



Scratch im Anfangsunterricht

Erfahrungen im Wahlfach
Informatik der Klasse 9

Markus Rohe

Dr. Boris Briehl

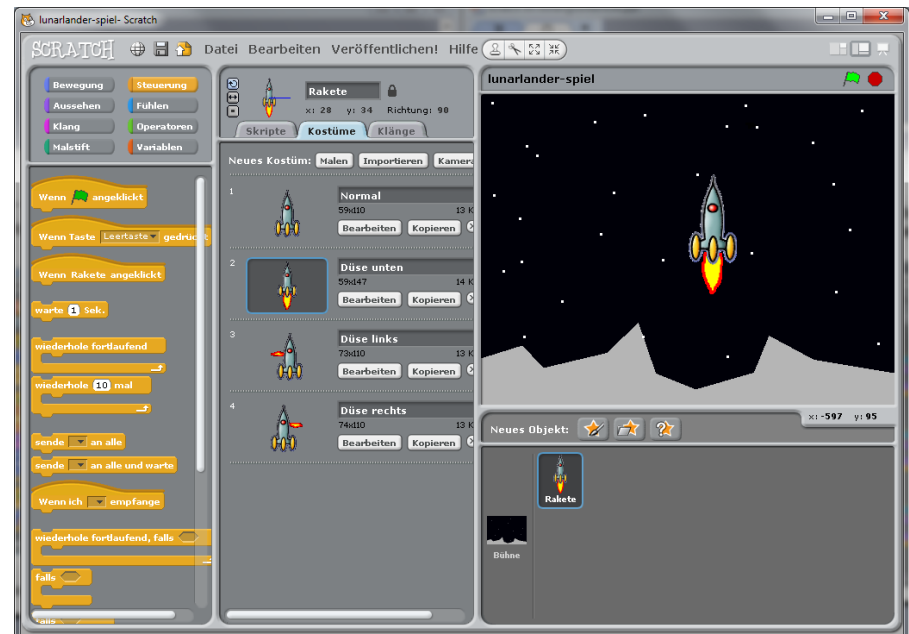
Gliederung

- Was ist Scratch, weshalb Scratch?
- einfache Einstiegsprojekte erstellen
- Demonstration fertiger Schülerprojekte
- Erfahrungen aus dem Unterricht
- Ausblick
-

Was ist SCRATCH ?



- entwickelt am MIT Media Lab
- Zielgruppe: Jugendliche zwischen 8 und 16 Jahren
- keine Installation erforderlich
- sehr stabil
- ressourcenschonend: läuft auch auf älterer Hardware flüssig
- Scratch-Homepage <http://scratch.mit.edu/>, kostenlos
- Windows, Linux, Mac OS X
-



Weshalb SCRATCH?

- Einführung des Wahlfachs Informatik am Gymnasium Ramstein-Miesenbach in Klasse 9 mit Beginn des laufenden Schuljahres
- Lehrplanentwurf Informatik
 - Inhaltsbereich „Algorithmisches Problemlösen“
 - Teilkompetenz „Abläufe mit Hilfe von algorithmischen Grundstrukturen beschreiben“
 - Hinweis: „Eine didaktische Programmierumgebung nutzen, in der die Ausführung von Elementaranweisungen veranschaulicht wird.“
 - Einzelschrittausführung
- Suche nach geeigneter Programmierumgebung zur Einführung in die algorithmische Denkweise und imperative Programmierung
- Nachmittagsunterricht
- graphisches Programmiersystem
-

Weshalb SCRATCH?

- einfache, intuitiv bedienbare Programmierumgebung
- lädt zum selbst entdecken ein

The image shows a screenshot of the Scratch programming environment. The interface is divided into several sections:

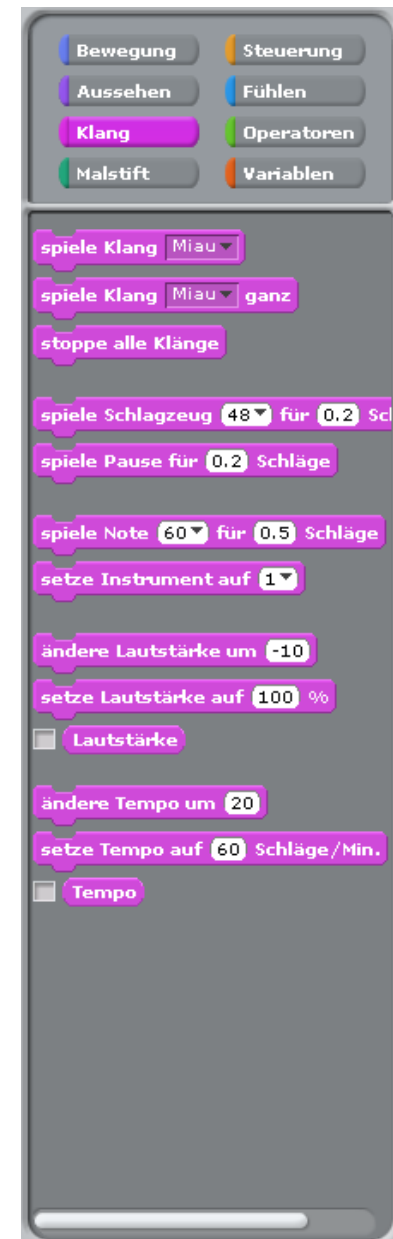
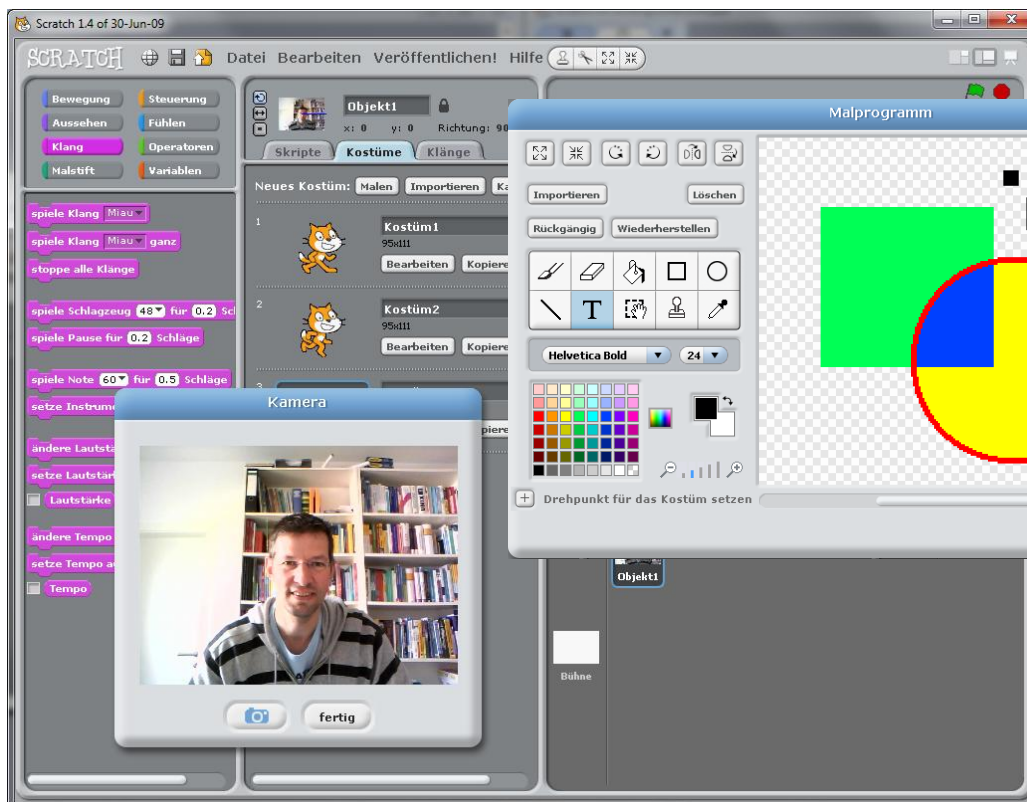
- Left Panel (Categories of Instructions):** A vertical list of categories for instructions, including Bewegung, Aussehen, Klang, Malstift, Steuerung, Fühlen, Operatoren, and Variablen. A callout box labeled "Kategorien von Anweisungen" points to this panel.
- Center Panel (Scripts):** A workspace for writing scripts. A callout box labeled "Programme (Skripte)" points to the script area. The script shown is a "Wenn angeklickt" block containing several actions: "ziehe Kostüm fragend an", "sage Nanu, was ist das denn? für 2 Sek.", "ziehe Kostüm dog2-c an", "gehe 40 -er Schritt", "warte 0.3 Sek.", "ziehe Kostüm dog2-b an", "gehe 40 -er Schritt", "warte 0.3 Sek.", and "sage Hmm, doch nix für mich!".
- Right Panel (Stage):** A workspace for the stage. A callout box labeled "Bühne" points to the stage area. The stage contains a blue dog character and a bunch of bananas. A callout box labeled "Objekte" points to the objects on the stage.
- Bottom Panel (Object Palette):** A workspace for creating new objects. A callout box labeled "Objekte" points to this panel. It shows a "Neues Objekt:" section with icons for "Bello" and "Banene...".
- Bottom Panel (Code Blocks):** A workspace for code blocks. A callout box labeled "Programmierkacheln" points to this panel. It contains various code blocks for movement, rotation, and position.

Wozu überhaupt graphische Programmiersysteme?

