Eckart Modrow, Didaktik der Informatik, Universität Göttingen
emodrow@informatik.uni-goettingen.de

Materialien für den Informatikunterricht an Gymnasien und Gesamtschulen

Themenbereich 1: Information und Daten
- Teil 1: Information -

mit grafischen Programmiersprachen als Werkzeugen

- Orientierung an den KCs und Bildungsstandards
- mit didaktischer Analyse
- für handlungsorientierten Unterricht
- weitgehende Nutzung grafischer Programmiersprachen
- in unterschiedlichen Kontexten
- incl. Lösungsvorschlägen
- mit unterschiedlichen, frei zugänglichen Werkzeugen realisiert

Inhalt

1. Information und Daten

1.1 Formale Vorgaben für die Sekundarstufen[1]

Die GI gibt für beide Sekundarstufen Kompetenzkataloge an. An dieser Stelle betone ich von den prozessbezogenen Kompetenzen besonders das Modellieren und Implementieren (*Schülerinnen und Schüler aller Jahrgangsstufen erstellen informatische Modelle zu gegebenen Sachverhalten, implementieren Modelle mit geeigneten Werkzeugen und reflektieren Modelle und deren Implementierung.*) sowie das Begründen und Bewerten (*Schülerinnen und Schüler aller Jahrgangsstufen stellen Fragen und äußern Vermutungen über informatische Sachverhalte, begründen Entscheidungen bei der Nutzung von Informatiksystemen, wenden Kriterien zur Bewertung informatischer Sachverhalte an.*). Die GI begründet diese Kompetenzen kurz wie folgt:

> *„Der Prozess der Modellierung ist nicht nur Lerninhalt, sondern auch durchgängige Methode des Informatikunterrichts, wobei aber auch die Implementierung unverzichtbar ist, um das Ergebnis der Modellbildung erlebbar zu machen. Begründen und Bewerten fördern die Kommunikations- und Argumentationsfähigkeit des Lernenden, ohne diesen Bereich ist der Umgang mit Informatiksystemen nur intuitiv oder spielerisch und häufig durch Einflüsse aus Medien bestimmt."*

Im Inhaltsbereich fordert die GI für die Mittelstufe, dass Schülerinnen und Schüler aller Jahrgangsstufen den Zusammenhang von Information und Daten, verschiedene Darstellungsformen und Operationen auf Daten und deren Interpretation in Bezug auf die dargestellte Information verstehen und diese Operationen auf Daten sachgerecht durchführen. In der Oberstufe unterscheiden die Lernenden zwischen Zeichen, Daten und Information sowie zwischen Syntax und Semantik, analysieren Daten hinsichtlich ihrer Struktur, bilden Information als Daten mit Datentypen und in Datenstrukturen ab, verwenden, modellieren und implementieren Operationen auf statischen und dynamischen Datenstrukturen, erstellen zu einem Realitätsausschnitt ein Datenmodell und implementieren es als Datenbank, untersuchen und organisieren Daten unter Beachtung von Redundanz, Konsistenz und Persistenz, verwenden eine Abfragesprache zur Anzeige und Manipulation von Daten und interpretieren die Daten. Im erhöhten Anforderungsniveau verwenden, modellieren und implementieren sie Operationen auf komplexen Datenstrukturen und entwickeln zu einem Ausschnitt der Lebenswelt mit komplexen Beziehungen eine Datenbank.

Die aktuellen Kerncurricula übernehmen diese Vorgaben weitgehend.

[1] Die Vorschläge der GI für Standards: http://www.informatikstandards.de/index.htm
GI: Gesellschaft für Informatik

1.2 Didaktische Analyse[2]

In den Kerncurricula sind für Praktiker neben den genannten Inhaltsbereichen die Beispielaufgaben besonders interessant, weil sich aus ihnen besonders gut eine Vorstellung von dem intendierten Unterricht gewinnen lässt. Im betrachteten Gebiet finden sich traditionell behandelte Themen aus dem Bereich der Datenstrukturen und Datenbanken, aber praktisch nichts über Informationen. Dieser Begriff tritt weitgehend nur innerhalb von Wortkombinationen (*Informations*technik, *Informations*gesellschaft, …) auf, und er wird widersprüchlich verwandt. Im Oberstufen KC Hessen[3] z. B. wird Information definiert als „*die Semantik einer Aussage, Beschreibung, Anweisung, Mitteilung oder Nachricht*“. Wie diese Semantik „*durch Maschinen automatisch verarbeitet*“ werden soll (Oberstufen KC NRW[4]) erschließt sich mir nicht ganz. Aus einem nicht genügend scharf definierten Begriff lässt sich anscheinend nicht so leicht Unterricht ableiten, auch wenn er prominent in den Kompetenzbereichen benutzt wird. Es ist also lohnend, sich etwas eingehender mit dem benutzten Informationsbegriff zu beschäftigen. Die KCs scheinen diese Notwendigkeit auch zu sehen, da sie im Gegensatz zu den meisten anderen Begriffen für diesen Definitionen liefern.

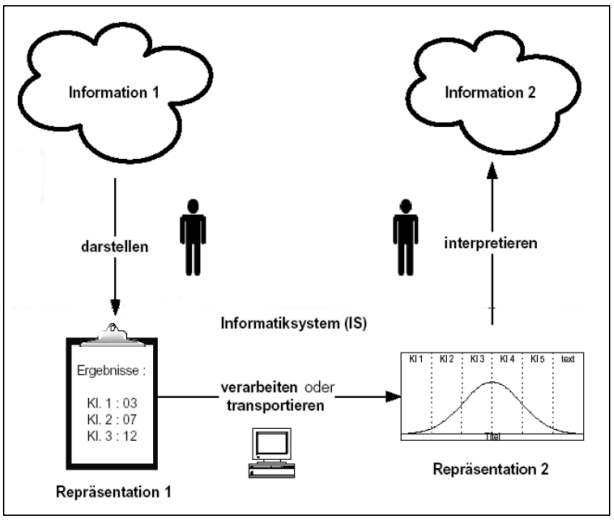


Bild 1: Testauswertung nach Hubwieser

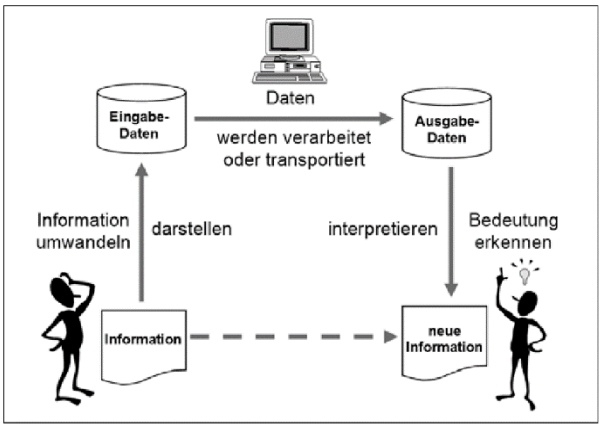


Bild 2: Schema der Informationsverarbeitung aus den GI-Bildungsstandards

1.2.1 Daten, Information und Wissen

Der Informationsbegriff wird im Rahmen der Informationszentrierten Informatikdidaktik im Wesentlichen von Norbert Breier und Peter Hubwieser[5] eingeführt. Breier meint schon 1994: „*In einem zeitgemäßen* Informatikunterricht *steht meines Erachtens nicht der Algorithmus, sondern die Information als Erscheinungsform der realen Welt im Mittelpunkt*.“ Hubwieser erläutert den Informationsbegriff anhand eines Bildes, das sich auf die Daten einer Testauswertung bezieht (Bild 1). In leicht abgewandelter Form taucht es dann in den GI-Bildungsstandards als allgemeines Modell der Informationsverarbeitung (Bild 2) und, daraus übernommen, in verschiedenen

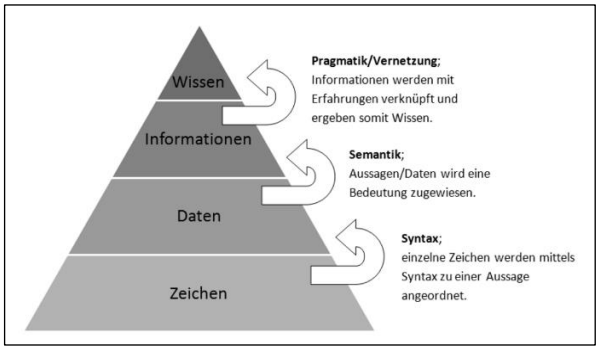


Bild 3: die Wissenspyramide[5]

[2] Dieser Abschnitt erscheint in ähnlicher Form in LOG IN 187/188 (2017)
[3] https://kultusministerium.hessen.de/schule/kerncurricula/gymnasiale-oberstufe/informatik
[4] http://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/KLP_GOSt_Informatik.pdf
[5] Beide Quellen aus Hubwieser, P., Didaktik der Informatik, Springer 1998

KCs auf. Leitet man aus diesem Schema Inhaltsbereiche ab, dann kommen schon Hubwieser, aber auch die GI in ihrem Gesamtkonzept für eine informatische Bildung[6], sehr schnell z. B. zur automatischen Verarbeitung und Vernetzung von *Repräsentationen* (Daten). Aus der „informationszentrierten" Didaktik wird sofort eine „datenzentrierte", wenn es um konkreten Unterricht geht.[7] Das Problem tritt also nicht erst in den Kerncurricula auf.

Aus den Diagrammen wird deutlich, dass der in der Informatikdidaktik benutzte Informationsbegriff weder mit dem Shannonschen der Informationstheorie noch mit der umgangssprachlichen Gleichsetzung von Information und Daten viel zu tun hat. Sie trennen die Informationen aber nicht eindeutig genug von den Daten, und die betroffenen Personen stehen eher hilflos daneben. Wir benötigen deshalb präzise und miteinander kompatible Definitionen für die benutzten Begriffe. Hilfreich scheint mir dafür die oft benutzte *Wissenspyramide*[8] (Bild 3) zu sein, die neben *Daten* und *Informationen* auch noch die Ebenen des *Wissens* und der *Zeichen* enthält. Als Ausgangspunkt wählen wir die Definition von Wissen aus der Wikipedia[9]:

> *Wissen wird […] als ein für Personen oder Gruppen verfügbarer Bestand von Fakten, Theorien und Regeln verstanden, die sich durch den größtmöglichen Grad an Gewissheit auszeichnen, so dass von ihrer Gültigkeit bzw. Wahrheit ausgegangen wird.*

Wissen ist somit an Personen gebunden und kann folgerichtig z. B. innerhalb heutiger Maschinen nicht existieren. Dort finden wir Daten. Da Wissen nicht vollständig und sogar falsch sein kann, ergeben sich Lücken in der Gewissheit, die durch Informationen geschlossen oder verringert werden können[10].

> *Information ist die Teilmenge von Wissen, die von einer bestimmten Person oder Gruppe in einer konkreten Situation benötigt wird und häufig nicht explizit vorhanden ist.*

Diese Definition entspricht in etwa der aus den Bildungsstandards, „*Information ist der kontextbezogene Bedeutungsgehalt einer Aussage, Beschreibung, Anweisung, Mitteilung oder Nachricht.*", allerdings auf das durch die Information geänderte Wissen bezogen. Information ist ebenfalls an Personen gebunden, die den Bedeutungsgehalt der Daten erkennen und bewerten. Sie ist zeit- und situationsabhängig. Erhält eine Person z. B. eine Nachricht zweimal, dann ist der Informationsinhalt beim zweiten Mal sehr viel geringer, denn die Wissenslücke wurde schon von der ersten Information geschlossen. Informationen hängen einerseits von den zu ihrer Übermittlung benutzten Daten ab, aber andererseits auch vom Zustand des Empfängers. Dieser baut *pragmatisch* Informationen in sein bestehendes Wissen ein, *vernetzt* sie mit diesem – oder auch nicht. Daten werden durch *Zeichen* des gewählten Zeichensatzes repräsentiert, den wir hier als Code auffassen können. Die *Syntax* dieser Repräsentation beschreibt die Struktur dieser Darstellung.

Bis hierher gibt es keine Probleme: Informationen befinden sich im Kopf, Daten im Rechner. Leider folgt jetzt der Versuch, die Daten informationsorientiert zu sehen. Der Zusammenhang zwischen Daten und Informationen wird dadurch inkonsistent definiert. In den Bildungsstandards und Kerncurricula finden wir:

> *Daten sind eine Darstellung von Information in formalisierter Art, geeignet zur Kommunikation, Interpretation und Verarbeitung.*

Die Wikipedia meint:

> *Daten werden als Zeichen (oder Symbole) definiert, die Informationen darstellen und die dem Zweck der Verarbeitung dienen.*[11]

Der oben genannte Informationsbegriff ist personenbezogen. Informationen können also nicht ohne die interpretierende Person gesehen werden, z. B. weil dieselben Daten für unterschiedliche Personen

---

[6] http://www.informatische-bildung.de/

[7] http://www.vlin.de/sek1/Mittelstufeninformatik.pdf

[8] https://derwirtschaftsinformatiker.de/2012/09/12/it-management/wissenspyramide-wiki/

[9] https://de.wikipedia.org/wiki/Wissen

[10] https://de.wikipedia.org/wiki/Information

[11] https://de.wikipedia.org/wiki/Daten

ganz unterschiedliche Informationen darstellen können. Auch in der Definition der Bildungsstandards sind Informationen kontextabhängig. Ohne diesen Kontext verlieren sie die Eigenschaft, Information zu sein. Sie werden auf das reduziert, was sie ohne Bedeutung sind: eben Daten. Im Modell der Wissenspyramide sind die Verhältnisse klarer: der Empfänger interpretiert die empfangenen Daten und versucht, sich ihre Semantik zu erschließen. Dieser Schritt erfolgt vor der Vernetzung mit seinem bestehenden Wissen und weitgehend unabhängig von dieser. Die Interpretation ist vom Empfänger und seinem Zustand abhängig, sie kann nicht ausschließlich aufgrund der Daten erfolgen. Nach der Interpretation entscheidet der Empfänger, ob die Bedeutung der Daten für ihn eine Information darstellt.

Wir sollten m. E. darauf verzichten, den Datenbegriff wie oben geschehen in das informationszentrierte Schema zu pressen. Daten sind eine Kategorie an sich, gebunden an eine physische Repräsentation. Misst z. B. eine Meeressonde Temperaturwerte, speichert diese und geht anschließend verloren, dann sind die physisch repräsentierten Messwerte als Daten existent, auch wenn sie bedauerlicherweise nie zu einer Information werden. Speichert ein Betriebssystem den Systemzustand in Protokolldateien, dann sind diese Daten vorhanden, auch wenn sie nie von Menschen ausgewertet werden. Die Definition von Daten als Repräsentationen von Informationen verwechselt den oben genannten Informationsbegriff mit dem umgangssprachlichen und führt zu der unschönen Situation, dass der Bedeutung der Information Genüge getan zu sein scheint, wenn Daten und ihre Strukturen betrachtet werden.

Unsere Untersuchung hat ein einfaches Ergebnis: Die beiden untersten Ebenen der Wissenspyramide sind der Fachwissenschaft Informatik zugänglich. Sie sind mit den tradierten Inhaltsbereichen verknüpft. Die beiden oberen sind zumindest teilweise intrapersonal, gehen über die reine Fachwissenschaft hinaus. Wie der Bereich „Informatik und Gesellschaft“ beziehen sie sich auf die Bedeutung der Informatiksysteme, diesmal nicht so sehr politisch und sozial gesehen, sondern auf die persönliche Betroffenheit bezogen. Der Informationsbegriff gehört zum allgemeinbildenden Beitrag der Schulinformatik. Dieser wird nicht erreicht, wenn die Behandlung von datenbezogenen Themen mit informationsbezogenen gleichgesetzt wird.

1.2.2 Unterschiedliche Szenarien

Wir wollen die Konsequenzen unserer Überlegungen in vier Situationen durchspielen. Dafür benennen wir die beiden Akteure des „Informationsübertragungsschemas“ als *Susi* (Sender) und *Emil* (Empfänger) und reduzieren die Beschriftung.

1. Fall: Susi sendet die Nachricht *„Bin angekommen!“* an Emil. (Bild 4)

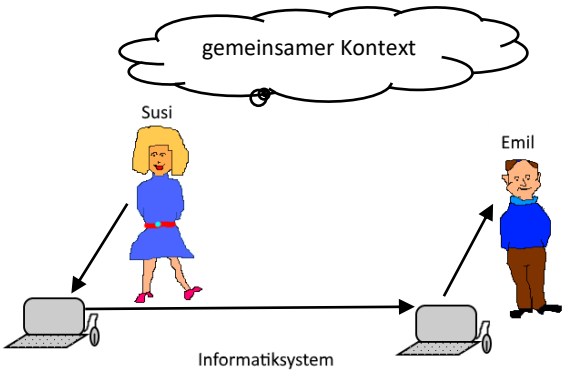


Bild 4: Kommunikation in gegebenem Kontext

Die Nachricht kann für Emil nur eine Bedeutung haben, wenn Susi und er sich über ihren Sinn im Klaren sind. Weiß also Emil, dass Susi entweder auf dem Weg nach Hannover oder zu sich selbst ist, dann kann er die Nachricht interpretieren, sogar inklusive von Subtexten wie dem fehlenden „gut“, die einige Komplikationen erwarten lassen. Susi wiederum weiß, dass Emil auf ihre Nachricht wartet und sie in ihrer Kürze verstehen wird. Sie kann ihre Information durch entsprechende Daten ausdrücken. Susi und Emil handeln innerhalb eines gemeinsamen Kontextes, der es gestattet, die Nachricht zu interpretieren. Ohne diesen Kontext ist das nicht möglich, und deshalb sollte der Kontext auch ins Schema aufgenommen werden.

