Computational Thinking

Jedes Kind verdient es, gut auf die Zukunft vorbereitet zu werden!

Jede Schule und jede Lehrkraft sollte dabei unterstützt werden.

Harald Zigala
h.zigala@tsn.at

Was ist Computational Thinking?

- Seymour Papert: Children, computers and powerful ideas (1980)

- Jeanette Wing: Artikel Computational Thinking (2006)

Was ist Computational Thinking?

WIKIPEDIA
The Free Encyclopedia

Rechnerisches Denken

20 Sprachen

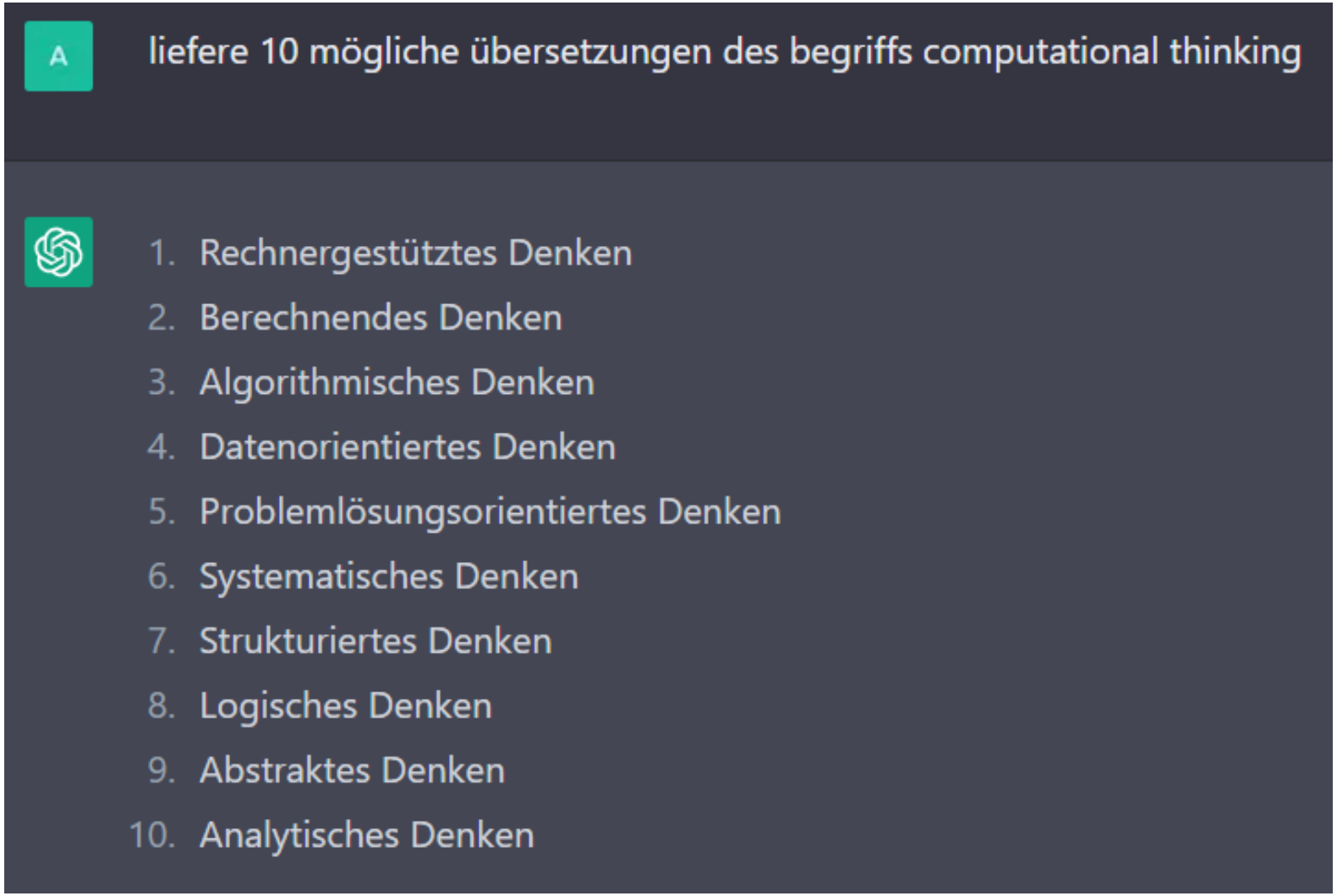
Artikel Sprechen Lesen Bearbeiten Siehe Verlauf

Aus Wikipedia, der freien Enzyklopädie

Computational Thinking (CT) ist die mentale Fähigkeit, Konzepte, Methoden, Problemlösungstechniken und logisches Denken anzuwenden, die aus Informatik und Informatik stammen, um Probleme in allen Bereichen, einschließlich unseres täglichen Lebens, zu lösen. [1] In der Bildung ist CT eine Reihe von Problemlösungsmethoden , bei denen Probleme und ihre Lösungen auf eine Weise ausgedrückt werden, die auch ein Computer ausführen könnte. [2] Es beinhaltet die Automatisierung von Prozessen, aber auch die Verwendung von Computern, um Prozesse (natürliche und künstliche) zu erforschen, zu analysieren und zu verstehen. [3] [4] [5]

Was ist Computational Thinking?

https://chat.openai.com/chat



Was ist Computational Thinking?

Swisscom.ch

Mit CT werden Probleme und deren Lösungen so formuliert, dass sie sowohl Menschen als auch Maschinen verstehen und ausführen können. Das Problem wird methodisch so zerlegt, dass Muster erkannt und Regeln formuliert werden können, die schliesslich zur Problemlösung führen. So vereinfacht, kann das Problem auch von einem Computer gelöst werden.

CT:= kreative Problemlösung, sodass sie von Menschen oder Computern gemeistert werden kann

Meist wiederkehrende Aufgaben wie …

- ❑ Aufstehen/Frühstück
- ❑ Anziehen
- ❑ Kochen/Rezept
- ❑ Einkauf

… aber auch wie …

- ❑ Hausübungen
- ❑ Wanderung
- ❑ Programmierungen
- ❑ …

bzw. computational thinking ist die große Schwester des Hausverstands

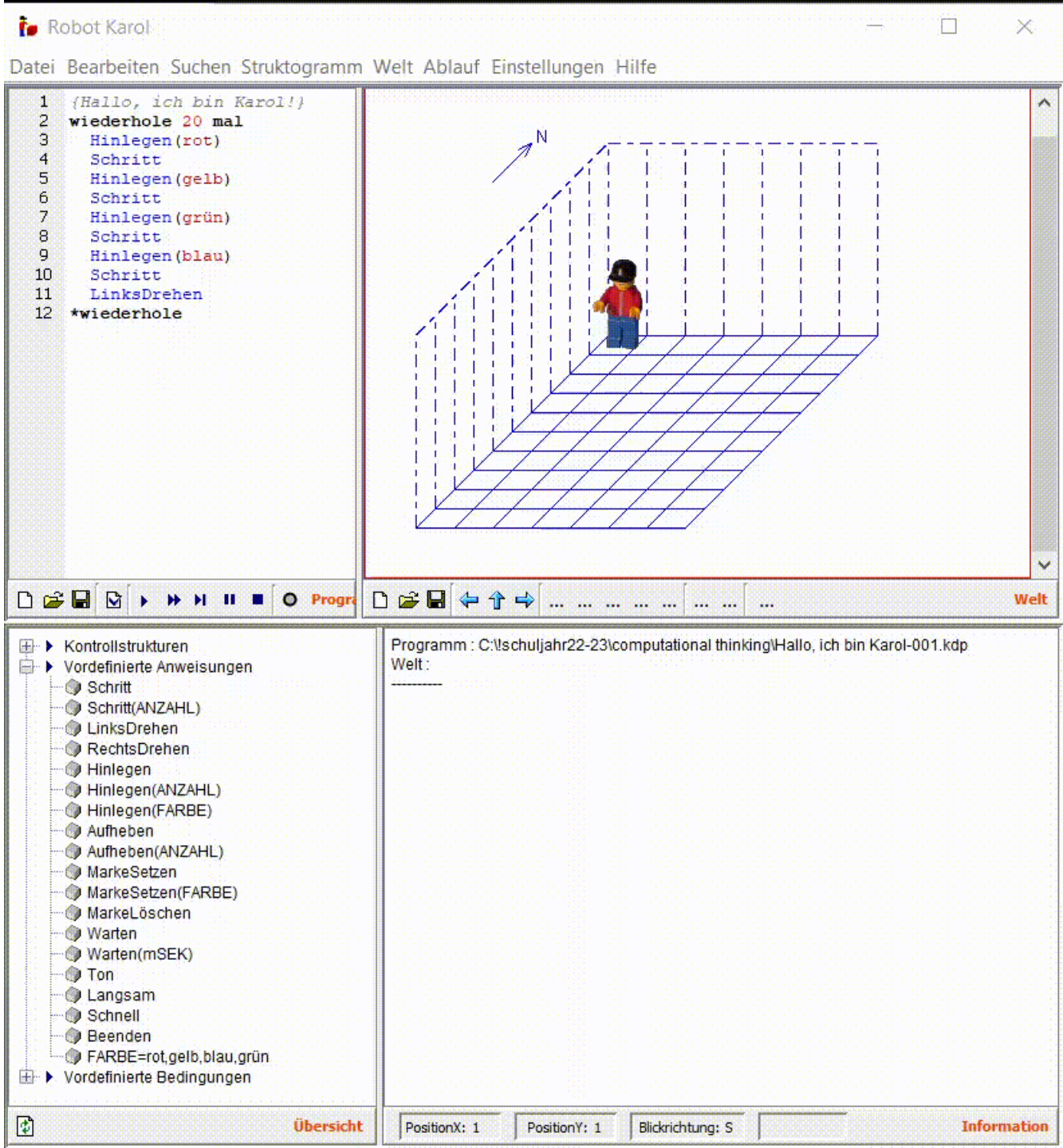
Robot Karol

- Mini-Language
- 2001 (D); aktuelle Version 3.0.4 (2019)
- kostenlos; keine Hardware
- erste Versionen nur Windows
- Java-basierter Karol für alle Plattformen (Win, iOS, Linux)

