen. Das Spiel ist endgültig vorbei, wenn kein verfügbares „Leben“ mehr vorhanden ist.<sup>52</sup>

## 5 DIDAKTISCHE ANALYSE

### 5.1 BEZUG ZUM RAHMENLEHRPLAN

Der Rahmenlehrplan für das Wahlpflichtfach Informatik legt sich in Bezug auf das Unterrichtsvorhaben nicht stark fest, allerdings lassen sich doch einige Ankerpunkte finden. Die Unterrichtsreihe hat ihren Schwerpunkt in den Wahlpflicht-Modulen 1 („Aufbau und Wirkungsweise von Informatiksystemen“) und 5 („Multimedia“),<sup>53</sup> da die Programmiersprache Scratch zur Erstellung von Animationen und kleinen Adventure-Spielen kennengelernt und eingesetzt werden soll.

Die Erstellung eines Adventure-Spiels mithilfe der vorgegebenen Bausteine der Programmiersprache Scratch findet sich im **Kompetenzbereich Fachwissen** wieder: „Die Schülerinnen und Schüler [...] verstehen Software als Komplex von Objekten [...] und erstellen kleine Programme unter Benutzung vorgegebener Bausteine[.]“<sup>54</sup> Durch die Verwendung von Scratch können somit wie oben beschrieben grundlegende informatische Konzepte anschaulich gemacht, kognitiv durchdrungen und kreativ ausprobiert werden.

Um die Handhabung der Programmiersprache zu erlernen, werden die Online-Dokumentationen zu Scratch, inklusive auf der Scratch-Homepage verfügbarer Videos, zum Einsatz kommen. Während des Entwicklungsprozesses ihrer eigenen Software und

---

<sup>52</sup> Je nach Auffassung kann es hierbei auch tatsächlich nötig sein, in den Bereich der negativen Zahlen vorzudringen: Werden nämlich die „Lebenswerte“ als Anzahl der **nach** zur Verfügung stehenden Versuche gewertet, so ergeben „0 Leben“ einen Sinn (nämlich kein **weiterer** Versuch) und damit kann erst bei „-1 Leben“ abgebrochen werden.

<sup>53</sup> Vgl. (Senatsverwaltung für Bildung, Jugend und Sport Berlin, 2006), S. 25f., 30

<sup>54</sup> (Senatsverwaltung für Bildung, Jugend und Sport Berlin, 2006), S. 10

im Anschluss daran erstellen die Schülerinnen und Schüler selbst Dokumentationen zu ihrem Projekt. Außerdem testen und verbessern sie ihr Produkt schrittweise. Diese Aspekte werden im **Kompetenzbereich Erkenntnisgewinnung** eingefordert: „Die Schülerinnen und Schüler [...] lesen Benutzerdokumentation (Handbücher) von Software und schreiben solche für von ihnen selbst erstellte Systeme, [...] analysieren eigene Arbeiten im Einsatz, gewinnen Erkenntnisse über deren Auswirkungen und dokumentieren sie, erkennen verschiedene Typen, Ursachen und Auswirkungen von Fehlern in eigenen Programmen [und] erfahren die extreme Begrenztheit der ‚Verständnisfähigkeit‘ der Maschine beim Entwurf und Test eigener Problemlösungen[.]“<sup>55</sup>

Während und nach Erstellung der Spiele werden diese von den Partnern und den anderen Mitgliedern der Lerngruppe getestet. Im Erstellungsprozess müssen sich die Partner ihre Aufgaben an verschiedenen Stellen aufteilen, um Leerlauf zu vermeiden. Beide Gesichtspunkte erfordern Überlegungen über die dazu nötigen Schnittstellen, einerseits im Bereich der Aufteilung und -zusammenführung, andererseits im Bereich der Ersteller-Nutzer-Kommunikation. Die Auseinandersetzung mit beidem sieht der **Kompetenzbereich Kommunikation** auch vor: „Die Schülerinnen und Schüler erkennen, analysieren und betreiben aktiv Kommunikationsvorgänge [...]: Kommunikation zwischen dem Programmierer und dem Programmbenutzer in Form der vom Programmierer vorhergesehenen Kommunikationswege und der entsprechend gestalteten Benutzeroberfläche, Kommunikation zwischen Arbeitsgruppenmitgliedern bei der Verabredung von Schnittstellen für Teilergebnisse, Präsentation und Diskussion von Arbeitsgruppenergebnissen.“<sup>56</sup>

Die Schülerinnen und Schüler erstellen im Rahmen der ihnen verfügbaren Zeit lediglich kleine Animationen bzw. Adventure-Spiele. Sie erkennen, dass bereits dafür ein relativ hoher Zeitaufwand nötig ist. Dies kann zu einer hohen Wertschätzung für die Arbeit anderer Programmierer, die ihre Software kostenfrei zur Verfügung stellen, führen. Ebenso kann die Auseinandersetzung damit dazu führen, dass Fragen nach dem Zweck einer kostenfreien Verbreitung von Software und den Motiven der Anbieter aufgeworfen werden, wie dies im **Kompetenzbereich Bewertung** angedacht ist: „Die Schülerinnen und Schüler [...] berücksichtigen bei der Bewertung von kostenloser Programmierarbeit den beabsichtigten Zweck[. ...]“<sup>57</sup>

---

<sup>55</sup> (Senatsverwaltung für Bildung, Jugend und Sport Berlin, 2006), S. 11

<sup>56</sup> (Senatsverwaltung für Bildung, Jugend und Sport Berlin, 2006), S. 12

<sup>57</sup> (Senatsverwaltung für Bildung, Jugend und Sport Berlin, 2006), S. 13

## 5.2 DIDAKTISCHE REDUKTION

Die Schülerinnen und Schüler der Lerngruppe sollen durch diese Unterrichtsreihe befähigt werden, in der Programmiersprache Scratch einfache Programme zu erstellen und dabei grundlegende informatische Konzepte zu verwenden. Aus der Vielzahl an möglichen Programmarten, die mit Scratch erstellt werden können, sind nun solche auszuwählen, welche nicht nur diese informatischen Konzepte umsetzen, sondern auch die progressive Entwicklung von selbstständigem Arbeiten fördern und dabei an den Wissensstand und nach Möglichkeit die Lebenswelt der Schülerinnen und Schüler anknüpfen. Hier bietet sich die Erstellung eines Adventure-Spiels an, in dem der Nutzer mehrere Handlungsoptionen und sogenannte „Leben“ (Versuche zum Durchspielen) hat und in dem „Items“ (gewisse Zusatzgegenstände) für die Lösung des Spiels benötigt werden.

Die Kompetenz zur Erstellung eines so komplexen Programms muss jedoch schrittweise entwickelt werden. Als Vorstufe eignen sich zunächst einfache Animationen mit zunehmend komplexem Aufbau. Für das daraufhin zu programmierende mehrdimensionale Adventure-Spiel erstellen die Schülerinnen und Schüler die benötigten Grafiken selbst oder besorgen sie sich aus anderen legalen Quellen. Sie stellen ihr Produkt inklusive einer von ihnen hergestellten Dokumentation vor und diskutieren mit anderen Mitgliedern der Lerngruppe die Ansätze, Vor- und Nachteile ihrer Lösung.

Da die Schülerinnen und Schüler schrittweise an das Programm und seine Möglichkeiten zur Umsetzung informatischer Konzepte und individueller Gestaltung herangeführt werden sollen, wird eine hierfür zu entwickelnde Unterrichtsreihe verhältnismäßig zeitintensiv sein. Geplant ist es, insgesamt vierzehn Unterrichtsstunden hierfür zu verwenden, wobei Präsentations- und Dokumentationsphasen unter Berücksichtigung der entsprechenden Arbeitshypothesen explizit einen großen Raum einnehmen sollen.

Differenzierung innerhalb der Lerngruppe ist vor allem durch die flexible Bearbeitungstiefe der Aufgabenstellung möglich. Hervorzuheben ist auch, dass die Programme von den Schülerinnen und Schülern in Partnerarbeit erstellt werden sollen, was Kommunikationskompetenz und die Fähigkeit, die eigenen Vorstellungen mit denen des Partners abzustimmen, gegebenenfalls zu revidieren und gemeinsam umzusetzen, fördert.

## 6 METHODISCHE ÜBERLEGUNGEN

### 6.1 PLANUNG DER UNTERRICHTSREIHE NACH KREATIVEN MAßSTÄBEN

Um einen Unterricht nach kreativen Maßstäben zu ermöglichen und damit die Fähigkeit zum selbstständigen Arbeiten zu fördern, müssen verschiedene, bereits aufgeführte

Gesichtspunkte beachtet werden.<sup>58</sup> Diese sollen nun für die geplante Unterrichtsreihe konkretisiert werden.

### ***Relevanz des Unterrichtsgegenstandes***

Scratch als Programmiersprache ist optisch ansprechend gehalten und leicht zu erlernen. Die mit Scratch programmierbaren Projekte lassen sich im Multimedia- und Spiele-Bereich einordnen und werden als solche für Schülerinnen und Schüler relevant erscheinen. Das beabsichtigte Thema - die Programmierung eines Mini-Adventures aus einer Animation heraus - sollten die Schülerinnen und Schüler als interessant und herausfordernd wahrnehmen.<sup>59</sup>

### ***Aufgabenstellungen mit Problemlösungs- oder Problemerschaffungsprozessen***

- *Subjektive Neuheit:* Die vorgesehenen Aufgaben wurden in dieser Form von den Schülerinnen und Schülern noch nicht bearbeitet. Die subjektive Neuheit ist schon dadurch gegeben, dass die verwendete Programmiersprache für alle Schülerinnen und Schüler neu ist.<sup>60</sup> Auch haben die Schülerinnen und Schüler im bisherigen Unterrichtsverlauf - und auch im Privatleben, soweit mir bekannt ist - noch keine Animationen oder Adventure-Spiele erstellt.
- *Offene Aufgabenstellungen:* Die Offenheit in der Aufgabenstellung kann dadurch erreicht werden, dass den Schülerinnen und Schülern die Inhalte der darzustellenden Geschichte inklusive des konkreten Ablaufs und der visuellen Ausgestaltung freigestellt werden, die Umsetzung eines informatischen Konzepts jedoch verlangt wird. Dies wird dann am besten gelingen, wenn dieses Konzept für das weitere Voranschreiten innerhalb des Projekts als notwendig, intuitiv und zwangsläufig erscheint.
- *Flexible Bearbeitungstiefe:* Es steht den Schülerinnen und Schülern frei, wie weit sie ihr Projekt über die Minimalanforderungen hinaus ausbauen. Die zu behandelnden informatischen Konzepte sind grundlegender Art und als solche können sie mehrfach an verschiedenen Stellen und auf unterschiedliche Art und Weise zum Einsatz kommen. Außerdem lassen sich in der optischen Ausgestaltung und im inhaltlichen Ablauf des Programms beliebige Verfeinerungen, Erweiterung oder Optimierungen vornehmen. Es ist damit zu rechnen, dass die Schülerinnen und Schüler wegen der ansprechenden Gestaltung der Programmiersprache sich nach Erreichen der Minimalanforderungen aus intrinsischer Motivation noch weiter mit ihrem Projekt und dessen Modifikationen auseinandersetzen wollen.

---

<sup>58</sup> Vgl. Kapitel 2.2 dieser Arbeit

<sup>59</sup> Spiele dieser Art sind den Schülerinnen und Schülern im Alltag als sogenannte „Flash-Online-Spiele“ bekannt.

<sup>60</sup> Dieser Sachverhalt wurde im Vorfeld sichergestellt.