Die Rolle des Informatiksystems ist in diesem Fall völlig nebensächlich, klar vom Informationsaustausch getrennt. Susi hätte auch laut rufen, eine Postkarte senden, die Nachricht trommeln oder per

Brieftaube transportieren lassen können. Und umgekehrt ist die Fähigkeit des Informatiksystems, Texte geeignet zu kodieren, die Zeichen zu transportieren und wieder darstellen zu können, völlig unabhängig vom Informationstransport. Die Interpretationsaufgabe des Systems besteht darin, die Zeichen so zu kennzeichnen, dass sie als Text erkannt und durch ein geeignetes Teilsystem dargestellt werden können. Diese Aufgabe erfolgt automatisch z. B. durch Kennzeichnung der Datenpakete oder der Datei aufgrund der vereinbarten Syntax. Mit Verständnis hat das nichts zu tun.

Insgesamt ist das Beispiel, das durch das überall verwendete Grundschema nahegelegt wird, unter informatischen Informationsgesichtspunkten unergiebig. Ich empfehle, das Schema nicht zu benutzen, wenn der Informationsaspekt im Unterricht hervorgehoben werden soll.

2. Fall: Susi sendet die Nachricht *„meistens nachmittags“* an Emil. (Bild 5)

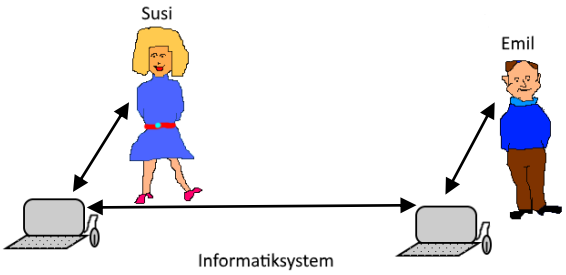


Bild 5: Kommunikation mit offener Fragestellung

In diesem Fall soll der gemeinsame, in vielen Krisen gestählte Kontext nicht vorhanden sein, weil Susi und Emil mehr oder weniger zufällige Kommunikationspartner im Netz sind. Da Susi ohne diesen Kontext keine angemessenen Daten zusammenstellen und übermitteln kann, muss Emil den Kontext zuerst herstellen. Der Kommunikationsprozess muss deshalb von ihm gestartet werden, indem er eine entsprechende Frage an Susi stellt. Die Frage wird von Susi so interpretiert, dass sie die gewünschte Information identifizieren und in Daten umsetzen kann. Die empfangenen Daten muss Emil wiederum als Antwort auf seine Frage interpretieren und als die gesuchte Information bewerten. Das wird im Schema durch Doppelpfeile dargestellt. Dabei kann auf beiden Seiten viel schiefgehen. Susi kann die Frage falsch verstehen, wenn diese nicht völlig eindeutig formuliert ist. Sie kann also falsche Informationen von Emil erhalten und entsprechend falsche Antworten erzeugen, die von Emil wiederum falsch verstanden werden können.

Auch in diesem Fall passieren die interessanten Dinge in den Köpfen der Beteiligten. Wir könnten die Bedeutung der nichtverbalen Kommunikation diskutieren und die Rolle der Emoticons thematisieren, das Textverständnis in unterschiedlichen sozialen oder kulturellen Kontexten untersuchen oder den Bedarf an Bildtelefonie. All das sind wichtige und diskussionswürdige Schulthemen. Ihnen gemeinsam ist aber, dass wir uns ihnen weder über Kenntnisse der Netzprotokolle noch der benutzten Datenstrukturen nähern. Die informatischen Fachthemen sind für die hier diskutierte Rolle der Information irrelevant.

3. Fall: Susi sendet die Nachricht *„Berlin, Bern, Bukarest“* an Emil. (Bild 6)

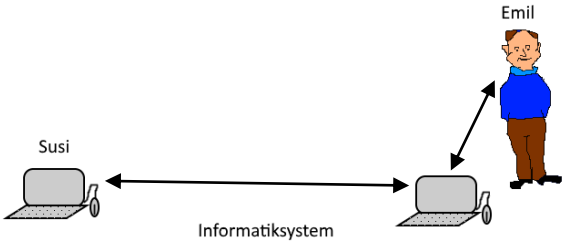


Bild 6: Kommunikation mit eindeutiger Fragestellung

In diesem Fall hat Emil seine Frage so präzise gestellt, dass Susi sie eindeutig auswerten kann. Eine Interpretation und somit ein für Verständnis geeigneter Kontext sind nicht erforderlich. Damit entfällt aber auch die Rolle von Susi als Person. Sie kann durch einen Computer ersetzt werden, der die Frage beantwortet, solange einige Syntaxregeln eingehalten werden. Emil kann z. B. fragen:

SELECT name FROM staedte WHERE istHauptstadt = „ja“ AND name like „B%“ LIMIT 3;

Die Information ist in diesem Fall sehr einseitig verteilt. Emil weiß, welche Information er benötigt. Er beschreibt die Daten, die zum Schließen der Wissenslücke erforderlich sind und ruft diese von einem Informatiksystem ab. Weder auf dem Weg von Emil zum System noch innerhalb des Systems ist auch nur ein Ansatz von Information zu finden. Diese entsteht erst in Emils Kopf, nachdem er Susis Antwort erhalten hat.

Da dieser dritte Fall sehr direkt der Kommunikation in und mit Informatiksystemen entspricht, ist seine Analyse für die Lernenden wichtig. Egal, ob sie digitale Assistenten benutzen, Datenbanken befragen oder Suchmaschinen benutzen: von ihnen wird eine eindeutige Beschreibung der zur Fragenbeantwortung erforderlichen Daten erwartet – ob es ihnen passt oder nicht. Die Systeme spiegeln zwar Verständnis vor oder es wird ihnen von den Benutzern zugesprochen, aber sie besitzen es nicht. Das Bewusstsein dafür verhindert die Überbewertung der erhaltenen Antworten und die Unterbewertung der Verantwortung des Benutzers für seine Fragestellung. Je mehr die Rolle der Kommunikationspartner verwischt wird, desto unklarer wird die Bewertung der Ergebnisse.

4. Fall: Emil überträgt seine Aufgaben an ein Programm und geht schwimmen. (Bild 7)

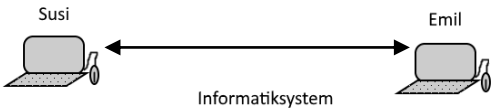


Bild 7: Kommunikation ohne menschliche Partner

Nachdem Susi schon durch einen Algorithmus, in diesem Fall durch einen SQL-Server, ersetzt wurde, könnte ja auch Emil auf die Idee kommen, dass seine Aufgaben besser und schneller von einem Algorithmus wahrgenommen werden können. Er behauptet, dass er seine Interpretation von Susis Daten ausreichend präzise durch ein Programm beschreiben kann, das den Daten Informationen entnimmt und auch gleich ggf. erforderliche Aktionen veranlasst. Stimmt das? Wir wählen als Beispiel den Hochfrequenzhandel im Bankensystem. Susi übermittelt die aktuellen Kurse an ihrem Börsenplatz, Emil bewertet die Unterschiede zu seiner Börse und veranlasst entsprechende Kauf- oder Verkaufsanweisungen.

Da Emil, jetzt als Maschine, über kein Wissen verfügt, können auch keine Wissenslücken bei ihm geschlossen werden. Um Information im definierten Sinne kann es sich also nicht handeln. Der Algorithmus Emil ist zwar aus dem Wissen der Person Emil über die Abläufe im Börsenhandel entstanden, er stellt dieses Wissen aber nicht vollständig dar, und vor allem, er vernetzt es nicht mit Emils restlichem Wissen. Die Lücken in diesem Wissen, die für konkrete Reaktionen im Börsenhandel geschlossen werden müssen, erfordern die aktuellen Börsenwerte. Der Algorithmus verfügt dafür über Variable, also Leerstellen, die von Susi aktualisiert werden. In Abhängigkeit von diesen Werten durchläuft Emil seine Anweisungsfolgen in unterschiedlicher Reihenfolge und löst die entsprechenden Aktionen aus. Dafür ist keinerlei Interpretation erforderlich. Es handelt sich um einen reinen Automatisierungsvorgang.

1.2.3 Folgerungen

Wir können aus den vier betrachteten Fällen einiges lernen. Die ersten beiden zeigen, dass menschliche Kommunikation problematisch sein kann, und zwar unabhängig vom benutzen Medium. Dieses erhält seine Bedeutung daraus, dass es Kommunikation ermöglicht und aus seiner Verbreitung. Für die dabei auftauchenden technischen Fragen der Datenverarbeitung ist der Informationsbegriff irrelevant.

Interessanter sind die beiden anderen Fälle. Der dritte beschreibt ganz gut die Rollen von Benutzer und Informatiksystem bei der Informationsbeschaffung. Die Intelligenz liegt dabei vollständig beim Benutzer. Dieser beschreibt die zur Erzeugung der gesuchten Information erforderlichen Daten und ist damit auch für diese Beschreibung verantwortlich. Ist die Beschreibung unpräzise, dann erhält er entsprechende Antworten. Befragt der Fragende wie in Szenario 2 einen menschlichen Experten, dann muss sich dieser die gesuchte Information aus dem Kontext erschließen und mithilfe des Informatiksystems die zur Beantwortung erforderlichen Daten zusammenstellen und übermitteln. Er übernimmt dann auch die Verantwortung für deren Relevanz. In Szenario 3 steigen die Anforderungen an den Frager erheblich, denn er muss jetzt Experte sein. Ausreden gibt es nicht mehr. Seine Frage wird immer ausgewertet, z. B. über statistische Zusammenhänge oder indem Übereinstimmungen mit dem Fragetext wörtlich im Netz gesucht werden, sie wird aber nicht verstanden. Um die anfallenden Daten überhaupt sortieren zu können, muss das System den fehlenden Kontext ergänzen, z. B. durch Auswertung vergangener Fragen oder ähnlicher Fragen anderer. Die Gefahr, dass so z. B. „Echokammern“

entstehen, die Daten immer der gleichen Tendenz erzeugen, wird als aktuelles, die Demokratie gefährdendes Problem diskutiert.

Der Informationsgesichtspunkt führt in diesem Szenario auf die Frage, was der Benutzer wissen muss, um angemessene Fragen stellen zu können; wissen sowohl vom Thema der Fragestellung wie von der Funktionsweise des benutzen Systems. Die traditionellen Fachthemen der Schulinformatik werden damit um einen Aspekt erweitert, der geeignet ist, die Relevanz eben dieser Fachinhalte vor dem Hintergrund des Lebens in einer durch Informatiksysteme geprägten Gesellschaft zu bewerten. Die Informationszentrierte Didaktik erfordert die Entwicklung neuer Unterrichtskomponenten, um das Fach in Richtung von aktuellem allgemeinbildendem Unterricht weiterzuentwickeln. Sie verzahnt die Fachinhalte mit deren gesellschaftlicher Bedeutung.

Der vierte Fall beschreibt die Übertragung von menschlichen Aufgaben an Informatiksysteme. Diese Menschen können aus ihrem Wissen und ihrer Erfahrung heraus beschreiben, wie in unterschiedlichen Situationen reagiert werden sollte. Im Rahmen dieser Beschreibung reagieren die automatisierten Systeme dann in den Menschen vergleichbarer Weise, meist sogar schneller und zuverlässiger. Was aber passiert, wenn die Beschreibung unvollständig ist oder neue Situationen auftreten? Da die ausgewerteten Daten ihren Datencharakter im gesamten Prozess beibehalten, also nie zu Informationen werden, wird ihre Semantik auch nie erschlossen. Bedeuten sie etwas anderes als im Algorithmus vorgesehen, dann versteht niemand diese Bedeutungsänderung, weil sie nicht mit vorhandenem Wissen aus vielleicht ganz anderen Gebieten verknüpft werden kann. (Nebenbei: auch die Nutzung Neuronaler Netze ändert an dieser Bewertung nichts.) Die klare Trennung von Daten und Informationen ermöglicht in diesen Fällen, z. B. die Verantwortlichkeit für die Konsequenzen der Automatisierung zu diskutieren (etwa beim autonomen Fahren) und die ethischen Grenzen auszuloten. Der Informationsaspekt schafft Klarheit bei der Argumentation und verhindert, dass gesellschaftlich relevante Fragen durch den Rückzug auf Fachinhalte vernebelt werden. Sie ermöglicht den politischen Diskurs über die Stellung der Informatiksysteme.

1.2.4 Fazit

Die Informationszentrierte Didaktik hat dazu geführt, dass der Informationsbegriff etwas inflationär in fast allen Gebieten der Schulinformatik benutzt wird. Er verliert dabei an Schärfe und vor allem die Funktion, Orientierung bei der Planung von Unterricht zu geben. Den traditionellen Inhaltbereichen wie z. B. dem der Daten und Datenstrukturen schadet es zwar nicht, dass ihnen jetzt zusätzlich der Anspruch zugeschrieben wird, ebenfalls dem Informationsaspekt Rechnung zu tragen. Es leidet aber Chance, den Informatikunterricht in Richtung seiner allgemeinbildenden Funktion zu akzentuieren. Reduzieren wir den Informationsbegriff dagegen auf seine ursprüngliche Bedeutung, dann erweitern wir den Themenkanon der Schulinformatik um gesellschaftlich relevante Aspekte, die sich direkt auf die Planung der Curricula auswirken können.

Betrachten wir z. B. den Begriff der „Wissensgesellschaft“, aus dem teilweise gefolgert wird, dass Wissen nicht mehr erworben werden muss, wenn es „im Netz“ allen zur Verfügung steht, dann sieht man sofort, dass es so einfach nun doch nicht geht. Im Netz finden wir kein Wissen, sondern Daten. Es ergeben sich stattdessen einige Fragen, die geklärt werden müssen, bevor es mit der Informationsgewinnung in der Wissensgesellschaft so richtig klappen kann:

- Über welches Grundgerüst von Wissen müssen die Lernenden verfügen, um ihre Wissenslücken überhaupt erkennen zu können?
- Welche Kompetenzen müssen die Lernenden erwerben, um die zum Schließen der Wissenslücken erforderlichen Daten präzise beschreiben zu können? Können sie überhaupt beschreiben, was sie nicht wissen?
- Welches Wissen über die Daten liefernde Informatiksysteme müssen die Lernenden erwerben?

- Wie lernen die Lernenden, die Relevanz der gelieferten Daten bezogen auf ihre Fragestellung einzuschätzen?
- Was passiert, wenn die Antworten „gefärbt“, z. B. auf die Fragenden abgestimmt werden?
- Welche Daten gewinnt das antwortende Informatiksystem aus den Fragen? Welche Informationen können daraus abgeleitet werden?

Der Informationsbegriff erweist sich also ziemlich eindeutig im Bereich „Informatik und Gesellschaft“ als wirksam. Wir sollten ihn da auch belassen. Diese Beschränkung beschneidet m. E. nicht seine Bedeutung, im Gegenteil: Wenn ein Begriff die Ausrichtung eines Schulfachs deutlich akzentuieren kann, dann ist das nicht wenig. Es ist viel.

1.3 Unterricht zum Informationsbegriff

Entsprechend der didaktischen Analyse zu diesem Bereich werden die Begriffe „Information“ und „Daten“ getrennt. Beiträge zum Informationsbegriff werden im Bereich „Informatik und Gesellschaft“ angesiedelt, aber nicht gesondert behandelt, sondern sie treten als Ergänzungen z. B. beim Thema der „Daten und ihrer Verarbeitung“ auf. Dabei soll handlungsorientiert gearbeitet werden, d. h. die Lernenden realisieren ihre Vorstellungen und Ideen mithilfe geeigneter Werkzeuge, hier: meist grafischer Programmiersprachen, innerhalb eines auch unter dem Informationsaspekt interessanten Kontextes.

Nach unseren didaktischen Vorstellungen[12] entwickeln wir in diesem Themenbereich die folgenden Kompetenzbereiche besonders:

Die Lernenden …

… *wählen in Abhängigkeit von der Problemstellung einen geeigneten Modelltyp aus und begründen ihre Wahl.*
… *reduzieren das modellierte System auf die im Modelltyp relevanten Eigenschaften und erstellen ein entsprechendes Modell.*
… *diskutieren die Möglichkeiten und Grenzen des Modells sowie die Auswirkungen seines Einsatzes.*
… *kooperieren miteinander sowie mit Informatiksystemen.*
… *formulieren ihre Lösungsideen im informatischen Begriffssystem.*
… *entwickeln eigene Algorithmen.*
… *identifizieren im gestellten Problem algorithmisierbare Abläufe.*
… *realisieren Algorithmen mithilfe geeigneter Werkzeuge als ausführbare Programme.*
… *dokumentieren die Ergebnisse ihrer Arbeit und präsentieren diese geeignet.*

Da der in den Kerncurricula benutzte Informationsbegriff thematisch kaum zur Kerninformatik gehört, müssen Erkenntnisse dazu weitgehend aus dem Kontext folgen, in dem die Fachinhalte angesiedelt werden. Die Unterrichtsplanung muss also Szenarien schaffen, aus denen sich Bedarf nach der inhaltlichen Klärung der Fachfragen ergibt, aber ebenso Diskussionsbedarf der gesellschaftlichen Folgen – hier: der übermittelten Informationen. Der betrachtete Einsatz von Informatiksystemen muss die Lernenden im fortgeschrittenen Unterricht möglichst im Wortsinn „betroffen“ machen, anfangs aber nur die Rollen der Systemteile verdeutlichen. Wir orientieren uns an den vier oben betrachteten Szenarien und wählen jeweils geeignete Kontexte, die unterschiedliche Fachfragen aufwerfen. Die jeweilige Diskussion der Rolle der Information sollte im gegebenen Kontext diskutiert werden. Dieser Aspekt wird aber meist explizit thematisiert werden müssen, weil er bedingt durch die unterschiedliche Verwendung der Begriffe nicht unbedingt von selbst akut wird.

[12] Eckart Modrow, Kerstin Strecker; Didaktik der Informatik, de Gruyter Studium, 2016

1.3.1 Zu Fall 1: Kommunikation in gegebenem Kontext

Wir benötigen Szenarien, in denen das Informatiksystem nur als Transportmittel für Nachrichten auftritt, die von den Beteiligten „verstanden“ werden. Selbst ist es damit erstmal nachrangig, im einfachsten Fall stellt es den Kontext nur dar, z. B. bei der Programmierung einer in Scratch[13] geschriebenen Geschichte. Das Informatiksystem ist aber trotzdem nicht irrelevant, weil einerseits dessen Gebrauch erlernt wird und somit spätere, eher „informatische“ Aufgaben vorbereitet werden. Andererseits stellt die Benutzung unterschiedlicher Objekte, die über Nachrichten kommunizieren, einen intuitiven Einstieg in die objektorientierte Modellierung dar.

Beispiel: **Im Gemüseladen**

Altersstufe: *frühe Sekundarstufe I*
Werkzeug: *Scratch2*
Material: *Im Gemüseladen.sb2*

Bild 8: Im Gemüseladen

Zwei Personen agieren in einem Laden, z. B. indem eine Kundin den Raum betritt (mit Beinbewegungen durch Kostümwechsel) und anschließend eine Botschaft („Ich bin da!“) sendet. Daraufhin erscheint die Verkäuferin, fragt nach den Wünschen, … - alles durch Botschaften gesteuert. Der Kontext ist in diesem Fall eindeutig, und da die Objekte nur auf bestimmte Botschaften reagieren, ist auch klar, was jeweils zu tun ist. Auch wenn die Situation trivial ist, besteht kein Zweifel an der Rollenverteilung: Botschaften im Informatiksystem bestehen aus Texten, die von den Handelnden interpretiert werden und ggf. Handlungen auslösen.

Das gegebene Animationsprogramm stellt einen einfachen Rahmen für den Anfangsunterricht bereit, der von den Lernenden modifiziert und ergänzt wird. Nicht zu unterschätzen ist dabei die Arbeit mit den Kostümen, z. B. um Bewegungen zu visualisieren. Der Umgang mit dem eingebauten Grafikeditor und anderen Grafikprogrammen, die dann wiederum unterschiedliche Grafikformate bereitstellen, was z. B. für die Transparenz des Hintergrunds wichtig ist, motiviert einen Teil der Lernenden mehr als der direkte Einstieg über Skripte. Sind verschiedene Kostüme erstellt, dann sollen sie natürlich auch benutzt werden – und dafür benötigt man dann Programmskripte. Der Umweg über die Grafik führt zur Algorithmik – allerdings anhand selbst erstellter (Teil-)Produkte, was die Motivation stark erhöhen kann.

Weshalb ist eigentlich die grafische Darstellung für die Lernenden so wichtig? Der Kontext erschließt sich eben nicht nur aus den Texten, sondern bei diesen einfachen Geschichten auch aus dem Aussehen, der Körperhaltung usw. der Akteure und aus deren Umfeld (siehe: „Im Bistro“ weiter unten). Was hier dargestellt wird, muss nicht mehr gesagt werden, ist aber entscheidend für die Interpretation. Es ist eben ein Unterschied, ob der Ausruf „Das ist ja Kohl!“ in einem Gemüseladen oder in einem Klassenzimmer erfolgt.

Das Beispiel führt zu unterschiedlichen Geschichten, die nach Herstellung eines definierten Anfangszustands („grüne Flagge gedrückt“) im Wechselspiel von gesendeten Botschaften und dadurch ausgelösten Ereignissen (mit deren Behandlung) in Anlehnung an die gegebenen Beispiele (siehe Bilder 9 und 10) entstehen. Die Qualität der Ergebnisse zeigt sich dann in der Phantasie, Komplexität und Witzigkeit der Geschichten.

[13] https://scratch.mit.edu/

Obwohl Botschaften, Ereignisse, ggf. Zustände, die sich durch lokale Variable („zusätzliche Attribute zu den schon vorhandenen“) beschreiben lassen, benutzt werden, sollte der Schwerpunkt des Geschehens nicht bei der Fachsprache liegen. Natürlich muss über die Vorgänge geredet werden, und dann kann man auch gleich die üblichen Begriffe benutzen. Das ist aber ein (durchaus wünschenswerter) Nebeneffekt. Ziel des Unterrichts ist es, die Kinder zu aktivieren und phantasievolles Agieren zu fördern. Die Benutzung einer formalisierten Sprechweise („Das Attribut *x-Wert* des Objekts *Kundin* wird durch Aufruf der Methode *ändere x um -5* neu gesetzt.“) gehört eher nicht dazu. Es reicht festzustellen, dass die Kundin nach links geht.

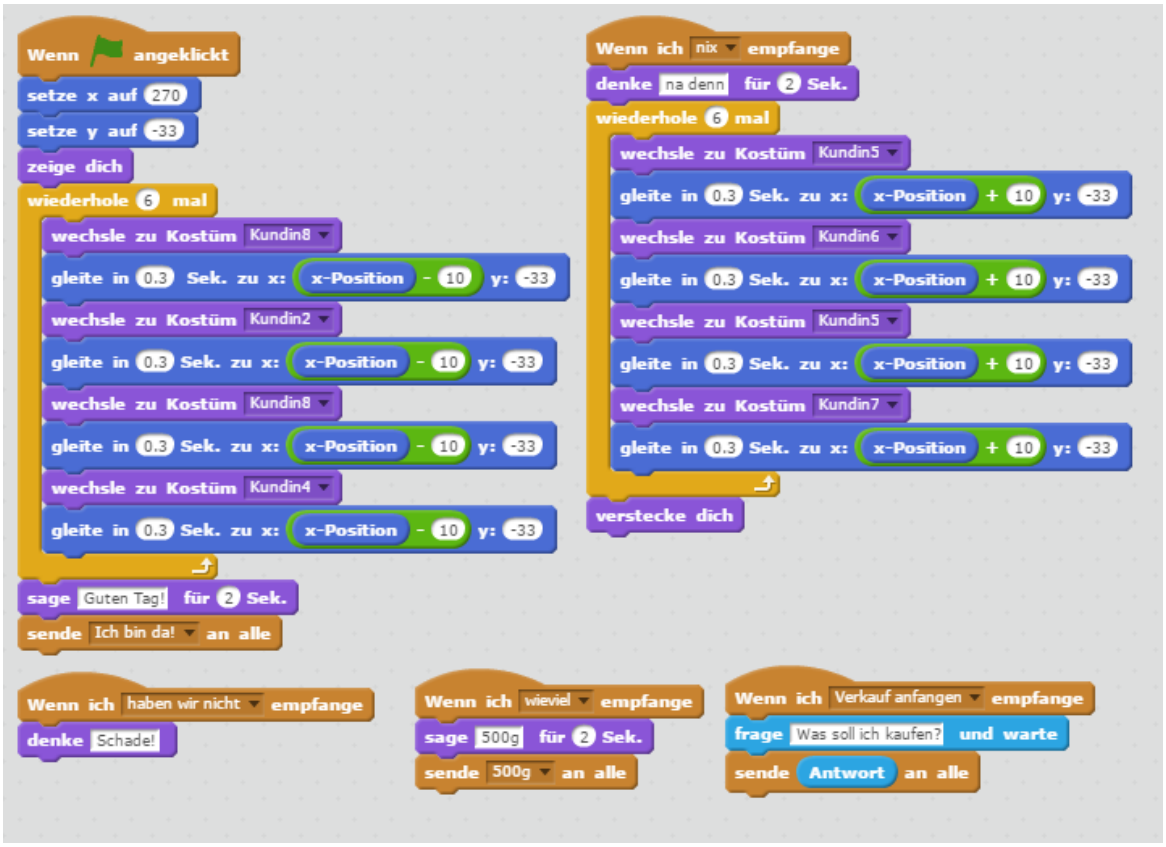


Bild 9: (erste) Skripte der Kundin

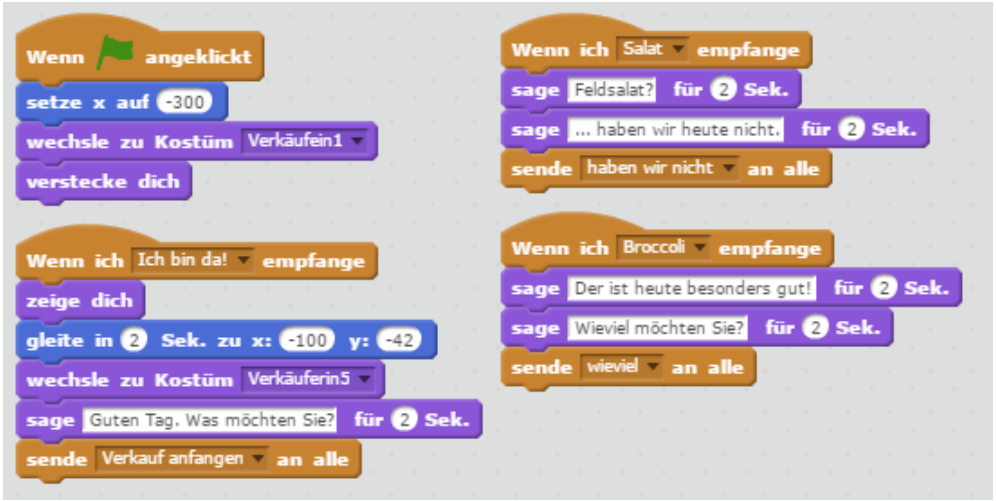


Bild 10: (erste) Skripte der Verkäuferin

alternatives Beispiel: **Schwimmer**

Altersstufe: *Sekundarstufe I*
Werkzeug: *Scratch2*
Material: *Schwimmer.sb2*

Wir zeichnen einen Schwimmer in verschiedenen Schwimmphasen und duplizieren ihn mehrmals. Auf eine Nachricht der Trainerin hin („los“) schwimmen die Schwimmer mit zufällig gewählten, anfangs veränderlichen Geschwindigkeiten bis zum anderen Ende der Schwimmbahn. Erreicht einer das Ende, dann freut er sich und stoppt durch eine Nachricht alle anderen Schwimmer (und sich selbst).

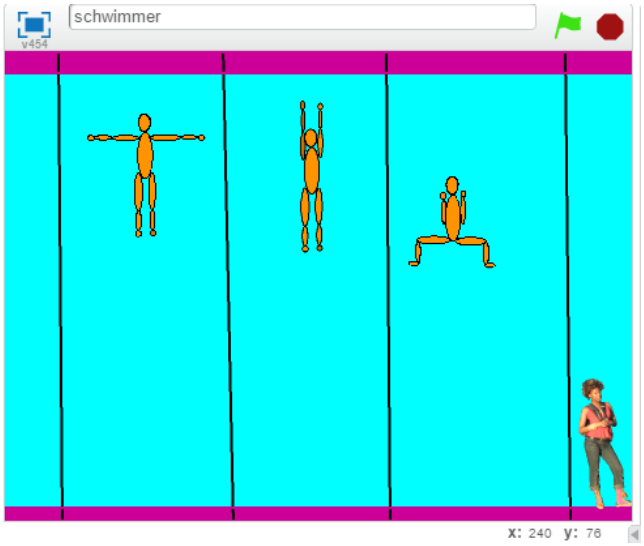
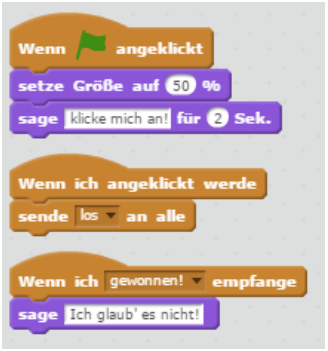


Bild 11: Schwimmer

Die Miniskripte der Trainerin sind trivial …



… und auch die Schwimmer haben nicht viel mehr zu tun, als zu schwimmen.

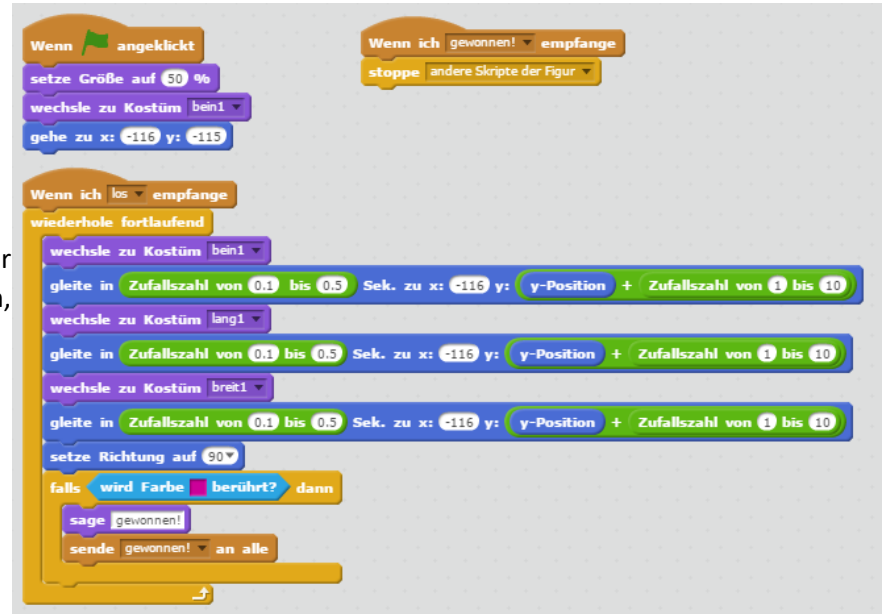


Bild 12: Skripte von Trainerin und Schwimmer

Auch dieses Beispiel dient nur dazu, die Unterschiede zwischen Botschaft und übermittelter Information deutlich zu machen. Es lässt sich allerdings leicht ausbauen, z. B. indem den Schwimmern feste Geschwindigkeiten zugewiesen werden (was ein neues Attribut, also lokale Variable erfordert) oder diese am Bahnende umkehren, um zum Startpunkt zurück zu schwimmen. Andere Schwimmstile sind leicht darstellbar, es lassen sich Erfolgsstatistiken führen, und auch der Anschlag am Ziel ist stark verbesserungswürdig. In welchem Zusammenhang die Siegesnachricht mit der Äußerung „Ich glaub‘ es nicht!“ steht, bleibt allerdings das Geheimnis der Beteiligten.

alternatives Beispiel: **Selbstportrait**

Altersstufe: *Sekundarstufe I*
Werkzeug: *Scratch2*
Material: *Selbstportrait.sb2*

Die Schülerinnen und Schüler stellen sich selbst vor: wo sie wohnen, ihren Schulweg, ihre Hobbies, …

In diesem Fall ist die Rollenverteilung noch viel klarer: das Informatiksystem stellt Bilder und Texte, also Daten, bereit. Deren Auswahl oblag der portraitierten Person, und deren Informationen sollen die Mitschüler/innen erreichen. Was die damit machen, ob sie z. B. Hunde mögen, ist offen.

Bild 13: Selbstportrait

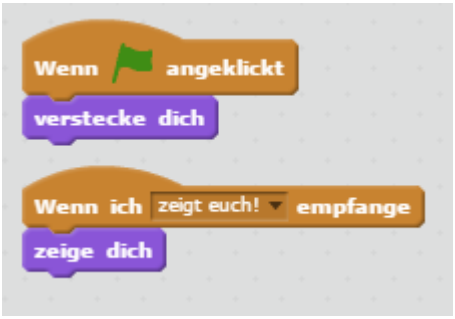


Bild 14: Skripte von Paula und den Hunden

alternatives Beispiel: **Im Bistro**

Altersstufe: *Sekundarstufe I/II*
Werkzeug: *BYOB*
Material: *Im Bistro.ypr*

Das Beispiel entspricht direkt dem des Gemüseladens, vielleicht für etwas ältere Schülerinnen und Schüler. Die Nutzung von BYOB eröffnet einige zusätzliche Möglichkeiten, z. B. bei der Animation der Sprites. Man könnte z. B. die Gliedmaßen gesteuert am gemeinsamen Objekt bewegen („attached parts“). Vor allem aber wird durch die Nutzung von BYOB schon für einfache Animationen der Wechsel des Werkzeugs überflüssig, wenn später komplexere Probleme bearbeitet werden. Und die Rolle der Körpersprache wird mehr als deutlich. Verlagert sich das Gespräch z. B. in den Bereich der Ironie, dann werden viele Äußerungen nur vor dem bildhaften Hintergrund richtig interpretierbar sein.

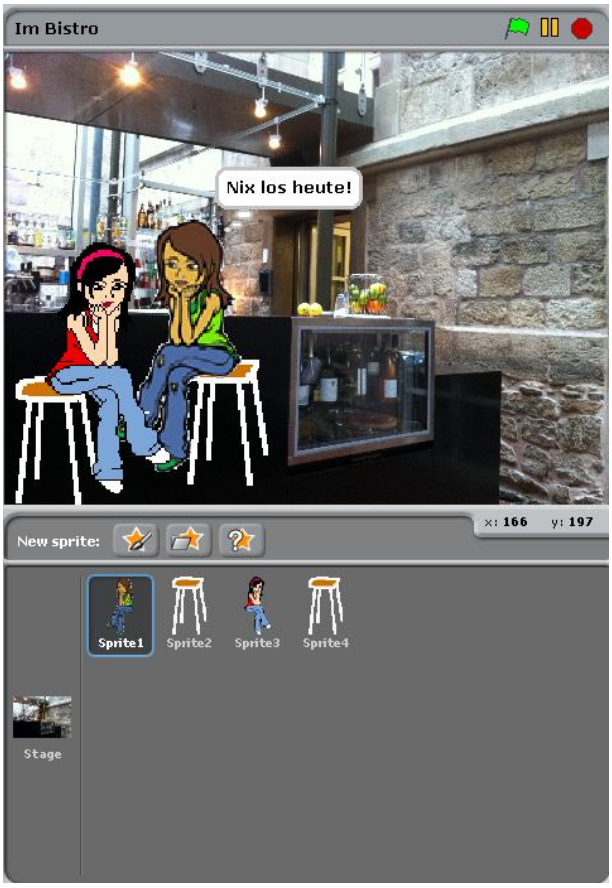


Bild 15: Im Bistro

alternatives Beispiel: **Searles chinesisches Zimmer**

Altersstufe: *Sekundarstufe II*
Werkzeug: *Snap!*
Material: *chinesisches Zimmer.xml*

Searls Beispiel dient zur Diskussion möglicher künstlicher Intelligenz. In einem Zimmer befindet sich eine Person, die kein Chinesisch kann, aber über ein Buch verfügt, das in ihrer Sprache Regeln zur Veränderung chinesischer Texte enthält. Sie bekommt chinesische Texte hereingereicht, wendet die Regeln an und gibt die Ergebnisse wieder nach außen.[14] Personen außerhalb des Zimmers glauben, dass die Person im Innern Chinesisch kann.

Das Beispiel ist angesichts der Diskussion über künstliche Intelligenz aktuell. Es ist aber natürlich auch ein exzellentes Beispiel zum Verhältnis von Information und Daten. Das „datenverarbeitende System“ im Innern versteht gar nichts, produziert aber Ergebnisse, die von den Benutzern als Erscheinungsformen von Intelligenz interpretiert werden.

Hinweis: Da ich leider kein Chinesisch kann, wurden im abgebildeten Beispiel die Texte von einem Programm übersetzt. Die Ergebnisse wurden dann in die Eingabefelder kopiert.

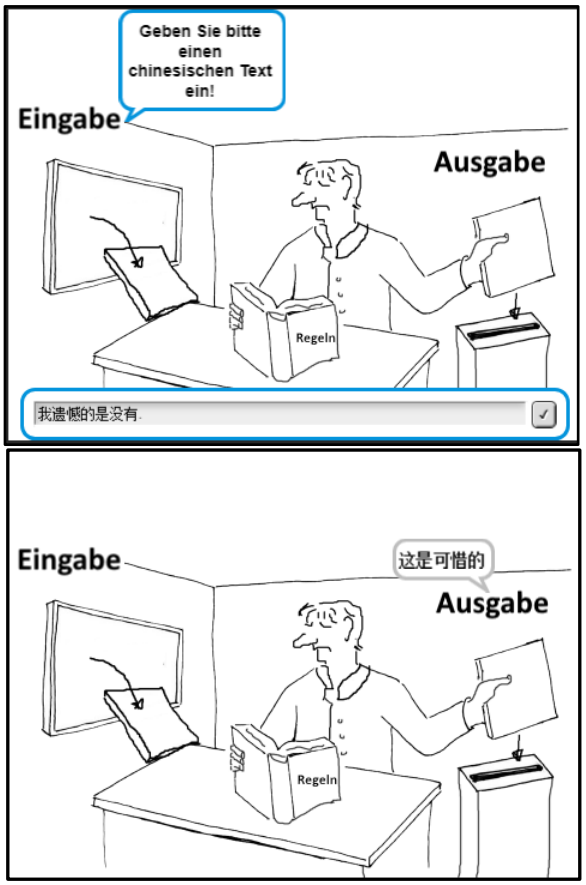


Bild 16: Das chinesische Zimmer

[14] https://de.wikipedia.org/wiki/Chinesisches_Zimmer

1.3.2 Zu Fall 2: Kommunikation mit offener Fragestellung

In diesem Szenario kommunizieren zwei menschliche Partner mithilfe eines Informatiksystems. Der eine stellt eine Frage, der andere hilft ihm bei der Antwort – hoffentlich nach bestem Wissen und Gewissen.

Beispiel: **Fernunterricht Astrophysik**

Altersstufe: *Sekundarstufe I/II*
Werkzeug: *BYOB*
Material: *Galaxien-Schueler.ypr, Galaxien-Lehrer.ypr*

Wir nutzen die Netzwerkfähigkeiten von BYOB aus und arbeiten mit zwei Instanzen des Programms, die sich auf verschiedenen Computern im gleichen Netzwerk befinden. Ein Schüler stellt eine Frage an den weit entfernten Astrophysiker, der liefert ihm Material[15], von dem er hofft, dass der Schüler sich die Antwort erschließen kann. In diesem Fall handelt es sich dabei um einige Galaxienbilder. Durch die Frage wird ein Kontext erzeugt, für den der Antwortende Daten zusammenstellt und übermittelt, von denen er glaubt, dass sich dem Fragenden daraus die gesuchten Informationen erschließen. Der Fragende interpretiert dann das Datenmaterial als Hilfestellung bezogen auf seine Frage – und nicht etwa als Deko-Material für den Klassenraum.

Die Partner kommunizieren wiederum über Botschaften und können sich Daten mithilfe globaler Variable *(<variablenname> sensor value*) oder (in diesem Fall) über „gesharete“ Sprites) übermitteln.

Bei etwas komplizierteren Fragen müssen die Daten natürlich zuerst ausgewertet, dargestellt und interpretiert werden, bevor entschieden wird, ob die ursprüngliche Frage sich damit beantworten lässt (s. nächstes Beispiel). Auch darüber kann die Kommunikation mit dem Lehrenden erfolgen.

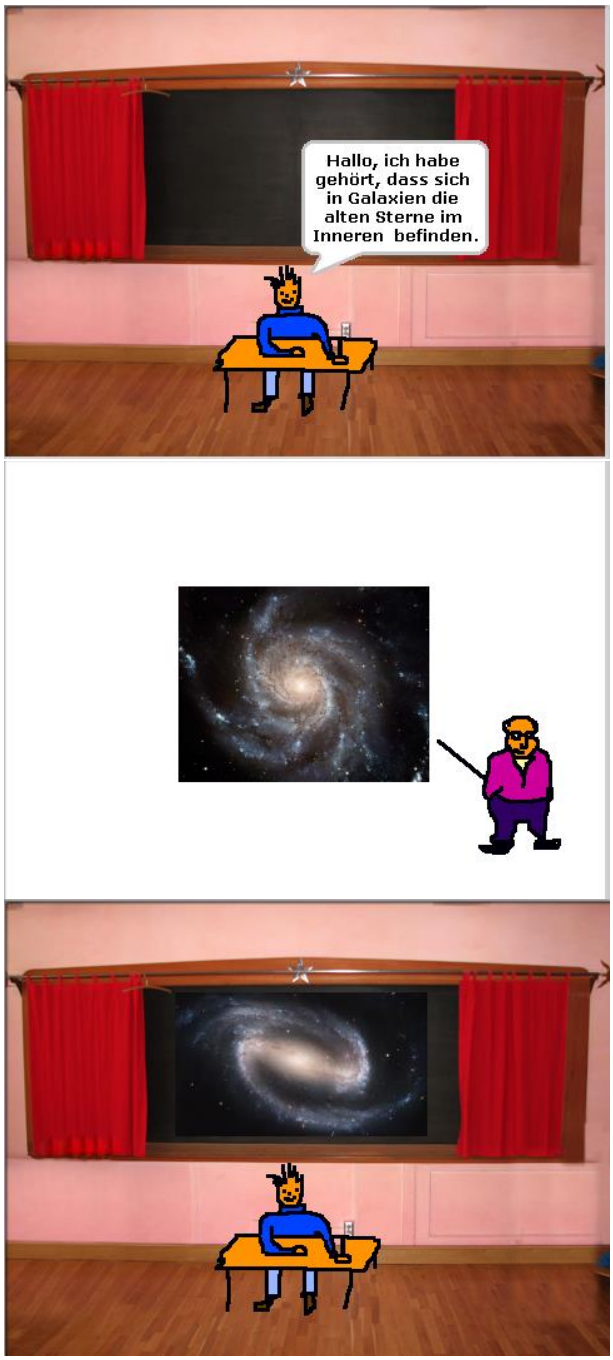


Bild 17: Fernunterricht in Astrophysik

[15] Aus https://en.wikipedia.org/wiki/Galaxy

Die Skripte der Beteiligten sind mehr als einfach, sie eignen sich für die ersten Stunden. Zu verstehen ist dagegen die Kommunikation zwischen entfernten Computern sowie der Datenaustausch.

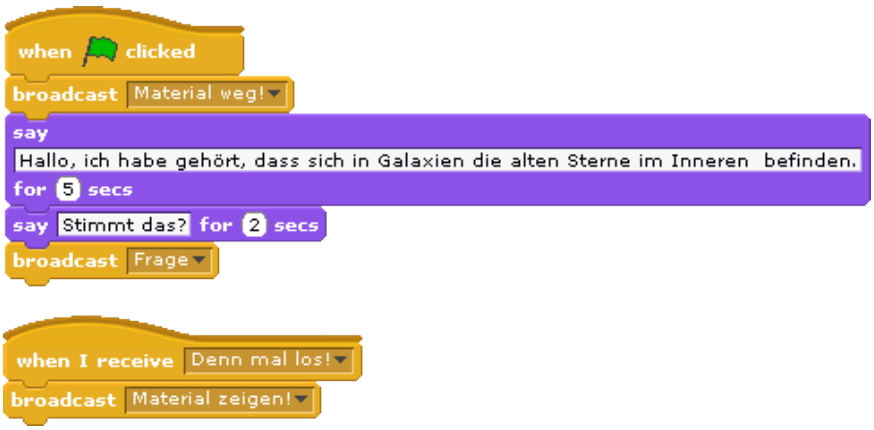


Bild 18: Skripte des Schülers in Astrophysik

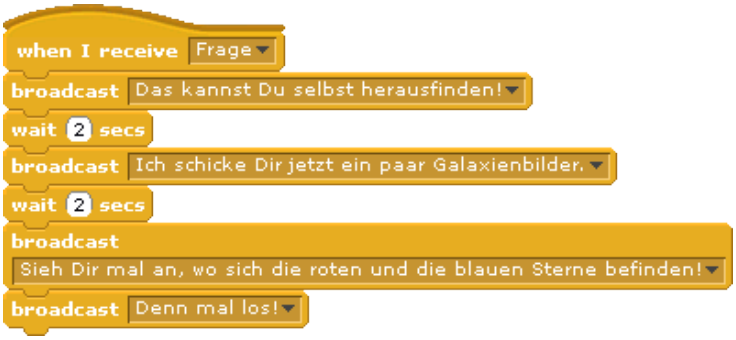


Bild 19: Skripte des Lehrers in Astrophysik

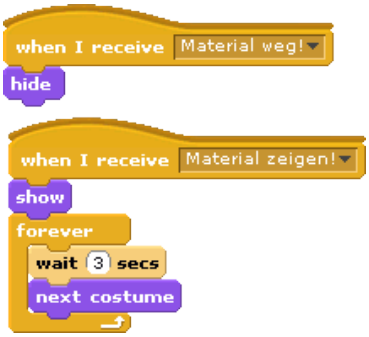


Bild 20: Skripte des Materials in Astrophysik

alternatives Beispiel: **Berechnung des Abstands der roten bzw. blauen Pixel vom Zentrum der Galaxis**

Altersstufe: *Sekundarstufe II*
Werkzeug: *GP*
Material: *Galaxien.gpp*

Wir wollen unser Astronomie-Beispiel fortsetzen und den Schüler in die Lage versetzen, die mittleren Abstände der roten bzw. blauen Pixel vom Zentrum der Galaxis zu messen. Dafür benötigen wir natürlich ein Werkzeug, dass einerseits überhaupt RGB-Werte lesen und wieder schreiben kann, und das ausreichend schnell ist, um die Daten eines ganzen Bildes zu verarbeiten. Mit GP („general purpose language“) ist ein entsprechendes Tool in der Entwicklung, aber noch nicht frei verfügbar.[16] Da dieses System noch weitgehend unbekannt ist, zeigen wir einmal den vollen Bildschirm, der Scratch, BYOB und Snap! sehr ähnelt. Gezeigt ist das Register zur Pixelbearbeitung.

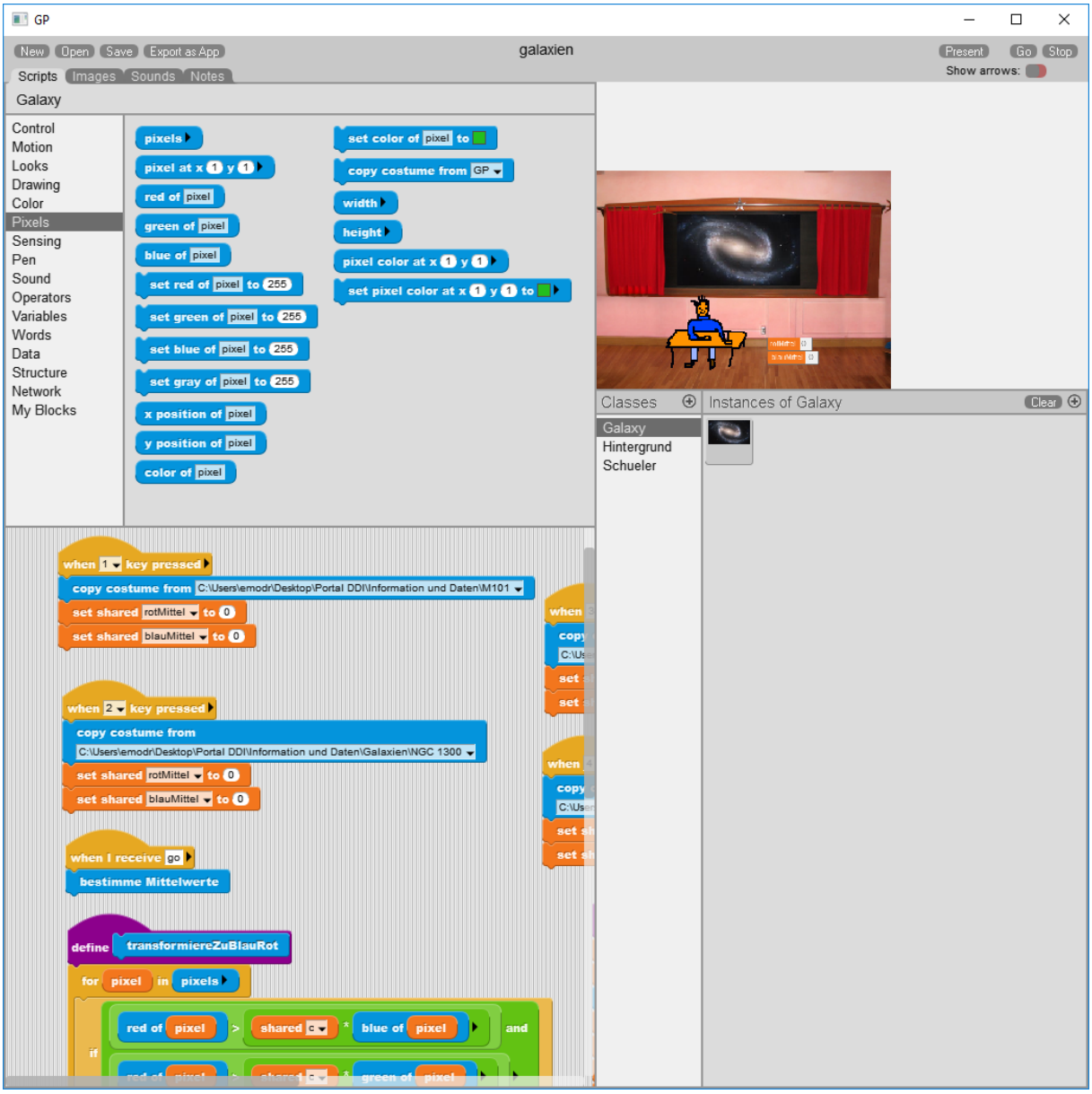
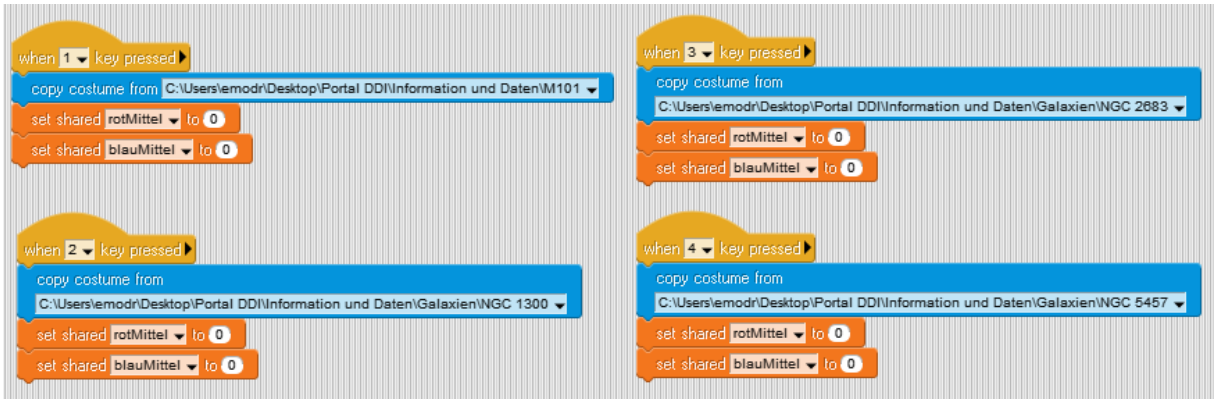


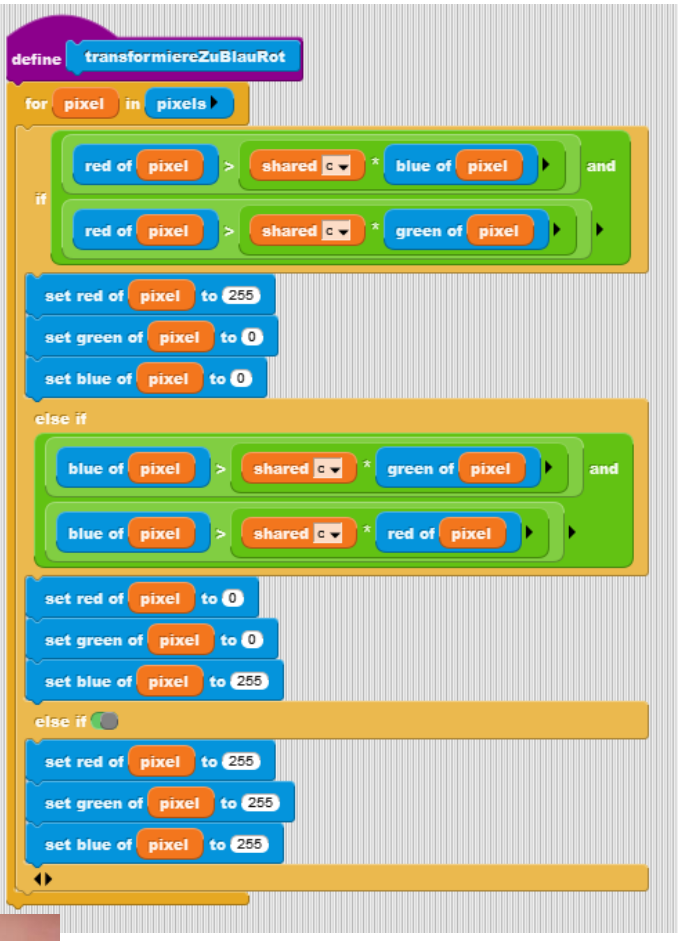
Bild 21: SScreenshot GP

[16] Wir benutzen hier die aktuelle (15.2.2017) Pre-alpha-Version 059a

Wir erzeugen drei Klassen namens *Galaxy*, *Hintergrund* und *Schueler* und laden die schon im vorigen Beispiel benutzten Bilder. Die Hauptarbeit erfolgt in der zugehörigen Instanz der Klasse *Galaxy*, die zumindest einmal unterschiedliche Galaxienbilder laden können muss. Dieses geschieht jeweils auf Tastendruck.



Anschließend wollen wir die „überwiegend roten bzw. blauen Pixel auf die vollen Farbwerte setzen; die anderen werden weiß. Der Begriff „überwiegend“ wird von einem Faktor *c* bestimmt, den der Schüler setzen kann, z. B. auf 1.5. Dieser wird als „shared c“ angezeigt, weil es sich in diesem Fall um eine globale, also geteilte Variable handelt.



Bilder 22, 23: Skripte des Galaxy-Objekts

Das Resultat ist ein stark reduziertes Farbbild.

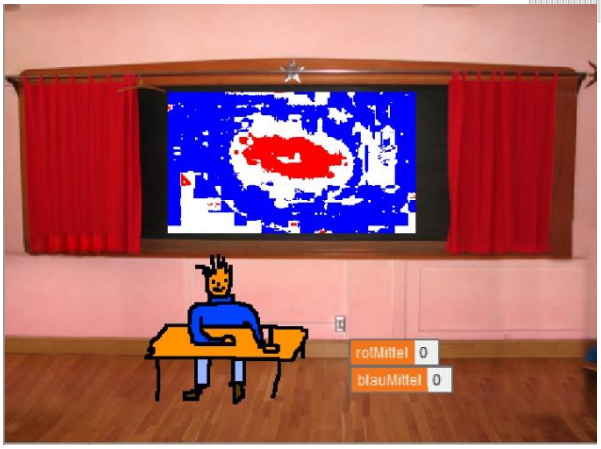


Bild 24: Bild nach der Transformation

In diesem können wir nun die Abstände der roten bzw. blauen Pixel zum Zentrum bestimmen und die jeweiligen Mittelwerte berechnen.

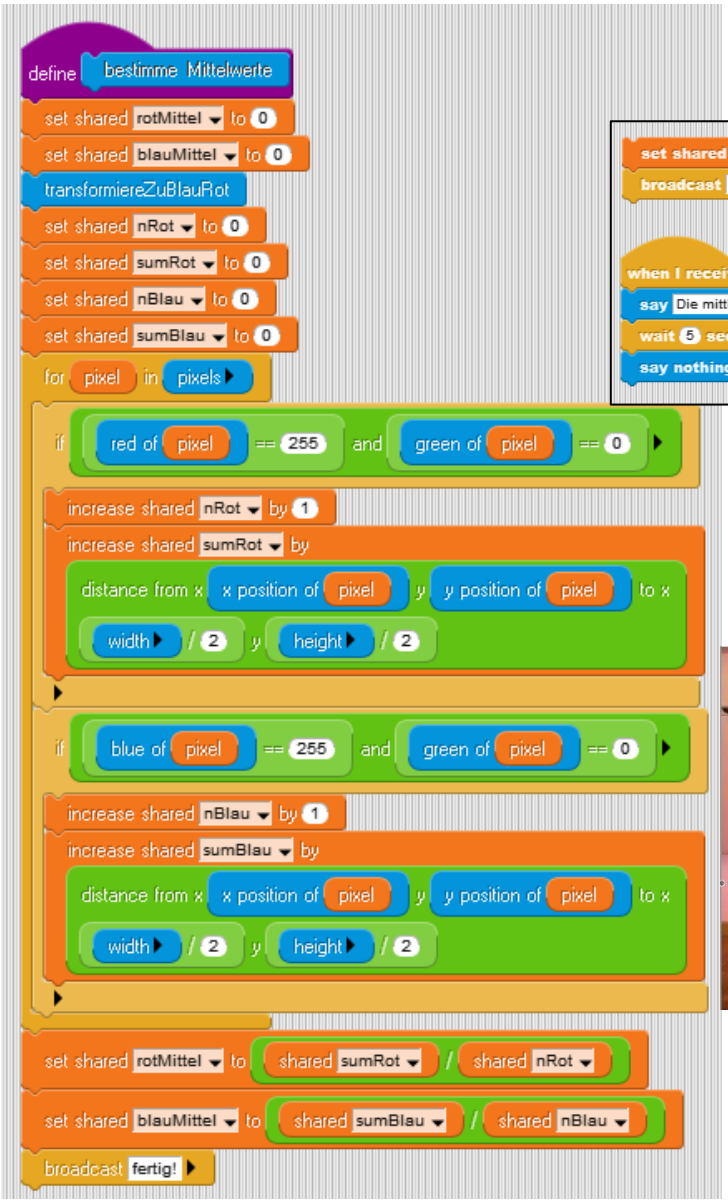
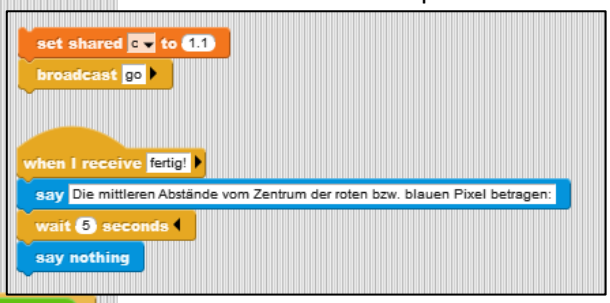


Bild 25: Berechnung der mittleren Abstände

Gesteuert wird der Ablauf wiederum über Botschaften. Der Schüler verfügt dazu über einfache Skripte.



… auf die das Galaxy-Objekt antwortet.

Bild 26: Ergebnisse

Soweit der informatische Teil. Erinnern wir uns daran, dass der Astronomielehrer die Bilder mit dem kleinen zusätzlichen Tipp als Antwort auf die Frage nach den alten Sternen geschickt hat, dann wurde diese Frage bisher nicht beantwortet. Der Schüler hat zwar zuerst per Augenschein, dann bestätigt durch ein kleines Programm festgestellt, dass im Innern der Galaxien mehr rot als blau strahlende Sterne zu finden sind - aber das wollte er doch gar nicht wissen. Er kann jetzt auf (mindestens) zwei Arten auf die Situation reagieren: entweder hält er den Lehrer für unfähig oder sich und seine Fragen nicht für ernst genommen und verlässt beleidigt das Fernunterrichtsprogramm, oder er vertraut dem Lehrer und schließt daraus, dass die alten Sterne die roten sind. Dieser zusätzliche Schluss hat aber nur teilweise mit den übermittelten Daten zu tun, Fakten dazu fehlen völlig, er ergibt sich wesentlich aus dem Kontext und der Situation der beteiligten Personen.

alternatives Beispiel: **Weizenbaums Eliza**[17]

Altersstufe: *Sekundarstufe II*
Werkzeug: *GP*
Material: *Eliza.gpp*

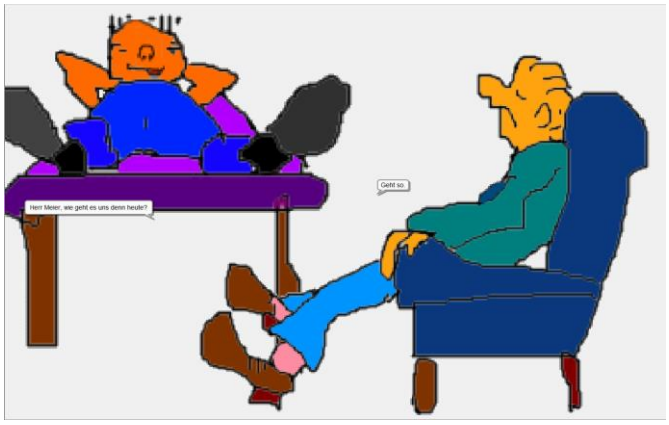


Bild 27: Psychiater und Patient im Gespräch

Das berühmte Beispiel beschreibt die Kommunikation zwischen Psychiater und Patient, wobei (in diesem Fall) beide zufallsgesteuert Platituden absondern. Die Koordination des „Gesprächs“ geschieht wieder durch entsprechende Botschaften.

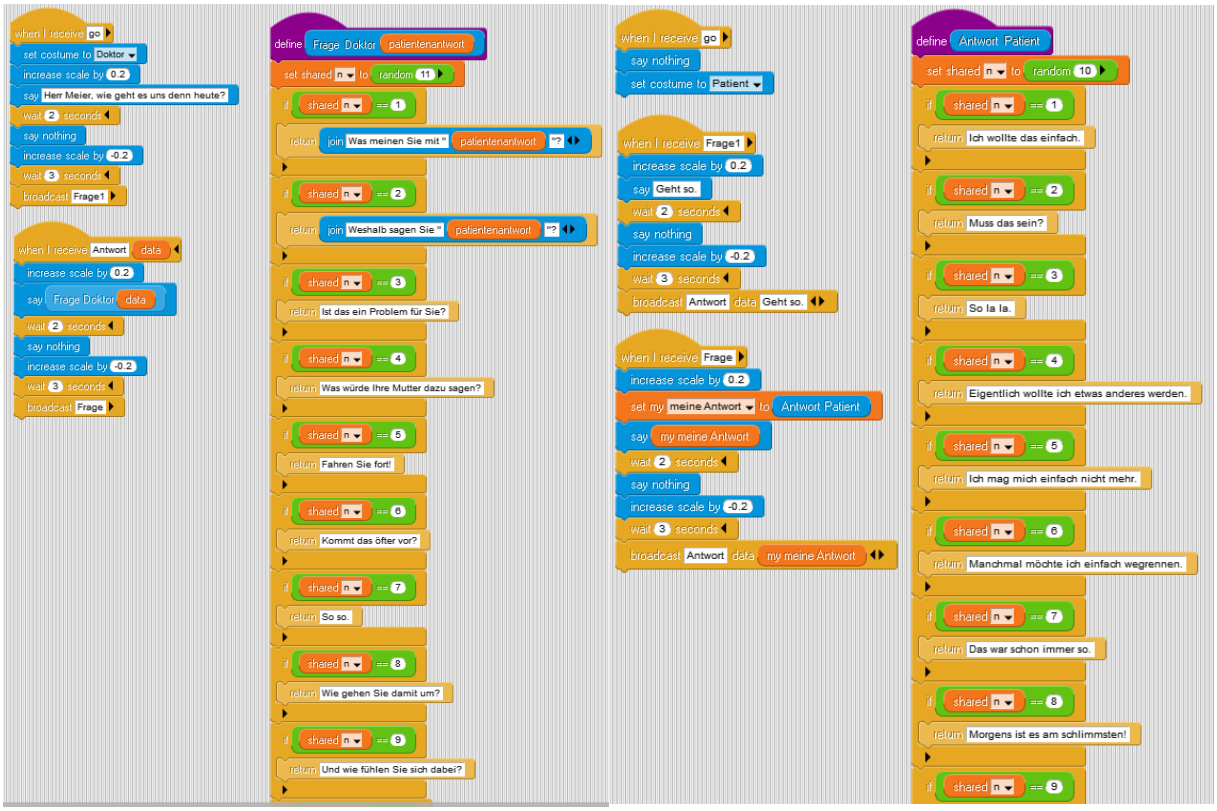


Bild 28: Skripte von Psychiater und Patient

Neben dem spielerischen Charakter des Beispiels ist auch der Informationsgehalt der Nachrichten interessant. Patient und Arzt reagieren inhaltlich überhaupt nicht aufeinander, sie senden aber Daten zum richtigen Zeitpunkt. Was ist denn nun die übermittelte Information? Wenn überhaupt etwas, dann erfährt der Patient, dass jemand da ist, mit dem er reden kann. Vielleicht hilft ihm das. Diese Information wird aber nicht durch die übermittelten Daten repräsentiert, sondern durch die Anwesenheit von Daten(müll). Die Daten selbst sind irrelevant.

[17] https://de.wikipedia.org/wiki/ELIZA

1.3.3 Zu Fall 3: Kommunikation mit eindeutiger Fragestellung

In diesem Szenario kommuniziert ein menschlicher Partner mit einem Informatiksystem. Will er angemessene Antworten haben, dann muss er seine Fragen entsprechend formulieren.

Beispiel: **Die Wissensgesellschaft**

Altersstufe:	*Sekundarstufe I/II*
Werkzeug:	*Browser*[18]
Material:	*Ergebnisse der Recherchen*

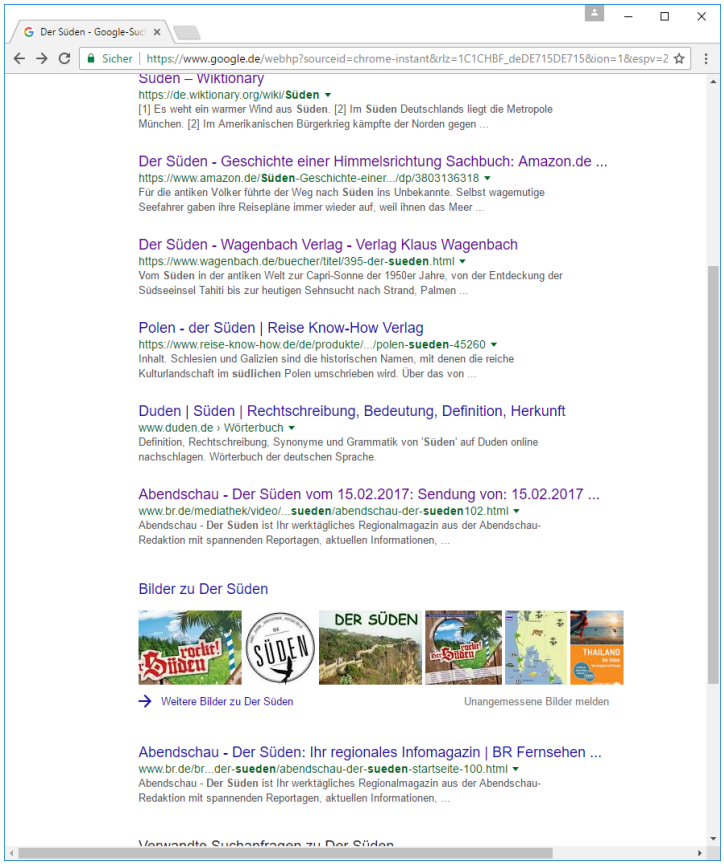


Bild 29: Suchergebnisse zu „Der Süden“

Man hört oft, dass es nicht mehr nötig sei, Wissen zu erwerben, weil sich dieses ja „im Netz“ befinde und jederzeit abgerufen werden könne.

Ist das so?

Wir gehen wie immer experimentell vor und versuchen, uns über einen nicht ganz trivialen Begriff zu informieren: den Süden. Die Antwort besteht aus den „relevantesten“ von 24 700 000 Ergebnissen. Geliefert werden die üblichen Wikipedia-Einträge zu dem Buch von Borges („El Sur“) und der Himmelsrichtung sowie Hinweise auf Reiseliteratur – und auf ein weiteres Buch zum Thema. Auf den nächsten 10 Seiten finden wir auch nicht viel mehr. Wir müssen also daraus schließen, dass „der Süden“ allein geografisch zu verstehen ist – oder wir müssen in den bitteren Apfel beißen und richtige Bücher lesen. Zwei davon scheint es nach Googles Meinung ja zu geben. Lehnen wir dieses üble Ansinnen ab, dann bleibt es beim geografischen Süden. Hinweise auf den Süden als Metapher, soziales oder ökonomisches Phänomen, narratives Element, Sehnsuchtsort, literarische Kategorie, Thema der bildenden Kunst usw. fehlen, und wir werden sie nicht einmal vermissen, es sei denn, wir wüssten, dass es sie gibt.

„Im Netz“ finden wir Fakten: die Einwohnerzahl von Hamburg oder das Bruttosozialprodukt von Burkina Faso, das Rezept für Frutti di Mare oder zur Reparatur des Staubsaugers. Aus diesen Fakten lassen sich Informationen gewinnen, wenn wir sie geeignet auswerten und einordnen. Doch worin „einordnen“? Infrage dafür kommt nur vorhandenes Wissen, im Kopf vorhandenes, und das muss erst einmal erarbeitet werden, bevor „das Netz“ angemessen benutzbar ist.

Fragen wir nach den Ergebnissen solcher Ordnungsprozesse, also den ausgewerteten Daten, dann erhalten wir natürlich auch Antworten: die Meinungen anderer. Diese können wir aber nur bewerten, also selbst wieder einordnen, wenn wir über entsprechende Fähigkeiten verfügen (s. o.). Fehlen diese, dann bleiben andere Bewertungskriterien: dass wir den Meinungsbildnern glauben (oder nicht), dass sie uns sympathisch sind (oder nicht), dass sie sind wie wir (oder nicht), dass andere ihnen glauben (oder nicht) … – wenn wir glauben, dass die anderen die sind, die sie angeben zu sein (oder nicht). Mit Rationalität hat das wenig zu tun.

[18] Die abgebildeten Screenshots stammen aus Google Chrome vom 16.2.2017

Wenn wir wissen, dass es noch andere Möglichkeiten gibt, unsere Fragen zu beantworten, als die zuerst von der Suchmaschine gelieferten, dann sind wir fein raus. Erweitern wir unsere Suche nach „dem Süden“ um den Begriff „Metapher“, dann erhalten wir ein völlig anderes Spektrum von Antworten – und es gibt auch nur noch 189 000 Ergebnisse. Welche Verarmung! Selbst der Süden als Sehnsuchtsraum ergibt 1 110 Antworten, die fast nichts mit den vorherigen gemein haben. Erst die Kombination mit der bildenden Kunst liefert wieder 352 000 Treffer. Präzisieren wir unsere Anfrage, indem wir erweiterte Einstellungsmöglichkeiten benutzen oder schon wissen, wie man z. B. Begriffe ausschließt, dann kommen die Suchergebnisse langsam dem näher, was wir von ihnen erwarteten. Auch hier liefert vorhandenes Wissen den Zugang zu neuem.

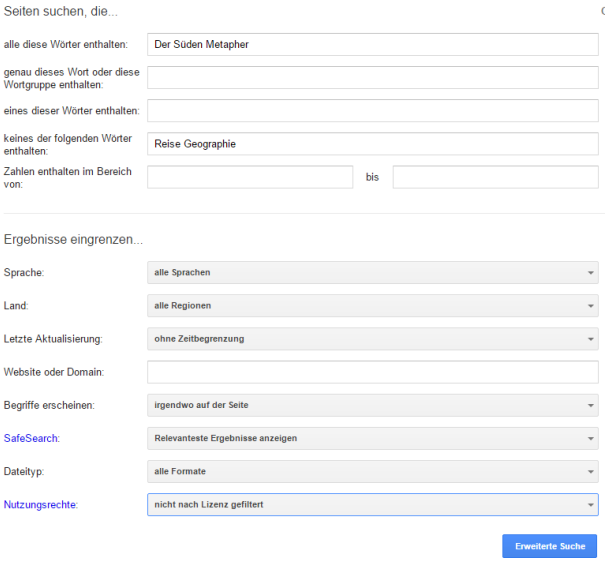


Bild 30: erweiterte Sucheinstellungen

Suchmaschinen sind ja nicht gehässig, sie arbeiten nur „wie vorgegeben“. Stellen wir präzise Fragen, dann liefern sie meist auch präzise Antworten. Stellen wir keine präzisen Fragen, dann benötigen sie Zusatzkriterien, um „die besten“ Antworten zu finden. Bei diesen Kriterien kann es sich um bezahlten Platzierungen handeln, meist aber um die „Bewertung“ der Antworten durch andere Benutzer, die gleiche oder ähnliche Suchanfragen gestellt haben. Die Bewertung findet bekanntlich in Form eines Klicks auf die Antwortzeile statt.

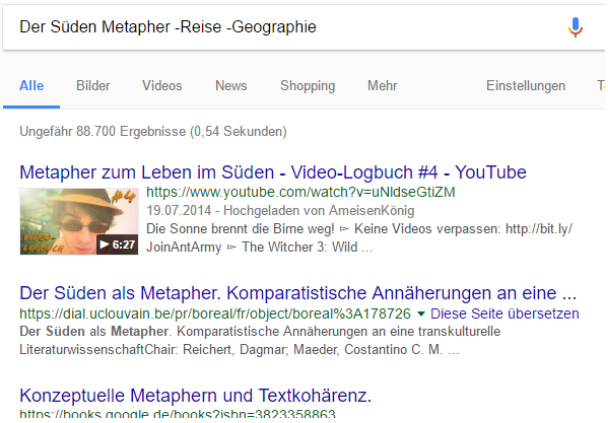


Bild 31: präzisere Suchergebnisse zu „Der Süden“

Dieses Verhalten hat Konsequenzen. Niemand kann die anfangs genannten 24 700 000 Antworten durchforsten, und selbst die 1 110 Treffer des Sehnsuchtsraums sind fast unüberschaubar. Also werden fast alle Klicks auf den ersten ein, zwei Seiten der Suchergebnisse erfolgen – und damit haben wir einen selbstverstärkenden Prozess: die meist angeklickten Seiten werden wieder am meisten angeklickt, womit sich ihre Platzierung weiter festigt. Die anderen sind zwar vorhanden, aber praktisch unsichtbar. „Im Netz“ erscheinen den Benutzern dauerhaft nur Seiten, die inhaltlich denen entsprechen, die ihnen anfangs angeboten wurden. Neue werden kaum dazukommen. Wurden z. B. die ersten Suchanfragen vom Anbieter gefiltert, ihm die Ergebnisse „ähnlicher“ Benutzer geliefert, die sich aufgrund seiner bisherigen Nutzung des Systems (oder zunehmend „des Netzes“ als Ganzem) leicht finden lassen, dann wird der Benutzer diesen „Informationsraum“ kaum wieder verlassen. Er sieht einfach nichts Anderes. Seitens des Anbieters ist dieses Verhalten verständlich, denn der möchte Suchergebnisse liefern, die mit hoher Wahrscheinlichkeit angeklickt werden - nur dann wird er bezahlt. Aber die so entstehenden „Echokammern“ sind z. B. politisch brandgefährlich, weil sie die Gesellschaft in disjunkte Gruppen spalten, die kaum noch diskursfähig sind, aber auch das Spektrum der sonst gelieferten „Informationen“ verarmen.

Das Resultat unserer Überlegungen ist ziemlich eindeutig: „Das Netz“ enthält kein Wissen, sondern Daten. Diese können unser Wissen bereichern, wenn wir über das Wissen verfügen, sie angemessen zu nutzen, sie zu bewerten, sie einzuordnen oder zu verwerfen. Eine entsprechende Bildung bildet die Basis für wunderbare neue Möglichkeiten. Fehlt sie, dann werden wir zu manipulierbaren Objekten.

alternatives Beispiel: **Zugriff auf Datenbanken**

Altersstufe: *Sekundarstufe I/II*
Werkzeug: *sqlSnap!*[19]
Material: *sqlBeispiel.xml*

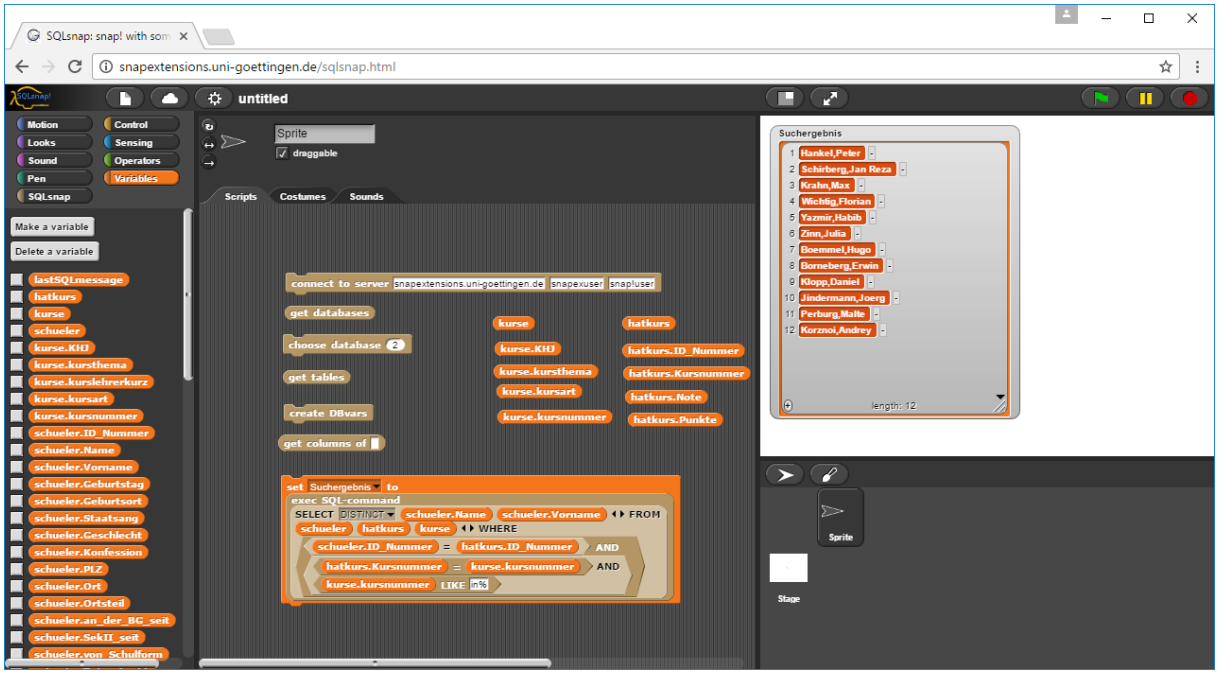


Bild 32: Screenshot „sqlSnap!“

Wir benutzen eine Variante von Snap! (sqlSnap!), die u. a. um die Möglichkeit ergänzt wurde, auf Datenbanken zuzugreifen. Damit können wir die üblichen Datenbankanfragen aus den zugehörigen Relationen, Attributen usw. mithilfe von Blöcken zusammensetzen und ausführen lassen. Das Ergebnis ist jeweils eine Liste. Im gezeigten Beispiel erhalten wir die Teilnehmenden an Informatik-Grundkursen. Ganz einfach.

Warum ist das so einfach – und für wen? Der Benutzer muss die SQL-Syntax kennen und sich an sie halten. Vor allem aber muss er die gewünschten Daten völlig eindeutig beschreiben, es darf keine Interpretationsmöglichkeit geben. Das ist gar nicht so einfach. Die Auswertung solcher Anfragen ist aber dann für die Maschinen einfach, das können sie schon seit Jahren.

Unser Benutzer beschreibt mit seiner Anfrage die Daten, die er vom System erhalten möchte. Er weiß zwar meist nicht, welche Daten genau er dann erhalten wird, er kennt aber ihre Bedeutung. ER kennt sie, nicht die Maschine. Die kann sie gar nicht kennen, weil sie nicht wissen kann, welche Bedeutung der Benutzer diesen Daten gibt, welche Information er ihnen entnehmen wird. Die Liste unser Kursteilnehmer z. B. kann viel bedeuten: vielleicht müssen sie wegen einer Exkursion vom restlichen Unterricht des Tages befreit werden, vielleicht wird nachgesehen, welche kleinen Kurse man streichen kann, vielleicht wird überprüft, ob die Bücher für den Kurs reichen. Man weiß es nicht …

SQL-Anfragen sind für Maschinen einfach auswertbar, weil sie eine klare Entscheidungsgrundlage liefern: Daten gehören entweder zur angeforderten Kategorie – oder nicht. Interessant wird es, wenn Fragen gestellt sind, die keine eindeutigen Entscheidungshilfen liefern. Sollen solche unscharfen Fragestellungen wie z. B. „Werden die Jungen (oder die Mädchen) in der Schule benachteiligt?“, „Haben es Stadtkinder (Landkinder, Kinder aus bürgerlichen Elternhäusern, Mitgliedern in Sportvereinen, Kin-

[19] snapextensions.uni-goettingen.de

dergartenkinder, Kinder mit Migrationshintergrund, …) in der Schule leichter (schwerer)?“ „Ist die Bewertung unterschiedlicher Kompetenzen in der Schule gerecht?“ mithilfe von (z. B.) SQL-Anfragen beantwortet werden, dann muss zwangsläufig eine Umformulierung und damit eine Interpretation erfolgen, die zu Antworten führt, bei denen zumindest fraglich ist, ob sie die ursprüngliche Frage beantworten oder eben nur die Interpretation. Eine Antwort gibt es immer, wenn eine Anfrage formuliert wurde, auch dann, wenn die ursprüngliche Frage aufgrund der Daten eigentlich nicht zu beantworten ist.

alternatives Beispiel: **Zugriff auf JSON[20]-Daten**

Altersstufe: *Sekundarstufe II*
Werkzeug: *GP*
Material: *JSONconverter.gpp*
stations.json (enthält die aktuellen Daten von Fahrrad-Entleihstationen in New York)
stationsshort.json (gekürzte Version der Datei)

Im Projekt sollen aktuelle und frei zugängliche Daten, die als JSON-Dateien gespeichert wurden, ausgewertet werden. Dazu müssen die Lernenden erst einmal recherchieren, um was es sich bei JSON handelt, welche Struktur die Dateien haben, welche Größen in ihnen darstellbar sind. Als Beispiel wählen wir die aktuellen Daten der Fahrrad-Entleihstationen in New York, die in einer Datei *stations.json* vorliegen.[21] (Die Aktualisierung dieser Daten ist ein gesondertes Thema, uns aus Kostengründen beschäftigen wir unseren Mitarbeiter aus dem Chinesischen Zimmer zeitweise in diesem Projekt.) Als Ziel der Transformation dient in diesem Fall eine Struktur, die entweder eine atomare Größe (Wahrheitswert, Zahl, Zeichenkette, …) oder eine Liste enthält, die aus atomaren Größen und/oder Teillisten besteht, die als ersten Eintrag den Typ der originalen Daten (Liste oder Dictionary) enthalten. In Dictionaries folgen als weitere Elemente zweielementige Listen mit Schlüssel/Wert-Paaren. Im Beispiel wird aus den Daten eine Tabelle erzeugt, die nur die Spalten *Stationsname*, *Status* und *verfügbare Fahrräder* enthält (Bild 33).

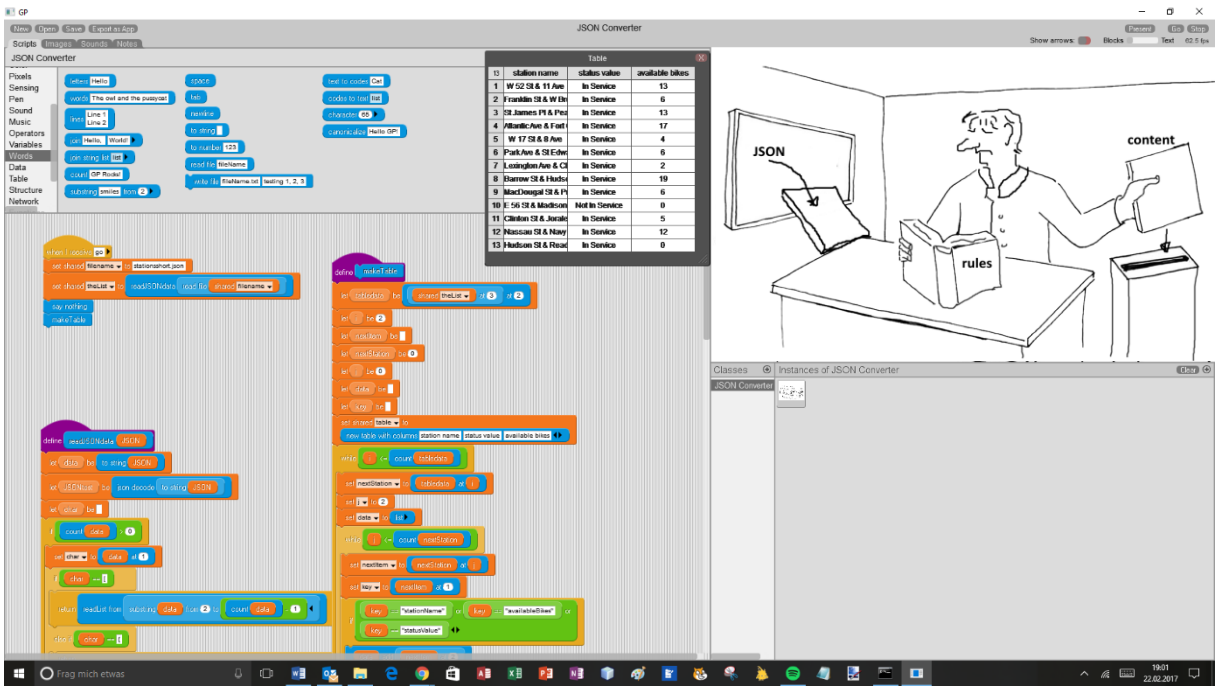


Bild 33: Screenshot „GP“

[20] JavaScript Object Notation
[21] https://catalog.data.gov/dataset/citi-bike-live-station-feed-json-d1c27

Wir wollen an diesem Beispiel eine Besonderheit von GP zeigen, die vielleicht einige Irritationen bzgl. des Verhältnisses von textbasierten und grafischen Sprachen beseitigt: GP kennt beide Darstellungen. Dafür wird derselbe Ausschnitt aus der Skriptebene einmal in grafischer und danach in Textform gezeigt.

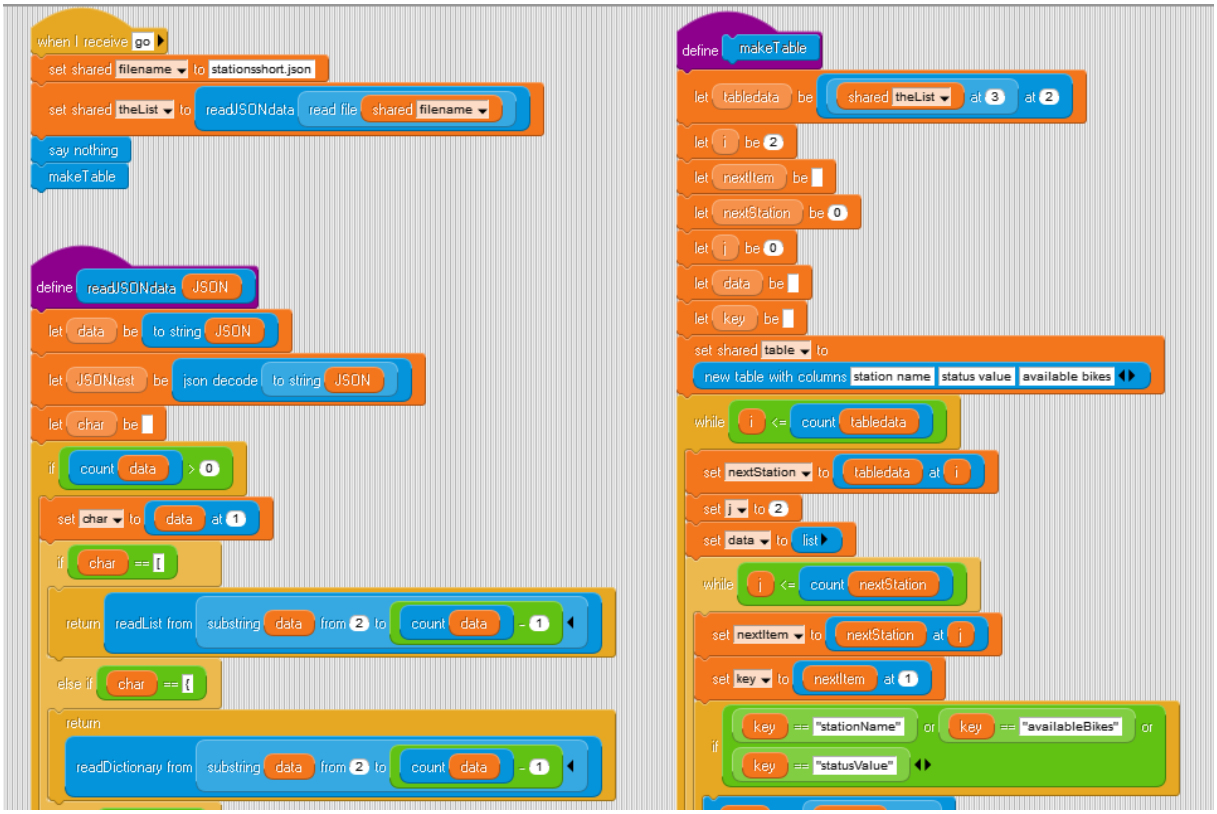


Bild 34: GP-Skripte in grafischer Darstellung

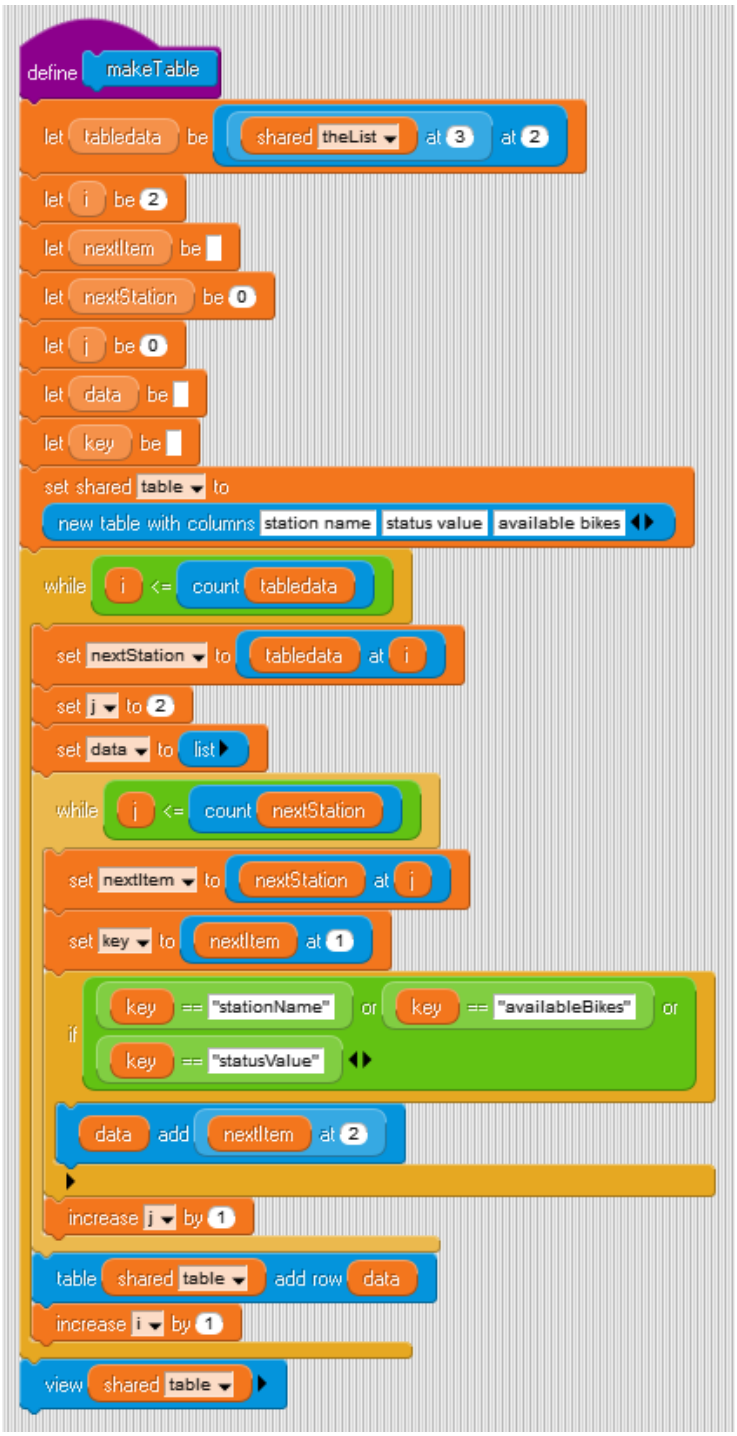
```
when I receive go
set shared filename to stationsshort.json
set shared theList to (readJSONdata (read file (shared filename)))
say nothing
makeTable

define (readJSONdata (JSON))
let (data) be (to string (JSON))
let (JSONtest) be (json decode (to string (JSON)))
let (char) be
if ((count (data)) > 0)
 set char to ((data) at 1)
 if ((char) == {)
  return (readList from (substring (data) from 2 to ((count (data)) - 1)))
 else if ((char) == {)
  return
  (readDictionary from (substring (data) from 2 to ((count (data)) - 1)))

define makeTable
let (tabledata) be (((shared theList) at 3) at 2)
let (i) be 2
let (nextItem) be
let (nextStation) be 0
let (j) be 0
let (data) be
let (key) be
set shared table to
(new table with columns station name status value available bikes)
while ((i) <= (count (tabledata)))
 set nextStation to ((tabledata) at (i))
 set j to 2
 set data to (list)
 while ((j) <= (count (nextStation)))
  set nextItem to ((nextStation) at (j))
  set key to ((nextItem) at 1)
  if
   (((key) == "stationName") or ((key) == "availableBikes") or ((key) == "statusValue"))
   (data) add ((nextItem) at 2)
  increase j by 1
 table (shared table) add row (data)
 increase i by 1
view (shared table)
```

Bild 35: GP-Skripte in textueller Darstellung

Liegen die Daten in Listenform vor, dann kann ihr relevanter Teil leicht extrahiert und ausgewertet werden – bloß, was ist hier „relevant“. Das hängt natürlich davon ab, was mit den Daten geschehen soll, welche Informationen gesucht sind. Interessiert uns die Zahl der verfügbaren Räder im Verlauf der Woche, dann ergibt sich eine andere Auswertung als beim Wunsch, die Verteilung der Räder über die Stadt in einem Stadtplan darzustellen. Und vielleicht suchen wir ja auch nur ein freies Rad in der Nähe des Hotels. In unserem Fall wollen wir einfach eine Tabelle der Stationen mit den verfügbaren Rädern anzeigen:



Eine neue eigene Methode ohne Parameter definieren.

Dazu einige lokale Skriptvariable vereinbaren und initialisieren.

Eine neue Tabelle mit drei Spalten vereinbaren.

Alle Stationen durchsuchen.

Eine Station auswählen, …

… die Liste der relevanten Daten leeren, …

… alle Einträge durchlaufen, …

… und die Daten mit den „interessanten“ Schlüsseln sammeln.

Die neue Tabellenzeile einfügen.

Die Tabelle anzeigen.

Bild 36: GP-Skripte zur Erzeugung einer Tabelle

1.3.4 Zu Fall 4: Kommunikation ohne menschliche Partner

Für dieses Szenario benötigen wir Beispiele, in denen Daten von einem System erfasst, an ein anderes, das durchaus im gleichen Rechner laufen kann, übermittelt und dort ausgewertet werden. Die Resultate dieser Auswertung werden dann diskutiert.

Beispiel: **Nummernschilderkennung**

Altersstufe: *Sekundarstufe I/II*
Werkzeug: *GP*
Material: *Nummernschildleser.gpp*

Bild 37: Nummernschildbild aus dem Netz

Wir wollen uns mit dem weiten Feld der Zeichenerkennung beschäftigen, d. h. aus einem Bild Texte entnehmen. Als Beispiel wählen wir die Nummernschilderkennung, wie sie z. B. in den Mautbarrieren an Autobahnen praktiziert wird. Wir wählen hier den einfachen, für die obere Mittelstufe geeigneten Fall, dass wir uns nur für die Nationalitäten der Fahrzeuge interessieren.[22] Damit müssen wir nur die Zeichen im blauen Bereich des Europa-Nummernschilds erkennen. Bilder für solche Aufgaben lassen sich im Internet generieren.[23]

Wir wählen einen sehr einfachen Ansatz und hoffen, dass sich die Anzahlen der Pixel zur Darstellung dieser Zeichen unterscheiden. Damit reduziert sich das Problem auf die Aufgabe, den blauen Bereich zu finden und in diesem die nichtblauen Pixel zu zählen. Wollen wir unabhängig von der Größe der Darstellung sein, dann können wir die Pixel im oberen Bereich („Euro-Sterne“) mit denen im unteren vergleichen.

Zuerst einmal generieren wir einige Nummernschilder unterschiedlicher Nationalitäten, importieren diese nach GP und schreiben einige Methoden, die die Teilaufgaben lösen, z. B.:

Bild 38: „deutsches“ Nummernschild untersuchen

Bild 39: Monitore für zwei Variable und Nummernschild in GP

Wir müssen zumindest den blauen Bereich im Kennzeichen finden. Ähnliches haben wir z. B. bei den Galaxienbilder schon gemacht. Bei den RGB-Grenzwerten muss man etwas experimentieren, dann klappt es gut.

Bild 41: blauer Anteil im Bild gefunden

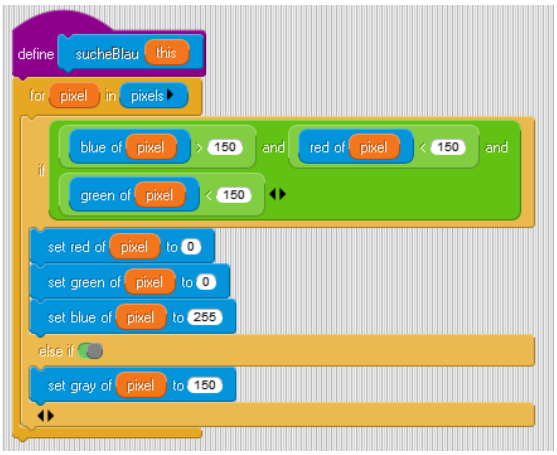


Bild 40: blauen Anteil im Bild suchen

[22] Ein ausführlicheres Beispiel finden wir auf http://snapextensions.uni-goettingen.de/beispielsupermarkt.pdf. Dort werden neben der Zeichenerkennung auch einfache Ansätze zur Gesichtserkennung usw. realisiert.
[23] Man kann z. B. einfach unter dem Stichwort „Nummernschild“ suchen.

Die Grenzen des blauen Bereichs lassen sich leicht finden, wenn wir ihn von links bzw. rechts, oben und unten beginnend durchsuchen. Zur Verdeutlichung markieren wir die untersuchten Pixel rot. Das entsprechende Skript ist einfach, aber (in dieser Form) lang.

Bild 42: Bild mit „Suchlinien“

In diesem Bereich können wir jetzt die inneren nichtblauen Pixel zählen, hier getrennt nach oberem und unterem Bereich.

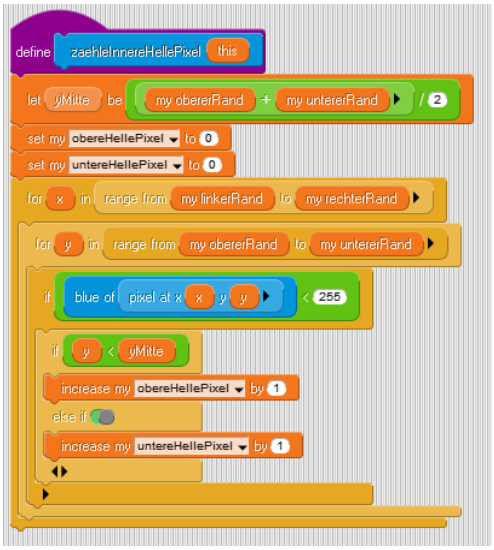


Bild 44: die „inneren“ Pixel zählen

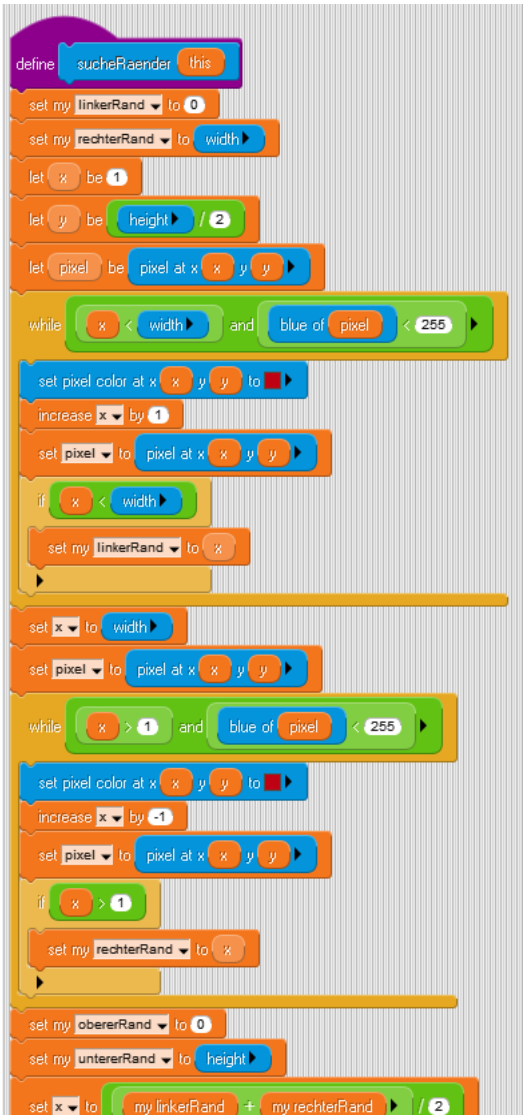


Bild 43: Skript zur Ermittlung der Bereichsgrenzen

Mit diesen Ergebnissen können wir jetzt einerseits untersuchen, ob der anfängliche Lösungsansatz überhaupt sinnvoll war, und wenn, aus welchem Land das untersuchte Kennzeichen stammt.

Bild 46: Ergebnisse

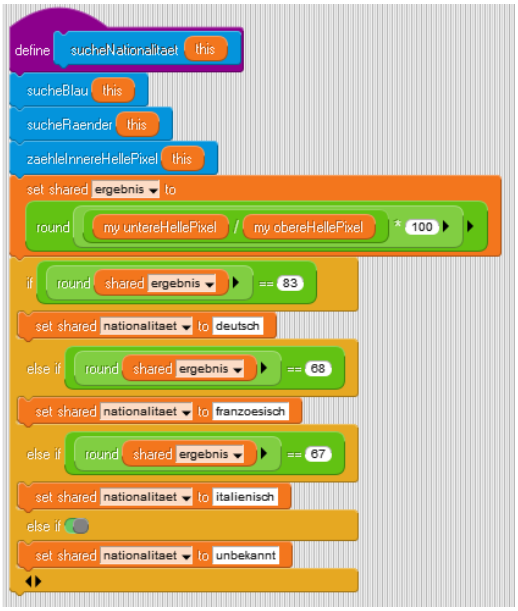


Bild 45: Nationalitätensuche

Soweit zum „technischen“ Teil. Wir können uns jetzt leicht vorstellen, dass sich auch der restliche Teil eines Kennzeichens mit etwas Aufwand bestimmen lässt. Das Ergebnis dieses Prozesses wird dann an eine andere Stelle übermittelt und dort ausgewertet. Dabei kann es sich um Mautstellen, Polizeicomputer, … handeln. Wir wollen zuerst den ziemlich „unkritischen“ Fall einer Mautstelle behandeln.