- 2D & 3D-Ansicht
- Verschiedene Abmessungen der Welten

- Sprache Deutsch
- Code direkt neben Ausführungsfenster
- z.T. Parameterübergabe
- Syntaxprüfung
- Einzelschritt & Schnelldurchlauf & Stopp

- Automatische Formatierung
- Struktogramm möglich
- Hilfe vorhanden



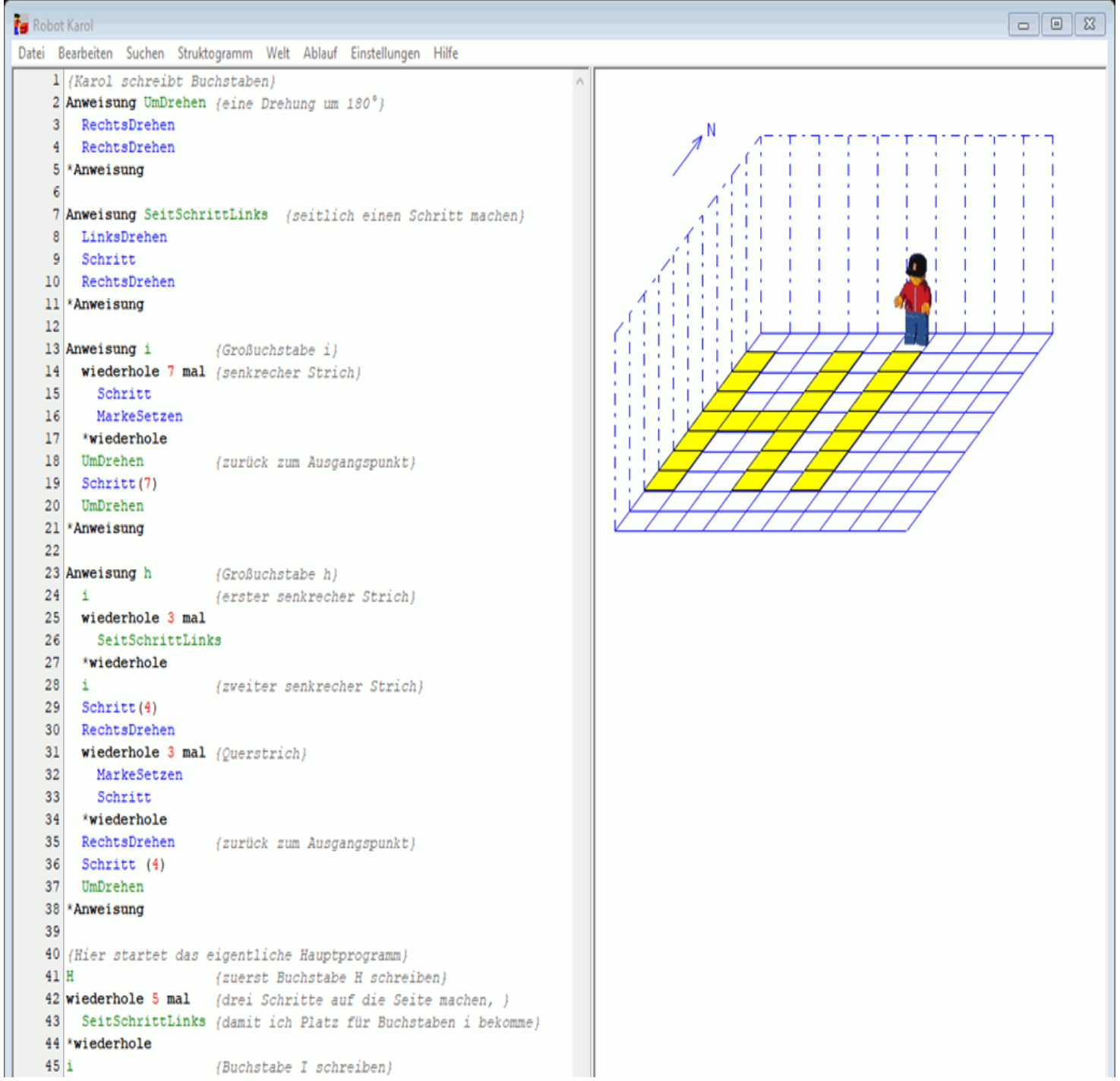
Computational Thinking

1. Dekomposition

- Ein komplexes Problem wird in verschiedene verstehbare Teilprobleme unterteilt.

2. Mustererkennung

- Innerhalb der Teilprobleme werden Muster gesucht. Oft sind die erkannten Abläufe bereits ganz oder in Teilen aus anderen Problemlösungsprozessen bekannt.