- *Vorhandensein grundlegenden Konzeptwissens:* Dadurch, dass in der Vorgeschichte mehrere Programmiersprachen zum Einsatz kamen, dürfte ein gewisses Konzeptwissen (beispielsweise über Programmabläufe, Steuerungssequenzen, Aufbau von Quellcode usw.) bereits vorhanden sein, an das angeknüpft werden kann.
- *Ideenanregende Aufgabenstellung:* Die Aufgabenstellung bezieht die Fantasie der Schülerinnen und Schüler explizit mit ein, da sie sich die Geschichte und ihre Ausgestaltung selbst ausdenken müssen. Dadurch dürfte die Motivation zur kreativen Umsetzung erhöht werden.

### ***Experimentieren***

Grundsätzlich ist es vorgesehen, dass die Schülerinnen und Schüler in Einzel- oder Partnerarbeit an ihren Projekten arbeiten, wobei Partnerarbeit als sinnvoller erscheint, weil sich so die Fähigkeiten zweier Partner ergänzen und effizienter als in Einzelarbeit Lösungen gefunden werden.<sup>61</sup> Dabei ist es geplant, dass die Gruppen, die sich zu Beginn finden, möglichst auch bis zum Schluss zusammenarbeiten, um Kontinuität bei der inhaltlichen und technischen Fortführung des Adventures zu erreichen. Dies kann außerdem dazu führen, dass sich eine gewisse Konkurrenz um das Finden der originellsten Story oder der beeindruckendsten technischen Lösung herausbildet, was den kreativen Prozess noch verstärken kann.<sup>62</sup> Grundsätzlich soll die Unterrichtsreihe so angelegt werden, dass die Schülerinnen und Schüler die Möglichkeit haben, ihre Ideen kurzschrittig und direkt in der Programmiersprache umzusetzen. Dies ermöglicht das schnelle Auffinden möglicher inhaltlicher Fehler oder Probleme und eine angemessene Reaktion darauf. Eventuelle Phasen der (ausschließlichen) Ideenfindung ohne direkte programmiertechnische Überprüfung sollten möglichst eng mit den anderen Phasen verzahnt werden.

### ***Zeitlicher Raum***

Es ist damit zu rechnen, dass mehr Zeit für die Unterrichtsreihe eingeplant werden muss, als dies bei einem Unterricht ohne Betonung kreativer Maßstäbe der Fall wäre. Der Motivationsgewinn durch einen Unterricht nach kreativen Maßstäben wird sich auch dadurch ausdrücken, dass die Schülerinnen und Schüler an Stellen verändern, gestalten und optimieren wollen, die für das zu vermittelnde informatische Konzept nur unter-

---

<sup>61</sup> Es ist zu erwarten, dass zu manchen Terminen Schülerinnen oder Schüler fehlen (z. B. durch Krankheit), insofern muss mit Phasen der Einzelarbeit gerechnet werden. Die Alternative, dass die betreffende Schülerin bzw. der betreffende Schüler dann in einer anderen Gruppe mitarbeitet, ist abzulehnen, da dadurch schrittweise die „getrennten“ Gruppen ihre eigenen Projekte in der Zukunft nicht mehr fortführen würden. Ebenso kann es in Einzelfällen zur Bildung von Dreier-Gruppen kommen, für die trotzdem der Begriff Partnerarbeit verwendet wird, da die intendierten ablaufenden Prozesse vergleichbar mit denen der anderen Zweier-Gruppen sind.

<sup>62</sup> Es kann natürlich auch dazu führen, dass dadurch Hemmschwellen bei vermeintlich „zurückfallenden“ Gruppen aufgebaut werden, hier ist durch eine angemessene Steuerung des Unterrichtsgeschehens gegebenenfalls zu intervenieren.

geordnet von Bedeutung sind.<sup>63</sup> Auch die Kommunikation in der Gruppe und zwischen den Gruppen, sowie Präsentation und Dokumentation fordern zusätzlichen zeitlichen Raum ein. Im Gegenzug ist damit zu rechnen, dass die Ergebnisse des Lernprozesses tiefer verankert werden und kritischer reflektiert werden können.

### ***Unterrichtsklima der Vielfalt***

Es ist notwendig, den Schülerinnen und Schülern zu verdeutlichen, dass eine „richtige“ Lösung der Aufgabenstellung sehr unterschiedlich aussehen kann. Dieser Ansatz ist im bisherigen Informatikunterricht in weniger intensiver Ausprägung häufig präsent gewesen, beispielsweise können verschiedene Algorithmen ein vorgegebenes Problem lösen. Meist stellt sich jedoch heraus, dass ein spezieller Algorithmus „besser“ als ein anderer ist, beispielsweise weil er ressourcenschonender, sicherer oder schneller ist. Dieser Effizienzgedanke ist nicht abzulehnen und wird auch in der vorgesehenen Unterrichtsreihe eine Rolle spielen. Es muss jedoch verdeutlicht werden, dass eben nicht nur Effizienz bestimmend sein kann, sondern auch Problemangemessenheit: Für verschiedene Umsetzungen eines Problems und seiner Lösung kann es durchaus sinnvoll oder sogar notwendig sein, völlig verschiedene Wege zu beschreiten. Wichtig ist dann, dass in Phasen der Darstellung der Resultate eine fundierte Diskussion möglich ist, in der die Entscheidungsgrundlagen für oder gegen ein bestimmtes Vorgehen begründet dargelegt werden können, sodass die gesamte Lerngruppe die Möglichkeit bekommt, neue Sichtweisen wahrzunehmen und abzuwägen. Diese Phasen sollten erst nach angemessener Zeit im Unterricht stattfinden, damit Reifungsprozesse möglich sind. In der kritischen Diskussion muss die individuelle Umsetzung gewürdigt werden, die „eigene Note“ darf nicht Bestandteil der Kritik werden.<sup>64</sup>

### ***Lehrerrolle***

In weiten Teilen des Unterrichts in dieser Reihe wird es notwendig sein, dass der Lehrer sich in seiner Kontrolle des Unterrichtsgeschehens zurücknimmt. Dies bedeutet nicht, dass Chaos oder Beliebigkeit intendiert sind. Vielmehr muss es den Schülerinnen und Schülern freistehen, auf welche Art sie sich dem Problem und seiner Realisierung nähern, wie viel Zeit sie für die einzelnen Phasen veranschlagen und wie sie sich in der Umsetzung ihrer Ziele selbst kontrollieren. Hierbei wird zu Beginn noch eine stärkere Steuerung notwendig sein, die im weiteren Verlauf mehr und mehr zurückgenommen wird. Umgekehrt muss der Lehrer in der für ihn hinzugewonnenen Zeit als möglicher Berater und Impulsgeber zur Verfügung stehen und im Blick behalten, dass Ablenkungen

---

<sup>63</sup> Diese Motivation ist notwendig für Unterricht nach kreativen Maßstäben. Insofern wäre es nicht sinnvoll, diese Problematik vermeiden zu wollen. Auf der anderen Seite muss darauf geachtet werden, dass Prozesse wie das Ausgestalten von Bildern zu den Charakteren möglicherweise teilausgelagert werden, um den zur Verfügung stehenden zeitlichen Rahmen nicht vollständig zu sprengen.

<sup>64</sup> Ausnahmen sind hier vorstellbar: Sollten die Schülerinnen und Schüler in der Umsetzung gesellschaftliche Normen oder Grundwerte verletzen, müsste natürlich eingegriffen werden.

keine zu große Rolle spielen. Die Grundstrukturierung der Unterrichtsstunden wird außerdem nach wie vor im Aufgabenfeld des Lehrers liegen. Hier ist es vorgesehen, vor allem in Phasen der Problemstellung und der Reflexion bzw. Präsentation moderierend tätig zu werden, um sicherzustellen, dass Ausgangslagen klar verstanden werden und in Präsentationsphasen Redezeiten gerecht verteilt sind.<sup>65</sup>

## 6.2 ONLINE-MEDIENEINSATZ

### 6.2.1 EINSATZ DER ONLINE-COMMUNITY SCRATCH-ONLINE

Bei der Arbeit mit Scratch bietet es sich an, die Anbindung an Scratch-Online zu nutzen. Die Schülerinnen und Schüler können auf diesem Weg den aktuellen Stand ihrer Programme sichern und den anderen Scratch-Nutzern präsentieren. Auch sind gegebenenfalls Online-Diskussionen mit anderen Nutzern von Scratch-Online möglich. Bezogen auf die eigene Lerngruppe ist es außerdem von Vorteil, dass sich „befreundete“ Scratch-Nutzer verlinken können (als sogenannte „Buddies“) und so schnell die Ergebnisse dieser Nutzer eingesehen werden können.

Da den Schülerinnen und Schülern die Programmiersprache Scratch bisher unbekannt ist, kennen sie auch Scratch-Online noch nicht. Sie haben allerdings im privaten Bereich bereits mehrere Online-Communities<sup>66</sup> kennengelernt und nutzen sie teilweise sehr intensiv. Insofern dürfte sich die Nutzung von Scratch-Online nicht als problematisch herausstellen. Im Gegenteil, die Schülerinnen und Schüler haben dadurch die Möglichkeit, solche Online-Communities als eine Möglichkeit zur Präsentation und Überprüfung eigener Produkte vor einer breiten Öffentlichkeit von interessierten Nutzern zu erfahren.

### 6.2.2 EINSATZ DES ONLINE-ARBEITSBEREICHES BSCW

Für die Dokumentation von Projekten, also die Beschreibung der Ergebnisse, bietet es sich an, einen Online-Arbeitsbereich, beispielsweise ein BSCW<sup>67</sup>, zu verwenden, da sich diese nicht auf Scratch-Online ablegen lässt. Dabei handelt es sich um einen geschlossenen Bereich im Internet, in dem angemeldete Benutzer ihre Arbeitsergebnisse abspeichern und austauschen können. Darüber hinaus ist es beispielsweise möglich,

---

<sup>65</sup> Bei einem längeren Andauern solchen Unterrichts ließen sich auch diese Phasen in die Schülerverantwortung übertragen. Es erscheint mir jedoch derzeit in dieser Lerngruppe noch nicht sinnvoll, diese Aufgabe den Schülerinnen und Schülern zu übertragen, weil die Moderationsfähigkeiten noch entwickelt werden müssen.

<sup>66</sup> Im Vorfeld stellte sich heraus, dass die Schülerinnen und Schüler beispielsweise häufig die Portale „SchülerVZ“ (<http://www.schuelervz.net/>, Zugriff am 28. März 2008), YouTube (<http://www.youtube.com/>, Zugriff am 28. März 2008) oder MySpace (<http://www.myspace.de/>, Zugriff am 28. März 2008) benutzen.

<sup>67</sup> Mehr Informationen unter <http://public.bscw.de/>, Zugriff am 28. März 2008. „BSCW“ steht für „Basic Support for Cooperative Work“ (Übersetzung A. S.: „Grundlegende Unterstützung für die Zusammenarbeit“) und wurde vom Fraunhofer-Institut für Angewandte Informationstechnik FIT entwickelt.

Online-Diskussionen zu führen, Abstimmungen durchzuführen oder Ergebnisse anderer Nutzer zu bewerten.

Die Schülerinnen und Schüler kennen die Arbeit mit dem BSCW bereits. Es ist in den vorangegangenen Unterrichtseinheiten unterschiedlich intensiv benutzt worden. Der Einsatz des BSCWs soll also gleichzeitig für Kontinuität und damit Verlässlichkeit in der Benutzung von Medien sorgen.

## 7 VERLAUFSKONZEPTION DER UNTERRICHTSREIHE

### 7.1 GROBVERLAUF

Wie bereits beschrieben wurde, drängte sich zum Ende der Unterrichtsreihe zur Programmiersprache „LOGO“ der Eindruck auf, dass viele der Schülerinnen und Schüler im bisherigen Wahlpflichtunterricht Informatik ein negatives Bild von Programmierung entwickelt hatten. Programmierung wurde als Lösung von irrelevanten und eng gefassten Aufgaben ohne persönlichen Bezug wahrgenommen. Auch führte die Programmierung mit „LOGO“ bei vielen Schülerinnen und Schülern zu der Ansicht, dass es im Wesentlichen darauf ankam, eine spezielle Lösung für ein spezielles Problem zu finden, ohne dass ihnen systematische Grundlagen vertraut wurden. Vor allem der Bereich der Programmierung mit Variablen ist hier zu nennen, dieses Konzept konnte der Mehrheit der Schülerinnen und Schüler nicht nähergebracht werden.

