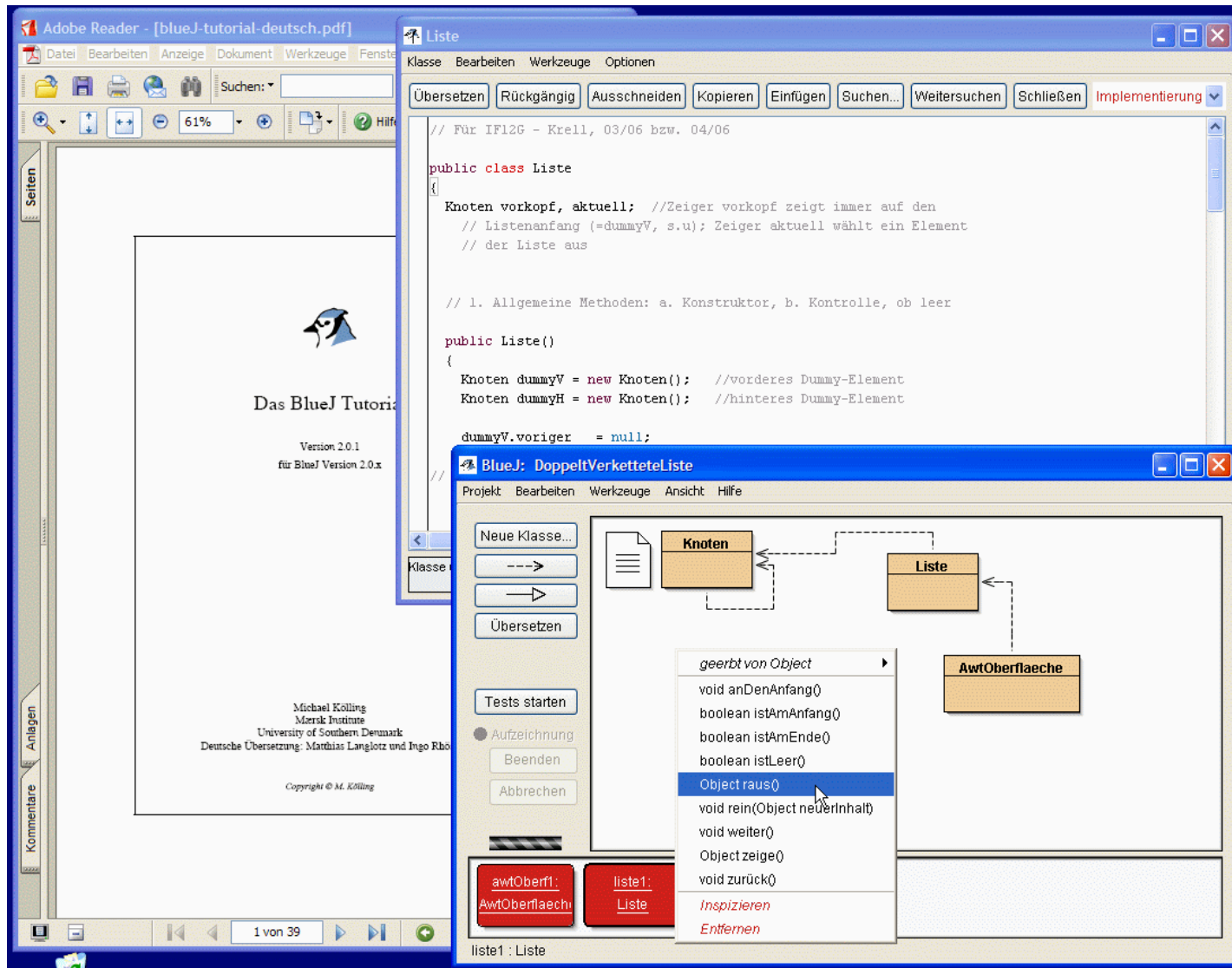


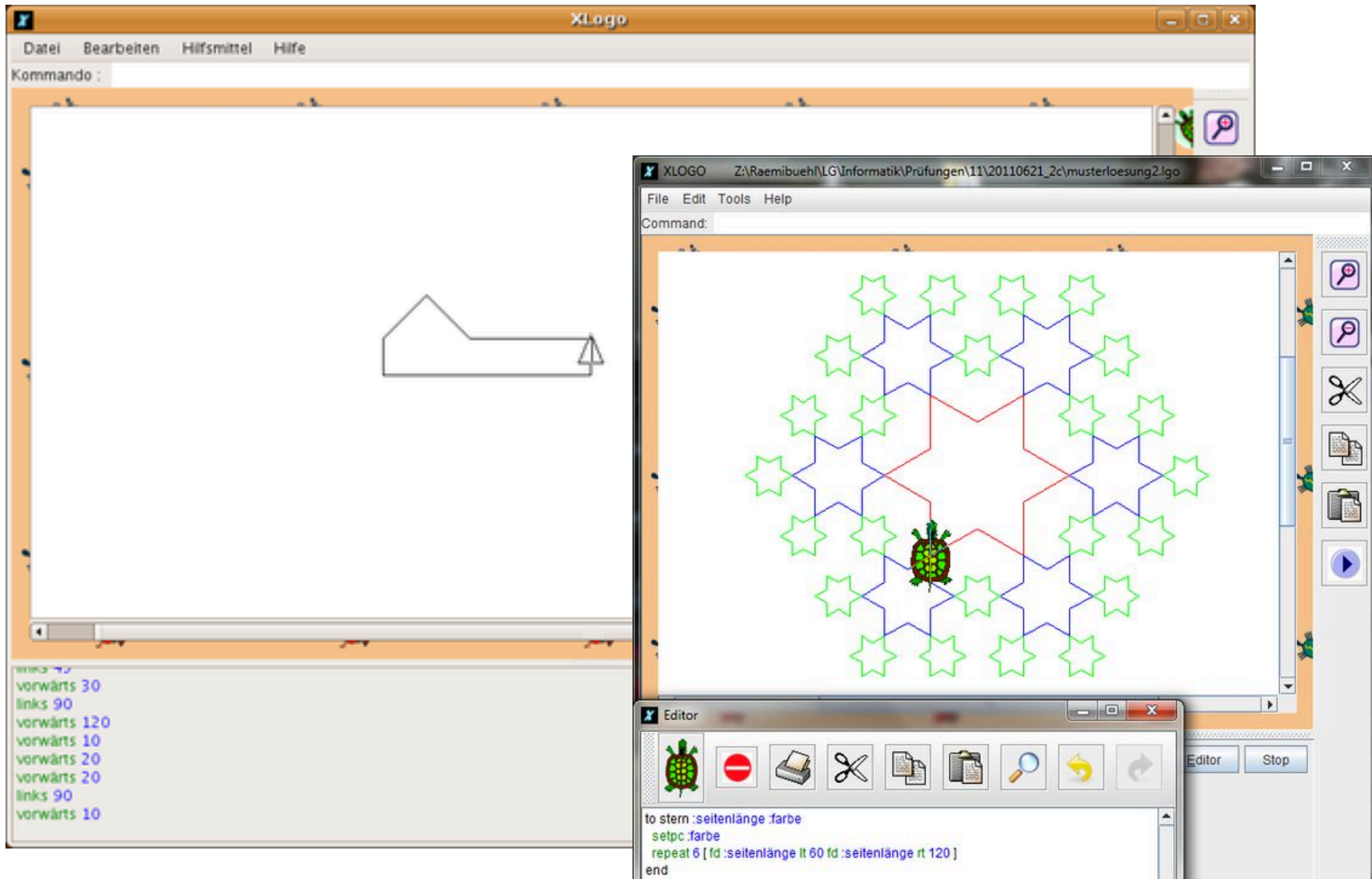
# Programmierungsumgebungen für Kinder



# Programmierumgebungen



# Programmierumgebungen



# Programmierumgebungen



Taktile Programmierumgebung: Primo.io



# Programmierumgebungen



Brettspiele zum Thema Programmieren: RobotTurtles

# Was davon verwenden?

## Klassifikation von Programmier-Lernumgebungen



## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



## 3. Koordinationskriterium



## 4. Ausführungskriterium



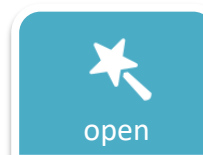
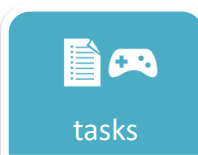
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium

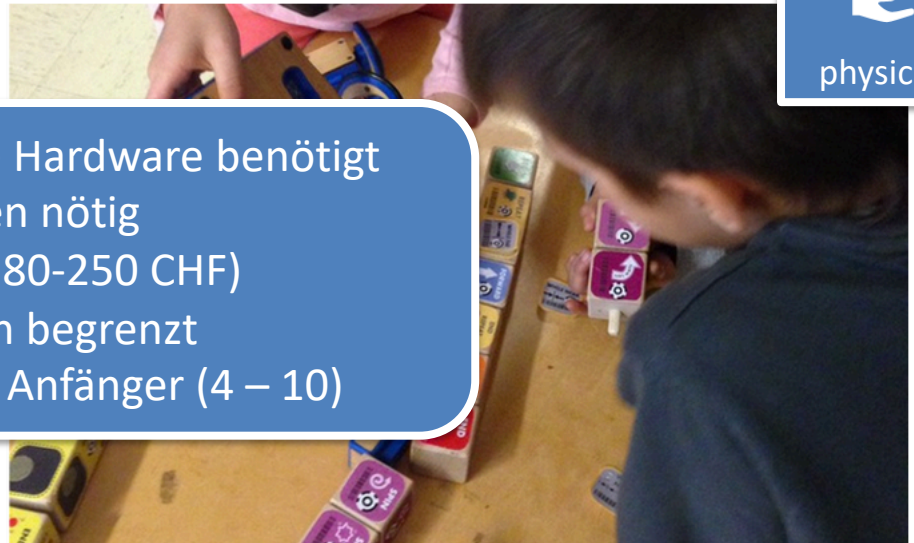




physical



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

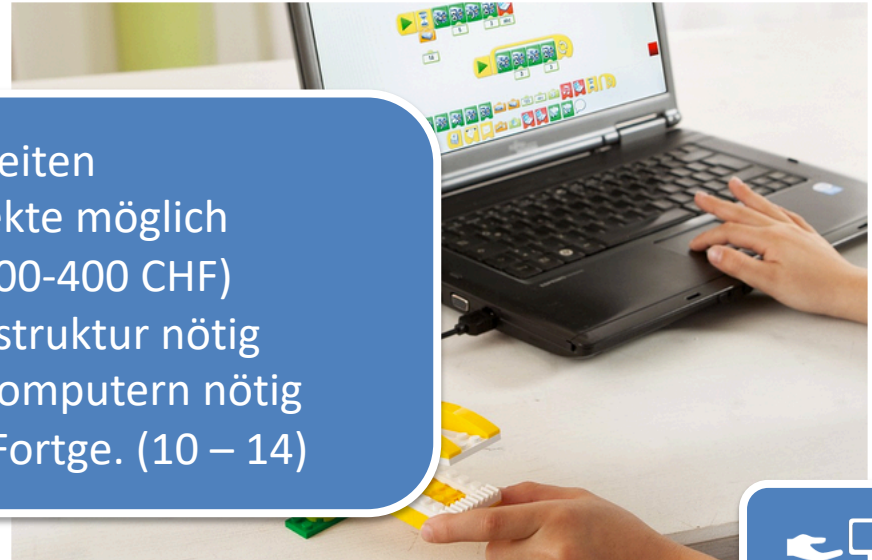


physical

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)