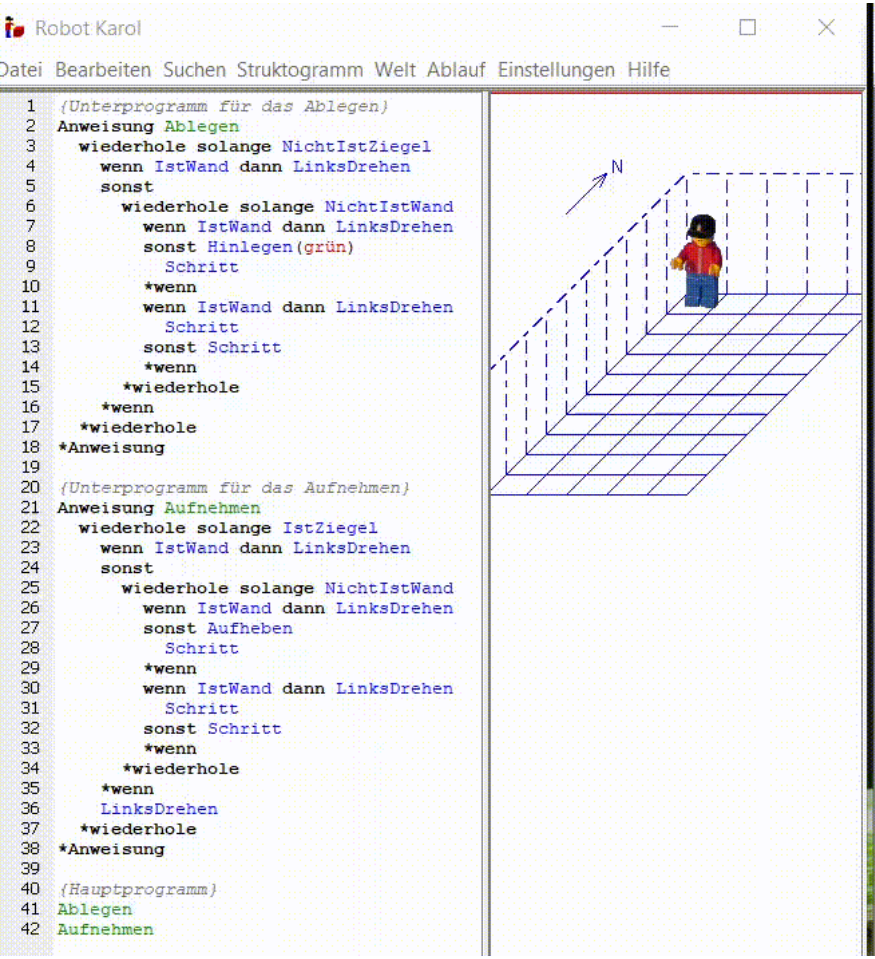
Computational Thinking

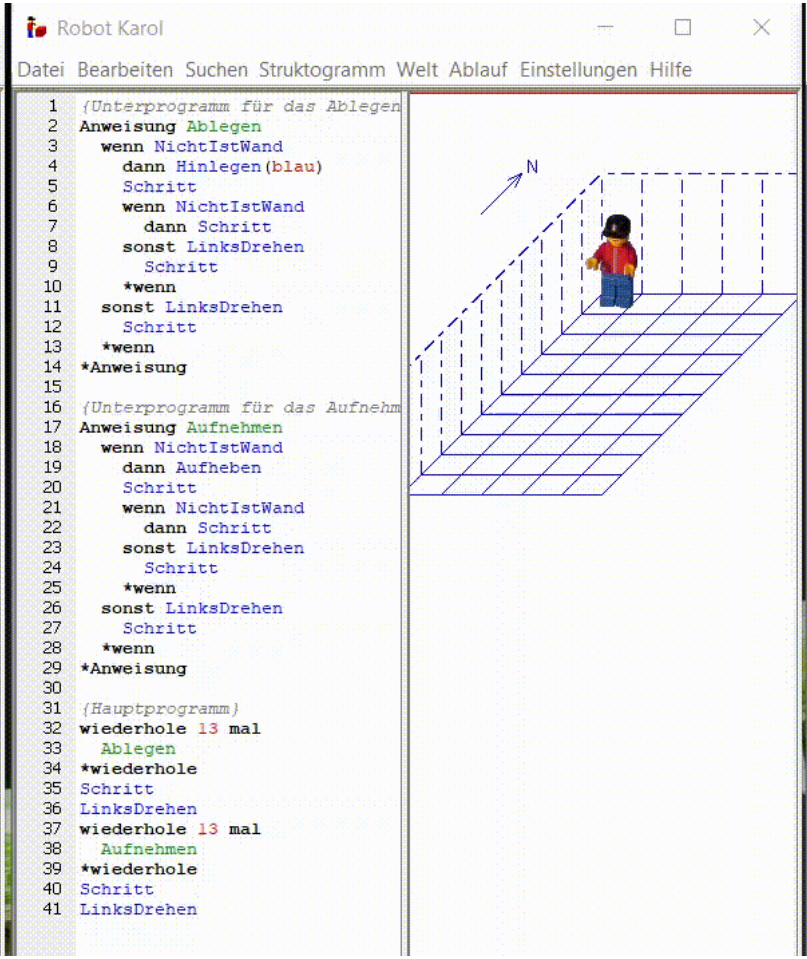
3. Abstraktion

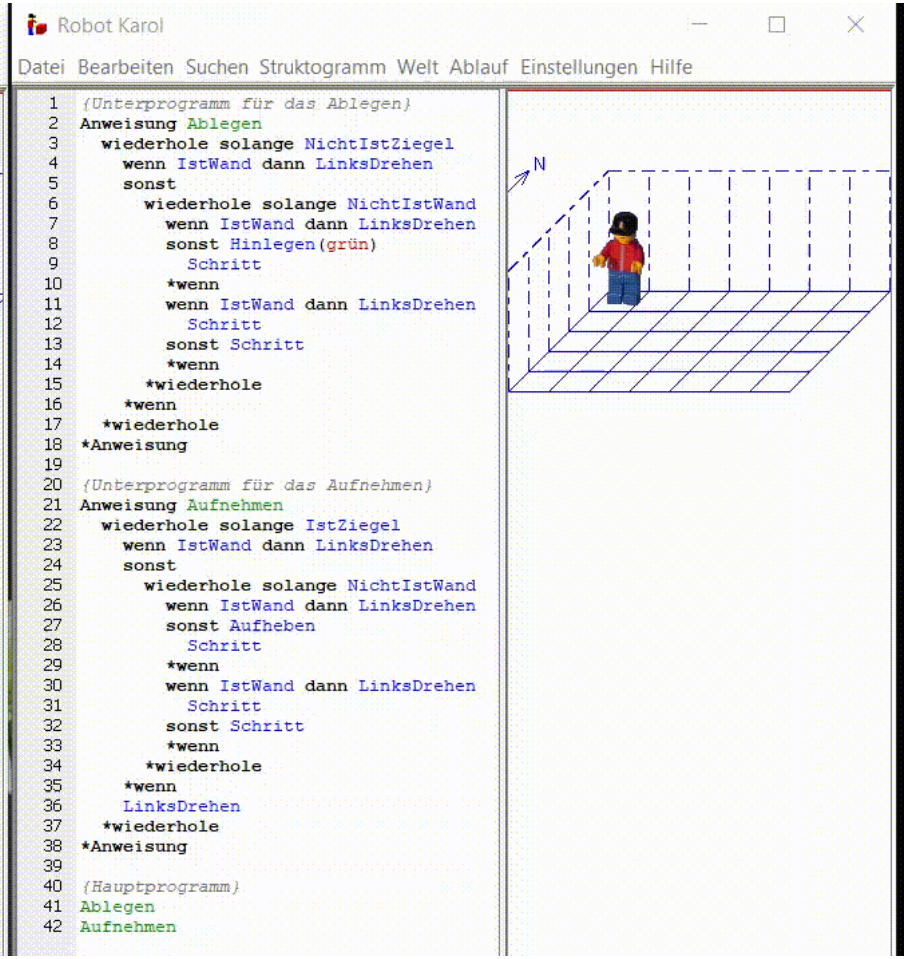
- Unwichtige Aspekte des Problems werden ausgeblendet. Der Fokus liegt auf den relevanten Details. Das Ausgangsproblem wird stark vereinfacht.

4. Algorithmen

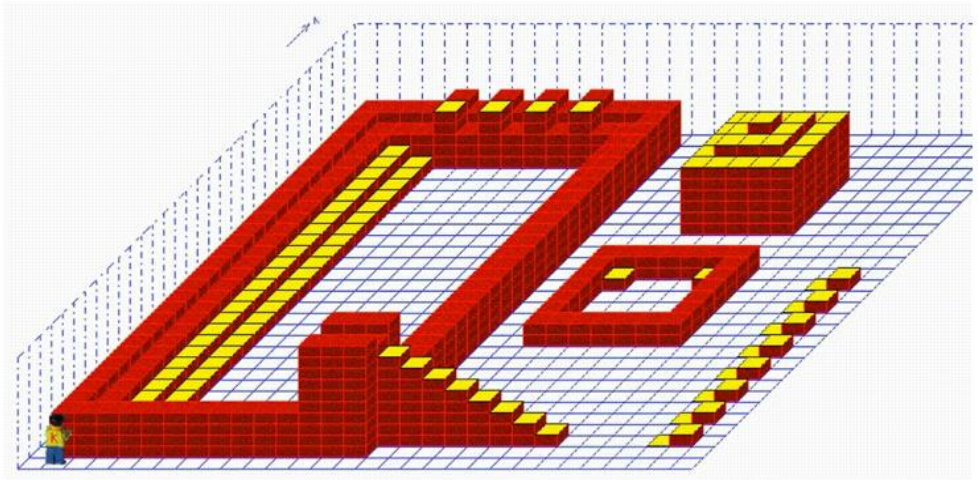
- Durch die vorangehenden Denkschritte wissen wir, nach welchen Mustern das Problem funktioniert und wie wir diese beeinflussen können. Nun können Regeln respektive Algorithmen definiert werden, die zur Problemlösung führen.