Um dieses Bild von Programmierung zu relativieren und möglicherweise eine veränderte Einstellung zu ihr anzustoßen, sollte in den folgenden Unterrichtsstunden Scratch zum Einsatz kommen.

Hierbei erschien es mir wichtig, den Schülerinnen und Schülern die Möglichkeit zu geben, ihre Einstellungen zur Programmierung von sich aus zu verändern, ihnen also mehr Verantwortung für ihren Lernprozess zu übergeben. Dies war insofern riskant, als dass die Gefahr bestand, dass die Schülerinnen und Schüler diese hinzugewonnene Verantwortung nicht in dem Umfang positiv für sich nutzen würden, wie es intendiert war. Um das Risiko zu minimieren, aber gleichzeitig die Chancen weitestgehend zu erhalten, also eine hohe und zielgerichtete Schüleraktivität zu erreichen, erschien es mir sinnvoll, den Unterricht nach kreativen Maßstäben zur Förderung des selbstständigen Arbeitens auszurichten. Dies ermöglicht eine hohe innere Bindung an den Unterrichtsgegenstand.

Gleichzeitig musste aber sichergestellt werden, dass informatische Konzepte zum Zuge kämen und zwar möglichst so, dass sie von den Schülerinnen und Schülern als notwendig und sinnvoll verinnerlicht werden würden. Reflektierende Elemente als Möglichkeit der Nachsteuerung mussten also eingeplant werden.

Im Wesentlichen sollte die Unterrichtsreihe deshalb in drei Grobphasen verlaufen:<sup>68</sup>

### ***Kennenlernen der Programmierumgebung (K)***

In dieser Phase setzen sich die Schülerinnen und Schüler mit der für sie neuen Programmiersprache auseinander. Es soll zunächst geklärt werden, was die Schülerinnen und Schüler von einer für den Wahlpflichtunterricht Informatik geeigneten Programmiersprache erwarten würden. Außerdem orientieren sich die Schülerinnen und Schüler in der für sie neuen Programmiersprachumgebung und versuchen, durch Veränderung einzelner Programmteile in einem mitgelieferten Beispielprojekt die Bedienungsweise der Umgebung zu verstehen. Es erfolgt die Vereinbarung der Dokumentationspflichten und die Anmeldung im BSCW und auf Scratch-Online. Abschließend soll - aus aktuellem Anlass - eine eigene Weihnachtsgeschichte als kleine Animation mit Scratch programmiert werden.

### ***Programmierung (P)***

Dieser Teil bildet den Schwerpunkt der Unterrichtsreihe und beruht auf dem schrittweisen Ausgestalten einer Animation hin zu einem kleineren Adventure-Spiel.

- Zuerst wird eine lineare Handlung als Kurzanimation mit Scratch umgesetzt.
- Anschließend werden mehrere Handlungsstränge eingefügt, sodass nichtlineare Handlungen<sup>69</sup> möglich werden. Das dazugehörige Konzept des „Regisseurs“ wird erarbeitet und mittels des informatischen Konzepts „Botschaften“ umgesetzt.
- Danach sollen gewisse Gegenstände („Items“) im Verlauf der Handlung von Bedeutung sein, d. h. es ist für den Verlauf der Geschichte wichtig, ob der „Held“ der Geschichte einen Gegenstand besitzt oder nicht. Hier kommt das informatische Konzept der „booleschen Variablen“ zum Einsatz.
- Letztlich werden dem „Helden“ mehrere Versuche zum Finden eines „richtigen“ Endes der Geschichte eingeräumt. Das zugehörige Konzept „mehrere Leben“ findet sich im informatischen Konzept „Ganzzahl-Variablen“ wieder.

### ***Reflexion (R)***

Zu Dokumentations- und Reflexionszwecken erstellen die Schülerinnen und Schüler eine maximal vierseitige Zusammenfassung aus ihren bisher erstellten Dokumentationen und erklären damit, worum es in ihrem Spiel geht. Wichtige Bestandteile können gesondert herausgestellt werden. Nach einer Abstimmung in der Lerngruppe wird das beste Plakat im Schaukasten ausgestellt. Abschließend wird zusammengefasst, ob (und wie weit) die Programmiersprache Scratch den anfangs geforderten Kriterien entspricht.

---

<sup>68</sup> In der konkreten Planung der einzelnen Unterrichtsstunden können die Phasen überlappen.

<sup>69</sup> Diese nichtlinearen Handlungsstränge ergeben zusammen eine mehrdimensionale Kurzgeschichte.

## 7.2 SYNOPSE

Stunde Phase	Thema der Stunde	Grobziel	Medien	Unterichtsformen	Hausaufgabe	Didaktischer Kommentar		
<b>1</b> K	Einführung in Scratch	Kennenlernen der Programmierumgebung	Computer, Internet, Beamer, Tafel	gel. UG EA, PA	Installation zu Hause, Anmeldung bei Scratch-Online	Motivation herstellen		
<b>2</b> K/P	Eine erste eigene Animation erstellen	Systematisches Untersuchen und Benutzen wichtiger Bausteine von Scratch		PA gel. UG	Dokumentation vervollständigen	Beschränkung auf wichtige Bausteine, um Funktionsweise zu analysieren		
<b>3</b> K	Die Scratch-Homepage und das BSCW	Vorbereitung der Sicherung von Ergebnissen		gel. UG PA	* keine (Weihnachtsferien)	Experimente zur Bereitstellung der Ergebnisse		
<b>4</b> K/P/R	Eine Weihnachtsanimation erstellen	Klang und Bilder als Möglichkeiten der individualisierten Darstellung nutzen		PA gel. UG		Sinnvolle Nutzung der letzten Stunde vor Weihnachten → Nutzung des kreativen Potentials		
<b>5/6</b> P	Eine mehrdimensionale Kurzgeschichte: Das Konzept des „Regisseurs“ mittels Botschaften	Verbindung vom informatischen Konzept „Botschaften“ und inhaltlicher Ausgestaltung		PA gel. UG	Dokumentation vervollständigen	Projektartige Zusammenarbeit mit längeren Phasen des selbstständigen Lernens	Konzeption	
<b>7/8</b> P/R	Eine mehrdimensionale Kurzgeschichte (Fortsetzung)	Auseinandersetzung mit programmier-technischer Ausgestaltung sowie Präsentation und Diskussion der Projekte		PA gel. UG	Projekte und Dokumentationen begutachten		Umsetzung und Vorstellung	

Stunde Phase	Thema der Stunde	Grobziel	Medien	Unterrichtsformen	Hausaufgabe	Didaktischer Kommentar
9/10 P	Die Realisierung von Items mittels boolescher Variablen	Verbindung vom informatischen Konzept „boolesche Variable“ und inhaltlicher sowie programmiertechnischer Ausgestaltung		<b>PA</b> gel. UG	Zeichnen der neuen Grafiken	Erweiterung der schon bestehenden Projekte → Erhöhung der Identifikation
11 P/R	Items (Fortsetzung)	Vervollständigung, Dokumentation und Optimierung der Programme		PA	* Projekte der anderen Gruppen begutachten	
12 P	Die Realisierung von „Leben“ mittels Ganzzahlvariablen	Verbindung vom informatischen Konzept „Ganzzahlvariable“ und inhaltlicher sowie programmiertechnischer Ausgestaltung sowie Präsentation und Diskussion der erweiterten Projekte		<b>PA</b> gel. UG		Ebenfalls Erweiterung, Vorstellung des gesamten Projekts inklusive Entwicklungsstufen
13 R	Eine Projektdokumentation für den Schaukasten	Eigenständige Reflexion und Darstellung des Gelernten		<b>PA</b> gel. UG		*
14 R	Scratch-Auswertung	Reflexion der Unterrichtsreihe		gel. UG	Rückblick und weitere Möglichkeiten erkunden	

Anmerkung: Dominante Unterrichtsformen wurden fett markiert. Die genannten Medien kamen in jeder Stunde zum Einsatz.

Legende: gel. UG - gelenktes Unterrichtsgespräch (auch Präsentationsphasen), EA - Einzelarbeit, PA - Partnerarbeit

\* Diese Stunden fanden hintereinander statt, hatten aber getrennte Inhalte.

## 8 DARSTELLUNG UND ANALYSE EINZELNER STUNDEN

Bei den im Folgenden ausgewählten Stunden liegt der Fokus darauf herauszustellen, inwieweit ein Unterricht nach kreativen Maßstäben stattgefunden hat. Ich werde aus jeder der drei vorgesehenen Grobphasen eine Stunde beziehungsweise Doppelstunde beschreiben. Dabei wird zunächst jede der Stunden in ihrem Verlauf dargestellt und eine Begründung der Unterrichtsphasen vorgenommen. Dem schließt sich eine Analyse der Unterrichtsstunden unter Berücksichtigung der Arbeitshypothesen an.

### 8.1 STUNDE 1: „EINFÜHRUNG IN SCRATCH“

#### 8.1.1 DARSTELLUNG UND BEGRÜNDUNG DER PHASEN

Diese Stunde sollte der Einführung in die Programmiersprache Scratch dienen.

Die vorhergehende Unterrichtsreihe zur Programmierung in „LOGO“ sollte sich klar von der folgenden Unterrichtsreihe abgrenzen. Mögliche Motivationshindernisse waren auszuräumen. Deshalb wollte ich zunächst die Schülerinnen und Schüler überlegen lassen, was ihrer Meinung nach eine gute Programmiersprache für den Wahlpflichtunterricht Informatik kennzeichnen würde. Dabei war die Frage absichtlich nicht präziser formuliert, um die Schülerinnen und Schüler nicht in eine vorgegebene Richtung, wie zum Beispiel „technische Aspekte“, „Erlernbarkeit“ usw., zu drängen. In einem gemeinsamen Brainstorming mit gleichzeitiger Visualisierung über den Beamer wurden die einzelnen Vorstellungen gesammelt (siehe Abbildung 2). Ich stimmte die Schülerinnen und Schüler dann darauf ein, dass es in den nächsten Stunden darum ginge zu überprüfen, ob die Programmiersprache Scratch ihre Anforderungen erfüllen würde.

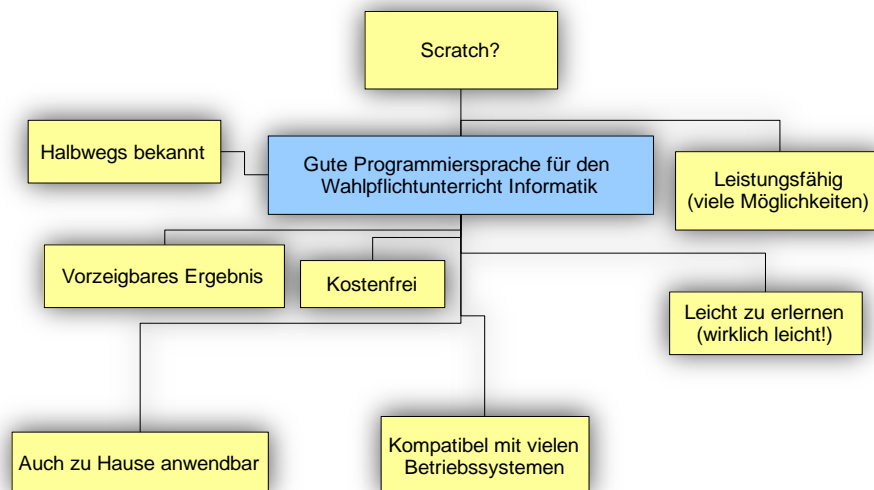


ABBILDUNG 2: BRAINSTORMING "GUTE PROGRAMMIERSPRACHE"

In der daran anschließenden Phase wurde ein kurzes Einführungsvideo zu Scratch gezeigt, um die Schülerinnen und Schüler auf die Inhalte der nächsten Unterrichtsstunden vorzubereiten.<sup>70</sup> Dieses Video hat ein überraschendes Ende: Es stellt sich heraus, dass es (nahezu) ausschließlich mit Scratch produziert wurde. Damit sollte eine positive Einstellung zur beginnenden Arbeit mit Scratch vorbereitet werden. Die Schülerinnen und Schüler äußerten dann im Plenum ihre Vermutungen, wozu Scratch benutzt werden könnte (siehe Abbildung 3). Ohne falsche Versprechungen machen zu müssen, konnte den Schülerinnen und Schülern mitgeteilt werden, dass alle ihre Vermutungen in den kommenden Stunden praktisch realisiert werden würden.

