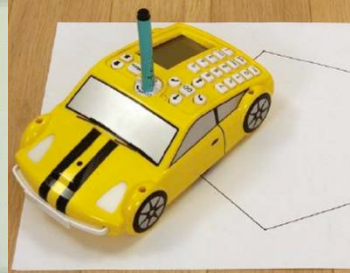
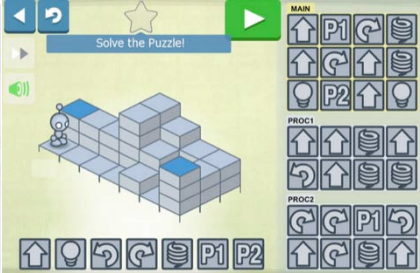


Programmierlernumgebungen für Kinder



Michael Hielscher
pädagogische hochschule schwyz

- 30 Minuten: Arten von Programmierumgebungen
 - 60 Minuten: selbst experimentieren an Posten
-

Programmierumgebungen

The image displays a programming environment with three main windows:

- Adobe Reader - [blueJ-tutorial-deutsch.pdf]**: Shows a PDF document titled "Das BlueJ Tutorial" (Version 2.0.1 für BlueJ Version 2.0.x) by Michael Kölling, Merik Inuttine, and Matthias Langlotz.
- Liste**: A code editor window showing the implementation of a `Liste` class. The code includes comments in German and Java code for a doubly-linked list structure.
- BlueJ: DoppeltVerketteteListe**: A class diagram window showing the relationship between `Knoten`, `Liste`, and `AwtOberflaeche`. A context menu is open over the `Liste` class, listing methods such as `Object raus()`, `void rein(Object neuerInhalt)`, and `void weiter()`.

```
// Für IF12G - Krall, 03/06 bzw. 04/06

public class Liste
{
    Knoten vorkopf, aktuell; //Zeiger vorkopf zeigt immer auf den
    // Listenanfang (=dummyV, s.u); Zeiger aktuell wählt ein Element
    // der Liste aus

    // 1. Allgemeine Methoden: a. Konstruktor, b. Kontrolle, ob leer

    public Liste()
    {
        Knoten dummyV = new Knoten(); //vorderes Dummy-Element
        Knoten dummyH = new Knoten(); //hinteres Dummy-Element

        dummyV.voriger = null;
    }
}
```

Programmierumgebungen

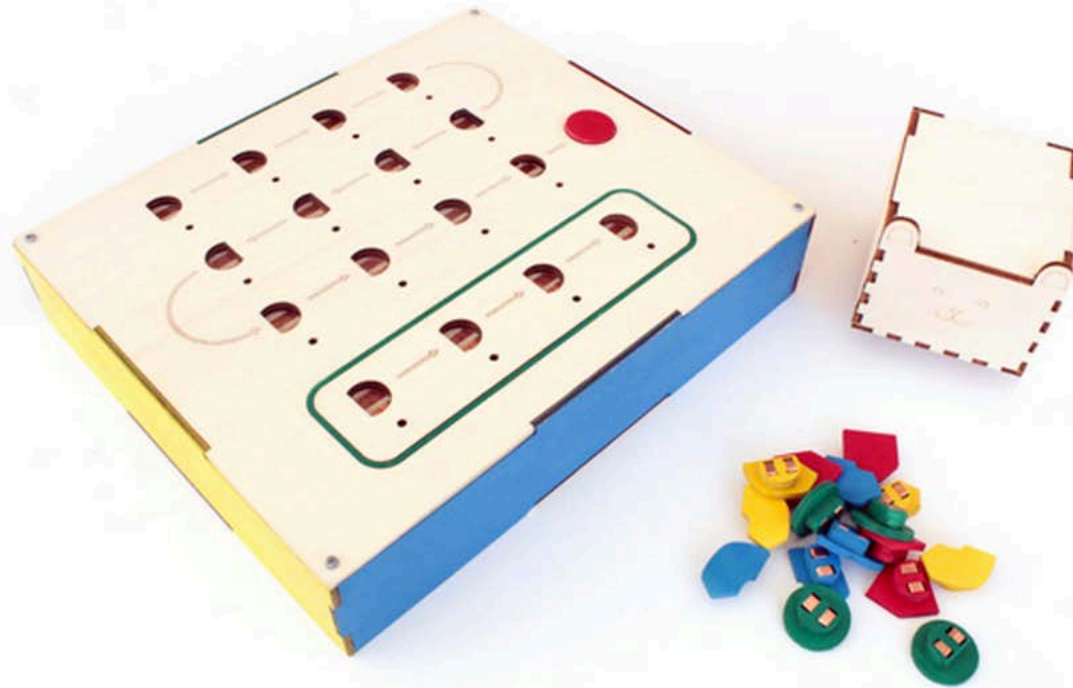
The image displays the XLogo programming environment. The main window shows a simple polygon on the left and a complex star pattern on the right. The star pattern is composed of green stars connected by blue and red lines, with a turtle icon at the center. The command window at the bottom left contains the following code:

```
vorwärts 30  
links 90  
vorwärts 120  
vorwärts 10  
vorwärts 20  
vorwärts 20  
links 90  
vorwärts 10
```

The editor window at the bottom right shows the following code:

```
to stern :seitenlänge :farbe  
setpc :farbe  
repeat 6 [ fd :seitenlänge lt 60 fd :seitenlänge rt 120 ]  
end
```

Programmierumgebungen



Taktile Programmierumgebung: [Primo.io](https://primo.io)

Programmierumgebungen



Brettspiele zum Thema Programmieren: RobotTurtles

Was davon verwenden?