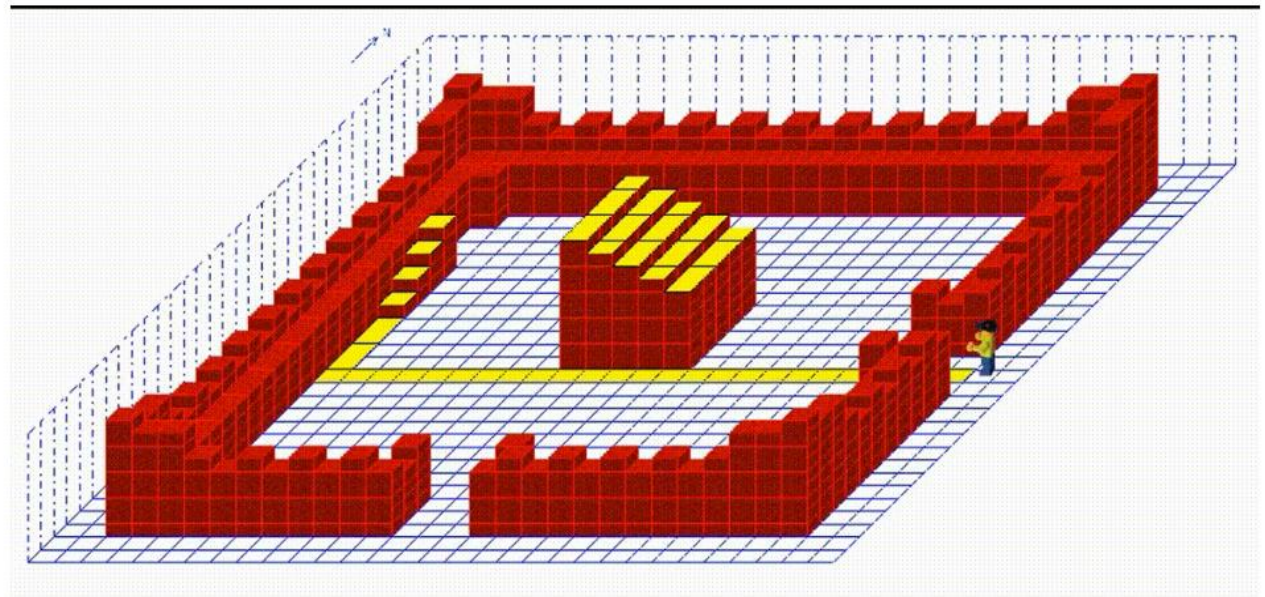




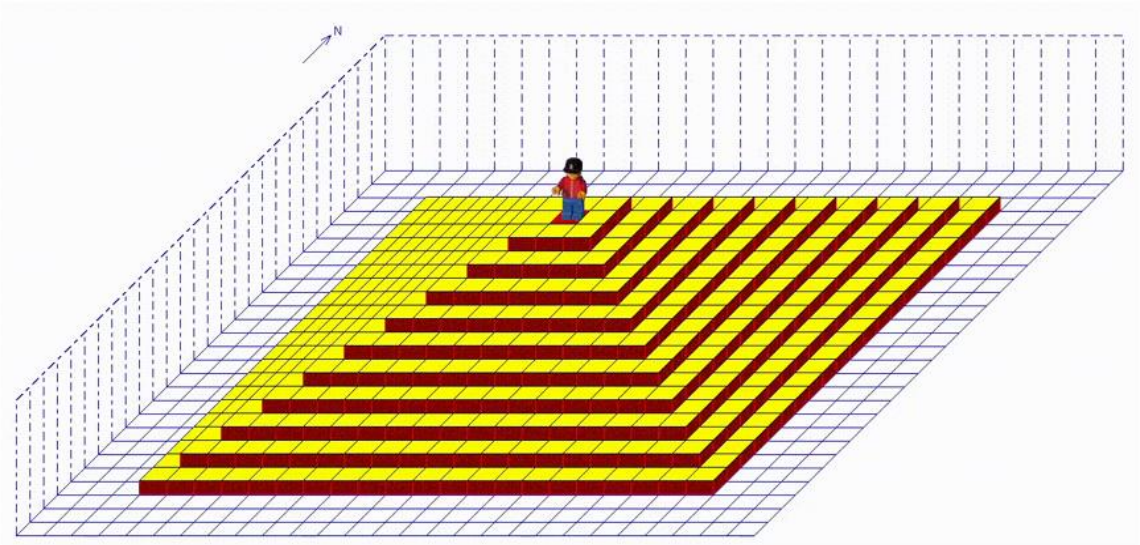


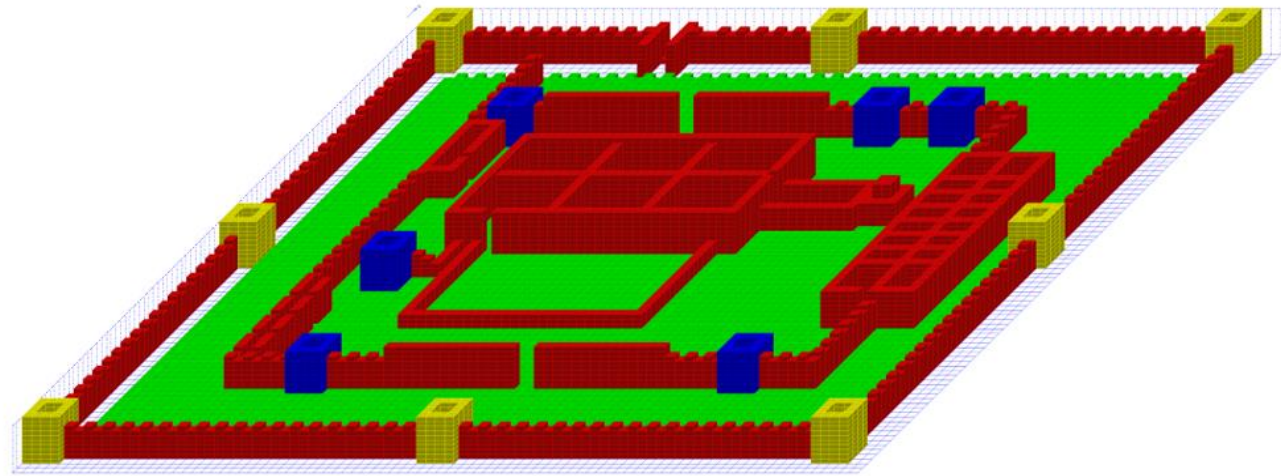
Ideen



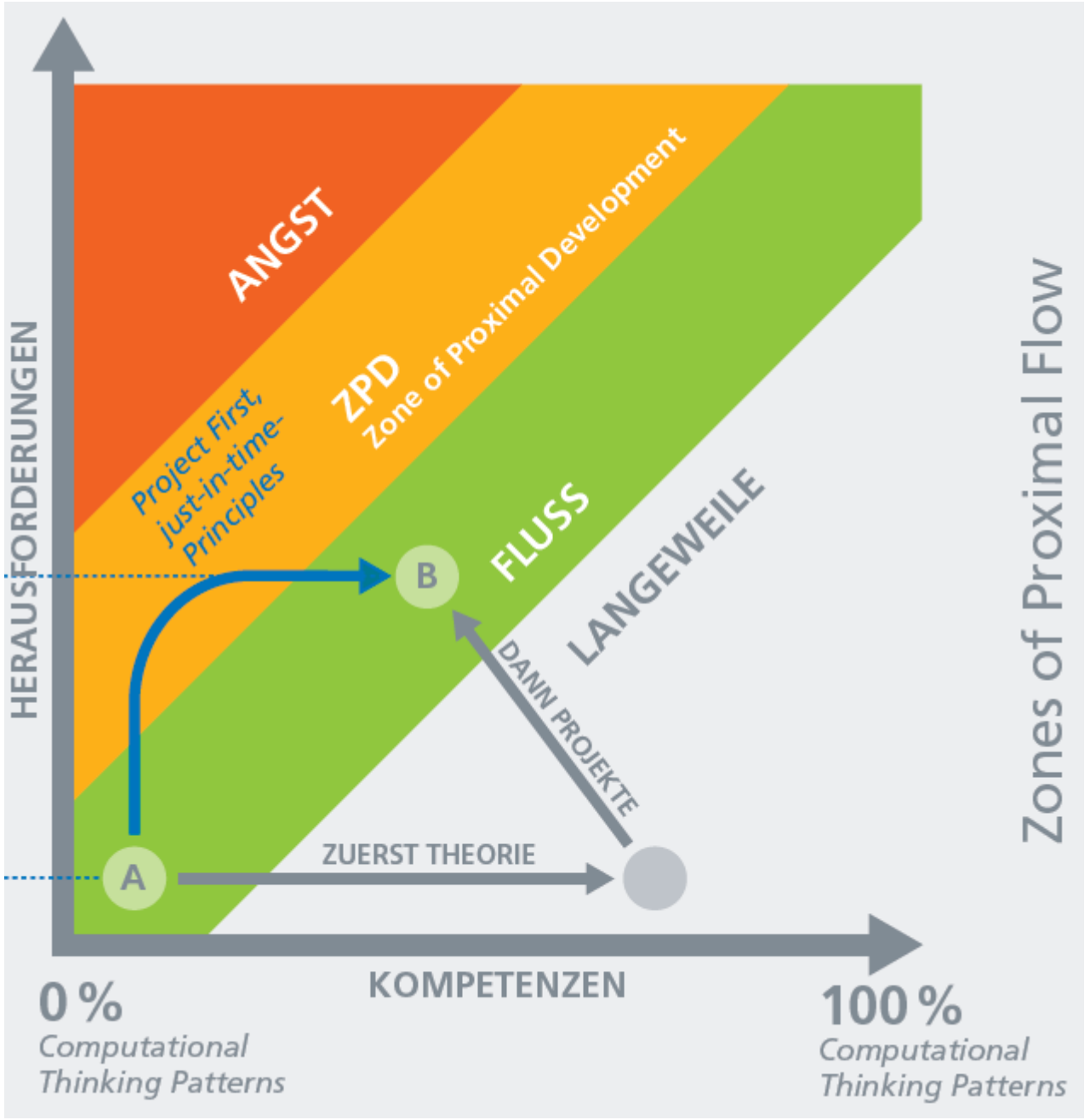


https://mebis.bycs.de/beitrag/robot-karol





Vorgehen

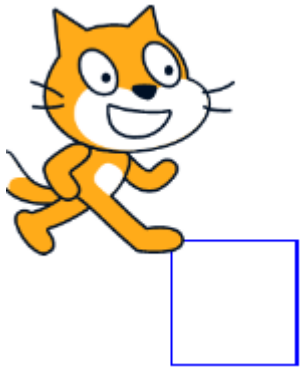

ANGST
ZPD
Zone of Proximal Development
Project First,
just-in-time-
Principles
FLUSS
LANGEWEILE
B
A
DANN PROJEKTE
ZUERST THEORIE
HERAUSFORDERUNGEN
KOMPETENZEN
0 %
Computational
Thinking Patterns
100 %
Computational
Thinking Patterns
Zones of Proximal Flow

Scratch & Snap!

- 2007 veröffentlicht
- 3-Klausel BSD-Lizenz (OpenSource)
- Windows, macOS, Linux
- läuft seit 2019 auch mit JavaScript im Webbrowser

- ca. 90.000.000 registrierte Accounts
- ca. 102.000.000 Projekte veröffentlicht

- wenige Features, für Kinder gedacht
  - z.B. Scratch Jr. Ab 5 Jahren

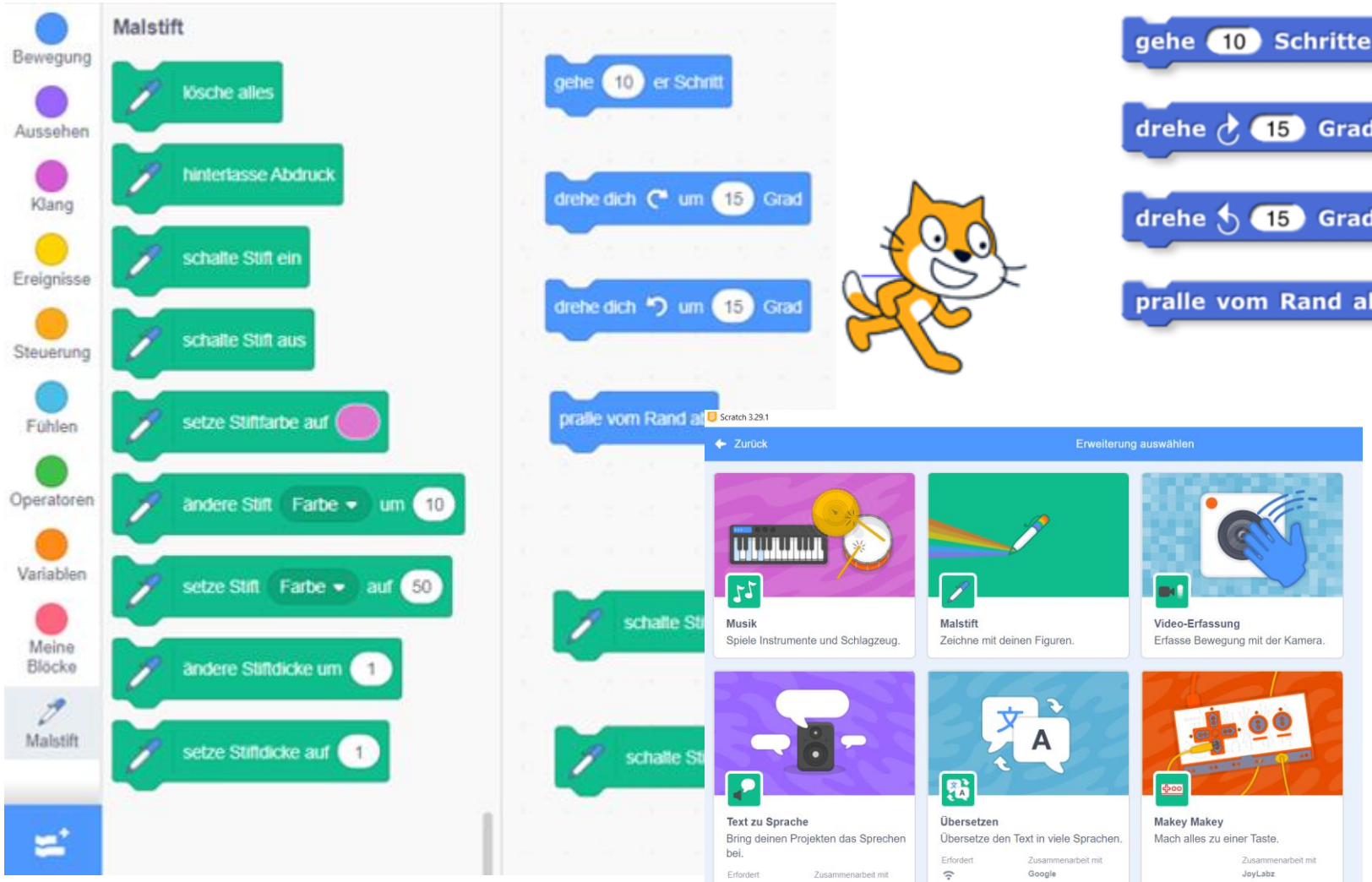


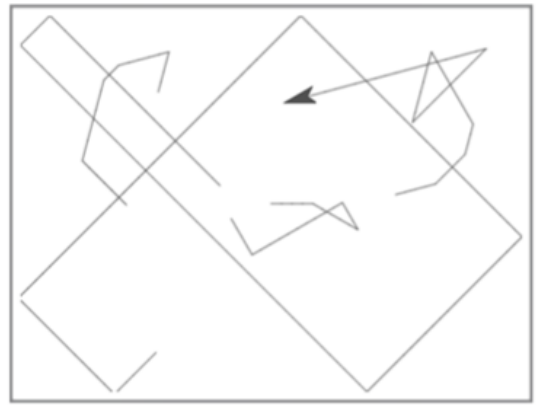
- Erscheinungsjahr 2011
- OpenSouce-Lizenz
- Windows, macOS, Linux
- läuft als BYOB im Webbrowser (JavaScript)

