

Fachbereich Mathematik und Informatik
Institut für Didaktik der Mathematik und Informatik
Sommersemester 2011
1. Prüfer Prof. Dr. Marco Thomas
2. Prüfer Prof. Dr. Achim Clausing
Abgabetermin 27.07.11

Konzeption einer kontext-orientierten Unterrichtsreihe im Fach Informatik zum Thema Haussteuerung

-Masterarbeit-

vorgelegt von:
David Beckmann
Studiengang: Master of Education
Studiengänge: Mathematik, Informatik

Inhaltsverzeichnis

1 Hinführung zum Thema.....	4
1.1 Fragestellung der Arbeit.....	5
2 Informatische Konzepte.....	5
2.1 Informatik im Kontext.....	6
2.2 Objektorientierter Ansatz.....	8
2.2.1 Objektorientierung als Einstieg in die Informatik	9
2.3 Informatiksysteme.....	11
3 Haussteuerung.....	12
3.1 Didaktische Konzepte zur Haussteuerung.....	13
3.2 Zugang über eine Entwicklungsumgebung (Lernumgebung).....	15
3.2.1 Lego Mindstorms.....	16
3.2.1.1 Simulation einer Haussteuerung mit Lego Mindstorms.....	18
3.2.1.2 Beurteilung für den schulischen Kontext.....	19
3.2.2 Das PUMA-System.....	20
3.2.2.1 Beurteilung für den schulischen Kontext.....	21
3.2.3 Scratch.....	22
3.2.3.1 Simulation einer Haussteuerung mit Scratch.....	24
3.2.3.2 Beurteilung für den schulischen Kontext.....	24
4 Entwurf einer Unterrichtsreihe.....	27
4.1 Bedingungsanalyse.....	27
4.2 Sachanalyse.....	29
4.2.1 Systemsteuerung.....	29
4.2.2 Sensoren und Aktoren.....	31
4.2.3 Struktogramme.....	32
4.2.4 Kontrollstrukturen.....	32
4.3 Didaktische Analyse	33
4.3.1 Didaktische Reduktion.....	33
4.3.2 Sequenzierung auf Makroebene.....	36
4.3.3 Sequenzierung auf Mikroebene.....	38
4.3.4 Lernziele.....	42
4.3.4.1 Inhaltliche Lernziele.....	42

4.3.4.1.1 Didaktische Auswahlkriterien für die Lerninhalte.....	43
4.3.4.2 Bildungsstandards.....	44
4.3.4.2.1 Bildungsstandards für das Fach Informatik.....	45
4.3.4.3 Methodische Lernziele.....	49
4.3.5 Motivationale Aspekte.....	51
4.3.5.1 Lernmotivation/ Leistungsmotivation.....	52
4.3.5.2 intrinsische/extrinsische Motivation.....	54
4.3.5.3 Eingangs-/ Verlaufsmotivation.....	56
5 Ausblick.....	57
LITERATURVERZEICHNIS.....	60
ANHANG.....	64

1 Hinführung zum Thema

Im Jahre 2009 titelt die ZEIT „Ein bisschen Word. An vielen Schulen fristet der Informatikunterricht ein Schattendasein.“¹ und fordert einen längst überfälligen Neustart. Für besonders beklagenswert hält die Autorin das Fehlen curricularer Bestimmungen sowie einheitlicher Bildungsstandards der Kultusminister-Konferenz. Diese sind für eine Gleichstellung des Fachs mit den anderen Naturwissenschaften notwendig. Sie fehlen bis heute.

Problematisch ist dies vor allem, da die fachlichen Inhalte nicht eindeutig definiert sind und es somit keinen Konsens über die Frage gibt, was die Schüler² innerhalb des Informatikunterrichts erlernen sollen beziehungsweise welche Inhalte unverzichtbar sind. Neben diesen fehlenden Konkretisierungen kommt der Mangel an guten Lehrmitteln erschwerend hinzu:

Zwar existieren Lehrmittel zur Nutzung von ICT (Anwendungswissen), produktorientierte Informatikanwendungen oder Einführungen oder aber eher abstrakte, nicht der Lebenswelt von Kindern und Jugendlichen angepasste Informatiklehrmittel für Studierende.³

Dabei erreicht der Informatikunterricht eine immer größere Bedeutung, wenn man bedenkt, wie viel Schüler mit Computer, zum Beispiel durch Internet oder Mobiltelefone, in Berührung kommen. Fothe/ Friedrich (2011) kommentieren dies damit, dass die Schüler die Konzepte nur verstehen, wenn sie „hinter die Kulissen schauen“ und sich in den neuen Informations- und Kommunikationssystemen die Chancen und Risiken sowie Möglichkeiten und Grenzen bewusst machen.⁴ Weiterhin stellen sie fest, dass die Vor-

¹ Hanselmann, Uta: »Ein bisschen Word«. In: Die Zeit. Nr.9. (19/02/2009). <http://www.zeit.de/2009/09/C-Informatik> (Stand: 27.7.2011).

² Innerhalb dieser Arbeit werden unter der Zusammenfassung "Schüler" selbstverständlich Schülerinnen und Schüler gefasst.

³ Döbeli Honegger, Beat/ Frey, André/ Braxmeier, Philipp: Mit iLearnIT.ch spielerisch das Interesse an Informatik wecken. In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber. Bonn: Köllen2009, S.305.

⁴ Vgl. Fothe, Michael/ Friedrich, Steffen: Informatik in die Schule! - ein erneutes Plädoyer. (05/2011). <http://www.gi.de/fileadmin/redaktion/Vorstandsglossen/GI-Vorstandsmitglied-Fothe110523.pdf> (Stand: 27.07.2011).

aussetzung für eine erfolgreiche Berufsausbildung oder ein Studium nur durch einen guten und anspruchsvollen Informatikunterricht sichergestellt werden kann.⁵

Auch wenn der Informatikunterricht es den Schülern erleichtert, im Umgang mit Informatiksystemen kreativ zu sein, ist das Anwenden von kreativen Lehrmethoden im Unterricht notwendig.⁶

Nach Romeike (2007) ist es sinnvoll, ein Thema zu wählen, welches sich aus der Lebenswelt der Schüler ergibt. Nur so kann der Schüler einen persönlichen Bezug und somit eine Relevanz für sein eigenes Leben herstellen.⁷

Es gilt insbesondere den Schülern plausibel zu machen, welche starken Auswirkungen die Informatik auf die Gesellschaft hat, einer Gesellschaft, die von Informationen und Informationsflüssen geprägt ist.⁸

1.1 Fragestellung der Arbeit

Die Arbeit widmet sich der Erarbeitung eines Unterrichtskonzepts für den Bereich "Informatik im Kontext".

Nachdem zunächst die informatischen Grundlagen in Form von drei ausgewählten Konzepten Informatik im Kontext, Objektorientierung und Informatiksysteme gelegt werden, wird ein Bezug zu dem Sachzusammenhang Haussteuerung hergestellt.

Vor allem müssen hierfür verschiedene Umsetzungsmöglichkeiten vorgestellt und bewertet werden, so dass daraus begründete Entscheidungen für die Wahl einer Lernumgebung für die Unterrichtsreihe getroffen werden können.

Schließlich wird im weiteren Verlauf der Arbeit die Unterrichtsreihe entwickelt und hinsichtlich theoretisch-fachlicher sowie methodisch-didaktischer Gesichtspunkte beurteilt.

⁵ Vgl. Fothe / Friedrich (2011).

⁶ Romeike, Ralf: Kriterien kreativen Informatikunterrichts. In: Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn: Köllen2007, S.59.

⁷ Vgl. Romeike 2007, S.59.

⁸ Vgl. Schubert, Sigrid/Schwill, Andreas: Didaktik der Informatik. 2. Auflage. Heidelberg: Spektrum2011, S.6.

2 Informatische Konzepte

Hier sollen nun zunächst einige didaktische Konzepte vorgestellt werden, die im Sachzusammenhang des Themas Haussteuerung eingeflochten und in der späteren Unterrichtsreihe berücksichtigt werden. Im Hinblick auf die erfolgreiche Vermittlung der Lerninhalte haben diese ihren jeweiligen Nutzen.

2.1 Informatik im Kontext

Unter „Informatik im Kontext“ werden Unterrichtsinhalte im Fach Informatik verstanden, die sich an alltagsbezogenen Lebenssituationen der Schüler orientieren. Zu diesem Thema hat sich eine Arbeitsgruppe⁹ zusammengefunden, die Materialien entwirft und auf ihren Nutzen für den Unterricht hin untersucht. Durch den Kontext werden die Schüler motiviert und der Sinnzusammenhang der zugehörigen fachlichen Inhalte wird ihnen bewusst.¹⁰

Nach Krapp (1998) handeln Schüler nach dem Modell der Interessengenese. Das heißt, wenn Schüler einem konkreten Handlungsablauf besondere Aufmerksamkeit schenken, kann sich daraus ein dauerhaftes Interesse herausbilden, sich näher und intensiver damit zu beschäftigen.¹¹ Somit können informatische Inhalte für Schüler über einen längeren Zeitraum hinweg attraktiv gemacht und es kann dafür das Interesse geweckt werden.

Die Konzepte sollten den Kriterien der Mehrdimensionalität, Breite, Tiefe, Lebenswelt und Stabilität standhalten.¹² Zum einen werden dann zu einem Thema sowohl informatische als auch zum Beispiel ökologische, ökonomische, rechtliche oder ethische Aspekte mit eingebracht. Hierbei ist dann besonders zu beachten, ob der Kontext wirklich gesellschaftlich relevant ist oder nur künstlich aufgesetzt worden ist. Auch in anderen natur-

⁹ Homepage Informatik im Kontext. <http://www.informatik-im-kontext.de/> (Stand: 27.07.2011).

¹⁰ Vgl. Weigend, Michael: Algorithmik in der Grundschule. In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber. Bonn: Köllen2009, S.106.

¹¹ Vgl. Krapp, A.: Entwicklung und Förderung von Interessen im Unterricht. In: Psychologie in Erziehung und Unterricht, Ernst Reinhardt Verlag München Basel1998, S.185-201.

¹² Vgl. Homepage Informatik im Kontext. <http://medienwissenschaft.uni-bayreuth.de/inik/konzepte.html> (Stand: 27.07.2011).

wissenschaftlichen Fächern findet eine Kontextorientierung statt. In der Biologie wird der Kontextgedanke so gesehen:

Den Lernenden werden Anknüpfungspunkte an ihr Vorwissen und ihre Alltagserfahrungen geboten sowie wissenschaftliche Anwendungsfelder vermittelt.¹³

Für die Informatik bedeutet das die Möglichkeit, genuine Inhalte, die oftmals nur sehr abstrakt zu erfassen sind und keinen wirklichen Lebensweltbezug haben, durch einen motivierenden Kontext, der auf Vorerfahrungen der Schüler fußt, den Schülern näher zu bringen.¹⁴

Des Weiteren müssen die Kompetenzen aus den Bildungsstandards¹⁵ verankert sein, um fachlichen Kriterien gerecht zu werden.¹⁶ Natürlich sollte man nicht versuchen, allen Kriterien zur gleichen Zeit gerecht zu werden, da dies einerseits kaum möglich ist, andererseits zur Überforderung führen kann.

Das pädagogische Gegenstromprinzip der Orientierung sowohl an Ereignissen an der Lebenswelt (bottom-up) als auch an Fachprinzipien und Kompetenzen (top-down) zeichnet den kontextorientierten Unterricht aus.¹⁷

Im weiteren Verlauf des Unterrichts ist es jedoch ratsam, eine Thematik von verschiedenen Seiten zu betrachten, damit die in den Bildungsstandards verankerten Kriterien eingehalten werden können.

Der Kontext soll also ganz konkret die Lebenswelt der Schülerinnen und Schüler berühren und somit potentiell erlebbar sein, um den Unterricht für die Schüler anregend zu gestalten. Hier müssen Genderaspekte berücksichtigt werden.¹⁸

¹³ Homepage Biologie im Kontext. <http://bik.ipn.uni-kiel.de/typo3/index.php?id=30> (Stand: 27.07.2011).
¹⁴ Vgl. Weigend 2009, S.106.

¹⁵ Auf die Bildungsstandards wird im Verlauf dieser Arbeit noch näher eingegangen.
Vgl. Kapitel 4.3.4.2 dieser Arbeit.

¹⁶ Homepage Biologie im Kontext. <http://bik.ipn.uni-kiel.de/typo3/index.php?id=30> (Stand: 27.07.2011).

¹⁷ Koubek, Jochen/ Schulte, Carsten/ Schulze, Peter/ Witten, Helmut: Informatik im Kontext (IniK). Ein integratives Unterrichtskonzept für den Informatikunterricht. In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber Bonn: Köllen2009, S. 275.

¹⁸ Für den Bereich "Informatik im Kontext" lassen sich verstärkt Aspekte finden, welche sich positiv auf die Motivation von Mädchen bezüglich des Unterrichtsfachs Informatik auswirken. Hierzu zählen die Orientierung an der Lebenswelt sowie kommunikative Aspekte.
Nähere Aufführungen hierzu sind zu finden bei Crutzen, Cecile K. M.: Dekonstruktion, Konstruktion und Inspiration. Gender in der Informatik. In: Dokumentation wissenschaftlicher Kolloquien 1999-2002. Band 2. Hrsg. von Ursula Paravicini und Maren Zempel-Gino. Hannover: NFFG2003.

Schließlich muss der Kontext auch zeitlich stabil sein und Relevanz für einen längeren Zeitraum besitzen.¹⁹

2.2 Objektorientierter Ansatz

Die Softwareentwicklung im objektorientierten Ansatz gliedert sich in Analyse, Entwurf und Implementierung.

In der Analyse werden die Ansprüche an das Informatiksystem im Groben erörtert. Auf der Basis der Beschreibungen der Analyse wird dann im Entwurf eine Konkretisierung der Komponenten des Informatiksystems²⁰ vorgenommen. Der Entwurf stellt somit die Überlegungen auf einem "höheren Abstraktionsniveau" dar.²¹ Schließlich wird mittels der Implementierung der Entwurf in einer Programmiersprache realisiert.

In allen Phasen stehen die Objekte im Fokus.

Unter Objekten werden Personen, Dinge oder Begriffe gefasst.²²

Jedes Objekt der Realität, das Informationen aufnimmt, speichert, verarbeitet und abgibt, kann man als Informationsobjekt beschreiben. Objekte der Realität können untereinander interagieren, diese Interaktion kann man als Kommunikation zwischen Informationsobjekten beschreiben. Ein Informationssystem ist die Realisierung einer Anzahl von Informationsobjekten – und damit ein Objekt der Realität, dass man (hier schließt sich ein Kreis) wiederum als Informationsobjekt beschreiben kann.²³

Das bedeutet, man ordnet jedem Objekt, welches in der Realität existiert, Zustände und Verhaltensweisen zu, analysiert deren Interaktionen, wodurch sich diese in einem Modell beschreiben lassen.²⁴

¹⁹ Vgl. Homepage Informatik im Kontext. <http://www.informatik-im-kontext.de/> (Stand: 27.07.2011).

²⁰ Alle Objekte innerhalb des Informatiksystems.

²¹ Vgl. Balzert, Heide: Lehrbuch der Objektmodellierung. Analyse und Entwurf. Berlin/Heidelberg: Sprektrum1999, S. 538.

²² Vgl. Balzert 1999, S.18.

²³ Crutzen, Cecile K. M./ Hein, Hans-Werner: Objektorientiertes Denken als didaktische Basis der Informatik. In: Innovative Konzepte für die Ausbildung. Hrsg. von Sigrid Schubert. Berlin/Heidelberg: Springer1995, S.150.

²⁴ Vgl. Crutzen 1995, S.150f.

Interaktionen lassen sich in zeitlich begrenzte oder zeitlich unbegrenzte gliedern²⁵, folglich können zwei Objekte eine unterschiedliche Interaktionsdauer aufweisen, was zur Folge hat, dass ebendiese Dauer genau bestimmt und im Modell²⁶ berücksichtigt werden muss.

Weiterhin werden den Objekten Attribute und Methoden zugeordnet. Attribute sind bestimmte Eigenschaften von Objekten, Methoden hingegen bezeichnen Operationen.²⁷ Beispielsweise hätte das Objekt 'Student' die Attribute *Matrikelnummer*, *Alter*, *Name* und die Methode *belegeVorlesung()*.

Zur Implementierung eines Informatiksystems können ausschließlich solche Objekte einbezogen werden, die zum unmittelbaren Nutzen sind. Das bedeutet, aus dem Ausschnitt der Realität werden lediglich die relevanten Objekte betrachtet, alle anderen werden ausgeblendet.²⁸

Ein wichtiges Objekt der Realität sind die künftigen Benutzer des Informatiksystems. Deshalb kommt der Interaktion zwischen System und Benutzer ein besonderer Stellenwert zu, da ein Informatiksystem nicht selbstreferentiell ist. Vielmehr ist es auf eine Nutzung ausgerichtet.²⁹

2.2.1 Objektorientierung als Einstieg in die Informatik

Da die Objektorientierung im Moment das vorherrschende Programmierparadigma ist, scheint es durchaus angemessen, die grundlegenden Konzepte, wie Objekte und Klassen, möglichst früh im Unterricht zu behandeln, damit diese intensiv betrachtet und somit auch ausreichend eingeübt beziehungsweise vertieft werden können.³⁰

²⁵ Vgl. Crutzen 1995, S.150.

²⁶ Auf die unterschiedlichen Modelle und deren speziellen Nutzen für die Modellierung wird im Verlauf der Arbeit noch Bezug genommen.

²⁷ Vgl. Balzert 1999, S.544.

²⁸ Vgl. Crutzen 1995, S.150.

²⁹ Vgl. Crutzen 1995, S.150f.

³⁰ Vgl. Böstler, Jürgen: Objektorientiertes Programmieren – Machen wir irgendwas falsch? In: Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn: Köllen2007, S.11.

Schon früh stellt Schwill (1995) drei Gründe vor, die nahelegen, objektorientierte Programmierung gerade im Anfangsunterricht einzuführen. Zunächst erfüllt es die Forderung nach einem „zeitgemäßen Unterricht mit mächtigen Konzepten, wie Erweiterbarkeit, Anpaßbarkeit, Rekonfiguration, Vererbbarkeit, Kapselung“³¹ sowie eine stärkere Anwenderorientierung durch die Benutzung des Computers beziehungsweise einer Software.

Weiterhin lässt dieser Ansatz ein didaktisches Prinzip der Fortsetzbarkeit zu, das bedeutet, dass im schulischen Kontext stets auf die Inhalte zurückgegriffen werden kann und diese in einer Art „Spirale“³² stets, den kognitiven Leistungen eines Schülers entsprechend, vertieft werden können.³³

Als letzten Grund nennt Schwill (1995) die Möglichkeiten, die objektorientierte Orientierung bezüglich kognitiver Verarbeitung bietet. Folglich nutzt der objektorientierte Ansatz gezielt vorteilhaft die menschliche Kognition aus, da elementare kognitive Prozesse des Denkens, Erkennens und Problemlösens ähnlich ablaufen.³⁴

Obwohl Informatik nicht mit Programmieren gleichzusetzen ist, wird zur Vermittlung von informatischen Grundkonzepten auch eine Programmiersprache benötigt.³⁵

Programmieren ist, so Börstler (2007), eine komplexe Fertigkeit, welche sehr schwierig zu erlernen ist. Daher muss insbesondere im Anfangsunterricht des Fachs Informatik eine intensive Befassung mit Programmierung vorgenommen werden.

Eine besondere Herausforderung für Anfänger ist die Programmierung auch deshalb, weil sie ein vollständig abstraktes Gebiet ist, zu welchem sie kein Vorwissen haben beziehungsweise ein schlüssiges Erklärungsmodell aufbauen können.³⁶

³¹ Schwill, Andreas: Programmierstile im Anfangsunterricht. In: Innovative Konzepte für die Ausbildung. Hrsg. von Sigrid Schubert. Berlin/Heidelberg: Springer1995, S.183.

³² Hier wird auf das Spiralkurrikulum nach Skinner eingegangen. Nach diesem finden Themen stets erneut Einzug in den Unterrichtsalltag und werden immer stärker vertieft beziehungsweise abstrahiert.

³³ Vgl. Schwill 1995, S.183.

³⁴ Vgl. Schwill 1995, S.183f.

³⁵ Döbeli/Frey/Braxmeier 2009, S.310.

³⁶ Vgl. Böstler 2007, S.14.

Bezüglich der Erschließung von Programmieren im Informatikunterricht, vor allem im Anfangsunterricht, erscheint die Nutzung von Lernumgebungen, seien es Kara, Puma oder Scratch³⁷, als sehr hilfreich. Schüler können so einfache Probleme lösen, ohne über das Wissen von vielen „Programmierplänen“ zu verfügen.

AnfängerInnen können ihre Pläne nicht so leicht generalisieren wie ExpertInnen. (...) Das Hauptproblem von AnfängerInnen liegt jedoch nicht in der Anzahl der bekannten Pläne, sondern in ihren begrenzten Fähigkeiten diese zu komplexen Programme zusammenzusetzen.³⁸

Geeignete Lernumgebungen können gerade diese Probleme aussetzen, indem sie den Schülern eine möglichst einfache und überschaubare Sammlung an Operatoren zur Verfügung stellt, mit welchen sie einfache Baupläne intuitiv beziehungsweise experimentell herstellen können. Es wird so ein einfacher Zugang zu der Materie ermöglicht.

2.3 Informatiksysteme

Eine Leitlinie der Schulinformatik sind die Wirkprinzipien von Informatiksystemen.³⁹ Daran sollen die Schüler die Funktionen der Systemkomponenten nachvollziehen. Um dies im Unterricht angemessen durchführen zu können, sind Systeme vonnöten, die eine geeignete Komplexität aufweisen und dabei aber nicht die Wirkungsweisen der Komponenten verfälschen.⁴⁰

Dabei handelt es sich nicht um eine Programmierung von Geräten wie DVD-Player oder das Einschalten von Fernsehern sowie das Steuern von Autos, sondern um das Erlernen der Organisation geistiger Tätigkeiten.⁴¹

Als Informatiksystem bezeichnet man die spezifische Zusammenstellung von Hardware, Software und Netzverbindungen zur Lösung eines Anwendungsproblems. Eingeschlossen sind alle durch die Einbettung des Systems in den Anwendungsbereich beabsichtigten

³⁷ Auf diese Lernumgebungen wird im Verlauf der Arbeit unter 3.2 eingegangen.

³⁸ Böstler 2007, S.14.

³⁹ Vgl. GI: Grundsätze und Standards für die Informatik in Schulen. Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen der GI erarbeitet vom Arbeitskreis „Informatik“. In: LOG IN. Heft 150/151. Beilage zur LOG IN. Berlin: LOG IN Verlag2008, S.12-14.

⁴⁰ Vgl. Herper, Henry/Hinz, Volkmar: Analyse eines Informatiksystems durch unterschiedliche Modellierungsansätze. In: Unterrichtskonzepte für informative Bildung. Hrsg. von Steffen Friedrich Bonn: Kölle2005, S.253.

⁴¹ Vgl. Schubert/Schwill 2011, S.213.

oder verursachten nichttechnischen Fragestellungen und ihre Lösungen, also Fragen der Gestaltung des Systems, der Qualifizierung der Nutzer, der Sicherheit sowie der Auswirkungen und Folgen des Einsatzes. Informatik ist dann die Wissenschaft von Entwurf und Gestaltung von Informatiksystemen.⁴²

Dieses Produkt der Informatik wird von Syrbe als soziotechnisches System bezeichnet und für die Informatik nutzbar gemacht.⁴³

Das Informatiksystem ist hierbei nicht nur bezüglich seines technologischen Sachsystems zu analysieren, sondern auch in Bezug auf soziale Dimensionen. Das bedeutet, dass innerhalb des Informatiksystems zwischen der Repräsentationsebene, der Kommunikationsebene und der Wissensebene unterschieden wird, also die Bereiche Maschine, Interaktion in der Gruppe und der individuelle Wissensstand zu differenzieren sind.⁴⁴

3 Haussteuerung

As a very basic definition of home automation you could consider it to be anything that is able to be controlled in your home. If you have a remote or you have automatic control over the function within your home, you have home automation.⁴⁵

Der Gebrauch von unterschiedlichen Sensoren und Aktoren, deren Werte und Zustände ständig geprüft werden müssen, da sie davon abhängige Aktionen auslösen, macht eine moderne Haussteuerung aus. Als Sensoren wären zum Beispiel Temperatur-, Luftfeuchte-, Einstrahlungs- oder Bewegungssensoren zu bezeichnen. In den Bereich der Aktoren fallen Heizungsventile, Motoren zum Öffnen der Fenster oder die Steuerung von Wasserleitungen über Magnetventile.⁴⁶

⁴² Claus, Volker/ Schwill, Andreas (Hrsg.): Duden Informatik. Ein Fachlexikon für Studium und Praxis.- Mannheim: Duden Verlag 2001, S. 301.

⁴³ Syrbe steht hier in Beziehung zu Ropohl, der das soziotechnische System als eines von vier Modellen ansieht.