- Voraussetzungen: Wenig bis gar keine Programmiererfahrung
- Trennung von Problemlösung und Codierung
- Vermeidung von Frustration durch Syntaxfehler zu Beginn des Lernprozesses
- Lernen einer Programmiersprache wird verschoben
 - graphisch ist intuitiver
 - Korrekturen wirken sich in Echtzeit aus
- Der „Code-Schock“ kommt später, aber viele Schüler haben dann schon die Programmierkonzepte wie Variablen und Schleifen verinnerlicht
- wenig Ablenkung durch technische Details bei der Bedienung der Oberfläche: schlank, leicht, intuitiv (z.B. keine komplizierten Koordinaten- und Bewegungsabläufe)
-

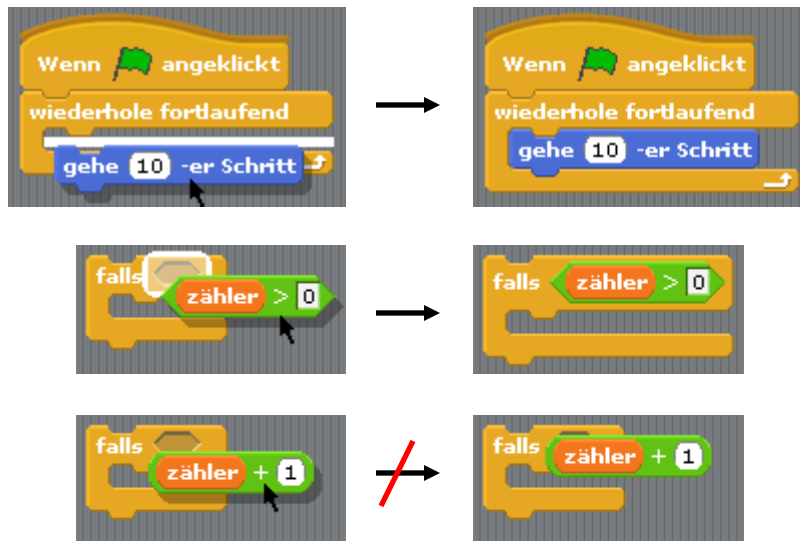
Weshalb SCRATCH?

- multimedial: Bilder, Grafiken, Klänge, Animationen, ...
- sehr ansprechend und motivierend für Schüler
- umfangreiches Online-Archiv mit fertigen Projekten verfügbar



Weshalb SCRATCH?

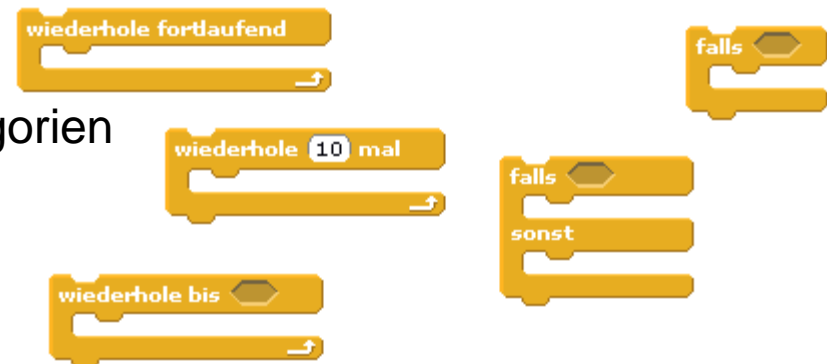
- Programme werden aus Programmierkacheln zusammengesetzt (Puzzle)
- Vermeidung von Syntax- / Grammatikfehlern



```
Wenn  angeklickt
  ziehe Kostüm geschlossen an
  wiederhole fortlaufend
    falls Entfernung von Ball < 20
      ziehe Kostüm offen an
      sende weg an alle
      warte 3 Sek.
      ziehe Kostüm geschlossen an
      sende reset an alle
      warte 0.1 Sek.
```

- farbliche Zuordnung der Kacheln zu Kategorien

| | |
|----------|------------|
| Bewegung | Steuerung |
| Aussehen | Fühlen |
| Klang | Operatoren |
| Malstift | Variablen |