- ca. 600.000 registrierte Accounts
- ca. 5.000.000 Projekte

- Weiterentwicklung von Scratch mit mehr Features
  - BYOB = eigene Funktionen; Rekursion möglich
  - Listen, Sprites und Funktionen sind first-class, d.h. können in anderen Listen gespeichert werden
  - externe Ansteuerung möglich Lego, Arduino, …
  - ausgefeilte Parameterübergabe – mächtiger als manche Text-Programmierumgebung

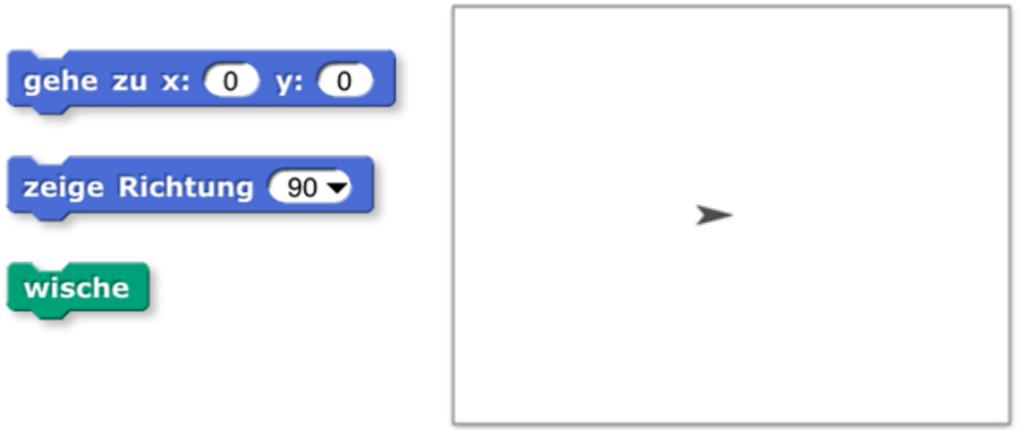
Direktmodus bei beiden Sprachen möglich





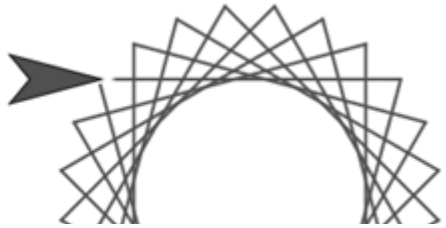
Malstift = Erweiterung (microbit, Lego, …)