Zitiert nach Magenheim, Johann S.: Informatiksystem und Dekonstruktion als didaktische Kategorien – Theoretische Aspekte und unterrichtspraktische Implikationen einer systemorientierten Didaktik der Informatik. Universität Paderborn 2000, S. 3.

⁴⁴ Vgl. Magenheim (2000), S. 3.

⁴⁵ Lindsay, Stephen: Home Automation 101. Discover How To Create Or Transform Your Current Home Today, S.5. http://books.google.de/books?id=1-WJhqL0H6EC&printsec=frontcover&dq=home+automation+101&hl=de&ei=GyUsTtCKHsPngam2IDtDQ&sa=X&oi=book_result&ct=book-thumbnail&resnum=2&ved=0CD0Q6wEwAQ#v=onepage&q&f=false (Stand: 27.07.2011).

⁴⁶ Vgl. Lerch, Reinhard: Elektrische Messtechnik. Analoge, digitale und computergestützte Verfahren. 5.Auflage. Heidelberg/ Dordrecht/ London/ New York: Springer-Verlagh2010, S.638.

Die Steuerung übernimmt eine speicherprogrammierbare Steuerung (SPS), die in einer geeigneten Sprache programmiert werden muss. Dabei ist zu bedenken, dass Messwerte der Sensoren und Parameter der Aktoren eingelesen und möglicherweise angepasst werden müssen. Die Verarbeitung erfolgt in kleinen Modulen, deren Ergebnisse dann Aktionen in Form von Aktorensteuerung nach sich ziehen. Die Steuerung sollte auch kommunikationsfähig sein, indem ein Benutzer durch Angaben von Parametern innerhalb eines Bedienprogramms mit ihr interagieren kann. Womöglich gibt es auch noch weitere Steuerungen, die dann auch untereinander interagieren müssen.⁴⁷

3.1 Didaktische Konzepte zur Haussteuerung

Beleuchtet man den Gegenstand Haussteuerung näher, so erkennt man, dass es sich hierbei um ein Informatiksystem nach Schwill⁴⁸ handelt. Dabei ist die Hardware die speicherprogrammierbare Steuerung sowie deren Peripheriegeräte und die Software die Programmierung. Haussteuerung ist ein Anwenderproblem, zu dem man im Speziellen auch die Qualifizierung der Nutzer, die Gestaltung des Systems, Sicherheitsaspekte sowie Folgen des Einsatzes diskutieren sollte.

Es ist also zu fragen, was alles zu beachten ist, wenn man dieses System umsetzen möchte, und wo Schwierigkeiten auftreten könnten.

Soziale Aspekte der Haussteuerung im Sinne eines soziotechnischen Systems zu benennen fällt schwer, da keine Nutzerinteraktion stattfindet. Dennoch ändert sich das Verhalten der Nutzer insofern, dass sie konkrete Handlungen, wie etwa Änderungen in der Heizungssteuerung, dem System überlassen.

Auch die Objektorientierung lässt sich gut auf den Gegenstand Haussteuerung anwenden, indem man die Sensoren und Aktoren jeweils als Objekte ansieht, die Informationen austauschen und mit Hilfe der Steuerung miteinander kommunizieren. Es handelt sich um in der Realität vorhandene Objekte, die verschiedene Zustände und Verhaltensweisen einnehmen. Man kann ihnen Attribute wie Laufzeiten und Richtwerte, beispielsweise die Lichtintensität eines Lichtsensors, oder Methoden wie messen() und drehe-

⁴⁷ Vgl. Lerch 2010, S.638.

⁴⁸ Vgl. Claus/Schwill 2001, S. 301.

Motor() zuweisen. Schließlich ist bei der Implementierung auch der Nutzer zu beachten. Da die Haussteuerung auch manuell bedienbar sein soll, müssen hier Kommunikationsmöglichkeiten zwischen Nutzer und System geschaffen werden.

Gerade wenn man an Konzepte für den Anfangsunterricht nach Schwill mit Erweiterbarkeit, Anpaßbarkeit, Rekonfiguration, Vererbbarkeit, Kapselung⁴⁹ denkt, lässt sich in Bezug auf die Haussteuerung einiges benennen. In der häuslichen Umgebung liegen immer bauliche Unterschiede, so dass kein Installationsort dem anderen gleicht. Außerdem soll die Möglichkeit bestehen eine vorhandene Haussteuerung durch weitere Funktionen erweitern zu können oder aber das System teilweise neu zu konfigurieren.

Weiterhin ist Haussteuerung ein Thema zur alltagsbezogenen Lebenssituation der Schüler, das sich eignet, im Sinne von Informatik im Kontext⁵⁰ eine Unterrichtseinheit zu werden. Dabei ist einerseits darauf zu achten, dass informatische Inhalte aus den Bildungsstandards sowie außerinformatische Inhalte ihren Platz haben und damit eine gewisse Mehrdimensionalität geschaffen wird.

Zum Bereich der informatischen Inhalte empfiehlt es sich, zum Gebiet der einfachen Algorithmen Einführungen in Schleifen und If-Anweisungen zu geben, da diese für die Behandlung und Umsetzung der Sensorwerte und die Ansteuerung der Aktoren essenziell sind. Die Nutzung beziehungsweise Einführung in eine Programmiersprache oder Programmierumgebung, um damit Problemlösungen innerhalb des Systems zu realisieren, scheint für die Kompetenzen im Bereich „Sprachen und Automaten“ sinnvoll. Des Weiteren ist der Umgang mit dem Informatiksystem Haussteuerung zu nennen und dabei zwischen der Soft- und Hardwareebene zu trennen. Zum Bereich „Informatik, Mensch und Gesellschaft“ können Auswirkungen des Systems Haussteuerung benannt und diskutiert werden. Hier lässt sich auch sehr gut der Bogen zu den außerinformatischen Inhalten, schlagen um die von der Arbeitsgruppe Informatik-im-Kontext geforderte Breite⁵¹ zu realisieren.⁵²

⁴⁹ Vgl. Schwill 1995, S.183.

⁵⁰ Vgl. Homepage Informatik im Kontext. <http://www.informatik-im-kontext.de/> (Stand: 27.07.2011).

⁵¹ Vgl. Homepage Informatik im Kontext. <http://www.informatik-im-kontext.de/> (Stand: 27.07.2011).

⁵² Vgl. GI 2008.

Insgesamt lassen sich also einige didaktische Konzepte mit dem Themenkomplex Haussteuerung verankern. Zur genauen Umsetzung erfolgen nun einige Umsetzungsvorschläge.

3.2 Zugang über eine Entwicklungsumgebung (Lernumgebung)

Gerade im Anfangsunterricht empfiehlt sich der Zugang über eine Entwicklungsumgebung. Ein besonderer Vorteil hier ist, dass Schüler schon mit geringen Programmierkenntnissen leistungsfähige Produkte erstellen können.⁵³ Durch eine meist objektorientierte Herangehensweise lässt sich die Problemstellung leicht auf Objekte und deren Operationen übertragen. Vorgegebene Konstrukte erlauben es dem Programmierer, sich auf das Zusammenstellen der einzelnen Bausteine zu beschränken, und es ähnelt dem Finden einer Konfiguration. Einfaches Bedienen und Erlernen der Umgebung befähigen die Schüler schon nach wenigen Unterrichtsstunden komplexe Aufgabenstellungen zu bearbeiten.⁵⁴

Eine Entwicklungsumgebung bietet die Gelegenheit, das Erlernen von Programmierung stufenweise zu strukturieren und womöglich Eingrenzungen vorzunehmen, wenn weiterführende Programmierungserfahrungen nicht nötig sind.⁵⁵ Die Vorgehensweise folgt einem modernen Vorgehen, da in ihr mehrere Faktoren wie Multimedia und Objektorientiertheit verankert sind.⁵⁶

Ein weiteres Merkmal ist der große Kreativitätsfaktor, der sich für die Schülerinnen und Schüler ergibt. Kreativität meint nicht

irgendeine besondere Fähigkeit, sondern größeres Wissen (in der Form praktisch angewandter Kenntnisse) und der Wille, sich dieses anzueignen und zu benutzen.⁵⁷

⁵³ Vgl. Schubert/Schwill 2011, S.294.

⁵⁴ Vgl. Kiesmüller, Ulrich: Prozessbegleitende, automatisierte Identifizierung der Problemlösestrategien von Lernenden beim Lösen algorithmischer Probleme In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber. Bonn: Köllen2009, S.170.

⁵⁵ Vgl. Schubert/Schwill 2011, S.294.

⁵⁶ Vgl. Schubert/Schwill 2011, S.294.

⁵⁷ Boden, M. A.: Die Flügel des Geistes: Kreativität und künstliche Intelligenz. München: DTV1995, S.39.

Vgl. Romeike 2007, S. 85.

Als Beispiel für eine kreative Entwicklungsumgebung nennt R. Romeike "Scratch".

Scratch ermöglicht das intuitive Erstellen von interaktiven Programmen und Animationen durch Exploration und Experimentieren, eine visuelle Darstellung der Strukturen und Programmierbausteine, Inspiration, schrittweises Entwickeln und das problemlose Veröffentlichen und Präsentieren der Ergebnisse auf einer Plattform im Internet [...].⁵⁸

Auf Scratch soll im weiteren Verlauf der Arbeit noch besondere Aufmerksamkeit gelegt werden.

Abschließend muss noch erwähnt werden, dass eine ebensolche Entwicklungsumgebung eine "vollwertige" Programmiersprache nicht ersetzen kann, da in ihr nicht alle unterrichtsrelevanten Konzepte einspielen. Das heißt Datenstrukturen wie Bäume, Listen oder das Deklarieren von Variablen finden in diesem Umfang keine Relevanz. Um diesem gerecht zu werden, ist ein späterer Schwenk zu einer "richtigen" Programmiersprache unausweichlich.⁵⁹

3.2.1 Lego Mindstorms

Eine Möglichkeit der Umsetzung bietet der Baukasten von Lego Mindstorms NXT 2.0⁶⁰. Hiermit können Roboter und andere autonome, interaktive Systeme selbst gebaut und programmiert werden. Es gibt eine zentrale Steuereinheit, NXT genannt, die mit dem Computer verbunden werden kann, um Programme darauf zu laden. Die Verbindung kann über Bluetooth sowie USB erfolgen. An dieser Steuereinheit befinden sich vier Eingänge für Sensoren und drei Ausgänge für Motoren, mit denen die Komponenten über Kabel verbunden werden können. Außerdem besitzt der NXT eine LCD-Anzeige aus 100 mal 64 Pixeln, die auch über ein Programm angesteuert und verändert werden kann.⁶¹

⁵⁸ Schubert/Schwill 2011, S.371.

⁵⁹ Vgl. Schubert/Schwill 2011, S.294.

⁶⁰ Vgl. Homepage Lego Mindstorms. <http://mindstorms.lego.com/en-us/Default.aspx> (Stand: 27.07.2011).

⁶¹ Vgl. Valk, Laurens: LEGO® - ROBOTER. Bauen und programmieren mit LEGO® MINDSTORMS® NXT 2.0. Heidelberg dpunkt.Verlag2011, S.38.

Der Baukasten enthält eine große Anzahl an Technic-Bausteinen, aus denen der Roboter gebaut werden kann. Diese lassen sich direkt an den NXT anbauen, da dieser über Löcher für Steckverbindungen verfügt.⁶²

Des Weiteren enthält der Baukasten eine Software "NXT-G", mit der die Programme geschrieben werden können. Hierbei handelt es sich um eine grafische Programmierumgebung, in der der Programmierer ein Programm erstellt, indem er vorgegebene Programmierblöcke hintereinander reiht und anschließend der Steuereinheit übergibt. Es gibt Blöcke für Bewegungen (Ansteuern von Motoren), Tonausgabe, Anzeigen auf der Steuereinheit, Warten, Wiederholen und Entscheidungen. Auch besteht die Möglichkeit eigene Blöcke zu entwerfen. Die Blöcke können auch parallel angeordnet werden, wenn der Roboter Anweisungen gleichzeitig ausführen soll.⁶³

Zwischen den einzelnen Blöcken können logische, numerische oder textliche Datenleitungen gesetzt werden, mit denen Daten weitergegeben werden können. Jeder Block besitzt ein Datenhub, so dass auch mehrere Datenleitungen parallel laufen können.⁶⁴

Man kann auch andere Programmiersprachen nutzen wie zum Beispiel NXC, was für Not eXactly C steht. Wie der Name schon sagt, ist diese Programmiersprache der Sprache C sehr ähnlich, aber wesentlich einfacher.⁶⁵ Diese Sprache ist noch sehr jung⁶⁶ und wurde genauso wie die Entwicklungsumgebung BricxCC von John Hansons erfunden. Ein Vorteil hierbei ist, dass die Sprache, ähnlich wie bei NXT-G, auf die Hardware ausgerichtet wurde und somit keine weitere Firmware benötigt wird.⁶⁷

Entscheidend für das autonome Verhalten des Roboters sind die zugehörigen Sensoren, die von der zentralen Steuereinheit über ein geladenes Programm angesteuert werden können. Es gibt Berührungs-, Farb-, Ultraschall- und Rotationssensoren, wobei die Rotationssensoren direkt im Motor integriert sind, um die Anzahl der Umdrehungen mes-

⁶² Vgl. Valk 2011 S.3.

⁶³ Vgl. Valk 2011, S.31-66.

⁶⁴ Vgl. Valk 2011, S.158ff.

⁶⁵ Vgl. Braun, Daniel: Roboter programmieren mit NXC für LEGO® MINDSTORMS® NXT. Heidelberg/München/Landsberg/Frechen/Hamburg mitp2009, S.16.

⁶⁶ Vgl. Braun 2011 , S.16.

NXC wurde 2007 von Hansons entwickelt.

⁶⁷ Vgl. Braun 2011, S.17.

sen zu können. Mittels dieser können die Roboter auf die Außenwelt reagieren. Es lassen sich auch noch weitere Sensoren einbeziehen wie Geräusch- oder Temperatursensoren, die aber nicht standardmäßig enthalten sind. Außerdem gibt es noch HiTechnic-Sensoren wie, Kompass-, Beschleunigungs-, Infrarot-, Gyrosensoren, zur genauesten Ermittlung von Drehungen, sowie einen Sensor zur Infrarotkommunikation.⁶⁸

Wie lässt sich dieses System für eine Haussteuerung nutzen?

3.2.1.1 Simulation einer Haussteuerung mit Lego Mindstorms

Wichtig für eine Hausteuerung ist die Verwendung von Sensoren und Aktoren, sowie eine geeignete Programmierumgebung. Dabei sollen die Sensoren einzeln angesprochen werden können, deren Informationen verarbeitet werden und im Anschluss Befehle an die Aktoren gegeben werden können. Hierbei sollte das Verarbeiten von Text, Zahlen und Wahrheitswerten möglich sein.

Im Großen und Ganzen erfüllt Lego Mindstorms diese Voraussetzungen, doch es lassen sich noch einige Schwächen feststellen. Im Bereich der Aktoren kann nur auf Motoren zurückgegriffen werden. Lampen, Summer unter anderem werden nicht mitgeliefert. Das heißt, man kann nur motorgesteuerte Geräte im Haus ansprechen, keine Lichter oder Klingeln bedienen.

Des Weiteren fehlt eine Umgebung, in der Komponenten zum Einsatz kommen. Hier bräuchte man zusätzlich ein Miniaturhaus oder Ähnliches, in dem die Sensoren und Aktoren installiert werden können. Die Programmierumgebung NXT-G eignet sich sehr gut, da mit ihr die wichtigsten Programmstrukturen wie einfache Anweisungen, Schleifen und Verzweigungen umgesetzt werden können. Auch können Variablen und Konstanten benutzt werden, um Sensorwerte festzulegen, zu verändern oder zu vergleichen.⁶⁹

Alles in allem besteht die Möglichkeit, eine Haussteuerung mit kleinen Einschränkungen zu simulieren. Doch wie sieht es mit einer Umsetzung in der Schule aus?

⁶⁸ Vgl. Homepage Lego Mindstorms Education. <http://www.nxt-in-der-schule.de/lego-mindstorms-education-nxt-system/nxt-hardware/sensoren/hitechnic-sensoren> (Stand: 27.07.2011).

⁶⁹ Vgl. Braun 2011 S.49ff.

3.2.1.2 Beurteilung für den schulischen Kontext

Zuerst einmal ist zu sagen, dass Lego Mindstorms bereits an vielen Schulen eingesetzt wird.

Das mittlerweile zum Kultstatus avancierte System LEGO MINDSTORMS für Schulen und ROBOLAB wurden 1998 erstmals lanciert und hilft seitdem unzähligen Schülern Naturwissenschaften, Technik, Konstruktion und Mathematik durch selbständiges Arbeiten, motivierende Baukästen, Programmiersoftware und Lernkonzepte besser zu begreifen.⁷⁰

Die Schüler erhalten beim Bauen und Programmieren der Roboter kontinuierlich neue Kenntnisse, wenden diese sofort in der Praxis an und entwickeln eigene neue Ideen. Hierbei wird das Problemlösen zur zentralen Kompetenz, welche von den Kompetenzen "Wissen erwerben", "Kooperationsfähigkeit", "Kommunikationsfähigkeit" und "Kreatives Denken" gespeist werden.⁷¹

Die Planungen zum Unterricht mit LEGO Mindstorms beziehen sich auf den Informatikunterricht der Sekundarstufe I und II. In vielen Bundesländern wird in den Lehrplänen das Thema "Messen, Steuern, Regeln" empfohlen, es bestehen insofern direkte Bezüge zur unterrichtlichen Auseinandersetzung mit Computern.⁷²

Damit ist das Arbeiten mit dem System an sich legitimiert. Jetzt muss man sich fragen, ob dies auch für eine Simulation einer Haussteuerung gilt.

Da bei einer Haussteuerung genauso wie beim Programmieren der Roboter das Behandeln technischer Systeme mit Ein- und Ausgabe sowie Verarbeitung im Vordergrund stehen, lassen sich hier deutliche Parallelen erkennen.

Allerdings muss man sich fragen, ob die Anschaffung des Systems für die Simulation einer Haussteuerung von Schulen zu tragen ist, wenn man bedenkt, dass die Technic-Bausteine gar nicht zum Einsatz kommen. Preislich liegt ein Satz, an dem maximal drei Schüler arbeiten können, bei circa 300 Euro.⁷³ Hinzu kommt dann noch die Schullizenzen

⁷⁰ Homepage Lego Mindstorms Education. <http://www.nxt-in-der-schule.de/> (Stand: 27.07.2011).

⁷¹ Vgl. Homepage Lego Mindstorms Education. <http://www.nxt-in-der-schule.de/lego-mindstorms-education-nxt-system> (Stand 27.07.2011).

⁷² Winkel, Jens: Gedankenstürme – LEGO-Mindstorms im Unterricht. Informationen über die Einsatzmöglichkeiten von LEGO Mindstorms im Unterricht und Erfahrungen aus der Praxis. (28/05/2003). <http://www.lehrer-online.de/mindstorms.php> (Stand: 27.07.2011).

⁷³ Vgl. Homepage Lego Mindstorms Education. <http://shop.lego-in-der-schule.de/> (Stand: 27.07.2011).

für die Software NXT-G, die bei 330 Euro liegt. So müssen mindestens 3500 Euro für einen Klassensatz bezahlt werden.

Es scheint also nur sinnvoll, eine Haussteuerung mit Lego Mindstorms umzusetzen, wenn dieses System bereits an der Schule vorhanden ist.

3.2.2 Das PUMA-System

Zum Kontext Haussteuerung hat Strecker (2009) eine in ein Puppenhaus der Firma playmobil® integrierte Steuerung entwickelt. Das PuMa-System⁷⁴ steht hierbei für **Pup-**penhaus-Hausautomation. Dabei sind Sensoren und Aktoren der Firma Conrad verwendet worden. Die Sensoren beinhalten Schalter, Licht- und Temperatursensoren sowie Druck- und Berührsensoren. Als Aktoren fungieren Summer, Glühbirnen, LEDs und Sollarmotoren. Mittels eines velleman-K8055Board können die Sensoren und Aktoren über einen Computer durch USB angesprochen werden. An dieser Platine befinden sich zwei analoge und fünf digitale Eingänge beziehungsweise acht digitale und zwei analoge Ausgänge.⁷⁵

Mit Hilfe einer dynamischen Bibliothek, die in Form einer DLL-Datei etwa durch die Programmiersprache Delphi eingebunden werden kann, können Ausgangsdaten gesetzt, gelöscht und Eingangsdaten abgefragt werden.⁷⁶

Des Weiteren verfügt das System über zwei Platinen mit mehreren Anschlussmöglichkeiten für Sensoren und Aktoren. Die Platinen sind so konzipiert, dass Spannungsver-

⁷⁴ Vgl. Strecker, Kerstin Maike: Informatik für Alle – Wie viel Programmierung braucht der Mensch. Dissertation zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades. Göttingen: 2009, S.85ff.

⁷⁵ Vgl. Strecker 2009, S.86f.

⁷⁶ Vgl. Strecker 2009. S.87.

stärkungen für kräftigere Motoren integriert sind. Auch werden die Sensorwerte unter Verwendung eines Potentiometers⁷⁷ in einheitliche digitale Signale umgewandelt.⁷⁸

Die Konfiguration kann über verschiedene Oberflächen erreicht werden, von denen aber zur zwei konkret von Strecker umgesetzt wurden. Zu den Vorschlägen zählen Schaltnetze und Schaltwerke, eine Mealy-Maschine, eine graphische imperative Programmiersprache, UML-Aktivitätsdiagramm sowie die Angabe von Delphi-Code.⁷⁹

In den in Delphi programmierten Hardwaresimulator "HASI" von Eckart Modrow wurde die velleman-DLL integriert, mit deren Hilfe die Eingaben und Anzeigen der Schaltnetze und Schaltwerke auch von außen kommen können und nach außen gehen.⁸⁰

Die zweite Umsetzung wurde mit Delphi realisiert, indem ein lauffähiges Delphi-Programm mit integrierter DLL-Datei sowie einer Verbindung zur Platine besteht. Hier können Konfigurationen durch Angabe von Alternativen und Zuweisungen von Variablen vorgenommen werden. Integriert wurde außerdem, dass alle 100 ms die Werte der Eingänge automatisch in Variablen gespeichert wurden.⁸¹

3.2.2.1 Beurteilung für den schulischen Kontext

Die Lernumgebung PUMA stellt bezogen auf den Sachzusammenhang Haussteuerung eine gelungene Umsetzung dar. Strecker(2009) hat eine Unterrichtsreihe mit PUMA in einer achten Klasse durchgeführt und diese auch von den Schülern evaluieren lassen.

Dabei ist herausgekommen, dass die Unterrichtsreihe durchweg „gut“ angekommen ist und den Schülern Spaß bereitet hat. Auch wurde ermittelt, welche Teilbereiche der Einheit am besten gefallen haben. Hier stellte sich heraus, dass gerade die Bastarbeiten an

⁷⁷ "Ein Potentiometer besteht aus einem speziellen Widerstand, an dem ein sogenannter Schleifer angebracht ist, der sich durch die Drehung entlang der leitenden Oberfläche bewegt. Dadurch vergrößert oder verkleinert er auch den Widerstandswert und somit die Spannung, die am Potentiometer anliegt."

Aus: Odendahl, Manuel/ Finn, Julian/ Wenger, Alex: Arduino. Physical Computing für Bastler, Designer & Geeks. 2. Auflage. Köln: O'Reilly Verlag2010. S.52.

⁷⁸ Vgl. Strecker 2009, S. 87f.

⁷⁹ Vgl. Strecker 2009 , S. 99f.

⁸⁰ Strecker nimmt hier Bezug auf Modrow, Eckart: Technische Informatik mit Delphi. Norderstedt: emu-online Verlag2004.

Vgl. Strecker 2009, S.100.

⁸¹ Vgl. Strecker 2009, S. 100.

dem Puppenhaus am häufigsten (knapp zwei Dritteln der Schüler) angegeben wurden. Die anderen Kategorien Hardware, Erfolgserlebnisse haben und Programmierung wurden ungefähr gleich oft genannt, nahmen aber insgesamt nur ein Drittel der Schülerstimmen ein.⁸²

An dieser Einheit lässt sich festhalten, dass Schüler der Sekundarstufe I ein großes Interesse am Erstellen eigener Produkte haben, die einen haptischen Charakter aufweisen. Es ist dann eben nicht nur eine Bildschirmausgabe, die dem Schüler einen erfolgreichen Algorithmus bestätigt, sondern das Ergebnis ist fassbar und real. Auch das Präsentieren der eigenen Leistung gegenüber den Mitschülern ist als großer Motivationsfaktor zu sehen.⁸³

Zur Umsetzung dieser Reihe sind einige Anschaffungen zu tätigen, die einen nicht zu vernachlässigen Kostenfaktor bilden. Das verwendete Playmobil-Haus kostet Stück 100 Euro⁸⁴, wobei eines nur für eine kleine Gruppe von Schülern zu nutzen ist. Hier könnte man vielleicht auch eine kostengünstigere Variante finden. Außerdem müssen die Platinen, Sensoren und Aktoren vorhanden sein, um die man aber bei einer Haussteuerung nicht herumkommt.

