tiny.cc/polymath_e

Computational Thinking ≠ Programming

The Rise of the Digital Polymath

Prof. Dr. Alexander Repenning

Fachhochschule Nordwestschweiz
Pädagogische Hochschule

University of Colorado
Boulder

tiny.cc/polymath_e

Computational Thinking for All

Prof. Dr. Alexander Repenning

Fachhochschule Nordwestschweiz
Pädagogische Hochschule

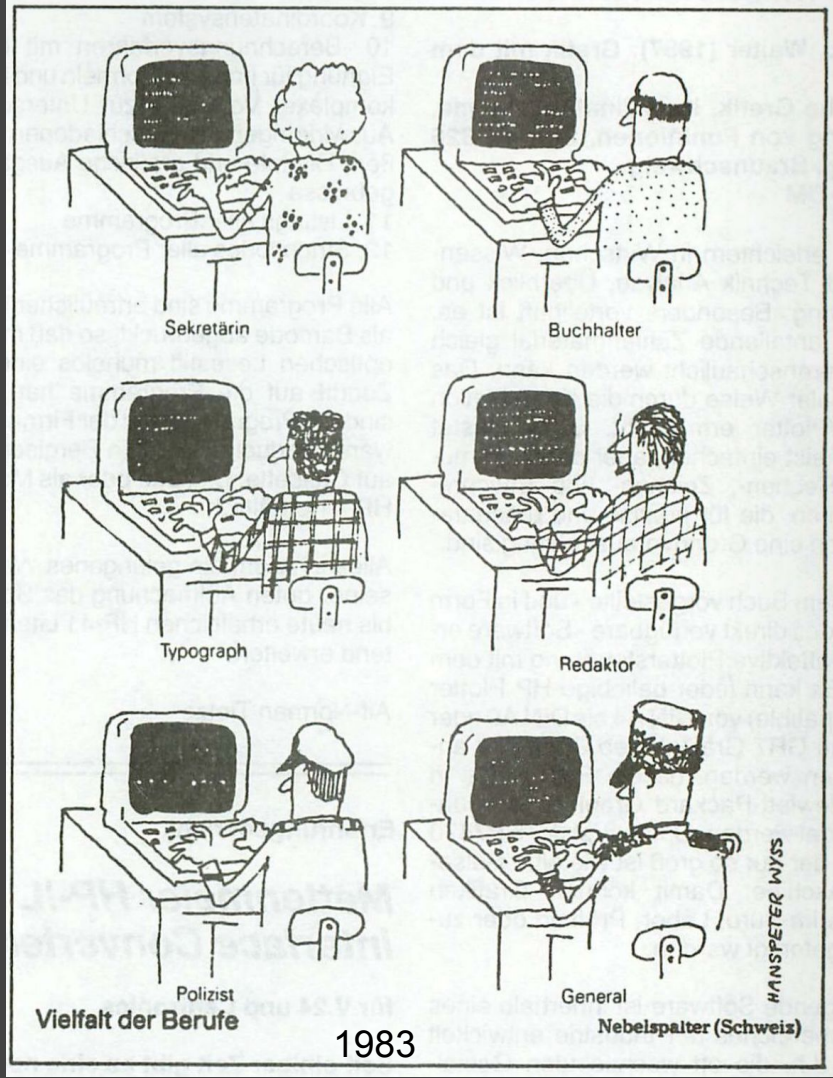
University of Colorado
Boulder

Horror **Vision**

36 years ago ...

“The variety of jobs”

– Hanspeter Wyss, Nebelspalter




Technology and Education
Time Travel

Renaissance **Polymath**

Person able to draw on complex bodies of knowledge from different subject areas to solve difficult problems

Leonardo DaVinci

- Greek: polymathēs, “having learned much”
- Latin: homo universalis
- German: Universalgelehrter

Hildegard von Bingen

1760-1820 **Industrial Revolution**

The Industrial Revolution has launched public education but has popularized ***specialization*** rather than ***discipline integration***

- This concept has remained so until the 21st century
- Students today have great difficulty with independently …
  - … connecting knowledge between disciplines
  - … deepening knowledge within discipline


The Digital Revolution eats the grandchildren of the Industrial Revolution

Digital **Polymath**

	Renaissance Polymath 	**Digital Polymath** 
Competencies	Has expert-level competencies in many disciplines	**Meta-Competence:** Has competence to acquire new competencies Peripheral perspectives of many disciplines including the attitude and ability to deepen knowledge–just in time–through the use of digital tools
Problem Solving Skills	Connects knowledge from different disciplines to solve problems	**Computational Thinker*:*** can *think with a computer* combines human abilities with computer affordances

Mandatory Pre-service Teacher Education

Innovators
tools

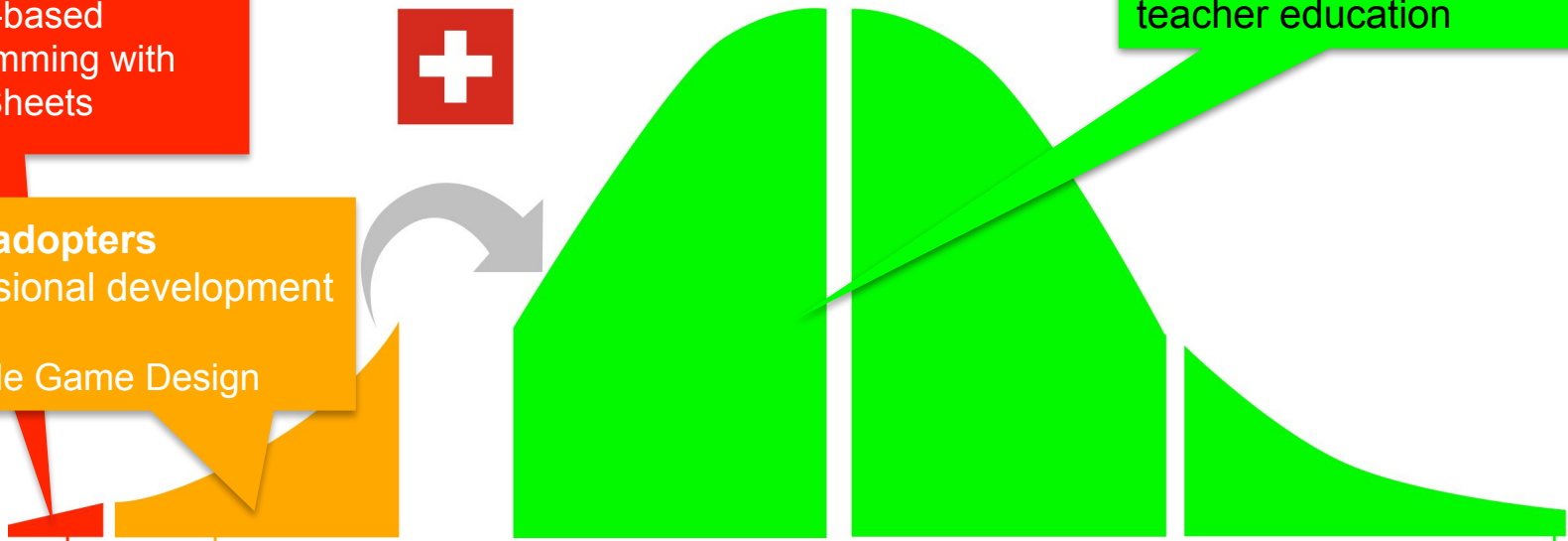
Blocks-based programming with AgentSheets

