

Robot Karol 3.0, Scratch und Co.

Alternative didaktische Zugänge zur Algorithmik

Johannes Neumeyer, Korbinian-Aigner-Gymnasium Erding
Klaus Reinold, Rupprecht-Gymnasium München

Ziel des Workshops

Aus Urheberrechtsgründen können hier Bilder aus dem Vortrag nicht veröffentlicht werden!

Neuerungen im Lehrplan Plus

1. Realschule
(Informationstechnologie)
2. Gymnasium
(Natur und Technik)

▼ IT 2.6 Programmierung – Algorithmen und Objekte

▼ IT 2.6.1 Modellieren und Codieren von Algorithmen (ca. 14 Std.)

Die Schülerinnen und Schüler analysieren Abläufe, modellieren sie mit algorithmischen Grundelementen und setzen sie mit einem geeigneten Programmierwerkzeug um.

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- analysieren Abläufe (z. B. bedingte Bewegung eines Roboters) und gliedern diese in sinnvolle Teilschritte, um dazu eindeutige Handlungsvorschriften zu formulieren.
- verwenden algorithmische Grundstrukturen, um Abläufe zu modellieren und stellen sie mit geeigneten Notationsformen dar.
- setzen algorithmische Grundstrukturen und Variablen ein, um Programmabläufe zu codieren und in einer geeigneten Programmierumgebung zu implementieren.
- testen und optimieren ihre einfachen Programme.

Inhalte zu den Kompetenzen:

- Algorithmus: Begriff und Beispiele
- algorithmische Grundstrukturen: Anweisung, Sequenz, Auswahl und Wiederholung
- Notationsformen, z. B. Programmablaufplan, Struktogramm, Pseudocode, Aktivitätsdiagramm
- Variablenkonzept: Bezeichner, Datentypen, Wertzuweisung

G1: Modellieren

Aufbauend auf ...
beschäftigen sich ...
hierfür verwenden ...
Diese Struktur ...

- Abläufe verknüpfen
- Die Grundstrukturen für Problemlösungen
- Algorithmen

(14 Std.)

...ten Systemen be-
nennen, dass sich die
beschreiben lassen.
t.

...ellierung geeigne-

NT 7.2 Schwerpunkt Informatik (ca. 28 Std.)

NT 7.2.1 Vernetzte Informationsstrukturen - Internet (ca. 13 Std.)

NT 7.2.2 Austausch von Information – E-Mail (ca. 2 Std.)

NT 7.2.3 Beschreibung von Abläufen durch Algorithmen (ca. 13 Std.)

Aktueller
LP G8

▶ NT7 2 Schwerpunkt Informatik (ca. 28 Std.)

▶ NT7 2.1 Vernetzte Informationsstrukturen (ca. 9 Std.)

▶ NT7 2.2 Projekt: Erstellen einer Hypertextstruktur (ca. 4 Std.)

▶ NT7 2.3 Chancen und Risiken digitaler Kommunikation (ca. 5 Std.)

▶ NT7 2.4 Beschreibung von Abläufen durch Algorithmen (ca. 10 Std.)

veröffentlichter
LP+ G9

2 Schwerpunkt Informatik (ca. 28 Std.)

2.1 Vernetzte Informationsstrukturen (ca. 8 Std.)

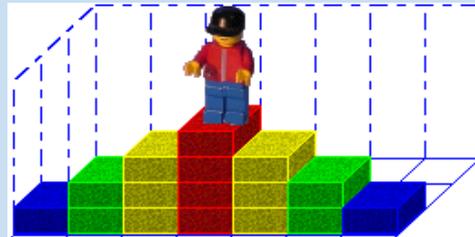
2.2 Chancen und Risiken digitaler Kommunikation (ca. 5 Std.)

2.3 Beschreibung von Abläufen durch Algorithmen (ca. 11 Std.)

2.4 Projekt (ca. 4 Std.)

LP+ G9
(Verlagsentwurf)

Übersicht vorgestellter Werkzeuge und Aufgaben



3.0



SCRATCH





Blocksprachen vs.



Speichern:

- Download als *.sb2 Datei
- Speichern in der Cloud

Zentral:

- Veröffentlichen in der Cloud
- Projekte Remixen
- Community, Forum
- Nicht (ohne Weiteres) auf dem Smartphone ausführbar (Flash-basiert!)

Schnittstellen zur „physical world“:

- Pico Board

Speichern:

- Speichern im Browser
- Export als XML Datei
- Speichern in der Cloud

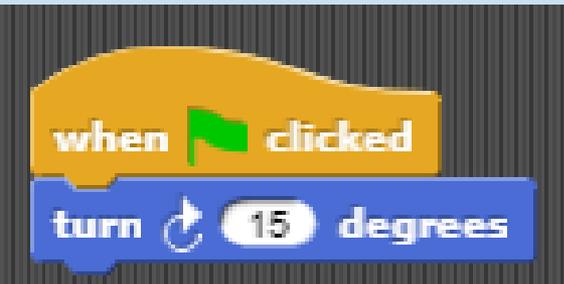
Zentral:

- Linkfreigabe via „share“
- weniger Spielereien; übersichtlicher
- nach Speichern in der Cloud auch direkt auf Smartphone ausführbar
- Möglichkeit zur Einbindung von JavaScript

Schnittstellen zur „physical world“:

- Arduino microcomputer (Snap4Arduino)
- Lego Mindstorms NXT

JS in Snap



[oder umfangreicher](#)

[oder andersherum](#)

Blockly

Offene Bibliothek zur Integration eines blockbasierten Editors in verschiedenen Umgebungen:

- <https://lab.open-roberta.org/>
(Fraunhofer Institut Intelligente Analyse- und Informationssysteme) → Calliope
- www.Code.org → Hours of code
- **BWInf: Jugendwettbewerb Informatik**
(Lernmaterialien -> Einführungsaufgaben)

The screenshot displays the Blockly interface for a robot navigation task. At the top, there are three version indicators: 'Version ☆☆☆', 'Version ☆☆☆', and 'Version ☆☆☆'. Below this, a text box contains the instructions: 'Programmiere den Roboter so, dass er das grüne Feld erreicht, ohne gegen die Hindernisse zu laufen. Um zum Beispiel den Roboter nach unten zu bewegen, falls sich rechts von ihm ein Hindernis befinden, kannst du schreiben:'. A small code snippet shows a 'falls' block containing a 'Hindernis rechts' block and a 'mache' block containing a 'gehe nach unten' block.

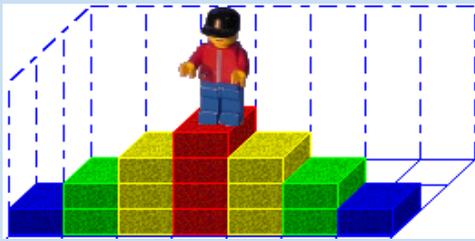
The main workspace features a 20x20 grid with a path of white squares leading from a green robot at the bottom left to a green goal square at the top right. Below the grid are navigation controls: '<|>', '|>', '>', '>>', '>>>', and '>|'. A blue button at the bottom reads 'Ausführen und bewerten'.

On the right, a 'Roboter-Programm' palette contains several blocks: 'gehe nach rechts', 'gehe nach oben', 'gehe nach unten', 'Hindernis rechts', 'wiederhole 10 mal: mache', and 'falls'. A status message above the palette reads 'Noch 10 von 10 Bausteinen verfügbar.'

Blockly

Offene Bibliothek zur Integration eines blockbasierten Editors in verschiedenen Umgebungen:

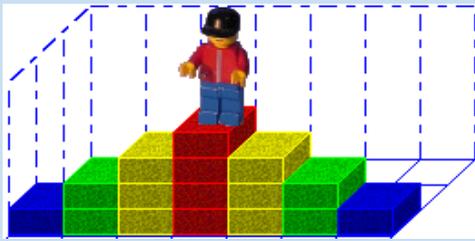
- <https://lab.open-roberta.org/>
(Fraunhofer Institut Intelligente Analyse- und Informationssysteme) → Calliope
- www.Code.org → Hours of code
- **BWInf: Jugendwettbewerb Informatik**
(Lernmaterialien → Einführungsaufgaben)
- [MIT App Inventor](https://www.mitsubishicorp.com/education/learning/learning-experiences/learning-experiences-2018/learning-experiences-2018-10-10-2018-10-10)



Karol 3.0

- Vorliegende Version ist noch nicht endgültig!
- Die endgültige Version wird nach Fertigstellung unter www.informatikschulbuch.de veröffentlicht.
- Kursteilnehmer können die Beta-Version per Mail anfordern unter

reinold "at" rupprecht-gymnasium.de



3.0 – Bewährtes

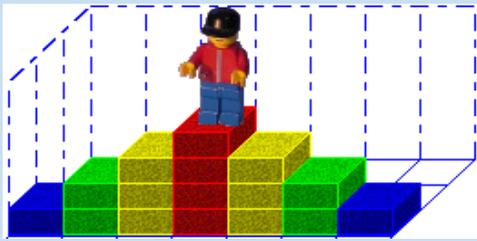
- alte Projekte weiterhin nutzbar
- Aufbau bewusst analog zu Karol 1.0

Robot Karol

Datei Bearbeiten Suchen Struktogramm Welt Ablauf Einstellungen Hilfe

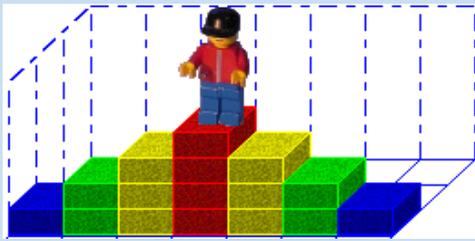
```
1 Methode Doppelschritt
2   karol.Schritt()
3   karol.Schritt()
4 EndeMethode
5
6 Doppelschritt
7
```