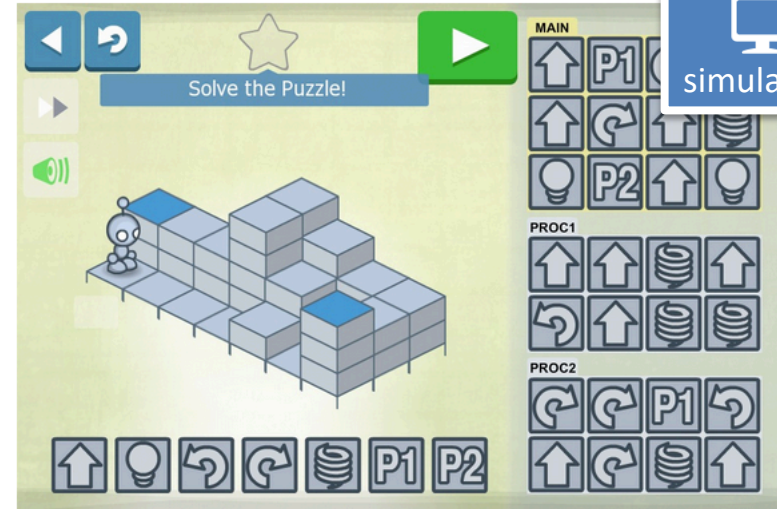
physical



physical

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

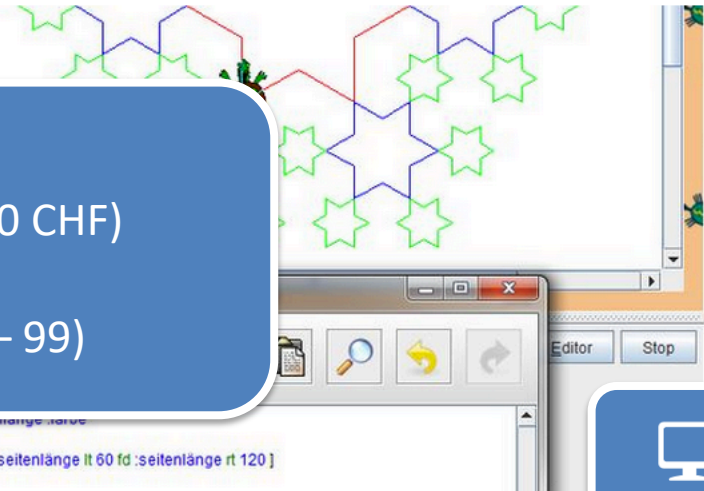




virtuelle Roboter mit ikonischer Programmierung  
Puzzlets mit physischer Eingabe (links), Lightbot mit virtueller Tastatur (rechts)

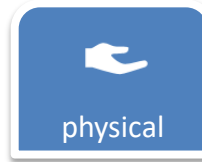


viele Möglichkeiten  
häufig sehr günstig (0-50 CHF)  
ikonisch, symbolisch  
=> ganzes Spektrum (6 – 99)



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

## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



## 3. Koordinationskriterium



## 4. Ausführungskriterium



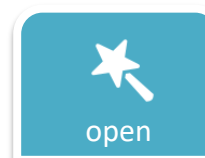
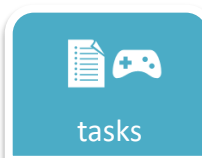
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium





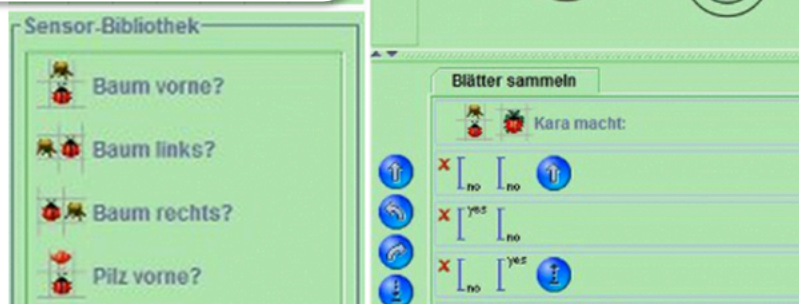
none



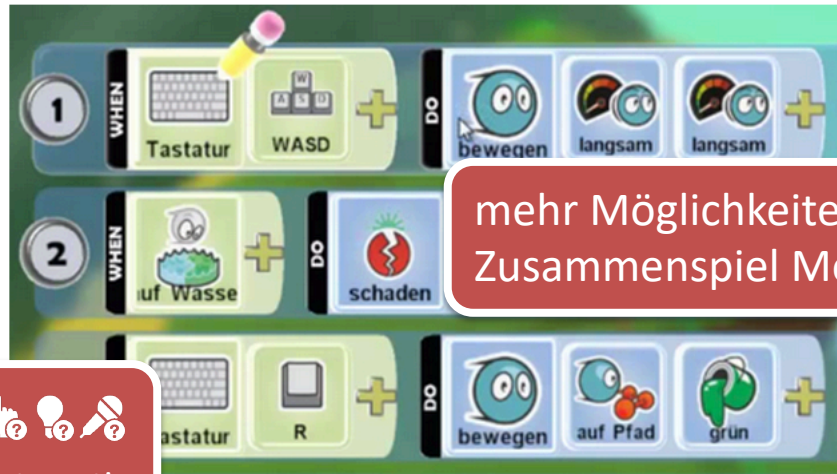
eventuell kann die „sture“ Variante für den Anfänger einfacher sein und das Konzept klarer aufzeigen.