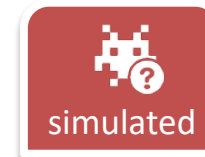
Klassifikation von Programmier-Lernumgebungen



1. Repräsentationskriterium



2. Interaktivitätskriterium



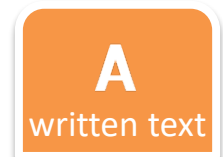
3. Koordinationskriterium



4. Ausführungskriterium



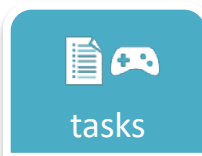
5. Notationskriterium



6. Mächtigkeitskriterium

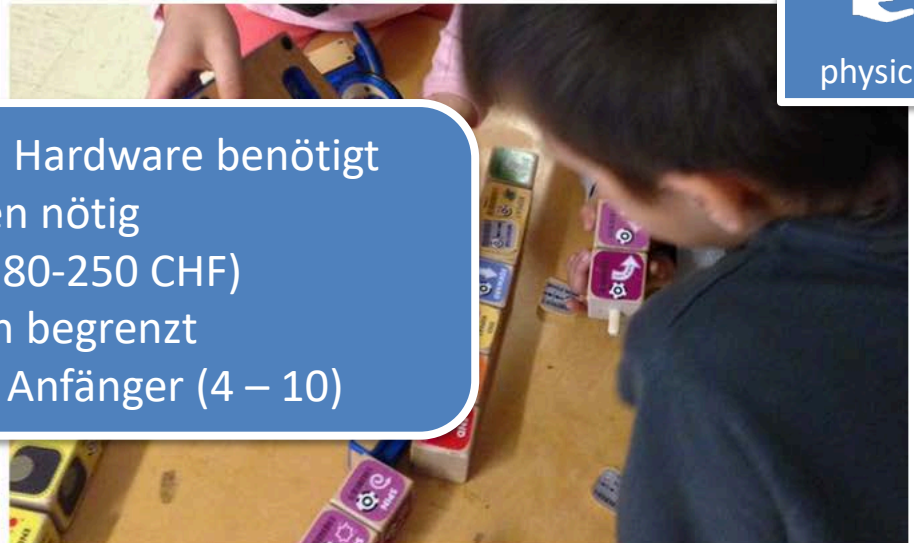


7. Didaktisierungskriterium





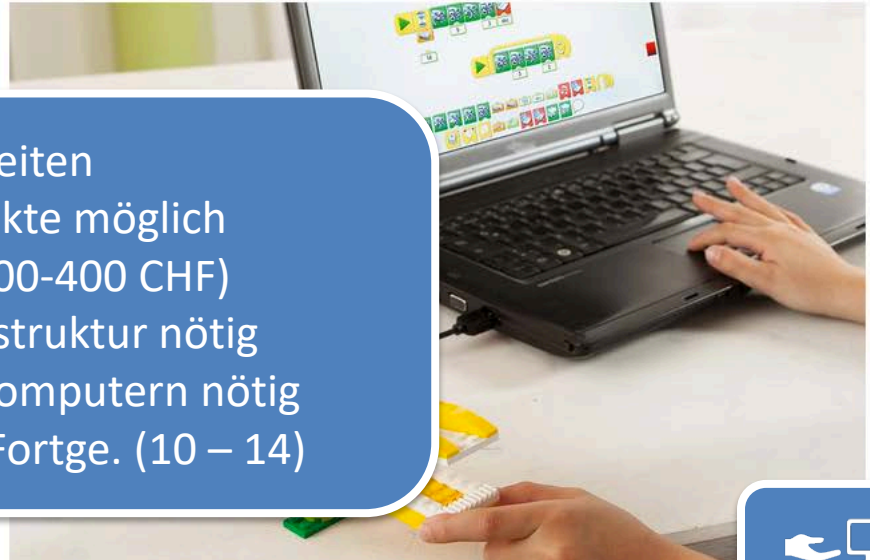
physical



physical

keine weitere Hardware benötigt
kein Vorwissen nötig
relativ teuer (80-250 CHF)
Möglichkeiten begrenzt
=> primär für Anfänger (4 – 10)

Enaktive physische Roboter: Bee-Bots (links), KIBO mit Bauklötzchen (rechts)



mehr Möglichkeiten
komplexe Projekte möglich
relativ teuer (100-400 CHF)
zusätzlich Infrastruktur nötig
Vorwissen zu Computern nötig
=> Anfänger + Fortge. (10 – 14)