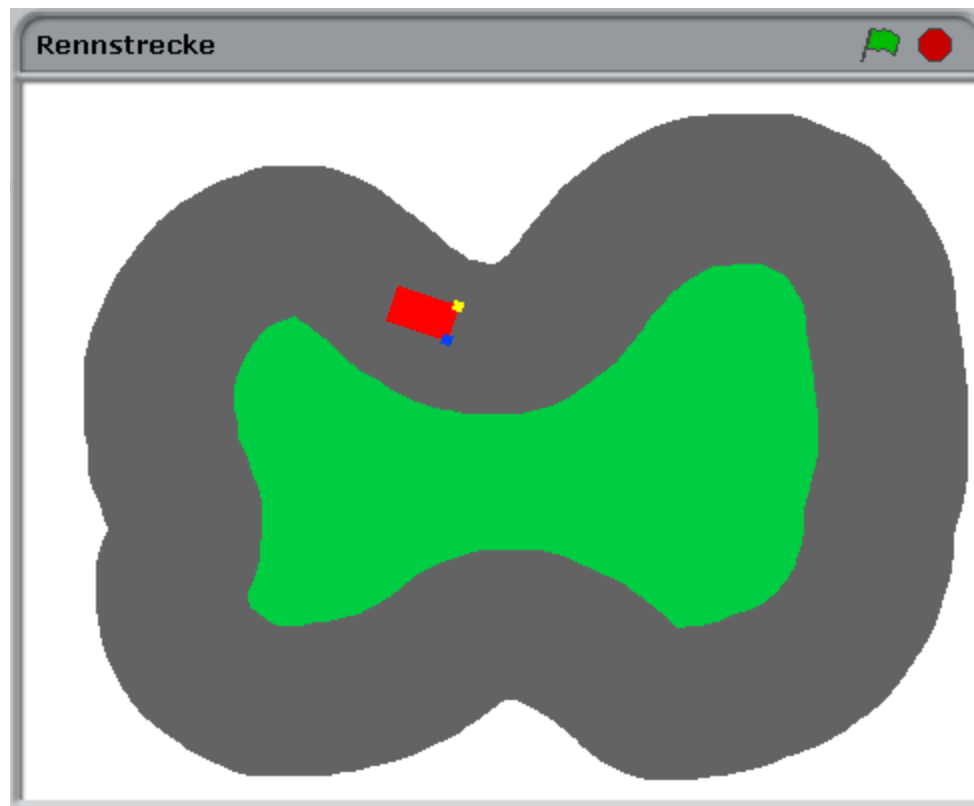
Eine einfache Animation



- einfache Animationen durch Kostümwechsel
- anfangs Aufgabenstellung = Regieanweisungen → Drehbuch
- Einzelschrittausführung

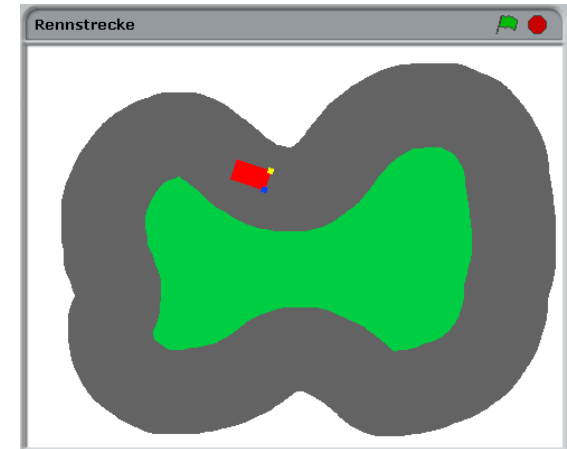
Ein einfaches Programm

Ein rotes Auto soll selbstständig auf einer Rennstrecke entlang fahren.



Ein einfaches Programm

- Rennstrecke als Hintergrund der Bühne malen:



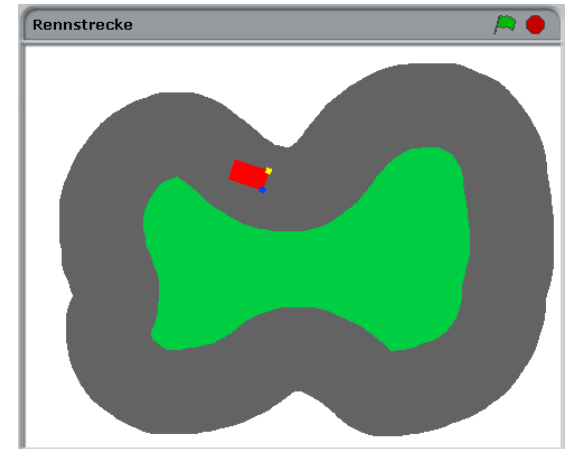
- Auto-Objekt erstellen und Kostüm malen:



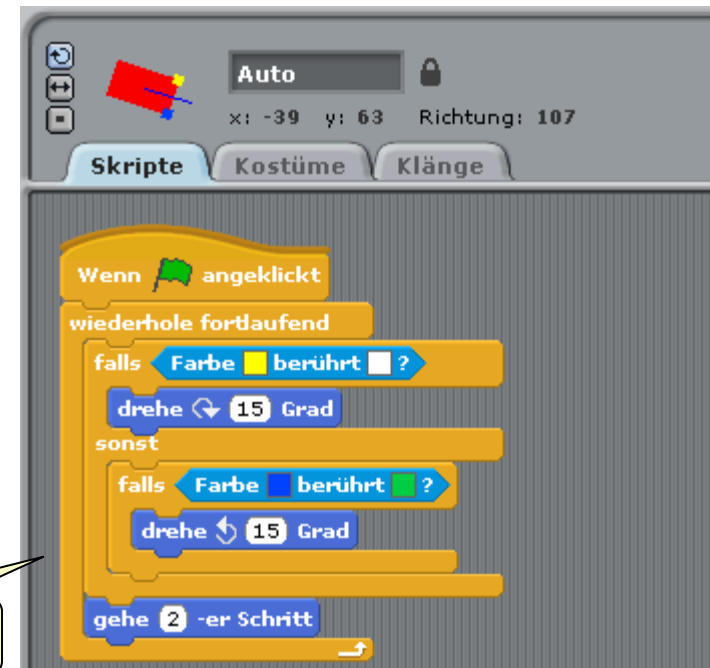
Eigenschaften des Objekts:
Koordinaten, Richtung,
augenblickliches Kostüm