simulated



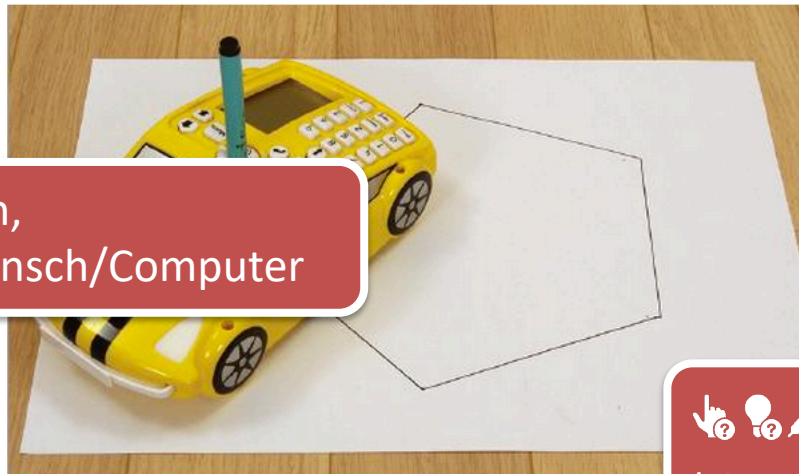
keine Sensoren im BeeBot (links), simulierte Sensoren in Kara (rechts)



mehr Möglichkeiten, Zusammenspiel Mensch/Computer



interaktiv

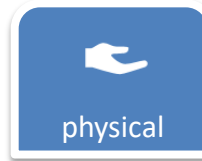


interaktiv

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



## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



## 3. Koordinationskriterium



## 4. Ausführungskriterium



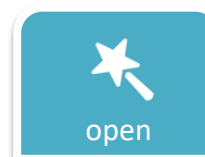
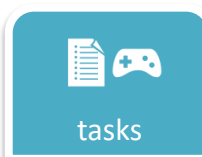
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium

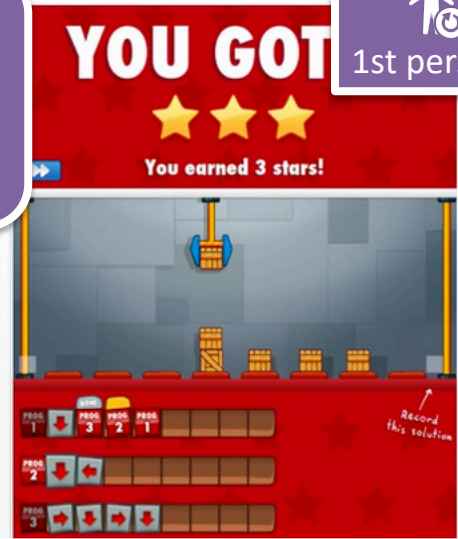
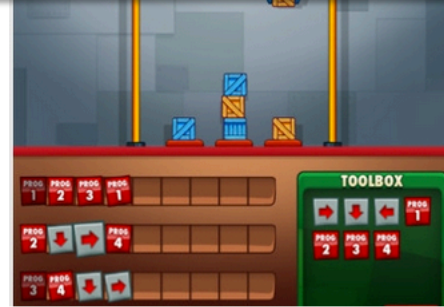
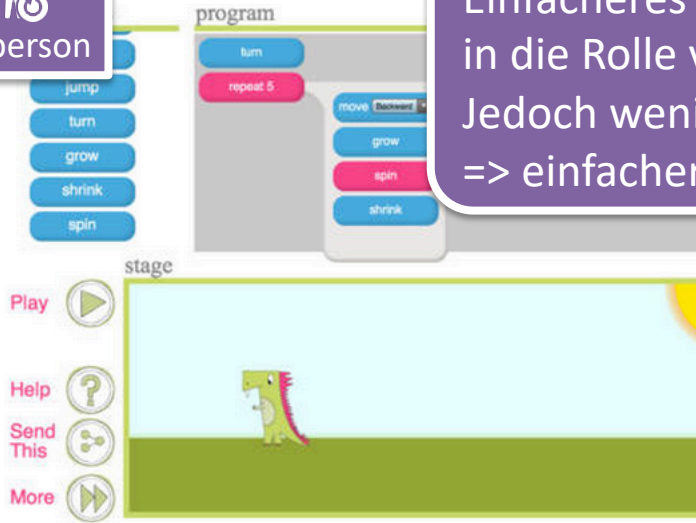