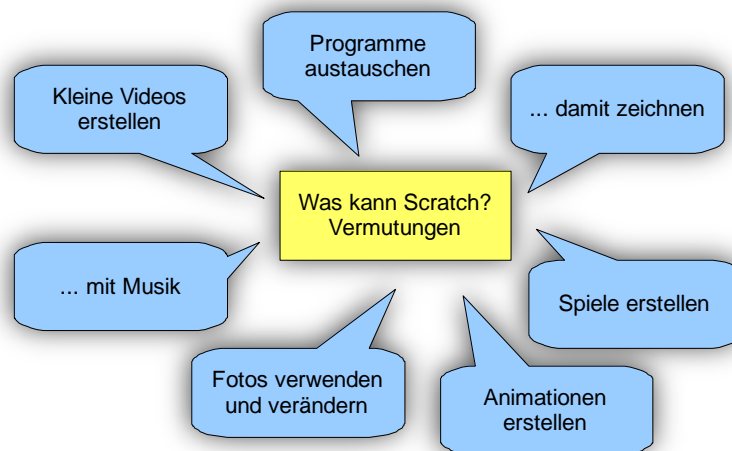


ABBILDUNG 3: VERMUTUNGEN ZU SCRATCH

Die im Informatikraum vorhandenen Rechner installieren bei jedem Neustart automatisch ein vordefiniertes Musterabbild, bei dem Scratch nicht installiert ist.<sup>71</sup> Deshalb musste nun geklärt werden, wie Scratch im Rechnernetz bezogen und installiert werden kann, was parallel von den Schülerinnen und Schülern an den Rechnern nachvollzogen wurde. Dies erfolgte explizit nicht in Partnerarbeit: Ziel war es, sicherzustellen, dass auch im Falle der Abwesenheit eines der Partner der andere gegebenenfalls am jeweiligen Projekt weiterarbeiten konnte. Anschließend wurde im Plenum besprochen, wie man

<sup>70</sup> Bei dem Video handelt es sich um ein Einführungsvideo, das direkt von Scratch-Online heruntergeladen werden kann (<http://ilk.media.mit.edu/projects/scratch/videos/ScratchIntro.wmv>, Zugriff am 29. März 2008). In ihm wird die Programmiersprache und die Programmierung mit Scratch in groben Zügen vorgestellt, wobei es (aus deutschsprachiger Sicht) problematisch ist, dass diese Einführung in englischer Sprache erfolgt. Dies stellte jedoch für die Schülerinnen und Schüler kein größeres Problem dar, weil die Audio-Informationen in dem Video lediglich unterstützenden Charakter haben und der Video-Teil dominierend ist.

<sup>71</sup> Dieser Vorgang stellt sicher, dass eventuelle Neuinstallationen von kritischen Programmen während der Rechnerlaufzeit durch Schülerinnen und Schüler beziehungsweise Lehrerinnen und Lehrer die Systemkonsistenz nicht ernsthaft gefährden können. Die Installation von Scratch ist nicht sehr zeitaufwendig.

sich Scratch zu Hause installieren kann und wie die Anmeldung bei Scratch-Online funktioniert.

In den verbleibenden ca. 20 Minuten hatten die Schülerinnen und Schüler die Aufgabe, eines der mitgelieferten Scratch-Beispiele in Partnergruppen zu öffnen und sich mittels kleinerer Veränderungen im Quelltext der Programmierung mit Scratch zu nähern. Durch diese experimentellen Veränderungen eines bestehenden Projektes sollten sich die Schülerinnen und Schüler mit der Programmieroberfläche vertraut machen. Außerdem konnte dadurch eine Brücke von den im Video geweckten Erwartungen hin zu den „echten“ Programmiererfahrungen der folgenden Stunden geschlagen werden.

Zum Schluss konnten einige der Gruppen (auf freiwilliger Basis) ihre Projekte kurz vorstellen, wobei sie vor allem demonstrieren sollten, was sie verändert hatten und welche Auswirkungen dies hatte. Es diente zum einen der Würdigung der Ergebnisse der Schülerinnen und Schüler und sollte zum anderen schon darauf vorbereiten, dass auch zukünftig erzielte Resultate von den Schülerinnen und Schülern reflektiert und dargestellt werden sollten.

Die Schülerinnen und Schüler erhielten die Hausaufgabe, Scratch auf ihrem Rechner zu Hause zu installieren und sich bei Scratch-Online anzumelden.

#### 8.1.2 ANALYSE

##### ***Motivation***

Das Vorhaben, mit den beiden Brainstorming-Phasen und dem Einführungsvideo eine Verbundenheit mit der Programmiersprache zu bewirken, scheint gelungen zu sein. Vor allem in der Phase der experimentellen Veränderung des Quelltextes mussten die Schülerinnen und Schüler nicht besonders motiviert werden, die Aufgabenstellung zu erfüllen. Im Gegenteil, sie setzten sich unverzüglich und konzentriert mit den Beispielen auseinander und versuchten, jedoch noch relativ ziellos, Veränderungen im Programmablauf herbeizuführen.

Besonders hervorheben möchte ich den Ausspruch des Schülers S13, der mich während dieser Phase ansprach und meinte: „Endlich muss man nicht mehr schreiben und kann keine Fehler mehr machen.“ Ich interpretiere dies so, dass S13 die bisherigen Versuche der Programmierung in den anderen Programmiersprachen des Wahlpflichtunterrichts als sehr schreibintensiv und Fehler provozierend wahrgenommen hatte.<sup>72</sup> Offenbar war es in dieser Stunde gelungen, die Schülerinnen und Schüler hinreichend zu motivieren, sodass sie sich in den folgenden Stunden freiwillig mit den Unterrichtsgegenständen auseinandersetzen würden.

---

<sup>72</sup> Seine Einschätzung ist inhaltlich leider nicht ganz treffend, da bei Scratch lediglich Syntaxfehler (nahezu) ausgeschlossen sind, inhaltliche Fehler jedoch naturgemäß trotzdem auftreten können.

### ***Kreativität***

Auch wenn der Aspekt kreativen Arbeitens in dieser Stunde noch keinen Schwerpunkt bildete, wurde trotzdem, vor allem in der Präsentationsphase, deutlich ersichtlich, dass die Schülerinnen und Schüler die ihnen durch die Aufgabenstellung zur Verfügung stehenden Möglichkeiten der unterschiedlichen Bearbeitung nutzten: Dadurch, dass nicht vorgegeben war, welches Beispiel näher betrachtet werden sollte, wurden verschiedene Beispiele ausgewählt und - bei gleich gewählten Beispielen - mit völlig verschiedenen Ansätzen verändert.<sup>73</sup>

### ***Progression***

Hier lässt sich, da es sich um den Beginn der Unterrichtseinheit handelt, am ehesten eine Ausgangsbeobachtung beschreiben. Es fiel auf, dass die Schülerinnen und Schüler anfangs Schwierigkeiten im Umgang mit unscharfen Zielerwartungen hatten. Den Schüleräußerungen ließ sich entnehmen, dass sie offenbar eine stärkere Lenkung hinsichtlich der Wahl und der Bearbeitung des Beispiels erwarteten. Dieses irritierende Moment spielte jedoch spätestens mit Beginn der Bearbeitung der Beispiele keine Rolle mehr.

### ***Zeit***

Der zeitliche Rahmen spielte vor allem beim Experimentieren mit dem Quelltext eine Rolle, weil er vorgab, wie intensiv sich mit den Beispielen auseinandergesetzt werden konnte. Einige der Schülerinnen und Schüler hätten vermutlich noch ausführlicher an der Manipulation der Programme gearbeitet, wenn das möglich gewesen wäre. Dies zeigte sich daran, dass einige Schüler zum Ende der Stunde noch im Raum blieben und ihr Beispielprojekt weiterbearbeiteten. Aus meiner Sicht war ein Abbruch an dieser Stelle jedoch gerechtfertigt, da die Schülerinnen und Schüler genügend Einblick gewinnen konnten.

### ***Lehrerrolle***

Der Anforderung, vornehmlich beratend tätig zu sein, wurde ich in den ersten Teilen dieser Stunde nicht gerecht, was mir aber bedingt durch das Thema vertretbar erscheint. In den letzten Teilen der Stunde gelang es mir schon eher mich zurückzuhalten, allerdings forderten die Schülerinnen und Schüler mehrfach eine stärkere Vorgabe der Arbeitsabläufe ein. Im Sinne der Progression sollte dies in den folgenden Stunden abnehmen.

---

<sup>73</sup> Einige Gruppen konzentrierten sich mehr auf die Manipulation des Ablaufs der Programme, während andere eher das Erscheinungsbild in Form und Farbe veränderten. Beides ist im Sinne der Aufgabe zulässig gewesen, da beides zu einem Grundverständnis der Programmierung in dieser Sprache beiträgt.

## **Präsentation**

In der Präsentationsphase zeigten sich noch Unsicherheiten in den einzelnen Gruppen. Die Ergebnisse wurden wie erwartet eher widerwillig präsentiert,<sup>74</sup> dann aber mit großem Nachdruck gegen Kritik von anderen verteidigt. Hier fiel auf, dass einige Schüler, vor allem S14, offenbar vor allem eine Wertschätzung des eigenen Produkts erreichen wollten und entsprechend negativ über andere Produkte sprachen. Diese Ansätze wurden jedoch überzeugend von den Mitschülerinnen und -schülern korrigiert.

## **Dokumentation**

In dieser Stunde fand noch keine weitere Dokumentation der Ergebnisse statt. Dies hing vor allem damit zusammen, dass die Beispiele noch nicht zielgerichtet untersucht wurden und erst einmal ein erster Eindruck von Scratch gewonnen werden sollte. Die Alternative, Scratch an sich einführend dokumentieren zu lassen, erschien mir nicht angemessen, da dies auf eine Dokumentation der Programmieroberfläche hinausgelaufen wäre. Ein erheblicher Mehrwert in dieser Dokumentation erschloss sich mir nicht.

## **8.2 STUNDE 5/6: „EINE MEHRDIMENSIONALE KURZGESCHICHTE“**

### **8.2.1 DARSTELLUNG UND BEGRÜNDUNG DER PHASEN**

In dieser und der folgenden Doppelstunde haben die Schülerinnen und Schüler an einem zusammenhängenden Projekt gearbeitet.

In den vorangegangenen Stunden wurden Kurzanimationen erstellt, zunächst mit dem Schwerpunkt der systematischen Erkundung einiger essentieller Bausteine in Scratch (Stunde 2: „Eine erste eigene Animation erstellen“), anschließend unter dem Gesichtspunkt des individualisierten Einsatzes von Grafiken und Klängen (Stunde 4: „Eine Weihnachtsanimation erstellen“). Jetzt, da den Schülerinnen und Schülern wichtige Grundlagen zur Arbeit mit Scratch bekannt waren, konnte ein komplexeres Projekt in Angriff genommen werden: die „mehrdimensionale Kurzgeschichte“.

