

Computational Thinking  $\neq$  Programming

# The Rise of the Digital Polymath

Prof. Dr. Alexander Repenning

# Computational Thinking for All

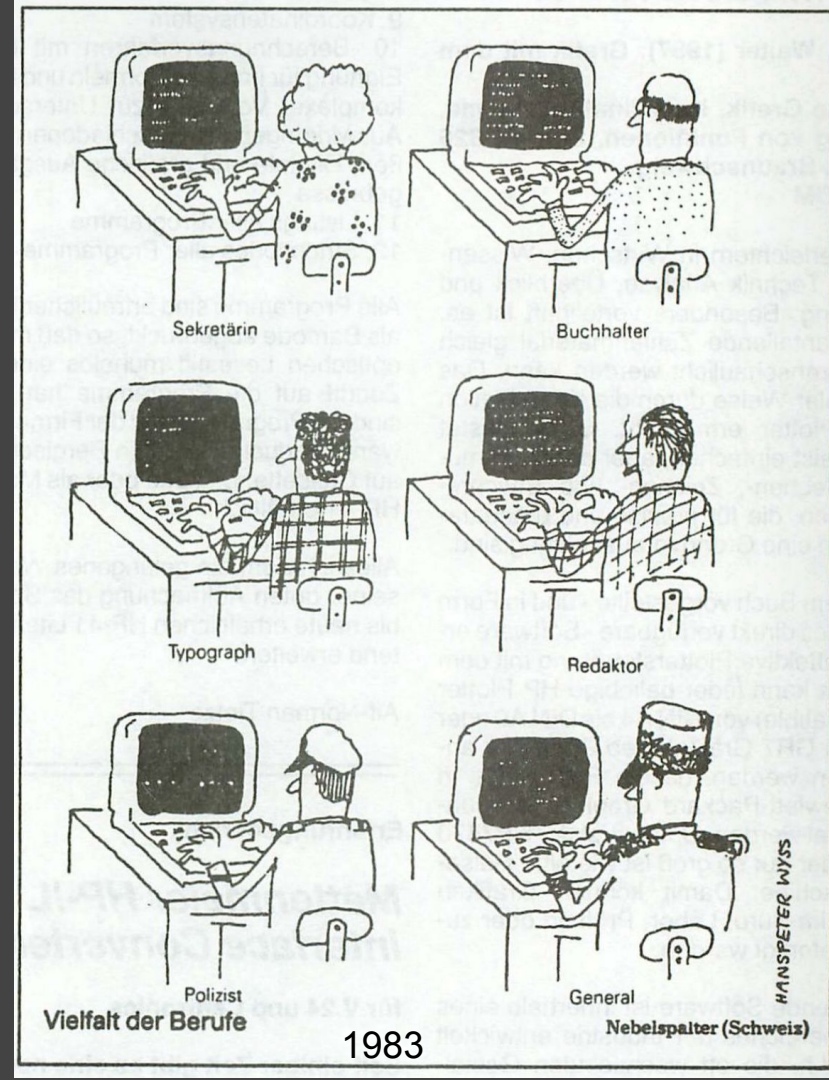
Prof. Dr. Alexander Repenning

# Horror Vision

36 years ago ...

“The variety of jobs”

– Hanspeter Wyss, Nebelspalter



1983



# Technology and Education

Time Travel

# Renaissance Polymath

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

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



Leonardo DaVinci



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 21<sup>st</sup> 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