1st person

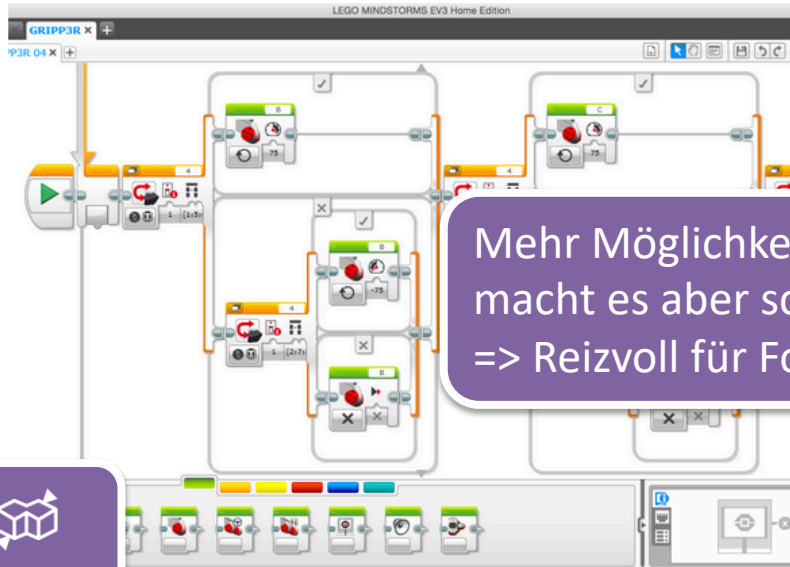


Einfacheres Debugging (sich selbst in die Rolle versetzen)  
Jedoch weniger Möglichkeiten  
=> einfacher für Anfänger

1st person



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



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



n-objects

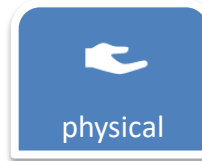


n-objects



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

## 1. Repräsentationskriterium



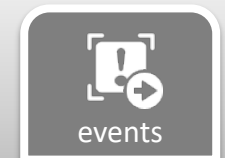
## 2. Interaktivitätskriterium



## 3. Koordinationskriterium



## 4. Ausführungskriterium



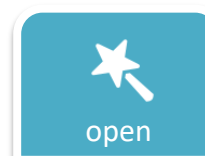
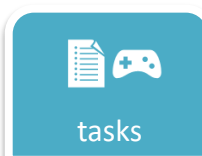
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium





sequential



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



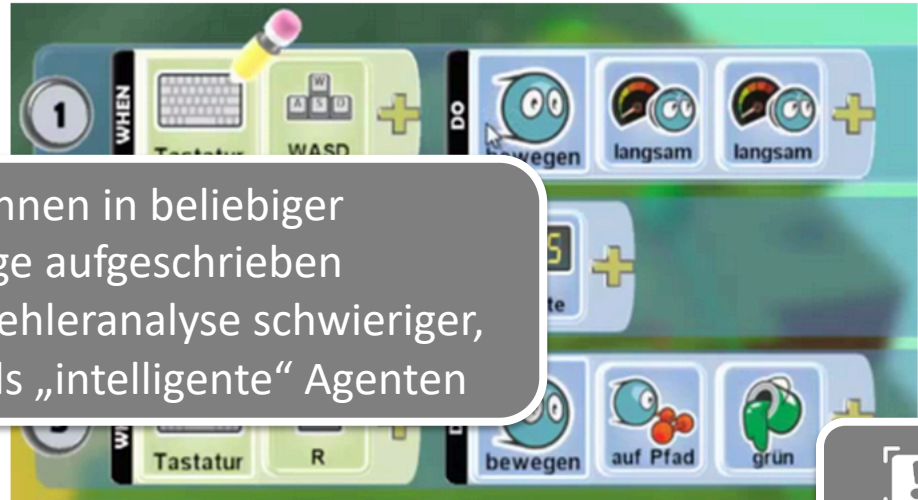
seq./evt.