Early adopters
professional development

Scalable Game Design

All

mandatory pre-service CS teacher education



Stage I Self-Selected Students / Self-Selected Teachers

Stage II All Students / Self-Selected Teachers

Stage III All Students / All Teachers

Stage I: Self-Selected Students / Self-Selected Teachers

1996

Stage II: All Students / Self-Selected Teachers

2011

Stage III
All students
All teachers

Mandatory pre-service teacher education: every future elementary school teacher must pass a computer science course

Computer Science Education

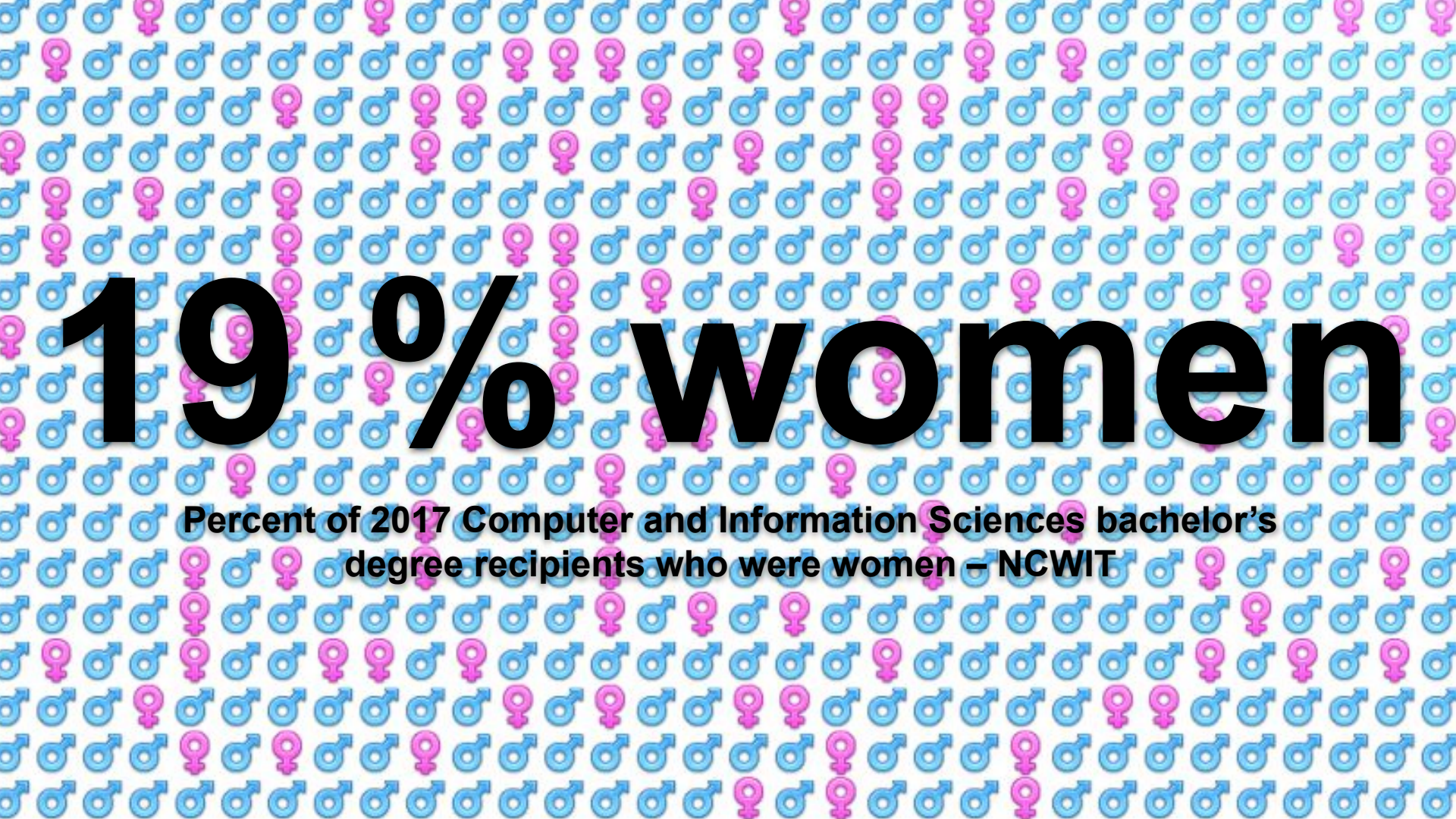
PH FHNW

1000+ teachers educated

Fachhochschule Nordwestschweiz
Pädagogische Hochschule

But who are **ALL** these people?

a demographic shift


19% women
Percent of 2017 Computer and Information Sciences bachelor's degree recipients who were women – NCWIT

75% women

Pre-service elementary school teachers=school of ed students

0.2% can program

2.7% of Swiss workforce are software developers

Strongly agree: 1
(n = 539)

Misconceptions **CS is about...**

- Application skills
  - How to use computers
  - How to use apps: MS Office, browsers
- Media skills
  - Learn about internet dangers

Course Concepts

1. **Motivation:** Game Design

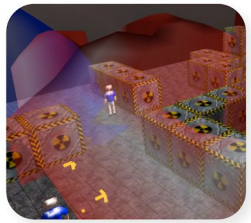
2. **Tools:** Computational Thinking Tools