**Innovators**  
tools

Blocks-based  
programming with  
AgentSheets

**Early adopters**  
professional development

Scalable Game Design

**Mandatory  
Pre-service  
Teacher Education**



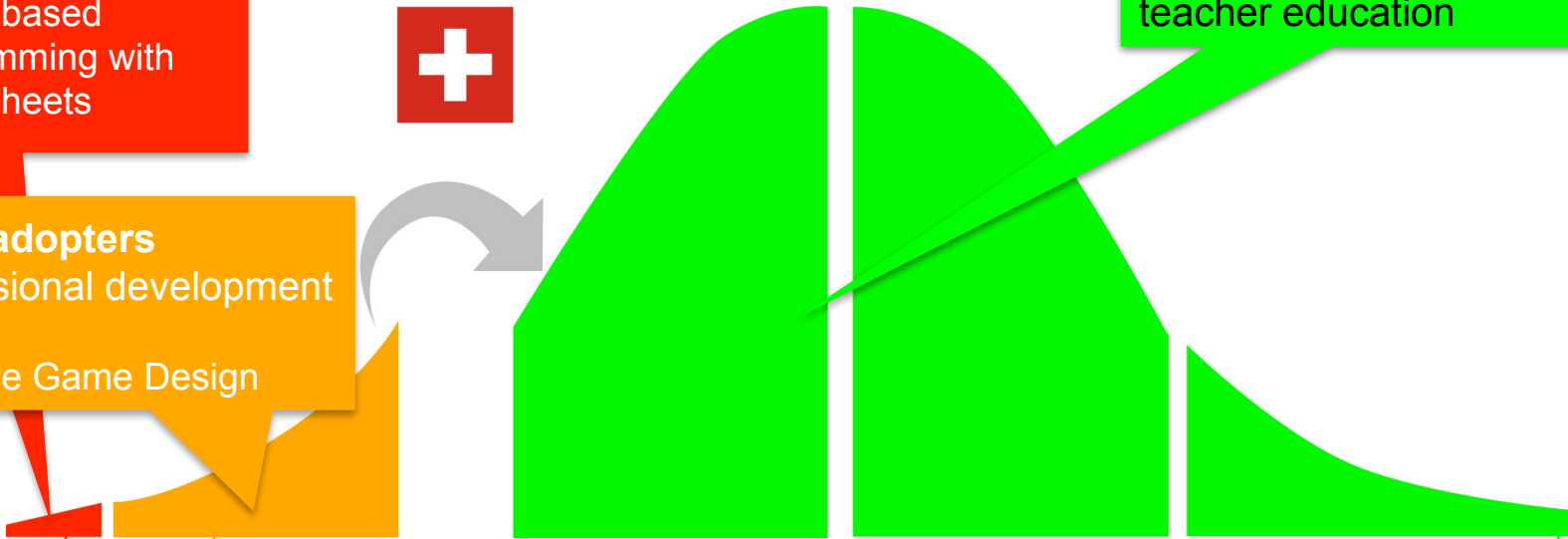
**All**

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



# Stage II: All Students / Self-Selected Teachers

**9NEWS.com**  
COLORADO'S NEWS LEADER



TEACHERS & VIDEO GAMES 9NEWS.com

5:18 81°

SCALABLE  
GAME DESIGN  
New York

SCALABLE  
GAME DESIGN  
México

**Stage III**

**All students**

**All teachers**

SCALABLE  
GAME DESIGN

Switzerland

**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...

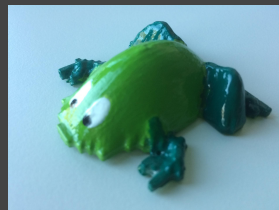
- Application skills
  - How to use computers
  - How to use apps: MS Office, browsers
- Media skills
  - Learn about internet dangers

# Course Concepts

1. **Motivation:** Game Design



2. **Tools:** Computational Thinking Tools



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

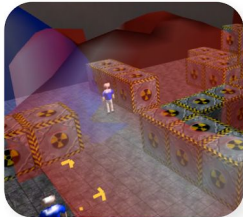


# Course Concepts

# AgentSheet & AgentCubes projects



City Traffic



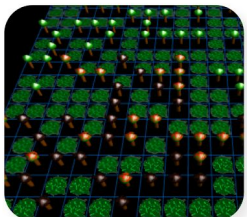
the Sims



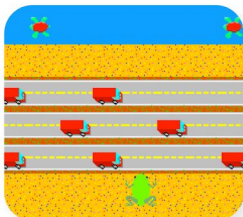
Bridge Builder



Pac Man



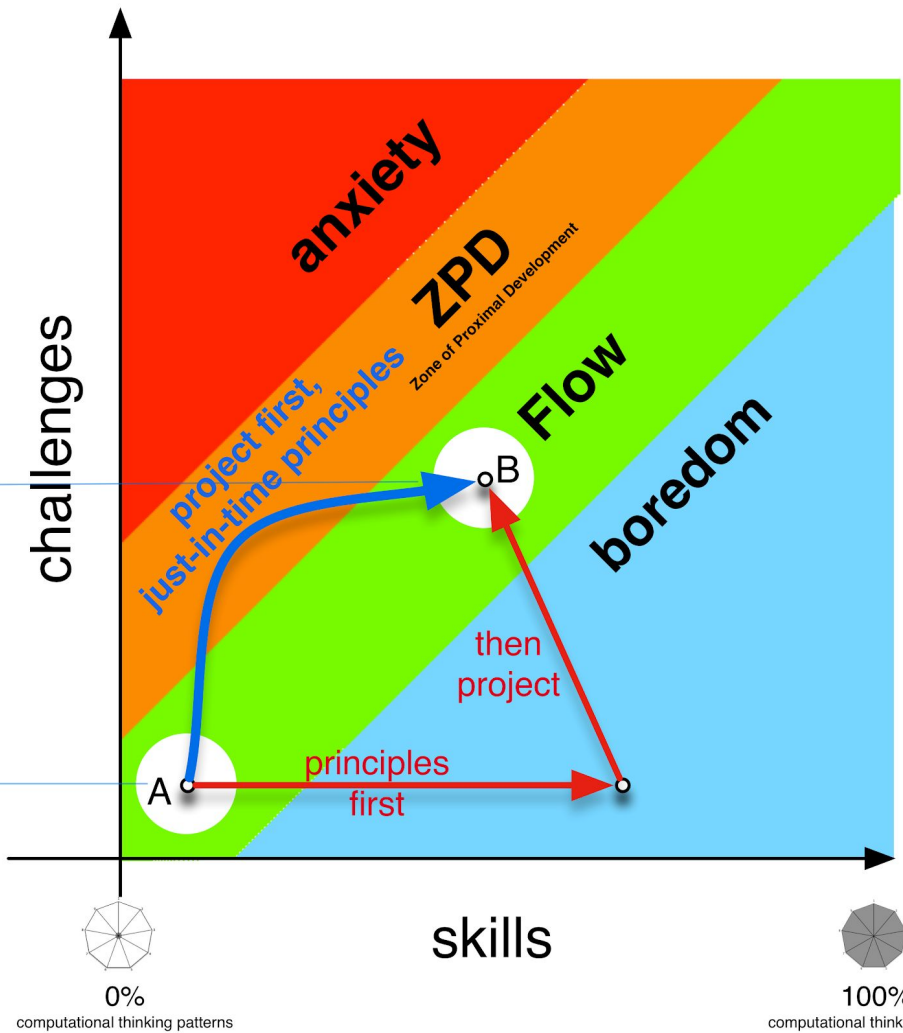
Forest Fire



Frogger

simulations

games

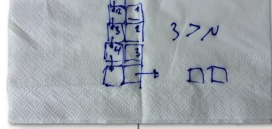




# Computational Thinking

## Abstraction

Problem Formulation



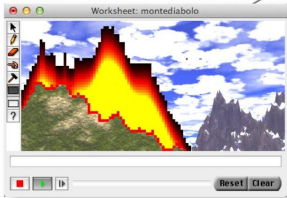
"how does a mudslide work?"

