Towards deeper understanding of the roles of CS/ Informatics in the curriculum
TC3 Curriculum Task Force
Panel Discussion Vilnius July 2015

Mary Webb, *mary.webb@kcl.ac.uk* King's College London, UK.
Niki Davis, *Niki.Davis@canterbury.ac.nz* University of Canterbury, Christchurch, New Zealand.
Yaacov J Katz , yaacov.katz@biu.ac.il Bar-Ilan University, Israel
Nicholas Reynolds, nreyn@unimelb.edu.au Melbourne Graduate School of Education, The University of Melbourne, Australia.
Maciej M. Sysło, syslo@mat.umk.pl UMK Toruń, University of Wrocław, Poland

Draft Position Paper for comment at:
http://ifip-education.ning.com/group/task-force-curriculum-deeper-understanding-of-role

The Challenge and Key Questions

Curriculum change driven by resurgence of focus on Computer Science / Informatics as a key academic discipline.

TC3 (Education Committee of IFIP) needs to:

- review recent developments
- identify key issues and dilemmas
- propose ways forward.

- What is the range of skills and understanding that should be developed?
- Are such skills and understanding necessary for everyone?
- At what age should such education commence and to what extent should it be/remain compulsory?
- What pedagogical approaches are likely to be appropriate?

Tackling The Challenge in this panel

Curriculum Theory

Views from:

- UK
- New Zealand
- Israel
- Australia
- Poland

Emerging themes

What can we learn from curriculum theory? LEARNERS' ENTITLEMENT TO KNOWLEDGE

(Young, 2013)

Resolution to crisis in curriculum theory?

- “**Powerful knowledge**" is key

  specialised discipline-based knowledge which is different from the experience-based knowledge that pupils bring to school

- to harness the emancipatory capacities of learners the curriculum should take learners beyond their own experience

So curriculum design should start from the learner's entitlement to knowledge.

(Young, 2013)

Considerations and Constraints

We need:

- a coherent view of "epistemic ascent“ (growth of expertise)
- to introduce early in the curriculum (e.g. at primary level) all three major types of knowledge.
  - knowledge of individual propositions   implies
  - some understanding of the concepts that such propositions express   and this in turn implies
  - a significant ability to understand and make inferences within the subject (Knowing How to do something).
- a structured approach to progression in learning the basic facts and central concepts of the subject because knowledge is systematic in terms of 1) classification of its various conceptual elements; 2) the relationships between the elements and 3) the procedures required to gain and validate knowledge.
- understanding of constraints that the conceptual structure of the subject might impose on pedagogically and cognitively coherent schemata of epistemic ascent
- To clarify the relationship between the ways in which pupils learn by simulating procedures for the acquisition of knowledge in their learning and the actual processes of expansion of disciplinary knowledge e.g. project work involving the systems development life cycle

(Winch 2013)

The discipline of Computer Science
(The Royal Society, 2012)

- The discipline of Computer Science encompasses foundational principles, widely applicable ideas and concepts as well as techniques and methods for solving problems and advancing knowledge as well as a distinct way of thinking and working

Key Concepts
Programs
Algorithms
data structures
architecture
communication

Techniques and Methods
Modelling
Decomposition
Generalising with algorithms or data
Designing, writing, testing,
explaining and debugging programs

UK Computing National Curriculum Aims

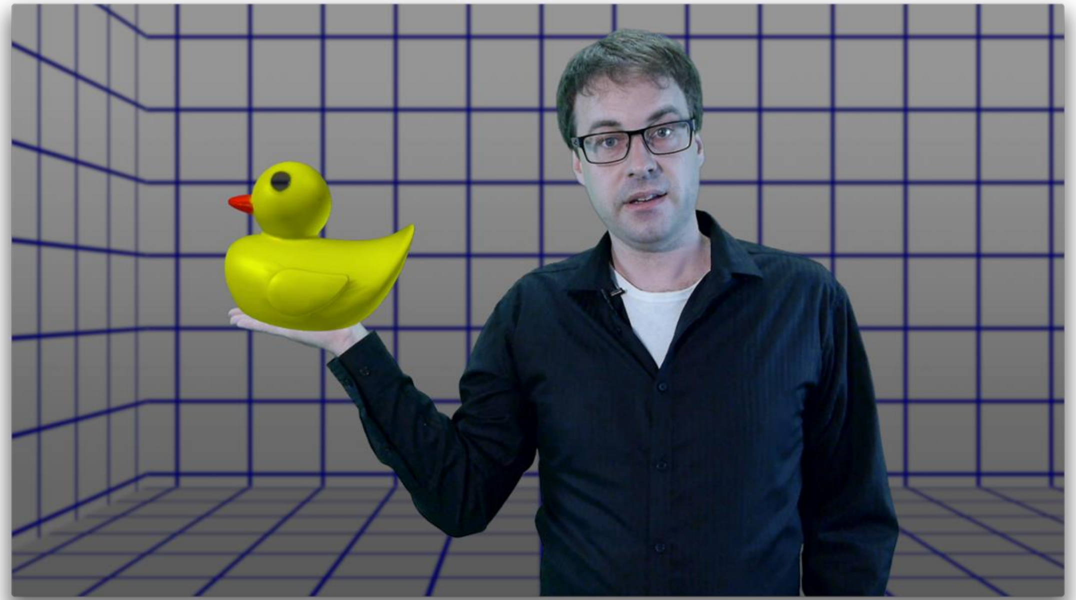
to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

Starting Age 5

Goals in NZ

Tim Bell



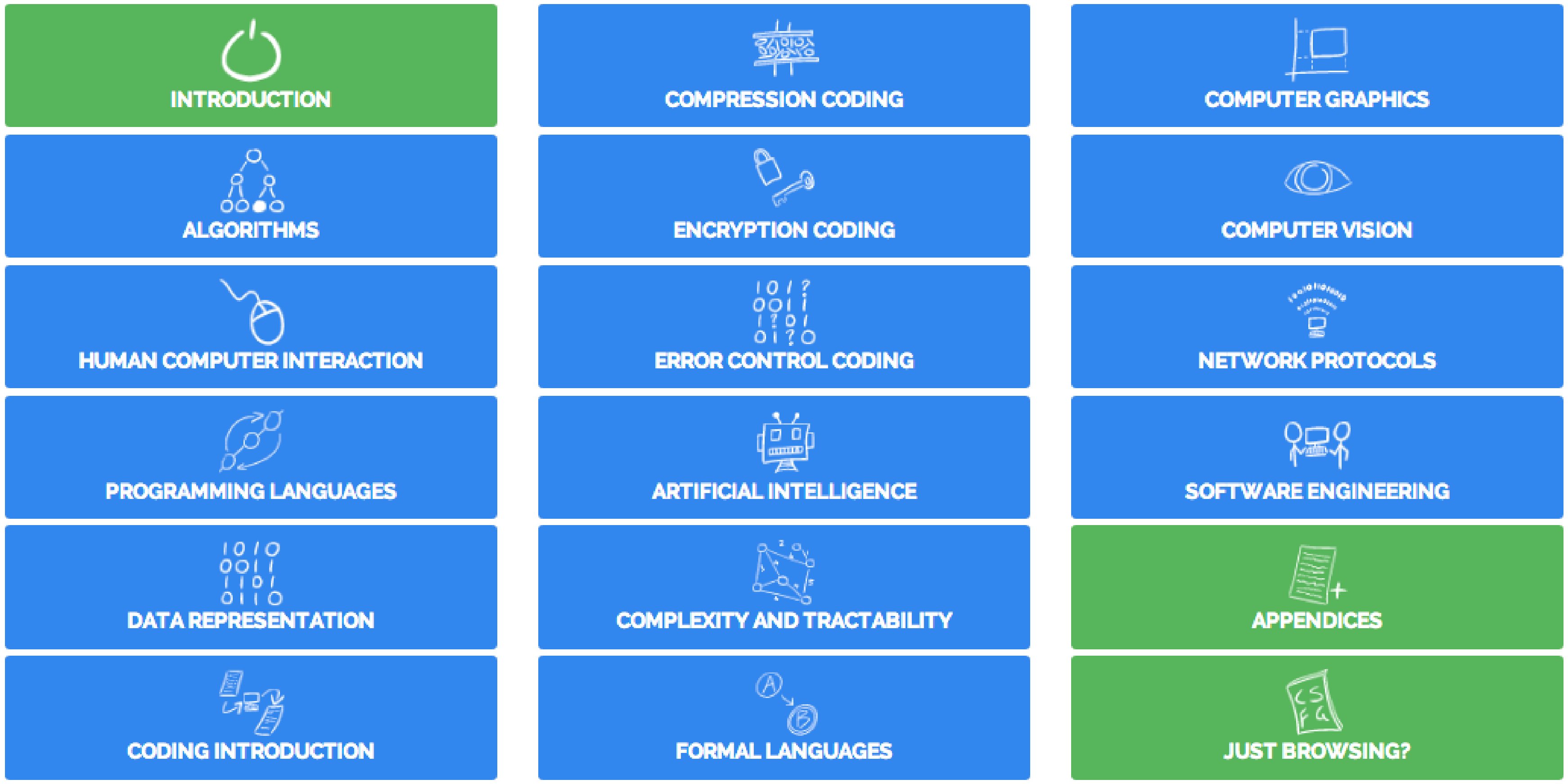
- Understand what the topic is
- Provide background skills
- Breadth, not depth

High schools

- US Advanced placement CS principles (pilot 2010-2016)
- IB diploma (2014)
- New Zealand (2011)
- Alberta CTS (2009) http://cacsaic.org/HowAlbertaGotCS
  https://education.alberta.ca/media/1074890/cse_sum.pdf
  https://education.alberta.ca/media/2160632/cse.pdf
- Germany (some states)
- South Korea
- Israel

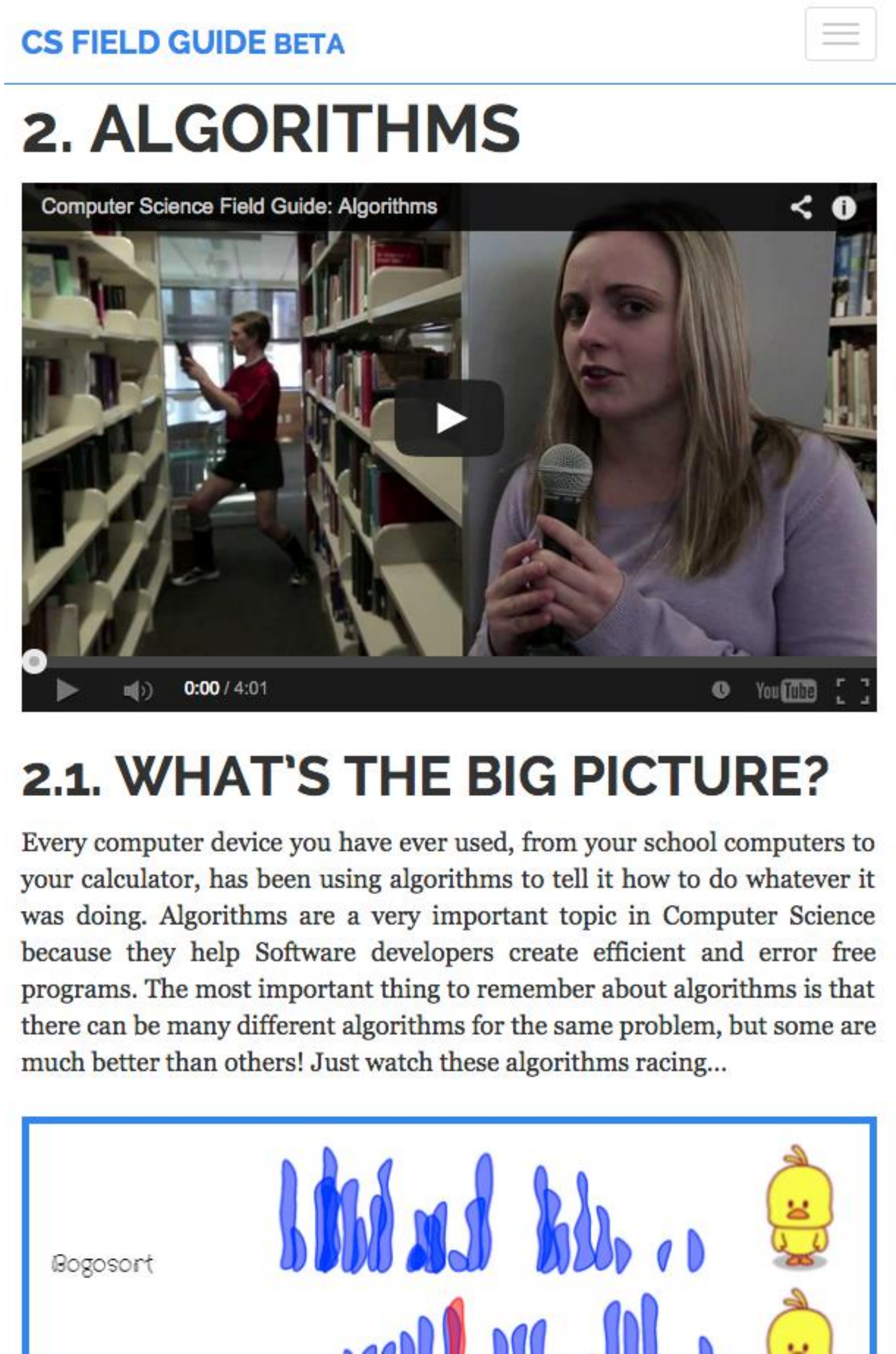
CS Field Guide: Open source “Text book”

http://csfieldguide.org.nz



NZ CS Field Guide

- Typically 150 to 300 users each weekday



CS FIELD GUIDE BETA

2. ALGORITHMS

2.1. WHAT'S THE BIG PICTURE?

Every computer device you have ever used, from your school computers to your calculator, has been using algorithms to tell it how to do whatever it was doing. Algorithms are a very important topic in Computer Science because they help Software developers create efficient and error free programs. The most important thing to remember about algorithms is that there can be many different algorithms for the same problem, but some are much better than others! Just watch these algorithms racing...

Girls' schools

THE IT GIRLS

SIMON ESKOW, TECHBLOG EDITOR 15 APRIL 2014, 2:44 PM

Like 12 Share 17 Tweet 27

Dunedin-based girls' school Columba College is taking on the notorious IT gender gap and winning, with many graduates entering a field typically dominated by men.

So what is the school doing differently and why is it working? IITP TechBlog Editor Simon Eskow talks to Columba's Digital Technologies teacher Julie McMahon to discover their approach.

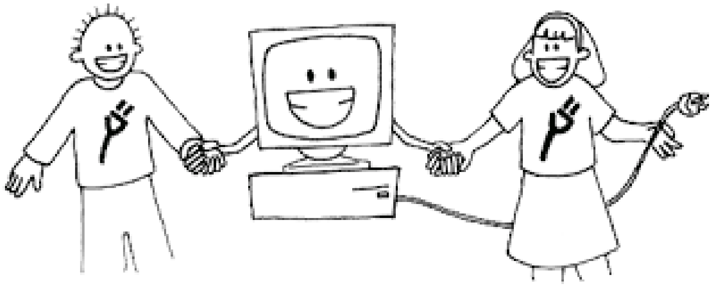
University of Canterbury Postgraduate Courses for teachers

csfieldguide.org.nz

cosc.canterbury.ac.nz/cs4hs

csunplugged.org

csanz.ac.nz



nzacditt.org.nz

tim.bell@canterbury.ac.nz

Panel - Computer Curriculum Considerations: A View from Israel

Prof Yaacov Katz
Michlala – Jerusalem Academic College
and
Bar-Ilan University

Paper Presented at IFIP TC3 Working Conference,
1-3 July 2015, Vilnius, Lithuania

Computer Science in Israeli Schools

- "Computer Science" is a major subject offered in a small but significant number of Israeli high schools
- These schools are located at the upper end of the high school technology track
- These are usually elite institutions where very talented students study computer science, physics, chemistry and biology (biotechnology)
- The graduates of these schools are snapped up by prestigious engineering faculties at Israeli universities

Computer Literacy in Israeli Schools

- Mainstream of high school students in public high schools do not study computer science per se as a stand-alone subject but are instructed in “computer literacy”
- Computer literacy is a major medium and methodology that contributes to instruction and learning of students in subjects included in the school curriculum
- Teachers develop knowledge and computer skills that allow them to indulge in teaching based on the transferral of knowledge within their sphere of expertise while utilising available technology

CS or Not CS

- The Australian Curriculum subject is not Computer Science.
  - What it is (or at least what we hoped it would be), is a way to create logical and clearly articulated pathways towards the study of CS, and a way in which all children can be exposed to and develop the knowledge, skills and understanding necessary to participate meaningfully, safely and ethically in our modern society.
- **This is an entitlement**

What is CS and why is it important?

- Political debate in Australia

"Coding is core skill for all children"

Is it really? Does this help our cause? How do we nuance this debate without contradicting those who are 'on our side'?

What are the other 'core skills'?

Some Questions from Australia and Elsewhere

- Is it reasonable to expect that 5 year olds can be taught CS? Is this actually what is or should be happening? Is it okay to hand over the teaching to Google and to Code.org?
  - What is different about Computational Thinking and why has it become so important?
  - How can the language and practice of the discipline be demystified? Is that what we want or need?
  - Is there such a thing as the 'crowded curriculum'? How do we get passed it?
- **Perhaps these are questions forTC3 and3.3**

ifip

Panel Emerging Questions

1) What is the range of skills and understanding that should be developed?

2) Are such skills and understanding necessary for everyone?

3) At what age should such education commence and to what extent should it be/remain compulsory?

4) What pedagogical approaches are likely to be appropriate?

1. What is the range of skills and understanding that should be developed?

Australia’s five Key Concepts:

Abstraction, which underpins all content, particularly the content descriptions relating to the concepts of *data representation* and *specification, algorithms and implementation*

Data collection (properties, sources and collection of data), **data representation** (symbolism and separation) and **data interpretation** (patterns and contexts)

Specification (descriptions and techniques), **algorithms** (following and describing) and **implementation** (translating and programming)

Digital systems (hardware, software, and networks and the internet)

Interactions (people and digital systems, data and processes) and **impacts** (sustainability and empowerment).

1. What is the range of skills and understanding that should be developed?

- Digital Technologies aims to develop the knowledge, understanding and skills to ensure that, individually and collaboratively, students:
  - design, create, manage and evaluate sustainable and innovative digital solutions to meet and redefine current and future needs
  - use computational thinking and the key concepts of abstraction; data collection, representation and interpretation; specification, algorithms and implementation to create digital solutions

1. What is the range of skills and understanding that should be developed?

- confidently use digital systems to efficiently and effectively automate the transformation of data into information and to creatively communicate ideas in a range of settings
- apply protocols and legal practices that support safe, ethical and respectful communications and collaboration with known and unknown audiences
- apply systems thinking to monitor, analyse, predict and shape the interactions within and between information systems and the impact of these systems on individuals, societies, economies and environments.

- **2. Are such skills and understanding necessary for everyone?**
  - Yes. But perhaps the better question is '**Are all students entitled to this knowledge and skill**'?
- **3) At what age should such education commence and to what extent should it be/remain compulsory?**
  - This must happen in the first years of schooling and continue to the end of compulsory schooling (or the point when students can make informed choices about their own pathways). Certainly up until the point in time when specialisation is allowed. **(Why?)**

4) What pedagogical approaches are likely to be appropriate?

- To answer this question it requires a strong understanding of specific contexts and students
- TC3's role and 3.3's role is to provide evidence about what works (in specific situations and contexts) and why. Not to create an argument about it
- This can be done through building examples of successful practice.

A New Computer Science Curriculum for All School Levels in Poland

Maciej M. Sysło

University of Wrocław, University of Toruń,

syslo@mat.umk.pl, http://mmsyslo.pl/

The School System in Poland (2008)

19 - 18
16 - 18
13 - 15
7 - 12
6

Tertiary education – University

Upper – high scho

Secondary education

Lo *mnazjum,* middle schoo

imary educati

stage

1st stage
integrated

Pre-school year

Informatics
anything rel omputers
chools

nformatics for nts, 1h
Informati – elective, 6h

and Informatics for all
with elements of algorithmics

From 2015-2016:
Informatics for all students
with elements of programming

computer Science education

Informatics introduced in our curriculum in 1985 for HS has never been removed !!!

2014: ICT/Informatics as a stand-alone subject on all school levels !!!

A new curriculum (as a ministerial document)

Structure:

- Introduction on the importance of computer science for our society in general and for our school students in particular
- Then follow the curricula for each level of education. Each curricula consists of three parts:
- 2nd part is the same in all curricula. It includes Unified aims which define five knowledge areas in the form of general requirements
- 1st part is a description of Purpose of study, formulated adequately to the school level.
- 3rd part consists of detailed Attainment targets. The targets grouped according to their aims define the content of each aim adequately to the school level. Thus learning objectives are defined that identify the specific computer science concepts and skills students should learn and achieve in a spiral fashion through the four levels of their education.

A new curriculum – Unified Aims at each Level

1. Understanding and analysis of problems (problem situations) based on logical and abstract thinking, algorithmic thinking, algorithms and representations of information.
2. Programing and problem solving by using computers and other digital devices – designing and programming algorithms; organizing, searching and sharing information; utilizing computer applications;
3. Using computers, digital devices, and computer networks – principles of functioning of computers, digital devices, and computer networks; performing calculations and executing programs;
4. Developing social competences – communication and cooperation, in particular in virtual environments; project based learning; taking various roles in group projects.
5. Observing law and security principles and regulations – respecting privacy of personal information, intellectual property, data security, netiquette, and social norms; positive and negative impact of technology on culture, social live and security.

ICT

A new curriculum – general comments

- remember: computer science ≠ programming
- concepts before tools, before programming

  problem (situation) – concepts – algorithms – programs (-ming)
- there are plenty of ways to introduce/teach computer science … without computers: computer science unplugged – Bebras tasks
- motivate and engage students by personalization

Programming

- programming is a tool, not a goal
- which programming language? – there are 3000
  - any, which can be used to introduce and illustrate concepts
  - introduce new constructs when needed
  - a program is a message for a computer and also to other people
  - different languages different programming methods
  - visual and textual languages and programming – when change visual for textual?

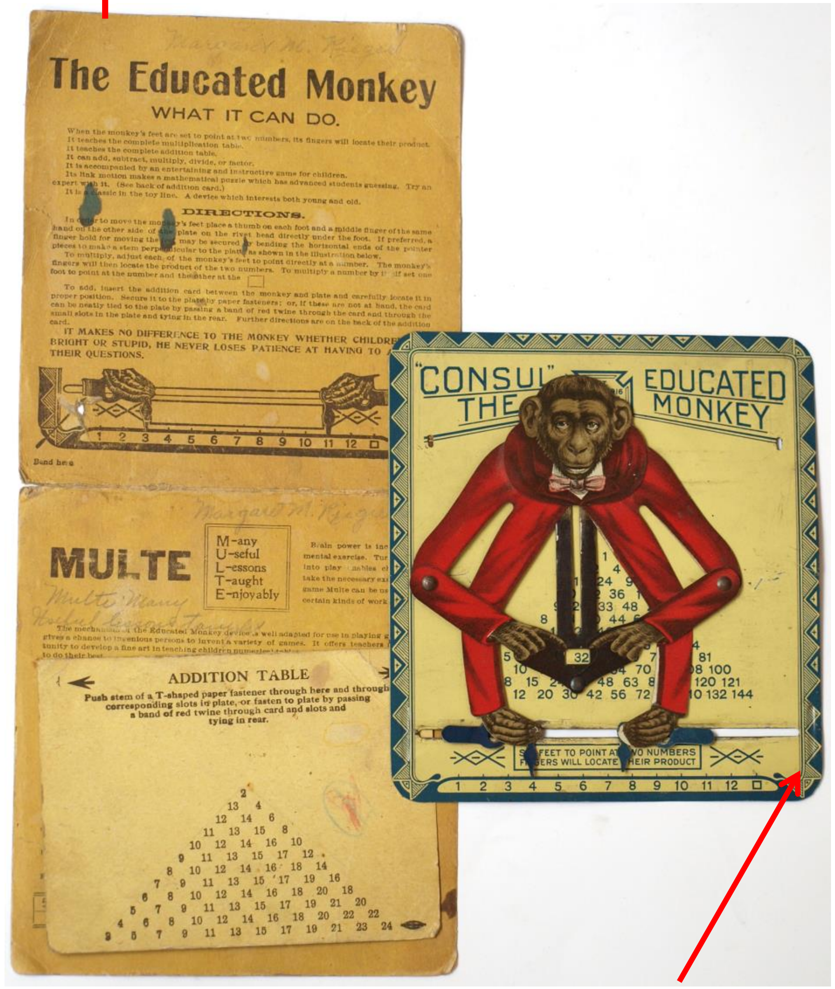
Introducing CS concepts – to kids (1-3, 4-6)

- the Childrens' universities
- We use all three forms of activities:
  - visual learning (pictures, objects, abstract and physical models, …)
  - auditory learning (exchange ideas, discussions, group work, …)
  - kinesthetic learning (physical activities)
- We work in environments consisting of three stages:
  - cooperative games and puzzles that use concrete meaningful objects – discovering concepts
  - computational thinking about the objects and concepts – algorithms, solutions
  - programming
- We extend, when appropriate, unplugged CS by adding … a computer
- Bebras tasks – the source of problem situations
- The Hour of Code – introduction to (visual) programming

Challenges

1. How to motivate and engage students through K -12, for 12 years!?
2. When and how to switch from visual to textual programming?

Playing with machines – the Educated Monkey


The Educated Monkey
WHAT IT CAN DO.
DIRECTIONS.
MULTE
M-any U-seful L-essons T-aught E-njoyably
ADDITION TABLE
"CONSUL" THE EDUCATED MONKEY

1916

For 5 x 5

How to:

- multiply two numbers
- divide two numbers
- factor a number

With another table, can be used for additions

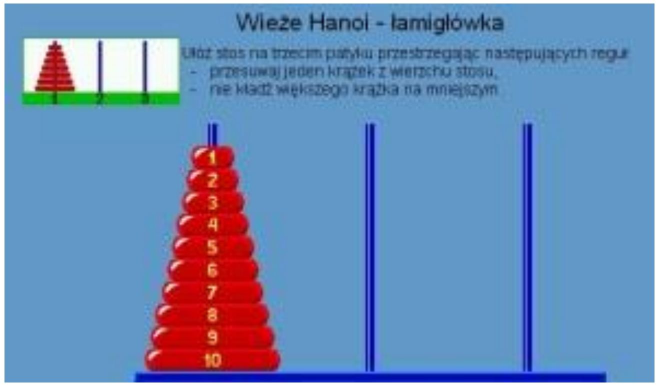
Concepts:

- maths basic operations,
- the use of a calculating instrument,
- algorithms

Children: where we can buy these instruments !!!