Ein weiterer Punkt ist die noch nicht eingesetzte grafische Programmierumgebung⁸⁵, welche den Schülern einen gewissen Halt beziehungsweise Struktur gibt. Darauf soll in weiteren Vorschlägen eingegangen werden.

3.2.3 Scratch

Scratch ist eine Programmierumgebung, in der über eine graphische Oberfläche programmiert werden kann. Scratch wurde am MIT Media Laboratory⁸⁶ innerhalb einer kleinen Forschungsgruppe, der Lifelong Kindergarten Group, entwickelt und ist im Frühjahr 2007 gestartet. Im Folgenden beziehe ich mich auf die Version 1.4.⁸⁷

⁸² Vgl. Strecker 2009, S.92.

⁸³ Vgl. Strecker 2009, S.92.

⁸⁴ Vgl. Homepage mytoys. <http://www.mytoys.de/PLAYMOBIL-PLAYMOBIL-4279-Neues-Wohnhaus/Bausteine/Kleinkindspielzeug/KID/de-mt.to.ca02.29.08/1722332> (Stand: 27.07.11).

⁸⁵ Vgl. Strecker (2009) setzte diese in weiterführenden Einheiten ein, etwa S.98ff.

⁸⁶ Fakultät der Universität Massachusetts, Institute of Technology.

⁸⁷ Vgl. Homepage Scratch. http://info.scratch.mit.edu/Scratch_1.4_Download (Stand: 27.07.2011).

Die Programmoberfläche ist in drei Teile unterteilt. Auf der linken Seite befinden sich vorgegebene Programmstrukturen, die von Steuerungsstrukturen wie Anweisung oder Schleife, mathematische Operatoren, Variablen, Fühloperationen wie Maustaste gedrückt, über Bewegungsoperationen, grafische Animationen sowie Geräusche reichen. Die Grafiken und Geräusche können aus einer mitgelieferten Bibliothek entnommen werden oder aber selbst gezeichnet beziehungsweise aufgenommen werden. Natürlich können auch importierte Medien genutzt werden.

Im mittleren Teil kann der Programmierer per „drag & drop-Verfahren“ die Programmstrukturen der linken Seite zu einem Script zusammenfügen. Scratch folgt dem objektorientierten Ansatz, da ein Script immer zu einem Objekt gehört, welches dann zur Laufzeit die im Script beschriebenen Operationen durchführt. Mehrere Objekte können miteinander agieren, indem Nachrichten zwischen ihnen hin und her gesendet werden. Ein Objekt kann hier eine beliebige Grafik sein, die aus der Bibliothek entnommen oder selbst gezeichnet worden ist.

Die graphische Umsetzung des Scripts wird auf der "Bühne", die dem rechten Teil der Programmoberfläche entspricht, visualisiert. Die "Bühne" ist zunächst eine weiße Fläche, die aber mit einem Hintergrund versehen werden kann. Hinter ihr befindet sich ein nicht sichtbares Koordinatensystem, so dass jeder beliebige Pixel angesprochen werden kann. Fährt man mit der Maus über die Fläche, werden die Koordinaten des Mauszeigers in der unteren rechten Ecke des Bildschirms angezeigt.

Es zielt, im Gegensatz zu herkömmlichen Programmiersprachen, darauf ab, das technische Know-How in den Hintergrund zu stellen. In Scratch können Programmierbausteine durch einfaches Klicken zusammengestellt werden. Dies funktioniert ähnlich wie bei einem Baukasten, wobei nur Bausteine zusammengefügt werden können, die einen syntaktischen Sinn ergeben. Somit können Syntaxfehler ausgeschlossen werden, und der Programmierer kann sich auf die Semantik konzentrieren.

Weiterhin werden die einzelnen Datentypen durch unterschiedliche Formen repräsentiert, so dass auch hier möglichen Fehlanpassungen vorgebeugt wird. Mit Grafiken, Sounds, Musik und Animationen steht ein weites Feld der Möglichkeiten offen, in dem experimentiert werden kann. Ein weiterer Vorteil dieser Programmierumgebung ist eine

große Menge von vorhandenen Projekten, die online zum Austausch bereit stehen. Der Nutzer kann diese Projekte herunterladen und mit ihnen weiterarbeiten oder auch nur Teile herausgreifen und in einem eigenen Projekt umsetzen. Außerdem kann er selbst auch sein fertiges Projekt für die Allgemeinheit zur Verfügung stellen.

3.2.3.1 Simulation einer Haussteuerung mit Scratch

Wie kann man mit Scratch eine Haussteuerung simulieren?

Dazu bietet der Einzelhandel für Elektroartikel "SparkFun Electronics" das sogenannte Picoboard⁸⁸ an. Hierbei handelt es sich um eine Platine, die über ein MiniUSB Kabel mit dem Rechner verbunden werden kann. Diese Platine beinhaltet einen Licht- und Geräuschsensor sowie einen Knopf und einen Regler. Außerdem sind vier Eingänge eingebaut, die mit weiteren Kabeln elektrische Widerstände messen können. Allerdings kann man mit dem Picoboard keine Ausgaben in Form von Motorensteuerung produzieren. Ausgaben können entweder mit Hilfe einer grafischen Ausgabe simuliert werden oder mittels des LEGO Education WeDo Robotics Kit⁸⁹ realisiert werden.

Eine weitere Möglichkeit zur Ausgabe bietet das Arduino, ein Microcontroller Board, welches auch über USB mit dem PC verbunden werden kann. Es beinhaltet 14 digitale Ein- und Ausgangspins sowie sechs analoge Eingänge. Mit diesem Board lassen sich dann Motoren ansteuern. Auch können Picoboard und Arduino mit Scratch kombiniert werden, so dass eine Haussteuerung in einer geeigneten Miniwelt⁹⁰ installiert werden kann.

3.2.3.2 Beurteilung für den schulischen Kontext

Losgelöst von dem Gedanken der Haussteuerung bietet die Arbeit mit Scratch innerhalb des Informatikunterrichts noch weitere Vorteile:

⁸⁸ Vgl. Homepage sparkfun. <http://www.sparkfun.com/products/10311> (Stand: 27.07.2011).

⁸⁹ Vgl. Homepage Lego Education.
http://www.legoeducation.us/eng/product/lego_education_wedo_robots_construction_set/2096
(Stand: 27.07.2011).

⁹⁰ Eine in diesem Zusammenhang als geeignet geachtete Miniwelt stellt beispielsweise Strecker (2009) vor.

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). In developing Scratch, we put high priority on simplicity, sometimes even sacrificing functionality for understandability.⁹¹

Gerade der hohe Wert, den die Entwickler auf die Einfachheit legen, rechtfertigt Scratch als Einstiegsprogrammierumgebung für die Schule. Aufbauend darauf ermöglicht Scratch facettenreiche Möglichkeiten für Projekte, den Schülern sind kaum Grenzen in der Kreativität bei der Umsetzung von Problemlösungen gesetzt. Für den Mittelstufenbereich ist die Komplexität angemessen, da sich einerseits die Bildungsstandards der GI abdecken lassen und andererseits die kognitiven Voraussetzungen der Schüler in der 9. Klasse zur Nutzung des Programms ausreichen.

Romeike (2010) betont die positiven Aspekte der Nutzung von Scratch gerade im Anfangsunterricht. Insbesondere stellt er die "Intuitivität, Verständlichkeit, Ideen-Anregung und Vielfalt" als förderliche Wirkung heraus.⁹²

Die Programmierumgebung Scratch unterscheidet sich von anderen Programmiersprachen vor allem durch folgende Punkte:

1. Early programming languages were too difficult to use, and many children simply couldn't master the syntax of programming;
2. Programming was often introduced with activities (such as generating lists of prime numbers and making simple line drawings) that were not connected to young people's interests or experiences;
3. Programming was often introduced in contexts where no one could provide guidance when things went wrong – or encourage deeper explorations when things went right⁹³

Besonders hervorzuheben ist sicherlich die Vermeidung von kontextlosen mathematischen Problemstellungen, die selten motivationale Anregungen bei den Schülern hervorrufen. Durch die direkte visuelle Darstellung der Programme kann man sich daher Problemen zuwenden, die die Schüler mehr interessieren, etwa kontextbezogene Fragestellungen mithilfe von Scratch realisieren.

⁹¹ Homepage Scratch. <http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/docs/Programming-with-Scratch.pdf> (Stand: 27.07.2011).

⁹² Romeike, Ralf: "Das bessere Werkzeug" - Anmerkungen zur Diskussion Etoys vs. Scratch. In: LOG IN Zeitschrift für informatische Bildung und Computer in der Schule. 163/164. Berlin: Log-In-Verlag2010 S.43-45.

⁹³ Resnick, Mitchel (u.a.): Scratch: Programming for all. <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Stand: 27.07.2011).

As Scratchers program and share interactive projects, they learn important mathematical and computational concepts, as well as how to think creatively, reason systematically, and work collaboratively: all essential skills for the 21st century⁹⁴

Insgesamt lassen sich Scratch mit seinem Bastelcharakter, dem hohen Bedeutungsgehalt für Schüler und mit sozialen Aspekten drei positive Attribute zuweisen.

Der Bastelcharakter geht aus der Arbeitsfläche hervor, die ähnlich wie ein Desktop aufgebaut ist. Hier können Programmelemente beliebig zusammengesteckt werden oder auch einfach nur inaktiv und ungenutzt auf der Arbeitsfläche liegen und vielleicht später noch verwendet werden. Die Entwickler von Scratch vergleichen dies mit einem Legobaukasten. Durch das Zusammensetzen der Bausteine und das anschließende Testen des Programmteils kommen dem Schüler neue Ideen, die erst nicht bedacht worden sind.

Der hohe Bedeutungsgehalt für Schüler liegt in der Vielfältigkeit und der Personalisierung. Mit Scratch können Geschichten, Spiele, Animation oder Simulationen durchgeführt werden. Die starke Personalisierung ist vor allem durch den Import von eigenen Bildern oder Audiodateien sichergestellt.

Auch soziale Aspekte werden berücksichtigt, sobald die Projekte auf die Scratch Website hochgeladen werden beziehungsweise andere Projekte dort betrachtet werden. Die Schüler können so miteinander in Kontakt treten, sich andere Arbeiten/Programme ansehen und sich gegenseitig ein Feedback geben.

I like Scratch better than blogs or social networking sites like facebook because we're creating interesting games and projects that are fun to play, watch, and download. I don't like to just talk to other people online, I like to talk about something creative and new.⁹⁵

Die Projekte können mit einem javagestützen Player direkt abgespielt werden. Kommentierungen und Bewertungen können vorgenommen werden und die Programme lassen sich auch als Scratchdatei herunterladen. Diese daraus entstehende Community agiert weltweit und die Scratchsoftware ist bereits in 40 Sprachen übersetzt. Die Präsentation der eigenen Programme in die ganze Welt gibt einen zusätzlichen Anreiz, der Wettbewerbscharakter bietet eine zusätzliche Motivation. Das Weiterverwenden von

⁹⁴ Resnick, Mitchel (u.a.): Scratch: Programming for all.

<http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Stand: 27.07.2011).

⁹⁵ Aussage eines 13-jährigen Mädchen aus Kalifornien. Gefunden auf:

<http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Stand: 27.07.2011).

Programmstrukturen wird vom Verfasser eher mit Stolz gesehen, als dass es Verärgerung aufgrund von Ideenübernahme auslösen würde.⁹⁶

4 Entwurf einer Unterrichtsreihe

Klafki (1970) geht davon aus, dass didaktische Überlegungen nicht bloß die Methoden, sondern auch die sachlichen Inhalte betreffen sollten. Somit muss zunächst ein Überblick über die fachlichen Inhalte gegeben werden, damit daran anschließend die methodischen Überlegungen folgen können.⁹⁷

Im Folgenden wird eine Unterrichtsreihe aus dem Bereich „Informatik im Kontext“ vorgestellt; Thematisch widmet sich diese schwerpunktmäßig der Haussteuerung.⁹⁸

Nach einer kurzen Vorbemerkung zu den Rahmenbedingungen soll zunächst eine fachliche Analyse oder auch Sachanalyse, wie Klafki (1970) diese nennt, erfolgen, in welcher die wesentlichen fachlichen Inhalte wissenschaftlich fundiert dargestellt werden. Daran anschließend erfolgt die methodische Analyse, hier didaktische Analyse genannt, in der die didaktische Reduktion, die Stundenübersichten sowie Methoden-, Lern- und Motivationsziele abgesteckt werden.

4.1 Bedingungsanalyse

Die Unterrichtsreihe ist für eine neunte Klasse eines Gymnasiums konzipiert und auf acht Doppelstunden ausgelegt. Die Unterrichtsreihe kann als Einführung in die Informatik genutzt werden oder aber auch im Anschluss an vorangegangene Reihen, da gerade das erforderliche Vorwissen im Bereich der Informatik sehr begrenzt gehalten wurde.

Die Schüler sollten vor Beginn der Unterrichtsreihe in der Lage sein, die Computer zu nutzen, das heißt mit den installierten Systemen vertraut zu sein. Dies beinhaltet auch, eine sinnvolle Recherche mittels des Internets durchführen zu können. Die Lernumgebung Scratch muss den Schülern nicht bekannt sein, es schadet aber auch nicht, wenn der eine oder andere bereits auf Vorerfahrungen mit der Lernumgebung zurückgreifen kann. Im Bereich der Methodenkompetenz sollten die Schüler schon Erfahrungen mit

⁹⁶ Vgl. <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Stand: 27.07.2011).

⁹⁷ Vgl. Klafki, Wolfgang: Studien zur Bildungstheorie und Didaktik. Weinheim: Beltz1970, S.126ff.

⁹⁸ Materialien zur Unterrichtsreihe befinden sich im Anhang dieser Arbeit.

Gruppenarbeiten gemacht haben. Ebenso sollten die Schüler über Wissen bezüglich von Präsentationen sowie Feedback verfügen.

Der Kurs sollte eine Schülerstärke von 20 Schülern nicht überschreiten, da die Unterrichtseinheit sehr stark auf Gruppen- beziehungsweise Projektarbeit ausgerichtet ist, deren Ergebnisse dann im Plenum besprochen werden sollen. Dieses Vorstellen der Ergebnisse ist zeitlich umfangreich, aber sinnvoll, da den Schülern hier die Möglichkeit gegeben wird, ihre Präsentationskompetenz auszubauen; vor allem wird ihnen ein breiteres Spektrum von Lösungsmöglichkeiten vorgeführt, was die Variabilität der Lösungszugänge deutlich macht. Somit können die Schüler lernen, dass es zu einer Problemstellung unterschiedliche Herangehensweisen gibt, sie können weiterhin die Vor- und Nachteile abwägen.

Diesbezüglich ist es leicht einzusehen, dass die Schüler höchstens zu zweit an einem Computer arbeiten können. Die Unterrichtsreihe soll insbesondere die Eigenaktivität der Schüler fördern und deren Kreativität ausnutzen, anregen und dadurch auch Möglichkeiten zu variablen Lösungszugängen eröffnen. Das Arbeiten mit dem Programm Scratch sollte daher zunächst für jeden Schüler einzeln ermöglicht werden. Unter der Voraussetzung einer großen Lerngruppe und der Kapazität eines schulischen Computerraums ist jedoch auch eine Verteilung von zwei Schülern auf einen Rechner denkbar. Diese können sich in einem zeitlich angemessenen Rahmen mit der Programmierarbeit abwechseln.

Die Computer müssen alle über einen Internetzugang verfügen und die Lernumgebung Scratch muss vorinstalliert und betriebsbereit sein, sodass hierfür kein Zeitverlust entsteht, etwa durch Probleme des Aufrufens des Programms. Das Gleiche gilt für die Treiber für Adruino und das Picoboard.

Für die weitere Arbeit in den Projekten müssen mindestens fünf Miniwelten, sowie je fünf Adruino- und Picobards vorhanden sein, sodass die Schüler in Vierergruppen arbeiten können. Es empfiehlt sich, im Vorfeld einen Testlauf mit den verschiedenen Einheiten durchzuführen, um einen reibungslosen Ablauf besser abzusichern.

Welche Miniwelt man wählt, ist relativ frei zu entscheiden. Empfehlenswert ist der Gebrauch einer Miniwelt, die sich eignet, ein echtes Haus möglichst nah zu modellieren. In anderen Arbeitsgruppen hat sich das Puppenhaus von der Firma playmobil bewährt.⁹⁹ Weitere Hardwarekomponenten wie eine Auswahl an Motoren, Summer oder Lampen sowie zu nutzende Sensoren stehen jedem Schüler zur Verfügung. Ebenso wie auch in dem Bereich der Software muss die Lehrperson die Hardwarekomponenten prüfen, um den Schülern ein effizientes Arbeiten ermöglichen zu können.

4.2 Sachanalyse

Innerhalb der Sachanalyse werden nun die fachlichen Inhalte erläutert, welche der vorgestellten Unterrichtsreihe zugrunde liegen. Hierbei wird zunächst die informatische Perspektive sowie die technischen Sichtweisen auf die Inhalte erläutert, welche dann mit Hilfe geeigneter didaktischer Reduktion für den Unterricht handhabbar gemacht werden.

4.2.1 Systemsteuerung

Für den fachlichen Inhalt der Unterrichtsreihe ist vor allem die Betrachtung von Steuerungssystemen relevant, da der Schwerpunkt der Sequenz auf der Erstellung einer Haussteuerung liegt, welche eine spezielle Form von Steuerungssystemen, eine Speicherprogrammierbare Steuerung, ist.

Zunächst sind die Begriffe System und Steuerung von Prozessen näher zu definieren. Unter einem System wird nach Wedekind (1998) allgemein folgende Definition verstanden:

- (a) eine Menge von Elementen (Systembestandteilen), die
- (b) durch bestimmte Ordnungsbeziehungen miteinander verbunden und
- (c) durch klar definierte Grenzen von ihrer Umwelt geschieden sind.¹⁰⁰

⁹⁹ Vgl. Strecker 2009, S.86.

¹⁰⁰ Zitiert nach Hubwieser, Peter: Didaktik der Informatik. 3. Auflage. Berlin / Heidelberg: Springer2007, S.86.

Exemplarisch kann das System einer Haussteuerung genannt werden, zu welchem die Steuerung und die Peripheriegeräte, also Sensoren und Aktoren, gehören. Die Steuerung von Prozessen ist nach der Deutschen Industrienorm definiert.

Nach DIN 19226 versteht man unter der Steuerung eines Prozesses den Vorgang, bei dem durch Messung der Prozesszustände über bestimmte Gesetzmäßigkeiten Stellwerte zur Beeinflussung des Prozesses erzeugt werden. Die Gesetzmäßigkeiten werden in einem Programm formuliert, das im Steuerungsrechner, d.h. im Allgemeinen in einer Speicherprogrammierbaren Steuerung (SPS), abläuft.¹⁰¹

Es gibt verschiedene Steuerungsarten. Pritschow (2006) nennt hier mit der Verknüpfungssteuerung und der Ablaufsteuerung zwei verschiedene Formen von Steuerungen.¹⁰²

Bei der Verknüpfungssteuerung werden Ausgangssignale einer Verknüpfung von Eingangssignalen zugeordnet. Die Eingangssignale werden über Grundfunktionsglieder wie logische Funktionen, Zeit- und Speicherglieder sowie Zähler verarbeitet.¹⁰³

Die Ablaufsteuerung beinhaltet eine Kette von hintereinander auszuführenden Prozessen, die entweder eine zeitlich- oder prozessbedingte Reihenfolge einnehmen.¹⁰⁴

Diese beiden Steuerungsarten können innerhalb einer Speicherprogrammierbaren Steuerung, einem digital arbeitenden elektronischen System, eingesetzt werden. Sie beschreiben die spezifischen Funktionen, die durch digitale oder analoge Eingangs- beziehungsweise Ausgangssignale Maschinen oder Prozesse steuern. Der interne Speicher der SPS enthält die anwenderorientierten Steuerungshinweise. Die SPS mit den angeschlossenen Peripheriegeräten bilden das SPS-System.¹⁰⁵

Die Systemsteuerung agiert nach dem EVA-Prinzip, was den Prozess in Signaleingabe, Signalverarbeitung und Signalausgabe gliedert.¹⁰⁶ Wie können wir die Signaleingabe

¹⁰¹ Seitz, Matthias: Speicherprogrammierbare Steuerungen. System- und Programmentwurf für die Fabrik- und Prozessautomatisierung, vertikale Integration. 2. Auflage. München: Hanser2008, S.13.

¹⁰² Vgl. Pritschow, Günter: Einführung in die Steuerungstechnik. München/Wien: Hanser2006, S.43.

¹⁰³ Vgl. Pritschow 2006, S.43f.

¹⁰⁴ Vgl. Pritschow 2006, S.44.

¹⁰⁵ Vgl. Grötsch, Eberhard: SPS. Speicherprogrammierbare Steuerungen als Bausteine verteilter Automatisierung. 5. Auflage. München: Oldenbourg Industrieverlag2004, S.18.

¹⁰⁶ Vgl. Pritschow 2006, S.41.

und Signalausgabe näher beschreiben? Dazu gehen wir auf die Funktionsweise von Sensoren und Aktoren ein und beschreiben sie exemplarisch detailliert.

4.2.2 Sensoren und Aktoren

„Sensoren nehmen Informationen über einen Zustand eines technischen Prozesses durch Messung einer physikalischen Größe auf.“¹⁰⁷ Bezuglich der Haussteuerung können solche relevanten Größen beispielsweise Temperatur, Zeit, Licht, Wärme, Klang, Bewegung und Druck betreffen. So gibt es zur Erfassung jeder physikalischen Größe einen entsprechenden Sensor.

Wie funktioniert beispielsweise ein Lichtsensor? Ein Lichtsensor kann Helligkeitswerte etwa durch eine Photodiode messen. Hier wird der fotoelektrische Effekt genutzt, der Lichtintensität durch verschieden geladenes Material in eine elektrische Spannung transformiert. Das Licht, welches aus Photonen besteht, löst durch die Bestrahlung die Elektronen. Durch das Zusammentreffen der verschiedenen geladenen Atomschichten entsteht eine Ladungsverschiebung. Diese kann gemessen werden und ergibt das Eingangssignal.¹⁰⁸

Wie wird das Eingangssignal bei einem Klangsensor gewonnen? Bei einem Klangsensor handelt es sich um ein kleines Mikrofon, welches unterschiedliche Lautstärken beziehungsweise Tonhöhen erkennt. Hierbei werden die Schallwellen in entsprechende Spannungsänderungen umgewandelt, die dann das Eingangssignal bilden.¹⁰⁹

Schließlich soll noch kurz die Funktionsweise eines Temperatursensors erläutert werden. Es gibt zwei verschiedene Arten von thermischen Mikrosensoren, berührende sowie berührungslose Sensoren. Temperaturveränderungen führen zu Polarisierungsänderungen, das heißt der elektrische Widerstand, welcher im Sensor gemessen wird, ändert sich mit der Temperatur.¹¹⁰

¹⁰⁷ Werner, Dieter: Taschenbuch der Informatik. München: Hanser Verlag 2007, S.650.

¹⁰⁸ Vgl. Berns, Karsten/Schmidt, Daniel: Programmierung mit Lego® Mindstorms® NXT. Heidelberg/London/New York: Springer Verlag 2010, S.53.

¹⁰⁹ Albrecht, Frank-Werner: Beschallungstechnik. Theorie und Praxis. 3. Auflage. Renningen-Malmsheim expert Verlag 1995, S.21.

Aktoren ermöglichen eine aktive Beeinflussung des technischen Prozesses.¹¹¹ Dies können beispielsweise Motoren, Lampen oder Summer sein. Im Kontext der Haussteuerung können Motoren beispielsweise dazu eingesetzt werden, Jalousien oder Markisen zu bewegen.

4.2.3 Struktogramme

Struktogramme dienen der Modellierung von Programmabläufen, insbesondere der Veranschaulichung von Abläufen durch Graphen.¹¹²

Ihr Zweck besteht darin, algorithmische Ablaufstrukturen graphisch in einer Weise darzustellen, die anders als Programmablaufpläne zu gedanklicher Planung und strukturierter Programmierung zwingt.¹¹³

Die durch die Deutsche Industrienorm (DIN66261) festgelegte Vereinheitlichung von Struktogrammen besteht aus vier Grundelementen der Darstellung von Sequenzen, If-Anweisungen, Fallunterscheidungen und Schleifen.¹¹⁴

Mittels Struktogrammen werden Probleme in Teilschritte aufgeschlüsselt und können so mittels der Grundelemente gelöst werden. Insgesamt entspricht die Vorgehensweise eines Struktogramms einem Top-Down Verfahren, also der Auflösung von großen und komplexen Strukturen und deren Zerlegungen in kleinere, einfacher beschaffene.¹¹⁵

4.2.4 Kontrollstrukturen

Kontrollstrukturen ermöglichen das Formulieren von flexiblen Programmen mit sogenannten bedingten Verzweigungen sowie Laufschleifen. Es sind folgende Anweisungstypen zu unterscheiden: Ausdrucksanweisungen, Verbundanweisungen, leere Anweisun-

¹¹⁰ Vgl. Völklein, Friedemann/Zetterer, Thomas: Praxiswissen Mikrotechnik. Grundlagen Technologien Anwendungen. 2. Auflage. Wiesbaden. Vieweg & Sohn Verlag2006, S.203.