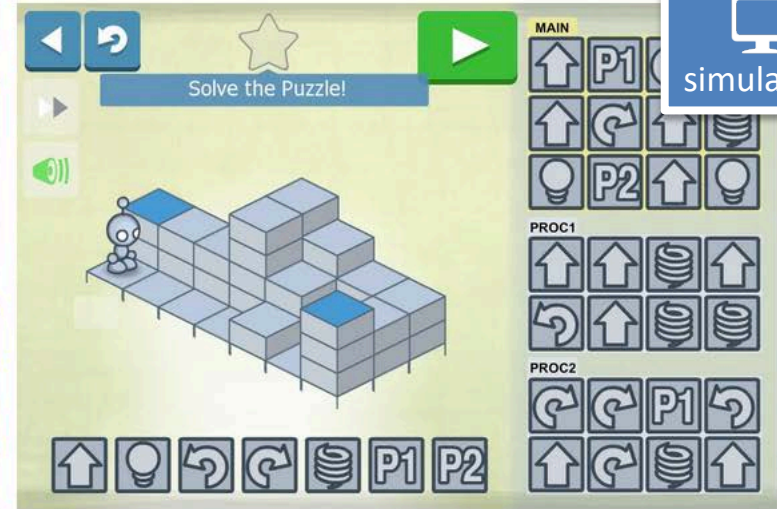
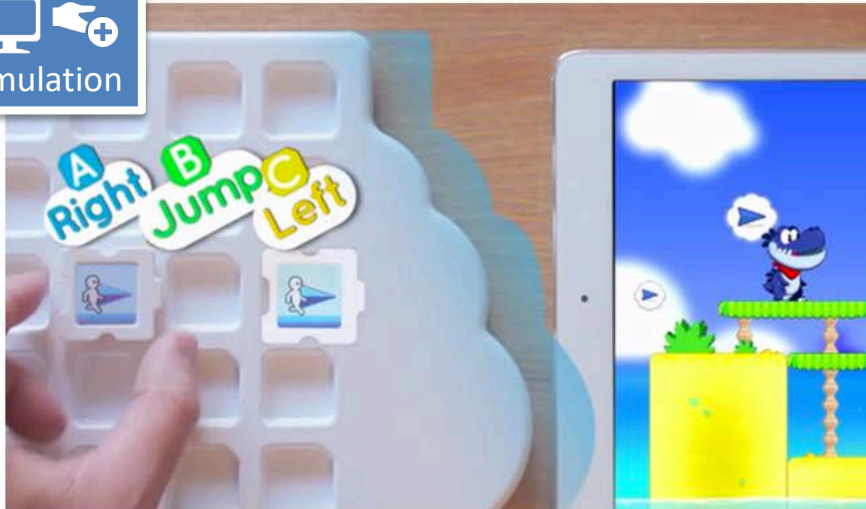


physical



physical

Enaktive physische Roboter mit Programmierung am Computer:
Thymio II (links), Lego WeDo (rechts),

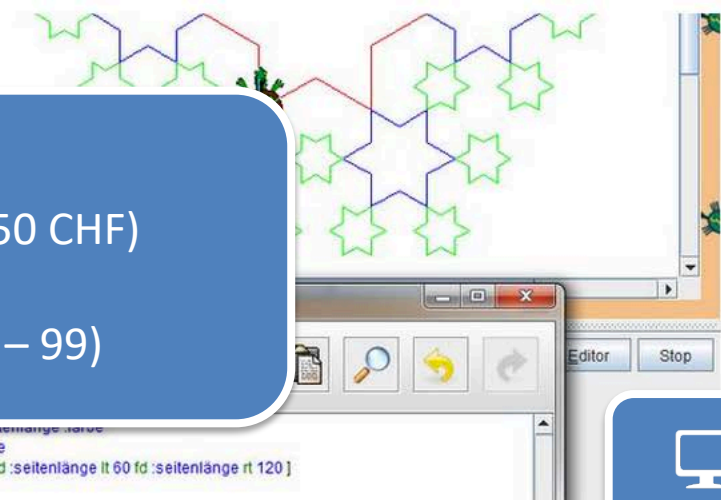


virtuelle Roboter mit ikonischer Programmierung

Puzzlets mit physischer Eingabe (links), Lightbot mit virtueller Tastatur (rechts)



viele Möglichkeiten
häufig sehr günstig (0-50 CHF)
ikonisch, symbolisch
=> ganzes Spektrum (6 – 99)



virtuelle Roboter mit symbolischer, textueller Programmierung ohne physische Komponenten: Codecombat.com (links), XLogo (rechts)