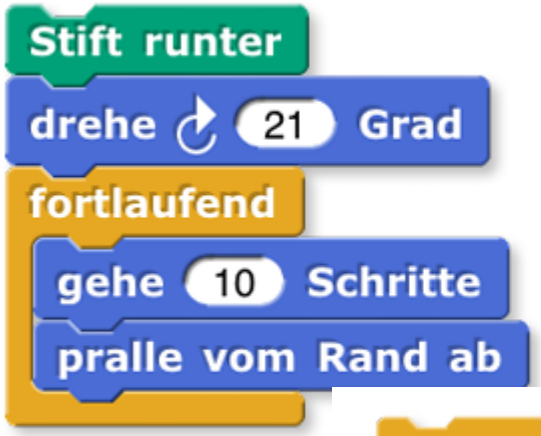
Zurücksetzen

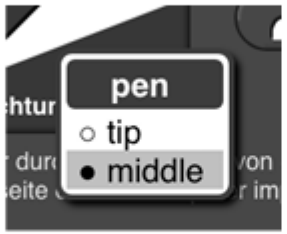


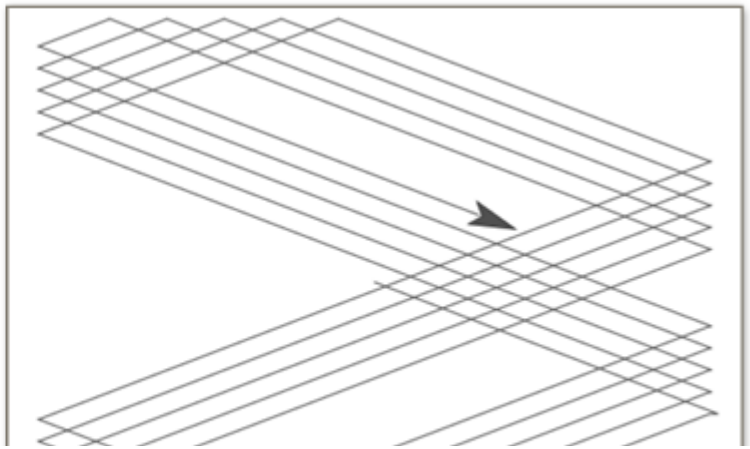
Schleifen


Stift runter
wiederhole 24 mal
gehe 100 Schritte
drehe 105 Grad

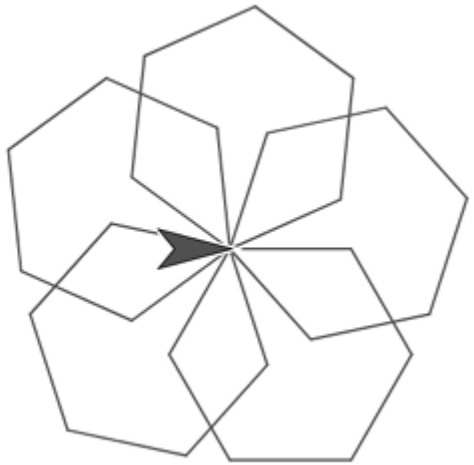



Stift runter
drehe 21 Grad
fortlaufend
gehe 10 Schritte
pralle vom Rand ab

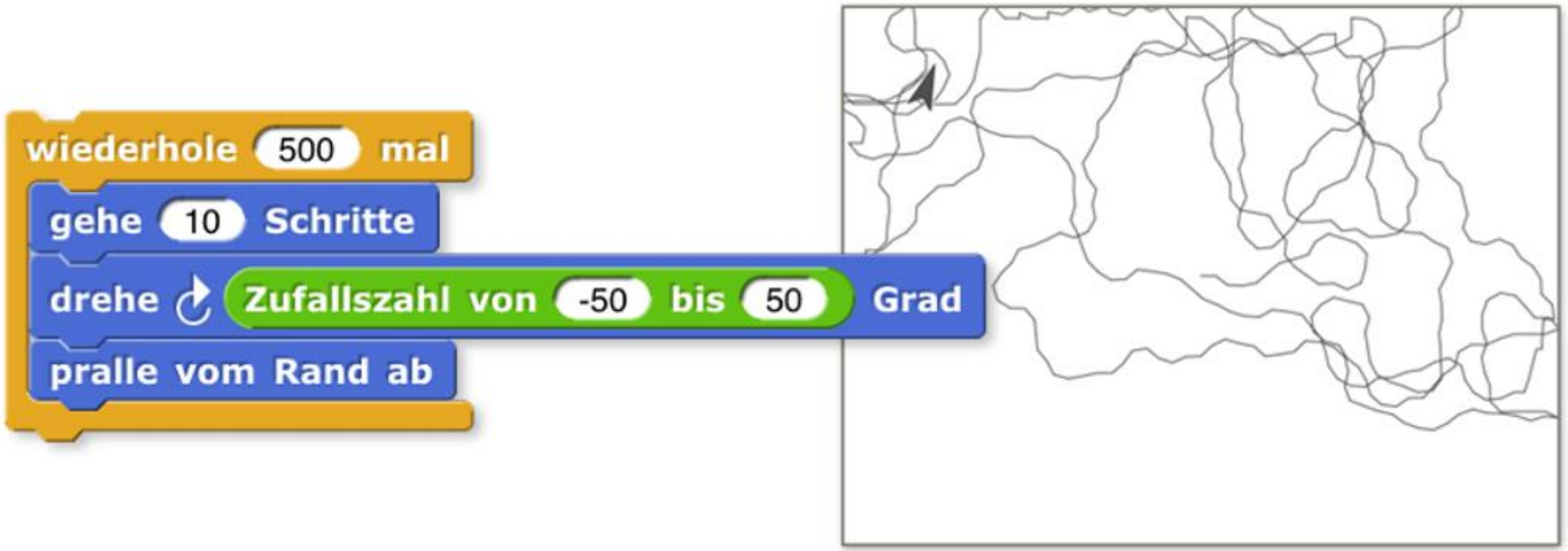

pen
tip
middle




wiederhole 5 mal
wiederhole 6 mal
gehe 50 Schritte
drehe 60 Grad
drehe 72 Grad

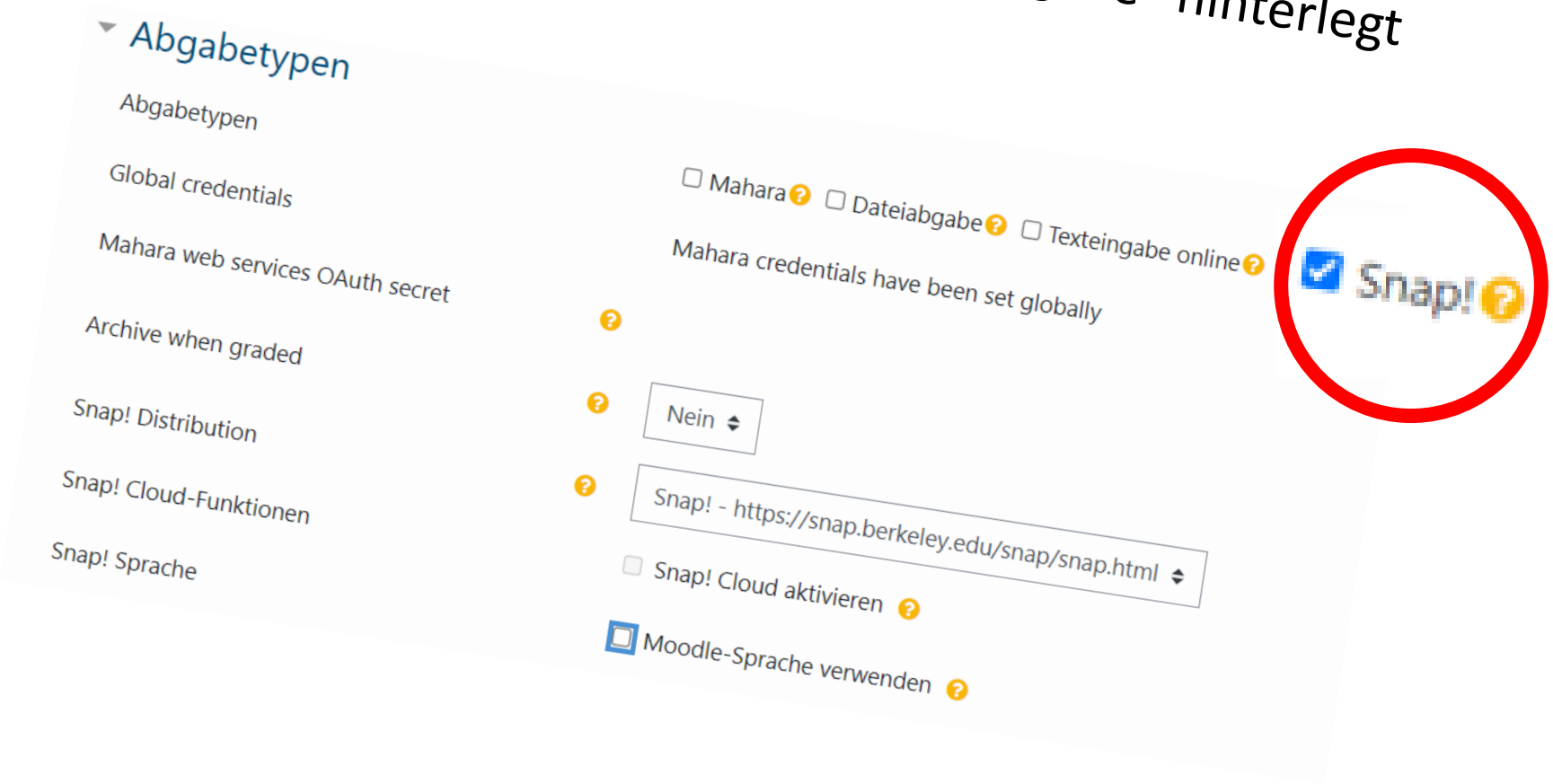


Zufall

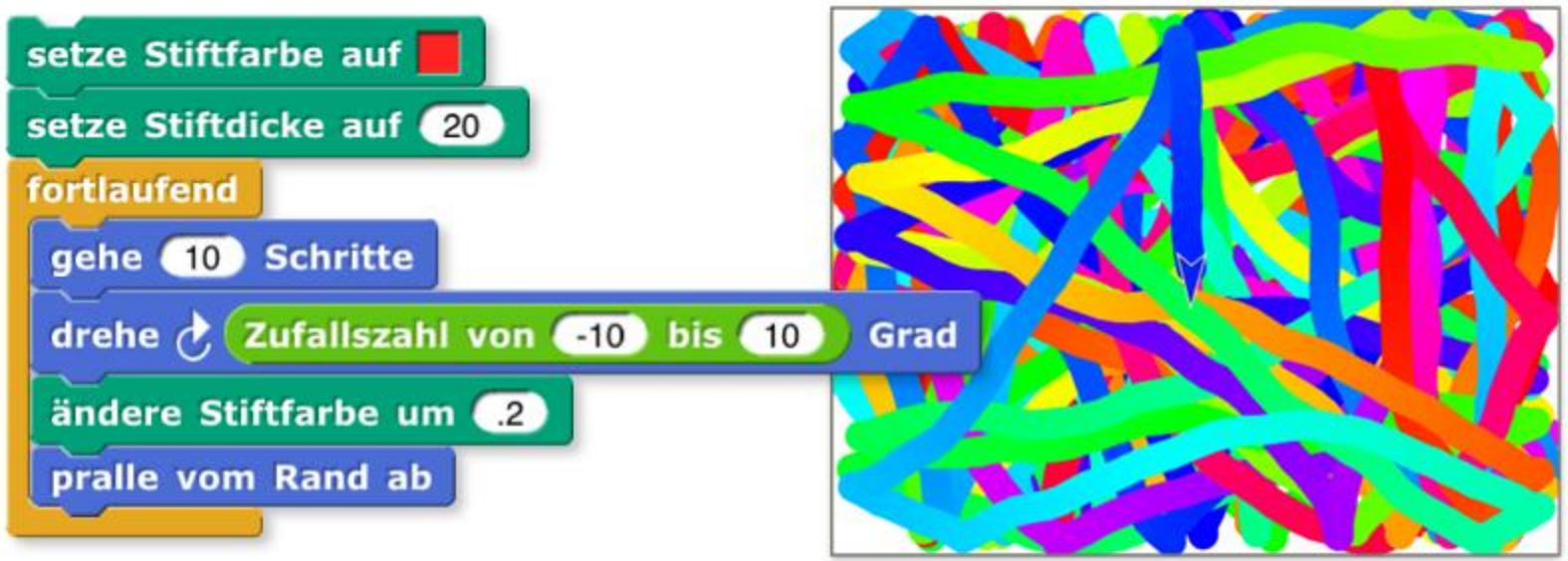

wiederhole 500 mal
gehe 10 Schritte
drehe Zufallszahl von -50 bis 50 Grad
pralle vom Rand ab

Wie funktioniert das?

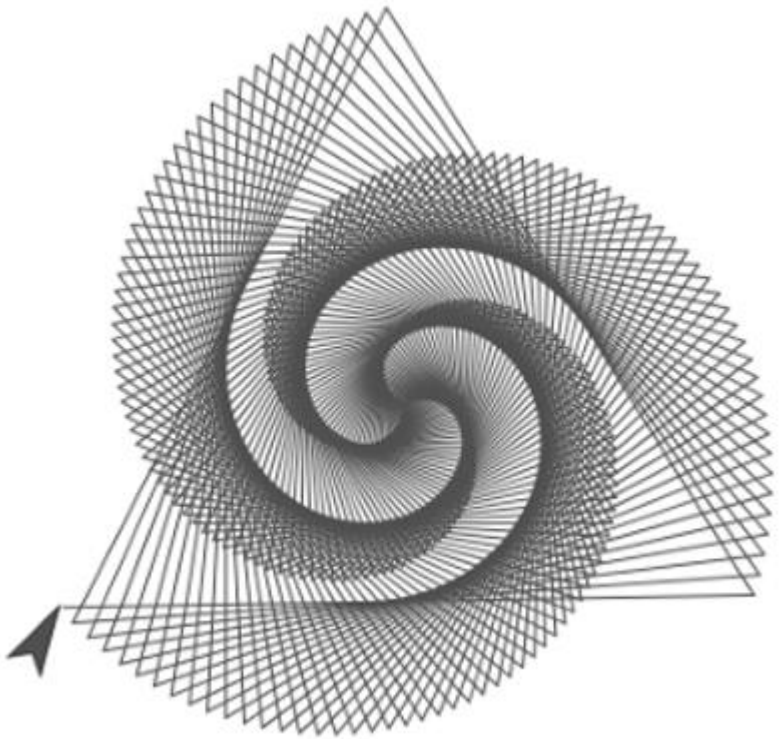
In TSNmoodle ist Snap! direkt in der Aktivität „Aufgabe“ hinterlegt

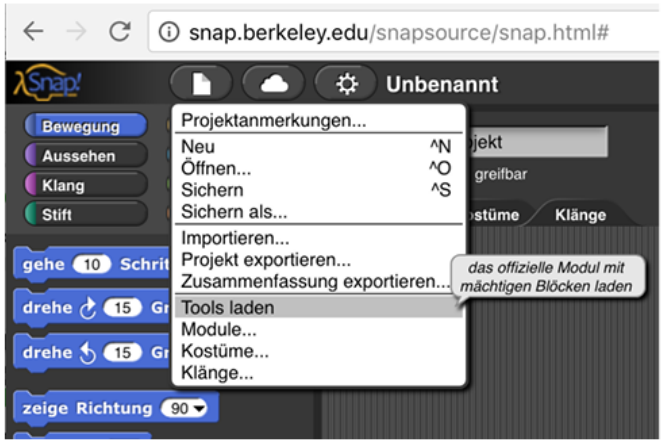


Farben


setze Stiftfarbe auf
setze Stiftdicke auf 20
fortlaufend
gehe 10 Schritte
drehe Zufallszahl von -10 bis 10 Grad
ändere Stiftfarbe um .2
pralle vom Rand ab

Mit Variablen – wie viele Zeilen?