¹¹¹ Vgl. Werner 2007, S.650.

¹¹² Vgl. Schubert/Schwill 2011, S.259.

¹¹³ Schubert/Schwill 2011, S.259.

¹¹⁴ Vgl. Schubert/Schwill 2011, S.259.

¹¹⁵ Vgl. Hering, Ekbert/Gutekunst, Jürgen/Dyllong, Ulrich: Handbuch der praktischen und technischen Informatik. 2. Auflage. Berlin/Heidelberg: Springer2000, S.210.

gen, if-Anweisungen, switch-Anweisungen, while-Anweisungen, do while-Anweisungen und for-Anweisungen.¹¹⁶

Für die Darstellung der für die Unterrichtsreihe notwendigen Inhalte werden im Folgenden die if-Anweisung sowie die while-Anweisung näher ausgeführt, da diese konkret im Unterricht benannt beziehungsweise behandelt werden.

Die if-Anweisung oder auch bedingte Verzweigung bietet sich immer dann an, wenn es zwei Möglichkeiten (Anweisungen) gibt, ein Programm fortzuführen. Welche Fortsetzung gewählt wird, hängt von einer Bedingung ab. Dabei wird die Bedingung zunächst ausgewertet und als Wahrheitswert interpretiert. Kommt wahr heraus, wird mit Bedingung eins ansonsten mit Bedingung zwei fortgesetzt.¹¹⁷

Eine while-Anweisung (while-Schleife) gliedert sich in einen Ausdruck und eine Anweisung. Bei dem Ausdruck handelt es sich um eine Bedingung, welche auch als Wahrheitswert interpretiert wird und zuerst geprüft wird. Solange die Bedingung wahr ist, wird der Schleifendurchlauf wiederholt. Ist der Ausdruck schon zu Beginn falsch, wird die Anweisung gar nicht ausgeführt.

4.3 Didaktische Analyse

Im Folgenden werden der Verlauf sowie die didaktischen Perspektiven der Unterrichtsreihe erörtert. Der Fokus liegt hier einerseits auf der Darstellung der Reihe in der Makro-/Mikroebene, andererseits auf den zugrundeliegenden Lernzielen und Bildungsstandards.

4.3.1 Didaktische Reduktion

Ein wichtiger Bestandteil der Unterrichtsplanung ist die didaktische Reduktion, welche die fachlich-komplexen Inhalte der Sachanalyse in schülergerechte Lerninhalte formt.

Die Behauptung Jerome S. Brunners, man könne die Grundstruktur jedes Gegenstandsbereichs prinzipiell jedem Kind auf jeder Altersstufe vermitteln, scheint den Gedanken einer

¹¹⁶ Vgl. Böttcher, Axel/Kneißel, Franz: Informatik für Ingenieure. Grundlagen und Programmierung in C. 2. Auflage. München: Oldenbourg Wissenschaftsverlag 2002, S.117.

¹¹⁷ Vgl. Böttcher 2002, S. 118.

didaktischen Transformation zu stürzen, in der ein Inhalt durch altersbezogene Darstellung faßbar wird.¹¹⁸

Gemäß dieser Forderung muss die Erfassung des fachlichen Inhalts bedingt durch die kognitiven Möglichkeiten der Schüler erreichbar sein. Allerdings gilt hier, wie auch Memmert (1995) bemerkt, dass keine simplen Vereinfachungen an die Schüler herangetragen werden, welche einen fachlich falschen Eindruck hinterlassen, sondern Inhalte über eine didaktische Entscheidung so reduziert werden, dass die wissenschaftlichen Grundsätze stets erhalten bleiben.¹¹⁹

Nach Aschersleben (1993) unterliegt die didaktische Transformation sechs verschiedenen Prinzipien: Eingrenzung, Vereinfachung, Verdichtung, Straffung, Veranschaulichung sowie Isolierung.¹²⁰ Diese sollen nun auf die Inhalte der Unterrichtsreihe angewendet werden.

Unter Eingrenzung wird die Reduktion des Umfangs des Unterrichtsgegenstandes gefasst, welche dann die Bearbeitung des fachlichen Inhalts in der zur Verfügung stehenden Zeit ermöglicht.¹²¹

Die hier vorgestellte Unterrichtsreihe ist generell sehr experimentell und schülerorientiert ausgerichtet, sodass die Reduktion des Umfangs des Unterrichtsgegenstands unumgänglich ist, da den Schülern viel Zeit für selbstorganisiertes Lernen eingeräumt wird. Somit können sich die Schüler beispielsweise die Lernumgebung Scratch selbst erschließen, allerdings wird ihnen durch den Lehrer eine kurze Einführung die Basisfunktionen von Scratch gegeben, was den Schülern den Einstieg in die eigenständige Arbeit erleichtert und somit auch Zeit erspart.

¹¹⁸ Memmert, Wolfgang: Didaktik in Grafiken und Tabellen. 5. Auflage. Bad Heilbrunn: Klinkhardt 1995, S.41.

¹¹⁹ Memmert (1995) nennt hier das Beispiel des Sonnensystems, welches von grundschülern über den Eindruck von kreisrunden Umlaufbahnen der Planeten zunächst fachlich falsch erlernt wird und sich somit ein falscher Eindruck aufgenommen wird, der später unbedingt korrigiert werden muss. Bei diesem Vorgang handelt es sich, so Memmert (1995) nicht um didaktische Reduktion. Vgl. Memmert 1995, S.40f.

¹²⁰ Aschersleben, Karl: Welche Bildung brauchen Schüler? Vom Umgang mit dem Unterrichtsstoff. Bad Heilbrunn: Klinkhardt 1993, S.91-106.

¹²¹ Aschersleben 1993, S.84.

Die Vereinfachung des Unterrichtsgegenstandes meint, dessen Schwierigkeiten zu umgehen, indem diese reduziert werden.¹²² In der Unterrichtsreihe finden sich vielerlei solche Vereinfachungen bezüglich des Unterrichtsgegenstands.

Das Einführen in das Programmieren wird den Schülern über die Lernumgebung Scratch ermöglicht. Wie schon Romeike (2008) bemerkt, ist der Einstieg in die Programmierung für Schüler sehr schwierig, weil komplex, und bedarf deswegen einer geeigneten didaktischen Reduktion, welche in diesem Fall mittels Scratch realisiert wird.¹²³ Dies hat den Vorteil, dass sich die Schüler, losgelöst von informatischen Grundgedanken, intuitiv mit Programmierung auseinandersetzen. Den Schülern werden über das Programm zentrale Programmier-/Kontrollstrukturen zur Verfügung gestellt, ohne dass diese im Vorhinein besprochen werden müssen.

Des Weiteren nutzen die Schüler Sensoren und Aktoren innerhalb der Projektarbeit, ohne in Gänze über den Aufbau und die informatisch-technischen Grundlagen derselben informiert zu sein. Dies ist jedoch für die Lernziele der Unterrichtsreihe nicht notwendig und ist deswegen zunächst gemäß der Vereinfachung zu vernachlässigen. Dies ist vor allem dadurch möglich, da die Nutzung des Picoboard sowie Adruinos, ebenso wie auch der Umgang mit Scratch, intuitiv¹²⁴ sind und die Schüler nach einer kurzen Einführungsphase experimentell mit diesen Elementen arbeiten können.

Mittels der *Verdichtung* wird die Komplexität eines Unterrichtsgegenstandes herabgesetzt und so die Konzentration auf die wesentlichen Aspekte ermöglicht.¹²⁵ Bezuglich der Haussteuerung soll den Schülern das Prinzip des Programmierens deutlich werden. Über die Bausteine in Scratch können sie mit einfachen Mitteln ein Programm erstellen, welches einen komplexen informatischen Vorgang simuliert. Die Komplexität des Unterrichtsgegenstands ist somit deutlich vereinfacht worden. Der Fokus liegt auf der Erstellung eines funktionstüchtigen, leistungsfähigen Programms, nicht jedoch auf der wissenschaftlich-informatischen Realität.

¹²² Aschersleben 1993, S.148f.

¹²³ Vgl. Romeike, Ralf: Kreativität im Informatikunterricht. Dissertation. Universität Potsdam. 2008, S.113.

¹²⁴ Romeike (2008) bezeichnet die Nutzung gar als „spielerisch“. Vgl. Romeike 2008, S.115.

¹²⁵ Aschersleben 1993, S.49f.

Mit der *Straffung* wird der Gehalt eines Unterrichtsgegenstandes herausgefiltert und so die Unübersichtlichkeit aufgehoben.¹²⁶ Eine Straffung wird innerhalb der Unterrichtsreihe über die Einschübe der verschiedenen Arbeitsblätter zu informatischen Themen erreicht. Die Arbeitsblätter „Picoboard“, „Adruino“, „Kontrollstrukturen“, „Struktogramme“ sowie „Struktogramme II“ stellen eine Informationsmöglichkeit dar, welche komplexe informative Inhalte darstellt, um so die im Unterricht behandelten Inhalte und Vorgehensweise fachlich zu fundieren. Dem Schüler wird auf diese Weise eine weitere Ebene der fachlichen Tiefe aufgezeigt.

Über Beispiele und geeignete Methoden wird die *Veranschaulichung* realisiert, das heißt die Milderung der Abstraktion.¹²⁷ Insbesondere die Projektarbeit fußt mittels des vorgegebenen Themas der Erstellung einer Haussteuerung auf einem schülernahen Beispiel. Somit wird das geforderte Abstraktionsniveau deutlich gesenkt und eine Basis für anschaulichkeit geliefert. Der Aspekt der Veranschaulichung zeigt sich auch darin, dass Schüler während der Programmierung immer direkte Ergebnisse in Form von Veränderungen in ihrer Umwelt, sei es durch Bewegungen oder Geräusche, erhalten.

Über die *Isolierung* kann ein Unterrichtsgegenstand aus komplexen Zusammenhängen herausgelöst betrachtet und fassbar gemacht werden.¹²⁸ Im Kontext der Unterrichtsreihe „Haussteuerung mit Scratch“ setzt die Isolierung an der Stelle an, an welcher das Programmieren von der allgemeinen Ebene auf eine Konkrete gehoben wird, da lediglich eine Haussteuerung programmiert wird. Folglich ist die Unterrichtsreihe tendenziell produktorientiert ausgerichtet und das Erlernen des Programmierens in einem allgemein informatischen Sinn diesem zunächst eher untergeordnet.

4.3.2 Sequenzierung auf Makroebene

In dieser Unterrichtsreihe sollen die Schüler einen Einblick in die Funktionsweise einer Haussteuerung bekommen. Das meint, erst einmal im Groben ein Verständnis dafür zu erlangen, wie eine Haussteuerung aufgebaut ist und inwiefern man unterschiedliche

¹²⁶ Vgl. Aschersleben 1993, S.137ff.

¹²⁷ Vgl. Aschersleben 1993, S.22.

¹²⁸ Vgl. Aschersleben 1993, S.159

Komponenten klassifizieren kann. Im Anschluss daran sollen die Schüler eine eigene Haussteuerung mit Hilfe der grafischen Programmierumgebung Scratch entwickeln. Dies benötigt eine kurze Einführung in des Programm, welche in zwei Doppelstunden vor der eigentlichen Reihe zu behandeln ist. Da das Programm sehr intuitiv aufgebaut ist und syntaktische Fehler wegen der vorgegebenen Programmierbausteine, die nur in syntaktisch richtiger Weise zusammengefügt werden können, nicht begangen werden können, bietet das Programm eine hohe Kreativität. Auch die Motivation ist sehr hoch einzustufen, da die Schüler durch nicht auffindbare syntaktische Fehler nicht demotiviert werden können und sich durch die sofortige grafische Umsetzung schnell Erfolgserlebnisse einstellen.

Im Bereich der Haussteuerung wird ein besonderer Akzent auf dem Kontextgedanken liegen. Das bedeutet für die Schüler, sich nicht nur mit der informatischen Sichtweise zu beschäftigen, sondern auch andere Sichtweisen wie zum Beispiel Sicherheitsaspekte oder Kostenfaktoren mit einfließen zu lassen. Der Kontextgedanke orientiert sich hier an den Unterrichtsentwürfen zu Informatik-im-Kontext¹²⁹, einer Initiative von Bildungsverantwortlichen aus Berlin und Brandenburg, die mittlerweile bundesweit durchgeführt wird. Wichtig hierbei ist, dass der Unterrichtsgegenstand einen Bezug zur Lebenswelt der Schüler bietet, was das Thema Haussteuerung zweifelsohne mitbringt.

Die genaue Umsetzung der Haussteuerung erfolgt mittels des Picobards, über welches Sensoren angesteuert werden können. Dieses Board kann über USB mit einem PC verbunden werden und lässt sich mit Scratch ansteuern. Des Weiteren werden Motoren über das Adruinoboard angesprochen. Auch hier besteht eine USB-Verbindung mit dem PC.

Nachdem sich die Schüler erfolgreich mit der zugrundeliegenden Software beziehungsweise Hardware beschäftigt haben, werden sie in eine kleine Projektarbeit von zwei Doppelstunden geschickt, in der sie anhand einer Miniwelt in Form von einem Miniturhaus, eine eigene Automation installieren und programmieren. Im Anschluss werden die Ergebnisse der Klasse präsentiert, wobei die Kleingruppen jeweils ihre Vorgehens-

¹²⁹ Vgl. Homepage Informatik im Kontext. www.informatik-im-kontext.de (Stand: 27.07.2011).

weise genau erläutern. Schließlich wird im Anschluss noch einmal die informative Sicht beleuchtet, in dem auf kennengelernte Programmierstrukturen eingegangen wird.

Insgesamt besteht die Unterrichtsreihe aus 8 Doppelstunden. Diese werden im Folgenden näher ausgeführt.

4.3.3 Sequenzierung auf Mikroebene

In der ersten Doppelstunde bekommen die Schüler eine Einführung in Scratch. Dabei stellt der Lehrer zunächst ein fertiges Scratchprodukt vor, um den Schülern zu verdeutlichen, wie vielfältig die Möglichkeiten in Scratch sind. Hierbei wird zunächst nur das Ergebnis präsentiert.

Im Anschluss sollen die Schüler die Programmierumgebung Scratch kennen lernen. Der Lehrer erläutert den Aufbau des Programms und geht auf den Begriff des Ereignisses sowie die Programmierstrukturen Wiederholung und Fallunterscheidung ein. Es soll herausgestellt werden, dass es sich um Aktionen von Objekten handelt, die jeweils ihr eigenes Script haben. Hierzu führt er eine kurze Präsentation vor. Beendet wird die Einführung durch ein einfaches Programm, welches über einen Beamer den Schülern vorgeführt wird.

Nun sollen die Schüler aktiv werden und in Partnerarbeit ein eigenes Scratchprogramm entwerfen, welches als einzige Auflage eine Interaktion von Objekten beinhalten soll. Dabei wird den Schülern bewusst ein großer Freiraum gelassen, um mit dem Programm zu experimentieren. Den Schülern wird bis zur nächsten Doppelstunde die Vervollständigung ihrer Programme aufgegeben, so dass zu Beginn der zweiten Doppelstunde die Ergebnisse präsentiert werden können.

In der zweiten Doppelstunde werden zunächst die Programme der Schüler exemplarisch vorgestellt. Hierbei wird darauf geachtet, dass die Schüler zu ihren Programmen genaue Erklärungen abgeben und detailliert erläutern, wie sie vorgegangen sind. Die Mitschüler sind an dieser Stelle dazu aufgefordert, sich kritisch zu den vorgestellten Programmen zu äußern, um so das gesamte Plenum in den Vortrag einzubeziehen.

Nach der Auswertungsphase widmet sich die Lerngruppe nun dem Thema „Haussteuerung“. Zu dieser Thematik soll jeder Schüler zunächst eine Mindmap erstellen. Die Schüler gehen hier assoziativ vor und beschränken sich allein auf Informationen des Vorwissens. Erst im Anschluss sind die Schüler dazu aufgefordert, die vom Lehrer mitgebrachten Fachbücher beziehungsweise das Internet als Informationsquelle zu nutzen, um die Mindmap zu ergänzen. Im Anschluss wird im Plenum eine gemeinsame Mindmap mit Hilfe einer Software erstellt, die dann jedem Schüler zugänglich ist.

Schließlich soll der Fokus auf technische Systeme gelegt werden, zu denen auch die Haussteuerung gehört. Dabei erklärt der Lehrer besonders die Dreiteilung von Sensoren, Aktoren und der Programmierung. Des Weiteren teilt der Lehrer ein Arbeitsblatt aus, in dem das technische System einer Waschmaschine näher betrachtet wird. Als Input bekommen die Schüler einen Informationstext sowie einen Link zu einem Lehrvideo im Internet. Hierbei soll die Funktionsweise der Sensoren und Aktoren von den Schülern erarbeitet und bis zur nächsten Stunde zu Hause fertiggestellt werden.

Zu Beginn der dritten Doppelstunde wird das Arbeitsblatt zu Sensoren und Aktoren besprochen. Im Anschluss daran beschäftigen sich die Schüler verstärkt mit dem Kontextgedanken der Haussteuerung. Die Schülergruppe wird in vier gleichgroße Gruppen unterteilt und widmet sich einer Gruppenarbeit zur Installation einer Haussteuerung, wobei zwei Gruppen möglichst viele Argumente für eine Installation und die anderen beiden möglichst viele Gegenargumente finden sollen. Der Arbeitsauftrag weist darauf hin, dass neben Sicherheitsaspekten, Nutzen, Aufwand auch Kostenfaktoren eine Rolle spielen. Zur Erweiterung der Argumente dürfen die Schüler auch das Internet verwenden. Die Argumentation wird mit Hilfe eines Plakats unterstützt, um den Gedankengang der Gruppe besser zu verdeutlichen. Außerdem bestimmt jede Gruppe einen Sprecher, damit im Anschluss eine Pro-und Contra-Diskussion vollzogen werden kann. Die übrigen Gruppenmitglieder dürfen sich auf Wortmeldung hin auch einschalten. Am Ende der Stunde werden greifende Argumente an der Tafel festgehalten und von den Schülern ins Heft notiert. Die Plakate können im Klassenraum aufgehängt werden.

Das Thema der vierten Doppelstunde beinhaltet nun die Einführung in die konkrete Umsetzung einer Haussteuerung mit Scratch. Dazu stellt der Lehrer zunächst die Kom-

ponenten Picoboard und Adruino vor und führt deren Funktionsweise an einem Beispielprogramm vor. Eine geeignete Miniwelt, an der die Haussteuerung durchgeführt werden kann, kann entweder ein umgebautes Puppenhaus¹³⁰ oder ein selbstgebautes Modell sein. Hier sind Kostenfaktoren abzuwägen.

Im Anschluss bekommen die Schüler die Möglichkeit, in Partnerarbeit die neue Hardware zu erkunden und sie an Beispielprogrammen auszuprobieren. Diese können von den Schülern zunächst leicht abgewandelt werden, bevor sie sich eigene Programme überlegen. Haben sich die Schüler hinreichend mit dem neuen System auseinandergesetzt, geht der Lehrer noch einmal auf den Aufbau solcher technischen Systeme ein und erläutert das EVA-Prinzip. Hier sollte ein kleines Tafelbild den Zusammenhang erläutern, welches von den Schülern anschließend ins Heft notiert wird.

In der fünften und sechsten Doppelstunde wird ein kleines Projekt durchgeführt. Das heißt, die Schüler werden dazu aufgefordert, Vierergruppen zu bilden und jeweils zu einer Miniwelt eine Haussteuerung zu bauen und zu programmieren. Hierbei werden den Schülern die benötigten Sensoren und Motoren zur Verfügung gestellt.

In jeder Gruppe bekommt jedes Mitglied eine besondere Aufgabe zugewiesen und wird somit zu einem Experten auf diesem Gebiet. Einer sollte für die Hardwareinstallation zuständig sein, einer für die Scratchprogrammierung; daneben braucht man einen Ideensammler sowie ein Mitglied, das die Arbeit der Gruppe in einem Lerntagebuch festhält. Dabei übernehmen der Ideensammler und der Lerntagebuchschreiber die Kontrolle der Gruppenarbeit. Insbesondere stärken sie den Fortschrittsgedanken der Gruppe, sodass diese in ihrem Arbeitsprozess stetig voranschreitet. Am Ende der Gruppenarbeit sollen die Gruppen in der Lage sein, in einem Kurzvortrag von fünf bis zehn Minuten ihr Projekt der Klasse vorzustellen. Die Arbeitsaufträge werden den Schülern per Aufgabenblätter ausgehändigt.

In der siebten Doppelstunde werden die Gruppenarbeiten präsentiert. Im Vorhinein werden die Schüler darüber unterrichtet, dass sie zu jeder Arbeit begründete Kritik äußern sollen. Hier erklärt der Lehrer, wie diese im Detail aussehen könnte. Zu der mündlichen Evaluation wird gleichzeitig eine schriftliche Einschätzung der Gruppenarbeiten vorge-

¹³⁰ Vgl. Strecker 2009, S.86.

nommen. Dies geschieht dadurch, dass die Schüler jeweils zu verschiedenen Bereichen auf einer Zahlenskala von sehr gut bis unzureichend die Gruppenarbeiten bewerten. Der Lehrer teilt diese zu Beginn der Präsentation aus und sammelt die Ergebnisse, die anonym abgegeben werden, am Ende wieder ein und kann sie als zusätzliches Indiz für die Notengebung mit einfließen lassen.

In der achten Doppelstunde wird das Augenmerk auf die Modellierung von Programmentwürfen gelegt, um strukturierte Programmierung zu verdeutlichen. Hierbei handelt es sich um einen Exkurs, der nicht direkt für diese Unterrichtsreihe existenziell ist, aber für zukünftige Einheiten strukturierte Programmabläufe und Modellierungsfähigkeiten schult. Über Informationsblätter lernen die Schüler Struktogramme und Programmscripte aus Scratch in Beziehung zu setzen. Im weiteren Verlauf der Stunde sollen die Schüler zu den Programmen ihres Haussteuerungsprojekts Struktogramme entwickeln, die dann exemplarisch von den Schülern vorgeführt werden. Gegen Ende der Stunde bekommen die Schüler ein weiteres Struktogramm ausgeteilt, welches sie bezüglich seiner Funktion beurteilen sollen. Falls sie eine weitere Hilfestellung benötigen, können sie das zugehörige Scratchprogramm aufrufen.

Zum Abschluss der Unterrichtsreihe soll ein Resümee vorgenommen werden. Dazu wird im Unterrichtsgespräch eine Mindmap entwickelt, in der alles aufgenommen wird, was in der Reihe besprochen wurde. Bevor die Stunde zu Ende geht, bittet der Lehrer die Schüler, die Reihe zu evaluieren, indem er Bögen austeilte, auf denen zu verschiedenen Items Bewertungen vorgenommen werden sollen. Auch sollen, wenn möglich, Kommentare abgegeben werden, um die Reihe für die Zukunft noch weiter zu verbessern.

Schließlich gibt es noch weitere Materialien unter der Überschrift „Überprüfe dich selbst“¹³¹. Diese umfassen die fachlichen Schwerpunkte der Unterrichtsreihe „Haussteuerung“, „Kontrollstrukturen“, „if-Anweisungen“, „while-Schleifen“, „Sensoren“, „Aktoren“ sowie „Struktogramme“. Die Arbeitsblätter sollen den Schülern eine eigenständige Überprüfung ihres Wissens ermöglichen und Gelerntes festigen. Das Einreichen der Materialien am Ende der Unterrichtsreihe kann auch als Vorbereitung auf einen

¹³¹ Siehe Anhang S.100-107.

Test oder eine Klassenarbeit verstanden werden. Ebenso ist es denkbar die Materialien zusätzlich innerhalb der laufenden Unterrichtsreihe einzubinden und als zusätzliche Arbeitsblätter an die Schüler auszuteilen. Die Materialien können also variabel und je nach Bedarf eingesetzt werden.

4.3.4 Lernziele

Im Informatikunterricht werden Konzept- und Produktwissen bis hin zu Fertigkeiten vermittelt. Für die Lernenden ist oft unklar, was die wesentlichen Ziele des Unterrichts sind; sie laufen Gefahr, sich in Details zu verlieren.¹³²

Eindeutige und klar verständliche Zielsetzung sind ein wesentlicher Bestandteil von gutem Unterricht. Sowohl für Lehrer als auch für Schüler müssen diese daher eindeutig formuliert werden. Dies gilt insbesondere für die Lernziele im Informatikunterricht, da die Schüler sich hier mit grundlegenden Konzepten, produktspezifischen Umsetzungen dieser Konzepte auseinandersetzen und Fertigkeiten im Umgang mit betreffenden Computerwerkzeugen einüben. In diesem komplexen Umfeld ermöglichen die Lernziele einen Überblick.¹³³

4.3.4.1 Inhaltliche Lernziele

Inhaltlich lernen die Schüler den Umgang mit Scratch sowie die Nutzung der Hardwarekomponenten Picoboard und Adruino. Des Weiteren erhalten sie Wissen über algorithmische Grundbausteine wie If-Anweisungen und Schleifen und machen sich mit der Grundidee von Algorithmen vertraut.