1. Repräsentationskriterium



2. Interaktivitätskriterium



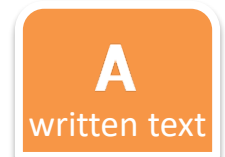
3. Koordinationskriterium



4. Ausführungskriterium



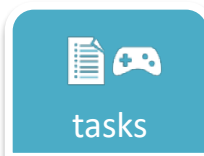
5. Notationskriterium



6. Mächtigkeitskriterium



7. Didaktisierungskriterium





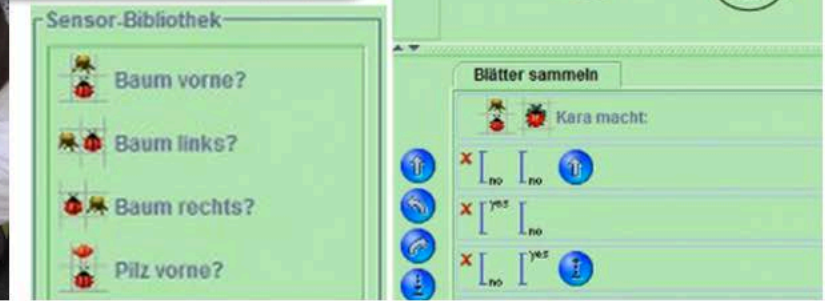
none



eventuell kann die „sture“ Variante für den Anfänger einfacher sein und das Konzept klarer aufzeigen.



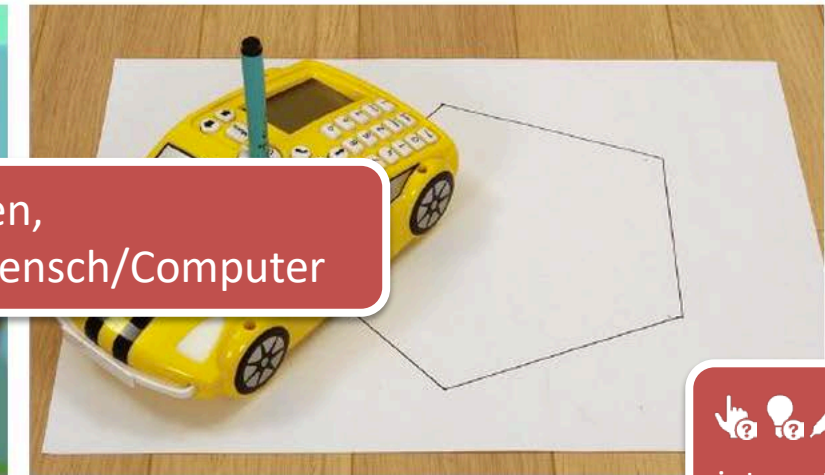
simulated



keine Sensoren im BeeBot (links), simulierte Sensoren in Kara (rechts)



mehr Möglichkeiten, Zusammenspiel Mensch/Computer



interaktiv



interaktiv

Tastatursensor in Kodu (links), Licht- und Berührungssensor im ProBot (rechts)

1. Repräsentationskriterium



2. Interaktivitätskriterium



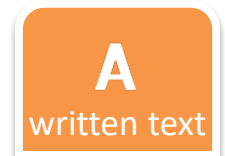
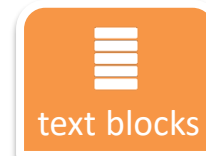
3. Koordinationskriterium



4. Ausführungskriterium



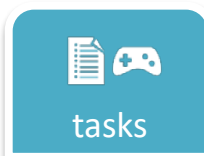
5. Notationskriterium



6. Mächtigkeitskriterium

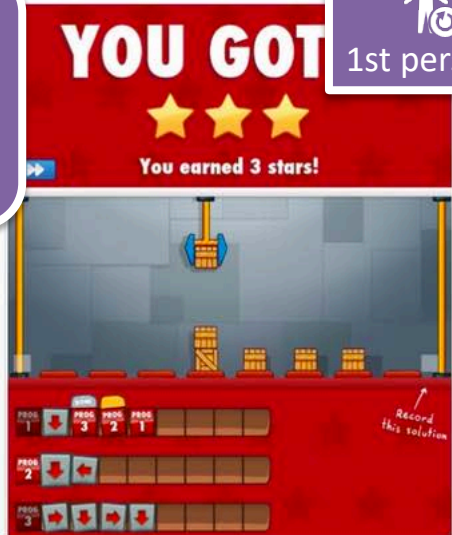
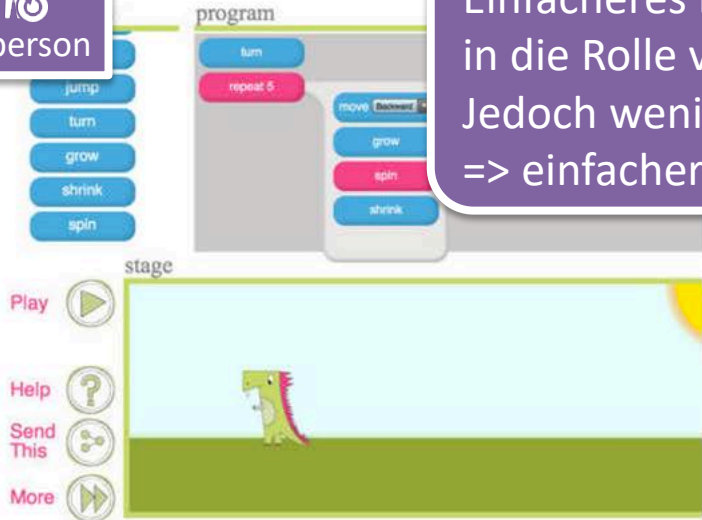