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



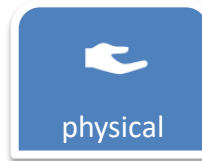
evt./seq.



events

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

## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



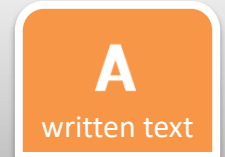
## 3. Koordinationskriterium



## 4. Ausführungskriterium



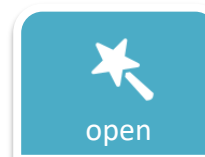
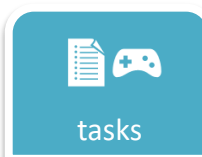
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



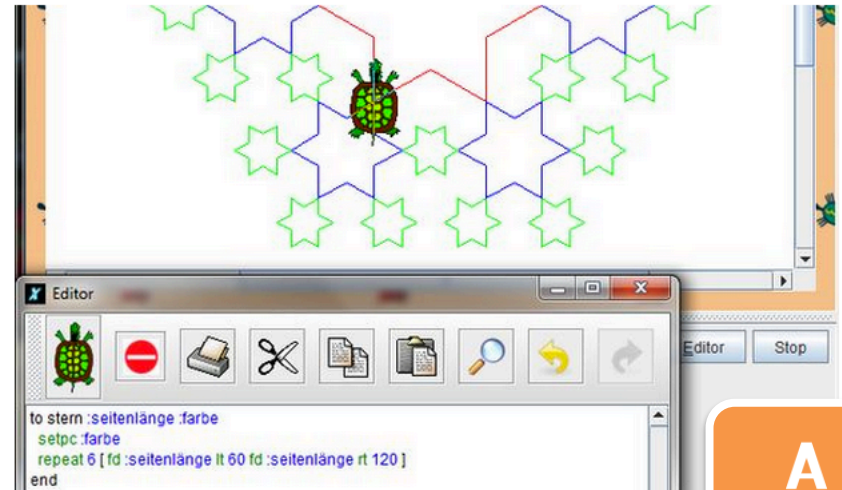
## 7. Didaktisierungskriterium





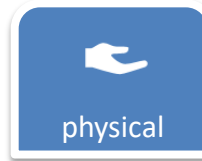


Ohne Notation: RoamerToo (links) und Icon-Notation: ScratchJR (rechts)



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

## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



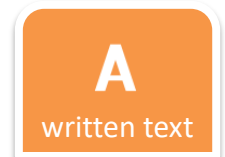
## 3. Koordinationskriterium



## 4. Ausführungskriterium



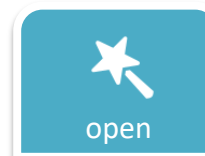
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium







Sequenz

Bedingte Anweisung

Prozeduren / Unterprogramme

Wiederholung

Rekursion

Objektorientierung

Datentypen

Variablen

## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



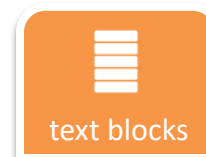
## 3. Koordinationskriterium



## 4. Ausführungskriterium



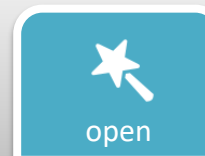
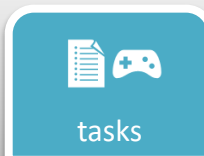
## 5. Notationskriterium