3. **Structure:** The 7 BIG Ideas of computer science

Course Concepts

AgentSheet & AgentCubes projects

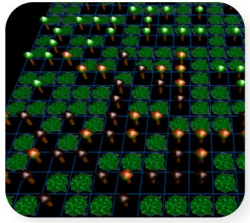
City Traffic



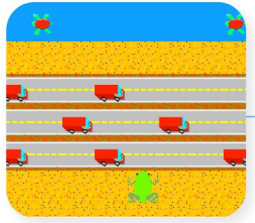
the Sims

Bridge Builder

Pac Man



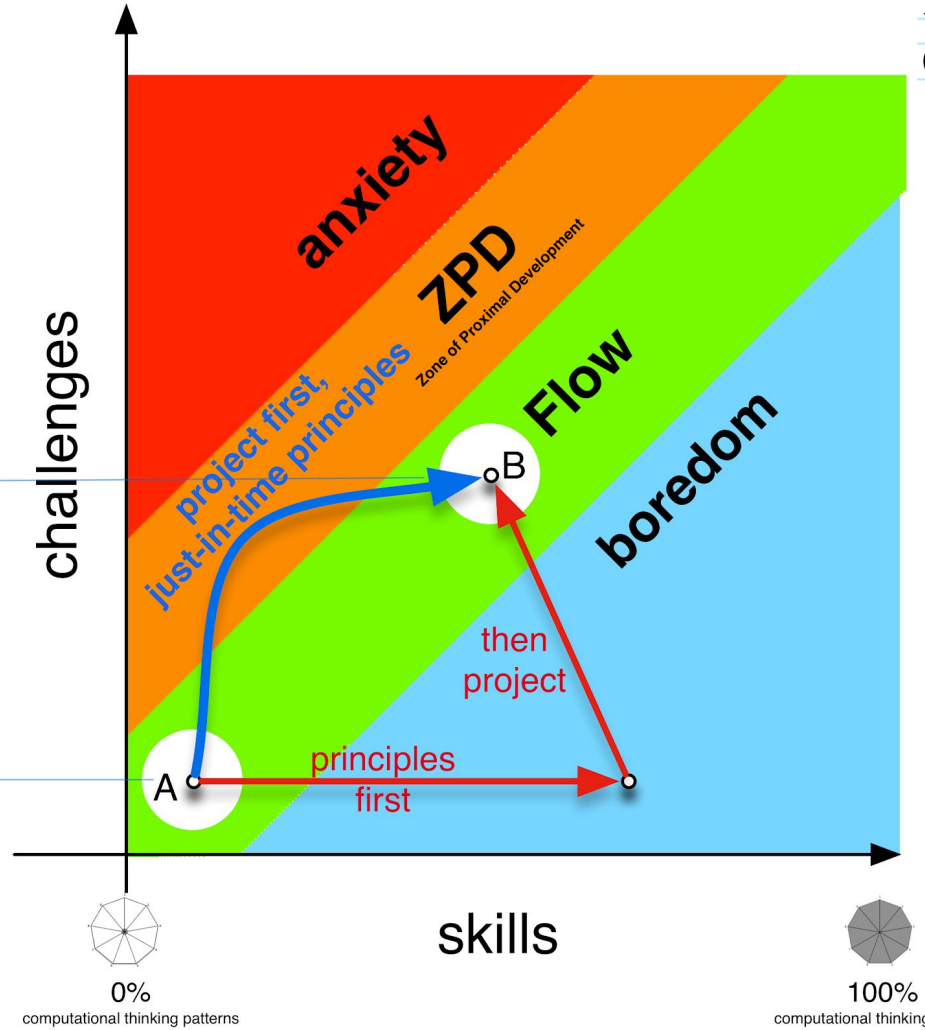
Forest Fire



Frogger

simulations

games



SCALABLE GAME DESIGN

1

KANTON solothurn

SCALABLE GAME DESIGN

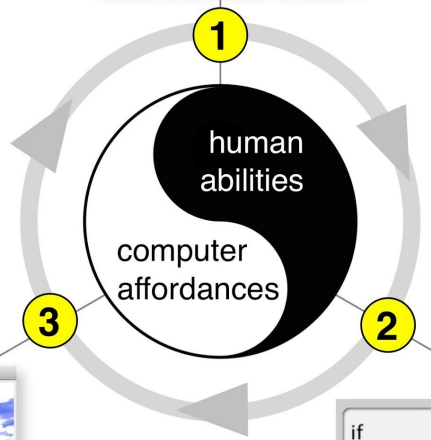
1

Computational Thinking

Abstraction

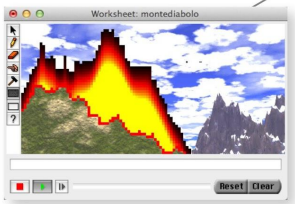
Problem Formulation

"how does a mudslide work?"