7. Didaktisierungskriterium

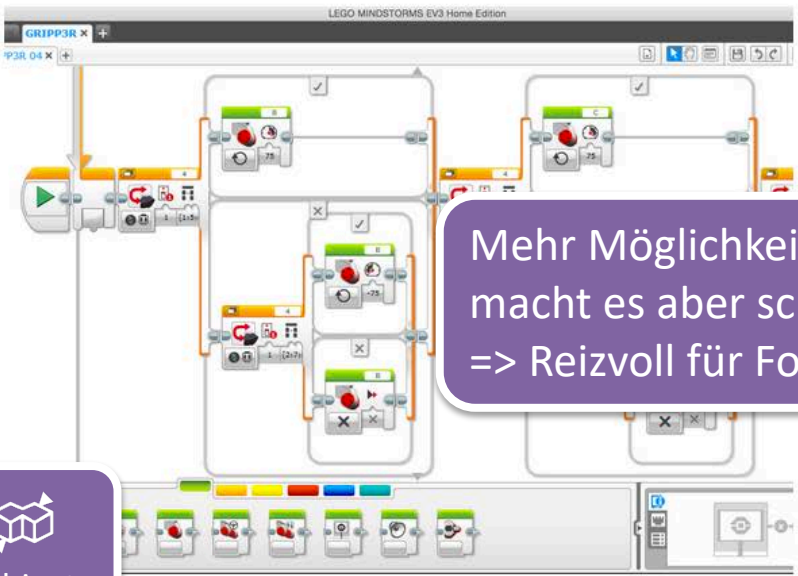




Einfacheres Debugging (sich selbst in die Rolle versetzen)
 Jedoch weniger Möglichkeiten
 => einfacher für Anfänger



1st-person am Beispiel Daisy the Dinosaur (links) und CargoBot (rechts)



Mehr Möglichkeiten, Koordination macht es aber schwieriger
 => Reizvoll für Fortgeschrittene



Programmierung mehrerer Motoren und Sensoren in Lego Mindstorms (links) und Multikara mit mehreren Käfern (rechts)



1. Repräsentationskriterium



2. Interaktivitätskriterium



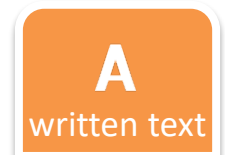
3. Koordinationskriterium



4. Ausführungskriterium



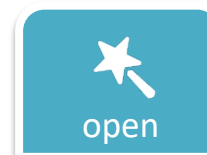
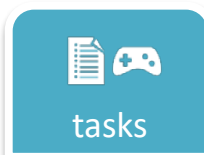
5. Notationskriterium



6. Mächtigkeitskriterium



7. Didaktisierungskriterium





sequential



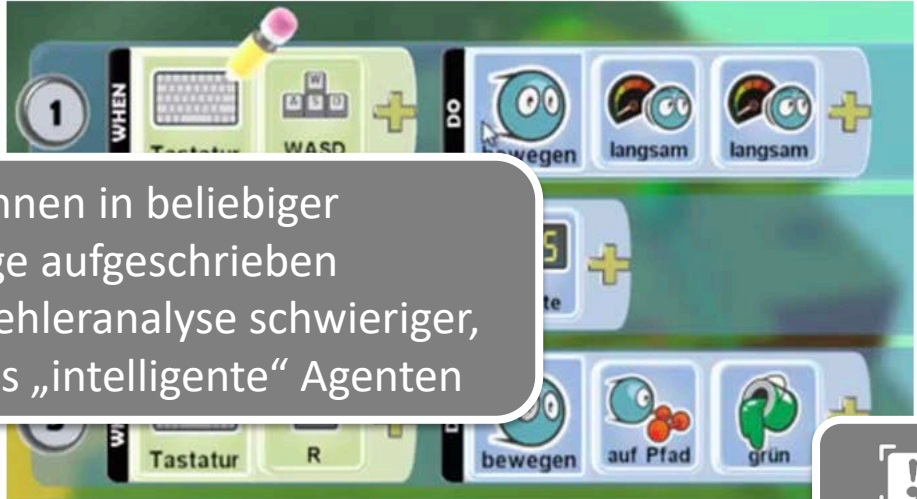
Programm lässt sich einfach nachspielen, debugging mit Schritt für Schritt Analyse



seq./evt.



Sequentielle Programmierung bei RoboTurtles (links) und sequentielle Programmierung mit Sensor-Events beim ProBot (rechts)



Regeln können in beliebiger Reihenfolge aufgeschrieben werden, Fehleranalyse schwieriger, Roboter als „intelligente“ Agenten



evt./seq.

Event-basierte Programmierung mit sequentiellen Teilen in Scratch (links) und ausschließlich Event-basierte Programmierung mit Kodu (rechts)



events

1. Repräsentationskriterium



2. Interaktivitätskriterium



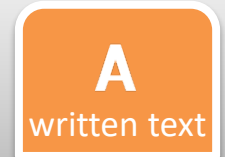
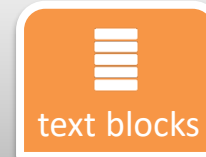
3. Koordinationskriterium



4. Ausführungskriterium



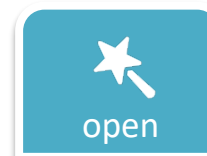
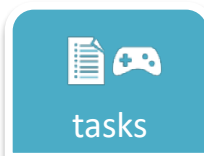
5. Notationskriterium

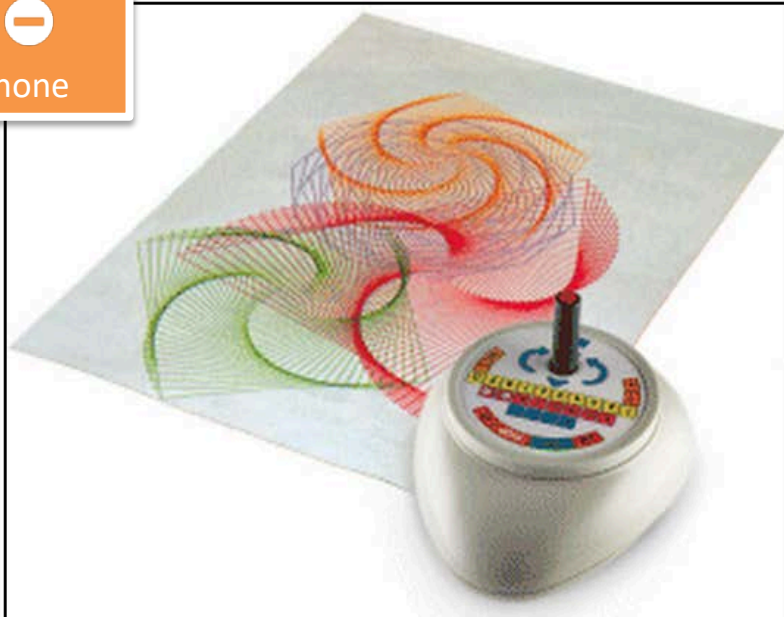


6. Mächtigkeitskriterium

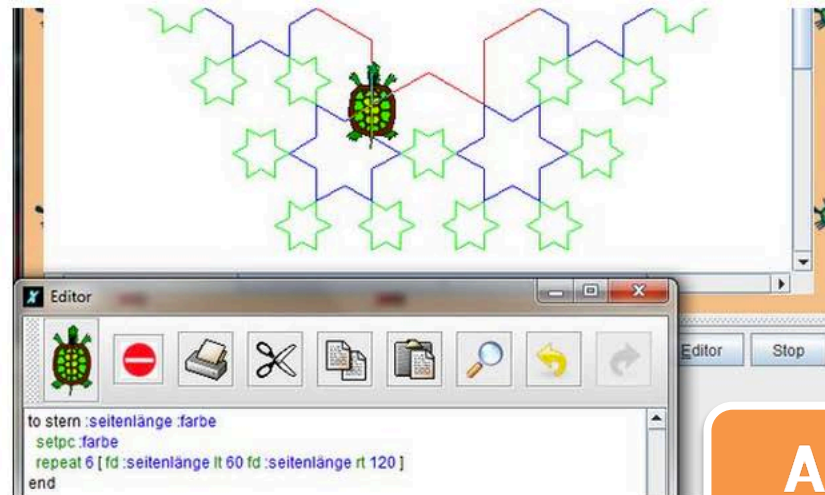


7. Didaktisierungskriterium





Ohne Notation: RoamerToo (links) und Icon-Notation: ScratchJR (rechts)



Text blocks: Blockly auf code.org (links) und textuelle Programmierung bei XLogo (rechts)

1. Repräsentationskriterium



2. Interaktivitätskriterium



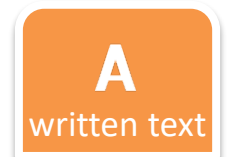
3. Koordinationskriterium



4. Ausführungskriterium



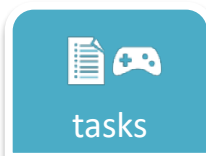
5. Notationskriterium



6. Mächtigkeitskriterium



7. Didaktisierungskriterium





Bedingte Anweisung

Sequenz

Prozeduren / Unterprogramme

Wiederholung

Rekursion

Objektorientierung

Datentypen

Variablen

1. Repräsentationskriterium



2. Interaktivitätskriterium



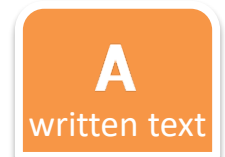
3. Koordinationskriterium



4. Ausführungskriterium



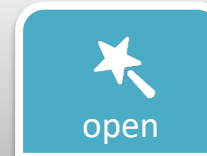
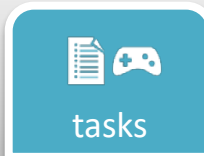
5. Notationskriterium

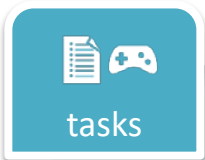


6. Mächtigkeitskriterium

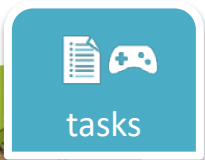
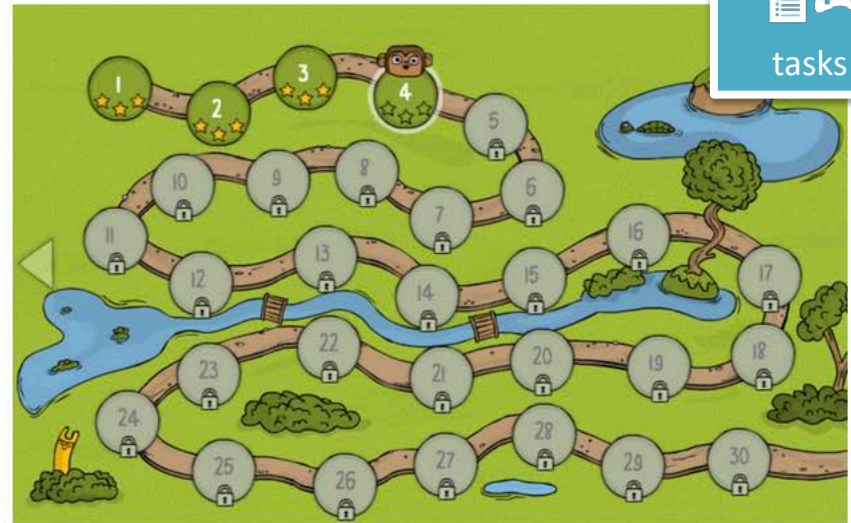


7. Didaktisierungskriterium



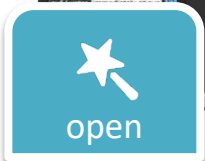
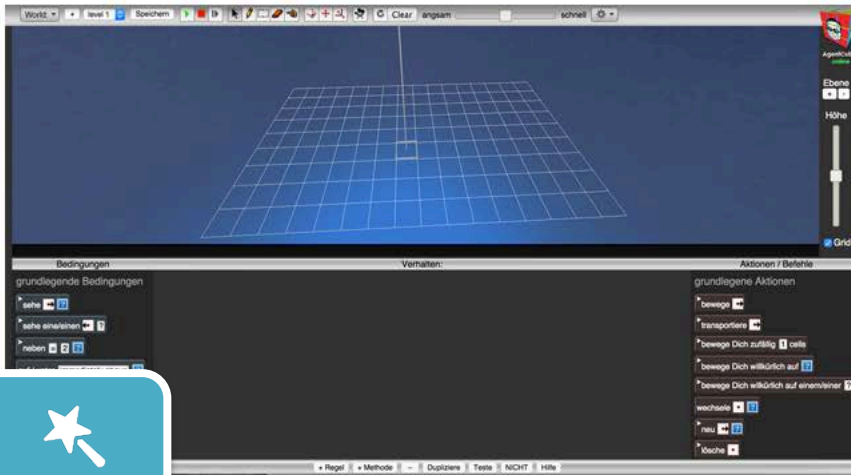


tasks

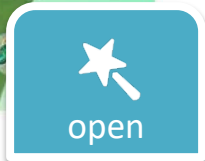
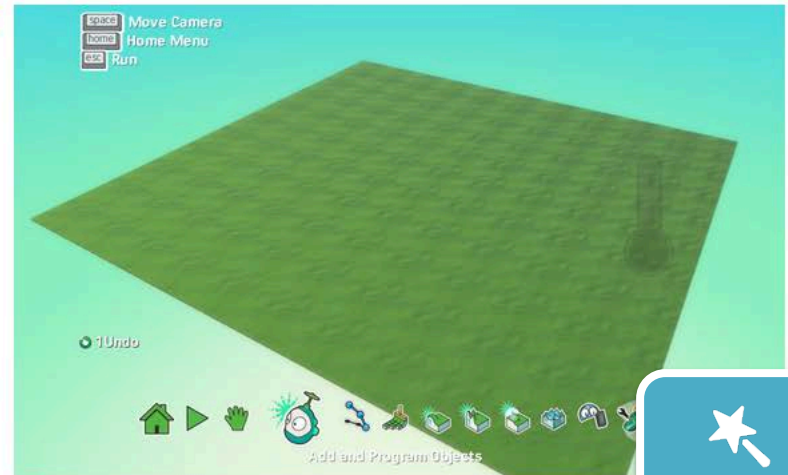


tasks

Aufgabenauswahl bei Lightbot (links) und CodeMonkey (rechts)



open











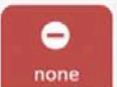






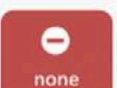



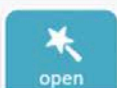

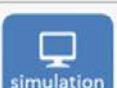
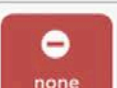
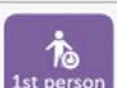




