HEINZ-BERGGRUEN-GYMNASIUM
BAYERNALLEE 4
BERLIN CHARLOTTENBURG

Unterrichtsentwurf

Informatik

Debora Baumann

13.04.2011

_____
(Debora Baumann)

Prüfungsvorsitzende:	Frau Pallik
Seminarleiter 1.SPS Charlottenburg-Wilmersdorf:	Herr Knittel
Seminarleiterin Mathematik:	Frau Glorius
Seminarleiter Informatik:	Herr Gramm
Schulleiter:	Herr Thimm
Lehrervertreter:	Herr Dietz

Lerngruppe:	Wahlpflichtkurs Informatik 9
Raum:	R022

Aufgabenstellung für diese Examensstunde:

Bereiten Sie Unterricht zum Thema „Ein Spiel mit Scratch programmieren“ vor und zeigen Sie einen Ausschnitt aus dem längerfristigen Lernprozess. Die Thematisierung, ggf. die Kontextualisierung der Lerninhalte und die Ausrichtung auf die Kompetenzentwicklung sind Teil der Aufgabenstellung.

Das Thema der Lehr- und Lernprozesse:

Thema der Einheit:	Ein Spiel mit Scratch programmieren
Thema der Stunde:	**Interaktion zwischen Objekten: Das Konzept der Botschaften in Scratch**

Der fachlich-inhaltliche Schwerpunkt:

In der objektorientierten Programmierung ist die Nachricht, auch Botschaft oder Message genannt, ein zentraler Begriff. Sie dient dem Informationsaustausch zwischen zwei oder mehreren Objekten. Ein Objekt *(der Sender)* schickt einem anderen Objekt *(dem Empfänger)* eine Nachricht, um dieses zu aktivieren. Die auszuführende Aktivität kann eine einfache Handlung oder eine Zustandsänderung sein. Die Interpretation der Nachricht nimmt das Empfängerobjekt in eigener Verantwortung vor, d. h., das sendende Objekt kennt grundsätzlich die realisierten Methoden des Empfängers nicht *(Geheimnisprinzip)*[1].
Für die Schülerinnen und Schüler[2] ist „die ‚Botschafts-Betrachtungsweise‘ erst einmal ungewohnt.“[3] Sie soll durch zwei Beispiele verdeutlicht werden:

1. Ein Hundebesitzer (Sender) gibt seinem Hund (Empfänger) bestimmte Befehle (Botschaften), wie „Hol den Ball!“, „Sitz!“, „Bei Fuß!“, die der Hund anschließend ausführt, sofern er gut erzogen ist. Dabei bestimmt der Hundehalter nicht über die Art und Weise, wie der Hund den Ball nehmen soll.
2. Ein Autofahrer (Empfänger), der sich an die Straßenverkehrsordnung hält, darf solange über Kreuzungen fahren, bis die Ampel (Sender) auf Rot schaltet und ihm dadurch die Botschaft zukommen lässt: „Anhalten und auf Grün warten!“.

Das Botschaften-Konzept ermöglicht somit eine interaktive Spielweise, die nicht notwendigerweise an einen Zeitfaktor gebunden ist, sondern eintritt, sobald der Empfänger eine bestimmte Botschaft erhält. Verschiedenste Objekte können also untereinander kommunizieren bzw. interagieren.

---

[1] Vgl. Claus/Schwill [2006]: Stichwort Nachricht, S.443.
[2] Im weiteren Verlauf werden die „Schülerinnen und Schüler“ mit SuS und „eine Schülerin oder ein Schüler“ mit SoS abgekürzt.
[3] Engelmann [2009]: S.61.

Dieses Konzept, bezogen auf das spätere Computerspiel, kann der Wechsel in ein neues Level sein. Der Sender (das vom Spieler benutzte Objekt) verschickt eine Botschaft, dass alle Punkte eingesammelt wurden, und der Empfänger (Bühne) erkennt, dass ein neues Bild erzeugt werden muss, um in das nächste Level zu überführen bzw. um weitere Spielfiguren (Gegenspieler) ins Leben zu rufen.
Das Konzept der Botschaften wird außerdem zum arbeitsteiligen Programmieren in der Softwareentwicklung verwendet.