Ein einfaches Programm

- Rennstrecke als Hintergrund der Bühne malen:



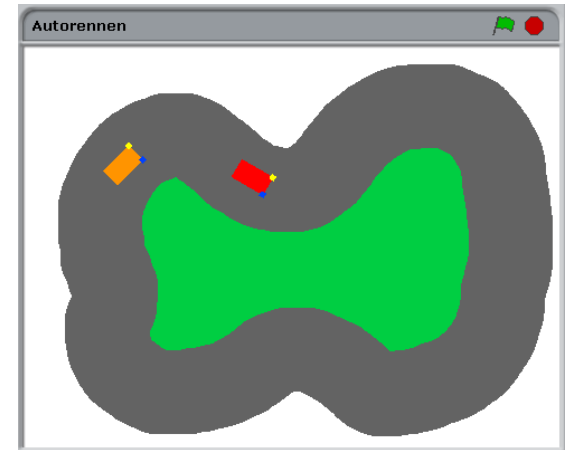
- Auto-Objekt erstellen und Kostüm malen:



Skript

Erweiterung: Autorennen

- Simultane Ausführung von Skripten



gleichzeitige ablaufende Skripte

Scratch script for "Auto" (red car). The script is triggered by "Wenn grünes Flagge angeklickt". It contains a "wiederhole fortlaufend" loop with two conditional blocks: "falls Farbe gelb berührt ?" (if yellow is touched) followed by "drehe 15 Grad" (turn 15 degrees), and "sonst" (otherwise) followed by "falls Farbe blau berührt ?" (if blue is touched) followed by "drehe 15 Grad" (turn 15 degrees). The loop ends with "gehe 2 -er Schritt" (move 2 steps).

Scratch script for "Auto2" (orange car). The script is triggered by "Wenn grünes Flagge angeklickt". It contains a "wiederhole fortlaufend" loop with two conditional blocks: "falls Farbe gelb berührt ?" (if yellow is touched) followed by "drehe 8 Grad" (turn 8 degrees), and "sonst" (otherwise) followed by "falls Farbe blau berührt ?" (if blue is touched) followed by "drehe 8 Grad" (turn 8 degrees). The loop ends with "gehe 4 -er Schritt" (move 4 steps).

Erfahrungen aus dem Unterricht

- eigenes Gestalten der Bühne, der Kostüme, etc. wirkt sehr motivierend
- visuell und auditiv ansprechende Projekte leicht realisierbar
- Programmierumgebung regt zu eigenem Entdecken an (Schleifen)
- Binnendifferenzierung leicht umsetzbar
 - Erweiterung von Projekten
 - Zusätzliche Aufgaben für leistungsstarke Schüler
 - eigene Ideen
- Trennung zwischen Algorithmik und Codierung
 - erleichtert Übergang zu einer „herkömmlichen“ Programmiersprache
 - Grundkonstrukte (Variablen, Verzweigungen, Schleifen) wurden bereits verwendet → Spiralprinzip, Wiedererkennungswert
- Ähnlichkeit zu Struktogrammen
- Wie viel Kreativität lasse ich zu? Wie eng sollte geführt werden?

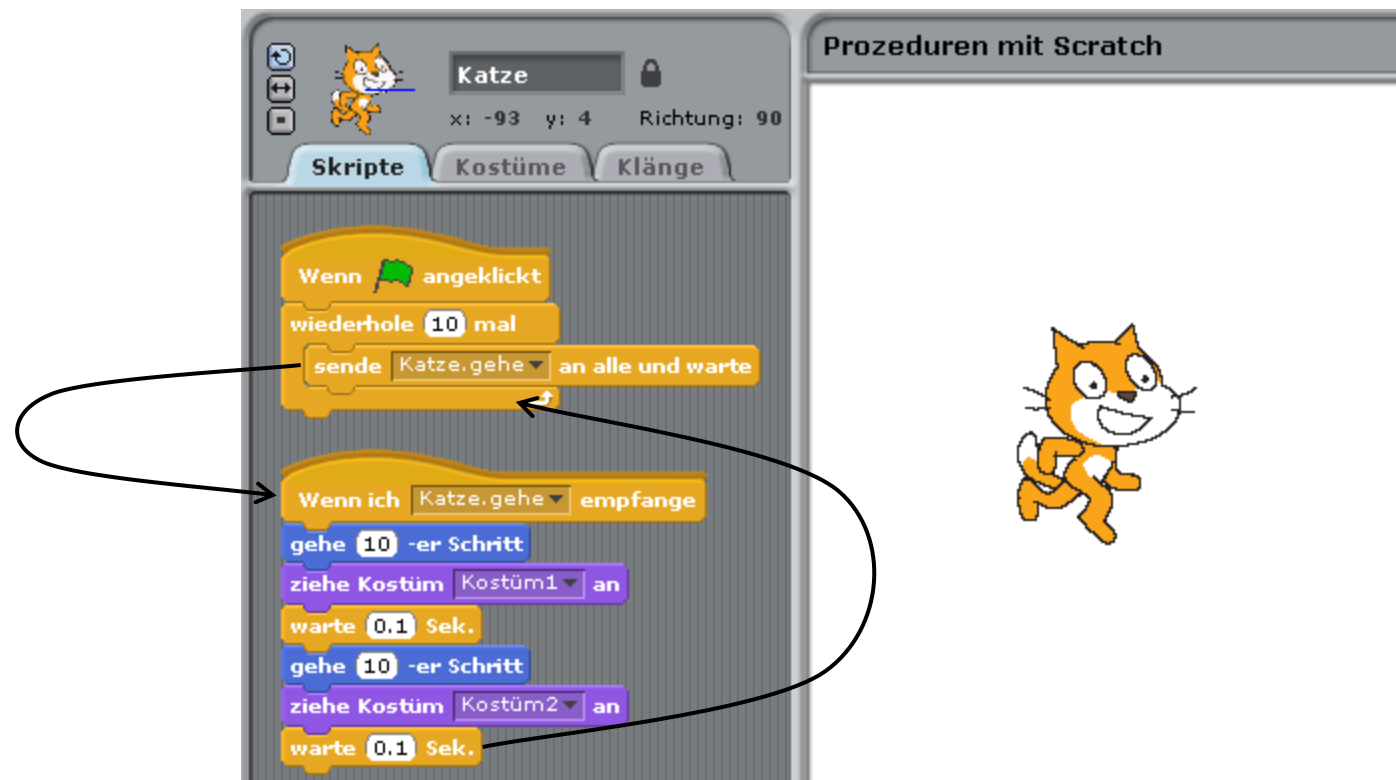


Weitere Programme

... zu folgenden Inhalte: Variablen, Ereignisse, ...

Ausblick: Prozedurale Programmierung

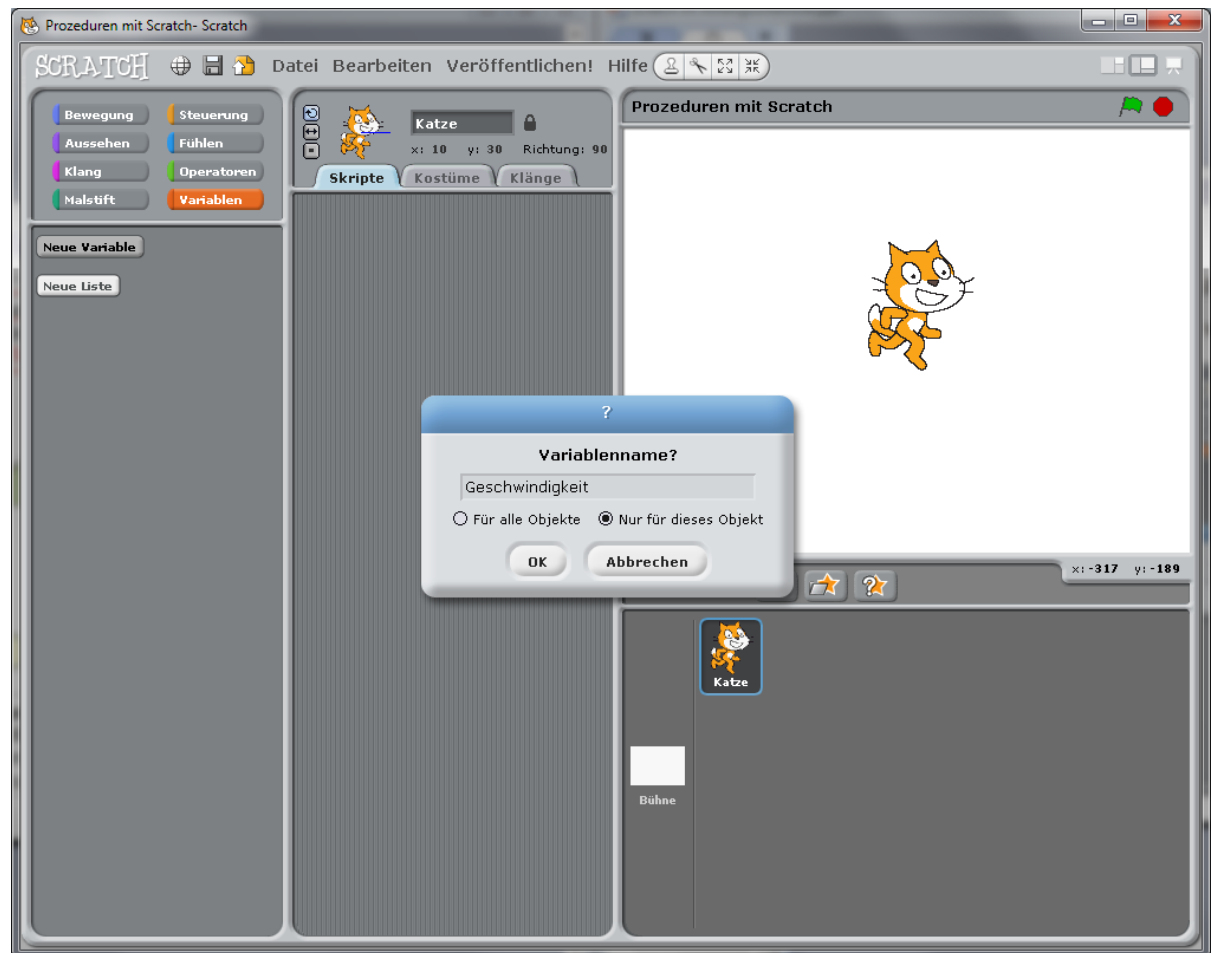
- Einführung in die prozedurale Programmierung
- Objekte schicken Nachrichten an „sich selbst“, um wiederkehrende Abläufe auszuführen → Modell für Unterprogramme
- Einschränkungen: keine Parameter, keine Funktionen
-



The screenshot shows the Scratch interface with a cat object named "Katze" at coordinates (-93, 4) and direction 90. The "Skripte" (Scripts) tab is active, displaying a script for the cat object. The script consists of two main blocks: a "Wenn angeklickt" (When clicked) block and a "Wenn ich empfangen" (When I receive) block. The "Wenn angeklickt" block contains a "wiederhole 10 mal" (Repeat 10 times) block, which in turn contains a "sende Katze.gehe an alle und warte" (Send Katze.gehe to all and wait) block. The "Wenn ich empfangen" block contains a sequence of actions: "gehe 10 -er Schritt" (Move 10 steps), "ziehe Kostüm Kostüm1 an" (Switch costume to Costume 1), "warte 0.1 Sek." (Wait 0.1 seconds), "gehe 10 -er Schritt" (Move 10 steps), "ziehe Kostüm Kostüm2 an" (Switch costume to Costume 2), and "warte 0.1 Sek." (Wait 0.1 seconds). A black arrow points from the "sende" block in the first script to the "Wenn ich empfangen" block in the second script, illustrating the message-passing mechanism. To the right of the script editor, the Scratch cat character is shown running.

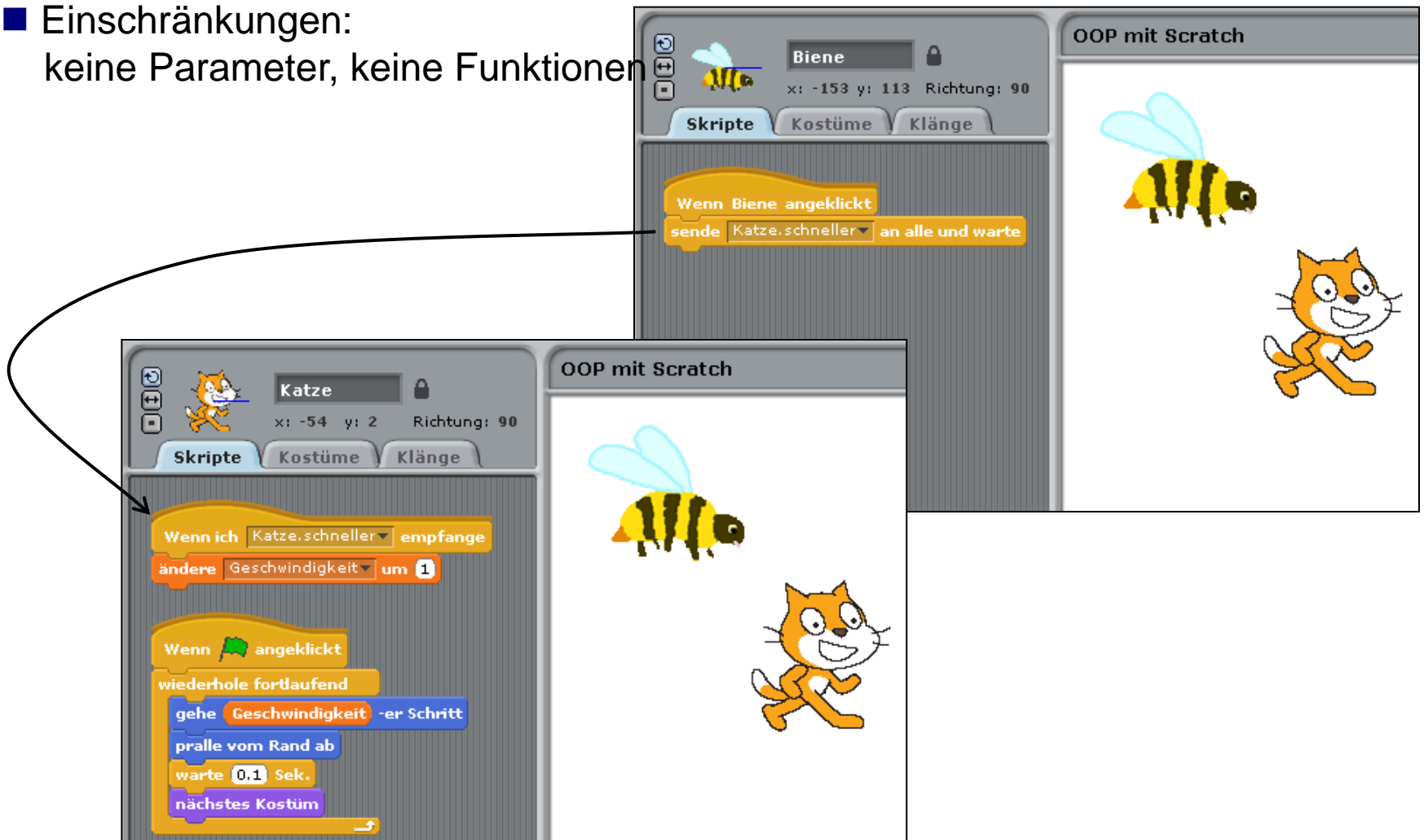
Ausblick: Objektorientierte Programmierung

- Einführung in die objektorientierte Programmierung
- Datenkapselung möglich mit Hilfe von Variablen, die nur für ein Objekt verfügbar sind



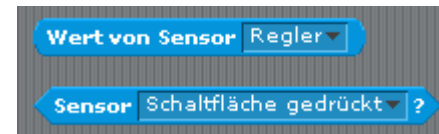
Ausblick: Objektorientierte Programmierung

- Objekte schicken Nachrichten an andere Objekte
- Einschränkungen:
keine Parameter, keine Funktionen

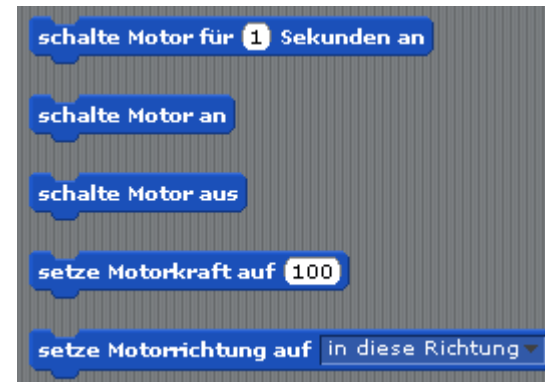


Ausblick: Weitere Möglichkeiten

- Anschluss an ein PicoBoard zur Verarbeitung externe Signale



- Steuerung von einfachen Maschinen aus dem Lego Education WeDo-Programm



Kontakt zu den Referenten

Markus Rohe, Dr. Boris Briehl

Gymnasium Ramstein-Miesenbach

Zum Kirchbühl 14

66877 Ramstein-Miesenbach

Telefon: 0 63 71 - 9 64 80

Telefax: 0 63 71 - 9 64 821

E-Mail: [gym.rm \[at\] t-online.de](mailto:gym.rm[at]t-online.de)

Beispielprogramme:

Online-Lehrbuch „Informatik in der Schule“, <http://www.inf-schule.de/>,

Kapitel „Algorithmisches Problemlösen mit Scratch“

Verweise

- Scratch Homepage <http://scratch.mit.edu/>, insbesondere <http://info.scratch.mit.edu/Research> (wissenschaftliche Publikationen zu Scratch)
- Dr. Klaus Becker, et al.: Online-Lehrbuch „Informatik in der Schule“, <http://www.inf-schule.de/>, Kapitel „Algorithmisches Problemlösen mit Scratch“ mit Übungsaufgaben der beiden Referenten
- Ralf Romeike: „Das bessere Werkzeug“, LOG IN Heft Nr. 163/164 (2010), S. 43-45
- Oliver Lau: „Programmieren lernen“, c't 2011, Heft 5, S. 148-153