Don't

- Think like a computer
- Think about computers

## Analysis

Solution Execution and Evaluation



visualize the consequence of thinking

## Automation

Solution Expression



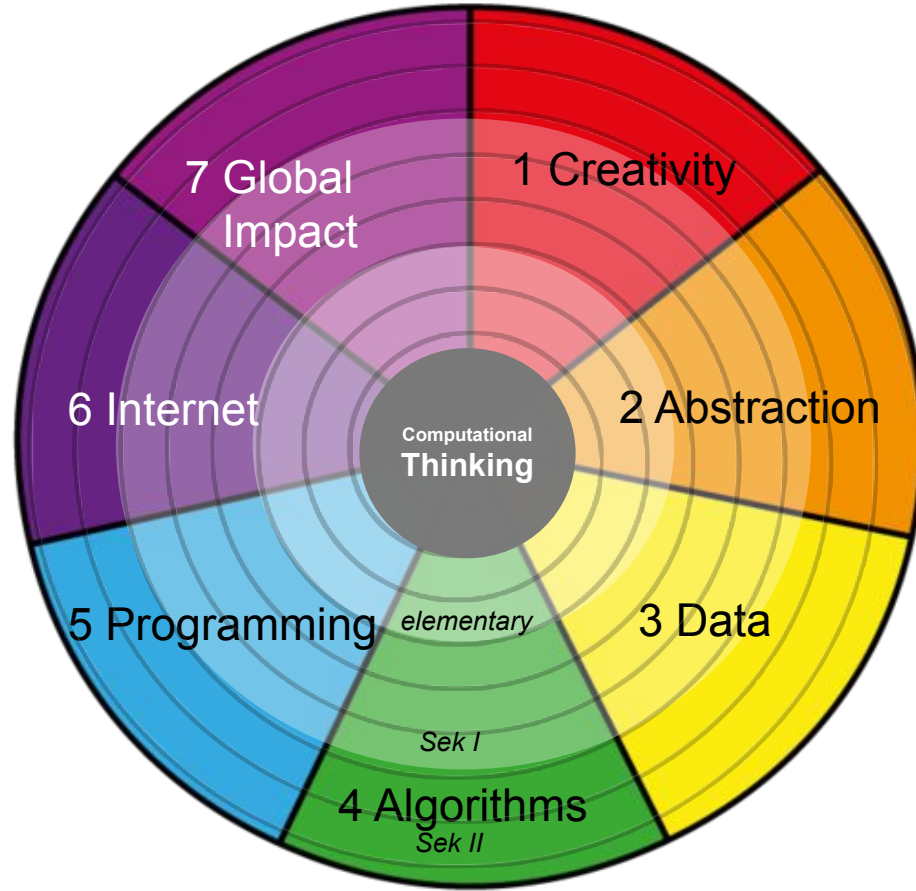
build simple model of gravity

but

- Think **WITH** the computer

2

# “7 BIG Ideas”





# Game Design & LP21



## LP21 Informatik

1. Die Schülerinnen und Schüler können Daten aus ihrer Umwelt darstellen, strukturieren und auswerten.

1	2	3
<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> <li>4. Daten interpretieren und auswerten (z.B. Schlussfolgerungen ziehen)</li> </ul>

2. Die Schülerinnen und Schüler können einfache Problemstellungen analysieren, mögliche Lösungsverfahren beschreiben und in Programmen einsetzen.

1	2	3
<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> <li>4. Daten interpretieren und auswerten (z.B. Schlussfolgerungen ziehen)</li> </ul>

3. Die Schülerinnen und Schüler verstehen Aufbau und Funktionsweise von informationstechnischen Systemen und können Konzepte per sicheren Datenaustausch anwenden.

1	2	3
<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> </ul>	<ul style="list-style-type: none"> <li>1. Daten aus der Umwelt darstellen und auswerten (z.B. Diagramm, Tabelle, Karte)</li> <li>2. Daten strukturieren und auswerten (z.B. Sortieren, Filtern, Gruppieren)</li> <li>3. Daten analysieren und auswerten (z.B. Korrelation, Regression)</li> <li>4. Daten interpretieren und auswerten (z.B. Schlussfolgerungen ziehen)</li> </ul>

Daten

### Scalable Game Design

Motivation -> Kompetenz

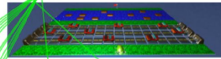


Sequenz von increasingly sophisticated games and situations designs

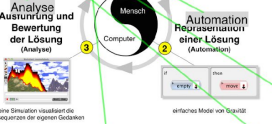
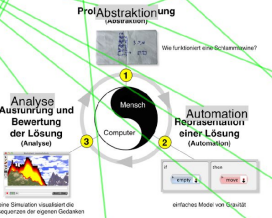
Zones of Proximal Flow