Die zentrale Aufgabenstellung für die Stunde ist die Programmierung eines Bienenstaates, angeführt von einer Königsbiene, die ihren Arbeiterbienen verschiedene Botschaften zukommen lässt, auf die jene reagieren sollen.
Die SuS bekommen als Vorlage ein Scratch-Projekt. In diesem Projekt ist bereits der Hintergrund programmiert und es sind die Objekte „Arbeiterbiene“ und „Königsbiene“ vorhanden (ohne Programmierung).
Jene/r SoS mit dem Arbeitsauftrag die Königsbiene zu programmieren (Sender), benutzt die Arbeiterbiene lediglich, um die Programmierung zu testen und ggf. zu verbessern/zu verändern. Die Königsbiene wird so programmiert, dass sie mit Beginn des Programmes (Klick auf die grüne Fahne) in einer Schleife („wiederhole fortlaufend“) über dem Bienenstock hin und her fliegt. Anschließend reagiert die Königsbiene auf das Betätigen einer Taste („Wenn Taste <n, v, f, h oder i>[4] gedrückt“), indem sie eine Botschaft an alle Objekte sendet („sende <Botschaft> an alle“). Hierauf reagiert die vorgegebene Arbeiterbiene, die von den SuS zum Testen eingesetzt wird. Die Arbeiterbienen nehmen die Botschaft auf („Wenn ich <Botschaft> empfange“) und geben zu Testzwecken etwas in einer Sprechblase aus („sage …“). Die Königsbiene kann zusätzlich mit Sprech- oder Denkblasen in ihrer Programmierung ergänzt werden, so dass die Handlung auch sinnvoll und sichtbar unterstützt wird.
Die SuS mit dem anderen Arbeitsauftrag (Empfänger), programmieren eine Arbeiterbiene und benutzen die vorgegebene Königsbiene, um ihre Algorithmen auf Richtigkeit zu überprüfen. Sie lassen ihre Arbeiterbiene solange versteckt, bis sie die Botschaft der Königsbiene erhalten. Die Arbeiterbiene soll bei den verschiedenen Botschaften entsprechend reagieren, dies wird mit einzelnen Sequenzen realisiert. Das Fliegen soll zufallsgesteuert („gehe <Zufallszahl>“ und „pralle vom Rand ab“) und kontinuierlich (Schleife) passieren. Das Suchen von Honig ist ein zielgerichtetes Bewegen („gleite“) hin zu verschiedenen Blumen. Das Vorstellen der Arbeiterbienen kann durch den „sage“-Baustein verwirklicht werden. Wenn die Arbeiterbienen die Botschaft „nacht“ empfangen verdunkelt sich der Himmel, die Bienen fliegen zum Bienenstock zurück und verschwinden („verstecke“) im Bienenstock (siehe antizipiertes SuS-Ergebnis).

[4] Jede Taste steht für eine Botschaft: n = nacht; v = vorstellen; f = fliegen; h = honig oder i = improvisieren.

Die Standards des Rahmenlehrplans und ihre Konkretisierung:

Standards aus dem Rahmenlehrplan[5]	***Entwicklungsstand der Lerngruppe***	***Konkretisierung der Kompetenzentwicklung***
Die SuS … … entwickeln selbstständig algorithmische Lösungen. … erstellen kleine Programme unter Benutzung vorgegebener Bausteine.	**Deklarativ:** Die SuS kennen algorithmische Grundstrukturen (Wiederholungen, Verzweigungen) und fast alle vorgegebenen Bausteine in Scratch. Sie kennen die Interaktion von Objekten nicht. Bisher wurde das Inerscheinungtreten von unterschiedlichen Objekten mit dem Zeitfaktor („warte bis“ oder „warte 20 sec.“) gelöst. **Prozedural:** Die SuS erstellen selbstständig einfache Programme und algorithmische Lösungen unter Benutzung der vorgegebenen Bausteine in Scratch. Bisher haben sie nur die Steuerungselement „Wenn grüne Fahne angeklickt“, „Wenn <Taste> gedrückt“ oder „Wenn <Objekt> angeklickt“ zum Start eines Programmes bzw. zur Aktivierung eines Objektes verwendet.	Die SuS erstellen selbstständig eine algorithmische Lösung zur ihnen vorliegenden Aufgabe (Arbeiterbiene oder Königsbiene). Sie benutzten dazu den (vorgegebenen) neuen Baustein „Wenn ich <Botschaft> empfange“, um den Empfänger (Arbeiterbiene) zu programmieren und in Interaktion mit dem Sender (Königsbiene) treten zu lassen. Des Weiteren benutzen sie den neuen Baustein „sende <Botschaft> an alle“, damit der Sender in eine Interaktion mit allen Empfängern treten kann. Johannes P. programmiert selbstständig eine Arbeiterbiene, die auf Botschaften der Königsbiene reagiert. Dazu benutzt er den vorgegebenen Baustein „Wenn ich <Botschaft> empfange“ und erstellt jeweils passende Algorithmen, die auf der Grundlage der algorithmischen Grundstrukturen basieren und die vorgegebenen Teilaufgaben erfüllen. Yuka programmiert eine Königsbiene, die später Botschaften an Arbeiterbienen sendet. Dazu benutzt sie in ihren jeweiligen Algorithmen zur Bearbeitung der einzelnen Teilaufgaben den Baustein „sende <Botschaft> an alle“.

[5] RLP [2006]: S. 10f. und S. 25.

Die angestrebte, längerfristige Kompetenzentwicklung der Lernenden:

Stunde [Datum] in der Einheit		*Thema*	*Kompetenz RLP Sek I*[6]
1./2. [08.03.]	**Ein Spiel mit Scratch programmieren**	**Was ist ein gutes Computerspiel?** Herausarbeitung von Kriterien als Grundlage für die Entwicklung des eigenen Computerspiels	**Programmabläufe strukturieren und umsetzen**: - Die SuS planen Anweisungsabfolgen zur Realisierung von Operationen auf Modellobjekten und setzen diese in lauffähige Programme um.
3./4. [15.03.]		**Kriterium 1: Bewegung und Aktion von Objekten** Kennenlernen der Programmierumgebung Scratch, indem vordefinierte Bausteine (Malstift; Bewegung; Steuerung) genutzt werden, um ein Bild mit dem Objekt Stift malen zu lassen.	
5./6. [22.03.]		**Exkurs: Algorithmische Grundstrukturen** Das Konzept der algorithmischen Grundstrukturen (der Schleifen und bedingten Anweisungen) nutzen, damit der Programmcode übersichtlich und klein bleibt. Struktogramme als Unterstützung zum Vordenken von algorithmischen Grundstrukturen.	
7./8. [29.03.]		**Kriterium 2: Spiel-Atmosphäre durch gute Grafik und passende Hintergrundmusik** Wie entwerfe ich eigene Objekte (Spielfiguren) und eigene Musik, um sie anschließend in Scratch einzubinden?	
9./10. [12.04.]		**Kriterium 3: Zielformulierung durch Punktesammeln** Das Variablenkonzept als Hilfe für gesammelte Punkte oder verlorene Leben, damit das Spiel unterhaltsam, spannend und sinnvoll wird.	
11. Examen [13.04.]		**Kriterium 4: Interaktion von Objekten und verschiedene Spiellevel (Welten)** Interaktion von Objekten mit dem Konzept der Botschaften in Scratch erarbeiten.	
12./13. [03.05.]		**Das Storyboard** – Der erste Schritt zur Erstellung meines eigenen Spiels mit Scratch.	
14.-17. [10.05.; 31.05.]		**Mein Spiel mit Scratch programmieren**	

Im Anhang ist der Advance Organizer zu finden, den die SuS in der dritten Stunde erhalten haben, nachdem zuvor die Kriterien für ein gutes Spiel entwickelt wurden.

[6] RLP [2006]: S. 16.

Die individuelle Kompetenzentwicklung der Lernenden:

Der Wahlpflichtkurs Informatik 9 setzt sich aus zehn SuS aus den Klassen 9c und 9d zusammen. Das Arbeiten im Kurs, egal ob nebeneinander (in Einzelarbeit) oder miteinander (in Gruppenarbeit), ist stets geprägt durch einen freundlichen, kameradschaftlichen und hilfsbereiten Umgang, obwohl die SuS aus zwei unterschiedlichen Klassen kommen. Vor allem Kara bemüht sich in den Arbeitsphasen ihren Nachbarinnen (Anni und Winona) bei der Erarbeitung der Aufgabe behilflich zu sein.

Johannes P. hat bereits im letzten Jahr den Wahlpflichtkurs Informatik absolviert. Er arbeitet überwiegend selbstständig und zielstrebig. Johannes hat ADHS, was jedoch in der kleinen Gruppe nicht so sehr auffällt. Lediglich in langen Erarbeitungsphasen beschäftigt er sich gelegentlich mit anderen Sachen, findet aber nach einer kleinen Pause und einer Ermahnung meinerseits stets zur Aufgabe zurück.
Er kennt alle vorgegebenen Bausteine und ihre Nutzung und konnte die bereits gestellten Aufgaben auf seine Weise zügig bearbeiten. Johannes kommt mit seiner Programmierung stets zum Ziel, muss jedoch erkennen, dass sie sehr aufwändig ist, weil er größtenteils intuitiv etwas programmiert und nur bedingt algorithmische Grundstrukturen benutzt. Er möchte möglichst schnell zum Ziel gelangen, weshalb er vorschnell gleiche Objekte dupliziert, obwohl sie noch nicht fertig programmiert sind.
Johannes ist schnell frustriert, wenn er die Aufgabe nicht beim ersten Ansatz lösen kann. Dennoch habe ich es nur selten erlebt, dass er aufgibt. Er versucht sich zu einer möglichst guten Lösung durchzuarbeiten und gibt sich dann mit ein paar Fehlern zufrieden.
Gerade bei dem Botschaften-Konzept und in der Arbeitsteilung muss Johannes mit voller Konzentration arbeiten, damit er die Schnittstellen gut programmiert und seine Arbeiterbiene, den Botschaften der Königsbiene folgen kann.

Yuka ist eine sehr introvertierte Schülerin. Sie arbeitet stets konzentriert und zuverlässig. Im Umgang mit der Programmierumgebung Scratch zeigte sie von Anfang an große Begeisterung, weil sie Scratch bereits zu Hause auf ihrem privaten Computer installiert hat. Sie hat am Anfang bei den ersten Programmieraufgaben viel Hilfe benötigt, um in die Denkweise, vor allem der algorithmischen Grundstrukturen, hineinzukommen und die verschiedenen Objekte zu programmieren.
Um ihr die benötigte Zeit zum Programmieren zu lassen, bekommt sie den Arbeitsauftrag eine Königsbiene zu programmieren. Sie kann sich auf die einzelnen Aufgaben konzentrieren, selbstständig einen Algorithmus zur Bearbeitung schreiben und anschließend noch Erweiterungen ihrer Königsbiene vornehmen. Zugleich wird ihr Selbstvertrauen gestärkt, weil die Königsbiene der Sender und somit Impulsgeber für den Kurs ist.

Die Begründung der Lehr- und Lernstruktur:

Die Stunde beginnt mit der Analyse des Scratch-Projektes „Ampel“[7], so dass die SuS erkennen, wie das Senden und Empfangen von Botschaften in Scratch realisiert wird. Damit die SuS in kurzer Zeit das Wesentliche des Botschaften-Konzepts verstehen, ist die Ampel auf zwei Zustände reduziert worden.