Analysis

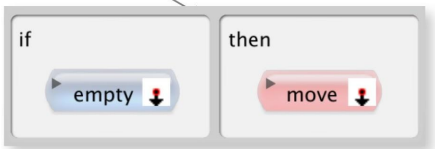
Solution Execution and Evaluation



visualize the consequence of thinking

Automation

Solution Expression



build simple model of gravity

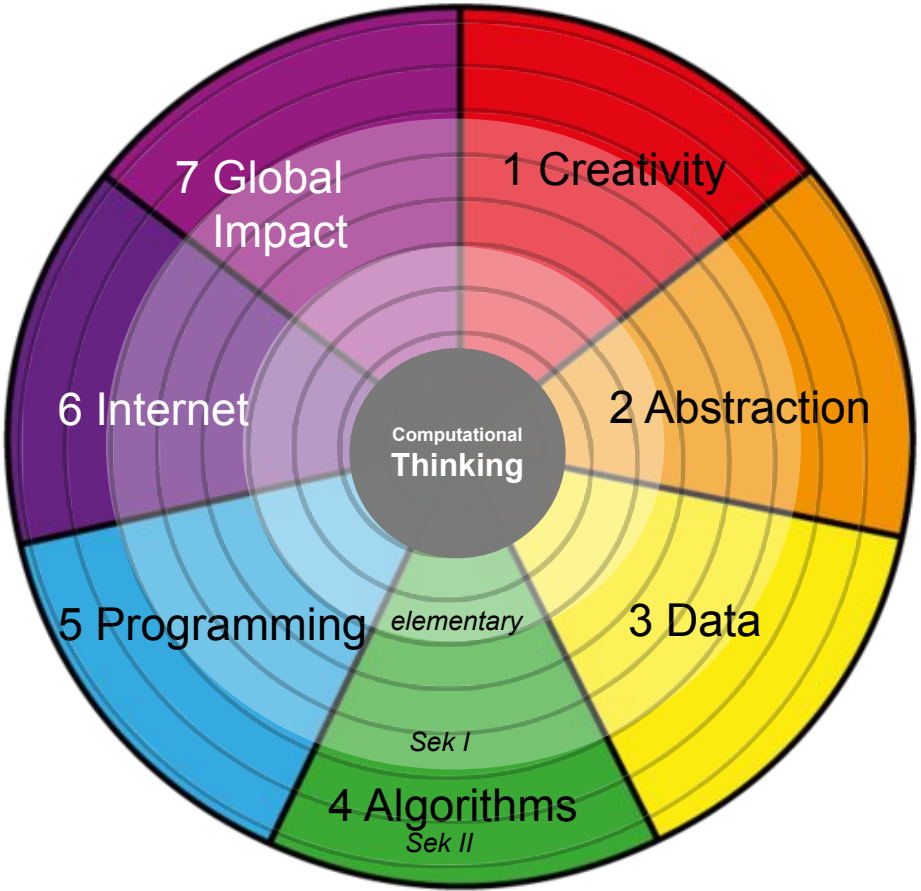
Don't

- Think like a computer
- Think about computers

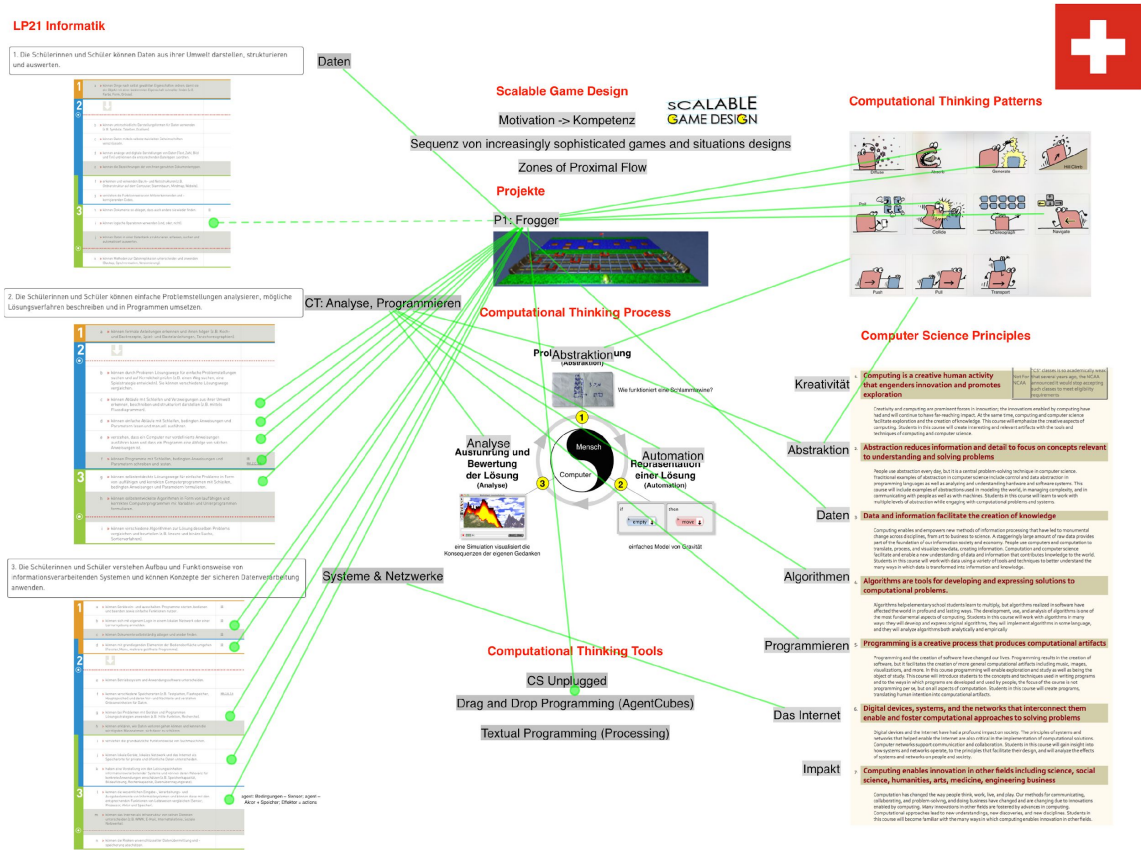
but

- **Think WITH the computer**

“7 BIG Ideas”


1 Creativity
2 Abstraction
3 Data
4 Algorithms
Sek II
Sek I
5 Programming
6 Internet
7 Global Impact
Computational Thinking
elementary

Game Design & LP21



44%

LP21 Informatik Kompetenzstufen

44%

A single game design activity covers a large percentage of the national curriculum (Lehrplan 21) Computer Science requirements

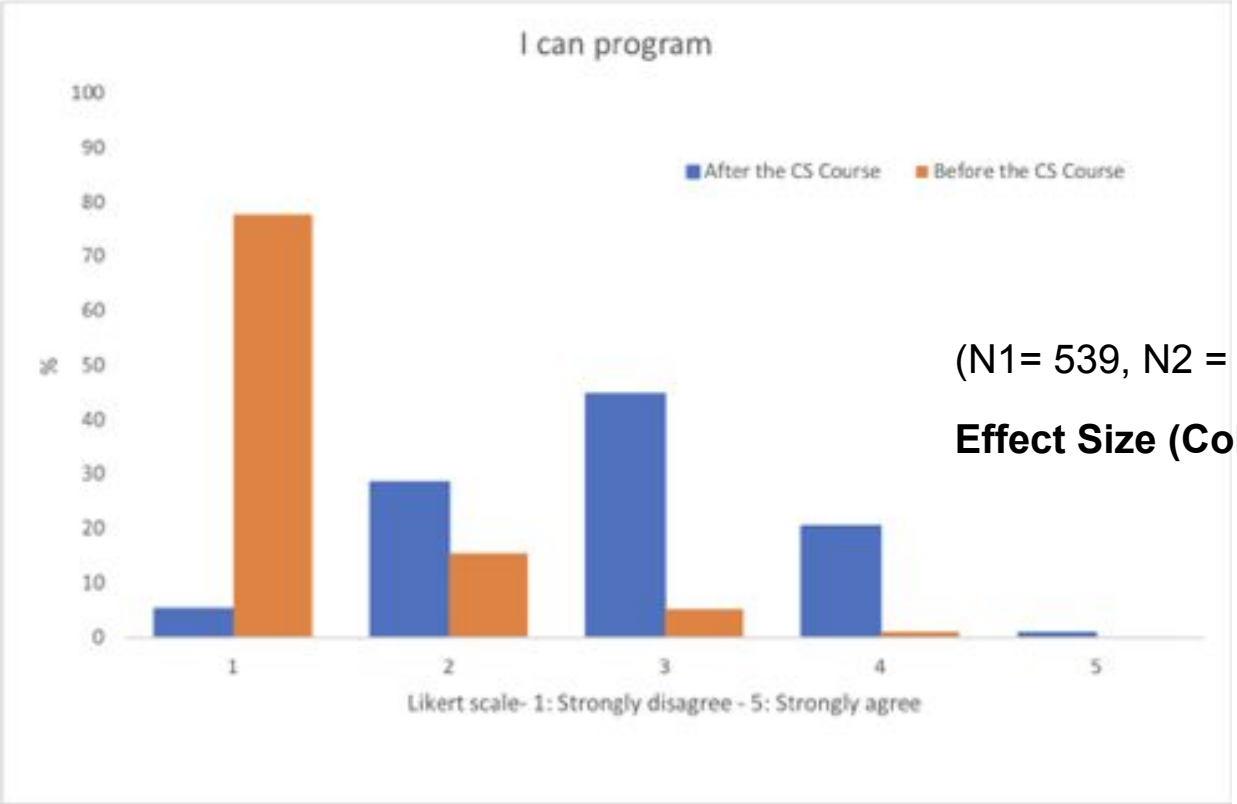
Results

1000+ students:

- Learned to write simple programs
- Learned to build STEM simulations and games
- Became Computational Thinkers

Resultate

More results



(N1= 539, N2 = 471)

Effect Size (Cohen‘s d): 2.05

Computational Thinking

≠

Programming

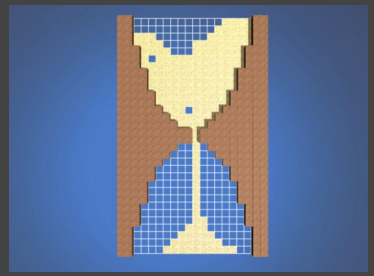
Computational Thinking Tools

- <switch to keynote slides>

STEM + C

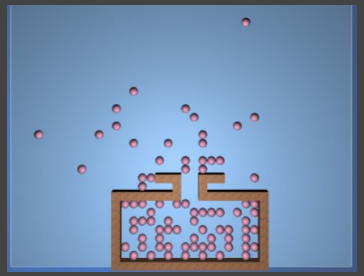
Hourglass

Build an hourglass. Sand should fall from an upper area of the vessel into the lower area and pile up there.