### Projekte

P1: Frogger



### Computational Thinking Process

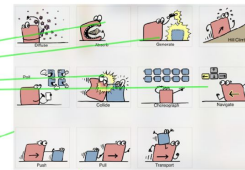


Systeme & Netzwerke

### Computational Thinking Tools

- CS Unplugged
- Drag and Drop Programming (AgentCubes)
- Textual Programming (Processing)

### Computational Thinking Patterns



### Computer Science Principles

- Kreativität**
  - Computing is a creative human activity that engages invention and promotes exploration.
- Abstraktion**
  - Abstraction reduces information and detail to focus on concepts relevant to understanding and solving problems.
- Daten**
  - Data and information facilitate the creation of knowledge.
- Algorithmen**
  - Algorithms are tools for developing and expressing solutions to computational problems.
- Programmieren**
  - Programming is a creative process that produces computational artifacts.
- Das Internet**
  - Digital devices, systems, and the networks that interconnect them enable and foster computational approaches to solving problems.
- Impakt**
  - Computing enables innovation in other fields including science, social science, humanities, arts, medicine, engineering, business.

# 44%

## LP21 Informatik Kompetenzstufen

# 44%

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

# Results

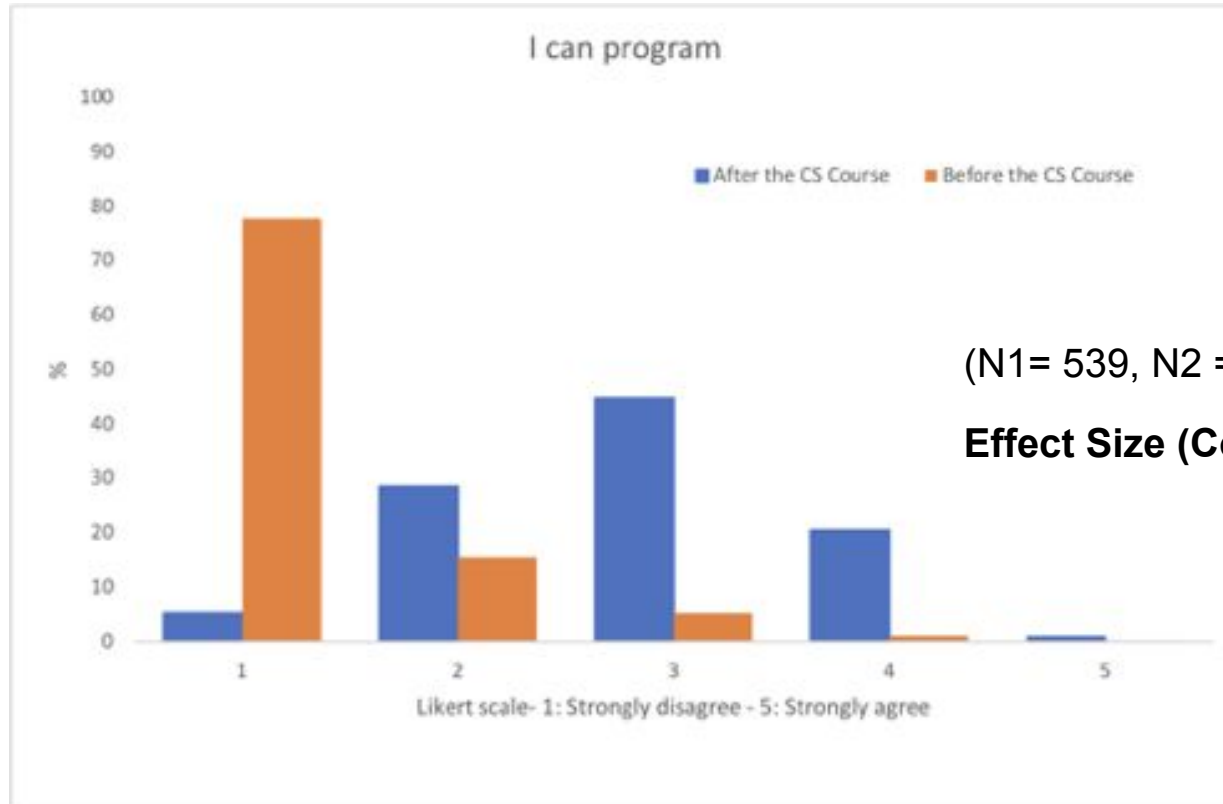
1000+ students:

- 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

A vertical decorative bar on the left side of the slide, consisting of a grid of small, colorful icons. Each icon is a stylized face with different colors and features, arranged in a repeating pattern.