[7] Vgl. Mickan, J. [2010]: S. 13.

Neu sind die Bausteine „sende <Botschaft> an alle (und warte)“ bzw. „Wenn ich <Botschaft> empfange“, welche in der Programmierung der Ampel bzw. des Autos erkennbar und für die SuS leicht zu analysieren sind. Dies zu besprechen und zugleich am Whiteboard festzuhalten, schließt die Einstiegsphase ab und führt direkt zur zentralen Aufgabenstellung.

In der Bearbeitungsphase erstellen die SuS mit Hilfe eines Arbeitsblattes (AB 1 bzw. 2) eine Biene in Scratch, die Botschaften sendet bzw. empfängt. Die SuS arbeiten zunächst allein, greifen ggf. auf eigene, selbst erstellte Programme aus ihrem Ordner zurück, um evtl. nach Lösungsstrategien zu suchen und diese auf die neue Aufgabenstellung zu übertragen. Zusätzlich steht den SuS, die eine erweiterte Erklärung zum Botschaften-Konzept benötigen, ein Übersichtsblatt „Botschaften in Scratch“ im Ordner austeilen-p_wpfinformatik9 zur Verfügung.
Die Struktur des Projekts erfolgt in zwei Parteien: Königsbiene (zwei SuS) und Arbeiterbiene (acht SuS).
Je eine Königsbiene übernehmen Yuka und Johannes G.[8], die im Kurs eher introvertiert sind. Beide sollten mit dieser Aufgabe weder unter- noch überfordert sein und die Anforderungen gut erfüllen, aus folgenden Gründen: Durch den Stellenwert der Königsbiene ist ihre Aufgabe sehr hoch einzuordnen und es hängt ein gewisser Leistungsdruck an beiden. Da am Ende aber nur eine Königsbiene verwendet werden kann, relativiert sich der Druck zugleich und sie haben entsprechend viel Zeit. Eine Differenzierung erlaubt ihnen kreativ ihre Biene zu programmieren.
Die anderen SuS erstellen je eine Arbeiterbiene, die auf die Botschaften individuell reagiert (je nach Programmierung der SoS). Im Sinne der Differenzierung ist es bei diesem Arbeitsauftrag nicht von Bedeutung, welche Anzahl von Botschaften von den SoS programmiert wurde. Ich bin optimistisch, dass alle eine Reaktion auf die Botschaften „fliegen“, „honig“ und „vorstellen“ am Ende der Arbeitszeit programmiert haben. Die leistungsstärkeren SuS (Kara, Claudius, Johannes P.) werden alle vorgegebenen Botschaften in Aktionen umsetzen, zusätzlich die Improvisationsaufgabe bearbeiten und diese individuell lösen.
Durch das selbstständige Lösen der Aufgabe sind am Ende der Arbeitszeit zehn individuelle Bienen zu erwarten, die auf die geforderten Botschaften reagieren, wenn eine fehlerfreie Programmierung vorliegt.

Nachdem alle SuS ihre Bienen im Tausch-Ordner gespeichert haben, werde ich am Ende der Arbeitsphase das Projekt zusammensetzten, damit die SuS Zeit zur Reflexion haben. Die Zusammensetzung erfolgt mit Hilfe der Scratch-Vorlage „Bienenstaat-Leer“, in welche die bearbeiteten Objekte „Königsbiene“ und alle „Arbeiterbienen“ importiert werden. Das Gelingen und Betrachten des Projekts hängt vor allem von der richtig programmierten Königsbiene ab, die in einem zentralen Projekt schnell durch die zweite Königsbiene ersetzt werden könnte. Weil die Projektzusammensetzung eine kurze Zeit in Anspruch nimmt, fordere ich die SuS auf, ihre möglichen Probleme und Fehlerquellen in der Programmierung zu reflektieren. Diese werden anschließend als Hinweis im Whiteboardbild festgehalten.
Im nächsten Schritt werden wir uns das Ergebnis des Bienenstaats ansehen und ggf. über Verbesserungen und Fehlerquellen diskutieren.
Falls noch Zeit ist, besteht die Möglichkeit, das Projekt erneut mit der zweiten Königsbiene zu testen und auf Unterschiede zu achten!

[8] Falls die genannten SuS nicht anwesend sind, werden sie durch Anni oder Tobias ersetzt.

Die Konkretisierung der geplanten Lehr- und Lernprozesse:

Legende: gUG – gelenktes Unterrichtsgespräch; EA – Einzelarbeit; UF – Unterrichtsform, AO – Advance Organizer