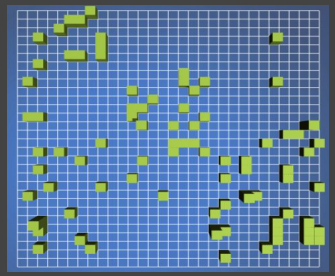
Perfume

Build a perfume bottle. The bottle should contain perfume particles. At the start of the simulation, the particles should escape and spread randomly into the simulation world.



Bacteria

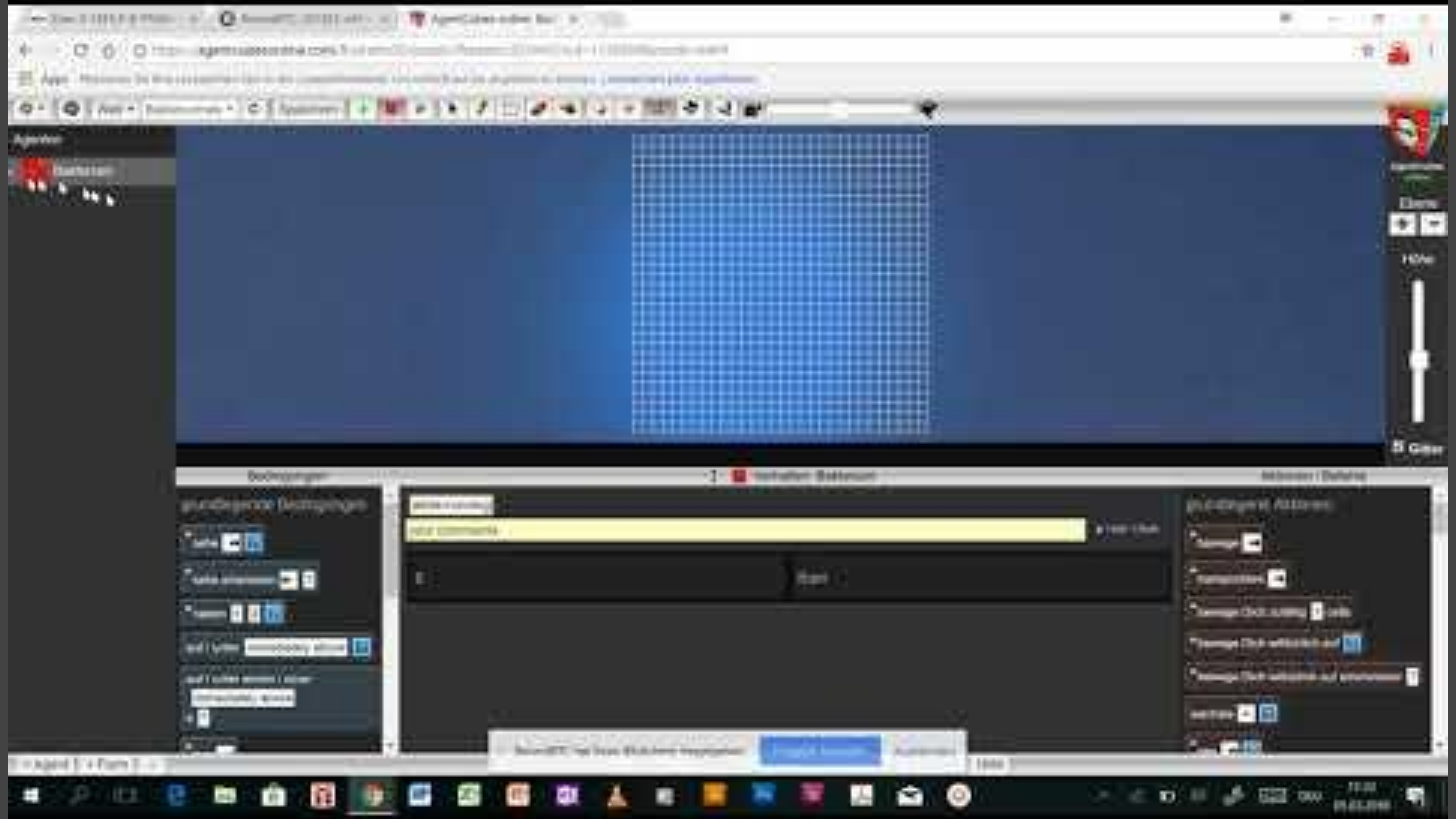
Build a simulation of proliferating bacteria. The simulation should begin with a randomly moving bacterium that divides from time to time.



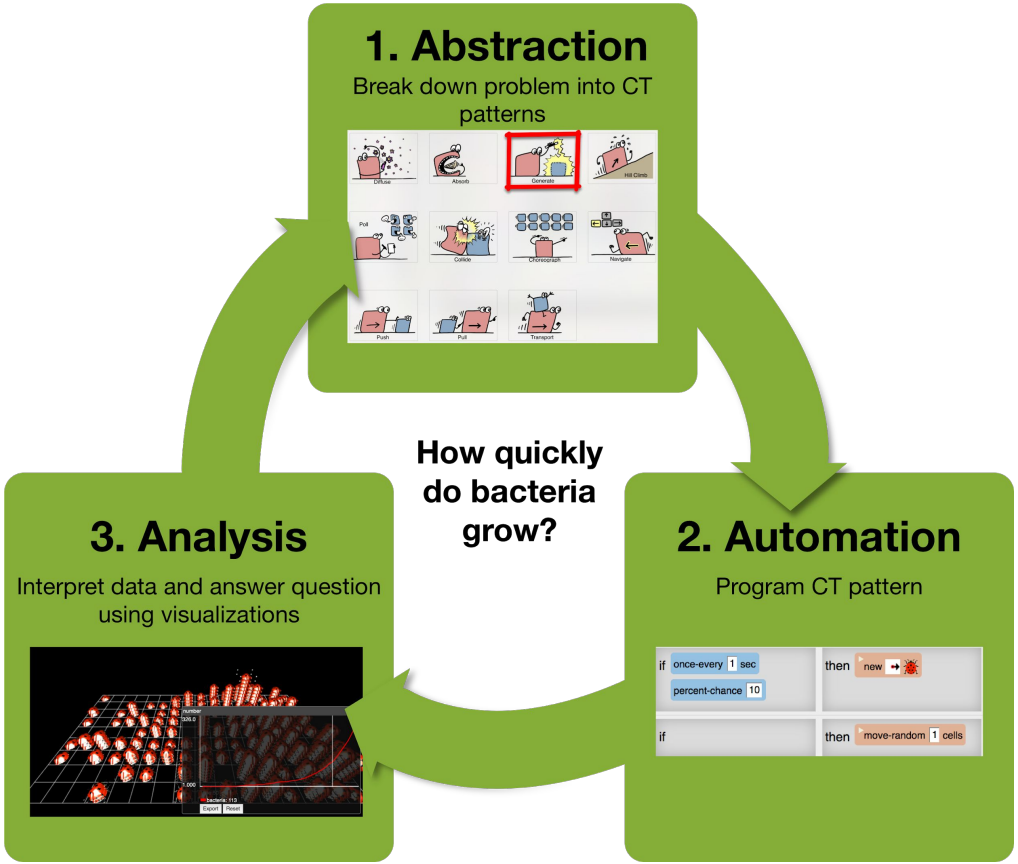
Using a Computational Thinking Tool

AgentCubes
online

Demo
https://agentcubesonline.com



Computational Thinking → Polymath



Conclusions

Is Switzerland in the Digital fast lane?