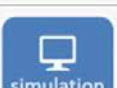

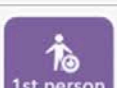











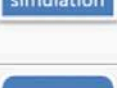
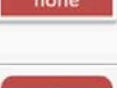
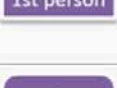
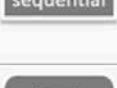

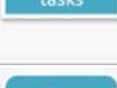

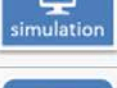
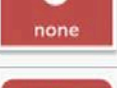
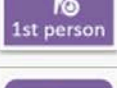
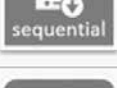
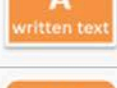
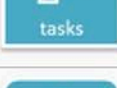

open

Freie Spielwelt für kreative Projekte bei AgentCubes (links) und Kodu (rechts)

<http://programmingwiki.de/Lernumgebungen>

Lernumgebungen

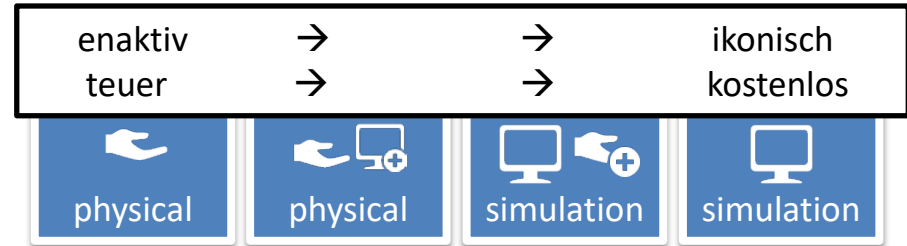
Die folgende Tabelle zeigt exemplarisch, wie die Kriterien angewendet werden können. Es besteht kein Anspruch auf Vollständigkeit oder einer trennscharfen Einordnung. Einige Lernumgebungen bieten Erweiterungen oder Zusätze und könnten unter deren Berücksichtigung entsprechend anders eingeordnet werden. Die Stufenempfehlungen basieren auf den Annahmen: Grundschule = 1-4 Klasse, Sekundarstufe I = 5-10 Klasse, Sekundarstufe II = 11-13 Klasse.

Lernumgebung	Repräsentation	Interaktivität	Koordination	Ausführung	Notation	Didaktisierung	Mächtigkeit	Stufenempfehlung
AgentCubes								Sekundarstufe I + II
Bee-Bot								Kindergarten / Grundschule
Blue-Bot								Kindergarten / Grundschule
Cargo-Bot								Sekundarstufe I + II
Code.org								Grundschule / Sekundarstufe I + II
Codecombat.com								Sekundarstufe I + II
CodeMonkey								Sekundarstufe I
Daisy the dinosaur								Kindergarten / Grundschule

Klassifikation von Programmier-Lernumgebungen

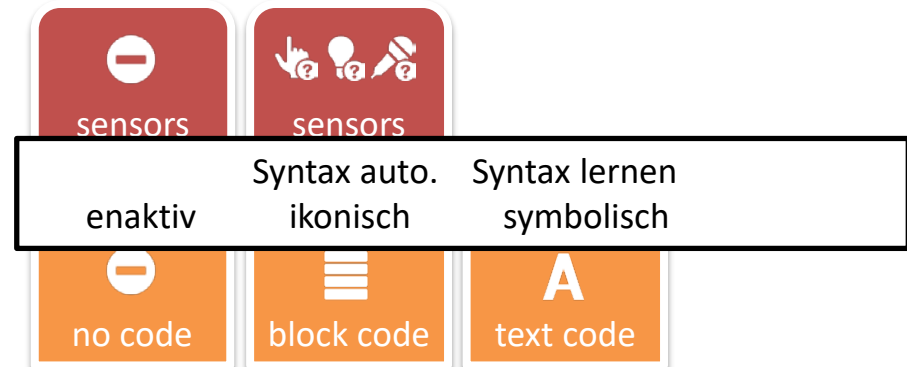
Robot vs. Robot-Simulator:

Ist das zu programmierende Objekt ein reales oder ein simuliertes? Braucht es einen Computer zum Programmieren?



Real-World-Sensors:

Kann man das Programm von außen beeinflussen?
Zum Beispiel durch Tastendruck, Geräusch machen, Helligkeit ändern ...



Representation of Code:

Wie wird ein entwickeltes Programm dargestellt, als Block-Diagramm, als Quelltext oder überhaupt nicht?



1st Person vs. n-Objects Programming:

Ist man praktisch selbst das zu programmierende Objekt und führt gedanklich die Bewegungen selbst aus bevor man sie aufschreibt?



Sandbox vs. Tasks & Game Levels :

Bietet die Lernumgebung fertige Levels oder Aufgaben, die abgearbeitet werden sollen oder stellt sie eine leere Welt zum experimentieren dar?



Posten zum Erkunden