Stempeln

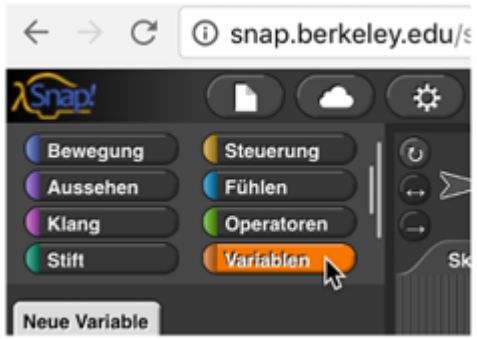

Skripte
Kostüme
Klänge
Richtungszeiger
Bilder durch hereinziehen von einer anderen Webseite or vom Computer importieren
Bearbeiten
Umbenennen
Duplizieren
Löschen
Exportieren
Unbenannt(2)

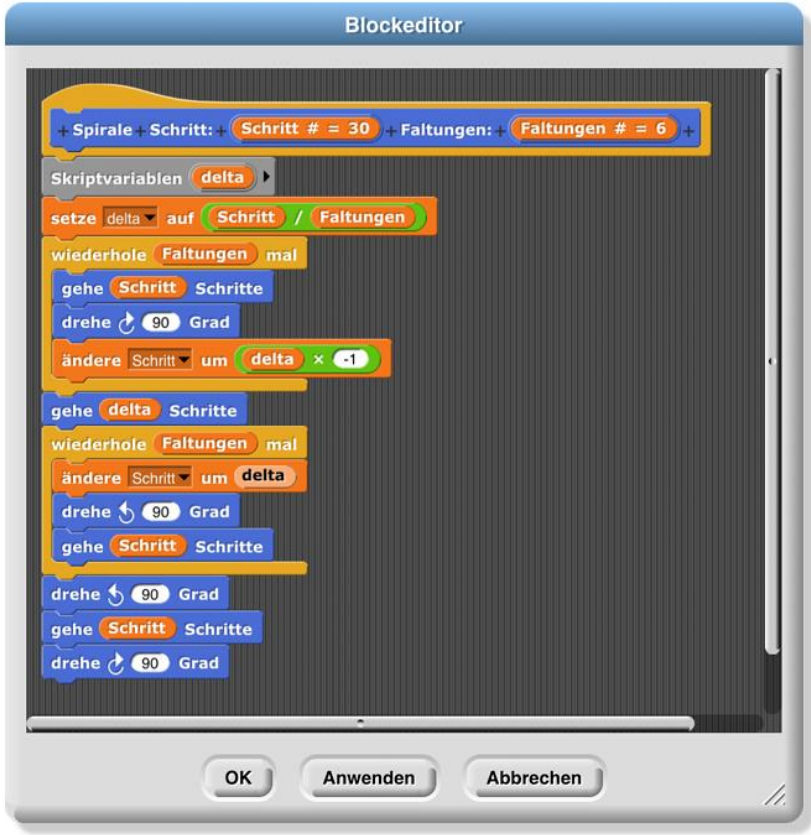

Wenn angeklickt
wische
Skriptvariablen Abstand
setze Abstand auf 30
setze y auf 160
wiederhole 6 mal
setze Größe auf 200 – y-Position × 0.8 %
setze x auf -200
wiederhole 20 mal
falls Zufallszahl von 1 bis 5 > 1
nächstes Kostüm
stemple
ändere x um Abstand
ändere y um 0 – Abstand
ändere Abstand um 20

ChatBot in Scratch

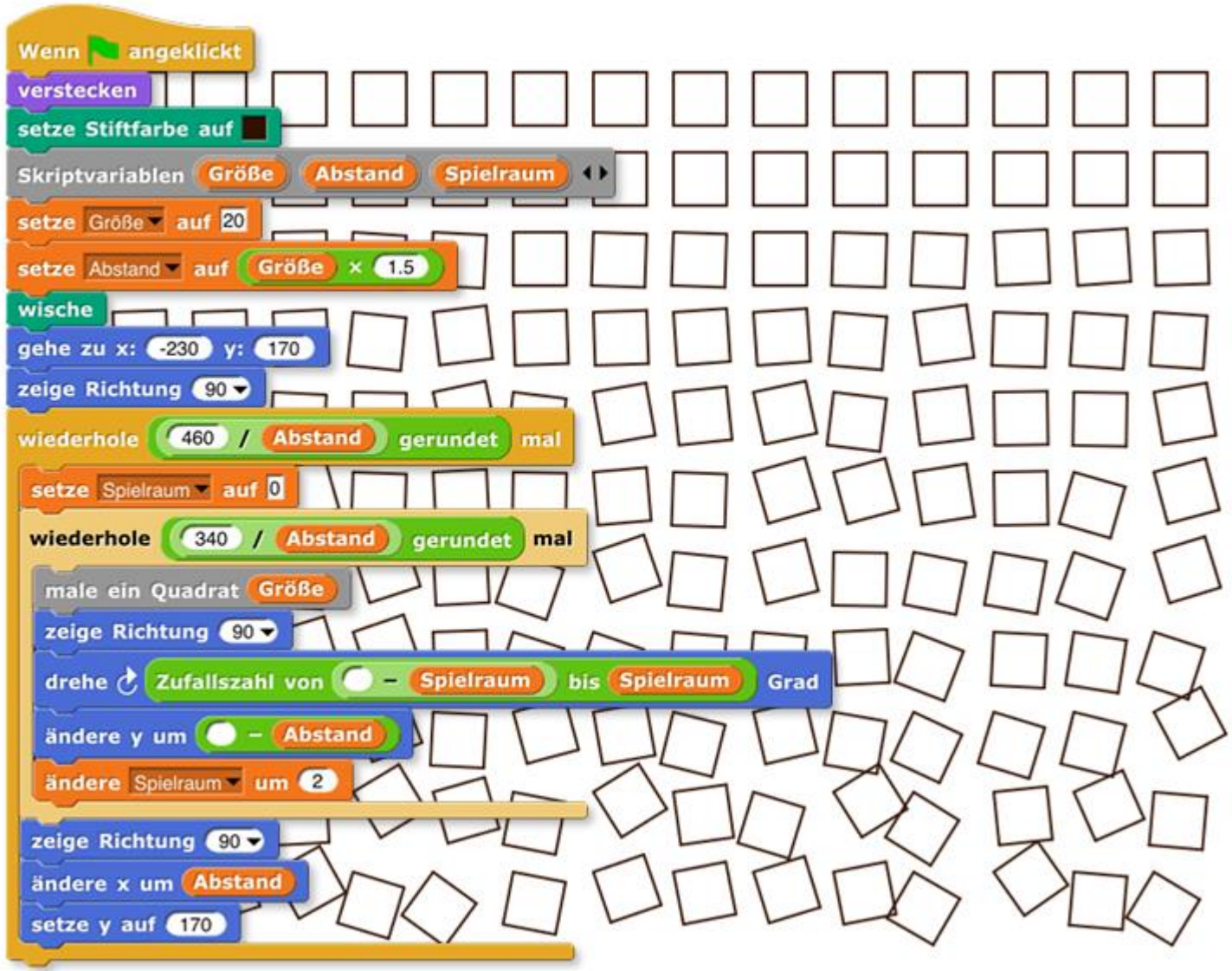
https://www.create-learn.us/blog/make-chatbot-with-ai-in-scratch/

Eigene Blöcke

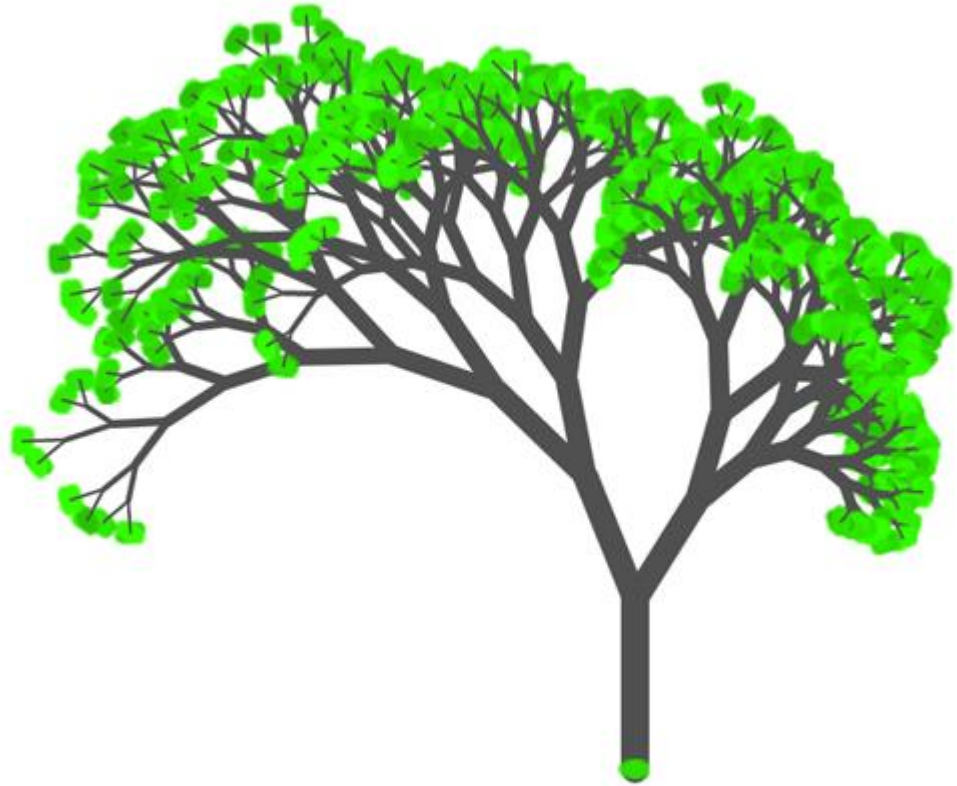




Eigene Blöcke



```
+ Baum + Level + [Level # = 3] + Länge + [Länge # = 80] +
falls [Level] > [0]
ziehe Kostüm [Richtungszeiger] an
setze Stiftdicke auf [Level]
Skriptvariablen [links] [rechts]
setze [links] auf [Zufallszahl von 15 bis 25]
setze [rechts] auf [Zufallszahl von 35 bis 60]
gehe [Länge] Schritte
drehe [links] Grad
Baum Level [Level – 1] Länge
[Länge] × [Zufallszahl von 0.7 bis 0.95]
drehe [rechts] Grad
Baum Level [Level – 1] Länge
[Länge] × [Zufallszahl von 0.5 bis 0.9]
drehe [rechts] Grad
drehe [links] Grad
gehe [Länge × -1] Schritte
sonst
ziehe Kostüm [Blatt] an
setze [Helligeit] -Effekt auf [Zufallszahl von 1 bis 30]
stemple
```