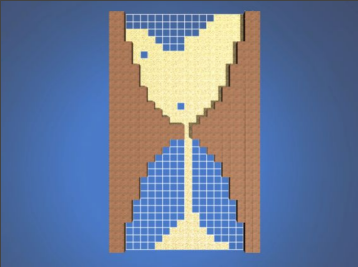
# Computational Thinking Tools

- <switch to keynote slides>

# STEM + C

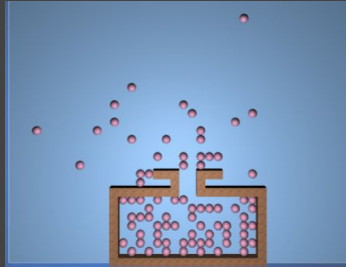
## Hourglass

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



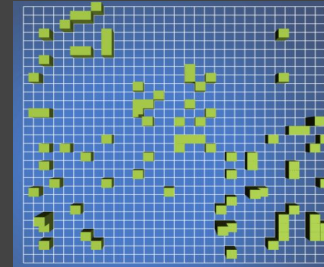
## Perfume

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

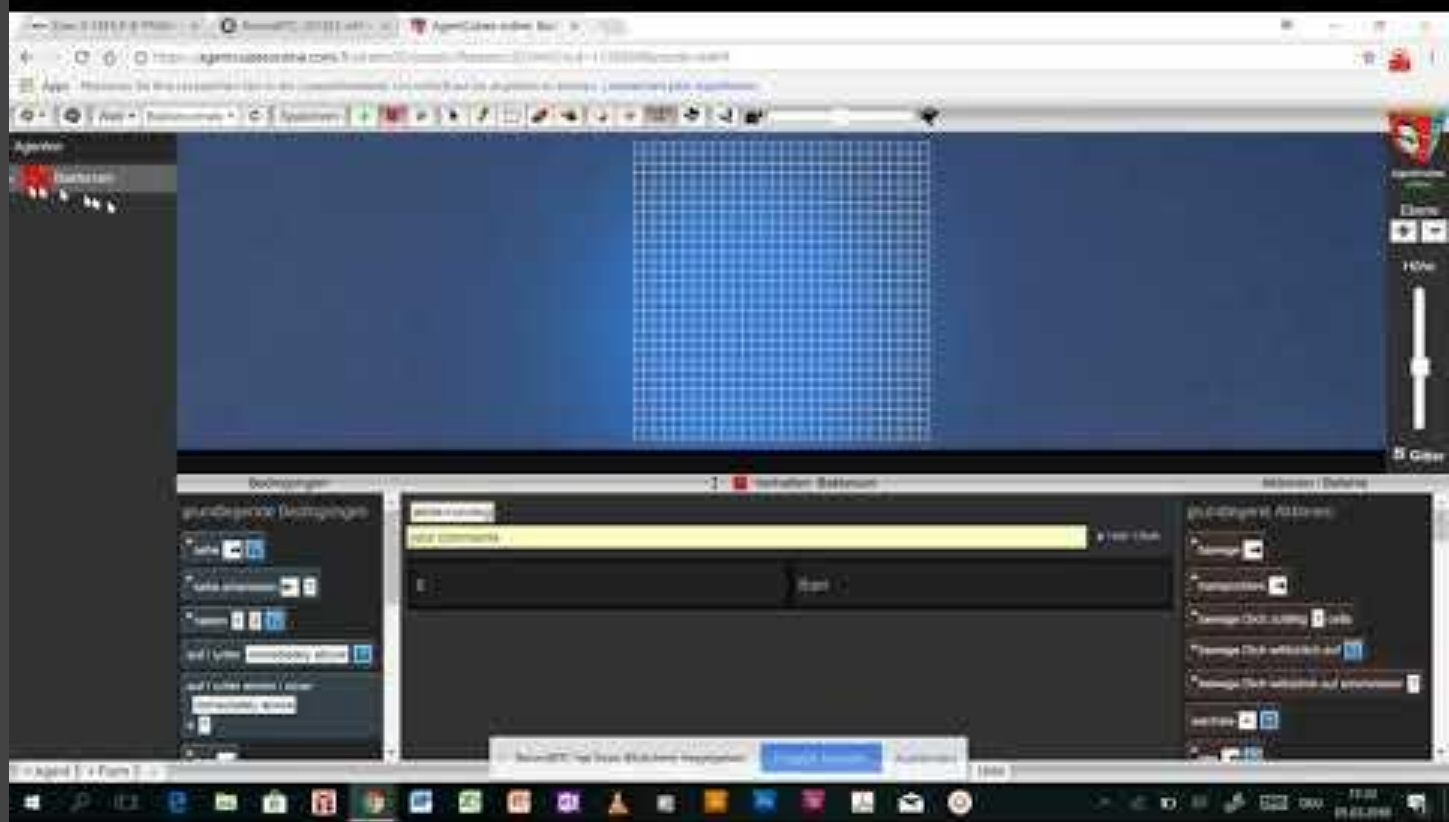


AgentCubes  
online

Demo

<https://agentcubesonline.com>

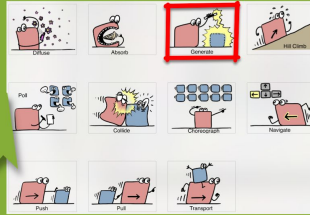




# Computational Thinking → Polymath

## 1. Abstraction

Break down problem into CT patterns



How quickly  
do bacteria  
grow?