## 6. Mächtigkeitskriterium

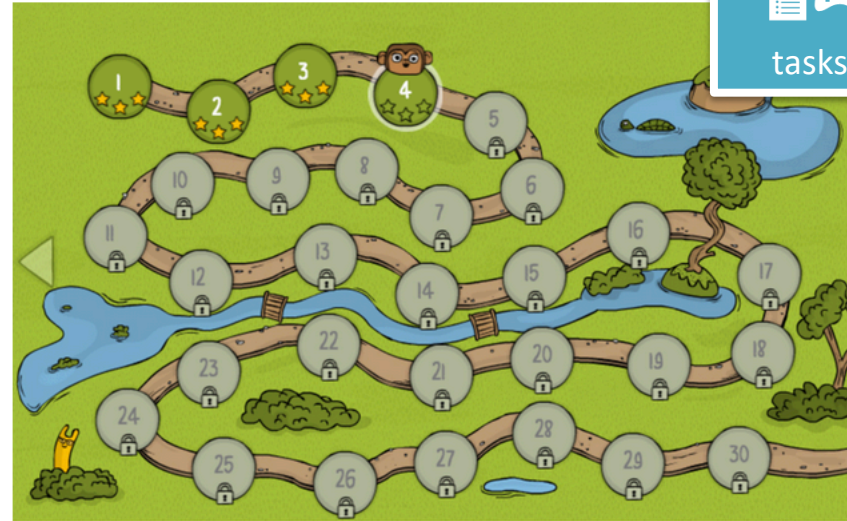


## 7. Didaktisierungskriterium

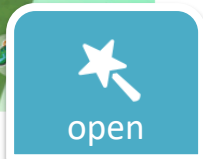
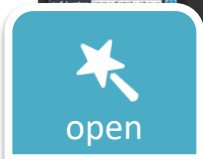
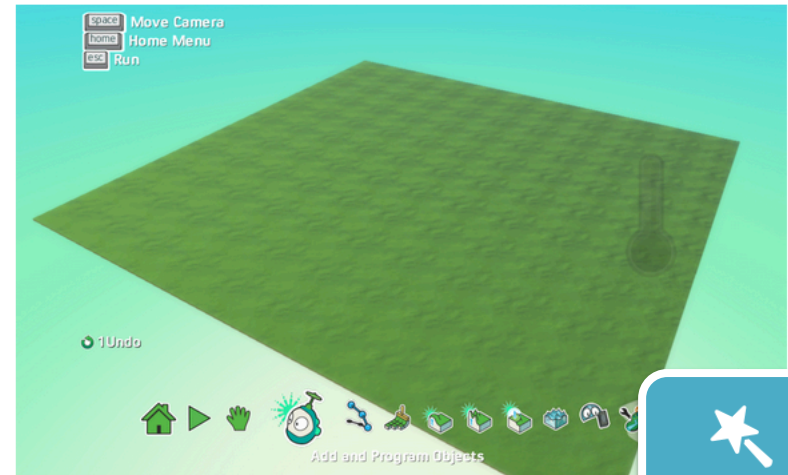
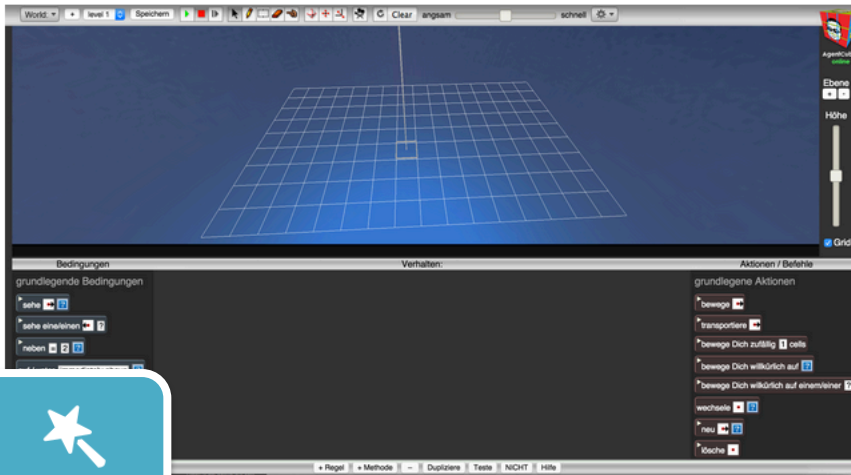




# 1 Basics



Aufgabenauswahl bei Lightbot (links) und CodeMonkey (rechts)



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

## 1. Repräsentationskriterium



## 2. Interaktivitätskriterium



## 3. Koordinationskriterium



## 4. Ausführungskriterium



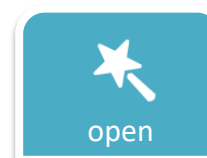
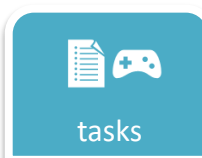
## 5. Notationskriterium



## 6. Mächtigkeitskriterium



## 7. Didaktisierungskriterium



# 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 können oder stellt sie eine leere Welt zum experimentieren bereit?