Ausgangspunkt war dabei die Frage, ob die Schülerinnen und Schüler Geschichten kannten, die nicht nur ein bestimmtes Ende besäßen. Erwartungsgemäß kamen von der Lerngruppe vornehmlich Bezüge zu Comics aus der Kinder- und Jugendzeit, teilweise aber auch zu anderen Büchern. S14 stellte kurz vor, wie solche Geschichten realisiert werden. Die Ausgangsfrage diente der motivationalen Anknüpfung an positive Erlebnisse der Schülerinnen und Schüler in der Vergangenheit und der inhaltlichen Verbindung zu bereits vorhandenem Vorwissen über die Struktur des zu verwendenden Konzepts.

Für die folgenden ca. 30 Minuten erhielten die Partnergruppen die Aufgabe, sich eine solche Geschichte mit mehreren Ausgängen zu überlegen und wesentliche Verlaufs-

---

<sup>74</sup> Vgl. Kapitel 3.2.2

punkte (z. B. Handlungsstränge) auf einem Konzeptpapier stichpunktartig zu beschreiben und anschließend die benötigten Grafiken (Charaktere und Hintergründe) zu erstellen oder aus dem Internet herunterzuladen.<sup>75</sup>

Zu Beginn der zweiten Stunde wurde gemeinsam besprochen, wie man grundsätzlich eine mehrdimensionale Geschichte programmieren könnte. Hierzu erhielten die Schülerinnen und Schülern eine einfache Umsetzung des Konzepts des „Regisseurs“ (ohne inhaltlichen Bezug zu einer Geschichte) vorgegeben und sollten es erklären. Es wurden dann die Vor- und Nachteile eines Einsatzes dieses Konzeptes besprochen.<sup>76</sup>

Bis zum Ende der Unterrichtsstunde hatten die Schülerinnen und Schüler nun Zeit, die von ihnen erdachte Geschichte unter Zuhilfenahme des Konzepts des „Regisseurs“ umzusetzen. Hierbei sollten sie sich noch einmal intensiv mit diesem Konzept auseinandersetzen und somit das Zusammenspiel der Botschaften-Bausteine als dessen Basis verstehen.

Als Hausaufgabe sollten die Dokumentation zum bisher Erreichten erstellt werden, um eine Grundlage für die folgende Doppelstunde mit der geplanten Fertigstellung der Projekte und den dazugehörigen Präsentationen der Projekte zu haben.

#### 8.2.2 ANALYSE

##### **Motivation**

In dieser Doppelstunde gelang es, die Schülerinnen und Schüler zur selbstständigen Auseinandersetzung mit dem Unterrichtsgegenstand zu motivieren. Der umzusetzende Sachverhalt, die mehrdimensionale Kurzgeschichte, war den Mitgliedern der Lerngruppe größtenteils inhaltlich bekannt, allerdings wurden solche Abläufe von ihnen bisher noch nicht realisiert. Dies schien Interesse bei ihnen hervorzurufen. Die Möglichkeit, sich den Inhalt der Kurzgeschichte selbst auszudenken und sie audiovisuell selbst zu gestalten, sorgte offenbar für einen zusätzlichen Motivationsschub. Größere Ablenkungen

---

<sup>75</sup> Hier musste, als willkürlich Grafiken aus dem Internet geladen wurden, kurz in Form einer Plenumsphase zum Thema „Urheberrecht“ beziehungsweise „Recht am eigenen Bild“ eingegriffen werden. Die Schülerinnen und Schüler wurden daran erinnert, dass die Ergebnisse ihrer Arbeit öffentlich verfügbar sein würden und insofern ausschließlich Bilder (und Klänge) verwendet werden dürften, die für diesen Zweck freigegeben wurden.

<sup>76</sup> In Abwägung der Möglichkeiten der Erarbeitung dieses Konzeptes kam ich zu dem Schluss, dass es selbst in diesem großzügigen zeitlichen Rahmen nahezu unmöglich sein würde, es die Schülerinnen und Schüler selbst erarbeiten zu lassen. Dies hätte bedeutet, dass sie erst mit einer einfacheren Ablaufsteuerung versucht hätten, das Ziel zu erreichen, diese dann schrittweise verfeinert oder verändert hätten, um letztendlich festzustellen, dass lediglich ein neuer Ansatz weiterhelfen würde. Dieses Vorgehen hätte außer dem Zeitaufwand auch noch den Nachteil gehabt, dass die Schülerinnen und Schüler möglicherweise ihr Selbstvertrauen in die Fähigkeit, Programme in Scratch komplett gestalten zu können, verloren hätten. Ungeachtet dieser fehlenden Möglichkeit des „Selber-Entdeckens“ hatten die Schülerinnen und Schüler meiner Meinung nach trotzdem ausreichend Raum für eine tiefere Auseinandersetzung mit diesem Konzept, da sie es analysieren, bewerten und außerdem in ihr geplantes Projekt integrieren sollten.

schiene nicht aufzutreten, mehrheitlich arbeiteten die Schülerinnen und Schüler konzentriert und motiviert an ihrem Projekt.

### ***Kreativität***

Im kreativen Bereich zeigte sich, dass die Lerngruppe offenbar die ihnen zugestandenen Freiheiten sinnvoll zu nutzen wusste. Die Konzepte für die Kurzgeschichte reichten vom Versteckspiel zweier Freunde über den Schulweg eines Schülers bis hin zu einer Geschichte, in der ein Löwe versucht, die Weltherrschaft zu erringen, während ein Papagei versucht, ihn davon abzubringen.

Auffällig viel Zeit verbrachten die einzelnen Gruppen mit der Ausgestaltung ihrer Charaktere, was auf eine hohe Verbundenheit zum jeweiligen Projekt schließen lässt. Es zeigte sich, dass dabei unterschiedliche Wege beschritten wurden: Einige Gruppen zeichneten ihre Charaktere komplett selbst, andere benutzten Vorlagen aus dem Internet und wieder andere modifizierten Beispielgrafiken von Scratch, bis sie ein individuelles Aussehen besaßen.

### ***Progression***

Inzwischen zeigte sich, dass die Lerngruppe für ihre selbstständigen Arbeitsphasen darauf vertraute, lediglich unscharfe Zielerwartungen zu erhalten („Erstelle eine mehrdimensionale Kurzgeschichte.“) und den Weg dorthin selbst gestalten zu dürfen, sobald die fachlichen Hürden - das neue Konzept des „Regisseurs“ - dem nicht mehr im Wege standen. Die Rückfragen hatten lediglich noch fachlichen Bezug, etwa wenn sich ein kniffliges Problem absolut nicht lösen ließ. Leichtere Hürden dagegen (etwa ein ungewollter Programmablauf oder ungewollte Effekte) versuchten die Schülerinnen und Schüler weitestgehend innerhalb der Gruppe zu lösen.

Auf der anderen Seite wurde von den Gruppen verstärkt die Würdigung besonders gelungener Charaktere oder Story-Elemente eingefordert, sowohl von den Mitschülerinnen und Mitschülern als auch vom Lehrer.

### ***Zeit***

Durch den projektartigen Charakter erschien mir in diesen beiden Doppelstunden der zeitliche Rahmen angemessen, zumal eine Weiterarbeit zwischen den beiden Doppelstunden zu Hause zusätzlich möglich war (was allerdings nur von einem Teil der Schülerinnen und Schüler genutzt wurde).

Aus Sicht der Lerngruppe wäre es - wenn die entsprechenden Wortäußerungen so interpretiert werden - wahrscheinlich wünschenswert gewesen, noch etwas mehr Zeit zur Erstellung der Grafiken aufwenden zu können. Dem steht jedoch entgegen, dass in diesem Teil der Unterrichtsstunde der informatische Bezug am schwächsten ausgeprägt war.

Im Nachhinein kann man feststellen, dass der zeitliche Rahmen angemessen gewählt wurde, da die Projekte innerhalb der Projektzeit fertiggestellt wurden.

### **Lehrerrolle**

Wiederum war im Einführungsteil der ersten Stunde der steuernde Anteil meinerseits sehr hoch, dafür war dieser Teil aber zeitlich eng eingegrenzt. Im restlichen Verlauf der ersten Stunde konnte ich im Wesentlichen der angedachten Rolle gerecht werden, hier wurden von mir lediglich Impulse bezüglich der Ausgestaltung der einzelnen Charaktere eingefordert. Selbst in der außerplanmäßigen Phase zu den Fragen des Urheberrechts, in der - da sie im Plenum stattfand - ein höherer Anteil von Lehrersteuerung erfolgte, führten die Schülerinnen und Schüler die Diskussion ohne größere Einwirkung meinerseits durch und kamen argumentativ zu vertretbaren Schlüssen bezüglich der Verwendbarkeit nicht selbsterstellter Grafiken.

Im ersten Teil der zweiten Stunde war - trotz der Einführung des neuen Konzepts - mein Steuerungsanteil begrenzt. Die Auseinandersetzung mit der vorgegebenen Grundstruktur nahmen die Schülerinnen und Schüler in ihren Gruppen vor und in der anschließenden Diskussion der Vor- und Nachteile setzten sie das Diskussionsmuster vom Ende der vorhergehenden Stunde fort. Im darauf folgenden Teil dieser Stunde - der Phase der Umsetzung des Konzepts für die eigene Geschichte - konnte ich mich erneut als Berater betätigen.

Hier wurde meine Aufmerksamkeit vor allem von einer Partner-Gruppe in Beschlag genommen: Während alle anderen Gruppen das Konzept des „Regisseurs“ verwendeten, nutzte die Gruppe von S16 und S14 ein alternatives Konzept, welches den Anforderungen im weiteren Verlauf nicht angemessen war.<sup>77</sup> Durch ihre Erläuterungen zu ihrem Ansatz ergab sich der Eindruck, dass sie das eigentlich angedachte Konzept noch nicht vollständig verstanden hatten. Es war nun möglich, im Gespräch mit dieser Gruppe das Problem zu thematisieren. Dadurch wurde ihnen der Anschluss an den allgemeinen Stand ermöglicht, wozu bei anders angelegtem Unterricht eine Plenumsphase benötigt worden wäre.

### **Präsentation und Dokumentation**

In dieser Doppelstunde waren keine Phasen für die Präsentation und Dokumentation vorgesehen, da das Projekt in der kommenden Doppelstunde fortgesetzt werden sollte und dort entsprechende Zeit dafür reserviert war. Allerdings machten sich die vorangegangenen Stunden mit den in ihnen enthaltenen Präsentationsphasen in dieser Doppelstunde bemerkbar; in den Phasen der Diskussion um die Konzepte beziehungsweise

---

<sup>77</sup> Sie benutzten statt der Botschaften, die im Konzept des „Regisseurs“ von Bedeutung waren, einen anderen Scratch-Baustein: „*Wenn Taste <TASTE> gedrückt*“. Dieser führt jedoch bei mehrdimensionalen Geschichten zum Konflikt, weil ab diesem Zeitpunkt die gleiche Taste so nicht erneut verwendet werden kann.

Einstellungen wurden die Beiträge der einzelnen Personen von den Schülerinnen und Schülern respektiert und umgekehrt wurden von den Redenden vermehrt Gründe für ihre Sichtweisen genannt. Dies lässt sich meiner Meinung nach auf die in den Präsentationsphasen entwickelten Umgangsformen zurückführen.

Eigentlich war es für das Ende der Doppelstunde von mir vorgesehen gewesen, bereits Dokumentationen für das bisher Erreichte anfangen zu lassen. Diesen Plan änderte ich jedoch, als sich abzeichnete, dass die einzelnen Gruppen mit dem weiteren Bearbeiten ihrer Projekte stark beschäftigt waren und offenbar kein günstiger Zeitpunkt für eine Unterbrechung vorlag. Entsprechend wurde dies zur Hausaufgabe.