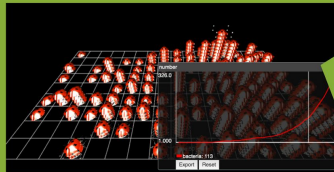
## 2. Automation

Program CT pattern

```
if once-every 1 sec then now → 🦠
percent-chance 10
if then move-random 1 cells
```

## 3. Analysis

Interpret data and answer question  
using visualizations



# 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.

# Thank you!

## Swiss Science Council SSC

- [Digital Competences](#)
- [Die Schweiz auf der digitalen Überholspur](#)



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

## Hasler Stiftung

- [Computational Thinking](#)

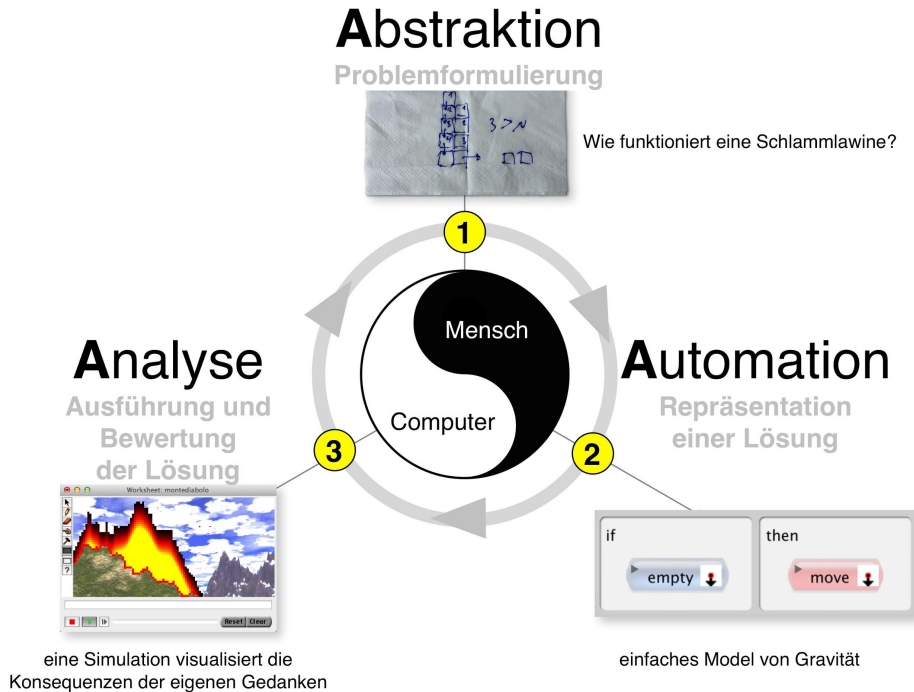
**HASLERSTIFTUNG**

# Which tool is better?





# Computational Thinking



NICHT