While Switzerland is still in the rear mirror of Digital Thought Leader nations such as the US and the UK in terms of digital education, it is now taking extraordinary measures to accelerate and has set the metaphorical turn signal.

tiny.cc/polymath_e

Thank you!

Swiss Science Council SSC

- Digital Competences
- Die Schweiz auf der digitalen Überholspur

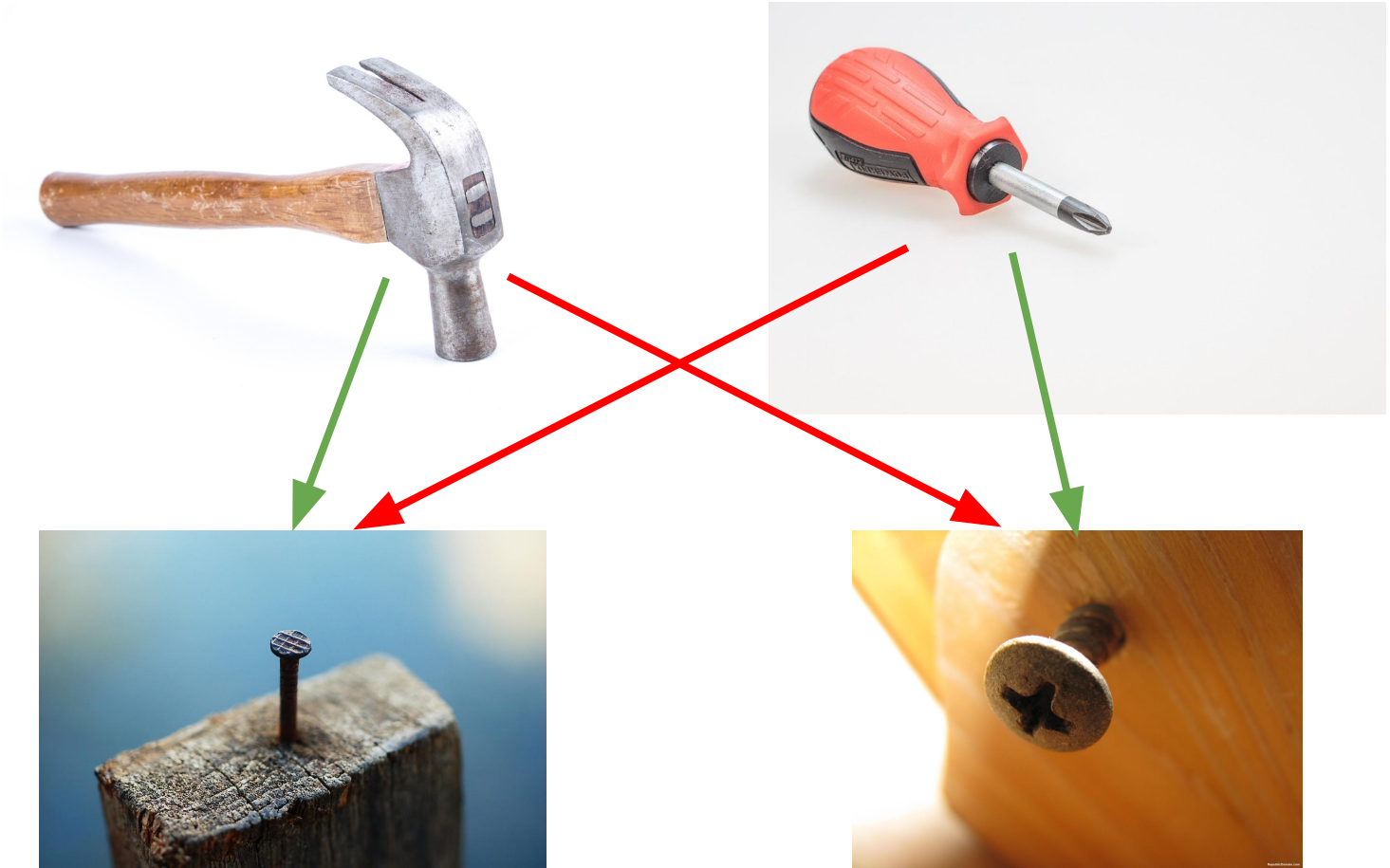
Hasler Stiftung

- Computational Thinking

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

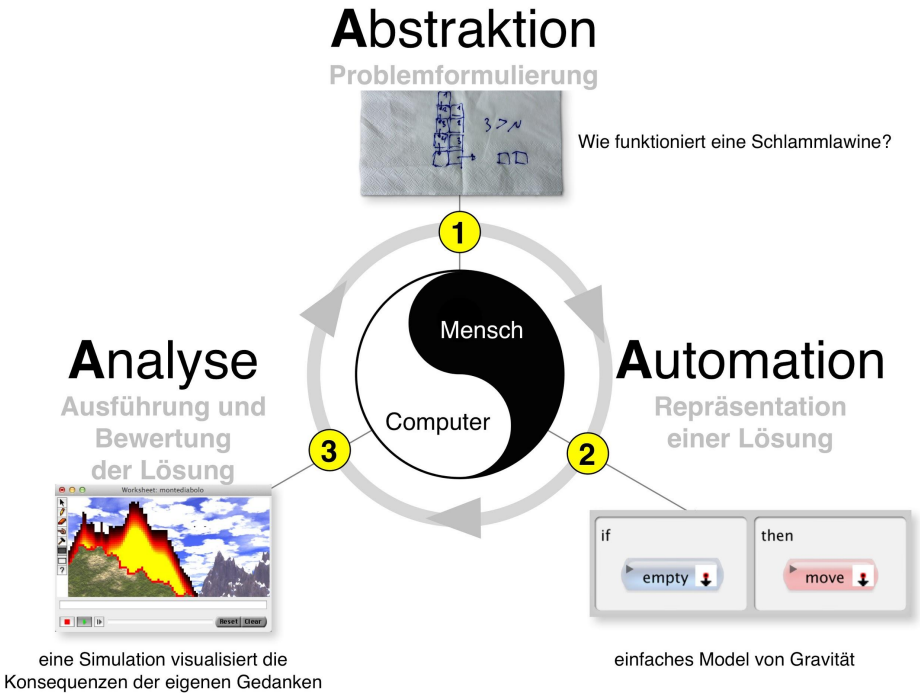
HASLERSTIFTUNG

Which tool is better?