PositionX: 1 | PositionY: 1 | Blickrichtung: S | Information



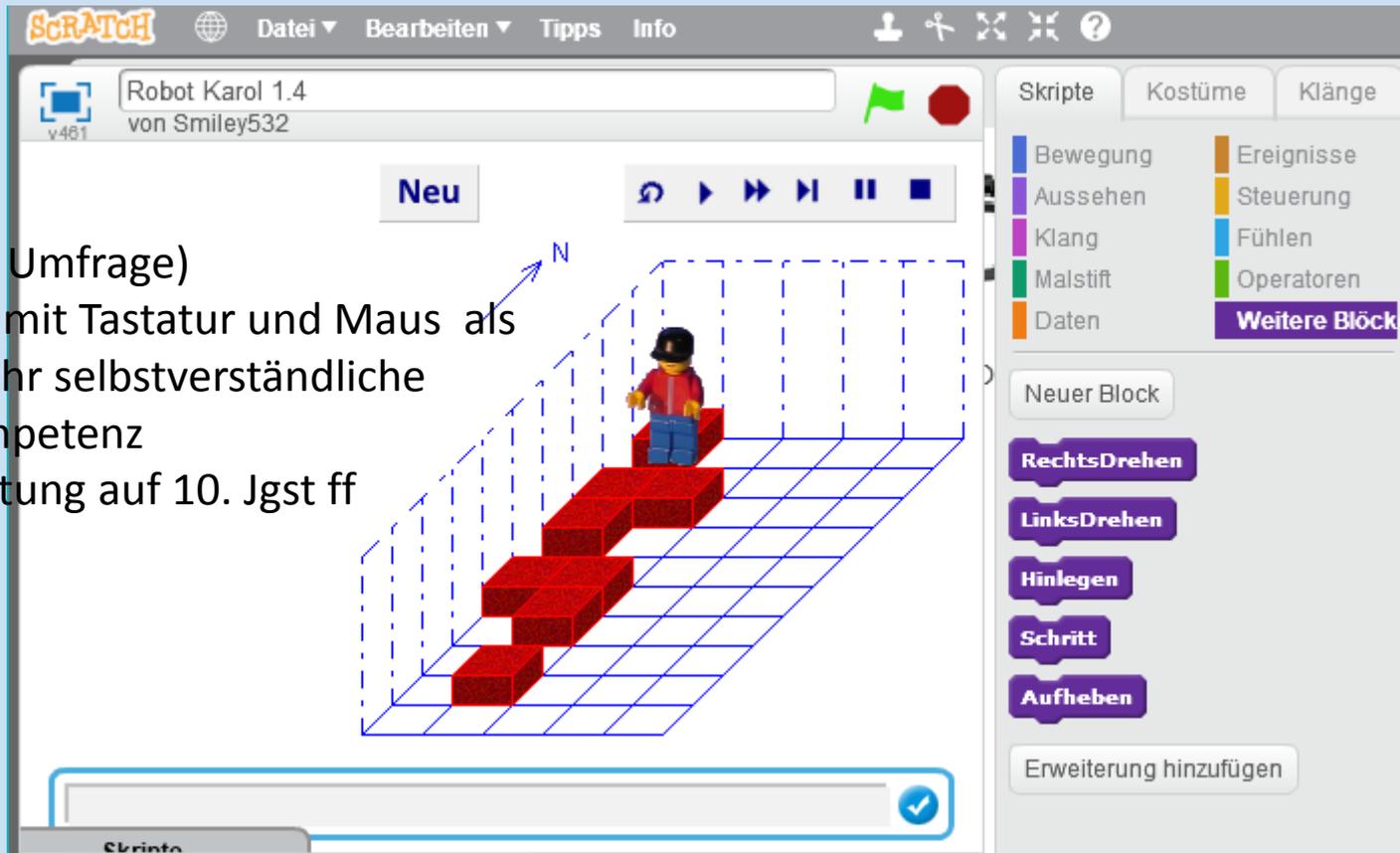
3.0 – Neuerungen

- Plattformunabhängigkeit auf der Basis von Java 1.8
- Hinlegen und MarkeSetzen alternativ mit Parameter für Farbe
- Auch bei Bedingungen Parameterwert möglich (z. B. IstZiegel(rot))
- EndeWiederhole statt *Wiederhole etc. möglich
- Methode statt Anweisung möglich
- Struktogramm in die Zwischenablage kopieren
- Konsequente Verwendung der Punktschreibweise möglich



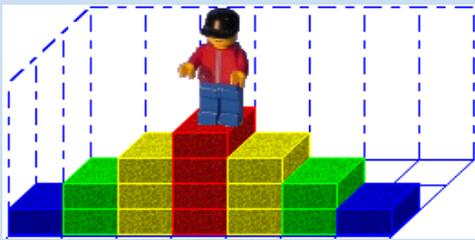
3.0

– Warum nicht auch blockbasiert?



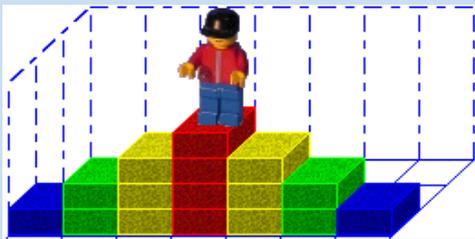
- (Bild aus Umfrage)
- Umgang mit Tastatur und Maus als nicht mehr selbstverständliche Basiskompetenz
- Vorbereitung auf 10. Jgst ff

Aus Urheberrechtsgründen können hier Bilder aus dem Vortrag
Teilweise nicht veröffentlicht werden!



vs. Blocksprachen

- Quelltext muss eigenständig strukturiert werden
 - Editor mit Syntaxhervorhebung
 - Arbeiten mit der Tastatur
 - Struktogramme sind integriert – Abläufe können auf's Wesentliche reduziert werden
 - zielgerichteteres Arbeiten: Endzustand oft klar definiert
- Programmierung nach Baukastenprinzip
→ keine Tippfehler/Syntaxfehler
 - Motivation u. U. höher, schnell schöne kreative Programme
 - eher offene Projekte



vs. Blocksprachen

Pong

**Vorsicht mit Objektkommunikation
– Lehrplan fordert reine Algorithmik!**

In einem Objekt „Sende“:



Im anderen Objekt
„Wenn ich Nachricht 1 empfangen“:



„Multitasking und Ereignissteuerung sind [bei Scratch] selbstverständliche Konzepte, die vom Programmier-Anfänger nebenbei erfasst werden.“

https://scratch-dach.info/wiki/Scratch#Werte_und_Variablen

Pong ohne Objektkommunikation

Vorgehen im Buch

Allgemeine Einstiege ohne
Programmiersprachenbezug

Aus Urheberrechtsgründen können hier Bilder aus dem Vortrag
nicht veröffentlicht werden!

Vorgehen im Buch

Programmiersprachensalat

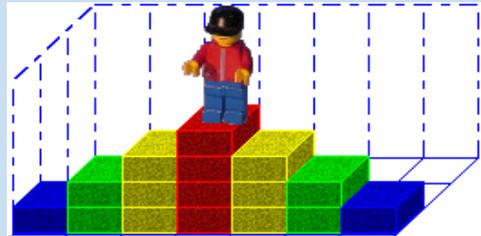
Aus Urheberrechtsgründen können hier Bilder aus dem Vortrag nicht veröffentlicht werden!

Vorgehen im Buch

Allgemeine und Sprachbezogene Aufgaben

Aus Urheberrechtsgründen können hier Bilder aus dem Vortrag nicht veröffentlicht werden!

Arbeitsaufträge (1 Std.)



3.0



Jugendwettbewerb
Informatik