### 8.3 STUNDE 13: „PROJEKTDOKUMENTATION FÜR DEN SCHAUkastEN“

#### 8.3.1 DARSTELLUNG UND BEGRÜNDUNG DER PHASEN

In dieser Stunde ging es darum, eine abschließende Projektdokumentation zu erstellen.

Die Stunden, die dieser vorausgingen, dienten der Vervollständigung und Erweiterung der mehrdimensionalen Kurzgeschichte hin zu einem Adventure-Spiel. Parallel wurden dazu - in unterschiedlicher Qualität - Dokumentationen erstellt, in denen neu hinzugewonnenes Wissen jeweils festgehalten werden sollte. Diese Dokumentationen sollten nun auf einer neuen Ebene zu einer Projektbeschreibung verschmelzen.

Dabei wurde das Anliegen durch einen Seitenaspekt sinnvoll unterstützt: Die Informatik hat im Foyer der Schule einen Schaukasten, der seit längerer Zeit nicht aktualisiert worden war. Es erschien mir für die Außenwirkung des Fachbereiches Informatik nützlich, die zu erstellenden Dokumentationen im Einverständnis mit den Schülerinnen und Schülern zum Aushang an dieser Stelle zu benutzen. Problematisch war, dass im Schaukasten lediglich Platz für ein bis maximal zwei Dokumentationen blieb. Es würde also eine Auswahl von Dokumentationen erfolgen müssen.

Mit Hinweis auf den geplanten Aushang und mit Verweis auf die bereits vorhandenen und als Grundlage verwendbaren Dokumentationen erhielten die Gruppen nun den Auftrag, eine Projektbeschreibung zu verfassen. Dabei nannte ich als Vorgaben lediglich, dass das Adventure-Spiel anschaulich dargestellt werden sollte (inklusive der Erklärung wichtiger Punkte) und dass wegen des Schaukastenaushangs aus Platzgründen dafür maximal vier DIN-A4-Seiten pro Projekt zur Verfügung stünden. Bezüglich der Wahl oder Ausgestaltung der Inhalte und der für die Erstellung der Beschreibung zu benutzenden Programme gab ich nichts Weiteres vor, um die Schwerpunktsetzung und den Arbeitsprozess der Schülerinnen und Schüler nicht unnötig zu beeinflussen.

Die Erstellungsphase sollte in Eigenregie der Gruppen organisiert werden, dafür waren etwa 25 Minuten eingeplant. Weitere 5 Minuten sollten für die Überarbeitung und Durchsicht in Bezug auf Fehler in Rechtschreibung oder Grammatik und optische Optimierung genutzt werden. Da die Schülerinnen und Schüler zu diesem Zeitpunkt bereits um den Wettbewerbscharakter wussten, nutzten sie diese Phasen sehr intensiv.

In der Präsentationsphase stellten die Schülerinnen und Schüler ihr Projekt inklusive der Dokumentation über den Beamer kurz vor, sodass sich alle Mitglieder der Lerngruppe einen Eindruck über die verschiedenen Ergebnisse verschaffen konnten. Die anschließende Abstimmung war so geplant, dass gruppenweise ein Favorit benannt wurde, wobei sich die Gruppen nicht selbst wählen durften. So sollte vermieden werden, dass die Abstimmung ohne Ergebnis bleiben würde. Tatsächlich gab es zwei Siegerdokumentationen, die daraufhin sofort im Schaukasten angebracht wurden, was die besondere Wertschätzung der Lerngruppe ausdrücken sollte (siehe Anhang).

### 8.3.2 ANALYSE

#### **Motivation**

Der Bereich der Dokumentation, der wie beschrieben Teil der Projekte war, gestaltete sich im bisherigen Unterrichtsverlauf für die Lerngruppe offenbar als der unangenehmste. Dies äußerte sich zum einen im Protest gegen die Erstellung<sup>78</sup> und zum anderen in der Qualität der Dokumentationen<sup>79</sup>. Obwohl die Schülerinnen und Schüler die Notwendigkeit einer Dokumentation einsahen, lehnten sie den Mehraufwand, der zur Erstellung nötig war, ab. Insofern erschien mir eine starke Motivation notwendig, um die Gruppen dazu zu bewegen, eine gute Abschlussdokumentation zu erstellen.

Die Aussicht auf den Aushang der Dokumentation im Schaukasten schien genau diese Motivation zu liefern. Die Schülerinnen und Schüler arbeiteten intensiv und relativ gewissenhaft sowohl an der Erstellung und Gestaltung der Dokumentationen als auch in der anschließenden Korrekturphase. Auch an der folgenden Präsentation und Abstimmung beteiligten sie sich engagiert, sodass hier ein hohes Maß an Motivation bescheinigt werden kann.

#### **Kreativität**

Außer den Punkten „maximal vier Seiten in DIN-A4“ und „aussagekräftige Projektbeschreibung“ wurde der Lerngruppe nicht vorgegeben, wie die Dokumentation der Projekte aussehen sollte. Entsprechend wählten die Schülerinnen und Schüler unterschiedliche Ansätze und Werkzeuge: Während einige die Projekte mit einer Präsentationssoftware erstellten, nutzten andere ein Textverarbeitungsprogramm. Einige Gruppen beschrieben ausschließlich den Verlauf ihres Adventure-Spiels, andere konzentrierten sich mehr auf die Vorstellung der im Spiel eingebauten Konzepte und wieder andere beschrieben ihr Adventure-Spiel im Kontext der Erstellung mit Scratch.

---

<sup>78</sup> Dies zeigte sich durch „Murren“ und Äußerungen wie „Müssen wir das wirklich machen?“ Die Unwilligkeit schien die gesamte Lerngruppe zu betreffen.

<sup>79</sup> Hier gab es allerdings starke Unterschiede: Während beispielsweise S3, S10, S11, S1, S4 und S5 meistens gute Ergebnisse lieferten, waren die Dokumentationen unter anderem von S13, S14 und S16 häufig von geringer Qualität.

Ich hätte mir gewünscht, dass wenigstens eine der Gruppen die Idee der Dokumentation durch Programmierung in Scratch umgesetzt hätte, auch dies wäre ausdrückbar gewesen und hätte eine Ausstellung im Schaukasten ermöglicht. Umgekehrt wäre es aber mit den gewählten Grundlagen dieser Unterrichtsreihe nicht redlich gewesen, die Schülerinnen und Schüler in diese Richtung zu drängen.<sup>80</sup>

### ***Progression***

In dieser Stunde zeigte sich deutlich die Progression im Bereich des selbstständigen Arbeitens. Alle Schülerinnen und Schüler waren in der Lage, mit den unscharfen Zielvorgaben angemessen umzugehen und die ihnen zur Verfügung stehenden Freiräume kreativ auszugestalten. Die Resultate und deren Präsentation sowie das Abstimmungsverhalten sprechen für eine hohe Identifikation der Gruppen mit ihren Projekten. Trotz geringer Lehrersteuerung in den einzelnen Phasen, vor allem in der Erstellungs- und der Präsentationsphase, ließen sich die Schülerinnen und Schüler nicht stark ablenken und erarbeiteten sich konzentriert ihre Ergebnisse.

### ***Zeit***

Der zeitliche Rahmen war auch in diesem Fall eher knapp gesetzt. Die Begrenzung auf vier DIN-A4-Seiten ließ den Zeitmangel für die Lerngruppe nicht so offensichtlich werden<sup>81</sup>, allerdings wäre mit mehr zur Verfügung stehender Zeit sicherlich eine detailliertere Projektbeschreibung möglich gewesen. Es stellt sich jedoch die Frage, ob dann die Motivation weiterhin aufrecht zu erhalten gewesen wäre.

### ***Lehrerrolle***

Bis auf den Abstimmungsprozess, den ich der Lerngruppe nicht in Eigenregie überließ, habe ich mich weitestgehend auf die Beraterrolle zurückgezogen. Auffällig war, dass in den Beratungsgesprächen keine kreativen Impulse nötig waren, sondern vor allem Fragen der Rechtschreibung und Grammatik thematisiert wurden beziehungsweise thematisiert werden mussten. Aufgrund von Schüleräußerungen ergab sich das Bild, dass ihnen offenbar ein Großteil der Fehler nicht bewusst war, sie aber keinesfalls eine fehlerhafte Dokumentation ausstellen wollten. Das führte dazu, dass von den Gruppen häufig mein Rat gesucht wurde.

### ***Präsentation***

Der Ablauf der Präsentation der Ergebnisse verlief inzwischen beinahe schon routiniert. Die Gruppen stellten nacheinander und annähernd selbstorganisiert ihre Ergebnisse vor

---

<sup>80</sup> Somit muss ich mir diese Hoffnung für die nächste Lerngruppe aufheben.

<sup>81</sup> Es gab nur vereinzelt die Bitte um einen kurzen Zeitaufschub.

und begründeten ihre Einzelentscheidungen zu bestimmten Gestaltungspunkten. Die Zuhörer verhielten sich respektvoll und kritisierten überwiegend konstruktiv.<sup>82</sup>

### **Dokumentation**

Die Dokumentationen erfüllten die formalen Anforderungen und zeigten, dass sich die Schülerinnen und Schüler mit ihrer Arbeit identifizierten. Mit den Verweisen auf die im Projekt enthaltenen Grundkonzepte zeigte zumindest ein Teil der Lerngruppe, dass diese Konzepte verinnerlicht wurden und anwendungsbereit waren.

## **9 GESAMTREFLEXION**

### **9.1 PROJEKTE UND DOKUMENTATIONEN**

Im Verlauf der Unterrichtsreihe sind pro Gruppe mehrere Projekte mit dazugehörigen Dokumentationen entstanden.<sup>83</sup> Diese stellen - als Gesamtheit - die Produkte der Unterrichtsreihe dar. Unter Einbeziehung der Arbeitshypothesen lässt sich an diesen Produkten überprüfen, ob und in welchem Umfang selbstständiges Arbeiten stattgefunden hat. Dabei ist zu beachten, dass die Produkte in der hier vorzustellenden Form den Endzustand des Arbeitsprozesses anzeigen. Der Schwerpunkt liegt dabei auf den Hypothesen „Kreativität“, „Progression“, „Zeit“ und „Dokumentation“.

Die entstandenen Produkte zeugen von einem hohen Maß an Kreativität im Arbeitsprozess (siehe Anhang). Die einzelnen Gruppen wählten sich unterschiedliche thematische Zusammenhänge, die dargestellt werden sollten. Sie verwendeten unterschiedliche Zugänge zur Beschaffung der grafischen Inhalte und waren in der programmiertechnischen Ausgestaltung, also der Wahl der Algorithmen, nicht festgelegt. Allerdings gab es Einschränkungen in der kreativen Umsetzung: Durch die zu vermittelnden informatischen Grundkonzepte wurden bestimmte Alternativzugänge ausgeschlossen, die ebenfalls für eine Umsetzung der gewählten inhaltlichen Zusammenhänge in Frage gekommen wären.

An den entstandenen Produkten lässt sich nicht bestimmen, wie sich der Verlauf der Fortschritts im selbstständigen Arbeiten gestaltete.<sup>84</sup> Dafür lässt sich aber herausstellen, ob und wie stark selbstständig gearbeitet wurde, also der Endpunkt der Progression festhalten. Daran gemessen, dass alle Arbeitsgruppen ein funktionierendes Ergebnis vorzeigen konnten, welches die behandelten Konzepte als Umsetzung enthielt, lässt sich

---

<sup>82</sup> Auch S14, der zu Beginn der Reihe durch destruktive Kritik aufgefallen war, zeigte ein angemessenes Kritikverhalten. Ob dies jedoch auf die Unterrichtsreihe oder auf die Tagesform zurückzuführen ist, lässt sich nur schwer einschätzen.