DO IT+GARDEN
MIGROS

Computational Thinking



NICHT

- Denken WIE ein Computer
- Denken über Computer

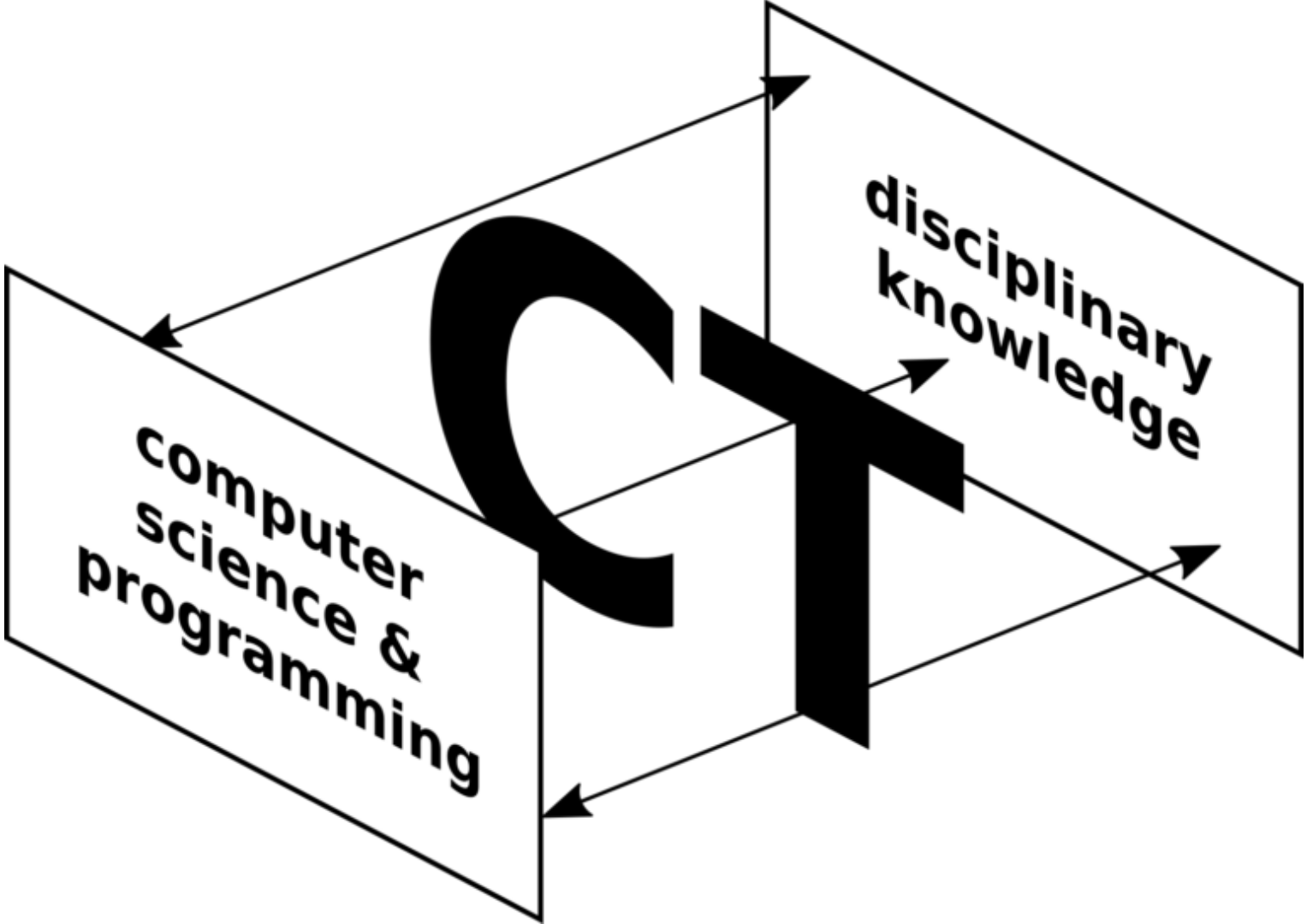
Sondern

- Denken MIT dem Computer

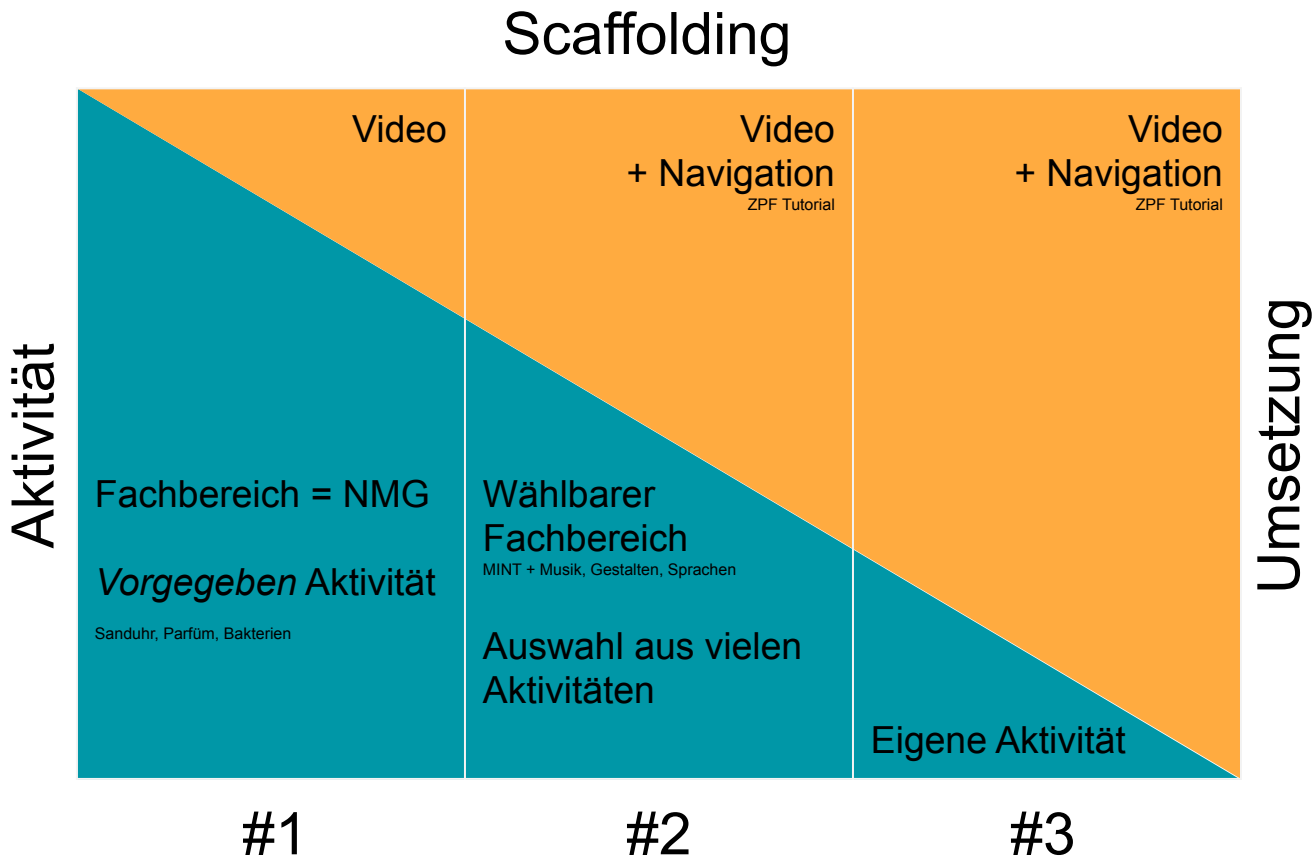
2

Fach

Didaktik


CT
disciplinary knowledge
computer science & programming

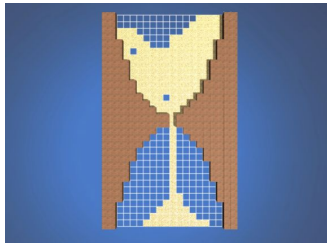
Projectlets



Informatik + NMG

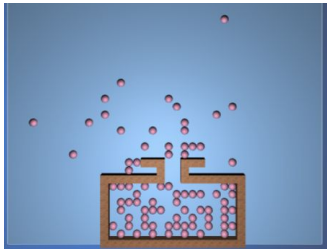
1: Sanduhr

Baue eine Sanduhr. Sand soll von einem oberen Bereich des Gefässes in den unteren Bereich fallen und sich dort häufen.