- Denken WIE ein Computer
- Denken über Computer

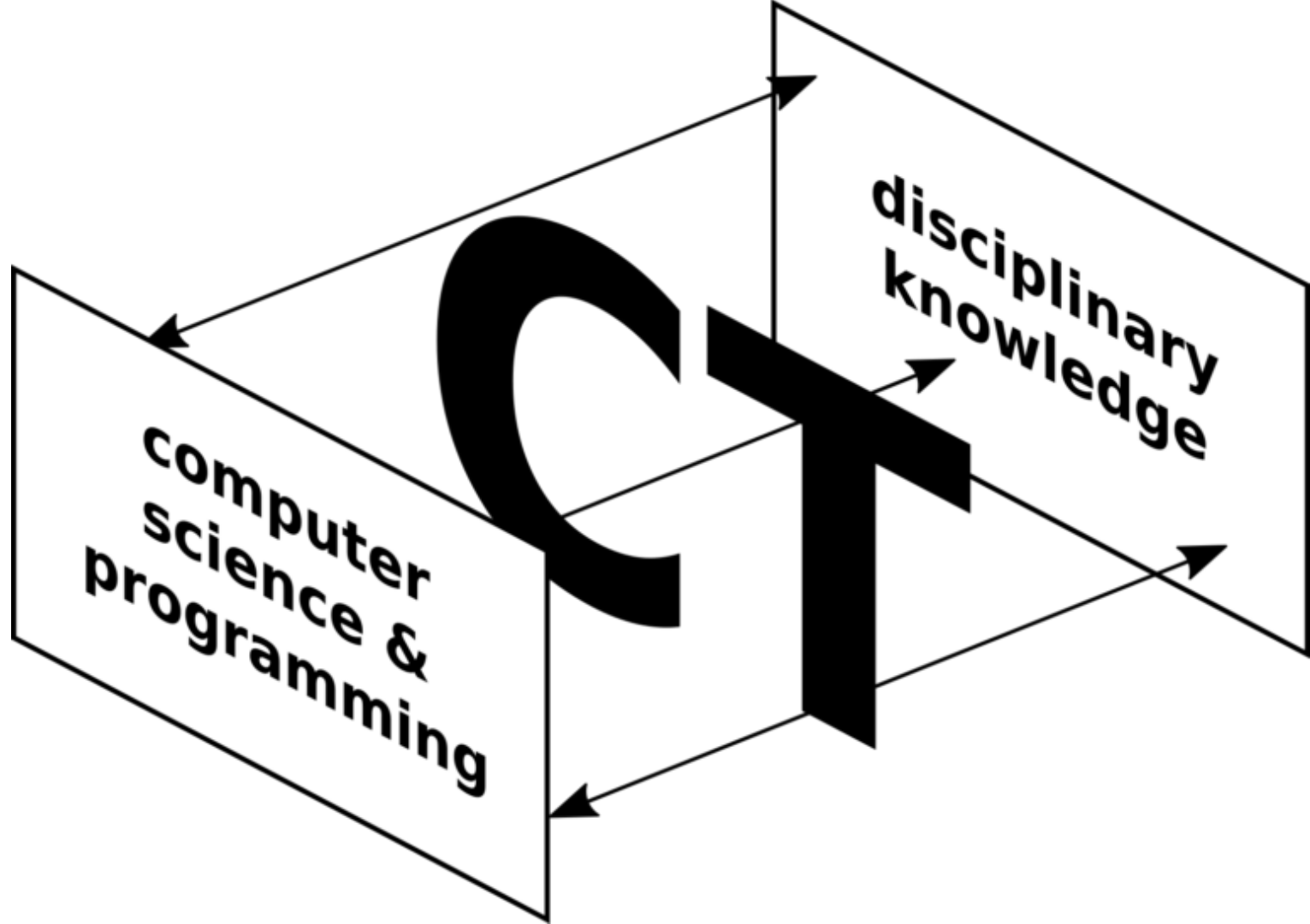
Sondern

- Denken MIT dem Computer

2

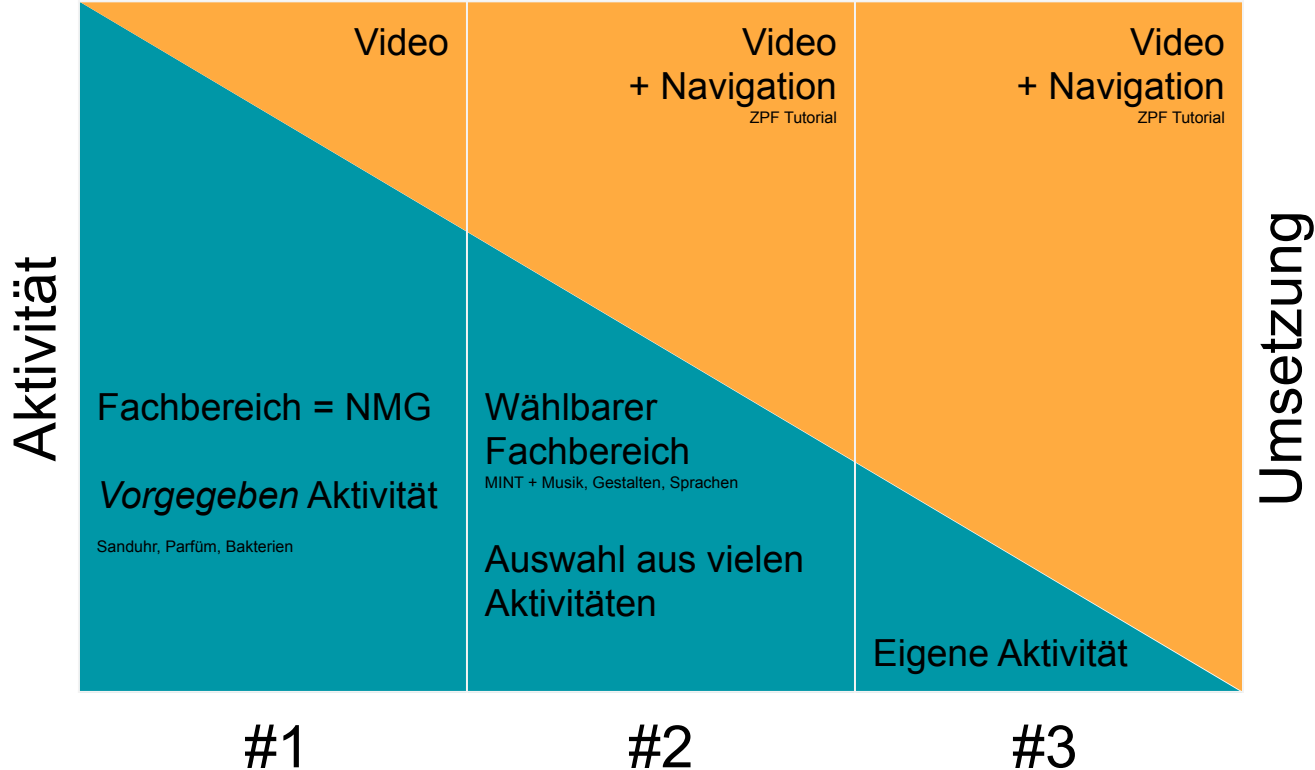
# Fach Didaktik





# Projectlets

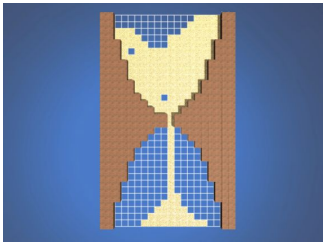
## Scaffolding



# Informatik + NMG

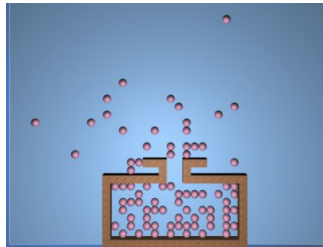
## 1: Sanduhr

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



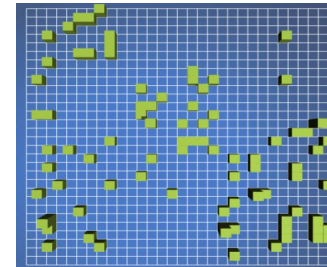
## 2: Parfüm

**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

### #2 create the agents



1. New Agent "Floor" as Tile
2. New Agent "Wall" as Cube
3. New Agent "Pac-Man" as inflatable icon
4. New Agent "Ghost" as inflatable icon

How Stop

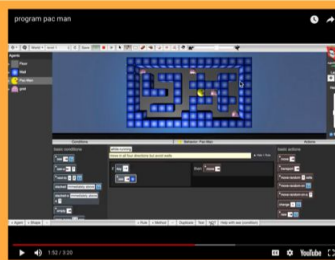
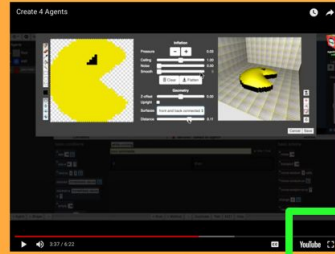
### #4 program Pac-Man

Program Pac-Man to be cursor key controlled (up, down, left, right)

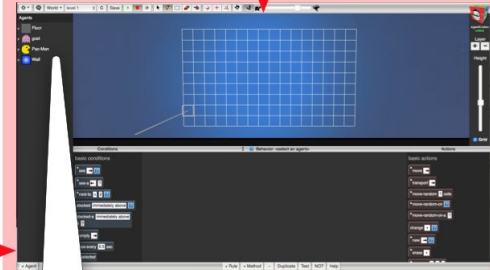
Pac-Man should not be able to jump onto walls

How Stop

## HOW Slides

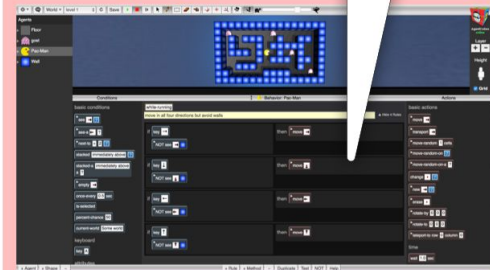


## prebaked Project



agents  
created

Pac-Man  
programmed



# Projectlet #2

- [Beispiel Musikinstrument mit MakeyMakey](#)

# Unterrichtseinheiten

- Alle

**Social Science A**  
**Computer Science**  
**Education**

Computational Thinking

course  
A

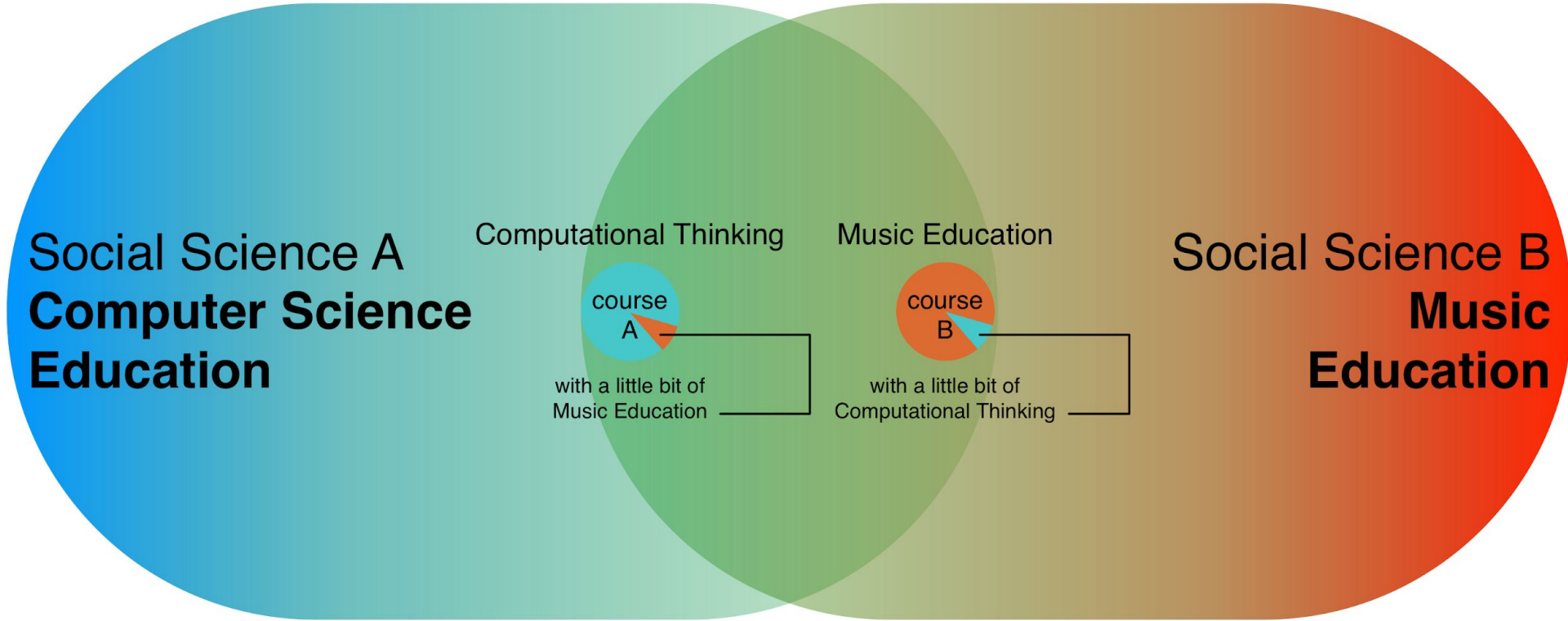
with a little bit of  
Music Education

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