Im Bereich der technischen Systeme lernen sie das EVA-Prinzip kennen und machen darüber hinaus Erfahrungen mit dem Aufbau solcher Systeme. Sie können Sensoren und Aktoren identifizieren, innerhalb der Haussteuerung ansteuern und eine datenbezogene Programmierung vornehmen. Schließlich erlangen sie ein fundiertes Wissen über den Gebrauch von Haussteuerungen und reflektieren diese auf Basis gesellschaftlicher Aspekte.

¹³² Hartmann, Werner/ Näf, Michael/ Reichert, Raimond: Informatikunterricht planen und durchführen. Berlin/Heidelberg: Springer-Verlagh2006, S.51.

¹³³ Vgl. Hartmann,/ Näf/ Reicher 2006, S.51.

4.3.4.1.1 Didaktische Auswahlkriterien für die Lerninhalte

Auf Basis der didaktischen Auswahlkriterien für Lerninhalte nach Hubwieser (2007), welcher sich an den fundamentalen Ideen Bruners (1960) orientiert, sollen die Lerninhalte den vier Kategorien allgemeine Bedeutung, Lebensdauer, Vermittelbarkeit sowie exemplarische Auswahl und Einflechtungen zugeordnet werden.¹³⁴

Das Kriterium der allgemeinen Bedeutung bezieht sich auf die in der Unterrichtsreihe angesprochenen Informatiksysteme, die in Form der programmierbaren Haussteuerung sowie der Lernumgebung Scratch betrachtet werden. Dabei lernen die Schüler Problemlösungsstrategien kennen, machen sich mit dem Lösen von Algorithmen vertraut und treffen Überlegungen zu sinnvollen Systemkonfigurationen. Diese Lerninhalte werden nach Hubwieser der Klasse eins zugeordnet, was der größten Anwendungsbreite entspricht, und sind damit gut für die Schulinformatik geeignet.

Das Kriterium der Lebensdauer korreliert eng mit dem der allgemeinen Bedeutung und macht eine Aussage über die Zukunftsbedeutung. Durch die große Anwendungsbreite ist somit auch die Lebensdauer als hoch einzuschätzen.

Die gute Vermittelbarkeit der Lerninhalte ist gegeben, da es sich hier um Inhalte handelt, die auch von der Gesellschaft für Informatik als Bildungsstandards proklamiert worden sind.¹³⁵ Des Weiteren wird in der Unterrichtsreihe großer Wert auf die intuitive Erschließung beziehungsweise auf geleitetes Experimentieren gelegt, was diesen Aspekt noch einmal unterstreicht.

Im Bereich der Haussteuerungssysteme wird hier eine exemplarische Auswahl für Haussteuerung an sich und für die Betrachtung von technischen Systemen getroffen. Bei der schrittweisen Erschließung wird bewusst darauf verzichtet, die Schüler durch lehrerzentrierten Unterricht einer enormen Stofffülle auszusetzen, sondern eine Reduktion der Informationsflut durch Einflechtung von Inputs¹³⁶ in die Gruppen- beziehungsweise Projektarbeit vorzunehmen.

¹³⁴ Vgl. Hubwieser 2007, S. 82-85.

¹³⁵ Vgl. GI 2008.

¹³⁶ Diese Inputs finden sich in den Arbeitszetteln zu If-Anweisungen, Schleifen und Struktogrammen. Siehe Arbeitsblätter „Infoblatt“, „Struktogramme“ im Anhang.

4.3.4.2 Bildungsstandards

Es existieren keine Bildungsstandards für den Bereich der Sekundarstufe I, von der Kultusminister Konferenz (KMK) wurden lediglich für die Sekundarstufe II solche Standards definiert.¹³⁷

Im Bereich der Sekundarstufe I kann man sich an den Forderungen der Gesellschaft für Informatik (GI) orientieren, welche im Gegensatz zur KMK keine Regelstandards¹³⁸ für die Schüler festlegen, in denen sie ein mittleres Anforderungsniveau erreichen. Vielmehr legt die GI Mindeststandards¹³⁹ fest, die ein Minimum der Kompetenzen festmachen, welches jeder Schüler erreichen soll.

Die GI gliedert die Bildungsstandards in Inhalts- und Prozessbereiche, die miteinander in Beziehung zu setzen sind.¹⁴⁰ Diese Aufteilung folgt dem Muster der NCTM-Standards¹⁴¹ und umfasst nach der GI wesentliche Inhalte der informatischen Bildung für die Sekundarstufe I.

Zu den Inhaltsbereichen zählt die GI "Informationen und Daten", "Algorithmen", "Sprachen und Automaten", "Informatiksysteme" und "Informatik, Mensch und Gesellschaft". Dazu passend werden die Prozessbereiche "Modellieren und Implementieren", "Begründen und Bewerten", "Strukturieren und Vernetzen", "Kommunizieren und Kooperieren" sowie "Darstellen und Interpretieren" formuliert.¹⁴² Insgesamt differenziert die GI zwischen den Kompetenzstufen der Bildungsstandards für die Klassen 5-7 und 8-10.

Das übergeordnete Ziel informatischer Bildung in Schulen ist es, Schülerinnen und Schüler bestmöglich auf ein Leben in einer Informationsgesellschaft vorzubereiten, das maß-

¹³⁷ Vgl. KMK. Einheitliche Prüfungsanforderungen für das Abitur (Informatik).

http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf (Stand: 27.07.2011).

¹³⁸ Vgl. Die KMK stellt lediglich Standards für die Oberstufe auf.

Vgl. <http://www.kmk.org/dokumentation/veroeffentlichungen-beschluesse/bildung-schule/allgemeine-bildung.html#c7540> (Stand: 27.07.2011).

¹³⁹ Vgl. GI 2008, S.2.

¹⁴⁰ Vgl. GI 2008, S.11.

¹⁴¹ Vgl. National Council of Teachers of Mathematics (NCTM). <http://www.nctm.org/> (Stand: 27.07.2011).

¹⁴² Vgl. GI 2008, S.12-14.

geblich durch den verbreiteten Einsatz von Informations- und Kommunikationstechnologie sowohl im privaten als auch im beruflichen Bereich geprägt ist.¹⁴³

Die GI legt besonderen Wert darauf, dass die Inhalts- und Prozessbereiche nicht sequentiell abgehandelt werden, da durch das isolierte Betrachten der Inhalte/Prozesse nur oberflächliches Wissen bei den Schülern evoziert wird. Besser ist es, nach der GI, die Bereiche zu kombinieren und diese in einen Kontext zu stellen. Insbesondere liegt hier das Augenmerk auf dem Lebensweltbezug und daraus resultierend auch auf dem Interesse von Schülerinnen und Schülern.¹⁴⁴

Somit werden im Folgenden die Bildungsstandards der GI für die Jahrgangsstufen 8 bis 10 im Fokus der Betrachtung stehen.

4.3.4.2.1 Bildungsstandards für das Fach Informatik

Gemäß der Unterteilung in Inhalts- und Prozessbereiche werden diese nun näher ausdifferenziert. Zu berücksichtigen gilt hierbei, dass viele der Bildungsstandards der Inhalts- und Prozessbereiche innerhalb der Unterrichtsreihe zwar angesprochen werden, diese sich jedoch nur über einen längeren Zeitraum realisieren lassen, das heißt, sie müssen auch über die Unterrichtsreihe hinaus geschult werden, um vollends realisiert werden zu können. In der Übersicht der Makroebene¹⁴⁵ werden zunächst alle relevanten Bildungsstandards aufgelistet, die innerhalb der jeweiligen Stunden angesprochen werden. Es gilt jedoch nicht, dass alle aufgelisteten Standards in den jeweiligen Stunden erfüllt werden, vielmehr werden bezüglich einiger Standards lediglich Grundsteine gelegt, damit diese im Verlauf des Informatikunterrichts weiter ausgebaut, verfestigt und schließlich realisiert werden können.¹⁴⁶ Innerhalb der Unterrichtsreihe werden vier der fünf Inhaltsbereiche abgedeckt.

¹⁴³ GI 2008, S.11.

¹⁴⁴ Vgl. GI 2008, S.12.

¹⁴⁵ Makroebene der Unterrichtsreihe „Haussteuerung mit Scratch“ im Anhang S. 77-89.

¹⁴⁶ Die Überlegungen zu den Bildungsstandards orientieren sich an dem Grundkonzept des Spiralcurriculums nach Bruner: „Bruner schlägt weiter vor, diese Grundelemente den Stadien der intellektuellen Entwicklung folgend vertiefend und verfeinernd immer wieder aufzunehmen, d.h. eine *spiralförmige* Curriculumorganisation zu Grunde zu legen, die dann zugleich der entwicklungsbedingt unterschiedlichen Akzentuierungen der verschiedenen Repräsentationsweisen des Wissens gerecht werden könnte.“

Vgl. Straka, Gerald/ Macke, Gerd: Lern-Lehr-Theoretische Didaktik. 4. Auflage. Münster: Waxmann Verlag2006, S.117.

Aus dem Bereich Algorithmen werden über die Unterrichtsreihe folgende Bildungsstandards angesprochen:

- Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
- „überprüfen die wesentlichen Eigenschaften von Algorithmen“
 - „lesen formale Darstellungen und setzen sie in Programmen um“
 - „stellen die algorithmischen Grundbausteine formal dar“
 - „verwenden Variablen und Wertzuweisungen“
 - „entwerfen, implementieren und beurteilen Algorithmen“
 - „modifizieren und ergänzen Quelltexte von Programmen nach Vorgaben“¹⁴⁷

Über die Nutzung von Scratch wird der Inhaltsbereich „Algorithmen“ automatisch angesprochen, da mittels der Lernumgebung algorithmische Grundbausteine genutzt werden, um Lösungen zu formulieren. Somit werden zentrale Aspekte wie Schleifen und Verzweigungen den Schülern näher gebracht, so dass diese zur Problembewältigung genutzt werden können. Durch Arbeitsformen wie Partnerarbeit beziehungsweise Gruppenarbeit sowie durch das Vorstellen von Programmstrukturen im Plenum werden diese von den Schülern reflektiert und adaptiert.

Zum Bereich „Sprachen und Automaten“ lassen sich folgende Kompetenzen benennen:

- Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
- „geben Problemlösungen in einer (...) Programmiersprache an“
 - „interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv“¹⁴⁸

In der Unterrichtsreihe wird die Programmierumgebung Scratch mit ihrer eigenen Programmiersprache eingeführt und dient im Folgenden zur Analyse, Darstellung und Lösung von Problemstellungen. Durch seine dynamische Struktur und die direkte Ausführbarkeit der Programmelemente erhalten die Schüler direkt Rückmeldungen zu ihren Lösungsansätzen. Mögliche Fehlermeldungen beschränken sich somit auf die Semantik und werden direkt erkannt.

Ein wichtiger Bestandteil der Reihe ist die Bedeutung von Informatiksystemen. Es werden folgende Standards abgedeckt:

- Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
- „klassifizieren Hardware und Software“

¹⁴⁷ GI 2008, S.15f.

¹⁴⁸ GI 2008, S.16.

„erweitern bestehende Informatiksysteme mit Soft- und Hardwarekomponenten“
„erschließen sich selbstständig neue Anwendungen und Informatiksysteme“¹⁴⁹

Insgesamt ist die Reihe experimentell angelegt. Das heißt, die Schüler sollen sich nach kurzen Einführungsphasen durch den Lehrer eigenständig Informatiksysteme erschließen. Informatiksysteme treten über die Software Scratch sowie die Hardwarekomponenten Picoboard sowie Adruino auf. Insbesondere bei der Entwicklung der Haussteuerung interagieren die Schüler mit dem System.

Dem Bereich Informatik, Mensch und Gesellschaft widmet sich vor allem der Kontextgedanke. Zentrale Aspekte sind hier:

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„kommentieren automatisierte Vorgänge und beurteilen deren Umsetzung“
„bewerten die Auswirkungen der Automatisierung in der Arbeitswelt [Haushalt]“¹⁵⁰

In der Doppelstunde zur Haussteuerung im Kontext werden gesellschaftliche Aspekte bezüglich der Haussteuerung von den Schülern benannt, bewertet und diskutiert. Der Lebensweltbezug der Thematik zu den Schülern legitimiert die Haussteuerung als Kontextgedanken. Die rein informative Sichtweise wird hier durchbrochen und durch gesellschaftliche Aspekte ergänzt.

Außerdem lassen sich die folgenden Prozessbereiche in der Unterrichtsreihe festmachen. Bezuglich des Prozessbereiches Modellieren und Implementieren lässt sich der folgende Standard finden:

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„verwenden bei der Implementierung die algorithmischen Grundbausteine“¹⁵¹

Die algorithmischen Grundbausteine stehen den Schülern durch die Lernumgebung zur Verfügung und können darüber zusammengesetzt und getestet werden.

Im Bereich Begründen und Bewerten soll folgende Kompetenz erreicht werden:

¹⁴⁹ GI 2008, S.17.

¹⁵⁰ GI 2008, S.18.

¹⁵¹ GI 2008, S.19.

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„stellen Vermutungen über Zusammenhänge und Lösungsmöglichkeiten im informatischen Kontext dar“¹⁵²

Dies lässt sich an der Erschließung der Haussteuerung im kontextuellen Sinn festmachen, da sie hier nicht lehrergeleitet sondern eigenständig den Sachzusammenhang erschließen.

Zum Aspekt Strukturieren und Vernetzen werden folgende Standards angebracht:

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„planen Arbeitsabläufe und Handlungsfolgen“
„erstellen netzartige Strukturen“
„nutzen Analogien zwischen informatischen Inhalten oder Vorgehensweisen, um Neues mit Bekanntem zu verknüpfen“¹⁵³

Hier sind vor allem die Projektarbeit und die Zusammenfassung der gelernten Inhalte zu nennen, die auch in Mindmaps strukturiert werden. Analogien finden sich insbesondere zu den Lehrerbeispielen, an denen sich die Schüler orientieren sollen und mit deren Hilfe sich eigene Programme entwickeln.

Der Prozessbereich Kommunizieren und Kooperieren findet sich in:

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„kommunizieren mündlich strukturiert über informative Sachverhalte“
„kooperieren in Projektarbeit bei der Bearbeitung eines informatischen Problems“
„dokumentieren Ablauf und Ergebnisse der Projektarbeit“
„reflektieren gemeinsam Ansatz, Ablauf und Ergebnis des Projekts“¹⁵⁴

Durch die weitgehende Vermeidung von Einzelarbeit kommt dieser Bereich besonders zum Tragen. Die Schüler müssen sich innerhalb der Gruppe arrangieren, das heißt sie sind permanent in einem Interaktionsprozess. Besonders innerhalb der Projektarbeit lernen die Schüler sich zu organisieren und arbeitsteilig vorzugehen. Selbstorganisation meint hier auch Selbstdokumentation, also die Sicherung von Wissen und Ergebnissen innerhalb der Gruppe.

Schließlich findet sich der Punkt Darstellen und Interpretieren in den Präsentationsphasen der Arbeitsaufträge.

¹⁵² GI 2008, S.19.

¹⁵³ GI 2008, S.21.

¹⁵⁴ GI 2008, S.21.

Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10
„wählen eine Darstellungsform auf der Basis allgemein akzeptierter und zweckdienlicher Kriterien aus“¹⁵⁵

Dabei wird vom Lehrer darauf hingewiesen, was für eine gute Präsentation beachtet werden muss und wie sich die übrigen Mitschüler einschalten können.

4.3.4.3 Methodische Lernziele

Nach der Erörterung der inhaltlichen Lernziele, stellt sich nun noch die Frage nach der Auswahl geeigneter Methoden für die konkrete Unterrichtsplanung.

Die Unterschiede bei den Lernenden, beim zu vermittelnden Stoff und die unterschiedlichen Charaktere der Lehrerinnen und Lehrer legen für den Unterricht eine Methodenvielfalt nahe.¹⁵⁶

Insbesondere der Informatikunterricht bietet eine große Möglichkeit für den Einsatz verschiedener Methoden, eine gezielte Auswahl der Methode ist daher zwingend erforderlich. Dies gilt vor allem, da man im Informatikunterricht oftmals heterogene Gruppen unterrichtet, das heißt die Schüler haben unterschiedliche Eingangsvoraussetzungen durch Vorwissen, Fähigkeiten sowie Interesse. Deshalb ist auf Individualisierung zu achten, der Unterricht sollte möglichst schülerorientiert gestaltet werden, um Überforderung oder Langeweile entgegenzuwirken.¹⁵⁷

Die eigenständige Erfassung der Lernumgebung Scratch sowie die Verwendung der Hardwarekomponenten der Haussteuerung¹⁵⁸ erfolgt, nach einer kurzen Einführungsphase durch den Lehrer, in Partner- beziehungsweise Gruppenarbeit.

Sowohl das grundsätzliche Design als auch die spezifische Ausprägung fachdidaktischer Methoden muss die weitgefächerte Dimension zwischen der Stringenz der Fachwissenschaft und der Individualität des Schülers berücksichtigen.¹⁵⁹

¹⁵⁵ GI 2008, S.22.

¹⁵⁶ Hartmann/Näf/Reichert 2006, S.63.

¹⁵⁷ Vgl. Hartmann/Näf/Reichert 2006, S.64.

¹⁵⁸ Picoboard, Adruino, Sensoren, Aktoren

¹⁵⁹ Timmerhaus, Winfried: Fachdidaktik als konstitutives Element universitärer Lehrerbildung. Bestandsaufnahmen, Analysen und Konzeptionen aus erziehungswissenschaftlicher Perspektive. Marburg: Tectum-Verlag2001, S.222.

Hier wird der Umstand beachtet, dass in der Informatik eine große Detailvielfalt, gerade im Bereich von Programmiersprachen, herrscht. Es ist nicht möglich alle diese Details mit den Schülern zu behandeln. Daher sollte der Informatikunterricht die Schüler dazu befähigen, sich nötiges Wissen mittels geeigneter Methoden „just in time“ anzueignen. Eigenständiges Arbeiten ist hierzu der Schlüssel.¹⁶⁰ In der hier vorgestellten Unterrichtsreihe finden sich die Methoden Gruppenarbeit, Projektarbeit sowie entdeckenden Lernen, besonders in den Arbeitsaufträgen zu Scratch, dem Picoboard und Adruino sowie in der Projektarbeit zur Hausteuerung.

Die Schüler werden über Partner-, Gruppen und Projektarbeit an eigenständiges Lernen gewöhnt und können darüber hinaus nach der Erschließung des Lernprogramms sowie der Hardwarekomponenten sich weitere Software- und Hardwarekomponenten erschließen und mit ihnen arbeiten.

Ein weiterer Aspekt bezüglich des eigenständigen Lernens ist das Erschließen von Sachverhalten und Sachinformationen. Somit sind die Schüler über Rechercheaufgaben, beispielsweise in der 3. Doppelstunde zum Thema Hausteuerung im Kontext, ebenso zu eigenständigem Arbeiten angehalten. Innerhalb der Projektarbeit wird die Schülergruppe in Vierergruppen eingeteilt, wobei jede Vierergruppe aus vier Experten besteht, die jeweils unterschiedliche Arbeitsschwerpunkte haben. Das vermittelt den Schülern den Bestandteil der vorgegebenen Binnendifferenzierung einer Gruppenarbeit, bei der die Aufgabenschwerpunkte aber noch selbst erarbeitet werden müssen. Das methodische Lernziel liegt hier in der Kommunikation innerhalb der Gruppe und in der Koordinierung der unterschiedlichen Bereiche, neben der Arbeitsteilung in Gruppenarbeiten ist also die Kooperation ein wichtiges methodisches Lernziel.

Ein weiteres methodisches Lernziel liegt in der Vorbereitung einer argumentgestützten Diskussion und deren anschließender Durchführung. Das Herauskristallisieren von Argumenten aus Informationsquellen und das Festhalten in aussagekräftigen Thesen auf

¹⁶⁰ Vgl. Hartmann/Näf/Reichert (2006), S.64.

einem Plakat wird geschult.¹⁶¹ Die Schüler gehen hier planvoll, strukturiert und logisch vor.

Die Diskussionsmethode ist im Bereich der Informatik eher selten anzutreffen, da sie meist nur in den sprach- oder gesellschaftswissenschaftlichen Fächern und weniger in den Naturwissenschaften eingesetzt wird.¹⁶² Da aber die Haussteuerung auch im gesellschaftlichen Kontext betrachtet wird, bietet sich hier diese Methode an.

Diskussionen innerhalb von Gruppen oder ganzen Klassen sind bei Jugendlichen oft recht beliebt, und zweifellos handelt es sich dabei um ein auf die Schülerinnen und Schüler zentriertes Verfahren.¹⁶³

Ein zentraler Bestandteil in der Methodenvielfalt der Unterrichtsreihe ist das Mind-Mapping. Diese Methode spiegelt die Gedankenstruktur im Gehirn wider und ist gut geeignet, einen Überblick über einen Sachzusammenhang zu erhalten.¹⁶⁴

4.3.5 Motivationale Aspekte

Die Beziehung zwischen Motivation und Lernen ist immer *wechselseitig*: Motivation ist Voraussetzung, aber auch Ergebnis von Lernprozessen.¹⁶⁵

Wie viele andere, betont auch Hubwieser (2007) in seiner Didaktik der Informatik den positiven Einfluss motivationaler Aspekte auf erfolgreiches Lernen. Er formuliert, dass ohne eine angemessene Motivierung der Versuch, den Lernwillen des Schülers anzuregen, sinnlos ist.¹⁶⁶ Motivierung steht somit im Zentrum jeglichen didaktischen Handelns und ist folglich auch zentral für die hier vorgestellte Unterrichtsreihe. Vor allem bedingt durch den Umstand, dass sich die Sequenz auf den Grundgedanken der Informatik im Kontext stützt, ist einzusehen, dass die Reihe auf einem motivationalen Fundament fußt.

¹⁶¹ Für die Gestaltung von Plakaten und auch für die Durchführung der Diskussionen lassen sich viele verschiedene Methoden finden.

Siehe hierzu Gugel, Günther: 2000 Methoden für Schule und Lehrerbildung. 1. überarbeitete Auflage. Weinheim/ Basel: Beltz Verlag2011, S.36ff.

¹⁶² Vgl. Klauer, Karl Josef/ Leutner, Detlef: Lehren und Lernen. Einführung in die Instruktionspsychologie, Weinheim/ Basel: Beltz Verlag2007, S. 173.

¹⁶³ Klauer /Leutner 2007, S. 173.

¹⁶⁴ Vgl. Gugel 2011, S.80.

¹⁶⁵ Hubwieser 2007, S.16.

¹⁶⁶ Vgl. Hubwieser 2007, S.15.

Die Schülerorienierung durch das Zentrieren auf Lebensweltbezüge, hier die Haussteuerung, macht die Besonderheit der Unterrichtseinheit aus.

Im Allgemeinen versteht man unter dem Terminus Motivation „einen kurz andauernden Zustand des Angetriebenseins“¹⁶⁷. Das bedeutet, der Fokus des Lernenden liegt für einen begrenzten Moment auf einem Gegenstand und eventuell dem Verstehen oder Befassen mit ebendiesem. Dieser Vorgang¹⁶⁸ ist jedoch eher spontaner Natur und schwer zu steuern. Hingegen wird unter dem Begriff Motivierung das aktive Bemühen um das Erreichen eines Motivationszustandes gefasst.¹⁶⁹ Der Schüler befasst sich mit einem Gegenstand oder einer Thematik angetrieben von dem eigenen Lernwillen.

Die Faktoren, welche in ebendiesem Lernwillen münden, können als Initiierungs- und Lenkungsfaktoren bezeichnet werden. Hierunter fasst man Momente wie beispielsweise Funktionslust, Schaffensfreude, Erfolg, Ehrgeiz und ähnliche. Diese werden als Motive bezeichnet.¹⁷⁰

Insbesondere wichtig für das Erlangen von Motivierung des Schülers, also das Erreichen eines finalen Zustandes von Motivation, sind drei verschiedene Ebenen: Lernmotivation/Leistungsmotivation, intristische/extrinsische Motivation sowie Eingangs-/Verlaufsmotivierung.¹⁷¹ Diese drei Aspekte sollen im Folgenden näher erläutert sowie bezüglich der ausgearbeiteten Unterrichtsreihe beurteilt werden.

4.3.5.1 Lernmotivation/ Leistungsmotivation

Die Lernmotivation wird entscheidend durch die Auseinandersetzung mit der Umwelt beeinflusst. Anreize aus der Realität werden verarbeitet und werfen Fragen auf, die dazu anregen, sich detailliert mit ihnen auseinanderzusetzen.