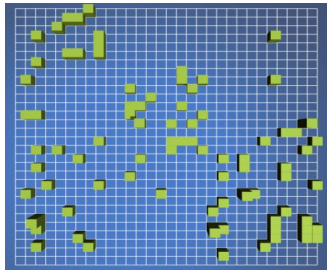
2: Parfüm

Baue eine Parfümflasche. Die Flasche soll Parfümpartikel enthalten. Beim Start der Simulation sollen die Partikel entweichen und sich zufällig in der Spielwelt verteilen.



3: Bakterien

Baue eine Simulation von sich vermehrenden Bakterien. Die Simulation soll anfangen mit einem zufällig umherwandernden Bakterium, das sich periodisch teilt.



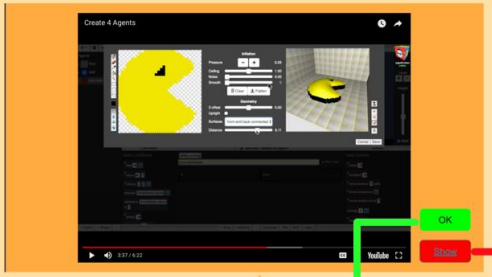
Projectlet #1

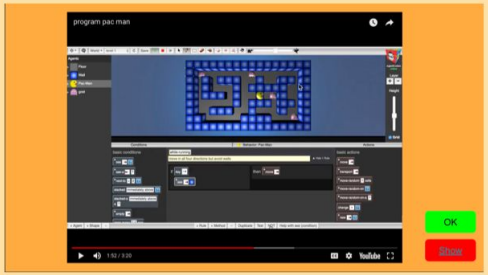
- Video Tutorials
- Beispiel Sanduhr

Zones of Proximal Flow Tutorials

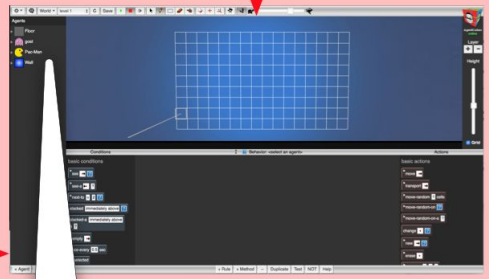
WHAT Slides

HOW Slides



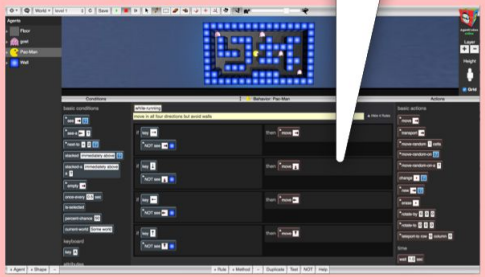


prebaked Project



agents created

Pac-Man programmed

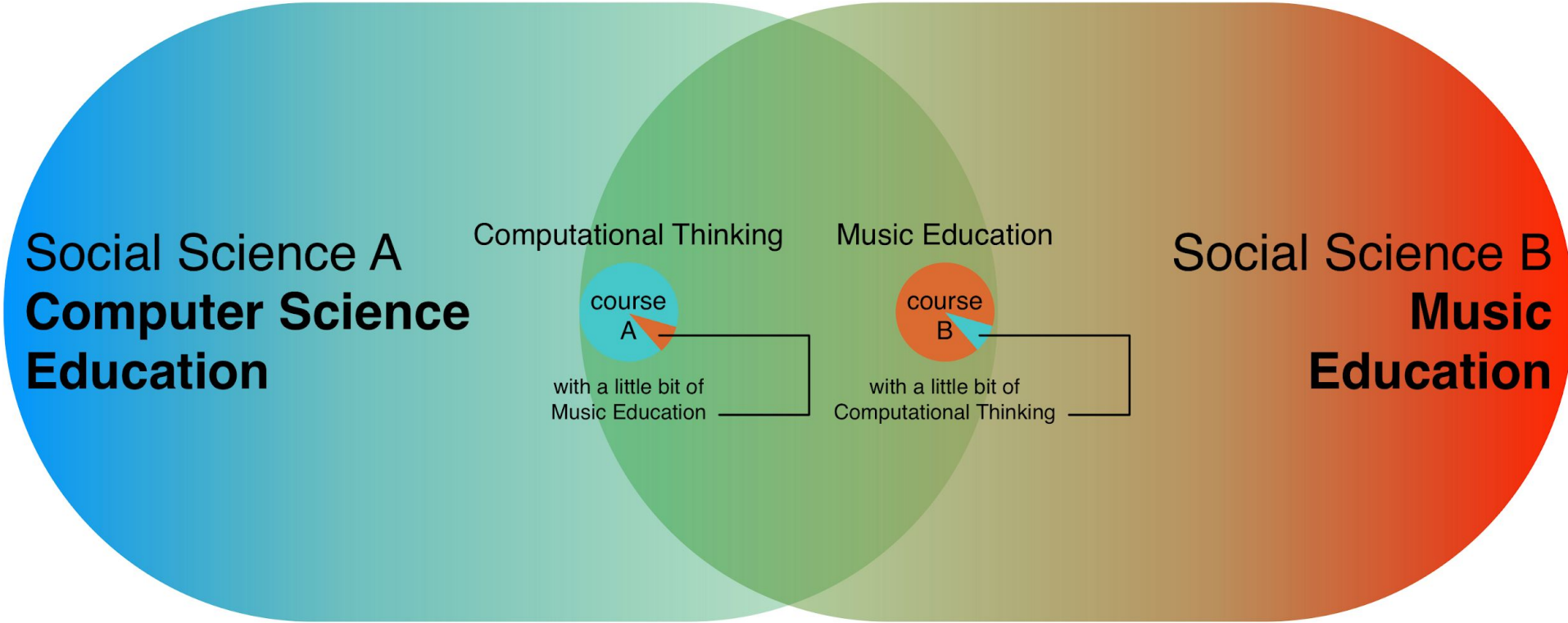


Projectlet #2

- Beispiel Musikinstrument mit MakeyMakey

Unterrichtseinheiten

- Alle


Social Science A
Computer Science
Education
Computational Thinking
course
A
with a little bit of
Music Education
Music Education
course
B
with a little bit of
Computational Thinking
Social Science B
Music
Education

Sinergia: Math in Game Design
Understanding Computational Thinking

- Tools: [PH FHNW] Computational Thinking Tools
- Pedagogy: [ETHZ, Manu Kapur, Math Ed] Productive Failure
- Practice: Design-Based Implementation Research