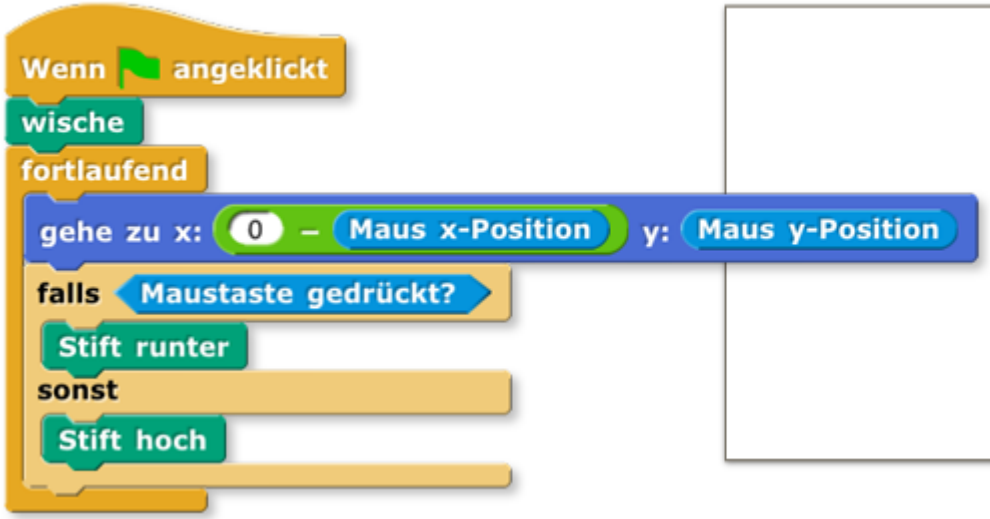


```
Wenn [flag] angeklickt
gehe zu x: [0] y: [-150]
zeige Richtung [0]
wische
Stift runter
Baum Level [10] Länge [60]
```

Interaktiv

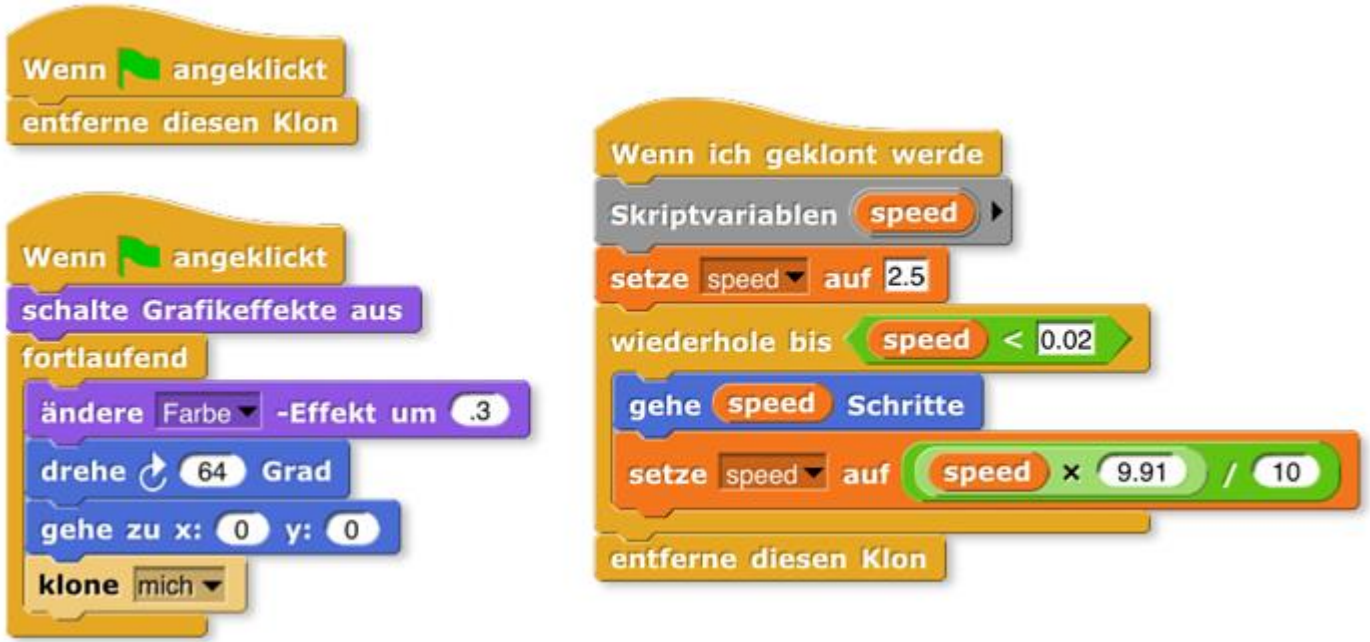

Wenn angeklickt
wische
fortlaufend
gehe zu Mauszeiger
falls Maustaste gedrückt?
Stift runter
sonst
Stift hoch

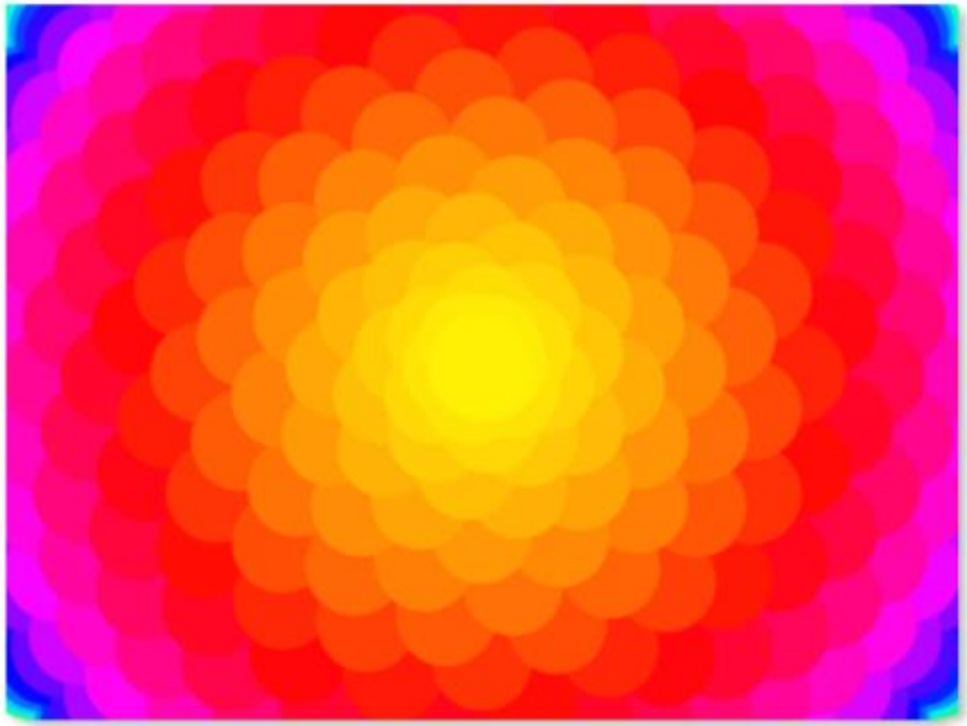

yolo


Wenn angeklickt
wische
fortlaufend
gehe zu x: 0 – Maus x-Position y: Maus y-Position
falls Maustaste gedrückt?
Stift runter
sonst
Stift hoch


yolo

Klonen


Wenn angeklickt
entferne diesen Klon
Wenn angeklickt
schalte Grafikeffekte aus
fortlaufend
ändere Farbe -Effekt um .3
drehe 64 Grad
gehe zu x: 0 y: 0
klone mich
Wenn ich geklont werde
Skriptvariablen speed
setze speed auf 2.5
wiederhole bis speed < 0.02
gehe speed Schritte
setze speed auf speed × 9.91 / 10
entferne diesen Klon




VIELE
N DANK
FUER D

Bild- und Quellennachweis

Seymour Papert: Folie 2 - https://de.wikipedia.org/wiki/Seymour_Papert#/media/Datei:Papert.jpg

Jeannette Wing: Folie 2 - https://de.wikipedia.org/wiki/Jeannette_Wing#/media/Datei:Jeannette_Wing,_Davos_2013.jpg

Wikipedia: Folie 3 - https://en.wikipedia.org/wiki/Computational_thinking

Swisscom.ch: Folie 5 - https://www.swisscom.ch/de/schulen-ans-internet/computational-thinking.html

chatGPT: Folie 4 - https://chat.openai.com/chat

ZPT: Folie 11 - Hasler Broschüre; Hasler Stiftung; Schriftenreihe Januar 2015; Computational Thinking in der Lehrerfortbildung

Kunst mit Snap! - https://eeducation.at/fileadmin/user_upload/Kunst_mit_Snap_JadgaHuegle.pdf

StarWars TextEngine: Folie 27 - https://scratch.mit.edu/projects/364079313

Robot Karol: https://mebis.bycs.de/beitrag/robot-karol