Lernmotivation bezeichnet nach Knörzer die

momentane Bereitschaft eines Individuums, sensorische, kognitive und motorische Funktionen in einer durch schulische Anforderungen vorstrukturierten Lernsituation darauf zu richten und derart zu koordinieren, dass ein vorgegebenes Lernziel erreicht wird.¹⁷²

¹⁶⁷ Hubwieser 2007, S.15.

¹⁶⁸ Vgl. Hubwieser 2007, S.15.

¹⁶⁹ Vgl. Hubwieser 2007, S.15.

¹⁷⁰ Vgl. Hubwieser 2007, S.15.

¹⁷¹ Diese drei Paare werden von Hubwieser unterschieden. Vgl. Hubwieser 2007, S.16.

Für den Unterricht im Allgemeinen ist es folglich wichtig, in der Aufarbeitung des Inhalts zu variieren, mit anderen Worten auf verschiedene Methoden und Herangehensweisen zurückzugreifen.

Bezüglich der hier vorgestellten Unterrichtsreihe zeigt sich eine Basis für die Anregung von Lernmotivation einerseits in dem gewählten Unterrichtsgegenstand der Haussteuerung. Das Phänomen einer Haussteuerung ist den Schülern aus ihrem Alltag bekannt und liefert so einen Lebensweltbezug. Andererseits ermöglichen die unterschiedlichen Methoden, wie beispielsweise die Partner- und Gruppenarbeit zu verschiedenen Programmieraufgaben in der 1. sowie 5./6. Doppelstunde, die Gruppendiskussion zum Kontext in der 3. Doppelstunde und die diversen Vortragssituationen, den Schülern vielfältige Herangehensweisen an das Thema. Die Schüler befassen sich nicht eindimensional mit der Thematik Haussteuerung, sondern erfassen diese auf kritisch-reflexiver Ebene durch die Diskussion sowie auf informatischer Ebene durch die Programmierung einer eigenen Haussteuerung.

Im Gegensatz zur Lernmotivation beschreibt die Leistungsmotivation eine Art Persönlichkeitsmerkmal. In der Leistungsmotivation bündeln sich einerseits Erfolgsorientierung und andererseits Anstrengungsbereitschaft.¹⁷³

Murray (1938) beschreibt Leistungsmotivation als

Die Tendenz, etwas so schnell und gut zu machen, wie möglich.
Die Tendenz, physikalische Objekte, Menschen und Ideen zu beherrschen.
Hindernisse zu überwinden.
Hohe Standards zu erreichen.
Sich selbst auszureichen.
Mit anderen konkurrieren und sie zu überwinden versuchen.
Durch geschickten Einsatz eigener Begabungen den Eigennutz zu erhöhen.¹⁷⁴

Dies zeigt, dass Leistungsmotivation nur schwer vom Lehrer beeinflusst werden kann, da diese Bereitschaft vom Schüler ausgehen muss.

¹⁷² Knörzer 1976, S.139.

¹⁷³ Vgl. Edelmann, Walter: Lernpsychologie. 6.Auflage. Weinheim: Beltz2000, S.256.

¹⁷⁴ Edelmann 2000, S.252. Vgl. Murray 1938, S.164.

Übertragen auf die Unterrichtsreihe lassen sich die Freiräume für Präsentationen der Ergebnisse anführen, in welchen den Schülern die Möglichkeit geben, ihre Ergebnisse vor dem Plenum darzustellen. Insbesondere die offenen Aufgabenstellungen¹⁷⁵ ermöglichen den Schülern eine individuelle Herangehensweise an die Thematik, sodass sich eine Vielzahl unterschiedlicher Lösungszugänge ergeben wird. Die Schüler können also aus ihrem eigenen Potential schöpfen und gemäß der Definition von Murray (1938) ihre Leistungsmotivation optimal ausnutzen.

Schließlich kommt sowohl der Lern- wie auch der Leistungsmotivation die didaktische Reduktion zugute, da die Schüler so nicht mit zu komplexen informatischen Inhalten überfordert werden, sondern sich gemäß ihres Leistungsniveaus mit den Inhalten auseinandersetzen können.¹⁷⁶

4.3.5.2 intrinsische/extrinsische Motivation

Ein weiterer wichtiger Aspekt der Motivation ist die intrinsische, beziehungsweise extrinsische Motivation.

Unter intrinsischer Motivation versteht man eine Tätigkeit, die nur wegen ihrer Handlung durchgeführt wird und nicht aufgrund von Folgen der Handlung. Das Hauptaugenmerk liegt somit auf der Handlung selbst und nicht auf deren Ergebnissen.¹⁷⁷

Die Schüler sind in der vorgestellten Unterrichtsreihe dazu aufgerufen sich in Partnerbeziehungsweise Gruppenarbeit ein Programm, Scratch, in Eigentätigkeit zu erschließen. Idealerweise empfinden die Schüler es als Herausforderung mit Scratch umgehen zu können und damit ein „schönes, sinnvolles, anspruchsvolles“ Programm erstellen zu können. Die Aufgabenstellung „Erstelle nun ein eigenes Programm in Scratch und ver-

¹⁷⁵ Zu denken ist hier an die Projektarbeit in der 5./6. Doppelstunde in welcher die Schüler eine Haussteuerung simulieren sollen. Hierbei wurde auf genaue Vorgaben verzichtet, damit die Schüler ihr kreatives Potential ausnutzen können und zu möglichst vielen unterschiedlichen Ergebnissen gelangen, was im Sinne eines gegenseitigen Lernens (Lernen von Schülern durch Schüler) förderlich ist.

¹⁷⁶ Vgl Seibert, N./Serve, H.J. (Hrsg.): Prinzipien guten Unterrichts. Kriterien einer zeitgemäßen Unterrichtsgestaltung. München PimS-Verlag1992, S.175f.

¹⁷⁷ Vgl. Geyer, Christoph: Intrinsisch motiviertes Streben nach sozialer Anerkennung. Eine Analyse der Ursachen und Wirkung Norderstedt: GRIN Verlag2008, S.3.

suche dabei die Objekte in Interaktion treten zu lassen“¹⁷⁸ ist bewusst offen gewählt, damit die Schüler sich nicht eingeschränkt fühlen und somit ihren persönlichen Ansprüchen an die Aufgabenlösung folgen können. Dies schafft eine motivationale Basis für einen persönlichen Lernschritt, der durch die geschaffenen Grundvoraussetzungen der Aufgabe idealer Weise intrinsisch ist.

Im Gegensatz dazu steht die extrinsische Motivation, die nur die Ergebnisse und Folgen der Handlung im Fokus sieht. Der Lernende handelt hier zweckorientiert, um beispielsweise eine Prüfung zu bestehen oder eine gute Note zu bekommen.¹⁷⁹

Im schulischen Kontext sollte die intrinsische Motivation überwiegen, da das aus eigenem Antrieb bedingte Interesse einen höheren Stellenwert einnimmt. Dennoch lässt sich auch innerhalb der vorgestellten Unterrichtsreihe ein gewisses Maß an extrinsischer Motivation nicht vermeiden, da die Ergebnisse der Schüler zu einem Teil die Notengebung beeinflussen. Vor allem wird die extrinsische Motivation durch den Druck des Vorstellens der Projektarbeit, das damit einhergehende Bewerten durch das Plenum sowie die Ergebnissicherung durch den Experten für das Lerntagebuch¹⁸⁰ gefördert. Allerdings ist nicht auszuschließen, dass sich innerhalb der Erarbeitung des Projekts auch intrinsische Motivation einstellt.

Bemerkenswert ist, dass intrinsische und extrinsische Motivation sich überlagern können und es zu Interferenzeffekten kommen kann.¹⁸¹ Das heißt, der Lernende ist nicht mehr dazu in der Lage zwischen beiden zugehörigen Motiven zu unterscheiden und verbindet beispielsweise direkt eine Prüfung mit einem Unterrichtsgegenstand statt den Gegenstand selbst.

Die negative Kopplung gilt es stets zu vermeiden. Bezuglich der geplanten Sequenz soll dies über das Führen eines Lerntagebuchs umgangen werden. Hier wird der Lernverlauf insgesamt festgehalten und kann in der Benotung berücksichtigt werden, so dass es keine einseitige Note nach der Beurteilung einer Momentaufnahme, der Präsentation, ge-

¹⁷⁸ Aufgabenstellung aus der 1. Doppelstunde.
Vgl. Anhang S.96.

¹⁷⁹ Vgl. Geyer 2008, S.3.

¹⁸⁰ Vgl. Seibert/Serve 1992 S.177.

¹⁸¹ Vgl. Hubwieser 2007, S.16.

ben wird. Dies kann Schülern Sicherheit geben, da sie über die gesamte Zeitspanne des Projekts sich ihre Note erarbeiten.

4.3.5.3 Eingangs-/ Verlaufsmotivation

Der Eingangsmotivation kommt eine besondere Bedeutung zu. Hier muss das Interesse der Schüler an einem Thema geweckt werden. Dabei kann auf bekanntes Vorwissen zurückgegriffen und damit durch vorhandene Vorkenntnisse Interesse hervorrufen oder durch einen völlig neuen Unterrichtsgegenstand Neugier erweckt werden. Die Neugier bildet den entscheidenden Faktor beim Lernen. Dabei ist wichtig, dass den Schülern zu einem Thema ein Lebensweltbezug verdeutlicht wird und Zusammenhänge zu ihrem jetzigen oder zukünftigen Leben hergestellt werden.¹⁸² Gelingt es dem Lehrer nicht, diesen größeren Zusammenhang bewusst zu machen und so dem Unterrichtsgegenstand einen Sinn zu geben, ist es schwer, die Aufmerksamkeit und die engagierte Arbeit der Schüler zu erlangen. Die Eingangsmotivation sollte einen klaren inhaltlichen Bezug zum zu lösenden Problem haben, welches von den Schülern, wenn möglich, so eingeschätzt wird, dass sie es entweder aus eigener Kraft oder in der Gruppe lösen können. Auf ein „Vorkauen“ des Lehrers¹⁸³ soll verzichtet werden. Verfehlt die Eingangsmotivation ihr Ziel, so hat der Lehrer im weiteren Verlauf es schwer, die Schüler zu erreichen. Dennoch reicht eine Anfangsmotivation nicht aus, um die Schüler dauerhaft für ein Thema zu begeistern. Vielmehr hat der Lehrer die Aufgabe die gesamte Unterrichtsstunde motivierend anzulegen.¹⁸⁴

Die Anfangsmotivation wird, bezogen auf die Unterrichtsreihe, über das Einführen eines neuen, unbekannten Programms sowie den offenen Arbeitsauftrag gefördert. Die Schüler befassen sich individuell mit einem neuartigen Komplex und können so experimentell vorgehen.

Um über den Verlauf der Unterrichtsstunde hinweg eine Motivation aufrecht zu erhalten, ist es sinnvoll, dass der Lehrer im Vorhinein, also bei der Planung der Unterrichts-

¹⁸² Vgl. Serve, Helmut J.: Das Unterrichtsprinzip der Motivierung. In: Prinzipien guten Unterrichts. Hrsg. von Seibert und Serve. München: PimS1992, S.186

¹⁸³ Serve 1992, S.187

¹⁸⁴ Vgl. Serve 1992, S.187

stunde, motivationale Aspekte zur Verlaufsmotivation bereithält, die er bei Bedarf oder vorbeugend einsetzt. Ein Bedarf könnte sich andeuten, wenn die Schüler sichtlich die Motivation verlieren, beispielsweise die Beteiligung nachlässt. Hierbei sollte, wenn möglich, auf extrinsische Motivation verzichtet werden, da der Fokus auf dem Problem und nicht zum Beispiel auf das Erreichen einer bestimmten Note liegt.¹⁸⁵

Für die Unterrichtsreihe kann die Verlaufsmotivation über die stets neuen Aufgabenbereiche gesteigert werden. Die Schüler müssen sich in nahezu jeder neuen Doppelstunde, mit Ausnahme der Projektarbeit, einer neuen Problematik stellen. Die Variation der Methoden sowie die individuelle Herangehensweise fördert die Verlaufsmotivation zusätzlich.

Obwohl die Notengebung in der Schule stets gegenwärtig und somit aus den Köpfen der Schüler nicht wegzudenken ist, kann innerhalb der hier vorgestellten Unterrichtsreihe der Fokus eher auf den Unterrichtsgegenstand gelegt werden, da die Schüler permanent beurteilt werden und nicht ein einzelnes Ergebnis nahezu die gesamte Note ausmacht.

Insgesamt muss der Lehrer auf eine gute Balance des Motivierungsniveaus achten, da eine Überdosierung eher kontraproduktiv ausfallen könnte. Die Schüler müssen sich ernst genommen fühlen.

Die Förderung der Lernmotivation bedeutet kein trickreiches Verhätscheln und Kögern der Schüler, sondern eine Unterstützung auf ihrem Wege zum mündigen Erwachsenen.¹⁸⁶

Schließlich ist zu sagen, dass der Unterricht so aufgebaut sein sollte, dass die Interessen der Schüler geweckt und über die Stunde hinweg gehalten werden sollten. Die Motivation darf nicht zur Manipulation der Problemstellung werden und somit das Problem an sich und dessen Sinn verdecken.

5 Ausblick

Informatik im Kontext verspricht damit, ein spannendes Projekt zu werden, bei dem das Bildungspotenzial des Informatikunterrichts stärker als bislang zur Geltung kommt. Es liegt an uns, dieses Versprechen zu erfüllen.¹⁸⁷

¹⁸⁵ Vgl. Serve 1992, S.190.

¹⁸⁶ Serve 1992, S.191

Vor allem in der heutigen Zeit ist ein selbstbestimmter, verantwortungsvoller und sicherer Umgang mit Informatiksystemen notwendig. Dieser fordert einerseits den technischen Sachverstand und andererseits die Kenntnisse über gesellschaftliche Rahmenbedingungen.¹⁸⁸

Während niemand im offenen Diskurs die Bedeutung der gesellschaftlichen Dimension anzweifelt, werden sie in der Ausbildung allzu oft vernachlässigt.¹⁸⁹

Ebenso wie Döbli Honegger/Frey/Braxmayer (2009)¹⁹⁰ finden auch Koubek/Kunz (2007) konkrete Gründe für diesen Mangel an Einbindung des Kontextgedankens im Informatikunterricht: den Mangel an Unterrichtsmaterialien.¹⁹¹ Auch die von ihnen erstellten Unterrichtsentwürfe im Rahmen der von ihnen gegründeten AG „Informatik – Mensch – Gesellschaft“¹⁹² sind nach eigener Aussage mangelhaft bezüglich einer geeigneten Sachanalyse, die es Lehrkräften ermöglichen würde, sich in die Thematik so einzuarbeiten, dass sie die Unterrichtseinheiten durchführen können.¹⁹³

Auch wenn der Informatikunterricht es den Schülern erleichtert, im Umgang mit Informatiksystemen kreativ zu sein, ist das Anwenden von kreativen Lehrmethoden im Unterricht notwendig.¹⁹⁴

Weiterhin hat sich gezeigt, dass sich mittels geeigneter Methoden und ausgewählten Beispielen eine gute Basis für motivierenden und kreativitätsfördernden Informatikunterricht herstellen lässt. In diesem Zusammenhang wird auch von Romeike (2007) betont, dass kreative Lehrmethoden absolut notwendig sind, um Kreativität, aber auch Motivation seitens der Schüler zu fördern.

¹⁸⁷ Homepage Informatik im Kontext. <http://www.informatik-im-kontext.de/> (Stand: 27.07.2011).

¹⁸⁸ Vgl. Koubek, Jochen/Kurz, Constanze: Informatik – Mensch – Gesellschaft im Schulunterricht.. In: Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn: Köllen2007, S.126.

¹⁸⁹ Koubek/Kurz 2007, S.126.

¹⁹⁰ Zur Förderung des Informatikwissens von Kinder und Jugendlichen sind gute Lehrmittel rar. Vgl. Döbli Honegger/Frey/Braxmeier 2009, S.305.

¹⁹¹ Vgl. Koubek/Kunz 2007, S.126.

¹⁹² Homepage der AG „Informatik – Mensche – Gesellschaft“ <http://waste.informatik.hu-berlin.de/Lehre/informatik-ag/index.html> (Stand: 27.07.2011).

¹⁹³ Vgl. Koubek/Kurz 2007, S.132.

¹⁹⁴ Romeike 2007, S.59.

An diese Aussage anschließend ergibt sich die Forderung nach ebensolchen brauchbaren Lehrmethoden. Folglich muss hinsichtlich der gezielten und begründeten Methodenauswahl noch weiter geforscht werden und die Methoden müssen in der praktischen Umsetzung im Informatikunterricht weiter entwickelt werden.

Des Weiteren fehlen neben einem nützlichen Methodenpool die Lehrmittel, etwa ausgearbeitete Unterrichtseinheiten und hierzu umfassende Lehrwerke, mit welchen kreativitäts- und motivationsfördernder Unterricht konzeptionell umgesetzt werden könnte.

Der in dieser Arbeit entwickelte Unterrichtsentwurf soll dazu einen Beitrag leisten und stellt eine mögliche Herangehensweise für das Erfassen und Erlernen informatisch-fachlicher Inhalte im Hinblick auf Kreativitäts- und Motivationsentwicklung dar.

LITERATURVERZEICHNIS

- Albrecht, Frank-Werner: Beschallungstechnik. Theorie und Praxis. 3. Auflage.
Renningen-Malmsheim expert Verlag1995.
- Aschersleben, Karl: Welche Bildung brauchen Schüler? Vom Umgang mit dem
Unterrichtsstoff. Bad Heilbrunn: Klinkhardt1993.
- Balzert, Heide: Lehrbuch der Objektmodellierung. Analyse und Entwurf.
Berlin/Heidelberg: Sprektrum1999.
- Berns, Karsten/Schmidt, Daniel: Programmierung mit Lego® Mindstorms® NXT.
Heidelberg/London/New York: Springer Verlag2010.
- Boden, M. A.: Die Flügel des Geistes: Kreativität und künstliche Intelligenz. München:
DTV1995.
- Böstler, Jürgen: Objektorientiertes Programmieren – Machen wir irgendwas falsch? In:
Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn:
Köllen2007.
- Böttcher, Axel/Kneißel, Franz: Informatik für Ingenieure. Grundlagen und
Programmierung in C. 2. Auflage. München: Oldenbourg Wissenschaftverlag2002.
- Braun, Daniel: Roboter programmieren mit NXC für LEGO® MINDSTORMS® NXT.
Heidelberg/München/Landsberg/Frechen/Hamburg mitp2009.
- Claus, Volker/ Schwill, Andreas (Hg.): Duden Informatik. Ein Fachlexikon für Studium
und Praxis.- Mannheim: Duden Verlag 2001.
- Crutzen, Cecile K. M./ Hein, Hans-Werner: Objektorientiertes Denken als didaktische
Basis der Informatik. In: Innovative Konzepte für die Ausbildung. Hrsg. von Sigrid
Schubert. Berlin/Heidelberg: Springer1995.
- Crutzen, Cecile K. M.: Dekonstruktion, Konstruktion und Inspiration. Gender in der
Informatik. In: Dokumentation wissenschaftlicher Kolloquien 1999-2002. Band 2.
Hrsg. von Ursula Paravicini und Maren Zempel-Gino. Hannover: NFFG2003.
- Döbeli Honegger, Beat/ Frey, André/ Braxmeier, Philipp: Mit iLearnIT.ch spielerisch
das Interesse an Informatik wecken. In: Zukunft braucht Herkunft. Hrsg. von
Bernhard Koerber. Bonn: Köllen2009.
- Edelmann, Walter: Lernpsychologie. 6.Auflage. Weinheim: Beltz2000.
- Fothe, Michael/ Friedrich, Steffen: Informatik in die Schule! - ein erneutes Plädoyer.
(05/2011).
<http://www.gi.de/fileadmin/redaktion/Vorstandsglossen/GI-Vorstandsmitglied-Fothe110523.pdf> (Stand: 27.07.2011).
- Geyer, Christoph: Intrinsisch motiviertes Streben nach sozialer Anerkennung. Eine
Analyse der Ursachen und Wirkung Norderstedt: GRIN Verlag2008.
- GI: Grundsätze und Standards für die Informatik in Schulen. Bildungsstandards
Informatik für die Sekundarstufe I. Empfehlungen der GI erarbeitet vom Arbeitskreis
„Informatik“. In: LOG IN. Heft 150/151. Beilage zur LOG IN. Berlin: LOG IN
Verlag2008.
- Grötsch, Eberhard: SPS. Speicherprogrammierbare Steuerungen als Bausteine verteilter
Automatisierung. 5.Auflage. München: Oldenbourg Industrieverlag2004.

- Gugel, Günther: 2000 Methoden für Schule und Lehrerbildung. 1. überarbeitete Auflage. Weinheim/ Basel: Beltz Verlag2011.
- Hanselmann, Uta: »Ein bisschen Word«. In: Die Zeit. Nr.9. (19/02/2009).
<http://www.zeit.de/2009/09/C-Informatik> (Stand: 27.7.2011).
- Hartmann, Werner/ Näf, Michael/ Reichert, Raimond: Informatikunterricht planen und durchführen. Berlin/Heidelberg: Springer-Verlagh2006.
- Hering, Ekbert/Gutekunst, Jürgen/Dyllong, Ulrich: Handbuch der praktischen und technischen Informatik. 2. Auflage. Berlin/Heidelberg: Springer2000.
- Herper, Henry/Hinz, Volkmar: Analyse eines Informatiksystems durch unterschiedliche Modellierungsansätze. In: Unterrichtskonzepte für informative Bildung. Hrsg. von Steffen Friedrich Bonn: Köllen2005.
- Homepage der AG „Informati – Mensche – Gesellschaft“ <http://waste.informatik.hu-berlin.de/Lehre/informatik-ag/index.html> (Stand: 27.07.2011).
- Homepage Biologie im Kontext. <http://bik.ipn.uni-kiel.de/typo3/index.php?id=30> (Stand: 27.07.2011)
- Homepage Informatik im Kontext. www.informatik-im-kontext.de/ (Stand: 27.07.2011).
- Homepage Informatik im Kontext. Konzepte. <http://medienwissenschaft.uni-bayreuth.de/inik/konzepte.html> (Stand: 27.07.2011).
- Homepage Lego Mindstorms. <http://mindstorms.lego.com/en-us/Default.aspx> (Stand: 27.07.2011).
- Homepage Lego Mindstorms Education. <http://www.nxt-in-der-schule.de/lego-mindstorms-education-nxt-system/nxt-hardware/sensoren/hitechnic-sensoren> (Stand: 27.07.2011).
- Homepage mytoys. <http://www.mytoys.de/PLAYMOBIL-PLAYMOBIL-4279-Neues-Wohnhaus/Bausteine/Kleinkindspielzeug/KID/de-mt.to.ca02.29.08/1722332> (Stand: 27.07.11).
- Homepage Scratch. http://info.scratch.mit.edu/Scratch_1.4_Download (Stand: 27.07.2011).
- Hubwieser, Peter: Didaktik der Informatik. 3. Auflage. Berlin / Heidelberg: Springer2007.
- Kiesmüller, Ulrich: Prozessbegleitende, automatisierte Identifizierung der Problemlösestrategien von Lernenden beim Lösen algorithmischer Probleme In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber. Bonn: Köllen2009.
- Klafki, Wolfgang: Studien zur Bildungstheorie und Didaktik. Weinheim: Beltz1970.
- Klauer, Karl Josef/ Leutner, Detlef: Lehren und Lernen. Einführung in die Instruktionspsychologie, Weinheim/ Basel: Beltz Verlag2007.
- KMK. Einheitliche Prüfungsanforderungen für das Abitur (Informatik).
http://www.kmk.org/fileadmin/veroeffentlichungen_beschluessel/1989/1989_12_01-EPA-Informatik.pdf (Stand: 27.07.2011).
- Koubek, Jochen/Kurz, Constanze: Informatik – Mensch – Gesellschaft im Schulunterricht.. In: Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn: Köllen2007.

- Koubek, Jochen/ Schulte, Carsten/ Schulze, Peter/ Witten, Helmut: Informatik im Kontext (IniK). Ein integratives Unterrichtskonzept für den Informatikunterricht. In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber Bonn: Köllen2009.
- Krapp, A.: Entwicklung und Förderung von Interessen im Unterricht. In: Psychologie in Erziehung und Unterricht, Ernst Reinhardt Verlag München Basel1998.
- Lerch, Reinhart: Elektrische Messtechnik. Analoge, digitale und computergestützte Verfahren. 5.Auflage. Heidelberg/ Dordrecht/ London/ New York: Springer-Verlagh2010.
- Lindsay, Stephen: Home Automation 101. Discover How To Create Or Transform Your Current Home Today. http://books.google.de/books?id=1-WJhqL0H6EC&printsec=frontcover&dq=home+automation+101&hl=de&ei=GyUsTtCKHsPn-gam2IDtDQ&sa=X&oi=book_result&ct=book-thumbnail&resnum=2&ved=0CD0Q6wEwAQ#v=onepage&q&f=false (Stand: 27.07.2011).
- Magenheim, Johann S.: Informatiksystem und Dekonstruktion als didaktische Kategorien – Theoretische Aspekte und unterrichtspraktische Implikationen einer systemorientierten Didaktik der Informatik. Universität Paderborn 2000.
- Memmert, Wolfgang: Didaktik in Grafiken und Tabellen. 5. Auflage. Bad Heilbrunn: Klinkhardt1995.
- Modrow, Eckart: Technische Informatik mit Delphi. Norderstedt: emu-online Verlag2004.
- National Council of Teachers of Mathematics (NCTM). <http://www.nctm.org/> (Stand: 27.07.2011).
- Odendahl, Manuel/ Finn, Julian/ Wenger, Alex: Arduino. Physical Computing für Bastler, Designer & Geeks. 2. Auflage. Köln: O'Reilly Verlag2010.
- Pritschow, Günter: Einführung in die Steuerungstechnik. München/Wien: Hanser2006.
- Resnick, Mitchel (u.a.): Scratch: Programming for all. <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (Stand: 27.07.2011).
- Romeike, Ralf: Kriterien kreativen Informatikunterrichts. In: Didaktik der Informatik in Theorie und Praxis. Hrsg. von Sigrid Schubert. Bonn: Köllen2007.
- Romeike, Ralf: Kreativität im Informatikunterricht. Dissertation. Universität Potsdam. 2008.
- Romeike, Ralf: "Das bessere Werkzeug" - Anmerkungen zur Diskussion Etoys vs. Scratch. In: LOG IN Zeitschrift für informative Bildung und Computer in der Schule. 163/164. Berlin: Log-In-Verlag2010.
- Schubert, Sigrid/Schwill, Andreas: Didaktik der Informatik. 2. Auflage. Heidelberg: Spektrum2011.
- Schwill, Andreas: Programmierstile im Anfangsunterricht. In: Innovative Konzepte für die Ausbildung. Hrsg. von Sigrid Schubert. Berlin/Heidelberg: Springer1995.
- Seibert, N./Serve, H.J. (Hrsg.): Prinzipien guten Unterrichts. Kriterien einer zeitgemäßen Unterrichtsgestaltung. München PimS-Verlag1992.

- Seitz, Matthias: Speicherprogrammierbare Steuerungen. System- und Programmentwurf für die Fabrik- und Prozessautomatisierung, vertikale Integration. 2. Auflage. München: Hanser2008.
- Serve, Helmut J.: Das Unterrichtsprinzip der Motivierung. In: Prinzipien guten Unterrichts. Hrsg. Seibert und Serve. München: PimS1992.
- Straka, Gerald/ Macke, Gerd: Lern-Lehr-Theoretische Didaktik. 4. Auflage. Münster: Waxmann Verlag2006.
- Strecker, Kerstin Maike: Informatik für Alle – Wie viel Programmierung braucht der Mensch. Dissertation zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades. Göttingen: 2009.
- Timmerhaus, Winfried: Fachdidaktik als konstitutives Element universitärer Lehrerbildung. Bestandsaufnahmen, Analysen und Konzeptionen aus erziehungswissenschaftlicher Perspektive. Marburg: Tectum-Verlag2001.
- Valk, Laurens: LEGO® - ROBOTER. Bauen und programmieren mit LEGO® MINDSTORMS® NXT 2.0. Heidelberg dpunkt.Verlag2011.
- Völklein, Friedemann/Zetterer, Thomas: Praxiswissen Mikrotechnik. Grundlagen Technologien Anwendungen. 2. Auflage. Wiesbaden. Vieweg & Sohn Verlag2006.
- Weigend, Michael: Algorithmik in der Grundschule. In: Zukunft braucht Herkunft. Hrsg. von Bernhard Koerber. Bonn: Kölken2009.
- Werner, Dieter: Taschenbuch der Informatik. München: Hanser Verlag2007.
- Winkel, Jens: Gedankenstürme – LEGO-Mindstorms im Unterricht. Informationen über die Einsatzmöglichkeiten von LEGO Mindstorms im Unterricht und Erfahrungen aus der Praxis. (28/05/2003). <http://www.lehrer-online.de/mindstorms.php> (Stand: 27.07.2011).

ANHANG

Makroebene.....	65
Mikroebene.....	69
Arbeitsblätter: Haussteuerung mit Scratch.....	77
PPT Kurze Einführung in Scratch.....	90
Normen bei Bewertung.....	97
Evaluation der Projekte.....	98
Evaluationsbogen.....	99
Arbeitsblätter „Überprüfe dich selbst“.....	100

Makroebene der Unterrichtsreihe zum Thema „Haussteuereung mit Scratch“

Stunden	Makrostruktur	Bildungsstandards (allg.)	Bildungsstandards 8.-10. Klasse
1. Doppelstunde	Einführung in Scratch	<p>SuS</p> <ul style="list-style-type: none"> - entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen - verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweisen - wenden Informatiksysteme zielgerichtet an - kommunizieren fachgerecht über informative Sachverhalte - kooperieren bei Lösungen informatischer Probleme - veranschaulichen informatischer Sachverhalte - wählen geeignete Darstellungsformen aus 	<p>SuS</p> <ul style="list-style-type: none"> - verwenden Variablen und Wertzuweisungen - geben Problemlösungen in einer (...) Programmiersprache an - interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv - erschließen sich selbstständig neue Anwendungen und Informatiksysteme - verwenden bei der Implementierung algorithmische Grundbausteine - kommunizieren mündlich strukturiert über informative Sachverhalte - wählen eine Darstellungsform auf der Basis allgemein akzeptierter und zweckdienlicher Kriterien aus
2. Doppelstunde	Haussteuerung (Zielbestimmung der UR)	<p>SuS</p> <ul style="list-style-type: none"> - entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar - verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweisen - stellen Fragen und äußern Vermutungen über informative Sachverhalte - erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik - kommunizieren fachgerecht über informative Sachverhalte 	<p>SuS</p> <ul style="list-style-type: none"> - überprüfen die wesentlichen Eigenschaften von Algorithmen - stellen die algorithmischen Grundbausteine formal dar - interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv - verknüpfen informative Inhalte und Vorgehensweisen mit solchen außerhalb der Informatik - kommunizieren mündliche strukturiert über informative Sachverhalte

3. Doppelstunde	Haussteuerung (Kontext)	<p>SuS</p> <ul style="list-style-type: none"> - verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweisen - benennen Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung - nehmen Entscheidungsfreiheiten im Umgang mit Informatiksystemen wahr und handeln in Übereinstimmung mit gesellschaftlichen Normen - reagieren auf Risiken bei der Nutzung von Informatiksystemen - stellen Fragen und Äußern Vermutungen über informative Sachverhalte - erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik - kommunizieren fachgerecht über informative Sachverhalte - nutzen geeignete Werkzeuge zur Kommunikation und Kooperation 	<p>SuS</p> <ul style="list-style-type: none"> - stellen die Veränderungen des eigenen Handelns in Schule und Freizeit dar - kommentieren automatisierte Vorgänge und beurteilen deren Umsetzung - bewerten die Auswirkungen der Automatisierung in der Arbeitswelt - stellen Vermutungen über Zusammenhänge und Lösungsmöglichkeiten im informatischen Kontext dar - stützen ihre Argumente auf erworbenes Fachwissen - formulieren angemessene Bewertungskriterien und wenden diese an - gewichten verschiedene Kriterien und bewerten deren Brauchbarkeit für das eigene Handeln - kommunizieren mündlich strukturiert über informative Sachverhalte - wählen eine Darstellungsform auf der Basis allgemein akzeptierter und zweckdienlicher Kriterien aus
4. Doppelstunde	Haussteuerung (mit Scratch)	<p>SuS</p> <ul style="list-style-type: none"> - kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsbereichen und lesen und interpretieren gegebene Algorithmen - entwerfen und realisieren Algorithmen mit algorithmischen Grundbausteinen - verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren 	<p>SuS</p> <ul style="list-style-type: none"> - kennen und verwenden die Datentypen Text, Zahl und Wahrheitswert - kennen und verwenden arithmetische und logische Operationen - verwenden Variablen und Wertzuweisungen - entwerfen, implementieren und beurteilen Algorithmen - interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv

		<p>Funktionsweisen</p> <ul style="list-style-type: none"> - wenden Informatiksysteme zielgerichtet an - erschließen sich weitere Informatiksysteme - erstellen informative Modelle zu gegebenen Sachverhalten - stellen Fragen und äußern Vermutungen über informative Sachverhalte - kommunizieren fachgerecht über informative Sachverhalte - kooperieren bei der Lösung informatischer Probleme 	<ul style="list-style-type: none"> - klassifizieren Hardware und Software Erweitern bestehende Informatiksysteme mit Soft- und Hardwarekomponenten Erschließen sich selbstständig neue Anwendungen und Informatiksysteme - verwenden bei der Implementierung die algorithmischen Grundbausteine - kommunizieren mündlich strukturiert über informative Sachverhalte
5. Doppelstunde/ 6.Doppelstunde	Haussteuerung (Projekt)	<p>SuS</p> <ul style="list-style-type: none"> - kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsbereichen und lesen und interpretieren gegebene Algorithmen - entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen - verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweisen - wenden Informatiksysteme zielgerichtet an - stellen Fragen und äußern Vermutungen über informative Sachverhalte - begründen Entscheidungen bei der Nutzung von Informatiksystemen - kommunizieren fachgerecht über informative Sachverhalte - kooperieren bei der Lösung informatischer Probleme 	<p>SuS</p> <ul style="list-style-type: none"> - kennen und verwenden arithmetische und logische Operationen - entwerfen, implementieren und beurteilen Algorithmen - interpretieren Fehlermeldungen bei der Arbeit mit Informatiksystemen und nutzen sie produktiv - verwenden bei der Implementierung die algorithmischen Grundbausteine - wählen begründet aus Alternativen aus - planen Handlungsabläufe und Handlungsfolgen - nutzen Analogien zwischen informatischen Inhalten oder Vorgehensweisen, um Neues mit Bekanntem zu verknüpfen - kommunizieren mündlich strukturiert über informative Sachverhalte - stellen informative Sachverhalte unter Benutzung der Fachsprache schriftlich sachgerecht dar - kooperieren in Projektarbeit bei der Bearbeitung eines informatischen Problems - dokumentieren Ablauf und Ergebnisse der Projektarbeit

		<ul style="list-style-type: none"> - veranschaulichen informatische Sachverhalte - wählen geeignete Darstellungsformen aus 	<ul style="list-style-type: none"> - reflektieren gemeinsam Ansatz, Ablauf und Ergebnis des Projekts - wählen eine Darstellungsform auf der Basis allgemein akzeptierter und zweckdienlicher Kriterien aus
7. Doppelstunde	Präsentation der Projekte	<p>SuS</p> <ul style="list-style-type: none"> - kommunizieren fachgerecht über informatische Sachverhalte - kooperieren bei der Lösung informatischer Probleme - nutzen geeignete Werkzeuge zur Kommunikation und Kooperation - interpretieren unterschiedliche Darstellungen von Sachverhalten - veranschaulichen informatische Sachverhalte 	<p>SuS</p> <ul style="list-style-type: none"> - kommunizieren mündlich strukturiert über informatische Sachverhalte - reflektieren gemeinsam Ansatz, Ablauf und Ergebnis des Projekts
8. Doppelstunde	Struktogramme Fazit	<p>SuS</p> <ul style="list-style-type: none"> - verstehen Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweisen - stellen Fragen und äußern Vermutungen über informatische Sachverhalte - erstellen informative Modelle zu gegebenen Sachverhalten - Reflektieren Modelle und deren Implementierung - kommunizieren fachgerecht über informatische Sachverhalte 	<p>SuS</p> <ul style="list-style-type: none"> - analysieren Sachverhalte und - erstellen netzartige Strukturen - nutzen Analogien zwischen informatischen Inhalten oder Vorgehensweisen, um Neues mit Bekanntem zu verknüpfen - begründen Vorgehensweisen bei der Modellierung informatischer Sachverhalte - kommunizieren mündlich strukturiert über informatische Sachverhalte - interpretieren Diagramme, Grafiken sowie Ergebnisdaten - gestalten Diagramme und Grafiken, um informative Sachverhalte zu beschreiben und mit andern darüber zu kommunizieren

1. Doppelstunde: Einführung in Scratch

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Einstiegsphase <i>5min</i>	Lernbereitschaft wecken		Filmvortrag (fertiges Scratchprodukt)	Präsentation	Beamer
Erarbeitungsphase I <i>20-25min</i>	Über den bevorstehenden Lernprozess informieren, Handhabung des Lernprogramms erklären	SuS verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise	kurzer Überblick über die Möglichkeiten in Scratch	Lehrervortrag/ Beamerpräsentation	Beamer, Arbeitsplätze mit installiertem Scratch PPT „Kurze Einführung in Scratch“
Erarbeitungsphase II <i>60-65min</i>	Aktive Auseinandersetzung mit dem Gelernten, Hilfestellung anbieten,	- SuS entwerfen und realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar - SuS kooperieren bei der Lösung informatischer - SuS nutzen formale Sprachen zur Interaktion mit Informatiksystemen und zum Problemlösen	SuS erarbeiten eigenständig in Partnerarbeit ein Scratchprogramm zum Thema "Mein erstes Scratchprogramm" Experimentieren eindeutig erwünscht!	Partnerarbeit/ Lehrer gibt Hilfestellungen	Arbeitsplätze mit installiertem Scratch AB „Kannst du das auch?“

Hausaufgabe: Vervollständigen des Lernprogramms. (Hinweise zur Installation von Scratch für Zuhause werden gegeben)

2. Doppelstunde: Haussteuerung

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Einstiegsphase <i>5min</i>			Aufbau der Programme		
Auswertungsphase <i>15min</i>	Sammeln von Lernergebnissen	SuS kommunizieren fachgerecht über informative Sachverhalte	SuS erklären den Mitschülern ihr Programm und beschreiben ihre Vorgehensweise	Schülerpräsentation	Beamer
Erarbeitungsphase I <i>30min</i>	Ideen sammeln für weiteres Vorgehen, Nutzung kognitiver Werkzeuge ermöglichen,	SuS erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik	Erstellen einer Mind Map zum Thema Haussteuerung	Einzelarbeit	Internetzugang, Fachbücher AB „Arbeitsauftrag“
Auswertungsphase II <i>30min</i>	Sammeln der Lernergebnisse,	SuS stellen Fragen und äußern Vermutungen über informative Sachverhalte	Anlegen einer gemeinsamen Mind Map	Plenum	Beamer Mind Map Manager
Einführungsphase <i>10min</i>	Ausgangssituation darstellen	SuS verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise	Einführung in Sensoren und Aktoren	Lehrervortrag, Plenum, Schülerfragen	AB „Hausaufgabe“
<u>Hausaufgabe:</u> Arbeitsblatt zu Sensoren und Aktoren					

3. Doppelstunde: Hausteuerung im Kontext

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Einstiegsphase <i>15min</i>	Kontrolle und Beurteilung des Lernerfolgs,	SuS verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise	Besprechung der Hausaufgabe Sensoren und Aktoren	Plenum	Tafelbild, Heft
Erarbeitungsphase <i>45min</i>	Vorwissen aktivieren, Ideen sammeln für weiteres Vorgehen, Abstrakte Informationen veranschaulichen	SuS benennen Wechselwirkungen zwischen Informatiksystemen und ihrer gesellschaftlichen Einbettung	Gruppenarbeit zur Haussteuerung	Gruppenarbeit	Internetzugang, Plakat, AB „Haussteuerung – eine sinnvolle Investition“
Auswertungsphase <i>30 min</i>	Sammeln von Lernergebnissen, Festigung des Gelernten,	SuS bewerten die Auswirkungen der Automatisierung in der Arbeitswelt (Haushalt) SuS kommentieren automatisierte Vorgänge und beurteilen deren Umsetzung	Pro- und Contra Diskussion mit Unterstützung des Plakats	Diskussion der beiden Wortführer (Wortstreit)	Plakate, Tafelbild, Heft

4. Doppelstunde: Hausteuerung mit Scratch (Picoboard/Adruino)

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Einführungsphase <i>30min</i>	Lernbereitschaft wecken, Ausgangssituation darstellen	SuS erschließen sich weitere Informatiksysteme, SuS klassifizieren Hardware und Software	Haussteuerung mit Scratch, Vorstellen der Komponenten	Lehrervortrag	Picoboard, Adruino, Miniwelt, Sensoren und Aktoren,
Erarbeitungsphase <i>35min</i>	Erkundendes Verhalten und Ausprobieren von Hypothesen ermöglichen	SuS kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten und lesen und interpretieren gegebene Algorithmen	SuS sollen sich mit der neuen Hardware auseinandersetzen. Zur Erleichterung können Beispielprogramme abgerufen werden.	Partnerarbeit	Picoboard, Adruino, Miniwelt, Sensoren und Aktoren, AB „Picoboard“ AB „Adruino“ Beispielprogramme zur Haussteuerung
Auswertungsphase <i>25min</i>	Vernetzung der Inhalte, Festigung des Gelernten		Klärung von Fragen, Lehrer gibt Anregungen Aufbau technischer Systeme: EVA-Prinzip	Plenum Frontalunterricht	Tafel, Heft,

5. Doppelstunde: Projektarbeit "Haussteuerung mit Scratch"

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Projektarbeit <i>90min</i>	Erkundendes Verhalten und Ausprobieren von Hypothesen ermöglichen, Nutzung kognitiver Werkzeuge ermöglichen, Transfer vorbereiten	SuS realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar, SuS erschließen sich weitere Informatiksysteme, SuS kommunizieren fachgerecht über informatische Inhalte, SuS kooperieren bei der Lösung informatischer Probleme,	Projektarbeit zum Thema: Haussteuerung Bildung von 4 Experten pro Gruppe: Lerntagebuch, Scratch, Adruino/Picoboard, Ideensammler	Projektarbeit in 4er Gruppen, Beratung durch den Lehrer	Picoboard, Adruino, Miniwelt, Sensoren und Aktoren, AB „Projekt: Haussteuerung“ AB „Gruppenarbeit“

Hausaufgabe: Abstimmen der Arbeitsschritte mit der Gruppe (Aufgabenverteilung)! Vorbereiten der Präsentation zur 7. Doppelstunde!

6. Doppelstunde: Projektarbeit "Haussteuerung mit Scratch"

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Projektarbeit 90min	Erkundendes Verhalten und Ausprobieren von Hypothesen ermöglichen, Nutzung kognitiver Werkzeuge ermöglichen, Transfer vorbereiten	SuS realisieren Algorithmen mit den algorithmischen Grundbausteinen und stellen diese geeignet dar, SuS erschließen sich weitere Informatiksysteme, SuS kommunizieren fachgerecht über informatische Inhalte, SuS kooperieren bei der Lösung informatischer Probleme,	Projektarbeit zum Thema: Haussteuerung Bildung von 4 Experten pro Gruppe: Lerntagebuch, Scratch, Adruino/Picoboard, Ideensammler	Projektarbeit in 4er Gruppen, Beratung durch den Lehrer	Picoboard, Adruino, Miniwelt, Sensoren und Aktoren, AB „Projekt: Haussteuerung“ AB „Gruppenarbeit“ <i>Input: AB „Kontrollstrukturen“</i>
<u>Hausaufgabe:</u> Fertigstellen der Arbeitsaufträge! Zur nächsten Stunde präsentationsbereit sein!					

7. Doppelstunde: Vorstellen der Ergebnisse der Projektarbeit

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Eingangsphase <i>15min</i>	Ausgangssituation darstellen		Klärung von Normen zur mündlichen Stellungnahme	Lehrervortrag	AB „Normen für die mündliche Stellungnahme“
Präsentationsphase <i>75min</i>	Sammeln von Lernergebnissen, Kontrolle und Beurteilung des Lernerfolgs	SuS wählen geeignete Darstellungsformen aus, SuS veranschaulichen informative Sachverhalte	Schüler präsentieren ihre Haussteuerungen, Evaluation durch SuS (Mündlich/Schriftlich)		PC, Beamer, AB „Evaluation der Projekte“
<u>Hausaufgabe:</u> Nochmaliges Lesen der Inputblätter!					

8. Doppelstunde: Struktogramme

Zeit/ Phase	Didaktische Funktion	Lernziele (Standards) Kompetenzen	Inhalt	Sozialform	Material
Erarbeitungsphasenphase <i>50min</i>	Vernetzung der Inhalte, Festigung des Gelernten, Weiterlernen ermöglichen,	SuS erkennen und nutzen Verbindungen innerhalb und außerhalb der Informatik, SuS nutzen Analogien zwischen informatischen Inhalten oder Vorgehensweisen, um Neues mit Bekanntem zu verknüpfen, SuS stellen Fragen und äußern Vermutungen über informative Sachverhalte,	Klären von Programmierstrukturen, Analogiebildung zu Scratch,	Plenum	AB „Struktogramme“ AB „Struktogramme II“ AB „Bit's Programm“
Resümee <i>20min</i>	Sammeln von Lernergebnissen		Rückblick auf die Inhalte der Unterrichtsreihe (Zusammenfassung)	Einzel/Partnerarbeit (MindMap) Plenumsdiskussion	Heft Tafelbild
Abschluss <i>20min</i>	Transparenz schaffen		Evaluation der Unterrichtsreihe aus Schülersicht	Schülerbeiträge (Mündlich/Schriftlich)	AB „Evaluationsbogen“

Haussteuerung mit Scratch



**KONZEPTION EINER KONTEXT-
ORTIENTIERTEN
UNTERRICHTSREIHE IM FACH
INFORMATIK ZUM THEMA
HAUSSTEUERUNG**

Kannst du das auch?



Bewege ein Objekt auf der Bühne!

Lasse das Objekt ein Viereck abschreiten!

Nutze hierzu die vorgegeben
Programmierstrukturen.

Erstelle mit einem Partner eine lustige
oder spannende Animation.

Präsentiert diese der Klasse!



Du kannst auch mehrere
Bewegungen hintereinander
schalten –
das sieht dann
aus wie ein Tanz!

Arbeitsauftrag



Was ist eine Haussteuerung?

Fertige zunächst eine Mind-Map zum Thema an, arbeite hierzu rein assoziativ.

Suche dann nach Informationen zum Thema „Haussteuerung“ - nutze Lexika, Fachbücher oder das Internet!

Notiere diese stichpunktartig unter die Mind-Map!



Haussteuerung, was soll denn
das sein Byte?

Meinst du eine

Haussteuerung oder eine

Ist das vielleicht dasselbe? **Hausautomation?**



Hausaufgabe



In einer Haussteuerung nehmen Sensoren und Aktoren einen besonderen Stellenwert ein.

Sensoren (z.B. Temperaturmesser) erzeugen dabei immer die Eingabedaten und Aktoren (z.B. Motor) werden durch Ausgabedaten gesteuert.

Lies dir den Text zu der Funktionsweise von Waschmaschinen durch.

Beschreibe die Funktionsweisen von Sensoren und Aktoren in einer Waschmaschine!

Nutze die Materialien dazu!

Welche verschiedenen Sensoren und Aktoren gibt es? Wozu sind sie gut und wie werden sie eingesetzt



Du kannst auch mal im Internet nach Lehrvideos suchen.

Haussteuerung – Eine sinnvolle Investition?



Verteilt euch gleichmäßig auf 2 Gruppen und erarbeitet folgenden Sachzusammenhang:

Gruppe I: Findet möglichst viele Argumente **für** die Installation einer Haussteuerung. Denkt dabei an Sicherheitsaspekte, Gefahren, Kosten, Nutzen, Aufwand. Informiert euch hierzu auch im Internet. Fertigt dazu ein Plakat an!

Gruppe II: Findet möglichst viele Argumente **gegen** die Installation einer Haussteuerung. Denkt dabei an Sicherheitsaspekte, Gefahren, Kosten, Nutzen, Aufwand. Informiert euch hierzu auch im Internet. Fertigt dazu ein Plakat an!

Findet jeweils einen Sprecher in eurer Gruppe. Am Ende wollen wir eine Pro- und Kontradiskussion führen!

Hey Byte! Also ich finde eine
Haussteuerung toll!



Ach ja? Also ich nicht!



Wieso denn das? Sie macht
doch alles leichter und
sicherer!

Für alles brauchen
Menschen Computer! Wenn du
mich fragst werden die ziemlich bequem...

Picoboard



Mit diesen Bausteinen kannst du das Picoboard in Scratch nutzen!

Wert von Sensor Regler ▾

Sensor Schaltfläche gedrückt ▾ ?

In Scratch werden die Sensoren des Picoboards über diese Module angesteuert.

Achte auf die Passform der Module.