<sup>83</sup> Die Projekte lassen sich auf Scratch-Online einsehen. Ausgangspunkt könnte beispielsweise die Seite <http://scratch.mit.edu/users/asrpdm/> (Zugriff am 2. April 2008) sein, von der aus u. a. auf die Seiten der Schülerinnen und Schüler verwiesen wird.

<sup>84</sup> Das geht aus der Analyse der vorgestellten Unterrichtsstunden hervor.

der Schluss ziehen, dass eine Progression stattgefunden hat. Begründet mit der unscharfen Zielerwartung und dem damit selbst zu findenden Weg zum Ergebnis ergibt sich eine Förderung zur Fähigkeit im selbstständigen Arbeiten.

Eine Förderung der Fähigkeit zum selbstständigen Arbeiten lässt sich im Hinblick auf die Zeit-Hypothese feststellen, wenn das Resultat den gestellten Anforderungen entspricht und es im Rahmen der zur Verfügung stehenden Zeit fertiggestellt werden konnte. Hier widerspricht der Eindruck aus den Resultaten den Darstellungen in der Analyse der vorgestellten Stunden. Offenbar war die Zeitvorgabe durchaus ausreichend, jedenfalls wurden alle Produkte in einem den Anforderungen entsprechenden und funktionierenden Zustand vorgelegt. Diese gegensätzlichen Feststellungen lassen sich verbinden, wenn die „Perfektion“ der Ergebnisse zusätzlich betrachtet wird. Hier hätte es bei zusätzlich zur Verfügung stehender Zeit noch Spielraum zur Optimierung gegeben, was eine tiefere Auseinandersetzung mit den Unterrichtsgegenständen und den behandelten Konzepten bedeutet hätte.<sup>85</sup>

Für die Betrachtung des Beitrags der Dokumentation zur Förderung der Fähigkeit zum selbstständigen Lernen ist eine Unterscheidung notwendig. Die Dokumentation der Teilprozesse gestaltete sich schwierig und wurde von den Schülerinnen und Schülern überwiegend widerwillig durchgeführt. Bei den Ergebnissen dazu überwog dementsprechend auch die Erfüllung der Minimalanforderungen. Hier fehlte, im Sinne von ROMEIKES Kriterien, die Relevanz für die Schülerinnen und Schüler.<sup>86</sup> Die Enddokumentation des Projekts dagegen ist als Erfolg im Blick auf den Darstellungsschwerpunkt zu werten. Die Gruppen setzten sich intensiv im Sinne einer Reflexion (auf der Meta-Ebene) mit ihrem Projekt auseinander und arbeiteten selbstständig die Resultate und besonderen Eigenschaften ihres Projekts heraus, um sie ansprechend zu präsentieren.

## 9.2 ABSCHLIEßENDE ANALYSE

Die beschriebene Unterrichtsreihe zur Einführung in die Programmierung mit Scratch sollte die Fähigkeiten der Schülerinnen und Schüler zum selbstständigen Arbeiten in einem Unterricht nach kreativen Maßstäben fördern. Die aufgestellten Hypothesen sollen nun im Kontext der gesamten Unterrichtsreihe im Hinblick darauf überprüft werden, ob und wie stark sie der Förderung der Fähigkeit zum selbstständigen Arbeiten zuträglich waren.

### **Motivation**

Die zu Beginn der Unterrichtsreihe vorgenommene Abklärung, was eine gute Programmiersprache für den Wahlpflichtunterricht kennzeichnen würde, wurde in der letzten Stunde der Unterrichtsreihe mit dem Zusammentragen von Vorteilen und

---

<sup>85</sup> Vgl. Kapitel 2.2, Zitat zur Kreativitätshemmern

<sup>86</sup> Die Einsicht, dass diese Dokumentation notwendig ist, war bei der Lerngruppe zu erkennen. Rückblickend hätte aber eine stärkere intrinsische Motivation aufgebaut werden müssen.

Grenzen von Scratch fortgeführt.<sup>87</sup> Die genannten Antworten (siehe Abbildung 4) lassen darauf schließen, dass die Lerngruppe eine insgesamt positive Einstellung zu Scratch entwickelt hatte und dass die Arbeit mit dieser Programmiersprache und die Umsetzung der Arbeitsanforderungen durch freiwillige Beschäftigung mit den

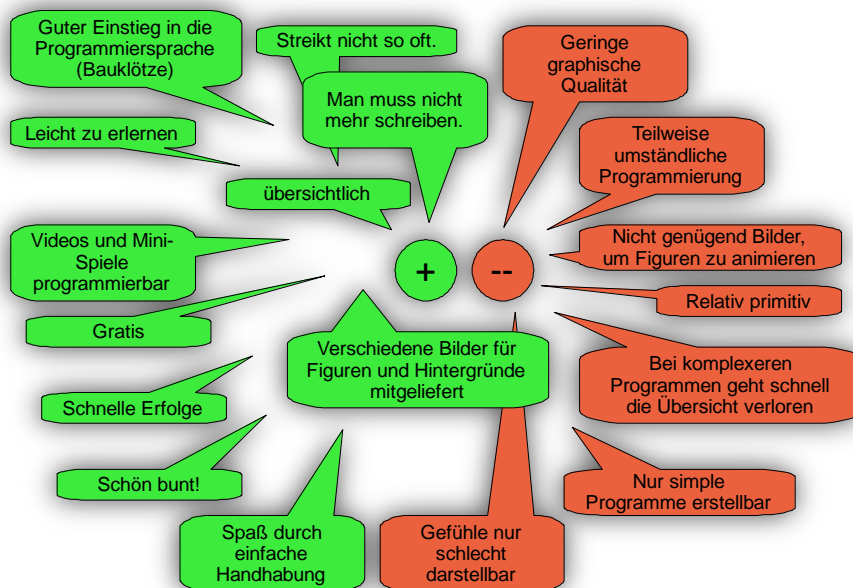


ABBILDUNG 4: VOR- UND NACHTEILE VON SCRATCH (SCHÜLERMEINUNGEN)

Unterrichtsgegenständen begünstigt wurde. Insofern lässt sich feststellen, dass selbstständiges Arbeiten durch freiwilliges Auseinandersetzen mit dem Unterrichtsgegenstand und seinem Umfeld gefördert wird.

### **Kreativität**

Der Einsatz unscharfer Zielvorgaben<sup>88</sup> und die Möglichkeit zur eigenverantwortlichen audiovisuellen und inhaltlichen Ausgestaltung der Produkte bewirkten eine hohe Identifikation der Gruppen mit ihren Projekten und ermöglichte eine durchgängige und weitestgehend ablenkungsfreie Auseinandersetzung der Schülerinnen und Schüler mit dem Unterrichtsgegenstand. Demnach wird selbstständiges Arbeiten durch die Möglichkeit zur kreativen Aneignung des Unterrichtsgegenstandes gefördert.

<sup>87</sup> Diesem Brainstorming schloss sich die Frage an, ob die Schülerinnen und Schüler einem Informatiklehrer beziehungsweise einer Informatiklehrerin den Einsatz der Programmiersprache Scratch im Wahlpflichtunterricht Informatik empfehlen würden. Dies beantworteten etwa 75% der Schülerinnen und Schüler zustimmend.

<sup>88</sup> Es sollte beachtet werden, dass unscharfe Zielvorgaben bei situationsunangemessenem Einsatz auch die Gefahr bergen, die Schülerinnen und Schüler zu überfordern.

### ***Progression***

Zu Beginn der Unterrichtsreihe forderten die Schülerinnen und Schüler noch eine starke Lenkung durch den Lehrer und klar definierte Zielvorgaben ein. Sie zeigten sich irritiert und verunsichert, weil dieses nicht erfolgte. Demgegenüber war die Lerngruppe zum Ende hin in der Lage, mit unscharfen Zielvorgaben vorzeigbare und dem Unterrichtsgegenstand entsprechende Produkte zu erzeugen, was bedeutet, dass die Fähigkeit zum selbstständigen Arbeiten eine Größe ist, die im Unterricht prozessartig entwickelt werden kann.

### ***Zeit***

Im Verlauf der Unterrichtsreihe entstand mehrfach der Eindruck, dass die zur Verfügung stehende Zeit dem Problem nicht angemessen war. Trotzdem erzielten die Gruppen vorzeigbare Resultate, die den Anforderungen entsprachen. Bei kreativen Prozessen ist es im Vorfeld grundsätzlich schwer einzuschätzen, wie viel Zeit zur Umsetzung nötig sein wird. Offenbar wirken sich auch motivationale Funktionen stark auf die notwendige Zeitspanne aus. Insofern müsste die „Zeit-Hypothese“ modifiziert werden: Die Fähigkeit zum selbstständigen Arbeiten kann nur dann gefördert werden, wenn die Schülerinnen und Schüler ausreichend Zeit zur Bewältigung der gestellten Aufgaben zur Verfügung haben, wobei diese von der Motivation zur Bearbeitung des Unterrichtsgegenstandes stark abhängt.

### ***Lehrerrolle***

Eine veränderte Lehrerrolle stellte nach Anfangsschwierigkeiten weniger für die Lerngruppe als vielmehr für den Lehrer eine Hürde dar. Die Balance zwischen Rückzug aus der Unterrichtskontrolle und Einbringen als Berater war schwer zu halten. Entscheidungen, ob bei Fehlentwicklungen eingegriffen werden musste oder (noch) abgewartet werden sollte, waren zeitnah zu treffen. Geling dies jedoch, so profitieren die einzelnen Mitglieder der Lerngruppe in mehrfacher Hinsicht (Lernfortschritt, Motivation usw.) und auch der Lehrer wurde entlastet, da er sich auf einzelne Problemfelder konzentrieren konnte. Es kann also festgestellt werden, dass eine Lehrerrolle, bei der die Beratungsfunktion im Vordergrund steht, einer Förderung der Fähigkeit zum selbstständigen Arbeiten zuträglich ist.

### ***Präsentation***

Nach dieser Unterrichtsreihe sehe ich die Bedeutung von Ergebnispräsentationen in Bezug auf die Förderung der Fähigkeit zum selbstständigen Arbeiten aus einem neuen Blickwinkel. Die Hypothese, selbstständiges Arbeiten würde durch die Möglichkeit zum Vergleich der Ergebnisse vor allem durch den zusätzlichen Motivierungseffekt gefördert, muss erweitert werden: Die mehrfache Präsentation von Ergebnissen kann als Prozess aufgefasst werden, in dem die Fähigkeit zum selbstständigen Arbeiten ebenfalls gefördert wird. Schließlich zeigten sich die Schülerinnen und Schüler zu Beginn der Reihe

noch wenig in der Lage, sinnvoll zu präsentieren und zu reflektieren beziehungsweise zu begründen. Zum Schluss gelang dies jedoch sehr gut, die Projekte konnten innerhalb der Lerngruppe ohne starke Steuerung durch den Lehrer präsentiert und diskutiert werden.

### **Dokumentation**

Bei den Dokumentationen zeigte sich, dass die aufgestellte Hypothese so nicht gehalten werden kann. Während sie für die Enddokumentationen zutreffend ist, muss sie für die Dokumentation von Zwischenergebnissen verändert werden: Eine aussagekräftige Dokumentation muss von den Schülerinnen und Schülern als Notwendigkeit und als eine Chance wahrgenommen werden, die Ergebnisse selbstständigen Arbeitens einschätzen und präsentieren zu können.

Die Unterrichtsreihe zeigte also, dass die Arbeitshypothesen mit den angeführten Modifikationen geeignet sind, einen Unterricht nach kreativen Maßstäben zu gestalten, in dem die Fähigkeit zum selbstständigen Lernen gefördert wird.<sup>89</sup> Erwähnenswert ist, dass mehrere Schüler auch außerhalb des Unterrichts weiter mit Scratch arbeiteten und beispielsweise Mini-Spiele erstellten und veröffentlichten. Ein Schüler nutzte Scratch sogar, um im Deutsch-Unterricht die Rezitation eines Gedichts visuell zu unterstützen. Auch wenn sich nicht alle Schülerinnen und Schüler dazu begeistern ließen, Scratch weiterhin zu verwenden, glaube ich, dass ihnen diese Unterrichtsreihe auch die allgemeine Programmierung nähergebracht hat.

Die verwendeten Konzepte können nun als bekannt vorausgesetzt werden und in andere Programmiersprachen übertragen werden. Somit dürften auch Einstiegshürden abgebaut worden sein. Als Ergänzung zu den bisherigen Themen in dieser Unterrichtsreihe ließen sich nun beispielsweise die Konzepte zum Zeichnen in Scratch weiterverfolgen (beispielsweise mit der Turtle-Grafik) oder andere Spiele erstellen.

Bei einer Wiederholung der Unterrichtsreihe sollten vor allem das zeitliche Konzept überprüft und die Dokumentationsansätze überarbeitet werden. Die damit verbundene mögliche Motivationssteigerung würde möglicherweise noch stärker zu einem selbstständigen Arbeiten der Schülerinnen und Schüler führen.<sup>90</sup> Auf jeden Fall werde ich in der Zukunft Scratch erneut im Zusammenhang mit kreativem Unterricht und der Zielsetzung der Förderung der Fähigkeit zum selbstständigen Arbeiten einsetzen, denn:

***„To be playful and serious at the same time is possible, in fact it defines the ideal mental condition.“ John Dewey<sup>91</sup>***

---

<sup>89</sup> Vgl. „Schüleraktivität“ und „erfolgreicher Informatikunterricht“ in Kapitel 2.2

<sup>90</sup> Es wird sich in der Zukunft zeigen, ob die gesteigerte Fähigkeit zum selbstständigen Arbeiten Bestand hat oder auf die Unterrichtsreihe beschränkt war.

<sup>91</sup> Zitiert in (Schubert & Schwill, 2004), Übersetzung A. S.: „Verspielt und ernst zum gleichen Zeitpunkt zu sein, ist möglich - das ist sogar der ideale Geisteszustand.“

## 10 QUELLEN- UND LITERATURVERZEICHNIS

Claus, V., & Schwill, A. (2003). *Duden Informatik*. Mannheim: Dudenverlag.

Claus, V., & Schwill, A. (2003). *Schülerduden Informatik*. Mannheim: Dudenverlag.

Deutsches PISA-Konsortium: Baumert, Jürgen (Hrsg.). (2001). *PISA 2000: Basiskompetenzen von Schülerinnen und Schülern im internationalen Vergleich*. Opladen: Leske + Budrich.

Klieme, E., Avenarius, H., Blum, W., Döbrich, P., Gruber, H., Prenzel, M., et al. (2007). *Zur Entwicklung nationaler Bildungsstandards - Expertise*. Berlin: Bundesministerium für Bildung und Forschung.

Knoke, F. (24. Juli 2007). *Netzwelt-Ticker: Login oder Leben! - Netzwelt - SPIEGEL ONLINE*. Abgerufen am 30. März 2008 von Spiegel Online:  
<http://www.spiegel.de/netzwelt/web/0,1518,496195,00.html>

Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A Sneak Preview. In *Second International Conference on Creating, Connecting, and Collaborating through Computing* (S. 104-109). Kyoto, Japan.

Romeike, R. (2007). Kriterien kreativen Informatikunterrichts - Beitrag zur INFOS 2007. Potsdam.

Romeike, R. (2007). *www.funlearning.de - Informatik macht Spass*. Abgerufen am 19. März 2008 von <http://www.funlearning.de/>

Rusk, N. (2007). Abgerufen am 23. März 2008 von Programming with Scratch:  
<http://scratch.mit.edu/files/Programming-with-Scratch.pdf>

Rusk, N., Resnick, M., & Maloney, J. (2007). Abgerufen am 24. März 2008 von Scratch 21st Century Skills: <http://ilk.media.mit.edu/projects/scratch/papers/Scratch-21stCenturySkills.pdf>

Schubert, S., & Schwill, A. (2004). *Didaktik der Informatik*. Heidelberg, Berlin: Spektrum Akademischer Verlag.

*Scratch | About*. (2007). Abgerufen am 21. März 2008 von Scratch:  
<http://scratch.mit.edu/about>

*Scratch-Download*. (2007). Abgerufen am 19. März 2008 von  
<http://scratch.mit.edu/download>

*Scratch-Homepage*. (2007). Abgerufen am 19. März 2008 von <http://scratch.mit.edu/>

Senatsverwaltung für Bildung, Jugend und Sport Berlin. (2006). *Rahmenlehrplan für die Sekundarstufe I ITG/Informatik Wahlpflichtfach*. Berlin.

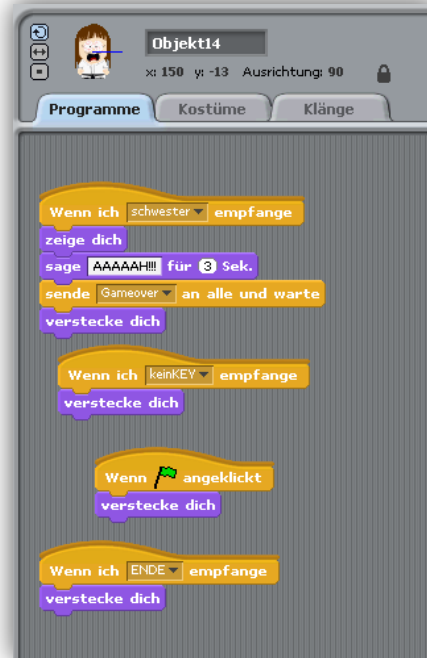
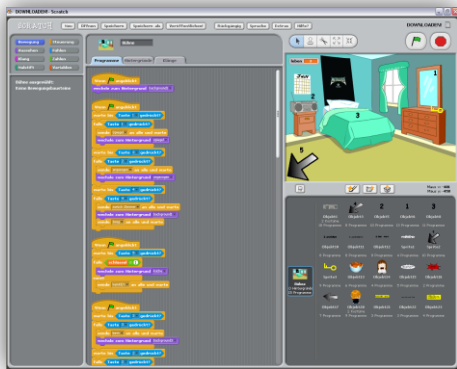
*Spotlight - Creating from Scratch*. (2008). Abgerufen am 21. März 2008 von Lifelong Kindergarten :: MIT Media Lab: <http://ilk.media.mit.edu/spotlight.php?id=3>

*Squeak*. (2008). Abgerufen am 23. März 2008 von Squeak-Homepage:  
<http://www.squeak.org/>

Weinert, F. E. (2001). Vergleichende Leistungsmessung in Schulen - eine umstrittene Selbstverständlichkeit. In F. E. Weinert, *Leistungsmessungen in Schulen* (S. 17-32). Weinheim: Beltz.

# 11 ANHANG

## A) Das Projekt „Bennys Schulweg“ von S5 und S4

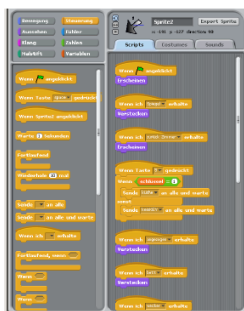


Szenario und Beispielquellcode

### Bennys Schulweg

In unserem mit Scratch erstellten Spiel Bennys Schulweg, handelt es sich um einen Jungen, den man pünktlich zur Schule bringen muss. Auf seinem Weg muss der Spieler Entscheidungen treffen, damit er rechtzeitig ankommt und davor z.B. nicht von seiner Schwester K.O. geschlagen oder auf eine andere Art ein Leben verliert. Die Zahlen, die man im Zimmer sieht, sind die Tasten, auf die man drücken muss, um mit dem jeweiligen Objekt zu interagieren. Der Spieler hat drei Leben, wenn alle verbraucht sind, ist das Spiel vorbei.

Scratch funktioniert mit einem Drag & Drop (ziehen und fallen lassen) – System. Man zieht die Programme aus der Box links in das Feld daneben und baut sich so Schritt für Schritt ein Spiel oder einen Kurzfilm.



Es gibt links oben acht Kategorien, in denen die Bausteine eingeordnet sind. Klickt man auf eine Kategorie, sieht man alle Bausteine, die dort eingeordnet sind. Das macht das System einfacher und übersichtlicher. Dann zieht man die Bausteine von der linken in die rechte Box und steckt sie dort zu einem Programmabau ein zusammen.

#### Erklärung von einem Baustein:

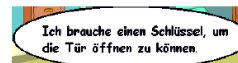


Wenn Flagge angeklickt:

Wenn die Flagge, also das Programm gestartet wird, angeklickt wird, dann warte bis Taste 5 gedrückt wird und falls die Variable, Schlüssel gleich 1 (das heißt der Schlüssel wurde aufgesammelt und kann verwendet werden) ist, dann wechselt das Programm zum Hintergrund Küche, wo man weitere Optionen hat.

Wenn jedoch kein Schlüssel aufgesammelt wurde dann sendet das Programm an alle „keinKEY“ und wartet.

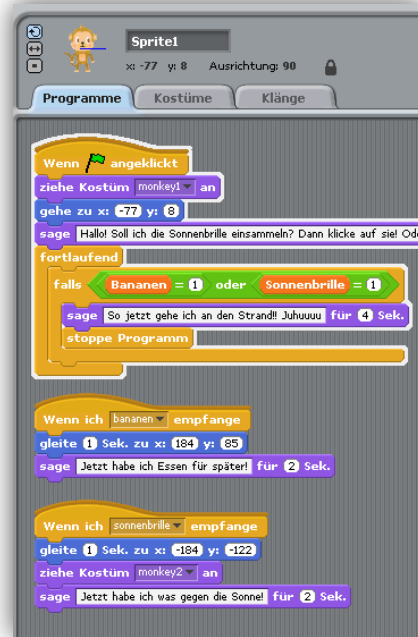
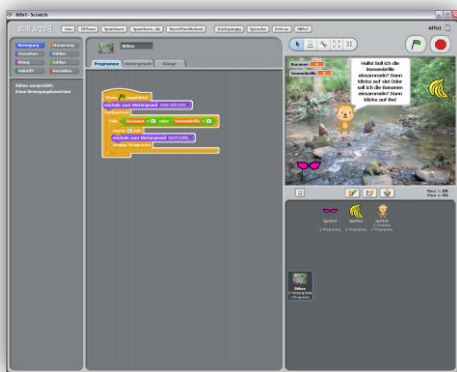
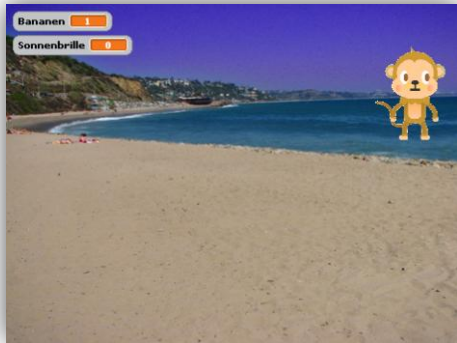
Kein Key bedeutet, dass man nicht in die Küche gehen kann und folgende Fehlermeldung erscheint:



Und so kann man den Raum nicht betreten und muss deshalb den Schlüssel finden, der sich in Benny's Zimmer befindet.

### Projektdokumentation

B) Das Projekt „Affe“ von S3, S15 und S8



Szenario und Beispielquellcode



**Bausteine**



- Über Bausteine wie diese funktioniert das Programm.
- Die einzelnen Bausteine sind durch Beschriftung leicht verständlich.

**Beispiel für einen Baustein**



- Ist die grüne Fahne angeklickt erscheint der Affe an der Position X:-77 und y:8.
- Er fragt: Hallo! Soll ich..
- Nun kann man sich zwischen den Bananen und der Sonnenbrille entscheiden.
- Klickt man auf die eins der beiden sagt der Affe: So jetzt gehe ich an den Strand.

Projektdokumentation