Zeit	Erwartetes SuS-Verhalten	Geplantes Lehrerverhalten	UF/Medien
gUG: Motivierung der SuS und Einführung in das Thema (5-10')			
9.55	SuS beschreiben das Programm: Eine Ampel steht auf Rot und schaltet auf Grün, sobald sie angeklickt wird. Daraufhin fährt das Auto los. Nach kurzer Zeit schaltet die Ampel auf Rot und das Auto stoppt, wie von alleine. Sie erkennen, dass die Ampel der Sender und das Auto der Empfänger ist und welche „Bausteine“ hierfür in Scratch verwendet werden. Das Ampelsignal (Rot oder Grün) ist die Botschaft, die in den Bausteinen integriert ist.	Warten bis alle am mittleren Tisch sitzen. Begrüßung! „Betrachtet zunächst das geöffnete Programm. [zweimal abspielen lassen] Beschreibt, was ihr gesehen habt! [Programmierung aller Objekte einblenden lassen] Betrachtet die Programmierung: Beschreibt, was neu für euch ist und welche Auswirkungen dies im gezeigten Programm hat!“ Heute geht es um das Botschaften-Konzept. Es wird für eine Interaktion zwischen Objekten verwendet. Ihr sollt, bezogen auf euer späteres Spiel, die Interaktion von Objekten verbessern und von der Zeitvariable unabhängig machen. Außerdem ist das Botschaften-Konzept für das Erzeugen von verschiedenen Spielleveln sehr hilfreich.“ [Whiteboardbild entwickeln] -> AB austeilen	Beamer, Scratch-Projekt Ampel
EA: Bearbeitung des Arbeitsauftrags (20-25')			
10.05	SuS bearbeiten ihren Arbeitsauftrag! Yuka und Johannes G. bearbeiten Königsbienen. Alle anderen programmieren eine Arbeiterbiene.	„Heute sollt ihr Botschaften selbst programmieren. Dazu habe ich das Beispiel eines Bienenstocks gewählt. Es gibt zwei Königsbienen und acht Arbeiterbienen. Lest euch genau den Arbeitsauftrag durch und beachtet die Bedingungen, die erfüllt werden müssen. Ihr habt 20 Minuten Zeit: Arbeitsende ist 10:25 Uhr, dann sollt ihr euer Objekt im Tausch-Ordner abspeichern!“	PC, AB 1 bzw. AB 2 AB 3 im Austeilen-Ordner
EA u. gUG: Sicherung (10')			
10.30	Die SuS überlegen, welche Schwierigkeiten sie bei der Erarbeitung festgestellt und bei der Zusammensetzung und	„Beendet bitte eure Arbeit! Wenn ihr jetzt alle eure Biene im Tausch-Ordner gespeichert habt, dann setzt euch bitte an die mittleren Tische zurück. Reflektiert, welche Schwierigkeiten ihr bereits beim Erstellen (und Testen) der	Beamer; AO der SuS

Ende: *10.40*	Durchführung des Programms erwarten könnten. SuS äußern die aufgetretenen und erwarteten Schwierigkeiten: - Alle Botschaften müssen richtig geschrieben sein Betrachten das entstandene Projekt und äußern evtl. Verbesserungsvorschläge für das nächste Mal. Notieren sich in ihrem AO das Stichwort „Botschaften“. SuS reflektieren das Ergebnis der Teamarbeit.	Sender bzw. Empfänger festgestellt habt.“ [setzt einige Bienen (aber nur eine Königsbiene) in das Projekt Bienenstaat_leer ein] „Nennt Schwierigkeiten, die euch aufgefallen sind! Wir wollen es als Aufforderung zur Achtsamkeit aufschreiben! Betrachten wir das entstandene Projekt.“ [ggf. wird die Königsbiene durch die zweite Königsbiene ersetzt und erneut getestet] „Holt euren AO heraus und ergänzt das neue Konzept, welches euch bei der Interaktion von Objekten und bei der Realisierung von neuen und weiteren Spielleveln helfen kann!“ (**Hinweis**: „Notiert euch bis zur nächsten Stunde ein erstes Konzept für euer eigenes Spiel! Ihr dürft im Team oder alleine ein Spiel programmieren!“)	
***Puffer: Vorarbeit für das eigene Spiel**			
	Notieren sich in ihrem Hefter erste Spielideen!	„Es ist noch etwas Zeit: Macht euch erste Notizen für euer eigenes Spiel. Hinweise zum Überlegen: Arbeite ich alleine? Was ist die Hauptspielfigur? Was ist das Ziel des Spiels? Wie sieht das Spielfeld aus? …“	

Quellen:

- Prof. Dr. Claus, V. / Prof. Dr. Schwill, A. [2006]: Duden Informatik A-Z; Dudenverlag; Mannheim.
- Dr. Engelmann, L. [2009]: Duden Informatik Lehrbuch S II; Duden Paetec GmbH, Berlin.
- Mickan, J. [2010]: Unterrichtsentwurf im Fach Informatik; Berlin.
- Senatsverwaltung für Bildung, Jugend und Sport [2006]: Rahmenlehrplan für die Sekundarstufe I, ITG und Informatik Wahlpflichtfach; Berlin.

Anlagen:

1. Antizipiertes Whiteboardbild:

Interaktion von Objekten: Das Konzept der Botschaften in Scratch

Ampel – Rotsignal – Autofahrer

Sender – Botschaft – Empfänger

ACHTUNG:

- Botschaften müssen stets gleich geschrieben werden
  - Groß- und Kleinschreibung beachten
  - Keine Rechtschreibfehler

2. Sitzplan:

	Whiteboard	
Chris		Tobias
Yuka		Claudius
Anni		Johannes P.
Kara		Luis
Johannes G.		
Winona		

3. Arbeitsblätter:

3.1. Arbeitsauftrag (Sender):

WPF [Ba]	Botschaften in Scratch	

Arbeitsauftrag (Sender):

Programmiere eine Königsbiene als Sender, die den Arbeiterbienen eine Botschaft sendet, wenn die entsprechende Taste auf der Tastatur gedrückt wurde.