Bei einer runden Passform gibt der Sensor einen *Zahlenwert* zwischen 0 und 100 zurück. Der Lichtsensor beispielsweise gibt die Intensität des Lichtes an. (0 = dunkel, 100 = hell)

Die sechseckige Passform enthält eine *Booleanvariable*. Bei dem Drucksensor beispielsweise gibt es nur die Möglichkeiten gedrückt oder nicht gedrückt. (wahr, falsch)



Drucksensor
Geräuschsensor
Lichtsensor
Widerstand
Regler

Adruino



schalte Motor für 1 Sekunden an
setze Motorkraft auf 100
schalte Motor an schalte Motor aus
setze Motornichtung auf in diese Richtung ▾

Über diese Bausteine
kannst du Adruino in
Scratch ansprechen!



Über das Adruinoboard werden die Motoren angesteuert.
Die Motorenbausteine müssen zunächst zugeschaltet
werden. Das machst du über Bearbeiten in der
Menüleiste.



So sieht das
Adruinoboard
aus!



Projekt: Haussteuerung



**Findet euch in Gruppen von 4 Personen zusammen.
Ihr arbeitet die nächsten 4 Stunden gemeinsam an
dem Projekt.**

**Gemeinsam sollt ihr eine Haussteuerung mit Hilfe
von Scratch simulieren.**

**Nutzt dabei eure Kenntnisse über das Programm
sowie die Informationen über Haussteuerungen
aus.**

**Was muss innerhalb eines Hauses gesteuert
werden? Was ist wichtig/unwichtig für eine
Steuerung? Welche Sensoren/Aktoren
braucht man?**

Nutzt das Picoboard bzw. Adruino!



**Um zum Ziel zu kommen habt ihr
alle unterschiedliche Aufgaben,
die alle zum Erfolg des Projekts beitragen!
Seht euch dazu die Einteilung
der Gruppen an!**

Gruppenarbeit



Jede Gruppe besteht aus 4 Experten für...

...Lerntagebuch

hält alles fest, was in der Gruppe erarbeitet wurde

...Experten für Scratch

ist für die Umsetzung in Scratch verantwortlich

...Experte für Picoboard/Adruino

ist mit der Hardware vertraut

...Ideensammler

überwacht die Gruppenarbeit und gibt neue
Denkanstöße

**Am Ende soll das Projekt der Klasse
präsentiert werden! Findet eine geeignete
Form für euren Kurzvortrag – er sollte
möglichst eindeutig, klar und präzise sein,
so dass euch alle verstehen!**

**Für die Präsentation habt ihr 5-10 Minuten
Zeit, versucht diese Vorgabe einzuhalten!**



Viel Spaß beim
Tüfteln!

Kontrollstrukturen



Wichtige Programmierstrukturen sind Anweisungen und Schleifen. Hier findest du if-Anweisungen und while-Schleifen.



In der „falls“-Anweisung musst du einen Wahrheitswert angeben. (wahr, falsch)



Interessieren dich diese if-Anweisungen und while-Schleifen? Gut, dann lies dir doch mal den Text dazu durch, vielleicht findest du dann noch andere interessante Fakten heraus!



Struktogramme



In der Informatik werden Programmabläufe unter anderem mit Struktogrammen dargestellt.
Hier die wichtigsten im Vergleich zu Scratch:

a := 6

setze a auf 6

gehe 10er Schritte
gehe 10er Schritte
gehe 10er Schritte

gehe 10 -er Schritt
gehe 10 -er Schritt
gehe 10 -er Schritt

Y X < Y N
"X ist kleiner als Y" "X ist größer als Y"

falls X < Y
sage X ist kleiner als Y
sonst
sage X ist größer als Y

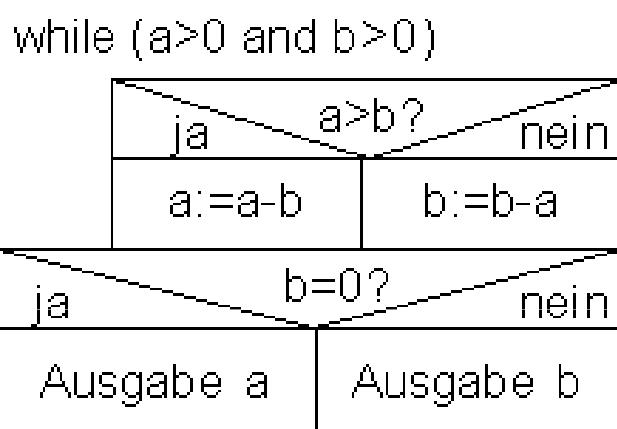
Wiederhole falls Maustaste gedrückt
gehe 10er Schritte

wiederhole fortläufend, falls Maustaste gedrückt?
gehe 10 -er Schritt

Struktogramme II



Struktogramme lassen sich beliebig verschachteln.



**Erkennst du was dieses Programm macht?
Probiere doch mal was passiert wenn $a = 6$
und $b = 3$ ist.**



Ich habe auch ein Programm
dazu geschrieben.
Du kannst es nutzen, wenn du willst!

Bit's Programm



A Scratch script consisting of the following blocks:

- setze a auf 6
- setze b auf 3
- wiederhole bis $b = 0$
 - falls $a > b$
 - setze a auf $a - b$
 - sonst
 - setze b auf $b - a$
- sage a für 2 Sek.

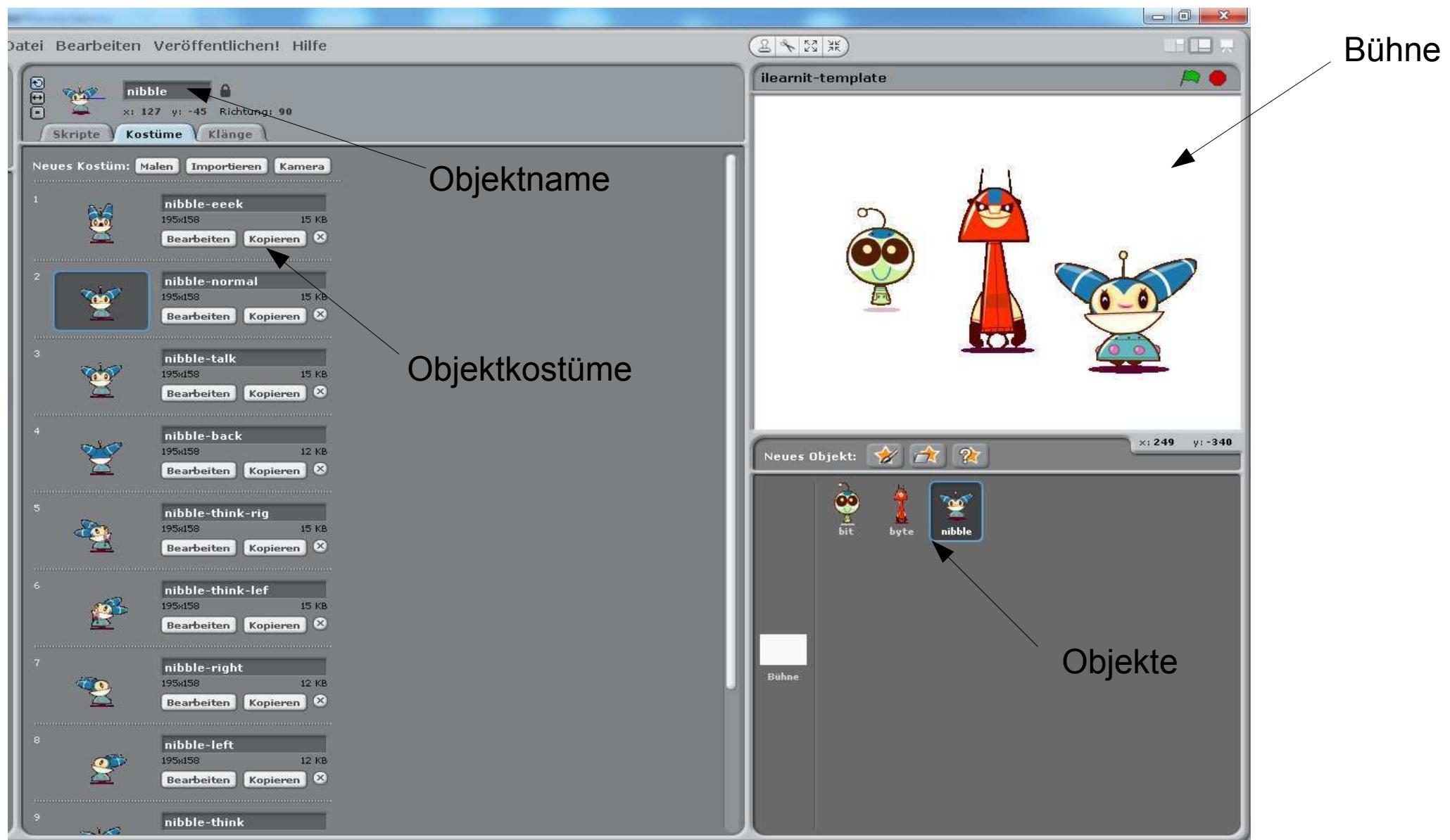


Toll oder? Aber verstehst du das auch?
Schreib doch einfach neben das Programm,
was es an der Stelle tut –
das macht es leichter!

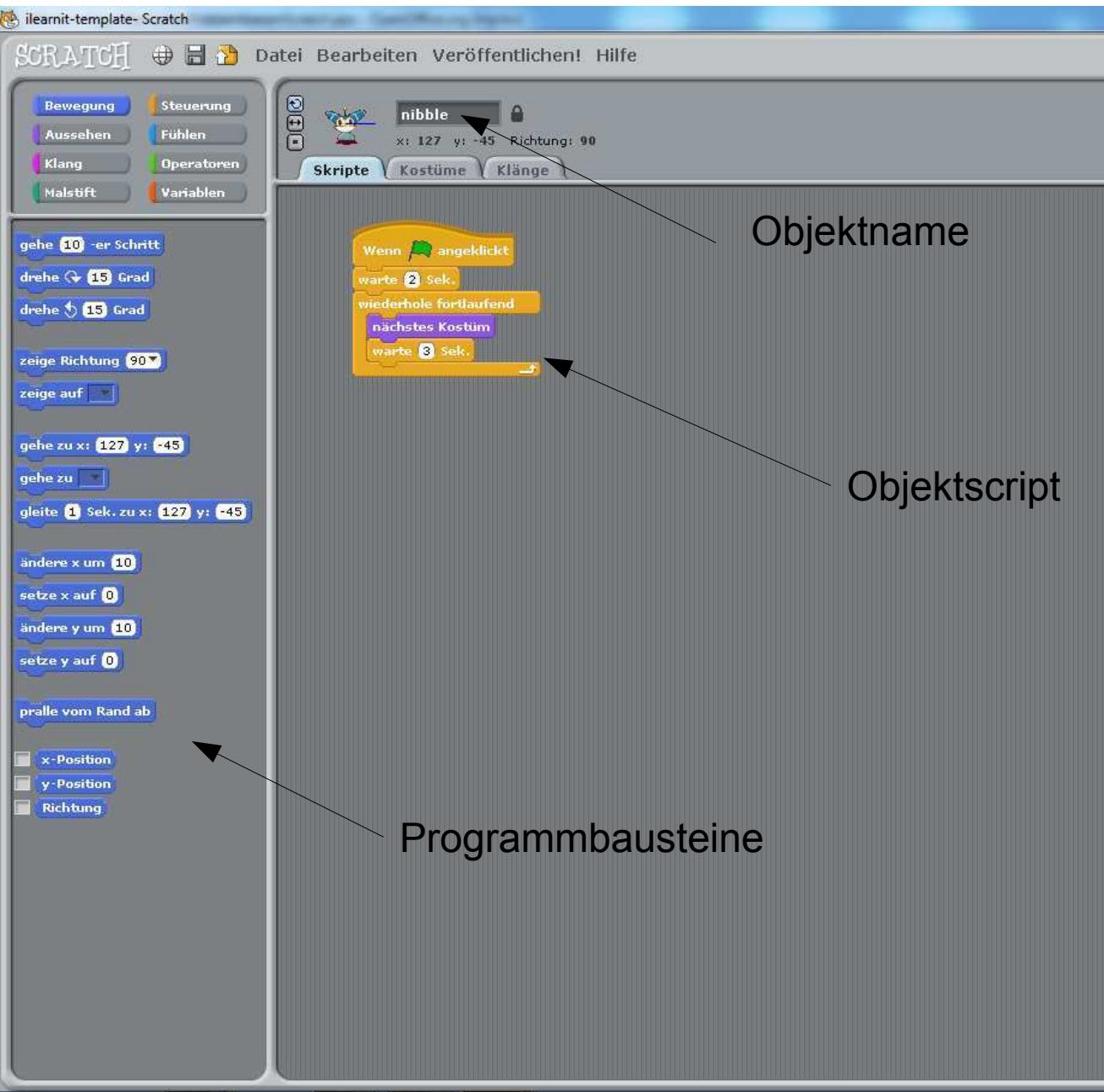
Kurze Einführung in Scratch



Die Welt von Nibble, Bit und Byte

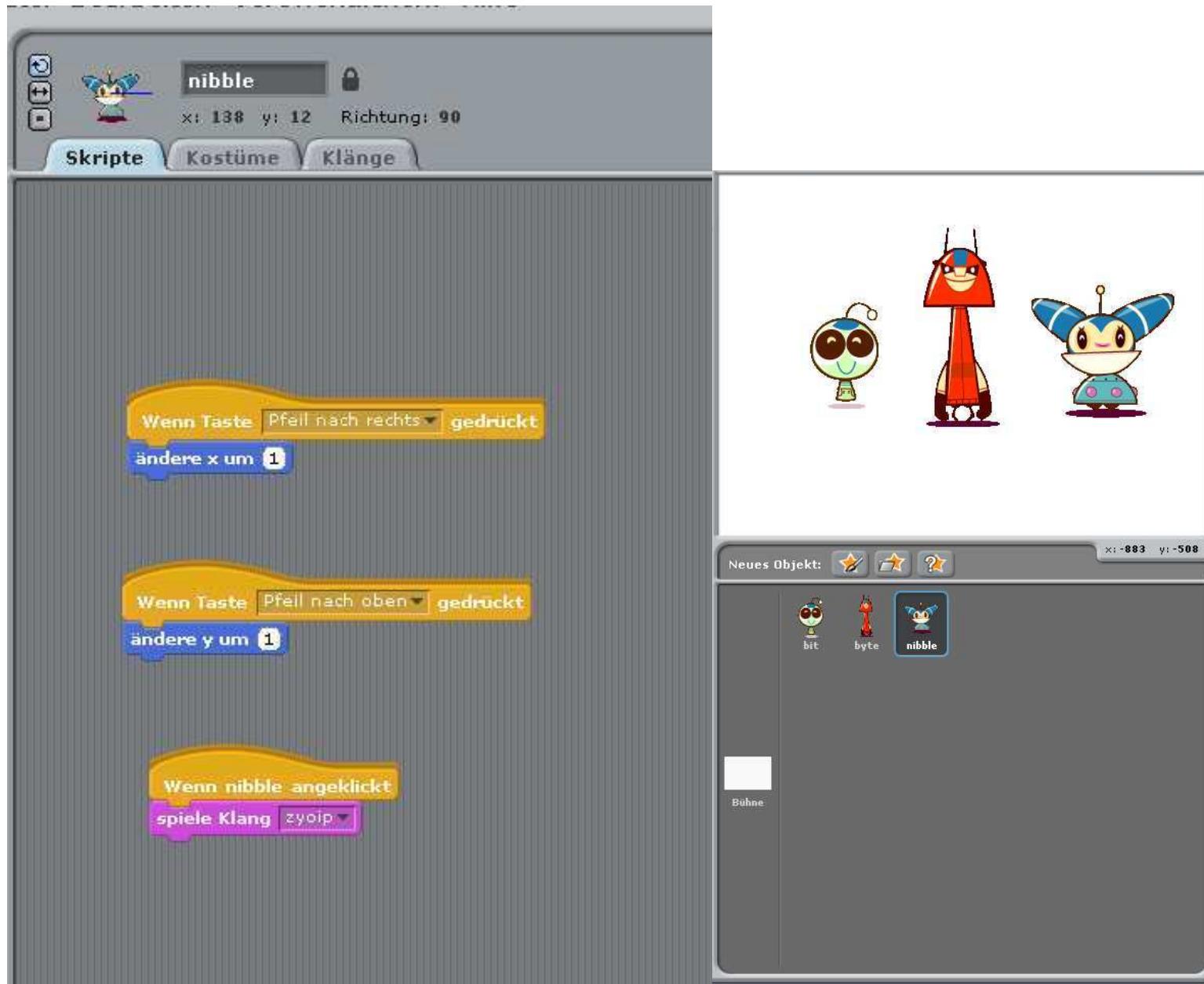


Das Objektskript



In diesem Script werden die Aktionen vom Objekt Nibble festgehalten.
Die Programmbausteine sind links zu finden.

Objekte sollen Aktionen ausführen



Ereignisse lösen Aktionen aus.
Ereignisse können sein:

- Taste gedrückt
- Objekt angeklickt
- Nachricht empfangen
- die grüne Fahne wird angeklickt

Fallunterscheidungen

Nibble soll auf Pfeil nach rechts 10 Schritte gehen, es sei denn er hat den Rand erreicht. Dann soll er sagen: „Ich bin am Rand angelangt.“



Wiederholungen

Soll Nibble direkt bis zum Rand laufen, nutzt man eine Wiederholung.



Arbeitsauftrag

Erstelle nun ein eigenes Programm in Scratch und versuche dabei die Objekte in Interaktion treten zu lassen!



Viel Spaß beim Ausprobieren!

Normen für die mündliche Stellungnahme

Dieses Merkblatt soll euch helfen Leitungen innerhalb eurer Schülergruppe zu bewerten.

Mit einer fairen und kompetenten Bewertung leistet ihr einen Beitrag zur Qualitätssteigerung.

Wichtige Regeln für die mündliche Bewertung:

- Kritik: Erst positive Aspekte benennen, dann negative Aspekte benennen!
 Negative Aspekte immer als Verbesserungsmöglichkeiten formulieren!
 Die Kritik klar und deutlich begründen!
 Nicht die Person/Gruppe beurteilen, sondern das Ergebnis!
 Nicht persönlich, herablassend oder verletzend sprechen!

Punkte die euch helfen können einen mündlichen Beitrag zur Bewertung zu leisten:

- ... hat Sachverhalte gut / sachkompetent und verständlich dargestellt und graphisch aufbereitet.
- ... hat das Thema gut analysiert und wichtige Punkte herausgearbeitet.
- ... hat sich treffende aktuelle Informationen besorgt.
- ... hat Informationen systematisch ausgewertet und zusammengefasst.
- ... hat einen kreativen Weg des Umgangs mit der Aufgabe gefunden.
- ... hat die Aufgabe treffend gelöst.
- ... hat Materialien und Geräte zur Präsentation systematisch vorbereitet.

Gebt eure Beiträge immer in einer Form ab, in der ihr selbst beurteilt werden wolltet.



Evaluation der Projekte

Gruppe: _____

Benote die Vorträge der Projekte durch deine Mitschüler nach den folgenden Punkten:

	1	2	3	4	5	6
Präsentationsform (Angemessene Wahl des Mediums)						
Art der Präsentation (Lautstärke/Geschwindigkeit/Wortwahl)						
Struktur des Vortrag (Verständnis/Kann ich dem Vortagenden folgen?)						
Inhalt des Vortrags (Anzahl der Umsetzungsmöglichkeiten/ Fachliche Richtigkeit)						
Kreativität der Umsetzungsform (Besonderheiten/Kreativer Umgang mit der Aufgabe)						

Weitere Bemerkungen zu dem Vortrag:



Evaluationsbogen

Benote die Unterrichtsreihe zu den verschiedenen Aspekten von 1 (trifft voll zu) bis 6 (trifft gar nicht zu).

	1	2	3	4	5	6
Das Unterrichtsthema "Haussteuerung" hat mir sehr gefallen						
Die Inhalte waren durchgängig gut verständlich						
Die Arbeit in Gruppen sagte mir zu						
Ich arbeite gerne in Projekten						
Die handwerklichen Arbeiten machten mir Spaß						
Die Programmierung mit Scratch hat mir gefallen						
Ich hätte gerne mehr mit Scratch gearbeitet						
Ich hätte gerne mehr zum Kontext Haussteuerung erfahren						
Das Präsentieren der Inhalte hat mir nichts ausgemacht						
Ich habe methodisch viel gelernt						
Ich habe inhaltlich viel gelernt						
Das Recherchieren von Informationen ist mir leicht gefallen						
Ich hätte mir mehr Hilfestellungen vom Lehrer gewünscht						

Am besten hat mir gefallen:

An dieser Stelle habe ich mich überfordert gefühlt:

An dieser Stelle habe ich mich unterfordert gefühlt:

Verbesserungsvorschläge/Lob/Kritik:

Überprüfe dich selbst



**MIT DIESER ZUSAMMENFASSUNG
KANNST DU DEIN WISSEN
NOCHMALΣ ÜBERPRÜFEN UND
FESTIGEN!**

**NUTZE DIE ARBEITSBLÄTTER
ALS WIEDERHOLUNG!**

Haussteuerung



Was ist eine Haussteuerung?



Bild: io-homecontrol. www.hausgarten.net

Welche Komponenten lassen sich in einem Haus steuern?

Kontrollstrukturen



Kontrollstrukturen ermöglichen dem Programmierer, in Abhängigkeit von Bedingungen, den Programmablauf zu beeinflussen. Zu Kontrollstrukturen, die in allen höheren Programmiersprachen verfügbar sind, zählt insbesondere die bedingte Verarbeitung von Kommandos (if-Anweisung) sowie die wiederholte Bearbeitung von Kommandos in Programmschleifen (z.B. while-Anweisungen). Fast alle Kontrollstrukturen arbeiten in Abhängigkeit von Bedingungen. Eine Bedingung ist ein Ausdruck wie z.B. $5 < 7$, der als Resultat einen Booleschen Wert liefert, also TRUE oder FALSE sein kann.

Aus: Krienke, Rainer: Programmieren in Perl. 2.Auflage. München/Wien:
HanserVerlag2002, S.59.

Erkläre mit deinen eigenen Worten, was Kontrollstrukturen sind!



If-Anweisung



Erkläre den Aufbau einer if-Anweisung!

Die Mehrzahl der Kontrollstrukturen enthält als wesentlichen Bestandteil eine Bedingung, die über die weitere Verarbeitung der in der Kontrollstruktur enthaltene Anweisungen entscheidet.

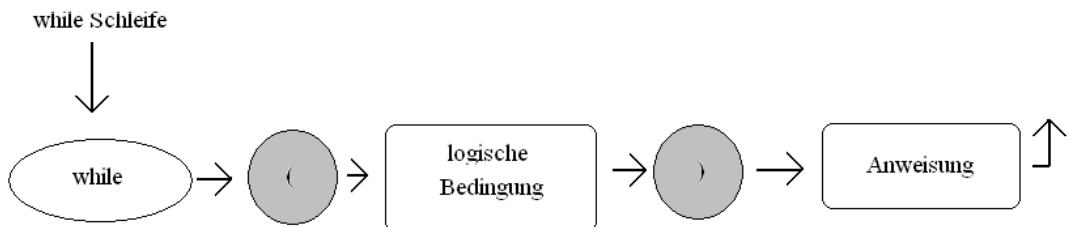
In einer if-Anweisung entscheidet z.B. eine Bedingung darüber, welche der beiden Gruppen von Anweisungen ausgeführt wird.

While-Schleife



Zuerst wird die Bedingung ausgewertet. Sollte diese wahr sein, wird die Anweisung, auch Rumpf der Schleife genannt, ausgeführt, an den Anfang der Schleife zurückgegangen und auf dieselbe Weise verfahren. Andernfalls ist die Ausführung der while-Schleife abgeschlossen und es wird mit der ersten Anweisung nach der while-Schleife fortgefahrene.

Aus: Müller, Heinrich/Weichert, Frank: Vorkurs Informatik: Der Einstieg ins Informatikstudium.
Wiesbaden: Teubner2005, S.68.



Was macht eine while-Schleife?



Sensoren



Ein- und Ausgabe über Sensoren und Aktoren

Sensoren nehmen Informationen über den Zustand eines technischen Prozesses durch Messung einer physikalischen Größe auf und leiten diese über die Messperipherie zum Computer.

Werner, Dieter: Taschenbuch der Informatik. München: Hanser Verlag 2007, S.650.

Welche Sensoren kennst du?

Welche Sensoren sind für eine Haussteuerung nützlich?



Aktoren



Ein- und Ausgabe über Sensoren und Aktoren

Will der Computer den technischen Prozess aktiv beeinflussen, so gibt er Informationen an die Stellperipherie aus, welche über Aktoren in den Prozess eingreift.

Werner, Dieter: Taschenbuch der Informatik. München: Hanser Verlag 2007, S.650.

Welche Aktoren kennst du?



Welche Aktoren sind für eine Haussteuerung nützlich?



Struktogramm



Entwirf ein Struktogramm für den Vorgang des Anmeldens bei deiner Email-Adresse.

Unter welchen Umständen klappt es und wann nicht?!
Welche Unterscheidung (wahr/falsch) ist zu machen?
Was sind die Konsequenzen?

