Computational Thinking ≠ 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
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
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

<table>
<thead>
<tr>
<th>Competencies</th>
<th>Renaissance Polymath</th>
<th>Digital Polymath</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Has expert-level competencies in many disciplines</td>
<td><strong>Meta-Competence:</strong> Has competence to acquire new competencies</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Peripheral perspectives of many disciplines including the attitude and ability to deepen knowledge–just in time–through the use of digital tools</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Solving Skills</th>
<th>Connects knowledge from different disciplines to solve problems</th>
<th><strong>Computational Thinker:</strong> can <em>think with a computer</em> combines human abilities with computer affordances</th>
</tr>
</thead>
</table>
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
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
But who are ALL these people?

da 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
AgentSheet & AgentCubes projects

City Traffic

the Sims

Bridge Builder

Pac Man

Forest Fire

Frogger

simulations
games

challenges

anxiety

ZPD

Flow

boredom

0% computational thinking patterns

100% computational thinking patterns

skills
Computational Thinking

Don’t

- Think like a computer
- Think about computers

but

- Think WITH the computer

Abstraction

Don’t

"how does a mudslide work?"

Analysis

Solution Execution and Evaluation

human abilities

Automation

Solution Expression

computer affordances

visualize the consequence of thinking

build simple model of gravity
“7 BIG Ideas”

1. Creativity
2. Abstraction
3. Data
4. Algorithms
5. Programming
6. Internet
7. Global Impact

Computational Thinking

Sek I

Sek II
Game Design & LP21

44%

LP21 Informatik Kompetenzstufen
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

(N1= 539, N2 = 471)

Effect Size (Cohen’s d): 2.05
Computational Thinking ≠ Programming
Computational Thinking Tools

- <switch to keynote slides>
**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

2. Automation
Program CT pattern

3. Analysis
Interpret data and answer question using visualizations

How quickly do bacteria grow?
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

Hasler Stiftung
- Computational Thinking
Which tool is better?
Computational Thinking

NICHT
- Denken WIE ein Computer
- Denken über Computer

Sondern
- Denken MIT dem Computer
Fach Didaktik
Fachbereich = NMG

Vorgegeben Aktivität

Sanduhr, Parfüm, Bakterien

Wählbarer Fachbereich

MINT + Musik, Gestalten, Sprachen

Auswahl aus vielen Aktivitäten

Eigene Aktivität

#1

#2

#3

Scaffolding
1: Sanduhr

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

#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 Slides

prebaked Project

agents created
Pac-Man programmed
Beispiel Musikinstrument mit MakeyMakey
Unterrichtseinheiten

- Alle
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