Unser Kennzeichen-Lesegerät liest also Nummernschilder und übermittelt das Ergebnis an die Zentrale – als Daten, z. B. „ABC-DE 123“. Diese Daten werden dort vom laufenden Programm ausgewertet, und zwar so, als ob es sich um Informationen handele. In diesem Fall wird z. B. angenommen, dass sich das Fahrzeug mit dem angegebenen Kennzeichen auf der Brennerautobahn befindet. Wenn die entsprechenden Voraussetzungen vorliegen, dann kann die Maut vom Konto des PKW-Halters abgebucht werden. Was passiert, wenn dieser der Abbuchung widerspricht, weil er angeblich gar nicht über den Brenner gefahren ist, sondern am Kochelsee gebadet hat? Menschliche Zeugen für beides gibt es nicht, nur „der Computer“ meint, das Kennzeichen auf einem Bild identifiziert zu haben. Ist dieser Fall justiziabel? Vermutlich so nicht, weil Computer als Zeugen nicht anerkannt sind. Wahrscheinlich wird in diesem Fall auch das Originalbild, das der Computer ausgewertet hat, gespeichert worden sein, sodass menschliche Experten überprüfen können, ob sich die Maschine geirrt hat. Es lassen sich aber leicht Szenarien angeben, wo diese Überprüfung nicht erfolgt oder auch gar nicht möglich ist, z. B. weil der Betroffene nichts von seiner „Identifizierung“ erfährt. Als Beispiel mögen die Bewegungsdaten eines Handys dienen, für die es sehr unterschiedliche Interessenten gibt.

Die von der Bildauswertung übermittelten Daten haben also auch in diesem Fall wenig mit den Informationen zu tun, die aus ihnen abgeleitet werden. Wird diese Interpretation an Maschinen ausgelagert, dann kommen wir schnell zu Szenarien, die im Bereich „Informatik und Gesellschaft“ anzusiedeln sind.

Alternatives Beispiel: **Streaming**

Altersstufe:	*Sekundarstufe I/II*
Werkzeug:	*BYOB*
Material:	*streaming-user.ypr*
	streaming-server.ypr

Bild 47: Streaming auf der Client-Seite

Wir liefern einen ersten Ansatz für zwei BYOB-Instanzen, die über die Mesh-Funktion als Server und Client gekoppelt sind. Auf der Server-Seite wird eine Liste von Kundendaten verwaltet, die das Einloggen ermöglicht und eine (im Beispiel noch nicht realisierte) Abrechnung durchführt, die von der Nutzungsdauer abhängt. Ist das Nutzerkonto leer, dann wir die Verbindung abgeschaltet. Auf der Client-Seite befinden sich Susi und ihr Laptop. Dieser baut die Verbindung auf, wenn der Einschaltknopf gedrückt wird, und beendet sie wieder beim zweiten Klicken.

Die Schülerinnen und Schüler müssen natürlich das Abrechnungssystem auf der Serverseite erst einmal einrichten. Ihr Hauptproblem sollte aber sein, eine sichere Verbindung zwischen Server und Client herzustellen, auf der die übermittelten Daten nicht mehr mitgelesen werden können. Da es dafür sehr unterschiedliche Lösungen gibt, handelt es sich um eine stark differenzierende Aufgabe.

Bild 48: Skripte von Susi, dem Einschaltknopf und dem Laptop

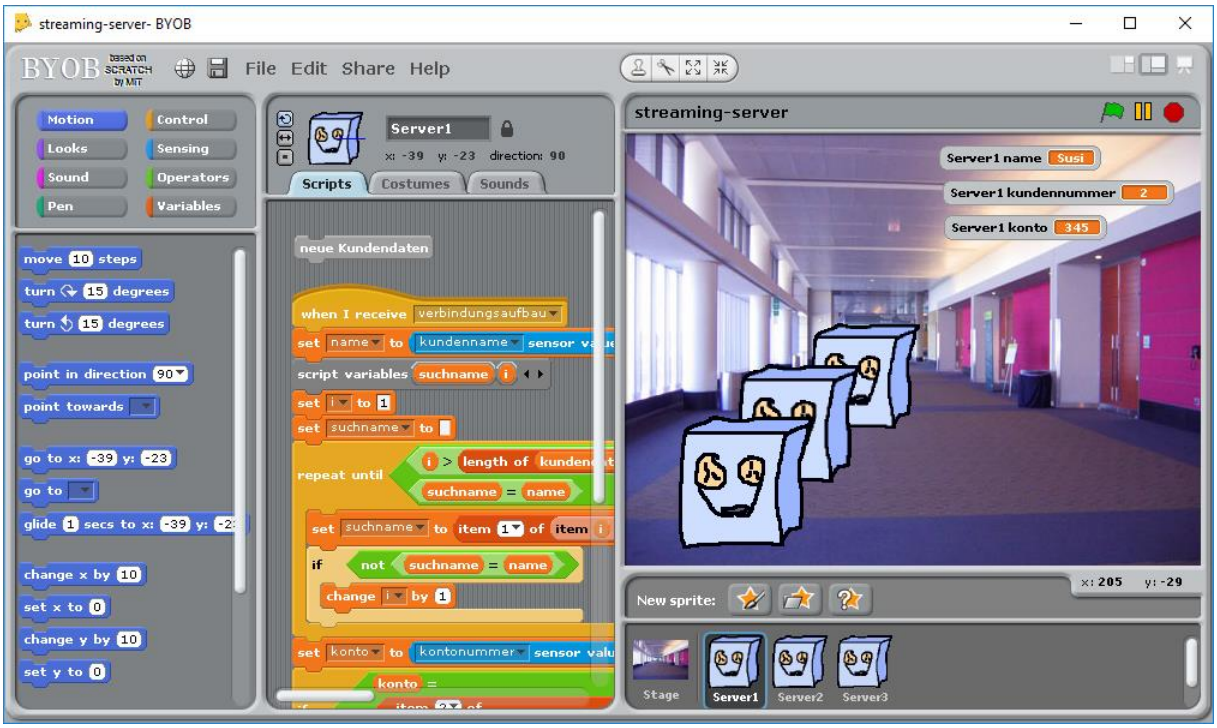


Bild 49: Streaming auf der Server-Seite

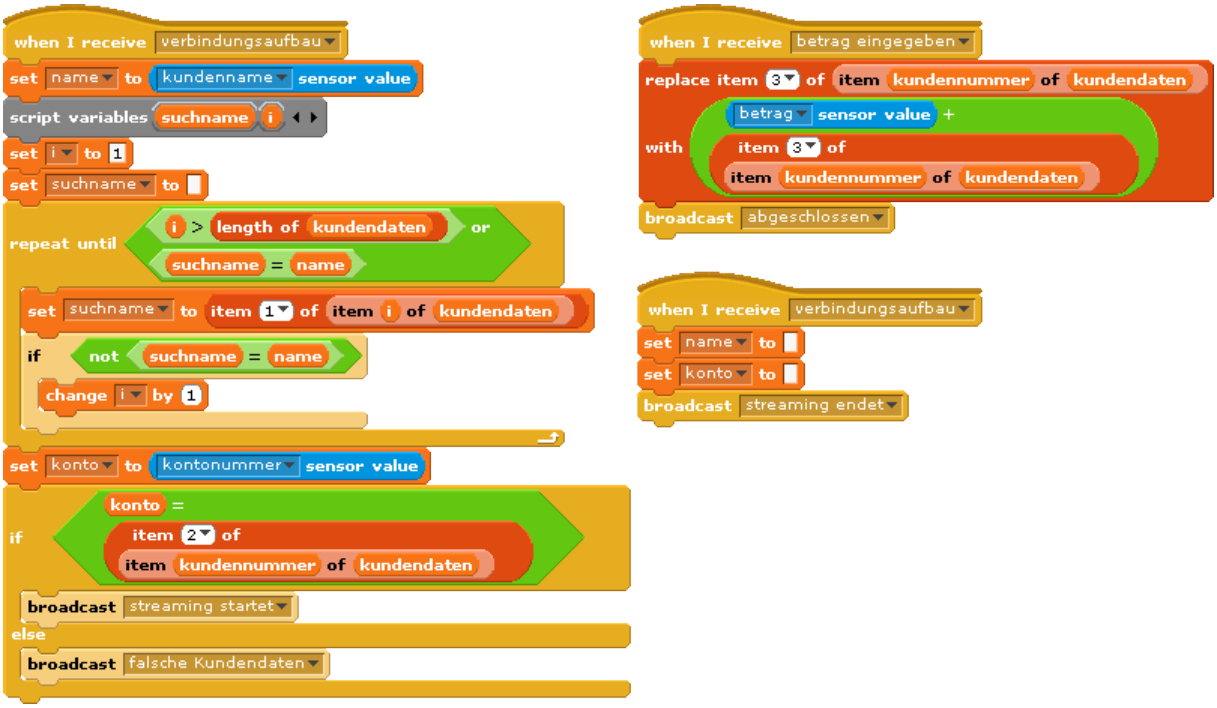


Bild 50: Skripte des Servers

Wie man sieht, sind die bisherigen Skripte trivial. Die Lösung lässt sich auf sehr unterschiedliche Weise stark ausbauen. Die beim Einlogprozess übermittelten Daten sollen die Information enthalten, dass es sich beim Benutzer (in diesem Fall) um Susi handelt. Offensichtlich kann diese Information stimmen, oder auch nicht. Die Daten alleine bestimmen also nicht den Informationsgehalt, sondern der gesamte Kontext ist wichtig, z. B. sein Sicherheitsaspekt, von dem abhängt, in wieweit den Daten zu trauen ist.

Alternatives Beispiel: **Zero Knowledge Authentifizierung**

Altersstufe: *Sekundarstufe II*
Werkzeug: *snap!*
Material: *zero knowledge protokoll.xml*

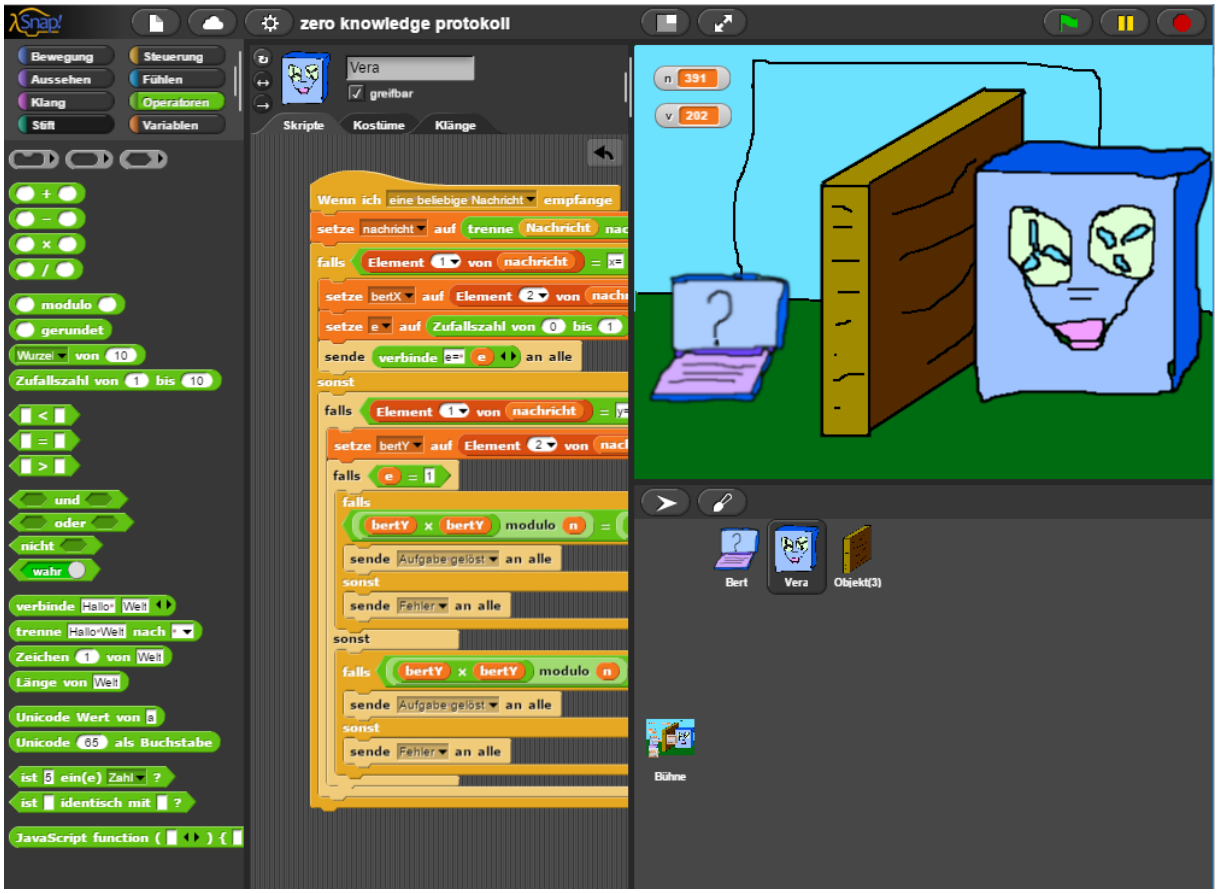


Bild 51: Zero Knowledge Authentifizierung

Die Idee des Zero-Knowledge-Protokolls[24] ist, dass ein Beweiser („Bert“) einem Verifizierer („Vera“) beweisen muss, dass er über bestimmte Informationen (den Schlüssel) verfügt, ohne dass der Beweiser diesen Schlüssel über das Netz mitteilt. Dazu stellt Vera dem Beweiser Aufgaben, deren Lösung man nur mit einer gewissen Wahrscheinlichkeit p erraten kann. Wäre $p = 0,5$ und die Anzahl der Fragen $n = 10$, dann wäre diese Fragenkette nur mit einer Wahrscheinlichkeit von $(0,5)^{10} = 0,00097..$ durch Erraten richtig zu beantworten. Wählt man n höher, dann lässt sich das antwortende System praktisch beliebig sicher authentifizieren. Wir wählen eine einfache Version des Fiat-Shamir-Protokolls, die wie folgt abläuft:

Voraussetzung:
Bert bestimmt eine große Zahl n als Produkt zweier großer Primzahlen: $n = p*q$. Dann wählt er eine zu n teilerfremde Zahl s und berechnet $v = s^2 \ mod \ n$. n und v veröffentlicht er.

Bild 52: Bestimmung der öffentlichen Werte n und v durch Bert

[24] https://de.wikipedia.org/wiki/Fiat-Shamir-Protokoll

Zur Authentifizierung werden dann die folgenden Schritte mehrfach durchlaufen:

1. Bert bestimmt eine Zufallszahl *r* und sendet $x = r^2 \ mod \ n$ an Vera.

2. Vera merkt sich *x*, bestimmt ein zufälliges Bit *e* (*0* oder *1*) und sendet dieses an Bert.
3. Bert berechnet $y = r*s^e \ mod \ n$ und sendet y an Vera.

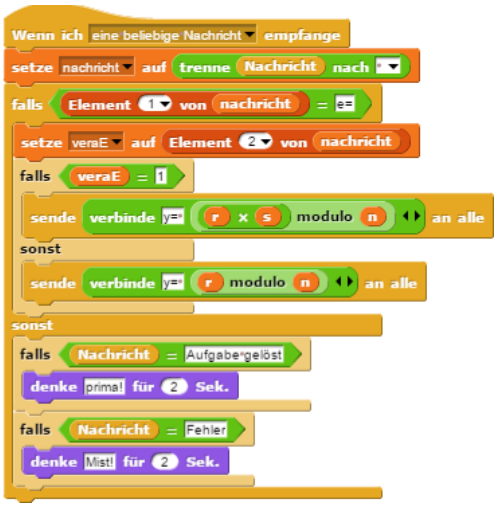


Bild 53: Skripte von Bert

4. Vera überprüft, ob $y^2 \ mod \ n = x*v^e \ mod \ n$ ist und teilt den Erfolg bzw. Misserfolg mit.

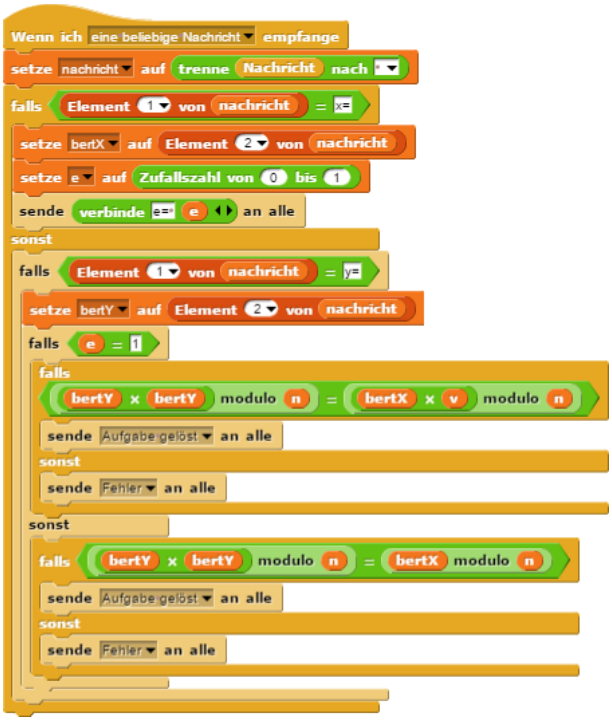


Bild 54: Veras Skript

Auch in diesem Fall werden Daten zwischen den Kommunikationspartnern übertragen. Allerdings ergibt sich deren Inhalt nicht aus den übertragenen Werten, sondern aus deren Stimmigkeit innerhalb des Rahmens des Protokolls, der über den reinen Datenaustausch herausgeht. Es geht also nicht um die Daten selbst, sondern um deren Eigenschaft, „richtig“ zu sein.