Vorbereitung:

- Öffne die Scratchvorlage Bienenstock_Vorlage aus deinem Ordner ***tausch-p_wpfinformatik9***!
- Speicher das Dokument in deinem Informatik-Ordner (Laufwerk H:/)!

Folgende **Bedingungen** sollen erfüllt sein:

1. Benenne die Königsbiene mit deinem Namen (z.B.: Königin_Clara) und lasse sie über dem Bienenstock immer hin und her fliegen.

2. Benutze die ***Arbeiterbiene***. Sie soll später die Botschaften empfangen und als Reaktion die Botschaften als Text in Sprechblasen ausgeben. So kannst du überprüfen, ob sich Fehler eingeschlichen haben.

3. Die ***Sender***-Königsbiene soll bei folgenden Eingaben (Tastendruck) die gegebenen Botschaften an alle anderen Arbeitsbienen (Objekte) schicken:

Eingabe	Botschaft
Taste „f“	fliegen
Taste „h“	honig
Taste „n“	nacht
Taste „v“	vorstellen
Taste „i“	improvisieren

<u>**Zusatz**</u>: Lass deine Königsbiene jeden sinnvollen Tastendruck mit einer kurzen Botschaft (Sprechblase) kommentieren. Ergänze das Verhalten der Königsbiene durch weitere sinnvolle Aktionen bei den verschiedenen Eingaben.

Wichtig am Ende:

Öffne die Programmierung deiner Königsbiene und gehe auf „***Datei/Objekt exportieren***“.
Speicher dein Objekt im Ordner ***tausch-p_wpfinformatik9*** ab! Achte darauf, dass deine Biene auch deinen Namen trägt.

3.2. Arbeitsauftrag (Empfänger):

WPF [Ba]	Botschaften in Scratch	

Arbeitsauftrag (Empfänger):

Programmiere eine Arbeiterbiene als Empfänger, die nur auf die Botschaft der Königin reagiert.

Vorbereitung:

- Öffne die Scratchvorlage Bienenstock_Vorlage aus deinem Ordner ***tausch-p_wpfinformatik9***!
- Speicher das Dokument in deinem Informatik-Ordner (Laufwerk H:/)!

Folgende **Bedingungen** sollen erfüllt sein:

1. Benenne die ***Arbeiterbiene*** mit deinem Namen und verändere das Aussehen (z.B. die Flügel einfärben), aber nicht die Größe, so dass du später sofort deine Biene erkennst.

2. Benutze die ***Königsbiene***, so dass sie später zu Testzwecken die Botschaften sendet. So kannst du überprüfen, ob sich Fehler eingeschlichen haben.

3. Die ***Arbeiterbiene*** soll bei folgenden Botschaften die angegebene Reaktion ausführen:

Botschaft	Reaktion
fliegen	zufälliges Umherfliegen
honig	fliege mehrere Blumen an
nacht	zurückfliegen zum Bienenstock, im Bienenstock verschwinden und das Programm beenden
vorstellen	Kurzen Text ausgeben, indem sich der Sprite vorstellt

<u>**Zusatz**</u>: Lass deine Arbeiterbiene auf die Botschaft „improvisieren“ reagieren. Denke dir selbst aus, wie dein Objekt reagieren soll.

Wichtig am Ende:

Öffne die Programmierung deiner Arbeiterbiene und gehe auf „Datei/Objekt exportieren“. Speicher dein Objekt im Ordner ***tausch-p_wpfinformatik9*** ab! Achte darauf, dass deine Biene auch deinen Namen trägt!

3.3. Infoblatt zu Botschaften im Ordner austeilen-p_wpfinformatik9:

WPF [Ba]	Botschaften in Scratch	

Ampel – Rotsignal – Autofahrer

Sender – Botschaft – Empfänger

Der Sender:

... hat zwei Methoden zur Auswahl, um Botschaften zu senden:

Sender 1

Sender 2

Genauso wie bei den Variablen legst du einen Namen für die Botschaft (Nachricht) fest, damit sie dir anschließend zur Verfügung steht (sowohl beim Sender, als auch beim Empfänger).

Neue Botschaft anlegen:

1. Klicke auf den kleinen Pfeil!

2. Wähle einen Botschaftsnamen! (z.B. Geh)

3. Die Botschaft wird eingefügt!

Der Empfänger:

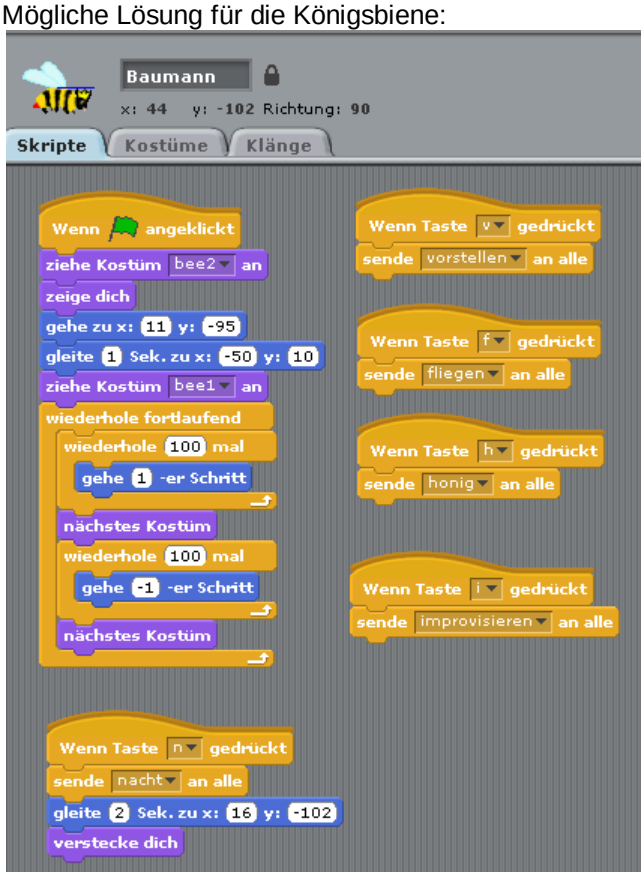
... hat nur eine Möglichkeit eine Botschaft zu empfangen:

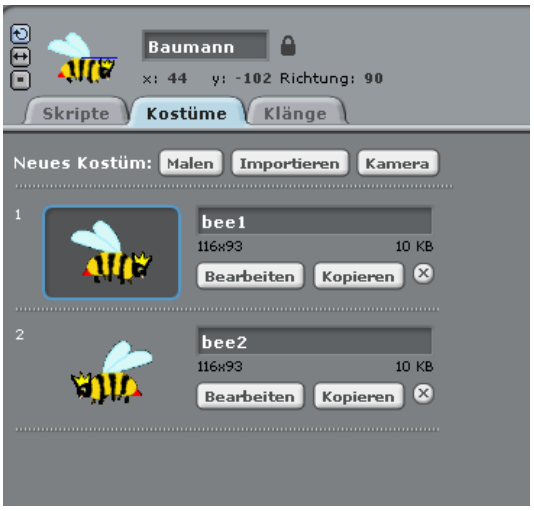
Empfänger (noch ohne Botschaft)

Empfänger mit eingefügter Botschaft

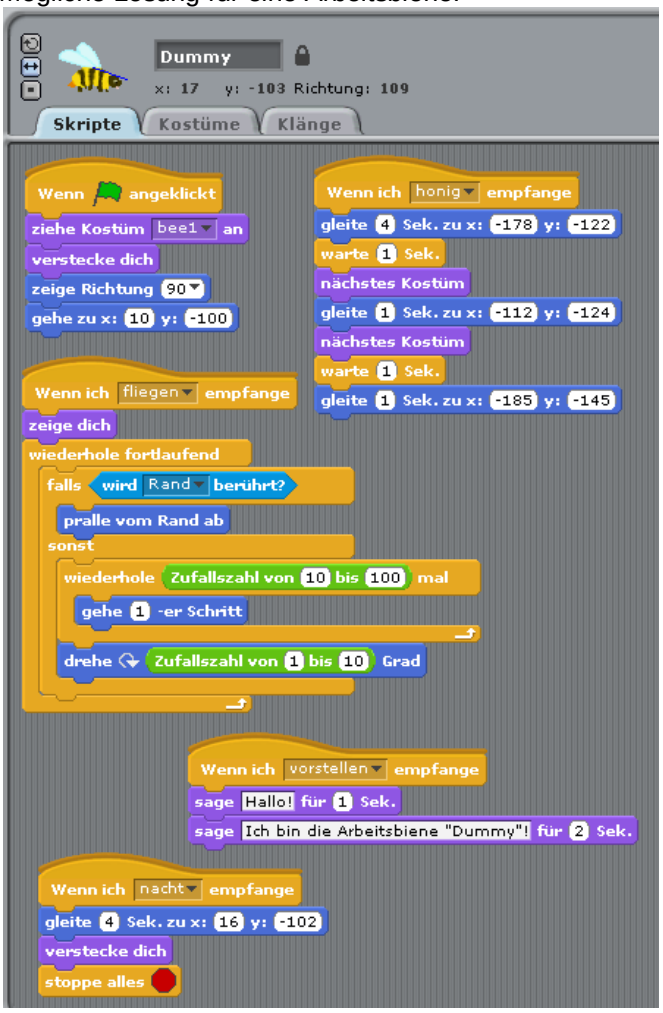
4. Antizipiertes SuS-Ergebnis:

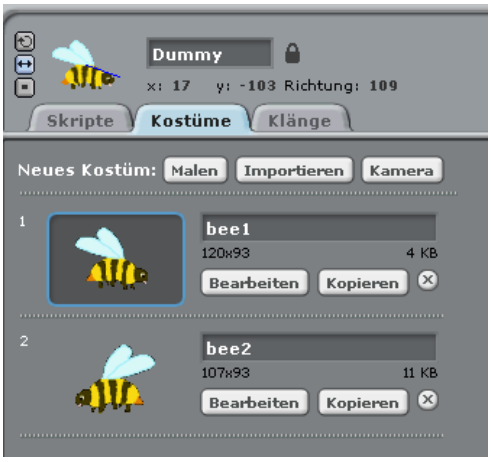
Mögliche Lösung für die Königsbiene:





Mögliche Lösung für eine Arbeitsbiene:

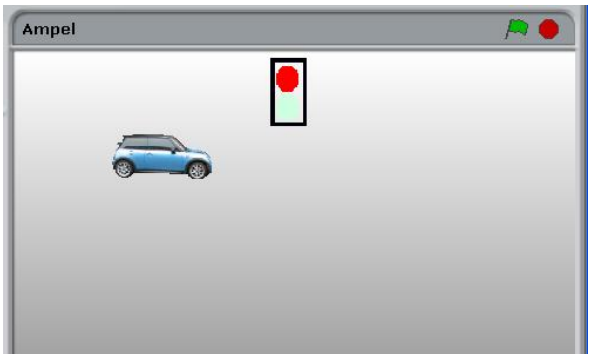




vorgegebene Lösung für die Bühne (den Hintergrund):

5. Einstiegsbeispiel Ampel:

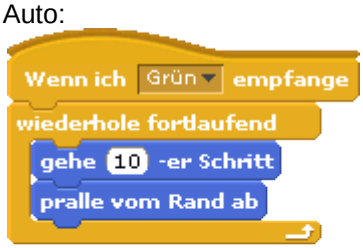
Bühne:

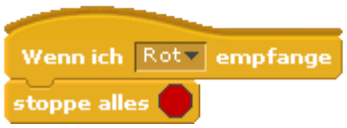
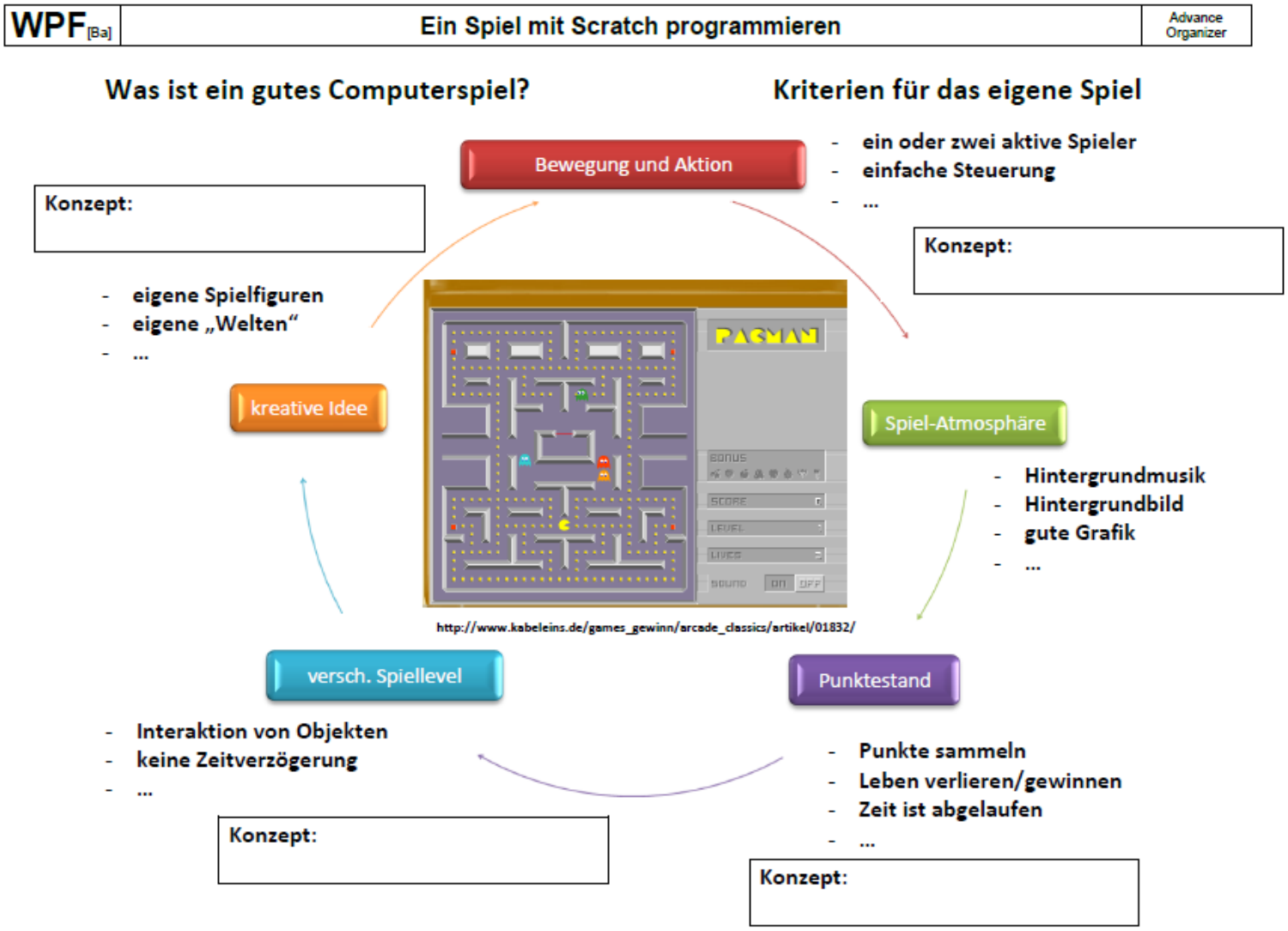


Programmierung der einzelnen Objekte:

Ampel:

Auto:





6. Advance Organizer: