Studienseminar 2007/09 am Adam-Kraft-Gymnasium Schwabach

Schriftliche Hausarbeit für das Lehramt an Gymnasien in Bayern

Entwicklung und Bewertung einer
Unterrichtssequenz zum Thema Algorithmen
im Informatikunterricht der 7. Jahrgangsstufe
mit Hilfe der Programmierumgebung

Verfasserin: StRefin Susanne Hoika

Fach: Informatik

Erstgutachter: StR Volker Denke

Vorwort

Ich möchte mich bei allen, die mich bei der Entstehung dieser Arbeit unterstützt haben, für ihre Hilfe, ihr Entgegenkommen und ihr Verständnis ganz herzlich bedanken.

Ganz herzlich möchte ich mich bei meinem Lebensgefährten und meinen Eltern, die mir während des Referendariats immer zur Seite gestanden haben, bedanken.

Besonderer Dank gilt meinen Betreuungslehrern im Zweigschuleinsatz - OStR Thomas Platzöder vom Dientzenhofer-Gymnasium Bamberg und OStR Jens Heublein vom Kaiser-Heinrich-Gymnasium Bamberg -, die mir u. a. wertvolle Tipps für die Erstellung und Korrektur der Stegreifaufgaben gegeben haben.

Bayreuth, im Februar 2009

Inhaltsverzeichnis

1	Ein	leitung	r 5	6			
2	Fachwissenschaftliche Analyse						
	2.1	·					
		2.1.1	Oberfläche	9			
		2.1.2	Kontrollstrukturen	12			
		2.1.3	Bedingungen				
		2.1.4	Methodenaufrufe				
		2.1.5	Variablen	19			
		2.1.6	Listen				
		2.1.7	Nebenläufigkeit	23			
		2.1.8	Ereignisverwaltung (Event-Handling)				
3	Did	aktiscl	he und methodische Überlegungen	24			
	3.1		oraussetzungen	24			
	3.2		iele				
	3.3	Didak	tische Überlegungen zu den Lerninhalten	24			
		3.3.1	Kontrollstrukturen, Bedingungen und Methodenaufrufe				
		3.3.2	Struktogramme	25			
		3.3.3	Variablen	26			
		3.3.4	Listen	29			
		3.3.5	Nebenläufigkeiten	29			
		3.3.6	Ereignisverwaltung (Event-Handling)				
	3.4	Die Pr	rogrammierumgebung				
		3.4.1		30			
		3.4.2	Beurteilung von Scratch	30			
	3.5	Metho	odische Vorüberlegungen	33			
	3.6		ufgaben				
	3.7		ische Vorüberlegungen				
		3.7.1					
		3.7.2	Kaiser-Heinrich-Gymnasium Bamberg				
4	Vor	gehen	im Unterricht	39			
	4.1	_	neine Anmerkungen	39			
	4.2		reibung von Abläufen - 1	41			
		4.2.1	Lernziele	41			
		4.2.2	Motivation	41			
		4.2.3	Problemfrage	41			
		121		49			

INHALTSVERZEICHNIS

	4.2.5	Sicherung	43
	4.2.6	Hausaufgabe	43
	4.2.7	Begründung und Evaluation	43
4.3	Beschr	reibung von Abläufen - 2	45
	4.3.1	Lernziele	45
	4.3.2	Verbesserung der Hausaufgabe und Wiederholung	45
	4.3.3	Motivation	45
	4.3.4	Problemfrage 1	45
	4.3.5	Erarbeitung 1	46
	4.3.6	Sicherung 1	46
	4.3.7	Problemfrage 2	46
	4.3.8	Erarbeitung 2	46
	4.3.9	Sicherung $\stackrel{\circ}{2}$	46
	4.3.10	Problemfrage 3	46
	4.3.11		46
	4.3.12	Sicherung 3	46
		Problemfrage 4	47
		Erarbeitung 4	47
		Sicherung 4	47
4.4		enlernen von Scratch	48
	4.4.1	Lernziele	48
	4.4.2	Wiederholung	48
	4.4.3	Einführung in Scratch	48
	4.4.4	Arbeitsauftrag	48
	4.4.5	Begründung	49
4.5	Kenne	enlernen von Scratch und Sequenz	50
	4.5.1	Lernziele	50
	4.5.2	Motivation	50
	4.5.3	Erarbeitung	50
	4.5.4	Sicherung	51
	4.5.5	Fortsetzung des Arbeitsauftrages und Präsentation der	
		Ergebnisse	51
	4.5.6	Begründung	51
4.6	Wiede	rholung mit fester Anzahl	52
	4.6.1	Lernziele	52
	4.6.2	Motivation	52
	4.6.3	Arbeitsauftrag	52
	4.6.4	Problemfrage	53
	4.6.5	Erarbeitung	53
	4.6.6	Sicherung	53
	4.6.7	Arbeitsauftrag	54

INHALTSVERZEICHNIS

	4.6.8	Begründung
4.7	Versch	achtelte Wiederholungen mit fester Anzahl
	4.7.1	Lernziele
	4.7.2	Arbeitsauftrag
	4.7.3	Begründung und Evaluation
4.8	Beding	gte Wiederholung
	4.8.1	Lernziele
	4.8.2	Motivation
	4.8.3	Problemfrage
	4.8.4	Erarbeitung
	4.8.5	Sicherung
	4.8.6	Arbeitsauftrag
4.9	Zweise	itig bedingte Anweisung 61
	4.9.1	Lernziele
	4.9.2	Motivation
	4.9.3	Problemfrage
	4.9.4	Erarbeitung
	4.9.5	Sicherung
	4.9.6	Arbeitsauftrag
	4.9.7	Begründung und Evaluation 63
4.10	Einseit	gig bedingte Anweisung
		Lernziele
		Motivation
	4.10.3	Problemfrage
		Erarbeitung
		Sicherung
	4.10.6	Arbeitsauftrag
	4.10.7	Evaluation
4.11	Metho	den und Botschaften
		Lernziele
	4.11.2	Motivation
		Problemfrage
	4.11.4	Erarbeitung
	4.11.5	Sicherung
		Arbeitsauftrag
4.12		len (Exkurs)
	4.12.1	Lernziele
	4.12.2	Motivation
	4.12.3	Problemfrage
		Erarbeitung
	4.12.5	Sicherung

INHALTSVERZEICHNIS

	4.13	4.12.7 Spielpr 4.13.1 4.13.2 4.13.3	Arbeitsauftrag	 	 73 74 74 74 74 75
5	Leis	tungsn	achweise und Evaluation		78
	5.15.25.35.4	Klasse 5.1.1 5.1.2 Klasse 5.2.1 5.2.2 Klasse 5.3.1 5.3.2 Klasse 5.4.1	7d am Dientzenhofer-Gymnasium Bamberg Auswahl der Aufgaben	 	 79 79 82 82 86 86 86 86 89
	5.5	5.4.2 Evalua 5.5.1 5.5.2 5.5.3	Ergebnisse		 89 92 92 92 92
6	Fazi 6.1 6.2 6.3	Schulfe Ausblie	est am Dientzenhofer-Gymnasium Bamberg ck		94 94 95 96
7	Anh	ang			97
	Erkl	lärung			135

1 Einleitung

Das Thema Algorithmen ist im Natur und Technik-Unterricht der 7. Jahrgangsstufe im Schwerpunkt Informatik unter dem Lehrplanpunkt NT 7.2.3 "Beschreibung von Abläufen" zu finden. "Die Schüler lernen eines der wichtigsten Grundprinzipien der automatischen Informationsverarbeitung kennen und erhalten einen ersten Einblick in seine Anwendung."¹.

Für die praktische Arbeit innerhalb dieses Themengebiets gibt es inzwischen mehrere schülergerechte Programmierumgebungen:

- Robot Karol ²
- Kara für Java, JavaScript, Phython oder Ruby ³
- Hamstersimulator⁴
- eToys von Squeak ⁵ ⁶
- Scratch ⁷

Bei diesen Lösungen sehen die Schüler das Resultat ihres Programms nur als Reaktion von Objekten auf dem Bildschirm, erleben diese aber nicht real. Es böte sich an dieser Stelle noch die sehr kostspielige Möglichkeit des Einsatzes von Robotern wie LEGO Mindstorms (RCX), LEGO Mindstorms NXT oder fischertechnik Robotern an.

Das Staatsinstitut für Schulqualität und Bildungsforschung München (ISB) schreibt im Vorwort seiner Handreichung: "Während des gesamten Unterrichts im Schwerpunkt Informatik laufen der inhaltliche Strang "Umgang mit Information" und der methodische Strang "objektorientierte Sichtweise" parallel zueinander – sie ergänzen sich gegenseitig beim Verstehen der verwendeten Werkzeuge und der bearbeiteten EDV-Dokumente."⁸.

Es wird im Kapitel 3.3 der Handreichung die Verwendung der Programmierumgebung "Robot Karol" folgendermaßen begründet: "Das folgende Unterrichtskonzept verwendet als Werkzeug "Robot Karol" und

¹[12, NT 7.2.3]

²http://www.schule.bayern.de/karol/, letzter Zugriff: 12.01.2009

³http://www.swisseduc.ch/informatik/karatojava/, letzter Zugriff: 12.01.2009

⁴http://www.u-helmich.de/inf/java2h/seite01.html, letzter Zugriff: 12.01.2009

⁵http://www.squeakland.org/, letzter Zugriff: 12.01.2009

⁶http://www.squeak.de/, letzter Zugriff: 12.01.2009

⁷[19, letzter Zugriff: 12.01.2009]

⁸[13, Seite 1]

verfolgt damit das didaktische Prinzip der objektorientierten Sichtweise auch in der Algorithmik."⁹.

In dieser Programmierumgebung ist jedoch nur ein ROBOTER-Objekt vorhanden, das keine Beziehungen zu anderen Objekten besitzt. Die Objektorientierung beschränkt sich hier lediglich auf das Ausführen von Methoden und Verändern von Attributwerten eines einzelnen Objekts. Die objektorientierte Sichtweise ist bei dieser Programmierumgebung kaum ausgeprägt.

Dieses Problem wird mit der Programmierumgebung behoben. Neben diesem ausschlaggebenden Kritierum findet man noch weitere Punkte, die für den Einsatz von Scratch sprechen. Diese werden in Kapitel 2 näher erläutert.

Die hier vorliegende Unterrichtssequenz verwendet und wurde im 2. Halbjahr des Schuljahres 2007/08 am Dientzenhofer-Gymnasium Bamberg erstmals erprobt und im 1. Halbjahr des Schuljahres 2008/09 am Kaiser-Heinrich-Gymnasium Bamberg mit ein paar Änderungen nochmals durchgeführt.

Bei der Evaluation werden zuerst die Ergebnisse der jeweiligen Klassen im Einzelnen betrachtet, dann innerhalb des Gymnasiums und zuletzt innerhalb der zwei Bamberger Gymnasien verglichen.

Dabei ist vor allem interessant, ob das Abschneiden in den Leistungsnachweisen einen Zusammenhang mit den Ausbildungsrichtungen des jeweiligen Gymnasiums hat.

⁹[13, Seite 107]

2 Fachwissenschaftliche Analyse

2.1 Vorstellung von Scratch

Scraff ist eine freie Programmierumgebung, die von der Lifelong Kindergarten Group am MIT Media Lab entwickelt und finanziell von der National Science Foundation, von Microsoft, von der Intel Foundation, von Nokia und von den MIT Media Lab Research Consortia unterstützt wird. Sie basiert auf der plattformunabhängigen Smalltalk-Entwicklungsumgebung Squeak.

Die "interpretierte dynamische visuelle Programmiersprache" der die Objektorientierung zugrunde liegt, nennt sich - genauso wie die Entwicklungsumgebung, die 2007 erschienen ist, - Scratch und leitet sich von der DJ-Scratchtechnik ab.

"Unter Scratchen (engl. to scratch = kratzen) versteht man die Erzeugung von Tönen durch rhythmisches Hin- und Herbewegen einer laufenden Schallplatte auf einem Schallplattenspieler bei aufgelegter Nadel. Dabei können die Töne mit dem Crossfader des Mischpultes rhythmisch ein- und ausgeblendet werden, um diese zu neuen Melodien zusammenzufügen." ¹¹

Genauso wie beim musikalischen "Scratchen" kann man in Scratch alle Objekte, Grafiken und Töne in mehreren Programmen verwenden und jedes Mal neu kombinieren. Die Schüler können ihrer Fantasie freien Lauf lassen. Sie haben schnell Erfolgserlebnisse, was sie wiederum motiviert.

Scratch ermöglicht es ihnen leicht, Animationsfilme, schöne Muster oder eigene Spiele zu programmieren.

Unter http://scratch.mit.edu/ kann man Scratch für die Betriebssysteme Microsoft Windows, Linux (bisher nur eine Experimentierversion) und Mac OS X kostenlos herunterladen.

¹⁰[20, letzter Zugriff: 13.01.2009]

¹¹http://de.wikipedia.org/wiki/Scratchen, letzter Zugriff: 13.01.2009

2.1.1 Oberfläche

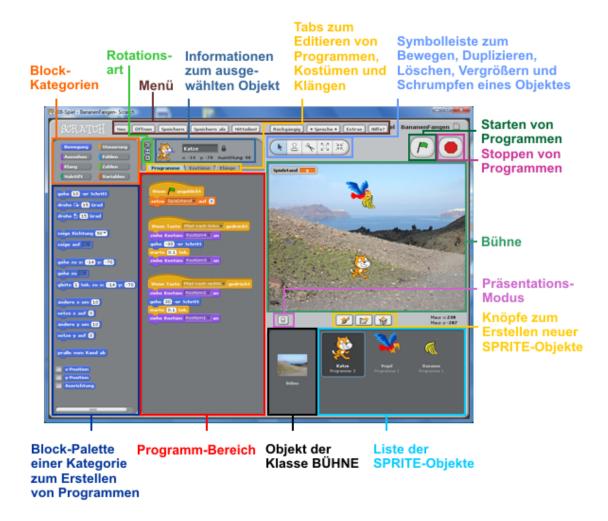


Abbildung 1: Scratch-Oberfläche

Die Programmoberfläche ist sehr übersichtlich aufgebaut, so dass sich vieles von selbst erklärt.

Unterhalb der Bühne findet man die Knöpfe zum Erstellen von Objekten. Man hat die Möglichkeit sich selbst ein Objekt zu malen, eines aus einer Liste auszusuchen oder sich ein zufälliges Überraschungs-Objekt aus den vorhandenen anzeigen zu lassen.

Darunter werden neben dem Objekt der Klasse BÜHNE noch alle weiteren Objekte angezeigt. Für das ausgewählte Objekt können Programme "geschrieben", Kostüme bzw. Hintergründe und Klänge hinzugefügt, bearbeitet oder gelöscht werden. Dafür muss im Programm-Bereich der jeweilige Reiter "Programme", "Kostüme" bzw. "Hintergründe" oder "Klänge" aktiviert sein.

Jedes Objekt kann mehrere Programme besitzen. Scratch ermöglicht es, dass mehrere Prozesse gleichzeitig ablaufen (siehe Nebenläufigkeit). Beispielsweise kann das Objekt Katze im "Hauptprogramm" überprüfen, ob bestimmte Bedingungen zutreffen, und gleichzeitig auf Ereignisse, wie Drücken der rechten oder linken Pfeil-Taste, reagieren.

Pro Kategorie ist eine Block-Palette vorhanden. Die einzelnen Blöcke, die je nach der Kategorie, der sie angehören, eine bestimmte Farbe besitzen, müssen nur in den Programm-Bereich gezogen werden. Sie passen wie LEGO-Steine aneinander.

Für das Objekt Bühne können ebenfalls Programme geschrieben werden, jedoch stehen nicht alle Methoden zur Verfügung - z. B. kann die Bühne nicht bewegt werden.

Die Blöcke kann man ihrem Aussehen nach folgendermaßen unterteilen:



Mit verändernden Methoden wird der Zustand des Objektes verändert. Sondierende Methoden geben Informationen über den Zustand des Objekts zurück. Bedingungen liefern einen Wahrheitswert zurück. Wenn die Bedingung in dem Moment zutrifft, ergibt es wahr ansonsten falsch.

Mit einem Kostümwechsel kann man die Bewegungen eines Objektes darstellen - z. B. kann eine Person tanzen, indem man unterschiedliche Bilder als Kostüme auswählt.

Es können Töne und Klänge abgespielt werden. Zusätzlich besteht auch die Möglichkeit, selbst Audioaufnahmen durchzuführen. Dafür muss jedoch ein Headset am Schülerrechner angeschlossen sein.

Die Schüler sehen auf der Bühne, ob die Programme das gewünschte Handeln der Objekte bewirken.

Wenn das Objekt nicht immer das gewünschte Verhalten zeigt, kann dies auch an der Einstellung der Rotationsart liegen. Es gibt drei Rotationsarten. Beim Abprallen eines Objektes vom Rand bewirkt die oberste Rotationsart, dass es sowohl horizontal als auch vertikal gespiegelt wird, die mittlere, dass das Objekt nur vertikal gespiegelt wird. Bei der untersten Rotationsvariante passiert nichts, d. h. das Objekt prallt ab und geht auf einmal "rückwärts".

Zum Starten der Programme muss der Knopf mit der grünen Fahne und zum Stoppen der Knopf mit dem roten Achteck gedrückt werden. Das Programm kann auch im Präsentationsmodus ablaufen, d. h. man sieht die Bühne auf dem kompletten Bildschirm.

Mit der Symbolleiste oberhalb der Bühne kann man Objekte bewegen, duplizieren, löschen, vergrößern oder verkleinern.

Das Menü ganz oben ermöglicht es, eine leere Programmoberfläche für ein neues Projekt zu erhalten, ein vorhandenes Projekt zu öffnen, das Projekt unter dem gleichen oder einem neuen Namen abzuspeichern, es online zu veröffentlichen, Schritte rückgängig zu machen, die Sprache zu ändern, unter Extras u. a. das Programm in Einzelschritten ablaufen zu lassen und die Hilfe anzusehen.

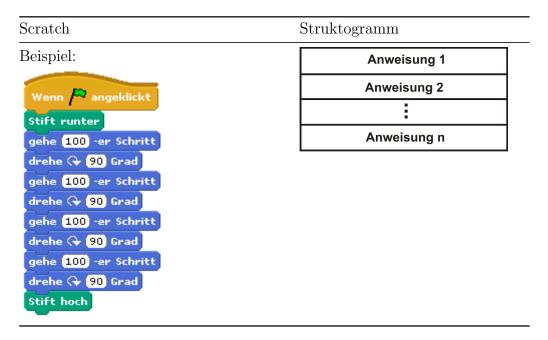
Läuft das Programm in Einzelschritten ab, so sieht man durch Hervorhebungen wie es abgearbeitet wird. Es ist nicht mehr möglich, dieses zu beeinflussen oder den Ablauf zu unterbrechen.

2.1.2 Kontrollstrukturen

Alle Kontrollstrukturen werden mit einem Beispiel in Scratch und dem zugehörigen Struktogramm aufgezeigt. Die hier aufgeführten Struktogramme - die sogenannten Nassi-Shneiderman-Diagramme - orientieren sich an der Norm DIN 66261^{12} .

Sequenz

Die Sequenz stellt eine Folge nacheinander auszuführender Anweisungen dar.

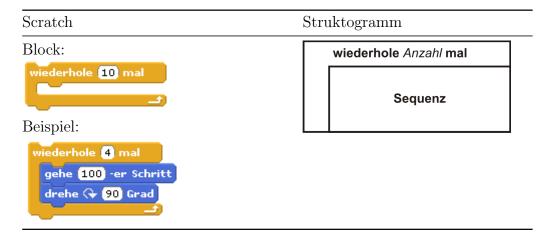


Weitere Kontrollstrukturen sind in der Kategorie "Steuerung" zu finden.

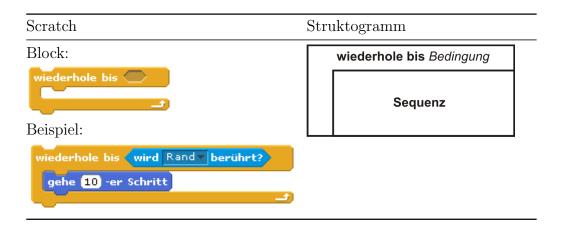
 $^{^{12} \}rm http://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm,$ letzter Zugriff: 26.01.2009

Wiederholung mit fester Anzahl

Soll eine Sequenz mehrmals wiederholt werden und steht dabei die Anzahl der Wiederholungen fest, verwendet man zur Vereinfachung die Wiederholung mit fester Anzahl.



Bedingte Wiederholung



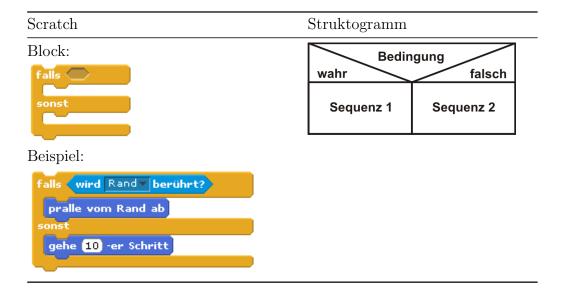
Die Bedingte Wiederholung ist in Scratch im Vergleich zu anderen Programmiersprachen eine Besonderheit. Es handelt sich um eine kopfgesteuerte bedingte Wiederholung mit Abbruchbedingung.

Normalerweise wird in einer kopfgesteuerten bedingten Wiederholung überprüft, ob eine Bedingung zutrifft. Nur wenn sie erfüllt ist, wird die Wiederholung ausgeführt. In Scratch wird überprüft, ob die angegebene

Bedingung zutrifft, und nur wenn diese Bedingung nicht erfüllt ist, wird die Wiederholung ausgeführt.

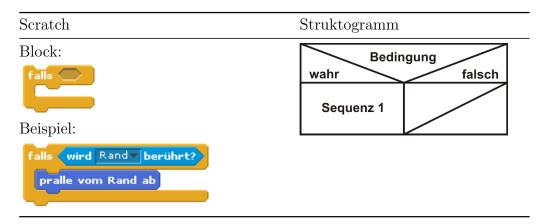
Zweiseitig bedingte Anweisung

Die zweiseitig bedingte Anweisung verwendet man, wenn abhängig von einer Bedingung unterschiedliche Sequenzen ausgeführt werden sollen.



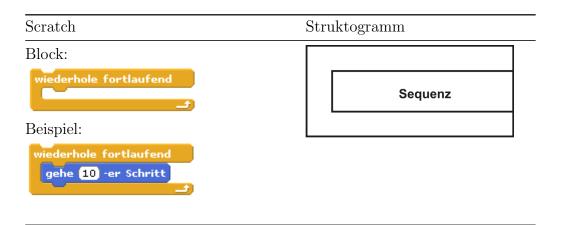
Einseitig bedingte Anweisung

Bei der einseitig bedingten Anweisung wird eine Sequenz nur dann ausgeführt, wenn die Bedingung den Wert wahr liefert.



Für viele Programme benötigt man eine unendliche Wiederholung.

Unendliche Wiederholung



Die unendliche Wiederholung ist nur ein Spezialfall der bedingten Wiederholung. Die verwendete Bedingung heißt **wahr** und ist somit immer erfüllt. Die unendliche Wiederholung zeugt eigentlich von schlechtem Programmierstil, ermöglicht es den Schülern aber auf einfache Weise eine Sequenz ständig wiederholen zu lassen.

2.1.3 Bedingungen

Bedingungen sind Methoden, die als Rückgabewert entweder wahr oder falsch liefern.

In der Kategorie "Fühlen" findet man folgende Bausteine:

Maustaste gedrückt?	Liefert wahr, wenn die Maustaste gedrückt wird.
Taste Leertaste ▼ gedrückt?	Liefert wahr, wenn die Leertaste (, eine Buchstabentaste, eine Zahlentaste oder eine Pfeiltaste) gedrückt wird.
wird Mauszeiger▼ berührt?	Liefert wahr, wenn das Objekt den Mauszeiger berührt.
wird Rand▼ berührt?	Liefert wahr, wenn das Objekt den Rand der Scratch-Bühne berührt.
wird Objekt1▼ berührt?	Liefert wahr, wenn das Objekt das Objekt "Objekt1" berührt.
wird Farbe berührt?	Liefert wahr, wenn das Objekt die angegebene Farbe berührt.
Farbe berührt ?	Liefert wahr, wenn eine bestimmte Farbe eines Objektes eine andere Farbe (eines anderen Objektes oder im Hintergrund) berührt.
laut?	Liefert wahr, wenn es laut ist.

In der Kategorie "Zahlen" sind noch weitere Bedingungen vorhanden.

Mit Hilfe der Boolschen Algebra kann man Bedingungen miteinander verknüpfen bzw. negieren.

und	Liefert wahr, wenn beide Bedingungen wahr sind.
oder	Liefert wahr, wenn eine der beiden oder beide Bedingungen wahr sind.
nicht	Liefert wahr, wenn die Bedingung nicht wahr ist.
Beispiel:	
nicht Taste Leertaste ▼ gedrückt?	Liefert wahr, wenn die Leertaste (, eine Buchstabentaste, eine Zahlentaste oder eine Pfeiltaste) nicht gedrückt wird.

Man kann überprüfen, ob eine Zahl oder eine erzeugte Zufallszahl oder ein Variablenwert kleiner, größer oder gleich einem anderen Wert ist.

	Liefert wahr, wenn der linke Wert kleiner ist als der rechte.
	Liefert wahr, wenn der linke Wert größer ist als der rechte.
	Liefert wahr, wenn der linke und der rechte Wert übereinstimmen.
Beispiel: falls spielstand > 5 sage Spielendel für 2 Sek. stoppe alles	Überprüft, ob die Variable spielstand größer als 5 ist. Wenn diese Bedingung zutrifft, wird "Spielende" ausgegeben und das ganze Programm gestoppt.

Scratch verwendet den "="-Operator als Vergleichsoperator und nicht als Zuweisungsoperator.

2.1.4 Methodenaufrufe

In Scratch werden Methoden durch Skripte beschrieben.

Es gibt vier Möglichkeiten eine Methode aufzurufen:

Wenn 🏳 angeklickt	Das Skript wird ausgeführt, wenn jemand auf den Knopf mit der grünen Fahne klickt. Man könnte es auch als Hauptprogramm "main" - wie in vielen Programmiersprachen - bezeichnen.			
Wenn Taste Leertaste ▼ gedrückt	Das Skript wird ausgeführt, wenn eine Taste - wie z. B. die Leertaste, eine Buchstabentaste, eine Zahlentaste oder eine Pfeiltaste - betätigt wird.			
Wenn Objekt1 angeklickt	Das Skript wird ausgeführt, wenn das Objekt namens "Objekt1" anklickt wird.			
Wenn ich ▼ empfange	Das Skript wird ausgeführt, wenn das Objekt eine Botschaft von einem anderen Objekt erhält.			
Beispiel				
	Das Skript wird ausgeführt, wenn das			
Wenn ich Botschaft▼ empfange	Objekt eine Botschaft namens			
spiele Klang Miauv	"Botschaft" von einem anderen Objekt erhält.			

Die ersten drei Möglichkeiten können nur vom Benutzer gesteuert werden. Bei der letzten Möglichkeit können Objekte durch Botschaften miteinander kommunizieren. Um so eine Methode aufzurufen, gibt es zwei Möglichkeiten.

Mit dem Block wird die Methode aufgerufen, das Programm läuft aber weiter. Das führt dazu, dass das Programm, das die Methode aufgerufen hat, weiter abläuft, während die aufgerufene Methode ausgeführt wird. Wenn man aber möchte, dass das Programm erst weiterläuft, wenn die aufgerufene Methode vollständig ausgeführt wurde,

dann muss folgender Baustein verwendet werden:

sende 💌 an alle und warte

Scratch hat an dieser Stelle eine weitere Besonderheit. Beim Senden einer Botschaft führen alle Objekte, die die angegebene Methode besitzen, diese aus. Möchte man dies verhindern, so muss man unterschiedliche Methodennamen verwenden.

2.1.5 Variablen

In der Kategorie "Variablen" können Variablen definiert werden. Es gibt eine Unterscheidung zwischen globalen und lokalen Variablen. Standardmäßig wird eine globale Variable erzeugt, auf die alle Objekte zugreifen können - also eine "Variable für alle Objekte". Soll nur ein Objekt auf eine Variable zugreifen dürfen, so benötigt man eine lokale Variable. Dafür muss beim Anlegen dieser die Option "Variable nur für dieses Objekt" ausgewählt sein.

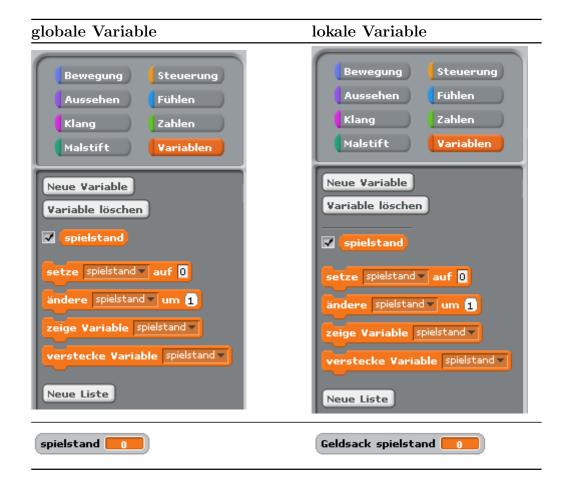


Nach dem Erzeugen einer Variable erscheinen eine sondierende Methode, die den Namen der Variable trägt, und darunter vier verändernde Methoden. Die globalen Variablen werden zuerst aufgelistet, die lokalen nach einem Trennstrich.

Zusätzlich erscheint auf der Bühne ein Baustein mit dem Namen der Variablen und ihrem momentanen Wert. Bei lokalen Variablen steht noch der Name des Objekts davor, dem diese Variable "gehört".

Mit den verändernden Methoden kann man die Variable initialisieren, d. h. einen Startwert festlegen, oder diese de- bzw. inkrementieren, indem man den Wert angibt, um den sie sich verändern soll. Und man kann den dazugehörigen Baustein auf der Bühne anzeigen oder verstecken lassen.

Durch die Verwendung der verändernden Methoden "setze ... auf 0" und "ändere ... um 1" wird der Zuweisungsoperator nicht explizit sichtbar.



2.1.6 Listen

Seit der Version 1.3.1 gibt es auch Listen in Scratch. Diese findet man in der Kategorie "Variablen". Bei Listen gibt es auch die Unterscheidung zwischen globalen und lokalen Listen. Auch hier wird eine Liste standardmäßig als global definiert.

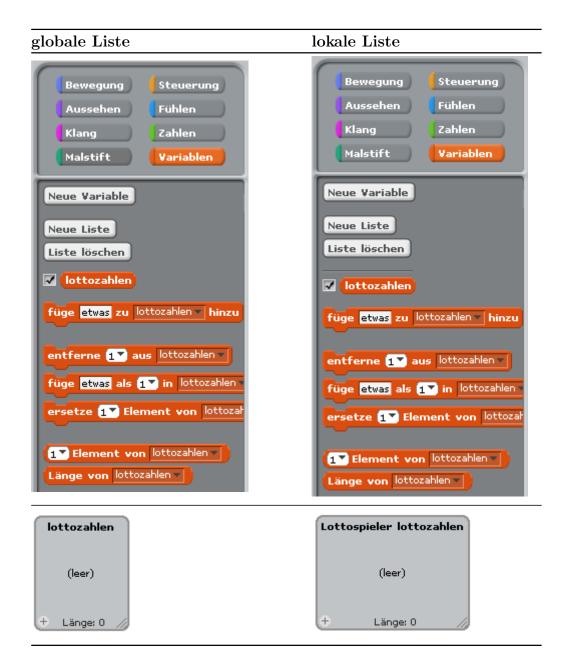


Nach dem Erzeugen einer Liste erscheinen eine sondierende Methode, die den Namen der Liste trägt, und darunter vier verändernde und zwei weitere sondierende Methoden. Die globalen Listen werden zuerst aufgelistet, die lokalen nach einem Trennstrich.

Zusätzlich wird auf der Bühne ein Baustein mit dem Namen der Liste und ihrem momentanen Inhalt angezeigt. Bei lokalen Listen steht noch der Name des Objekts davor, dem diese Liste "gehört".

Mit den verändernden Methoden kann man der Liste ein Element hinzufügen - auch Strings -, das erste oder letzte Element oder alle Elemente entfernen, Elemente an erster, letzter oder beliebiger Stelle einfügen, das erste, letzte oder ein beliebiges Element mit dem angegebenen ersetzen.

Mit den sondierenden Methoden kann man das erste, letzte oder ein beliebiges Element der Liste und die Länge der Liste zurückgeben lassen.



An dieser Stelle fehlt noch eine weitere sondierende Methode, die überprüft, ob ein bestimmtes Element in der Liste vorhanden ist. Somit werden einige Programme unnötig kompliziert.

Und auch dann stößt man an Grenzen, weil man kein bestimmtes Element aus der Liste entfernen kann, d. h. es fehlt noch eine weitere verändernde Methode, die ein bestimmtes Element aus der Liste entfernt.

Es ist bisher nicht möglich, diese Funktionen auf einfache Weise selbst zu

implementieren, da Scratch noch keine Möglichkeit bietet, Parameter an eigene "Methoden" zu übergeben. In den Scratch-Foren¹³ werden solche Erweiterungen jedoch diskutiert.

2.1.7 Nebenläufigkeit

Scratch ermöglicht es Objekten, gleichzeitig mehrere Skripte auszuführen.

Grundsätzlich müsste man die Skripte synchronisieren, um eine Verklemmung auszuschließen. Das erledigt jedoch Scratch im Hintergrund. Der Nutzer muss sich nicht darum kümmern.

2.1.8 Ereignisverwaltung (Event-Handling)

Scratch hat einen EventListener, der auf Tastatureingaben und Mausklicks reagiert.

Wenn man ein Skript programmiert hat, das etwas tun soll, wenn z. B. die "Pfeil nach oben"-Taste gedrückt wird, dann ruft Scratch das Skript selbstständig beim Eintreten des Ereignisses auf.

Das "ActionCommand" muss man nicht erst auswerten. Das erledigt Scratch im Hintergrund selbst.

 $^{^{13} \}texttt{http://scratch.mit.edu/forums/viewtopic.php?id=10887}, letzter Zugriff: <math display="inline">24.02.2009$

3 Didaktische und methodische Überlegungen

3.1 Lernvoraussetzungen

Die Schüler sollten die Begriffe Objekt, Klasse, Attribut, Attributwert und Methode kennen und diese voneinander abgrenzen können. Zudem sollte ihnen geläufig sein, das Objekte miteinander in Beziehung stehen können.

3.2 Lernziele

Die Grobziele, die mit der vorliegenden Unterrichtssequenz verfolgt werden, sind auch in den Bildungsstandards¹⁴ verankert:

Schülerinnen und Schüler

- kennen Algorithmen zum Lösen von Aufgaben und Problemen aus verschiedenen Anwendungsgebieten und lesen und interpretieren gegebene Algorithmen,
- entwerfen und realisieren Programme mit den algorithmischen Grundbausteinen und stellen diese geeignet dar.

3.3 Didaktische Überlegungen zu den Lerninhalten

3.3.1 Kontrollstrukturen, Bedingungen und Methodenaufrufe

Die Schülerinnen und Schüler lernen die wichtigsten Kontrollstrukturen kennen:

- Sequenz
- Wiederholung mit fester Anzahl
- (Kopfgesteuerte) Bedingte Wiederholung
- Zweiseitig bedingte Anweisung
- Einseitig bedingte Anweisung

¹⁴[14, Seite 30-32]

Zudem wird die unendliche Wiederholung als Spezialfall der bedingten Wiederholung verwendet.

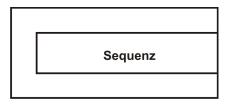
Für die bedingte Wiederholung und die nachfolgenden Kontrollstrukturen sind die Begriffe "Bedingung" und "Rückgabewert" erfoderlich.

Aufrufe von selbst geschriebenen Methoden werden ausschließlich nur für die Kommunikation zwischen Objekten verwendet.

Der Vergleichsoperator "=" behält seine Bedeutung wie in der Mathematik bei.

3.3.2 Struktogramme

Wie bereits unter 2. erwähnt orientieren sich die Struktogrammen grundsätzlich an der Norm DIN 66261¹⁵. Jedoch wurde bei der unendlichen Wiederholung davon abgewichen. Das normgerechte Struktogramm für die unendliche Wiederholung weicht von dem der Wiederholung mit fester Anzahl und der kopfgesteuerten bedingten Wiederholung ab:



Da die unendliche Wiederholung nur ein Spezialfall der bedingten Wiederholung ist, wurde kein großer Wert auf dieses spezielle Diagramm gelegt.

Zudem macht die Unterscheidung nur dann Sinn, wenn die Schüler neben der kopfgesteuerten bedingten Wiederholung - Bedingung wird vor dem Ausführen der Sequenz überprüft - auch die fußgesteuerte bedingte Wiederholung - Bedingung wird erst nach dem Ausführen der Sequenz überprüft - und ihre Struktogramme kennen. Dann ist auch das Struktogramm für die unendliche Wiederholung einleuchtend und benötigt keiner zusätzlichen Erklärung.

 $^{^{15} \}rm http://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm,$ letzter Zugriff: 26.01.2009

Damit wurde auch verhindert, dass die Schüler sich zu viele Diagramm-Typen merken müssen und diese auch noch versehentlich verwechseln.

Zudem wird sowohl bei Robot Karol als auch in allen Schulbüchern an dieser Stelle von der Norm abgewichen.

Daher wurde für die unendliche Wiederholung folgendes Struktogramm verwendet:



3.3.3 Variablen

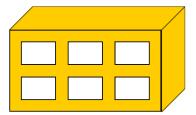
Der Themenbereich Variablen stellt eigentlich keinen Lerninhalt der 7. Jahrgangsstufe dar. Diese Entscheidung ist für quelltextbasierte Programmierumgebungen sinnvoll. Da hierbei insbesondere die Verwendung des Zuweisungsoperators "=" und seine zur Mathematik verschiedene Bedeutung für Probleme sorgen kann. Jedoch besteht dieses Problem bei Scratch nicht (siehe unter 2.1.5). Scratch ermöglicht den Schülern einen spielerischen Zugang, bei dem es für sie keine Verständnisschwierigkeiten gibt. Für die Schüler ist es selbstverständlich, dass der Spielstand eines Spiels in der Variable "spielstand" abspeichert werden muss. Zu Beginn des Spiels muss der Spielstand auf 0 gesetzt werden - die Variable "spielstand" muss also initialisiert werden. Im Laufe des Spiels muss dieser erhöht bzw. erniedrigt werden - die Variable "spielstand" muss somit inkrementiert bzw. dekrementiert werden.

Auch der Unterschied zwischen einer globalen und einer lokalen Variable ist den Schülern klar. Die lokale sieht nur das Objekt, dem es gehört, die globale sehen alle Objekte.

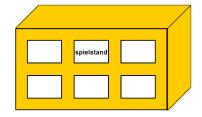
Wenn die Schüler die Standardeinstellung auf global belassen, können sie nichts falsch machen.

Das Geheimnisprinzip wird nicht besprochen.

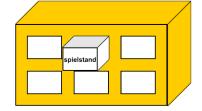
Die Verwendung von Variablen und eine Idee des Speicheraufbaus lässt sich den Schülern durch eine Analogie zu einem Schrank mit vielen Schubladen erklären.



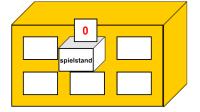
Erzeugt man eine Variable, so wird im Speicher entsprechend Platz reserviert. Der Speicherplatz wird durch eine Schublade repräsentiert, auf der der Variablenname steht.



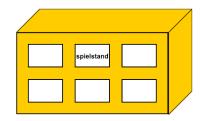
Am Anfang muss die Variable initialisiert werden, d. h. man öffnet die Schublade,



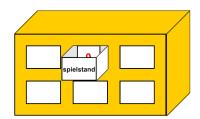
steckt einen Zettel mit dem Anfangswert der Variable in die Schublade



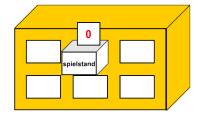
und schließt diese danach.



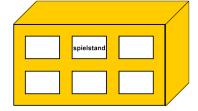
Will man auf die Variable zugreifen und ihren Wert erfahren, öffnet man die Schublade. In dieser findet man bereits einen Wert - beispielsweise den Wert 0.



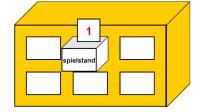
Es wird der Wert der Variablen nachgesehen.



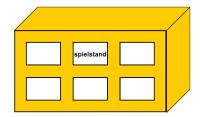
und die Schublade wieder geschlossen.



Soll die Variable einen neuen Wert erhalten, dann öffnet man die Schublade, wirft den Zettel, der drin liegt, weg und legt einen neuen Zettel mit dem aktuellen Wert hinein.



und schließt die Schublade danach.



Das Abspeichern von Informationen ist ein grundlegendes Konzept in der Informatik. Es ist sinnvoll, dass auch die Schüler, die ab der 8. Jahrgangsstufe nicht das naturwissenschaftlich-technologische Gymnasium besuchen, zumindest eine grobe Vorstellung des Variablenkonzepts entwickeln.

Daher wird in der vorliegenden Unterrichtssequenz auch auf das Thema Variablen eingegangen.

3.3.4 Listen

Selbst die Listen dürften den Schülern keine großen Verständnisschwierigkeiten bereiten, wenn man ihnen nur einfache Beispiele zeigen würde.

Da jedoch - wie bereits unter 2.1.6 erwähnt - ein paar entscheidende Methoden fehlen, werden dadurch Programme schnell zu komplex. Zudem besteht keine Notwendigkeit dieses Thema der 11. Jahrgangsstufe (Inf 11.1.1) zu behandeln.

3.3.5 Nebenläufigkeiten

Das Thema der Nebenläufigkeit wird zwar erst in der 12. Jahrgangsstufe (Inf 12.2) angesprochen, jedoch stellt dies für die Schüler überhaupt keine Schwierigkeit dar.

Die Schüler denken an dieser Stelle gar nicht daran, dass es eigentlich zu Verklemmungen kommen könnte. Für sie ist es aufgrund ihrer Alltagserfahrung selbstverständlich, dass mehrere Objekte gleichzeitig etwas tun. Genauso stellt es für sie auch kein Hindernis dar, dass ein Objekt gleichzeitig mehrere Skripte ausführt.

Die Möglichkeit einer Verklemmung spielt hier keine Rolle. Scratch ist im Hintergrund dafür zuständig, dass so etwas nicht passiert.

3.3.6 Ereignisverwaltung (Event-Handling)

Für die Schüler stellt auch die Verarbeitung von Events kein Problem dar. Scratch bietet bereits Konstrukte an, auf Maus- und Tastatureingaben zu reagieren. Die Schüler verwenden es intuitiv, auch wenn man dies noch gar nicht erklärt hat.

Den Schülern ist bewusst, dass ein entsprechender Programmblock nur ausgeführt wird, wenn ein Event - wie z. B. das Drücken der "Pfeil nach rechts"-Taste - eintritt.

3.4 Die Programmierumgebung

3.4.1 Anforderungen an eine Programmierumgebung

Eine schülergerechte Programmierumgebung sollte \dots^{16}

- einfach und klar aufgebaut und leicht verständlich sein.
- sich auf das Minimalste begrenzen und keine unnötigen Einstell- und Wahlmöglichkeiten besitzen.
- so aufgebaut sein, dass man mit bereits wenigen Anweisungen ein sinnvolles Programm erhält und dass nicht zuerst eine Menge an Syntax gelernt werden muss.
- Eingabehilfen enthalten, damit die Schüler nicht aufgrund von Tippfehlern am Programmieren scheitern.
- eine sofort nachvollziehbare grafische Umsetzung des Programms enthalten.
- für die Unterstufe ansprechend sein und einen spielerischen Zugang zur Algorithmik ermöglichen.
- die objektorientierte Sichtweise beibehalten.

3.4.2 Beurteilung von Scratch

Die Programmoberfläche ist sehr übersichtlich und dem Alter entsprechend aufgebaut, so dass sich vieles von selbst erklärt.

Man hat sich nur auf das Wesentliche begrenzt und alles Überflüssige weggelassen - wie z. B. Einstellmöglichkeiten für die Programmieroberfläche oder die Definition von eigenen Methoden mit Parameterübergabe. Das hat natürlich auch den Nachteil, dass Scratch nur für den Einstieg, aber nicht als weiterführendes Werkzeug für das Thema Algorithmen und Programmierung in höheren Jahrgangsstufen geeignet ist. Alles, was man für die 7. Jahrgangsstufe benötigt, ist jedoch vorhanden.

Es besteht die Möglichkeit, Objekte selbst zu malen oder auch Geräusche, Töne und Klänge selbst aufzunehmen - vorausgesetzt ein Headset ist am Rechner angeschlossen. Der spielerische Faktor ist auf jeden Fall vorhanden. Da die Schüler in diesem Alter häufig am Computer spielen, kann man

¹⁶[13, Seite 105-106]

sie damit motivieren, dass sie am Ende der Unterrichtssequenz das nötige Wissen haben, um eigene Spiele zu programmieren.

Aufgrund der vorgegebenen Bausteine müssen die Schüler kaum Syntax erlernen und nichts eintippen. Bei Robot Karol neigen die Schüler - trotz des vorhandenen Menüs - dazu, Methodenaufrufe und Kontrollstrukturen einzutippen. Dadurch entstehen Syntaxfehler, die dazu führen, dass sich die Schüler oft auf Fehlersuche begeben und somit vom Wesentlichen abgelenkt werden.

Bei Robot Karol gibt es die Möglichkeit, sich Struktogramme anzeigen zu lassen. Das ist bei Scratch nicht möglich. Dafür ähneln die Programme sehr den Struktogrammen, so dass diese Funktion in Scratch gar nicht nötig ist. Der Programmier-Vorgang in Scratch entspricht nahezu der Modellierung.

Die Schüler sehen sofort auf der Bühne, ob das Objekt das Gewünschte macht und müssen gegebenenfalls ihr Modell noch einmal überdenken.

Unter Extras können die Schüler die Option "Ausführung in Einzelschritten" auswählen. Wenn man das Programm startet, wird dieses etwas langsamer als sonst ausgeführt und man sieht durch die farbliche Hinterlegung den Programmablauf.

Jedoch ist das ein kleiner Nachteil gegenüber Robot Karol, bei dem man die Geschwindigkeit bestimmen kann und das Programm dabei auch jeder Zeit anhalten kann.

Im Gegensatz zu Robot Karol ist es in Scratch nicht so einsichtig, Teile des

Programms in einer Methode auszulagern. Das Konstrukt wenn ich empfange für die Kommunikation zwischen Objekten erschließt sich den Schülern. Die Programme in der 7. Jahrgangsstufe sind relativ kurz, so dass es gar nicht notwendig erscheint, Programmcode in Methoden auszulagern.

Das Entscheidende ist, dass Scratch einem ermöglicht, mehrere Objekte gleichzeitig auf dem Objekt Bühne agieren zu lassen. Im Gegensatz zu Robot Karol wird mit Scratch die Objektorientierung konsequent weitergeführt. Für diejenigen Schüler, die ab der 8. Jahrgangsstufe das naturwissenschaftlich-technologische Gymnasium besuchen werden, wird eine Grundlage für die 10. Klasse gelegt. Es besteht die Chance, dass den Schülern der Einstieg in die objektorientierte Programmierung mit Java dann leichter fallen wird.

Die Schüler haben dann auch bereits eine Vorstellung von Variablen entwickelt. Dieses Wissen kann in der 10. Klasse vertieft werden.

Selbst zur Demonstration und Veranschaulichung von Listen in der 11. Klasse und von Nebenläufigkeiten in der 12. Klasse kann Scratch dienen.

Scratch ist genau das richtige Werkzeug für den Einsatz in der Algorithmik in der 7. Jahrgangsstufe und vermittelt grundlegende Ideen zu weiteren ganz wichtigen Konzepten der Informatik. Zudem eröffnet es im Gegensatz zur eintönigen Welt vom Robot Karol die Möglichkeit zu immer wieder neuen Aufgabenstellungen mit neuen Akteuren. Damit lässt sich Langeweile aufgrund ähnlicher Aufgabenstellungen vermeiden.

3.5 Methodische Vorüberlegungen

Sowohl in der Motivation als auch in den Arbeitsaufträgen wurden immer wieder neue Problemstellungen gewählt.

Bei den Aufgabenstellungen ist der spielerische Faktor und der "Erfindergeist" mit berücksichtigt. Die Schüler können ihrer Kreativität freien Lauf lassen. Daher gibt es selten eine bestimmte Musterlösung, sondern eine mögliche Lösung.

Dabei wird viel Wert darauf gelegt, dass die Schüler für das reale Problem (Problemsituation) sich selbst eine Lösung (mentales Modell), beispielsweise als Struktogramm, überlegen und dieses in Scratch umsetzen. Die erhaltene informatische Lösung muss nun interpretiert werden und führt so zum Ergebnis. Im Rahmen der Validierung wird überprüft, in wie weit das Ergebnis wirklich die Lösung des Ausgangsproblems ist. Gibt es hier Differenzen, muss die Modellierung überdacht werden. Der informatische Problemlöseprozess läuft so lange ab, bis das erhaltene Ergebnis das eigentliche Problem löst¹⁷.

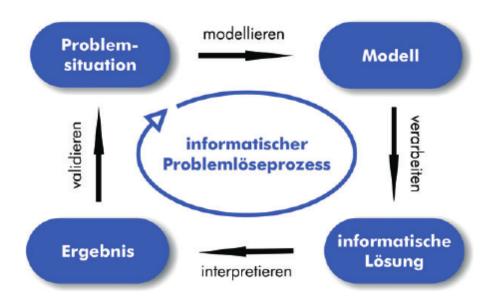


Abbildung 2: Problemlöseprozess

"Mittlerweise ist das Problemlösen mit Informatiksystemen eine der allgemein anerkannten Leitlinien der informatischen Bildung [...].

¹⁷[15, letzter Zugriff: 21.02.2009]

Die Struktur des Problemlöseprozesses stellt deshalb gerade für den Informatikunterricht ganz im Sinne von Roth [...] ein geeignetes Mittel zur Artikulation des Informatikunterrichts dar. Jeder neuer Stoff sollte anhand von Problemen aus dem Erfahrungsbereich der Schüler [...] eingeführt werden".¹⁸

Laut Hubwieser sollten Probleme ausgewählt werden, "deren Komplexität einerseits so hoch sein sollten, dass sie von den Schülern ohne die zu erlernenden Konzepte nicht oder nur unter erheblich höherem Aufwand gelöst werden können. Andererseits darf der intellektuelle Horizont der Schüler nicht überschritten werden. Im Sinne von Edelmann [...] wären Probleme, die sich durch Anwendung von Strategien oder durch Systemdenken lösen lassen, optimal. Das Ergebnis des Problemlöseprozesses ist dabei im Vergleich zu konventionellen Unterrichtsfächern aus dem mathematisch-naturwissenschaftlichen Bereich meist wesentlich offener." ¹⁹

3.6 Hausaufgaben

Da in der 7. Klasse Informatik nur einstündig ist und Verbesserungen von Hausaufgaben oft sehr viel Zeit kosten, werden schriftliche Hausaufgaben nur selten gestellt. Die Schüler haben während der Unterrichtsstunde die Möglichkeit das Gelernte gezielt einzuüben. Die Vorbereitung auf die nächste Stunde besteht für die Schüler darin, sich die Hefteinträge der letzten Stunden, in denen die theoretischen Hintergründe mit Beispielen untermalt sind, am Tag vorher anzusehen.

Es gibt eine Hausaufgabe zur Beschreibung von Abläufen und eine kurz vor der Stegreifaufgabe. Letztere wird korrigiert und muss ein Tag vor der nächsten Stunde bei der Lehrkraft abgegeben werden.

¹⁸[4, Seite 69]

¹⁹[4, Seite 69]

3.7 Technische Vorüberlegungen

Die hier vorliegende Unterrichtssequenz wurde im 2. Halbjahr des Schuljahres 2007/08 am Dientzenhofer-Gymnasium Bamberg erstmals erprobt und im 1. Halbjahr des Schuljahres 2008/09 am Kaiser-Heinrich-Gymnasium Bamberg mit ein paar Änderungen nochmals durchgeführt. Die technischen Voraussetzungen waren an beiden Schulen unterschiedlich.

3.7.1 Dientzenhofer-Gymnasium Bamberg





Im sogenannten Multimediaraum I214 sind 31 Rechner - einige mit dem Betriebssystem Windows 2000, einige mit Windows XP - vorhanden, die alle mit Hardware Sheriff-Karten ausgestattet sind. Die Monitore sind in den Tischen versenkt. Somit hat man die komplette Klasse im Blick. Der Raum ist jedoch sehr breit. Das bringt viele Nachteile mit sich. Der Beamer projiziert an die rechte Wand (vom Lehrerpult aus gesehen). Somit sehen die Schüler, die direkt davor sitzen gar nichts. Für die Schüler ganz hinten muss man alles vergrößern. Die Projektionsfläche für den Overheadprojektor ist genau hinter dem Lehrerpult in der Mitte. Hier sehen die Schüler an den Seiten gar nichts.

Das Whiteboard steht an der linken Wand (vom Lehrerpult aus gesehen). Es lässt sich frei verschieben. Allerdings glänzt es so sehr, dass die Schüler aufgrund von Spiegelungen nur wenig darauf erkennen konnten.

Wegen der Gegebenheiten musste ich in diesem Raum meinen Unterricht etwas umstellen. Ich habe komplett auf den Overheadprojektor und das Whiteboard verzichtet und dafür den Schülern häufig meinen Bildschirm für "Tafelabschriften" oder um Ihnen etwas zu zeigen mit dem didaktischen

Netz von Tandberg überblendet.

An einigen Rechnern war ein funktionsfähiges Headset vorhanden, so dass die Schüler auch Audioaufnahmen durchführen konnten.

Die Schüler melden sich alle mit dem gleichen Zugang an. Auf einem Netzlaufwerk befinden sich in einem Ordner alle Lehrer-Ordner. Jeder Lehrer hat in seinem Ordner seine Klassen selbst angelegt und für jeden Schüler in der jeweiligen Klasse einen Ordner vorbereitet. Dort legen die Schüler ihre Dateien ab. Es besteht immer die Gefahr, dass Schüler diese löschen.

Da dieser Computerraum mit der Tandberg-Steuerung ausgestattet ist und diesen eine Firma eingerichtet hat, wollte sich der Systembetreuer mit der äußerst komplizierten Tandberg-Steuerung nicht auseinandersetzen. Daher ist die Installation der Rechner in dem Raum sehr aufwändig, da er die Rechner schlecht klonen kann, ohne mit der Tandberg-Steuerung in Berührung zu kommen. Somit ist auf den Rechnern nur Microsoft Office 2000 installiert gewesen. Weitere benötigte Software musste man selbst in die Netzwerkumgebung legen, so dass die Schüler auch darauf zugreifen können.

Scratch lief in der Netzwerkumgebung problemlos, nur gab es Schwierigkeiten beim Abspeichern, weil den Netzlaufwerken kein Laufwerksbuchstabe zugeordnet war.

Da Scratch aber auf Squeak basiert und Squeak damit keine Schwierigkeiten hat, bot sich ein Trick an. Es wurde Squeak installiert und der Inhalt des kompletten Scratch-Verzeichnisses in das Squeak-Verzeichnis kopiert. Wenn man die Datei "Squeak.exe" startet, fragt Squeak nach einer Image-Datei. An der Stelle gibt man nicht die eigentliche Image-Datei von Squeak, sondern die von Scratch an. Man startet somit immer die Datei Squeak.exe. Das Image muss nur einmal angegeben werden. Squeak merkt sich den Speicherort der Image-Datei. Beim Starten von "Squeak.exe" erscheint zwar das Squeak-Logo, aber es startet in Wirklichkeit Scratch. Damit die Schüler dies nicht verwirrt, wurde eine Verknüpfung zur Datei Squeak.exe namens Scratch außerhalb des Squeak-Verzeichnisses angelegt.

Es ist für die Schüler ein großer Vorteil, dass jeder einen eigenen Rechner zur Verfügung hat. So sind alle Schüler gezwungen, sich mit dem Stoff auseinanderzusetzen und nicht auf den Partner zu vertrauen. Es kostet den Lehrer jedoch mehr Zeit in Eigenarbeitsphasen von Schüler zu Schüler zu

gehen und auch bei Problemen zu helfen. Das klappte aber sehr gut. Die Schüler haben sich auch gegenseitig unterstützt.

3.7.2 Kaiser-Heinrich-Gymnasium Bamberg





Im Computerraum A21 sind 15 Computer mit dem Betriebssystem Windows XP vorhanden gewesen. Der Raum ist mit einem Beamer, einem Overheadprojektor und einem Interwrite Board ausgestattet. Beim Interwrite Board hat man nur einen Stift, man kann jedoch die Stiftfarbe ändern. Es ist jedoch nicht möglich, mit dem Finger auf dem Board zu schreiben. Es sind zwei Stifte vorhanden. So kann immer ein Stift in der Ladeschale geladen werden. Die Software zum Interwrite Board kann man sich online herunterladen, so dass man genauso wie mit der SMART Board-Software Tafelanschriften vorbereiten kann.

Das KHG hat vor Weihnachten 27 neue Rechner bekommen. Einige waren zwar für die Ganztagesklassen gedacht, jedoch wurden alle neuen Rechner und zwei ältere Rechner (ca. 2 Jahre alt) im Computerraum installiert und die alten in die Klassenzimmer der Ganztagesklassen gestellt. Somit hatte nach Weihnachten jeder Schüler einen eigenen Rechner.

Der Raum selbst ist nicht klein, jedoch wird lediglich etwas mehr als die Hälfte der Fläche für die Tische und die Rechner verwendet. Somit sitzen die Schüler in größeren Klassen sehr eng aneinander. Daher ist es für die Lehrkraft schwierig in Eigenarbeitsphasen von Schüler zu Schüler zu gehen. Man muss die Schüler dazu erziehen, ihre Schultaschen unter den Tisch zu stellen, so dass man durch die Gänge gehen kann. Da die Schüler es bisher immer gewohnt waren, zu zweit am Rechner zu sitzen, haben sie sich nach Weihnachten weiterhin gegenseitig unterstützt.

3 DIDAKTISCHE UND METHODISCHE ÜBERLEGUNGEN





Sowohl Schüler als auch Lehrer können auf einem Samba-Laufwerk (mit Laufwerksbuchstaben) abspeichern. Die Schüler können auf die Dateien der Lehrer nicht zugreifen, können aber die Dateien ihrer Mitschüler löschen (nur innerhalb einer Klasse). Lehrer können Dateien eines Kollegen versehentlich löschen.

Die Schüler müssen sich an den Rechnern nicht anmelden.

Auf jedem Rechner ist die Software "PC Sheriff" installiert, so dass nach jedem Neustart der alte Zustand wieder hergestellt wird.

Auf den Rechnern ist bereits die neueste Scratch-Version installiert gewesen.

An einigen Rechnern waren bis Weihnachten noch Headsets angeschlossen gewesen, so dass einige Schüler etwas aufnehmen konnten.

4 Vorgehen im Unterricht

4.1 Allgemeine Anmerkungen

Folgender Stoffverteilungsplan ist in allen Klassen so umgesetzt worden. Die Stegreifaufgabe und die Verbesserung dieser sind die einzigen nicht festen Stunden.

- 1. Beschreibung von Abläufen 1
- 2. Beschreibung von Abläufen 2
- 3. Kennenlernen von Scratch
- 4. Kennenlernen von Scratch und Sequenz
- 5. Wiederholung mit fester Anzahl
- 6. Verschachtelte Wiederholungen mit fester Anzahl
- 7. Bedingte Wiederholung
- 8. Zweiseitig bedingte Anweisung
- 9. Stegreifaufgabe
- 10. Verbesserung der Stegreifaufgabe
- 11. Einseitig bedingte Anweisung
- 12. Methoden und Botschaften
- 13. Variablen (Exkurs)
- 14. Spielprogrammierung

Für jedes aufgeführte Thema ist je eine Unterrichtsstunde angedacht.

In der Stunde, in der die Stegreifaufgabe geschrieben wird, erhalten die Schüler die korrigierte Hausaufgabe zurück. Es wird eine mögliche Musterlösung besprochen und auf die Besonderheiten und Probleme der Schülerlösungen eingegangen. Danach wird in einem freien Klassenzimmer die Stegreifaufgabe geschrieben. Von der Stunde bleiben dann nur noch wenige Minuten, so dass es sich nicht lohnt, mit einem neuen Thema zu

4 VORGEHEN IM UNTERRICHT

beginnen. Die Schüler sollen ihren Animationsfilm erweitern und dabei die bisher gelernten Kontrollstrukturen verwenden.

Die Folgestunde wird für eine ausführliche Verbesserung der Stegreifaufgabe genutzt. Dabei werden auch unterschiedliche Lösungsmöglichkeiten angesprochen und es wird auf die gemachten Fehler hingewiesen. Die Stunde dient für diejenigen Schüler, die in der Stegreifaufgabe nicht gut abgeschnitten haben, als Wiederholung.

Auf diese zwei Stunden wird in der folgenden Planung nicht näher eingegangen.

In jeder Stunde wird alles bisher Gelernte entweder zu Beginn der Stunde oder in der Erarbeitungsphase wiederholt. In allen Erarbeitungsphasen helfen die Schüler mit Beiträgen zur Lösung des Problems mit.

In den Phasen, in denen die Schüler einen Arbeitsauftrag erledigen, arbeiten sie selbstständig. Der Lehrer dient nur als Ansprechpartner bei Fragen und Problemen.

An einigen Stellen sind Screenshots von besonders gelungenen Schülerlösungen zu finden. Diese sind auch auf der beiliegenden CD-ROM vorhanden. Leider haben einige Schüler oft Scratch einfach geschlossen und ihre gelungenen Lösungen vergessen abzuspeichern.

Alle Arbeitsblätter und ihre Lösungen sind sowohl im Anhang als auch auf der beiliegenden CD-ROM zu finden.

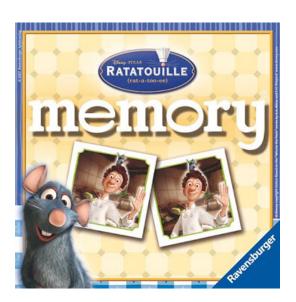
Im folgenden ist immer nur vom Schüler die Rede, jedoch sind natürlich Schülerinnen und Schüler damit gemeint.

4.2 Beschreibung von Abläufen - 1

4.2.1 Lernziele

- Die Schüler sollen erkennen, dass ein Ablauf in mehrere Schritte zerlegt werden könne.
- Sie sollen eine Handlungsvorschrift in Alltagssprache formulieren können.
- Sie sollen wissen, dass Abläufe durch mehrere Anweisungen beschrieben werden können.
- Sie sollen mehrere Beispiele für Verarbeitungsvorschriften kennen.

4.2.2 Motivation



So gut wie alle Schüler kennen das Spiel "memory". Damit diejenigen, die es nicht kennen, wissen, wie es abläuft, soll ein Schüler das Spiel erklären.

4.2.3 Problemfrage

Für den Menschen ist diese Erklärung verständlich. Könnte ein Roboter mit unserer Erklärung etwas anfangen? Was müssen wir an unserer Erklärung verändern?

4.2.4 Erarbeitung

Es wird gemeinsam eine Spielanleitung für das "memory"-Spiel verfasst. Dabei müssen die Schüler immer wieder daran erinnert werden, ob das ein Roboter bereits verstehen kann oder ob dieser nicht noch mehr Anweisungen benötigt.

Als nächstes wird das "memory"-Spiel gespielt. Um den vorhandenen Algorithmus zu testen, beschränkt man der Einfachheit halber das Spiel auf zwei Mitspieler. Das Spiel wird auf einem Tisch gespielt, der frei im Raum steht. Die Schüler begeben sich alle nach vorne und geben den zwei Spielern die notwendigen Anweisungen. Dabei stellen die Schüler schnell fest, dass die verfasste Spielanleitung noch nicht vollständig ist und ergänzen diese.

Die Schüler der Klasse 7a am KHG haben folgende Spielanleitung verfasst:

Spielvorbereitung:

- Öffne die Schachtel.
- Nimm die Karten aus der Schachtel.
- Drehe alle Karten so um, dass das Bild nach unten zeigt.
- Mische alle Karten.
- Lege alle Karten sorgfältig als Rechteck hin.
- Suche dir 1-7 Mitspieler.
- Der jüngste Spieler fängt an.

Spielablauf:

- 1. Decke zwei beliebige Karten auf.
- 2. Wenn die Karten gleich sind, greife sie dir, lege sie auf einen Stapel vor dir und gehe zu Schritt 1.
- 3. Wenn sie nicht gleich sind, merke dir ihre Position und drehe sie wieder um bzw. lege sie wieder an die gleiche Stelle.
- 4. Der nächste Spieler ist dran.

Wiederhole die Schritte 1-4 solange, bis keine Karten mehr da sind.

Der Spieler mit den meisten Karten gewinnt.

Während der Erarbeitungsphase, wird der Begriff der Anweisung bereits eingeführt. Die Schüler haben danach einen Spielablauf mit Anweisungen beschrieben. An dieser Stelle kann man den Begriff Algorithmus einführen und die Schüler damit motivieren, dass sie in den nächsten Stunden mit der Programmierumgebung Scratch Algorithmen programmieren werden.

4.2.5 Sicherung

Die Schüler erhalten das Arbeitsblatt "Beschreibung von Abläufen" und füllen die erste Lücke aus. Sie sollen weitere Handlungsanweisungen nennen, die alle festgehalten werden. In die nächste Lücke gehört der Begriff "Algorithmus".

4.2.6 Hausaufgabe

Die Schüler sollen einen Algorithmus verfassen, wie man sich mithilfe eines Wasserkochers einen Tee macht.

An dieser Stelle könnte man eine Hausaufgabenfolie verteilen. Da dies aber äußerst selten vorkommt, sollte an dieser Stelle kein Schüler benachteiligt werden. Daher wurde darauf verzichtet.

4.2.7 Begründung und Evaluation

Schüler in diesem Alter spielen gerne mit dem Computer. Was liegt näher, als selbst mit Scratch am Ende Spiele zu programmieren. Aus diesem Grund ist auch in dieser Stunde ein Spiel als Motivation gewählt.

4 VORGEHEN IM UNTERRICHT

In der ersten Erprobung wurde das Spiel "Mensch ärgere dich nicht" verwendet. Es hat sich aber als nicht geeignet erwiesen, da es zum einen Zufallszahlen enthält, die es unnötig kompliziert machen. Zudem gibt es zu viele Fälle, die man beachten muss.

"memory" dagegen ist einfach und überschaubar und hat sich in der Erprobung bewährt.

Da das Verfassen der Spielanleitung auch mit "memory" einige Zeit benötigt, ist das komplette Arbeitsblatt in einer Stunde nicht zu schaffen.

4.3 Beschreibung von Abläufen - 2

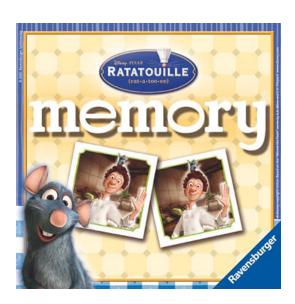
4.3.1 Lernziele

- Die Schüler sollen eine Handlungsvorschrift in Alltagssprache formulieren können.
- Sie sollen die Eigenschaften Endlichkeit, Eindeutigkeit und Ausführbarkeit eines Algorithmus kennen.

4.3.2 Verbesserung der Hausaufgabe und Wiederholung

Ein Schüler stellt seine Lösung vor. Diese wird - wenn nötig - ergänzt und es werden andere Lösungsvorschläge besprochen. Dabei werden die Begriffe Anweisung und Algorithmus wiederholt.

4.3.3 Motivation



Letzte Stunde habt ihr für das Spiel "memory" eine Spielanleitung verfasst.

4.3.4 Problemfrage 1

Woher wissen wir, wann das Spiel zu Ende ist?

4.3.5 Erarbeitung 1

Dabei stellt sich heraus, dass das Spiel zu Ende ist, wenn alle Kärtchen aufgedeckt sind und der Sieger feststeht. Somit enthält das Spiel endlich viele Anweisungen.

4.3.6 Sicherung 1

Auf das Arbeitsblatt "Beschreibung von Abläufen" von letzter Stunde wird der Begriff der "Endlichkeit" notiert und das entsprechende Beispiel zum "memory"-Spiel notiert.

4.3.7 Problemfrage 2

Wissen wir an jeder Stelle des Algorithmus, was wir zu tun haben?

4.3.8 Erarbeitung 2

Der Algorithmus legt die Reihenfolge fest. Wenn jemand beispielsweise ein "memory"-Paar findet, ist er noch einmal dran.

4.3.9 Sicherung 2

Das entsprechende Beispiel zum "memory"-Spiel wird unter 1. notiert.

4.3.10 Problemfrage 3

Unter 2. steht, dass der Algorithmus stets das gleiche Resultat liefert, wenn es mit den gleichen Voraussetzungen gestartet wird. Kann ich dann sagen, wenn die gleichen zwei Spieler gegeneinander spielen, dann gewinnt immer der gleiche? Was muss erfüllt sein, damit diese Aussage zutrifft?

4.3.11 Erarbeitung 3

Die Voraussetzungen müssen jedes Mal gleich sein. Die Karten müssen immer auf den gleichen Plätzen liegen und die zwei Spieler immer die gleichen Aktionen durchführen. Wenn das alles zutrifft, dann gewinnt immer der gleiche Spieler.

4.3.12 Sicherung 3

Das entsprechende Beispiel zum "memory"-Spiel wird auf dem Arbeitsblatt unter 2. notiert und über 1. der Begriff der "Eindeutigkeit".

4.3.13 Problemfrage 4

Woher wissen wir, dass jeder mit dieser Anleitung das Spiel spielen kann?

4.3.14 Erarbeitung 4

Die Anweisungen sind klar formuliert. Er muss gar nicht lange überlegen, wie er die jeweilige Anweisung ausführen muss.

4.3.15 Sicherung 4

Es wird der Begriff der "Ausführbarkeit" notiert und das entsprechende Beispiel zum "memory"-Spiel notiert.

4.4 Kennenlernen von Scratch

4.4.1 Lernziele

- Die Schüler sollen die Begriffe Klasse, Objekt, Attribut, Attributwert und Methode kennen.
- Sie sollen die Programmierumgebung Scratch kennen lernen.
- Sie sollen erkennen, was die einzelnen Anweisungen in Scratch bewirken.

4.4.2 Wiederholung

Bevor die Schüler mit Scratch in Berührung kommen, werden die Begriffe Klasse, Objekt, Attribut, Attributwert und Methode anhand von Grafik-Objekten (z. B. Objekte der Klasse RECHTECK, KREIS, ...) wiederholt.

4.4.3 Einführung in Scratch

Die Schüler erhalten eine kurze Einführung in Scratch:

- wie man Objekte erzeugen kann,
- wie man Objekte malen kann,
- dass ein Programms mit "Wenn Fahne angeklickt" beginnt,
- dass die Blöcke wie LEGO-Bausteine zusammengesetzt werden können,
- wie man ein Programm starten oder stoppen kann,
- wie man einem Objekt Kostüme hinzufügen kann und
- wie man ein Objekt Kostüme wechseln lassen kann.

4.4.4 Arbeitsauftrag

Erforsche Scratch weiter und erstelle einen Animationsfilm!

Untersuche dabei,

- ob alle Objekte, die du verwendest, der gleichen Klasse angehören und
- welche Attribute und Methoden die jeweiligen Objekte haben.

4.4.5 Begründung

Die Schüler sollen sich selbstständig in Scratch einarbeiten. Da Scratch sehr übersichtlich ist und sich vieles von selbst erklärt, stellt es für die Schüler kein Problem dar. Die Schüler entdecken viele Möglichkeiten, die ihnen die Programmierumgebung bietet. Sie werden in dieser entdeckenden Phase schon an die Arbeitsweise der kommenden Stunden herangeführt.

4.5 Kennenlernen von Scratch und Sequenz

4.5.1 Lernziele

- Die Schüler sollen wissen, dass das Objekt BÜHNE alle Objekte der Klasse SPRITE enthält.
- Sie sollen wissen, was man unter einem Methodenaufruf versteht.
- Sie sollen wissen, was man unter einer Sequenz versteht.
- Sie sollen das Struktogramm für eine Sequenz kennen.

4.5.2 Motivation



Die Katze soll zuerst "Hallo" sagen, ihr Kostüm wechseln, 100er Schritt gehen, ihr Kostüm wechseln und "miau" sagen. Wo findet man die einzelnen Bausteine, um dieses Programm zusammenzustellen?

4.5.3 Erarbeitung

Die Schüler nennen bzw. zeigen, wo die entsprechenden Bausteine zu finden sind. Sie bemängeln jedoch, dass alles viel zu schnell abläuft. Daher wird vor und nach dem Gehen der Baustein "warte 1 Sek." eingefügt.

Eine solche Folge von nacheinander auszuführender Anweisungen wird als Sequenz bezeichnet. Bei jeder Anweisung wird eine Methode aufgerufen.

Es werden die Methoden und Attribute der Objekte erarbeitet. Dabei wird festgestellt, dass das Objekt Bühne andere Attribute hat und einige Methoden gar nicht besitzt. Zudem besteht eine Beziehung zwischen dem BÜHNEN-Objekt und den SPRITE-Objekten.

4.5.4 Sicherung

Die Schüler erhalten das Arbeitsblatt "Algorithmen mit Scratch". Auf diesem ist das Klassendiagramm bereits abgebildet. Es müssen jedoch zur Wiederholung die Bestandteile einer Klassenkarte notiert werden.

Das Struktogramm der Sequenz wird eingeführt. Die Schüler müssen das links stehende Programm in das Struktogramm übertragen.

4.5.5 Fortsetzung des Arbeitsauftrages und Präsentation der Ergebnisse

Die Schüler haben nun Zeit, ihren Animationsfilm fertig zu stellen. Danach werden einige Animationsfilme vorgestellt.

4.5.6 Begründung

Die Erstellung des Animationsfilmes hat den Zweck, dass die Schüler zum einen Scratch kennen lernen und zum anderen ihrer Kreativität freien Lauf lassen. Das motiviert sie, so dass sie sich Scratch vielleicht auch zuhause herunterladen - Link auf Arbeitsblatt unten - und sich somit auch in ihrer Freizeit häufiger damit beschäftigen.

Ein Großteil der Schüler hat sich Scratch heruntergeladen.

Einige Schüler haben zu Hause ihrem Animationsfilm erweitert und neu Gelerntes mit eingebaut:

Eine Love-Story:



4.6 Wiederholung mit fester Anzahl

4.6.1 Lernziele

- Die Schüler sollen wissen, was man unter einer Wiederholung mit fester Anzahl versteht.
- Sie sollen das zugehörige Struktogramm kennen.
- Sie sollen wissen, was man unter einem Struktogramm versteht.
- Sie sollen wissen, dass Wiederholungen geschachtelt werden können.
- Sie sollen die Wiederholung mit fester Anzahl in Scratch umsetzen können.

4.6.2 Motivation



Den Schülern werden einige Methoden gezeigt, die sie für den nächsten Arbeitsauftrag benötigen. Mit der Methode "Stift runter" wird der Weg eines Objektes wie eine Spur markiert. Man muss es sich so vorstellen, als hätte das Objekt unter sich einen Stift. Mit "Stift hoch" wird diese Funktion wieder "deaktiviert". Mit der Anweisung "wische Malspuren weg" können die Malspuren gelöscht werden.

4.6.3 Arbeitsauftrag

Lasse die Schildkröte ein Quadrat laufen.

- 1. Zeichne zuerst ein Struktogramm!
- 2. Setze dein Struktogramm in Scratch um!

Nach ungefähr 10 Minuten stellt ein Schüler seine Lösung vor.

4.6.4 Problemfrage

Kann man das Programm vereinfachen?

4.6.5 Erarbeitung

Die Schüler erkennen, dass das Programm viermal die gleichen Anweisungsblöcke verwendet. Einige Schüler haben bereits beim Animationsfilm selbstständig eine Wiederholung verwendet und wissen die Lösung sofort. Es wird der Begriff der "Wiederholung mit fester Anzahl" eingeführt.

4.6.6 Sicherung

Die Schüler erhalten das Arbeitsblatt "Wiederholung mit fester Anzahl". Es werden die ersten Lücken ausgefüllt und das Struktogramm für die Wiederholung mit fester Anzahl eingeführt.

Die Schüler müssen das links stehende Programm in das Struktogramm übertragen.

Es wird festgehalten, dass Wiederholungen auch verschachtelt sein können. Dieses Thema wird in der nächsten Stunde weiter vertieft.

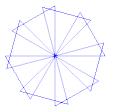
Zudem wird die Bedeutung des Begriffes Struktogramm notiert, die bereits in der letzten Stunde erwähnt wurde.

4.6.7 Arbeitsauftrag

1. Lasse ein Objekt ein Dreieck laufen.



- (a) Zeichne zuerst ein Struktogramm!
- (b) Setze dein Struktogramm in Scratch um!
- 2. Lasse das Objekt acht mal ein Dreieck laufen, so dass am Ende folgendes Muster entsteht:



- (a) Setze es in Scratch um!
- (b) Zeichne das Struktogramm dazu!

Die Lösung wird am Ende der Stunde vorgestellt.

4.6.8 Begründung

Die gewählten Beispiele sind Anwendungen der von Seymour Papert propagierten Turtle Grafik.

Der Begriff Struktogramm war bereits Gegenstand der letzten Stunde. Jedoch konnten die Schüler mit diesem Begriff noch nicht viel anfangen. Nachdem sie aber bereits eine weitere Kontrollstruktur und das zugehörige Struktogramm kennen gelernt haben, ist es sinnvoller an dieser Stelle, die Bedeutung des Begriffes festzuhalten. Trotzdem sehen die Schüler nicht viel Sinn im Zeichnen der Struktogramme, da ein Programm in Scratch fast genauso aussieht. Daher muss man damit argumentieren, dass es unterschiedliche (quelltextbasierte) Programmiersprachen gibt. In allen muss ein Programm anders umgesetzt werden. Das Struktogramm jedoch ist immer gleich.

4.7 Verschachtelte Wiederholungen mit fester Anzahl

4.7.1 Lernziele

- Die Schüler sollen wissen, wie verschachtelte Wiederholungen in Scratch umgesetzt werden.
- Sie sollen wissen, wie verschachtelte Wiederholungen im Struktogramm umgesetzt werden.
- Sie sollen die Abarbeitung einer verschachtelten Wiederholung erklären können.
- Sie sollen in der Lage sein, ein nicht ganz einfaches Problem mit verschachtelten Wiederholungen selbstständig zu lösen.

4.7.2 Arbeitsauftrag

In dieser Stunde sollen die Schüler die Möglichkeit haben, sich über die ganze Unterrichtsstunde mit einem nicht ganz einfachen Problem zu beschäftigen.

Damit der Arbeitsauftrag nicht zu schwer ist, ist er in mehrere Teilschritte vorstrukturiert.

Das bereits Gelernte soll auf einen neuen Sachverhalt angewendet werden.

Ziel: Die Katze soll mit der Malfunktion eine Blume zeichnen.

1. Lasse die Katze einen Halbkreisbogen zeichnen!



2. Lasse die Katze einen Halbkreisbogen in die andere Richtung zeichnen, so dass ein Blütenblatt entsteht!



3. Lasse nun die Katze das Blütenblatt so oft malen, bis eine Blume entsteht!



4. Versuche deine Blume bunt zu gestalten! Tipp: Dafür benötigst du eine weitere Methode!



5. Zeichne das Struktogramm dazu!

4.7.3 Begründung und Evaluation

Dieses Problem können die wenigsten Schüler auf Anhieb lösen. Einige tun sich bereits bei der 1. Aufgabe schwer. Daher war an dieser Stelle eine Hilfestellung nötig. Bei Aufgabe 2 sind nur wenige Schüler darauf gekommen, dass sich die Katze zuerst um 90 Grad drehen muss, bevor sie

4 VORGEHEN IM UNTERRICHT

den zweiten Halbkreisbogen zeichnet. Es hat geholfen, die Schüler zu fragen, was sie an Stelle der Katze tun müssten.

Ein Großteil der Schüler hat anschließend seiner Fantasie freien Lauf gelassen. Sie haben alle eine Blume gezeichnet, jedoch jeder eine ganz andere. Dabei haben alle verschachtelte Wiederholungen mit fester Anzahl verwendet, was der Sinn dieses Arbeitsauftrages war.



4.8 Bedingte Wiederholung

4.8.1 Lernziele

- Die Schüler sollen wissen, was man unter einer Bedingung versteht.
- Sie sollen Bedingungen in Scratch kennen.
- Sie sollen wissen, was man unter einer bedingten Wiederholung versteht.
- Sie sollen die bedingte Wiederholung in Scratch umsetzen können.
- Sie sollen das zugehörige Struktogramm kennen.

4.8.2 Motivation



Die Katze möchte bis zum nächsten Kaktus laufen. Anhand dieses Problems wird die Wiederholung mit fester Anzahl wiederholt.

4.8.3 Problemfrage

Wenn die Katze aber viel weiter vom Kaktus entfernt ist, dann gelangt sie mit diesem Programm nicht zum Kaktus. Ist die Katze viel näher am Kaktus, läuft sie an diesem vorbei. Wie können wir das Problem lösen?

4.8.4 Erarbeitung

Die Schüler kommen schnell darauf, dass die Katze solange laufen muss, bis sie den Kaktus bzw. die Farbe grün berührt. An dieser Stelle führt man den Begriff "Bedingung" ein und zeigt den Schülern alle möglichen Bedingungen

in der Kategorie Fühlen und die Möglichkeit, eine Bedingung zu verneinen.

Der eine oder andere Schüler hat bereits die bedingte Wiederholung ausprobiert und weiß somit die Lösung für das Problem.

4.8.5 Sicherung

Die Schüler erhalten ein Arbeitsblatt mit "Bedingungen" auf der Vorderseite und "Bedingte Wiederholung" auf der Rückseite. Es werden die ersten Lücken auf der Vorderseite ausgefüllt und die Bedingungen in Scratch durchgesprochen.

Auf der Rückseite wird die Definition einer bedingten Wiederholung festgehalten.

Die Schüler erraten das Struktogramm für die bedingte Wiederholung, das sich von der Wiederholung mit fester Anzahl kaum unterscheidet.

Die Schüler müssen das links stehende Programm in das Struktogramm übertragen.

4.8.6 Arbeitsauftrag

1. Lasse einen Ball auf den Rand der Bühne zu bewegen und von diesem abprallen!



2. Verwende folgenden Baustein aus der Kategorie Steuerung, um das Programm immer wiederholen zu lassen:



- 3. Lasse den Ball beim Abprallen drehen!
- 4. Verändere dein Programm so, dass der Ball in unterschiedlichen Winkeln vom Rand abprallt!

Tipp: Verwende den Baustein Zufallszahl von (1 bis 45) aus der Kategorie Zahlen!

5. Zeichne das Struktogramm dazu! Hinweis: Das Struktogramm für die Kontrollstruktur "wiederhole fortlaufend" sieht genauso aus, wie die Struktogramme für die Wiederholung mit fester Anzahl bzw. für die Bedingte Wiederholung. Der Text oben muss "wiederhole fortlaufend" heißen.

Die Lösung wird am Ende der Stunde besprochen. Dabei wird auch auf die unendliche Wiederholung eingegangen.

4.9 Zweiseitig bedingte Anweisung

4.9.1 Lernziele

- Die Schüler sollen wissen, was man unter einer zweiseitig bedingten Anweisung versteht.
- Sie sollen das zugehörige Struktogramm kennen.
- Sie sollen die zweiseitig bedingte Anweisung in Scratch umsetzen können.

4.9.2 Motivation



Als Motivation wird das Ergebnis des Arbeitsauftrages der letzten Stunde verwendet. Der Ball bewegt sich solange auf den Bühnenrand zu, bis er ihn berührt. Danach dreht er sich um einen zufälligen Winkel zwischen 1° und 45° und prallt vom Rand ab. Dies wird unendlich wiederholt. Wir wollen das Programm zu einem kleinen Spiel erweitern. Wird der Ball mit dem Mauszeiger berührt, sagt der Ball "Du hast mich erwischt!", ansonsten sagt er "Fang mich!".

4.9.3 Problemfrage

Wie können wir auf das Ereignis, ob der Ball vom Mauszeiger berührt wird oder nicht, unterschiedlich reagieren?

4.9.4 Erarbeitung

Das Problem wird mit der zweiseitig bedingten Anweisung gelöst. Wird der Ball vom Mauszeiger berührt, sagt er "Du hast mich erwischt!". Ansonsten sagt er "Fang mich!".

4.9.5 Sicherung

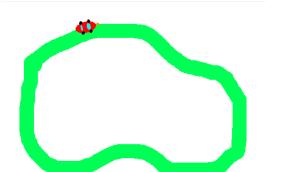
Die Schüler erhalten ein Arbeitsblatt zur "Zweiseitig bedingten Anweisung". Es wird die Definition der zweiseitig bedingten Anweisung im Lückentext festgehalten.

Das Struktogramm wird gemeinsam erarbeitet. Die Schüler werden darauf hingewiesen, dass es sich um das erste Struktogramm handelt, das von Scratch abweicht.

Die Schüler müssen das links stehende Programm in das Struktogramm übertragen.

4.9.6 Arbeitsauftrag

Diesen Arbeitsauftrag beginnen die Schüler im Unterricht. Er soll aber zuhause fertig gemacht werden und das Struktogramm einen Tag vor der nächsten Stunde der Lehrkraft ins Fach gelegt werden.



- 1. Male ein Objekt Rennstrecke.
- 2. Male ein Objekt Auto. Füge vor dem Auto noch einen kleinen Kreis hinzu, der eine noch nicht verwendete Farbe enthält. Dieser Kreis dient als Sensor, mit dem das Auto erkennen soll, ob er sich auf der Strecke oder daneben befindet.
- 3. Das Auto-Objekt soll sich nun mithilfe des Sensors auf dem Rennstrecke-Objekt bewegen. Überlege dir eine Strategie und erstelle zuerst das zugehörige Struktogramm! Tipp: Wie muss das Auto jeweils reagieren, wenn es links oder rechts
- 4. Setze das Struktogramm als Programm um!

4.9.7 Begründung und Evaluation

von der Strecke abkommt!

In der nächsten Stunde erhalten die Schüler ihre korrigierten Hausaufgaben. Die Hausaufgabe wird verbessert. Danach wurde in den meisten Klassen eine Stegreifaufgabe geschrieben. Diejenigen, die ihre Hausaufgaben gemacht haben, sind auch in der Stegreifaufgabe gut abgeschnitten.

Die Aufgabe wurde von den Schülern sehr gut bearbeitet. Viele hatten noch andere Ideen und haben sich eine bunte Linie in die Fahrbahn gemalt, an der sich das Auto orientieren konnte. Diese Lösung hat bewirkt, dass das Auto nicht entlang des Fahrbahnrandes gefahren ist, sondern auf der Fahrbahn.

4.10 Einseitig bedingte Anweisung

4.10.1 Lernziele

- Die Schüler sollen wissen, was man unter einer einseitig bedingten Anweisung versteht.
- Sie sollen das zugehörige Struktogramm kennen.
- Sie sollen die einseitig bedingten Anweisung in Scratch umsetzen können.

4.10.2 Motivation



Die Katze möchte gerne am Hürdenlauf teilnehmen.

4.10.3 Problemfrage

Was macht die Katze, um über die Hürden zu laufen?

4.10.4 Erarbeitung

Die Schüler beschreiben, dass die Katze die ganze Zeit vorwärts läuft. Wenn sie auf eine Hürde trifft, dann springt sie über die Hürde.

Zuerst wird das Vorwärtslaufen realisiert. Damit die Katze es andauernd macht, muss alles fortlaufend wiederholt werden. Die Katze springt über eine Hürde, wenn sie die Farbe blau berührt. Das Springen wird mit einer zweiseitig bedingten Anweisung realisiert. Wenn die Bedingung nicht zutrifft, wird nichts ausgeführt. Das bezeichnet man als einseitig bedingte Anweisung. Dafür gibt es in Scratch einen eigenen Block in der Kategorie

Steuerung.

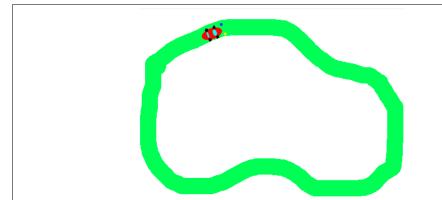
Das Aussehen des Struktogramms einer einseitig bedingten Anweisung wird diskutiert. Es handelt sich um das Struktogramm einer zweiseitig bedingten Anweisung, bei dem die Seite, in der die Bedingung nicht erfüllt ist, einfach gestrichen wird.

Der Begriff Kontrollstruktur wird bereits seit Wochen im Unterricht verwendet, wurde jedoch noch nicht festgehalten. Die einseitig bedingte Anweisung ist jetzt die letzte Kontrollstruktur, die die Schüler kennenlernen. Es werden in dem Zusammenhang alles bereits gelernten Kontrollstrukturen aufgezählt.

4.10.5 Sicherung

Die Schüler erhalten ein Arbeitsblatt zur "Einseitig bedingten Anweisung". Da die einseitig bedingte Anweisung als Variation der zweiseitig bedingten Anweisung eingeführt wurde, müssen die Schüler nichts mehr selbst ausfüllen.

4.10.6 Arbeitsauftrag



- 1. Male ein Objekt Rennstrecke.
- 2. Male ein Objekt Auto. Füge vor dem Auto noch zwei kleine Kreise unterschiedlicher Farbe hinzu, die jeweils eine noch nicht verwendete Farbe enthalten. Diese Kreise dienen als Sensoren, mit denen das Auto erkennen soll, ob es sich auf der Strecke oder daneben befindet.
- 3. Das Auto-Objekt soll sich nun mithilfe der zwei Sensoren auf dem Rennstrecke-Objekt bewegen. Überlege dir eine Strategie und erstelle zuerst das zugehörige Struktogramm!

 Tipp: Wie erkennt das Auto, ob es links oder rechts von der Strecke
- 4. Setze das Struktogramm als Programm um!

abgekommen ist? Wie muss es jeweils reagieren?

4.10.7 Evaluation

Die Aufgabe wurde von den Schülern sehr gut bearbeitet. Viele hatten noch andere Ideen und haben sich bunte Linien in die Fahrbahn gemalt, an denen sich das Auto orientieren konnte.

4.11 Methoden und Botschaften

4.11.1 Lernziele

- Die Schüler sollen wissen, durch welche vier Startbausteine das Ausführen von Methoden in Scratch veranlasst werden kann.
- Ihnen soll bewusst sein, dass gleichzeitig mehrere Methoden ausgeführt werden können.
- Sie sollen wissen, dass Objekte durch Botschaften miteinander kommunizieren können.
- Sie sollen erkennen, dass durch eine Botschaft alle gleichnamigen Methoden aufgerufen werden.

4.11.2 Motivation





Die Biene fliegt auf den Jungen zu und sticht ihn. Wir gehen davon aus, die Biene hat ihren Stachel vorne und nicht hinten.

4.11.3 Problemfrage

Was können wir tun, dass der Junge erst dann "Aua" sagt, wenn er tatsächlich gestochen wird?

4.11.4 Erarbeitung

Zuerst wird das Fliegen der Biene bis zum Jungen realisiert. Als Bedingung wird zum einen "wird Objekt Junge" berührt genannt oder "wird Farbe rot berührt". Da nicht bekannt ist, wohin die Biene den Jungen sticht, wird die

Bedingung "wird Objekt berührt" verwendet.

Der Junge soll "Aua!" sagen, wenn er gestochen wird. Wird das Programm des Jungen mit dem Startbaustein "Wenn Fahne angeklickt" verwendet, dann sagt der Junge "Aua" schon bevor er gestochen wurde.

Geht man davon aus, dass durch den Stich beim Jungen erst ein Nervensignal ausgelöst wird, dann muss dieses zunächst an das Gehirn geschickt werden. Erst dann verspürt der Junge einen Schmerz und signalisiert diesen mit dem Ausspruch "Aua".

In Scratch muss die Biene dem Jungen ein Signal - eine sogenannte Botschaft - schicken, dass sie ihn gestochen hat. Deswegen wird die zu sendende Botschaft "pieks" genannt und der Baustein "sende pieks an alle" verwendet. Somit wird die Methode "pieks" des Jungen aufgerufen.

Damit dies auch funktioniert, muss der Startbaustein beim Programm des Jungen durch "Wenn ich pieks empfange" ersetzt werden. Der Junge sagt erst dann "Aua", wenn die Biene ihn gestochen hat.

Durch das Senden und Empfangen einer Botschaft, kommunizieren die Objekte miteinander.

Nun wird der Junge dupliziert. Dem neuen Jungen-Objekt wird der Name Junge2 gegeben. Das Objekt Junge2 hat genau das gleiche Programm wie das Objekt Junge. Wenn das Programm jetzt ausgeführt wird, sagen beide Jungen-Objekte "Aua", obwohl das Objekt Junge gestochen wird. Versetzt man die Biene, so dass sie auf Junge2 zufliegt, fliegt sie an ihm vorbei.

Die Schüler erkennen sofort, dass es an der Bedingung liegt. Diese bezieht sich nur auf das Junge-Objekt.

Damit aber der Junge2 nicht "Aua" sagt, wenn das Objekt Junge gestochen wird, muss die Methode bei Junge2 auf beispielsweise "pieks2" umbenannt werden.

Die Schüler sehen, dass der Baustein "sende pieks an alle" dazu führt, dass von jedem Objekt, das die Methode namens "pieks" enthält, diese auch ausführt.

Die Schüler werden auf den Unterschied zwischen den Bausteinen "sende ... an alle" und "sende ... an alle und warte" aufmerksam gemacht.

4.11.5 Sicherung

Die Schüler erhalten ein Arbeitsblatt zu "Methoden und Botschaften". Da die Schüler viele der Konzepte bereits intuitiv richtig verwendet haben und das Beispiel mit dem Senden der Botschaften sehr einleuchtend ist, müssen die Schüler nichts mehr einfügen.

4.11.6 Arbeitsauftrag



- 1. Erzeuge drei unterschiedliche Personen-Objekte.
- 2. Füge jedem Objekt weitere Kostüme hinzu.
- 3. Lasse die drei Personen tanzen! Tipp: Kostümwechsel
- 4. Ändere dein Programm so ab, dass die Personen nacheinander tanzen. D. h. die erste Person beginnt zu tanzen und teilt der zweiten mit, wenn sie fertig ist, usw.
- 5. Erstelle Struktogramme!

Es wird eine mögliche Lösung vorgestellt.

4.12 Variablen (Exkurs)

4.12.1 Lernziele

- Die Schüler sollen wissen, dass eine Variable einen Wert abspeichert.
- Sie sollen den Unterschied zwischen einer lokalen und einer globalen Variablen erkennen.
- Sie sollen wissen, dass eine Variable initialisiert werden muss.
- Sie sollen wissen, dass man den Wert einer Variable verändern kann.

4.12.2 Motivation



Wir möchten ein Spiel programmieren, bei dem ein Geldsack herunterfallende Münzen einsammelt. Der Geldsack soll mit den Pfeiltasten nach links und rechts bewegt werden können. Es soll anfangs nur eine Münze herunterfallen, die aber jedes Mal von einer anderen - also einer zufälligen - Position herunterfällt.

4.12.3 Problemfrage

Es wäre interessant, zu zählen, wie viele Münzen bereits im Geldsack sind. Wie speichern wir den Spielstand ab?

4.12.4 Erarbeitung

Bevor das Abspeichern realisiert wird, wird erst einmal das eigentliche Spiel benötigt. Dazu werden zwei Objekte Geldsack und Münze gezeichnet.

Den Geldsack möchte man mit der Tastatur steuern können. Einige Schüler haben dies bereits verwendet und erläutern die Steuerungsmöglichkeit. Man benötigt dazu die Startbausteine "Wenn Taste Pfeil nach links gedrückt" bzw. "Wenn Taste Pfeil nach rechts gedrückt" und die entsprechenden Anweisungen.

Als nächstes soll die Münze herunterfallen, bis sie den Geldsack berührt. Um dies zu realisieren, muss der y-Wert erniedrigt werden. Damit es nicht zu schnell abläuft, muss noch "warte ... Sek." an die entsprechende Stelle hinzugefügt werden. Wenn die Münze den Geldsack berührt, versteckt sie sich. Jedoch soll sie immer wieder von oben starten. Somit muss die x- und y-Position der Münze immer auf einen bestimmten Wert gesetzt werden. Und man muss die Münze wieder anzeigen lassen.

Der Fall, dass die Münze nicht vom Geldsack eingesammelt wird, wird weggelassen. Das ist eine Erweiterungsmöglichkeit. Es wird davon ausgegangen, dass der Geldsack alle Münzen auffängt.

Da man den Geldsack bewegen kann, möchte man auch, dass die Münze von unterschiedlichen Stellen und unterschiedlich hoch zu fallen beginnt. Dafür muss man den Zufallszahl-Baustein aus der Kategorie "Zahlen" verwenden.

Jetzt fehlt nur noch der Spielstand. Dafür benötigt man eine Variable. Variablen werden im Speicher abgelegt. Man kann sich den Speicher wie einen Schrank mit vielen Schubladen vorstellen (siehe 3.3.3). Beim Erzeugen einer Variable erhält eine Schublade den Namen der Variable. Am Anfang benötigt die Variable einen Startwert, auf den sie gesetzt werden muss. Sie muss somit initialisiert werden, d. h. man steckt einen Zettel mit dem Anfangswert der Variable in die Schublade. Greift man auf die Variable zu, wird der Wert der Variablen nachgesehen. Erhält die Variable einen neuen Wert, wird der alte Zettel, der in der Schublade liegt, weggeworfen und ein neuer Zettel mit dem aktuellen Wert in die Schublade gesteckt.

Man möchte, dass der Geldsack den Spielstand verwaltet. Er soll wissen, wie viele Münzen in ihm sind. Daher wird das Objekt Geldsack ausgewählt und in der Kategorie "Variablen" eine lokale Variable namens "spielstand" angelegt.

Den Schülern wird der Unterschied zwischen der lokalen und globalen Variable so erklärt, dass auf die lokalen Variablen nur das Objekt zugreifen kann, dem es "gehört", auf globale können alle zugreifen.

Der Variable wird, wenn auf die Fahne geklickt wird, der Startwert 0 gegeben.

Sobald die Münze den Geldsack berührt, soll der Spielstand hochgezählt werden. Jedoch hat die Münze auf den Spielstand keinen Zugriff und muss somit dem Geldsack eine Botschaft schicken, dass sie bereits im Geldsack ist. Wenn der Geldsack diese Botschaft erhält, erhöht er den Spielstand.

Das Spiel soll enden, sobald mehr als fünf Münzen im Geldsack sind. Daher wird zuerst die Bedingung formuliert. In der Kategorie "Zahlen" gibt es weitere Bedingungen, mit denen man Werte vergleichen kann. Man kann als linken Wert die Methode "spielstand" einsetzen - es wird automatisch der aktuelle Wert eingefügt - und mit einem Zahlenwert - wie z. B. 5 - vergleichen. Somit ergibt sich die Bedingung "spielstand > 5". Mit der einseitig bedingten Anweisung kann diese Bedingung überprüft werden. Wenn diese wahr ist, dann sagt der Geldsack "Spielende!" und es wird alles gestoppt. Wählt man nur "stoppe Programm", so wird nur das Programm des Geldsacks, aber nicht das der Münze gestoppt.

Das Spiel könnte man noch folgendermaßen erweitern:

- Es fängt den Fall nicht ab, wenn die Münze vom Geldsack nicht aufgefangen wird und auf den unteren Rand fällt. Das Programm läuft an dieser Stelle nicht weiter.
- Man könnte eine Variable "leben" anlegen, der man als Startwert 3 gibt. Wenn eine Münze nicht aufgefangen wird, wird das Leben reduziert. Sind alle Leben verbraucht, ist das Spiel zu Ende.
- Wenn man das Münzen-Objekt mehrmals kopiert zehn Münzen-Objekte genügen -, wird das Spiel schwieriger.
- Man könnte eine Variable "level" einfügen. Ab einem gewissen Spielstand, gelangt man in ein höheren Level, in dem dann mehr Münzen fallen. Man könnte sie auch in jedem Level schneller fallen lassen.

• ...

Einige der Erweiterungen wurden in der folgenden Arbeitsphase von leistungsstärkeren Schülern im Rahmen der Binnendifferenzierung umgesetzt.

4.12.5 Sicherung

Die Schüler erhalten ein Arbeitsblatt zu "Variablen". Da dieses Thema einen Exkurs darstellt und auf dem Arbeitsblatt hauptsächlich das erarbeitete Spiel abgebildet ist, müssen die Schüler nichts mehr ergänzen.

4.12.6 Arbeitsauftrag

Überlege dir ein weiteres Spiel mit einem Spielstandzähler.

Im Rahmen der Binnendifferenzierung können sich die leistungsschwächeren Schüler an einem Spiel versuchen, das sehr dem vorherigen ähnelt. Leistungsstärkere können sich an ein anderes Spiel wagen oder das vorherige ausbauen.

In der nächsten Stunde wird ein Spiel programmiert. Wer jetzt schon eine gute Idee hat, kann sich auch nächste Stunde gerne daran weiterarbeiten. Wem bis nächste Stunde nichts einfällt, der erhält einen detaillierten Arbeitsauftrag.

4.12.7 Begründung

Mit dem gewählten Beispiel sollten viele gelernte Konzepte wiederholt werden. Zudem sollte damit gezeigt werden, wie man die gelernten Inhalte verknüpfen kann.

In der Erarbeitung wurden die Ideen der Schüler natürlich mit berücksichtigt und diese mit eingebunden.

Der Arbeitsauftrag ist sehr frei gestellt, damit wurde den Schülern Platz für Kreativität geschaffen. Die Schüler die sich schwer taten, haben das erarbeitete Spiel minimal verändert und selbst ausprobiert. Andere wiederum hatten tolle Ideen, die sie in der nächsten Stunde weiter verfolgt haben.

4.13 Spielprogrammierung

4.13.1 Lernziele

• Die Schüler sollen ein eigenes Spiel programmieren.

4.13.2 Motivation

Diejenigen Schüler, die keine Spielidee haben, sollen das Spiel "Bananen fangen" programmieren. Jeder Schüler erhält daher ein Blatt mit dem Arbeitsauftrag.



Den Schülern wird das Spiel "Bananen fangen" vorgestellt. Die Katze wird mit der linken und rechten Pfeiltaste nach rechts und links bewegt. Ein Vogel fliegt immer vom linken zum rechten Bühnenrand und hält Bananen im Schnabel. Sobald jemand auf die Leertaste drückt, lässt der Vogel die Bananen fallen. Die Katze muss die Bananen auffangen. Sie erhält für jedes gefangene Bananenbündel 10 Pluspunkte, für jedes nicht gefangene Bündel 10 Minuspunkte.

Den Schülern wird gezeigt, wie sie es schaffen, dass der Vogel die Bananen mit sich führt.

4.13.3 Arbeitsauftrag

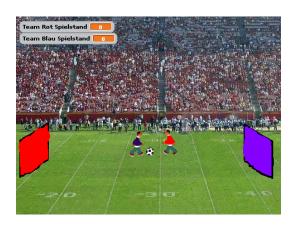
Die Schüler haben die Möglichkeit eine eigene Spielidee zu verfolgen oder diesen Arbeitsauftrag auszuführen.

Spiel "Bananen Fangen"

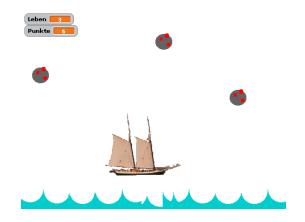
- 1. Kopiere die Kostüme, die das Objekt Katze bereits enthält!
- 2. Bearbeite jeweils die beiden kopierten Kostüme. Klicke dabei auf "Horizontal Spiegeln" und dann auf "Ok". Somit schaut die Katze in die andere Richtung!
- 3. Programmiere die Katze so, dass sie beim Drücken der linken Pfeiltaste sich nach links bewegt und beim Drücken der rechten Pfeiltaste nach rechts! Verwende dafür die vier Kostüme!
- 4. Füge ein Objekt Vogel mit zwei Kostümen hinzu!
- 5. Lasse den Vogel nun immer wieder vom linken zum rechten Rand fliegen. Das Fenster fängt bei x = -200 an und endet bei x = 200.
- 6. Setze die y-Koordinate vom Vogel mit einer Zufallszahl zwischen 0 und 200, damit der Vogel nicht immer auf der gleichen Höhe fliegen muss.
- 7. Füge ein Objekt Bananen hinzu!
- 8. Der Vogel hat die Bananen immer bei sich. Die Koordinaten des Vogels helfen dir beim Setzen der Koordinaten der Bananen.
- 9. Wenn jemand die Leertaste drückt, lässt der Vogel jedoch die Bananen fallen.
- 10. Die Katze will die Bananen fangen, möchte aber Punkte dafür erhalten. Erstelle beim Objekt Katze eine Variable "spielstand" und setze sie auf 0, wenn jemand die Fahne angeklickt hat!
- 11. Wenn die Bananen die Katze berühren, erhöht sich der Spielstand um 10 Punkte. Schafft sie es nicht, die Bananen zu fangen, dann werden 10 Punkte abgezogen.

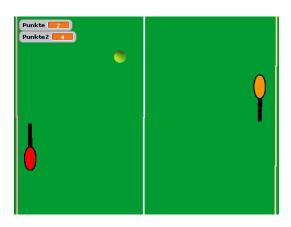
4.13.4 Evaluation

Die Schüler hatten schöne und sehr kreative Spielideen.













5 Leistungsnachweise und Evaluation

Auf den folgenden Seiten werden die Ergebnisse der Stegreifaufgaben vorgestellt und die von jeder Klasse im Einzelnen betrachtet. Danach werden sie innerhalb des Gymnasiums verglichen und zuletzt innerhalb der zwei Bamberger Gymnasien.

In den folgenden Diagrammen stellen die orangefarbenen Balken die Ergebnisse der Mädchen, die blauen die der Jungen, die grünen Balken die durchschnittlich erreichte Punktzahl und die roten die maximal erreichbare Punktzahl dar.

Die Angaben der Stegreifaufgaben und ihre Lösungen sind im Anhang zu finden.

Es wurden in allen Klassen unterschiedliche Stegreifaufgaben geschrieben. Die maximale Punktzahl betrug immer 20 Punkte. Es wurde folgender Notenschlüssel verwendet:

Note	von		bis
1	20	_	17
2	16,5	_	14
3	13,5	_	11
4	10,5	_	8
5	7,5	_	4
6	3,5	_	0

Die Stegreifaufgaben wurden nicht im Computerraum geschrieben, sondern in einem freien Klassenzimmer.

5.1 Klasse 7d am Dientzenhofer-Gymnasium Bamberg

Die Stegreifaufgabe wurde am 02.07.2008 geschrieben.

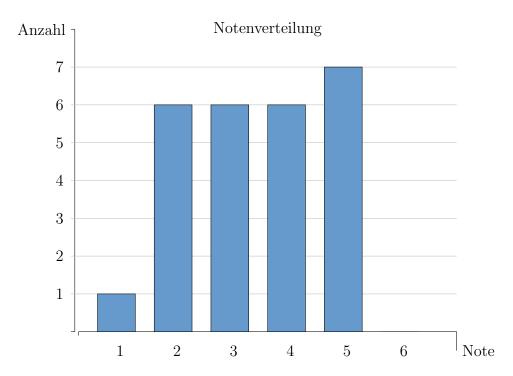
5.1.1 Auswahl der Aufgaben

Die vorangegangenen Stunden hatten die Themen "Zweiseitig bedingte Anweisung" und "Einseitig bedingte Anweisung".

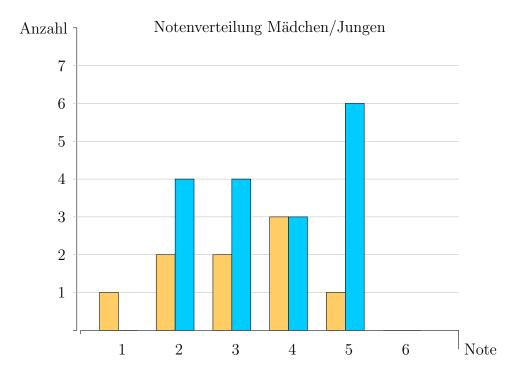
Die erste Aufgabe prüft das Grundwissen ab. Mit Aufgabe 2 soll festgestellt werden, ob der Schüler einen Überblick über den Stoff der letzten zwei Stunden hat (Reproduktion). Aufgabe 3a kann sowohl mit der Bedingten Wiederholung (Grundwissen) als auch mit dem Wissen aus den letzten zwei Stunden gelöst werden. Aufgabe 3b sollte mit der Einseitig oder der Zweiseitig bedingten Anweisung gelöst werden. Beide Aufgaben verlangen vom Schüler eine Anwendung des Gelernten auf neue Sachverhalte (Transfer).

5.1.2 Ergebnisse

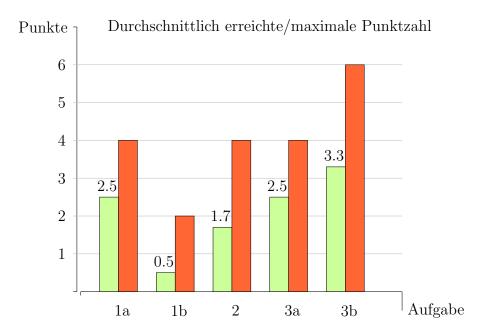
Von 30 Schülern haben 29 die Stegreifaufgabe mitgeschrieben. Darunter waren 9 Mädchen und 20 (anstatt 21) Jungen. Der Durchschnitt betrug $\emptyset 3, 46$.



Werden Mädchen und Jungen getrennt voneinander betrachtet, schneiden die Mädchen deutlich besser ab. Die Mädchen haben einen Durchschnitt von $\emptyset 3, 11$, die Jungen $\emptyset 3, 65$.



Wenn man die durchschnittlich erreichte Punktzahl mit der maximal erreichbaren vergleicht, sieht man, dass die Schüler im Durchschnitt bei den Theorieaufgaben 1b und 2 weniger als die Hälfte der Punkte haben. Darauf lässt sich schließen, dass der Großteil der Schüler sich nicht auf jede Stunde vorbereitet.



5.2 Klasse 7e am Dientzenhofer-Gymnasium Bamberg

Die Stegreifaufgabe wurde am 11.07.2008 geschrieben.

5.2.1 Auswahl der Aufgaben

Die vorangegangenen Stunden hatten die Themen "Einseitig bedingte Anweisung" und "Methoden und Botschaften".

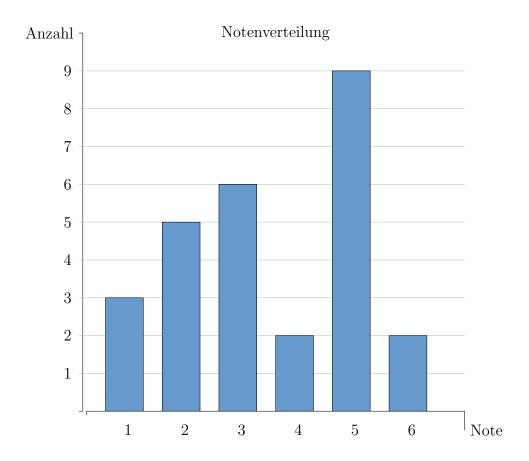
Die erste Aufgabe war Gegenstand der vorletzten Stunde und stellt somit eine Reproduktion dar. Aufgabe 2 ist eine Grundwissensaufgabe. In Aufgabe 3 wird eine Anwendung des Gelernten auf einen neuen Sachverhalt (Transfer) verlangt. Aufgabe 4 ist Grundwissen und wurde in der vorletzten Stunde wiederholt.

5.2.2 Ergebnisse

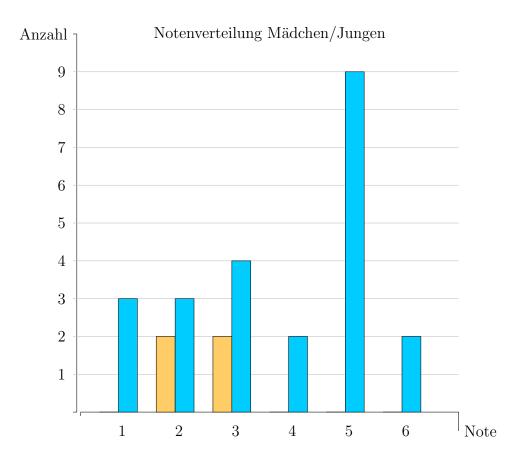
Alle 27 Schüler haben die Stegreifaufgabe mitgeschrieben. Darunter waren 4 Mädchen und 23 Jungen.

Der Durchschnitt betrug $\varnothing 3, 56$.

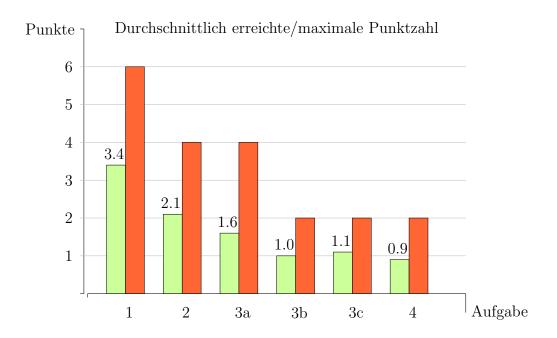
Das Ergebnis wurde jedoch durch eine Leistungsverweigerung (6) verfälscht. Ein Schüler war aufgrund eines Unfalls, bei dem er sich am Kopf verletzt hatte, für zwei Wochen nur von Schulaufgaben befreit.



Wenn man Mädchen und Jungen getrennt voneinander betrachtet, schneiden die Mädchen deutlich besser ab. Die Mädchen haben einen Durchschnitt von $\emptyset 2, 50$, die Jungen $\emptyset 3, 74$. Jedoch sind diese Ergebnisse aufgrund der geringen Mädchenanzahl nicht aussagekräftig.



Wenn man die durchschnittlich erreichte Punktzahl mit der maximal erreichbaren vergleicht, sieht man, dass die Schüler im Durchschnitt bei den Theorieaufgaben 3a und 4 weniger als die Hälfte der Punkte haben. Darauf lässt sich schließen, dass der Großteil der Schüler sich nicht auf jede Stunde vorbereitet.



5.3 Klasse 7a am Kaiser-Heinrich-Gymnasium Bamberg

Die Stegreifaufgabe wurde am 11.12.2008 geschrieben.

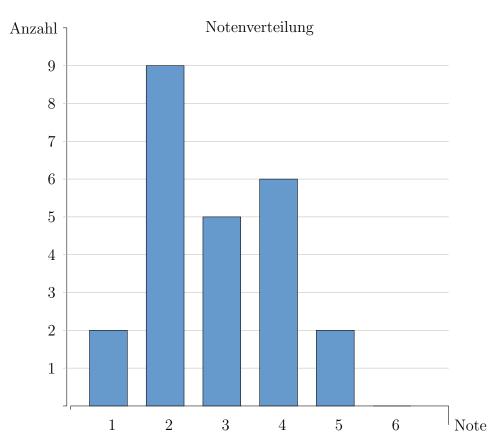
5.3.1 Auswahl der Aufgaben

Die vorangegangenen Stunden hatten die Themen "Bedingte Wiederholung" und "Zweiseitig bedingte Anweisung".

Die ersten zwei Aufgaben sind Grundwissensaufgaben. Aufgabe 3 ist eine Reproduktion. In Aufgabe 4 wird eine Anwendung des Gelernten auf einen neuen Sachverhalt (Transfer) verlangt.

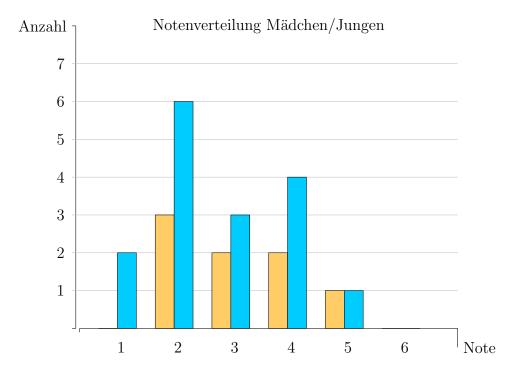
5.3.2 Ergebnisse

Von 27 Schülern haben 24 die Stegreifaufgabe mitgeschrieben. Darunter waren 8 (anstatt 9) Mädchen und 16 (anstatt 18) Jungen. Der Durchschnitt betrug $\emptyset 2, 88$.

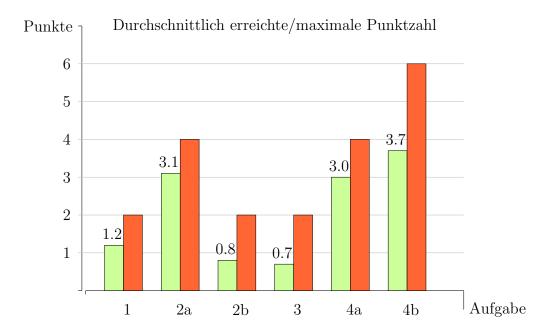


Die Klasse 7a ist eine sehr leistungsstarke Klasse, die sich für alles interessiert und sich von Stunde zu Stunde vorbereitet. Dies spiegelt sich in der Notenverteilung wieder.

Wenn man Mädchen und Jungen getrennt voneinander betrachtet, schneiden die Jungen deutlich besser ab. Die Mädchen haben einen Durchschnitt von $\emptyset 3, 13$, die Jungen $\emptyset 2, 75$.



Wenn man die durchschnittlich erreichte Punktzahl mit der maximal erreichbaren vergleicht, sieht man, dass die Schüler im Durchschnitt bei der Theorieaufgabe 1 mehr als die Hälfte und bei Theorieaufgabe 3 weniger als die Hälfte der Punkte haben. Jedoch schneidet diese Klasse in den Anwendungsaufgaben besser ab. Sie haben deutlich mehr als die Hälfte der Punkte.



5.4 Klasse 7b am Kaiser-Heinrich-Gymnasium Bamberg

Die Stegreifaufgabe wurde am 11.12.2008 geschrieben.

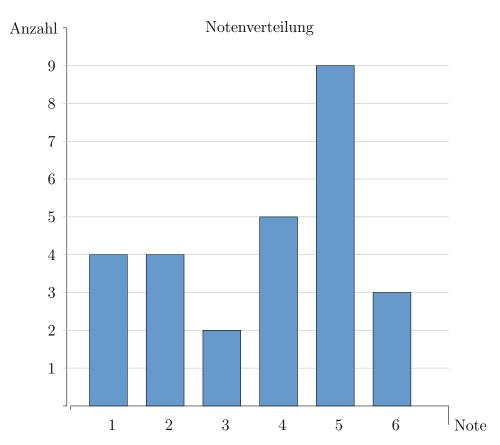
5.4.1 Auswahl der Aufgaben

Die vorangegangenen Stunden hatten die Themen "Bedingte Wiederholung" und "Zweiseitig bedingte Anweisung".

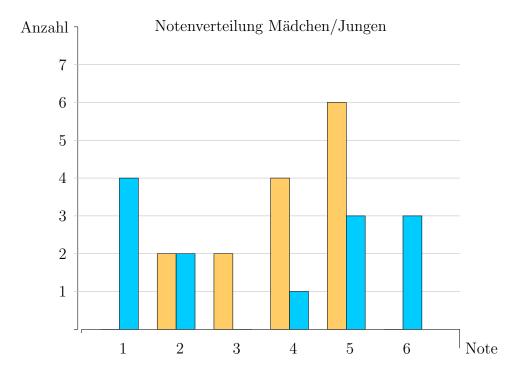
Die ersten zwei Aufgaben sind Grundwissensaufgaben. In Aufgabe 3 und 4 ist eine Anwendung des Gelernten auf einen neuen Sachverhalt (Transfer) verlangt. Aufgabe 5 ist eine Reproduktion.

5.4.2 Ergebnisse

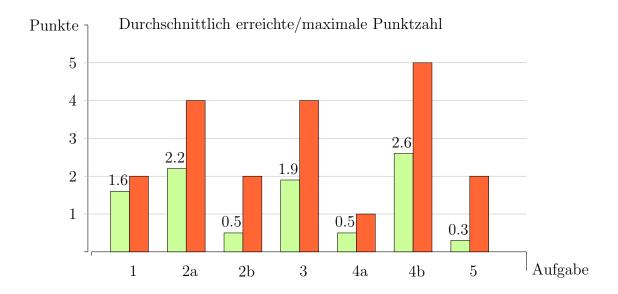
Von 29 Schülern haben 27 die Stegreifaufgabe mitgeschrieben. Darunter waren 14 (anstatt 15) Mädchen und 13 (anstatt 14) Jungen. Der Durchschnitt betrug $\emptyset 3, 74$.



Wenn man Mädchen und Jungen getrennt voneinander betrachtet, schneiden die Jungen deutlich besser ab. Die Mädchen haben einen Durchschnitt von $\emptyset 4,00$, die Jungen $\emptyset 3,46$.



Wenn man die durchschnittlich erreichte Punktzahl mit der maximal erreichbaren vergleicht, sieht man, dass die Schüler im Durchschnitt bei der Theorieaufgabe 1 noch gut abschneiden, dafür bei Aufgabe 1b und 5 jedoch fast nichts wissen. Daraus lässt sich schließen, dass die Schüler sich auf die ersten Stunden noch vorbereitet haben und dann nachlässig geworden sind. Bei den restlichen Aufgaben wurde etwas mehr als die Hälfte der Punkte erreicht.



5.5 Evaluation

5.5.1 Am Dientzenhofer-Gymnasium (DG) Bamberg

Die Klasse 7d und 7e schneiden ungefähr gleich ab.

Die Informatik wir nur mit einem Drittel für die Natur und Technik-Note gewichtet. Das ist ein Grund dafür, wieso einige Schüler - ganz besonders die Jungen - die Informatik als unwichtig erachten und sich nicht von Stunde zu Stunde vorbereiten.

Die Mädchen schneiden deutlich besser ab als die Jungen.

5.5.2 Am Kaiser-Heinrich-Gymnasium (KHG) Bamberg

Die Klasse 7a schneidet um fast eine Notenstufe besser als die Klasse 7b ab.

Die Klasse 7a ist eine sehr leistungsstarke Klasse, die sich für alles interessiert und sich von Stunde zu Stunde vorbereitet. Dies spiegelt sich auch in der Notenverteilung wieder.

Das ist bei der Klasse 7b nicht der Fall.

Die Informatik wir nur mit einem Drittel für die Natur und Technik-Note gewichtet. Das ist ein Grund dafür, wieso einige Schüler die Informatik als unwichtig erachten und sich nicht von Stunde zu Stunde vorbereiten.

Die Jungen schneiden deutlich besser ab als die Mädchen.

5.5.3 Vergleich zwischen den Schulen

Das Dientzenhofer-Gymnasium hat ungefähr 1500 Schüler und eine vorwiegend naturwissenschaftlich-technologische Ausrichtung. Das Kaiser-Heinrich-Gymnasium hat über 700 Schüler und bietet, wenn man von den Ganztagsklassen absieht, nur eine sprachliche Ausbildungsrichtung an.

Zwar spielt in der 7. Jahrgangsstufe die Ausbildungsrichtung keine Rolle, jedoch kann man davon ausgehen, dass die Schüler, die mehr sprachlich veranlagt sind, nicht das DG, sondern eher das KHG besuchen.

Wenn man die Leistungsnachweise beider Schulen miteinander vergleicht, sieht man - abgesehen von der Klasse 7a am KHG, die heraussticht - keine

großen Unterschiede.

Im Unterricht war jedoch ein gravierender Unterschied. Die Schüler am DG haben viel lieber etwas Neues ausprobiert.

Ein großer Teil der Schüler am KHG hat jedoch mehrmals nachgefragt, wie das jetzt geht, anstatt auch selbst zu experimentieren. Das hat sich mit der Zeit etwas gelegt.

Die Gründe dafür können sehr vielschichtig sein. Es könnte an der Ausbildungsrichtung der Schule liegen oder aber auch an dem Unterricht, den die Schüler in der 6. Klasse hatten.

Zusammenfassend kann man sagen, dass die Informatik in Natur und Technik an beiden Schulen zu wenig gewichtet wird. Ein Großteil der Schüler hat eine falsche Einstellung dem Fach gegenüber und bereitet sich zu wenig von Stunde zu Stunde vor. Zudem spielt in diesem Alter auch noch die Pubertät eine wichtige Rolle.

6 Fazit

6.1 Schulfest am Dientzenhofer-Gymnasium Bamberg

Ich habe beim Schulfest des Dientzenhofer-Gymnasiums in Bamberg einen Programmpunkt "GCRATCH-Programmieren" angeboten. Acht Schüler der Klasse 7e haben sich bereit erklärt, die Besucher in GCRATCH einzuweisen.

Sobald jemand den Computerraum betreten hat, wurde dieser von einer Schülerin oder einem Schüler in Empfang genommen und in eingewiesen. Kurze Zeit später sah man viele Besucher, darunter Eltern - nicht nur der aktiven Schüler - und ältere Schüler, die schüler, die kannten, begeistert am Rechner sitzen und programmieren.

Obwohl der Multimediaraum im 2. Stockwerk ist und in den oberen Stockwerken kaum Programmpunkte angeboten wurden, wurde dieser Programmpunkt so gut angenommen, dass er anstatt der geplanten einen Stunde, sogar zwei Stunden durchgeführt wurde.

6.2 Ausblick

Oft hört man von Kollegen, dass sie mit der Algorithmik viel früher fertig sind und nicht mehr wissen, was sie noch tun sollen. An dieser Stelle bietet sich ein Projekt an.

Zum einen findet man unter²⁰ Anregungen für Projekte.

Zum anderen gibt es zu ein Sensor Board. Die erste Version hieß noch ScratchBoard und war in den USA für 25 Dollar erhältlich. Das Nachfolgermodell heißt jetzt PicoBoard und kostet das doppelte²¹.

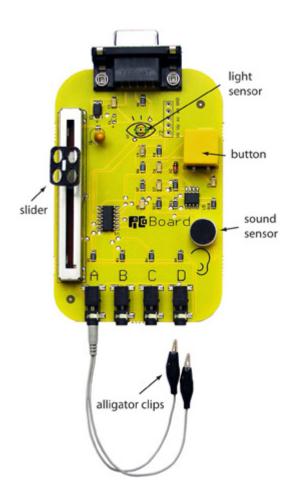


Abbildung 3: PicoBoard

²⁰[25, letzter Zugriff: 25.02.2009]

²¹[26, letzter Zugriff: 21.02.2009]

Mit dem Sensor Board hat man z. B. die Möglichkeit²²,

- ein Objekt nicht mit den Pfeiltasten, sondern mit dem "slider" zu steuern
- auf Helligkeitsänderungen zu reagieren ("light sensor")
- auf Lautstärkeänderungen zu reagieren ("sound sensor")
- durch das Drücken des "buttons" das Fallen der Bananen auszulösen, anstatt durch das Drücken der Leertaste
- mit "alligator clips" den Widerstand eines Stromkreises zu messen (z. B. an einem selbst gebauten "heißen Draht" anschließen)

In der Kategorie Fühlen sind noch weitere Bedingungen zu finden, die nur in Verbindung mit dem Sensor Board funktionieren:

Sensor Schaltfläche gedrückt▼?	Liefert wahr, wenn die Schaltfläche des ScratchBoards gedrückt wurde.
Sensor A verbunden ▼ ?	Liefert wahr, wenn etwas am Anschluss A (, B, C oder D) angeschlossen ist.

Den Treiber für das PicoBoard erhält man unter 23 Einsatzmöglichkeiten sind unter 24 zu finden.

6.3 Eigene Meinung

kann man sehr gut im Unterricht einsetzen. Es bietet viel mehr Möglichkeiten als Robot Karol. Die Programmierumgebung ist sehr übersichtlich aufgebaut, so dass sich vieles von selbst erklärt.

Den Schülern hat das Arbeiten mit sehr viel Spaß gemacht und sie waren motiviert bei der Sache.

Ich bin von überzeugt und werde in Zukunft die Algorithmik weiterhin mit GRATCH durchführen.

²²[27, letzter Zugriff: 21.02.2009]

²³[26, letzter Zugriff: 21.02.2009]

 $^{^{24}[11]}$

7 Anhang

Auf den folgenden Seiten sind die Arbeitsblätter und ihre Lösungen und die Stegreifaufgaben und ihre Lösungen zu finden.

Auf der beiliegenden CD ist folgendes enthalten:

- alle Arbeitsblätter und die Lösungen dazu als pdf- und docx-Dateien
- alle CRATCH-Projekte
 (Motivationen, Arbeitsaufträge und Stegreifaufgaben)
- alle geschriebenen Stegreifaufgaben und die Lösungen dazu
- ausgewählte SCRATCH-Projekte von Schülern
- dieses Dokument als PDF-Datei

Beschreibung von Abläufen

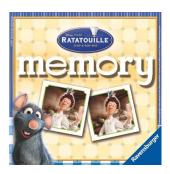
Abläufe werden durch	beschrieben.
Solche Handlungsanweisungen findet man in vielen Bereichen:	RATATOULLE
In der Informatik heißen solche Verarbeitungsvorschriften	·
Sie müssen u. a. folgenden Anforderungen genügen::	
Ein Algorithmus besteht aus endlich vielen Anweisungen. In a Algorithmus nach endlich vielen Schritten ein Resultat liefern z.B	
1. Mit jeder Anweisung ist auch die nächstfolgende festgelegt z.B	
2. Wird der Algorithmus mit den gleichen Voraussetzungen g Ausführung stets die gleichen Ergebnisse. z.B.	restartet, so liefert die
: Jede einzelne Anweisung muss für den Ausführenden des Algo eindeutig ausführbar sein. z.B	orithmus verständlich und

Beschreibung von Abläufen

Abläufe werden durch mehrere Anweisungen beschrieben.

Solche Handlungsanweisungen findet man in vielen Bereichen:

- Spielregeln,
- Gebrauchsanweisungen,
- Bastelanleitungen,
- Kochrezepten,
- mathematische Lösungsverfahren
- Beschreibung physikalischer Versuchsabläufe.



In der Informatik heißen solche Verarbeitungsvorschriften Algorithmen.

Sie müssen u. a. folgenden Anforderungen genügen:

Endlichkeit:

Ein Algorithmus besteht aus endlich vielen Anweisungen. In der Praxis soll ein Algorithmus nach endlich vielen Schritten ein Resultat liefern.

(z. B. Ein memory-Spiel endet, wenn alle Kärtchen aufgedeckt wurden und der Sieger feststeht.)

Eindeutigkeit:

- Mit jeder Anweisung ist auch die n\u00e4chstfolgende festgelegt.
 (z. B. Wer ein memory-Paar findet, ist noch einmal dran.)
- 2. Wird der Algorithmus mit den gleichen Voraussetzungen gestartet, so liefert die Ausführung stets die gleichen Ergebnisse.
 - (z. B. Wenn die Voraussetzungen immer gleich sind (Kärtchen liegen auf den gleichen Plätzen, gleiche Aktionen), dann gewinnt immer der gleiche Spieler.)

Ausführbarkeit:

Jede einzelne Anweisung muss für den Ausführenden des Algorithmus verständlich und eindeutig ausführbar sein.

(z. B. Jede Anweisung der memory-Spielanleitung ist so klar formuliert, dass die Spieler nicht lange überlegen müssen, wie sie richtig ausgeführt wird.)

Klassendiagramm:

Algorithmen mit Scratch

Die Objekte (Sprites) in Scratch bewegen sich auf einer Bühne:

SPRITE BÜHNE x-Position Hintergrund-Nr. y-Position nächster Hintergrund Ausrichtung setze ...-Effekt auf ... Größe < enthält ändere ...-Effekt auf ... aktuelles Kostüm drehe um ... Grad setze x auf ... ändere x um ... Damit ein Objekt eine ___ ____, muss man ihm eine Botschaft schicken, man sagt auch: die ___ Mit Methodenaufrufen gibt man dem Objekt Anweisungen. Programm aus Bausteinen in Scratch: age Hallo! für (2) Sek. ziehe Kostüm Kostüm2▼ an rte 1 Sek. gehe 100 -er Schritt e 1 Sek. ziehe Kostüm Kostüm1▼ an piele Klang miau ganz Eine Folge nacheinander auszuführender Anweisungen heißt ____

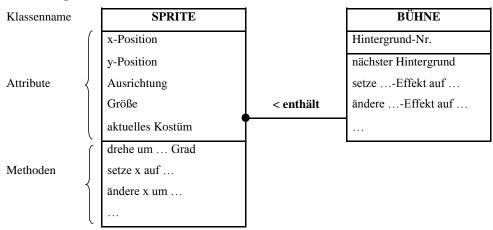
Das Programm Scratch kannst du unter http://scratch.mit.edu/ herunterladen.

100

Algorithmen mit Scratch

Die Objekte (Sprites) in Scratch bewegen sich auf einer Bühne:

Klassendiagramm:



Damit ein Objekt eine <u>Methode ausführt</u>, muss man ihm eine Botschaft schicken, man sagt auch: die <u>Methode aufrufen</u>.

Mit Methodenaufrufen gibt man dem Objekt Anweisungen.

Programm aus Bausteinen in Scratch:

Struktogramm:



sage "Hallo!" für 2 Sek.
ziehe Kostüm "Kostüm2" an
warte 1 Sek.
gehe 100-er Schritt
warte 1 Sek.
ziehe Kostüm "Kostüm1" an
spiele Klang "miau" ganz

Eine Folge nacheinander auszuführender Anweisungen heißt <u>Sequenz</u>.

Das Programm Scratch kannst du unter http://scratch.mit.edu/ herunterladen.

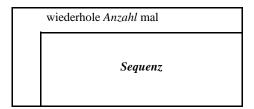
Wiederholung mit fester Anzahl

Soll eine Sequenz		
und steht		fest
verwendet man zur Vereinfachung die Wied	erholung mit fester Anzahl.	
Struktogramm für die Wiederholung mit fest	ter Anzahl:	
Wiederholungen können auch		sein.
Beispiel:		
Programm aus Bausteinen in Scratch:	Struktogramm:	
Wenn angeklickt Stift runter		
setze Stiftfarbe auf wiederhole 4 mal		
gehe 200 -er Schritt drehe ← 90 Grad		
warte 2 Sek.		
Stift hoch spiele Schlagzeug 72▼ für 2 Schläge		
wische Malspuren weg stoppe Programm		
Das Struktogramm ist eine		

Wiederholung mit fester Anzahl

Soll eine Sequenz mehrmals wiederholt werden und steht die Anzahl der Wiederholungen fest, verwendet man zur Vereinfachung die Wiederholung mit fester Anzahl.

Struktogramm für die Wiederholung mit fester Anzahl:



Wiederholungen können auch geschachtelt sein.

Beispiel:

Programm aus Bausteinen in Scratch:

Wenn angeklickt Stift runter setze Stiftfarbe auf wiederhole 4 mal gehe 200 -er Schritt drehe 90 Grad warte 2 Sek. Stift hoch spiele Schlagzeug 72 für 2 Schläge wische Malspuren weg stoppe Programm

Struktogramm:

Stift	Stift runter			
setze	e Stiftfarbe auf blau			
	wiederhole 4 mal			
	gehe 200-er Schritt			
	drehe nach rechts um 90 Grad			
	warte 2 Sek.			
Stift	Stift hoch			
spiel	spiele Schlagzeug 72 für 2 Schläge			
wisc	he Malspuren weg			

Das Struktogramm ist eine graphische Darstellung des Programmablaufs.

Bedingungen

Bedingungen sind	 .
die Fragen mit	beantworten.

Bei Scratch findest du in der "Fühlen"-Kategorie z. B. folgende Bausteine mit Bedingungen:

Maustaste gedrückt?	Liefert wahr, wenn die Maustaste gedrückt wird.
Taste Leertaste ▼ gedrückt?	Liefert wahr, wenn die Leertaste, eine
	Buchstabentaste, eine Zahlentaste oder eine
	Pfeiltaste gedrückt wird.
wird Mauszeiger▼ berührt?	Liefert wahr, wenn das Objekt den Mauszeiger
	berührt.
wird Rand▼ berührt?	Liefert wahr, wenn das Objekt den Rand der
	Scratch-Bühne berührt.
wird Objekt1 v berührt?	Liefert wahr, wenn das Objekt das Objekt
	"Objekt1" berührt.
wird Farbe berührt?	Liefert wahr, wenn das Objekt die angegebene
	Farbe berührt.
Farbe berührt ?	Liefert wahr, wenn eine bestimmte Farbe eines
	Objektes eine andere Farbe (eines anderen
	Objektes oder im Hintergrund) berührt.
laut?	Liefert wahr, wenn es laut ist.

Man kann die oben genannten Bedingungen mit dem nicht-Baustein in der "Zahlen-Kategorie" negieren.

z.B.

nicht Taste Leertaste ▼ gedrückt?	Liefert	wahr,	wenn	die	Leer	taste,	eine
	Buchstal	entaste,	eine	Zahler	ntaste	oder	eine
	Pfeiltaste	e nicht ge	drückt	wird.			

Bedingungen

Bedingungen sind Methoden, die Fragen mit wahr oder falsch beantworten.

Bei Scratch findest du in der "Fühlen"-Kategorie z. B. folgende Bausteine mit Bedingungen:

Maustaste gedrückt?	Liefert wahr, wenn die Maustaste gedrückt wird.
Taste Leertaste ▼ gedrückt?	Liefert wahr, wenn die Leertaste, eine
	Buchstabentaste, eine Zahlentaste oder eine
	Pfeiltaste gedrückt wird.
wird Mauszeiger berührt?	Liefert wahr, wenn das Objekt den Mauszeiger
	berührt.
wird Rand▼ berührt?	Liefert wahr, wenn das Objekt den Rand der
	Scratch-Bühne berührt.
wird ○bjekt1▼ berührt?	Liefert wahr, wenn das Objekt das Objekt
	"Objekt1" berührt.
wird Farbe berührt?	Liefert wahr, wenn das Objekt die angegebene
	Farbe berührt.
Farbe berührt ?	Liefert wahr, wenn eine bestimmte Farbe eines
	Objektes eine andere Farbe (eines anderen
	Objektes oder im Hintergrund) berührt.
laut?	Liefert wahr, wenn es laut ist.

Man kann die oben genannten Bedingungen mit dem **nicht**-Baustein in der "Zahlen-Kategorie" negieren.

z.B.

nicht Taste Leertaste ▼ gedrückt?	Liefer	t	wahr,	wenn	die	Leer	taste,	eine
	Buchs	tat	entaste,	eine	Zahler	ntaste	oder	eine
	Pfeilta	iste	e nicht ge	edrückt	wird.			

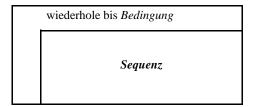
Bedingte Wiederholung

Soll das Objekt – z. B. die Katze – eine Seque	
	– also eine Bedingung den Wert
liefert –, so ver	wendet man die bedingte Wiederholung.
truktogramm für die bedingte Wiederholung	:
	1
eispiel:	
rogramm aus Bausteinen in Scratch:	Struktogramm:
Wenn 🏳 angeklickt	
wiederhole bis wird Farbe berührt?	
gehe 10 -er Schritt	
nächstes Kostüm	1
ziehe Kostüm cat1-a v an	
stoppe Programm	

Bedingte Wiederholung

Soll das Objekt – z. B. die Katze – eine Sequenz solange wiederholen, bis ein Ereignis eintritt – also eine Bedingung den Wert wahr liefert –, so verwendet man die bedingte Wiederholung.

Struktogramm für die bedingte Wiederholung:



Beispiel:

Programm aus Bausteinen in Scratch:

```
Wenn angeklickt
wiederhole bis wird Farbe berührt?
gehe (10) -er Schritt
nächstes Kostüm
ziehe Kostüm catl-a an
stoppe Programm
```

Struktogramm:

```
wdh. bis "wird Farbe 'grün' berührt?"

gehe 10-er Schritt

nächstes Kostüm

ziehe Kostüm "cat1-a" an
```

Zweiseitig bedingte Anweisung

Die zweiseitige bedingte Anweisung wird ver	wendet, wenn
Struktogramm für die zweiseitig bedingte Anv	weisung:
Beispiel:	
Programm aus Bausteinen in Scratch:	Struktogramm:
Wenn 🎮 angeklickt	
wiederhole fortlaufend	
wiederhole bis wird Rand berührt?	
gehe 10 -er Schritt	
falls wird Mauszeiger berührt?	
sage Du hast mich erwischt!	
sage Fang mich!	
drehe 🗘 (Zufallszahl von 1 bis 45) Grad	
pralle vom Rand ab	

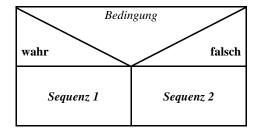
Wenn die Bedingung "wahr" ist, wird die _____ ausgeführt.

Ansonsten wird das Programm mit der _____ fortgesetzt.

Zweiseitig bedingte Anweisung

Die zweiseitige bedingte Anweisung wird verwendet, wenn abhängig von einer Bedingung unterschiedliche Sequenzen ausgeführt werden sollen.

Struktogramm für die zweiseitig bedingte Anweisung:



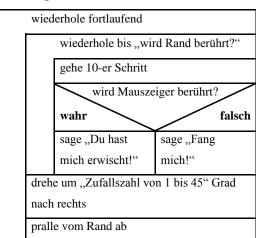
Beispiel:

Programm aus Bausteinen in Scratch:

ogramm aus Daustemen in Seruten.



Struktogramm:



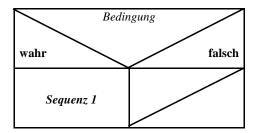
Wenn die Bedingung "wahr" ist, wird die Sequenz 1 ausgeführt.

Ansonsten wird das Programm mit der Sequenz 2 fortgesetzt.

Einseitig bedingte Anweisung

Die einseitig bedingte Anweisung wird verwendet, um eine Sequenz abhängig von einer Bedingung auszuführen:

Struktogramm für die einseitig bedingte Anweisung:



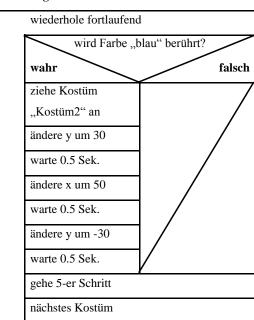
Beispiel:

Programm aus Bausteinen in Scratch:

Wenn angeklickt wiederhole fortlaufend falls wird Farbe berührt? ziehe Kostüm Kostüm2 → an ändere y um 30 warte 0.5 Sek. ändere x um 50 warte 0.5 Sek. ändere y um -30 warte 0.5 Sek.

nächstes Kostüm

${\bf Struktogramm:}$



Wiederholungen und bedingte Anweisungen werden als Kontrollstrukturen bezeichnet.

Methoden und Botschaften

Methoden

In Scratch wird eine Methode durch ein Skript beschrieben.

Das Ausführen einer Methode kann veranlasst werden durch:

- Start des Programms (Grüne Fahne)
- Tastendruck
- Anklicken des Objektes
- Erhalten einer bestimmten Botschaft



Dies hängt jeweils vom entsprechenden Startbaustein ab. Dabei können mehrere Methoden auch gleichzeitig durchgeführt werden.

Botschaften

Durch Botschaften können Objekte miteinander kommunizieren. So ist es z. B. möglich, dass ein Objekt anderen Objekten eine Botschaft schickt und sie auffordert, eine bestimmte Methode auszuführen.

Das Senden der Botschaften erfolgt mittels des Bausteins

Sende Botschafte an alle

Haben mehrere Objekte eine Methode mit dem gleichen Namen, dann werden durch das

Senden der Botschaft die Methoden all dieser Objekte aufgerufen.

Beispiel:



Struktogramm:

```
wiederhole bis "wird Junge berührt?"

gehe 2-er Schritt

sende "pieks" an alle und warte
```

Struktogramm:

sage "Autsch!" für 2 Sek.

Variablen

Variablen sind Platzhalter für Werte.

Sie werden z. B. bei einem Spiel als Zähler benötigt. Da sich ein Spielstand während des Spiels ändert – also variabel ist -, benutzt man Variablen, um den Spielstand zu speichern.

Erstellt man in der Kategorie Variable eine Variable, so muss man sich überlegen, welche

Objekte Zugriff auf die Variable benötigen.

Wenn nur ein Objekt die Variable benutzen soll, beschränkt man sich auf den Zugriff "Nur für dieses Objekt" (lokale Variable). Sollen alle Objekte die Variable verändern und benutzen können, verwendet man globale Variablen.



Initialisierung

Die Zählvariable muss bei jedem Programmstart von vorne anfangen zu zählen. Dafür muss sie **initialisiert** werden, d. h. auf einen Ursprungszustand gebracht werden.

Den Spielstand kannst du mit folgendem Baustein erhöhen:

```
ändere spielstand ▼ um 1
```

Rechnen

Mit Variablen kann man sehr gut rechnen. Scratch unterstützt die Grundrechenoperationen und einiges mehr. Das findest du in der Kategorie Zahlen.



Beispiel:

Programm des Objekts Münze:

Programm des Objektes Geldsack:

```
Wenn 🆰 angeklickt
                                                                                  etze spielstand▼ auf 0
Wenn 🎮 angeklickt
                                                 Geldsack spielstand 🔠
                                                                                Wenn ich münze▼ empfange
setze x auf (Zufallszahl von (-200) bis (200)
                                                                                 ndere spielstand v um 1
setze y auf Zufallszahl von 100 bis 200
                                                                                 alls spielstand > 5
 zeige dich
                                                                                 sage Spielende! für 2 Sek.
  ändere y um (-10)
    varte (0.2) Sek.
                                                                                 <mark>Venn Taste</mark> Pfeil nach links▼ <mark>gedrückt</mark>
                                                                                 ndere x um -10
   nde münze▼ an alle
                                                                                   nn Taste Pfeil nach rechts▼ gedrückt
                                                                                    re x um 10
```

7d	1. Stegreifaufgabe aus der Informatik 02	.07.2008
	Name: Note:	
1.	Meike hat folgendes Programm geschrieben.	
	a) Hilf ihr dieses zu vereinfachen und zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	(4 Punkte)
	Wenn angeklickt setze Stiftfarbe auf Stift runter warte 1 Sek. nächstes Kostüm gehe 10 -er Schritt drehe 15 Grad warte 1 Sek. nächstes Kostüm Stift hoch stoppe Programm	
	b) Welche Kontrollstruktur hast du an dieser Stelle verwendet?	(2 Punkte)
2.	Was ist der Unterschied zwischen der zweiseitig und der einseitig bedingten Anweisung?	(4 Punkte)

3. Hilf Jonas folgende Probleme zu lösen!

(4 Punkte) a) Ein Objekt Junge läuft geradeaus auf ein Haus zu. Es ist zu Hause angekommen, wenn es das Objekt Haus berührt. Das Programm soll immer - egal wie weit es vom Haus entfernt ist – funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)! b) Nun läuft das Objekt Junge einen anderen Weg. Anhand der gelben Markierung (6 Punkte) weiß es, dass es nach links abbiegen muss, um nach Hause zu finden! Es ist zu Hause angekommen, wenn es das Objekt Haus berührt. Das Programm soll immer - egal wie weit das Objekt Junge vom Haus entfernt ist funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!

Viel Erfolg!

7d

02.07.2008

Name:		Note:
-------	--	-------

1. Meike hat folgendes Programm geschrieben.

a) Hilf ihr dieses zu vereinfachen und zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!

(4 Punkte)



setze Stiftfarbe auf gelb

Stift runter

warte 1 Sek.

nächstes Kostüm

wiederhole 6 mal

gehe 10-er Schritt

drehe dich um 15 Grad nach rechts

warte 1 Sek.

nächstes Kostüm

Stift hoch

b) Welche Kontrollstruktur hast du an dieser Stelle verwendet?
Wiederholung mit fester Anzahl

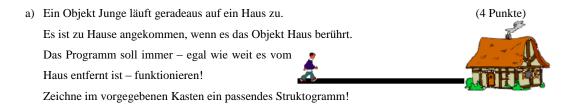
(2 Punkte)

2. Was ist der Unterschied zwischen der zweiseitig und der einseitig bedingten Anweisung? (4 Punkte)

Bei der zweiseitig bedingten Anweisung werden abhängig von einer Bedingung unterschiedliche Sequenzen ausgeführt.

Bei der einseitig bedingten Anweisung wird die Sequenz nur dann ausgeführt, wenn die Bedingung wahr ist.

3. Hilf Jonas folgende Probleme zu lösen!

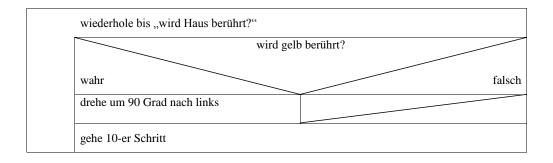


wiederhole bis "wird Haus berührt?"

gehe 10-er Schritt

b) Nun läuft das Objekt Junge einen anderen Weg. Anhand der gelben Markierung
weiß es, dass es nach links abbiegen muss, um nach Hause zu finden!
Es ist zu Hause angekommen, wenn es das Objekt Haus berührt.

Das Programm soll immer – egal wie weit das Objekt Junge vom Haus entfernt ist –
funktionieren!
Zeichne im vorgegebenen Kasten ein passendes Struktogramm!



Viel Erfolg!

7e	1. Stegreifaufgabe aus der Informatik	11.07.2008
	Name: Note:	
Hil	f Meike und Jonas folgende Probleme zu lösen!	
1.	Ein Objekt Ball bewegt sich fortlaufend in 10er Schritten. Berührt es den Rand, so prallt es von diesem ab und dreht sich um einen zufälligen Winkel von maximal 45 Grad (verwende folgenden Baustein: Zufallszahl von 1 bis 45). Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	(6 Punkte)
2.	Ein Objekt Regentropfen fällt solange senkrecht herunter, bis es das Objekt Smiley berührt, und versteckt sich danach.	(4 Punkte)
	Das Programm soll immer – egal von welcher Höhe der Regentropfen fällt – funktionierer Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	n!
		46

Regenschirm dabei. Sobald das Objekt Regentropfen das Objekt Smiley berührt, benachrichtigt es den Smiley. Daraufhin packt das Objekt Smiley den Regenschirm aus, d. h. es zieht das Kostüm "SmileyMitRegenschirm" an, und denkt sich "Hoffentlich wird das Wetter morgen besser!".	
a) Wie kommunizieren die zwei Objekte miteinander?	(4 Punkte)
b) Welche Anweisungen musst du dazu am Ende des Struktogramms in Aufgabe 2 ergänzen?	(2 Punkte)
c) Zeichne für die Reaktion des Objekts Smiley im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	(2 Punkte)
Nenne zwei Kontrollstrukturen!	(2 Punkte)

3. Der Smiley mag überhaupt keinen Regen! Deswegen hat er immer einen

Viel Erfolg!

'e	1. Stegreifaufgabe aus der Informatik

Name:	Note:

Hilf Meike und Jonas folgende Probleme zu lösen!

 Ein Objekt Ball bewegt sich fortlaufend in 10er Schritten. Berührt es den Rand, so prallt es von diesem ab und dreht sich um einen zufälligen Winkel von maximal 45 Grad (verwende folgenden Baustein: Zufallszahl von 1 bis 45).
 Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!



11.07.2008

	wird Rand bei	ührt?	
wahr			
pralle vom Rand ab			
drehe nach rechts/links			
um "Zufallszahl von 1 bis 45" Grad			

2. Ein Objekt Regentropfen fällt solange senkrecht herunter, bis es das Objekt Smiley berührt, und versteckt sich danach.

(4 Punkte)



Das Programm soll immer – egal von welcher Höhe der Regentropfen fällt – funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!

	wiederhole bis "wird Smiley berührt?"
	ändere y um -5-Schritt
ver	stecke dich



3. Der Smiley mag überhaupt keinen Regen! Deswegen hat er immer einen Regenschirm dabei. Sobald das Objekt Regentropfen das Objekt Smiley berührt, benachrichtigt es den Smiley. Daraufhin packt das Objekt Smiley den Regenschirm aus, d. h. es zieht das Kostüm "SmileyMitRegenschirm" an, und denkt sich "Hoffentlich wird das Wetter morgen besser!".



a) Wie kommunizieren die zwei Objekte miteinander? (4 Punkte)

Das Objekt Regentropfen schickt dem Objekt Smiley eine Botschaft, eine Methode auszuführen, die den Smiley das Kostüm wechseln lässt.

b) Welche Anweisungen musst du dazu am Ende des Struktogramms in Aufgabe 2 (2 Punkte) ergänzen?

sende "schirm" an alle (und warte)

c) Zeichne für die Reaktion des Objekts Smiley im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!

(2 Punkte)

schirm:

ziehe Kostüm "SmileyMitRegenschirm" an

denke "Hoffentlich wird das Wetter morgen besser!"

4. Nenne zwei Kontrollstrukturen!

(2 Punkte)

Wiederholung mit fester Anzahl, Bedingte Wiederholung, Einseitig bedingte Anweisung, Zweiseitig bedingte Anweisung

Viel Erfolg!

7a 1. Stegreifaufgabe aus der Informatik (NuT) 11.12.2008

Name: N	fote:
1. Was ist eine Sequenz?	(2 Punkte)
. Meike hat folgendes Programm geschrieben.	
a) Hilf ihr dieses zu vereinfachen und	(4 Punkte)
zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)	!
Wenn angeklickt setze Stiftfarbe auf Stift runter warte 1 Sek. nächstes Kostüm	
gehe 10 -er Schritt drehe (*) 15 Grad gehe 10 -er Schritt drehe (*) 15 Grad	
gehe 10 -er Schritt drehe & 15 Grad gehe 10 -er Schritt drehe & 15 Grad	
gehe 10 -er Schritt drehe (*) 15 Grad gehe 10 -er Schritt drehe (*) 15 Grad	
warte 1 Sek. nächstes Kostüm Stift hoch stoppe Programm	
b) Welche Kontrollstruktur hast du an dieser Stelle verwendet?	(2 Punkte
. Wann verwendet man die zweiseitig bedingte Anweisung?	(2 Punkte

4. Hilf Jonas folgende Probleme zu lösen!

(4 Punkte) a) Ein Objekt Junge läuft geradeaus auf ein Haus zu. Es ist zu Hause angekommen, wenn es das Objekt Haus berührt. Das Programm soll immer - egal wie weit es vom Haus entfernt ist – funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)! b) Nun läuft das Objekt Junge einen anderen Weg. Anhand der gelben Markierung (6 Punkte) weiß es, dass es nach links abbiegen muss, um nach Hause zu finden! Es ist zu Hause angekommen, wenn es das Objekt Haus berührt. Das Programm soll immer - egal wie weit das Objekt Junge von der gelben Markierung und die gelbe Markierung vom Haus entfernt ist – funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)! gelb

Viel Erfolg!

7a 1. Stegreifaufgabe aus der Informatik (NuT) 11.12.2008

	Name:	Note:	
1.	Was ist eine Sequenz?		(2 Punkte)
	Eine Folge nacheinander a	uszuführender Anweisungen.	
2.	Meike hat folgendes Progra	mm geschrieben.	
	a) Hilf ihr dieses zu verein	fachen und	(4 Punkte)
	zeichne im vorgegebene gelb	en Kasten ein passendes Struktogramm (mit Lineal)!	, ,
	Wenn P angeklickt	setze Stiftfarbe auf gelb	
	setze Stiftfarbe auf	Stift runter	
	Stift runter warte 1 Sek.	warte 1 Sek.	
	nächstes Kostüm gehe 10 -er Schritt	nächstes Kostüm	
	drehe (→ 15 Grad gehe 10 -er Schritt	wiederhole 6 mal	
	drehe (* 15 Grad gehe 10 -er Schritt	gehe 10-er Schritt	
	drehe (* 15 Grad gehe 10 -er Schritt	drehe dich um 15 Grad nach rechts	
	drehe (* 15 Grad gehe 10 -er Schritt	warte 1 Sek.	
	drehe (→ 15 Grad gehe 10 -er Schritt	nächstes Kostüm	
	drehe 🗣 15 Grad	Stift hoch	
	warte 1 Sek.		
	stoppe Programm		
	b) Welche Kontrollstruktur	r hast du an dieser Stelle verwendet?	(2 Punkte)
	Wiederholung mit fester	r Anzahl	
3.	Wann verwendet man die z	weiseitig bedingte Anweisung?	(2 Punkte)
	Die zweiseitige bedingte Aı	nweisung wird verwendet, wenn abhängig von einer Bedingung	

unterschiedliche Sequenzen ausgeführt werden sollen.

4. Hilf Jonas folgende Probleme zu lösen!

Struktogramm (mit Lineal)!

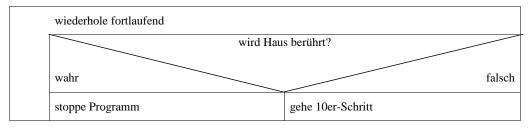
a) Ein Objekt Junge läuft geradeaus auf ein Haus zu. (4 Punkte)
Es ist zu Hause angekommen, wenn es das Objekt Haus berührt.

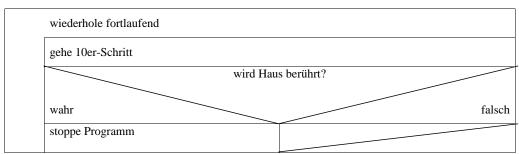
Das Programm soll immer – egal wie weit es vom Haus entfernt ist – funktionieren!

Zeichne im vorgegebenen Kasten ein passendes



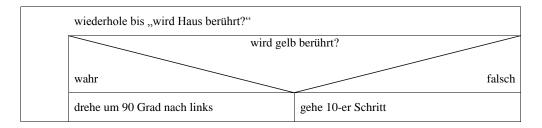
oder:





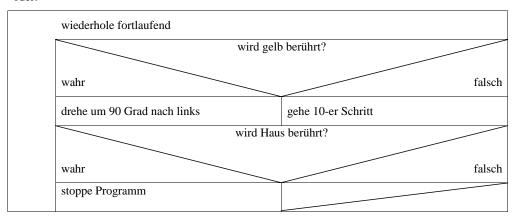
b) Nun läuft das Objekt Junge einen anderen Weg. Anhand der gelben Markierung
weiß es, dass es nach links abbiegen muss, um nach Hause zu finden!
Es ist zu Hause angekommen, wenn es das Objekt Haus berührt.

Das Programm soll immer – egal wie weit das Objekt Junge von der gelben
Markierung und die gelbe Markierung vom Haus entfernt ist – funktionieren!
Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!

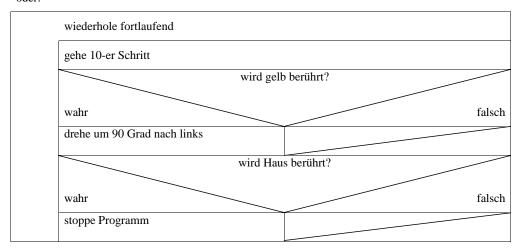


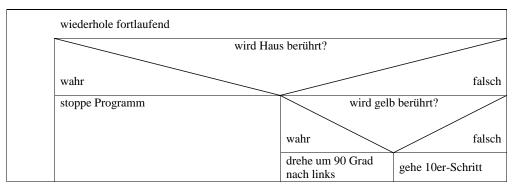
	wiederhole bis "gelb wird berührt?	
	gehe 10-er Schritt	
drehe um 90 Grad nach links		
	wiederhole bis "wird Haus berührt?"	
	gehe 10-er Schritt	

oder:



oder:





Viel Erfolg!

7b 1. Stegreifaufgabe aus der Informatik (NuT) 11.12.2008

Name: _______ Note:

1. Was ist eine Sequenz? (2 Punkte)

2. Meike hat folgendes Programm geschrieben.

a) Hilf Meike ihr Programm zu vereinfachen und zeichne im vorgegebenen Kasten ein (4 Punkte) passendes Struktogramm (mit Lineal)!

orange

wenn **Reagedickt**
wische Malspuren weg
estze Stiftfarbe auf

Wenn angeklickt
wische Malspuren weg
setze Stiftfarbe auf
Stift runter
gehe 50 -er Schritt
drehe 60 Grad
gehe 50 -er Schritt
stift hoch
stoppe Programm

b) Welche Kontrollstruktur hast du an dieser Stelle verwendet?

(2 Punkte)

I	Ein Objekt Elefant badet im Meer. Er ist inzwischen sauber und läuft zum Strand. Dort bleibt er stehen und sagt "Jetzt bin ich sauber!". Das Programm soll immer – egal wie weit der Elefant vom Stand entfernt ist – funktionieren! Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	(4 Punkte)
4.	Ein Objekt Pferd bewegt sich fortlaufend in 10er Schritten. Sobald es einen braunen Baumstamm sieht, springt es darüber. a) Wie kann man das Sehen realisieren, ohne dass das Objekt Pferd den braunen Baum	stamm bariibrt?
	wie kum man das senen reunsieren, omne dass das Gejekt i reid den bruthen Budin	(1 Punkt)
	b) Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!	(5 Punkte)
5.	Wann verwendet man die zweiseitig bedingte Anweisung?	(2 Punkte)

Viel Erfolg!

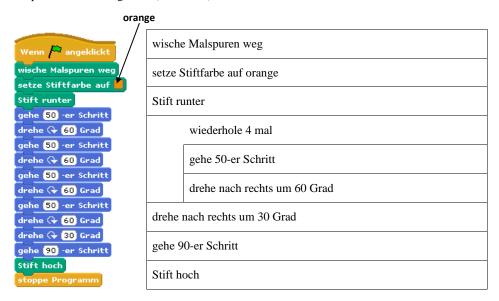
7b 1	. Stegreifaufgabe aus	der Informatik (NuT)	11.12.2008

Name: Note:

1. Was ist eine Sequenz? (2 Punkte)

Eine Folge nacheinander auszuführender Anweisungen.

- 2. Meike hat folgendes Programm geschrieben.
 - a) Hilf Meike ihr Programm zu vereinfachen und zeichne im vorgegebenen Kasten ein (4 Punkte) passendes Struktogramm (mit Lineal)!



b) Welche Kontrollstruktur hast du an dieser Stelle verwendet?

(2 Punkte)

Wiederholung mit fester Anzahl

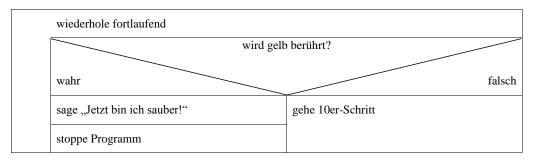
3. Ein Objekt Elefant badet im Meer. Er ist inzwischen sauber und läuft zum Strand.

Dort bleibt er stehen und sagt "Jetzt bin ich sauber!". Das Programm soll immer –
egal wie weit der Elefant vom Stand entfernt ist – funktionieren!

Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)!



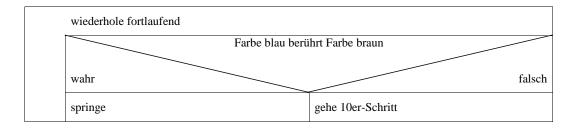
	wiederhole bis "wird Farbe gelb berührt?"
	gehe 10-er Schritt
sage "Jetzt bin ich sauber!"	



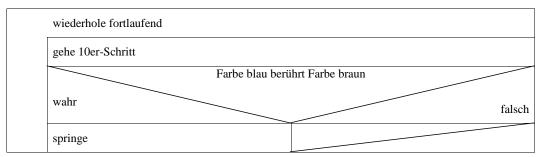
4. Ein Objekt Pferd bewegt sich fortlaufend in 10er Schritten. Sobald es einen braunen Baumstamm sieht, springt es darüber.



- a) Wie kann man das Sehen realisieren, ohne dass das Objekt Pferd den braunen Baumstamm berührt?
 - z. B. blauer Sensor vor dem Objekt (sollte jedoch einen gewissen Abstand haben) (1 Punkt)
- b) Zeichne im vorgegebenen Kasten ein passendes Struktogramm (mit Lineal)! (5 Punkte)



oder:



5. Wann verwendet man die zweiseitig bedingte Anweisung?

(2 Punkte)

Die zweiseitige bedingte Anweisung wird verwendet, wenn abhängig von einer Bedingung unterschiedliche Sequenzen ausgeführt werden sollen.

Viel Erfolg!

Literatur

- [1] Brichzin, P./Freiberger, U./Reinold, K. (2004). Ikarus. Natur und Technik. Schwerpunkt Informatik 6/7. Oldenbourg Schulbuchverlag
- [2] Engelmann, L. (2005). Natur und Technik. Schwerpunkt: Informatik. Bayern 6/7 Gymnasium. DUDEN PAETEC Schulbuchverlag. C. C. Buchner
- [3] Frey, E./Hubwieser, P./Winhard, F. (2004). Informatik 1. Objekte, Strukturen, Algorithmen. Ernst Klett Verlag
- [4] Hubwieser, P. (2001). Didaktik der Informatik. Grundlagen, Konzepte, Beispiele. Springer-Verlag
- [5] Schubert, S./Schwill, A.(2004). Didaktik der Informatik. Spektrum Akademischer Verlag
- [6] Humbert, L. (2005). Didaktik der Informatik mit praxiserprobten Unterrichtsmaterial. B. G. Teubner Verlag
- [7] Hartmann, W./Näf, M./Reichert, R. (2007). Informatikunterricht planen und durchführen. Springer-Verlag
- [8] Büdding, H./Homann, M. (2007). NXT-Roboter und mobile Endgeräte im Informatikunterricht. LOG IN. Informatische Bildung und Computer in der Schule. Nr. 145. LOG IN Verlag. Seite 44-51
- [9] Freudenberg, R./Hancl, M./Mietzsch, E. (2007). Es quiekt im Unterricht. LOG IN. Informatische Bildung und Computer in der Schule. Nr. 144. LOG IN Verlag. Seite 30-38
- [10] Seitz, F. (2008). Beschreibung von Abläufen durch Algorithmen unter Verwendung von Scratch im Fach Natur und Technik der 7. Jahrgangsstufe des G8. Erste schriftliche Hausarbeit für das Lehramt an Gymnasien in Bayern. Universität Bayreuth
- [11] Hofmann, M. (2008). Das Scratchboard. Einsatz des Scratchboards im Bereich "Beschreibung von Abläufen durch Algorithmen" im Fach Natur und Technik Schwerpunkt Informatik in Gymnasien. Erste schriftliche Hausarbeit für das Lehramt an Gymnasien in Bayern. Universität Bayreuth

Internetadressen:

- [12] Lehrplan Natur und Technik 7. Klasse http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/ index.php?StoryID=26436
- [13] Handreichung "Informatik im Fach Natur und Technik" http://www.isb.bayern.de/isb/index.asp?MNav=6&QNav=12&TNav=1&INav=0&Pub=427
- [14] Gesellschaft für Informatik (GI) e. V. (2008). Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. LOG IN. Informatische Bildung und Computer in der Schule. Nr. 150/151. LOG IN Verlag. Beilage http://www.sn.schule.de/~istandard/bildungsstandards_2008.pdf?PHPSESSID=c1146103b8373b9eeb1667a8fe1c3b11
- [15] Ehmann, M. (2006). Roboter zum Anfassen und virtuell. Problemlösendes Arbeiten im Informatikunterricht. Spektrum das Wissenschaftsmagazin der Universität Bayreuth, Ausgabe 02/2006. Seite 30-33

 http://www.uni-bayreuth.de/presse/spektrum/spektrum-pdf/ausgabe_02_06.pdf
- [16] Romeike, Ralf (2007). Animationen und Spiele gestalten. Ein kreativer Einstieg in die Programmierung. LOG IN. Informatische Bildung und Computer in der Schule. Nr. 146/147. LOG IN Verlag. Seite 36-44 (Gewinner des Unterrichtswettbewerbs der 12. GI-Fachtagung "Informatik und Schule INFOS 2007") http://www.funlearning.de/
- [17] Denker, M./Gälli, M. (2004). Von kleinen und großen Erfindern. Squeak: Lernumgebung und Smalltalk-System für Kinder und Erwachsene. c't Magazin für Computertechnik. Ausgabe 07/2004. Seite 216 ff. http://www.heise.de/ct/04/07/216/ (ohne Grafiken) http://www.squeak.de/ct-squeak-artikel.pdf
- [18] Homepage von Squeak http://www.squeak.org/ letzter Zugriff: 12.01.2009
- [19] Homepage von Scratch http://scratch.mit.edu/

- [20] Wikipedia-Artikel zu Scratch http://de.wikipedia.org/wiki/Scratch_(Programmiersprache)
- [22] Deutsche Übersetzung der Anleitung zu Scratch von Alex Oelinger http://www.olinger.net/iWeb/berufliches/Scratch_files/scratch_referenzhandbuch.pdf
- [23] Schlager, M. Abläufe und Algorithmen mit Scratch http://signalbscw.tcs.informatik.uni-muenchen.de/pub/bscw.cgi/d194654/ScratchInf7-Schlager.pdf
- [24] Güntner, T./Linse, P. Einführung in Scratch: Tutorial und Expertenpuzzle http://digitale-schule-bayern.de/dsdaten/348/96.pdf
- [25] Scratch-Werkstatt
 http://www.swisseduc.ch/informatik/programmiersprachen/
 scratch_werkstatt/
- [26] Homepage zum PicoBoard http://www.picocricket.com/picoboard.html
- [27] Getting Started with PicoBoard http://www.picocricket.com/pdfs/Getting_Started_With_ PicoBoards.pdf

Erklärung

Hiermit erkläre ich, dass ich die vorliegende schriftliche Hausarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Bayreuth, den 27. Februar 2009