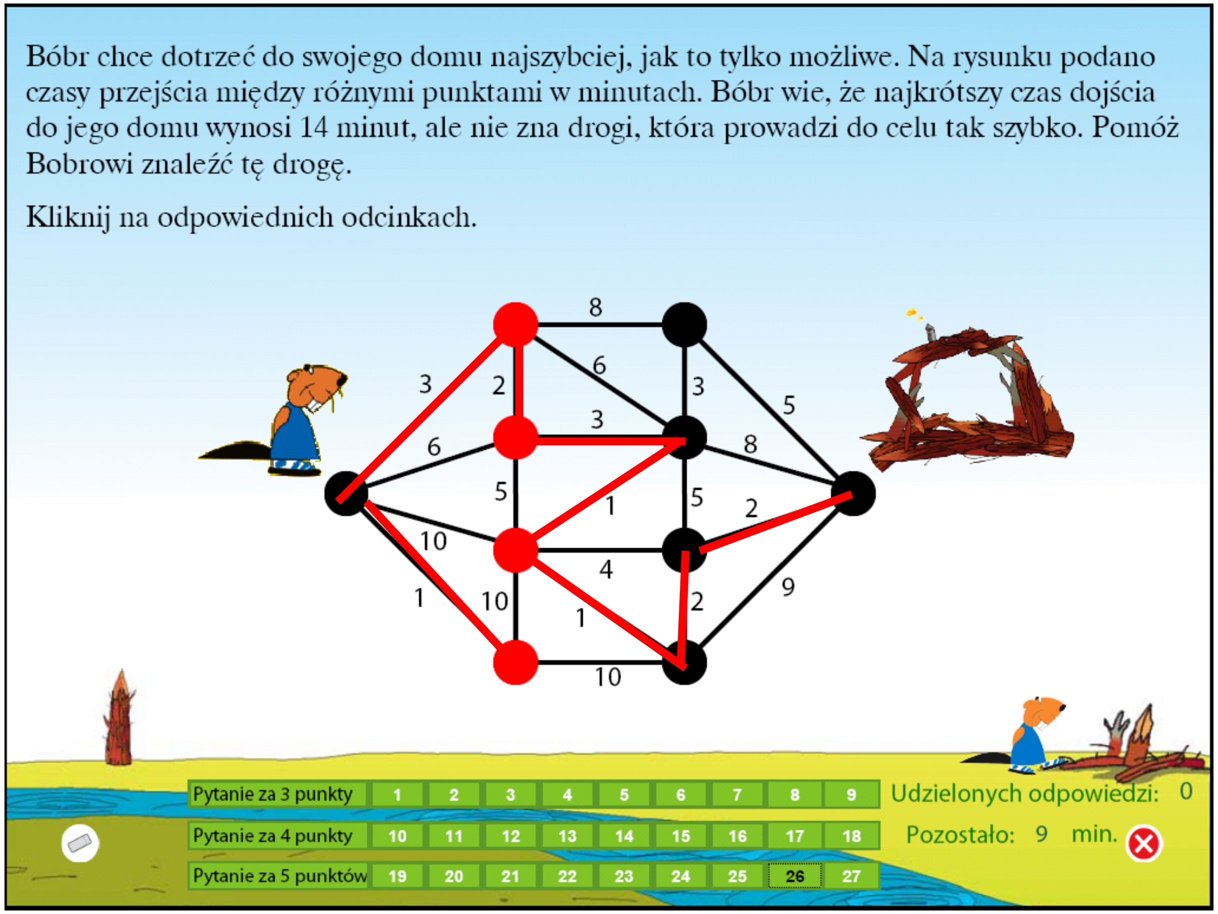
The Hanoi Towers

- The Hanoi Towers story
- first, kids play and try to find „an algorithm” and calculate the number of moves for different numbers of rings
- results: formulate algorithms and make a table with the number of moves
- then they play with (against) a computer program
- finally, they verify initial findings


Wieże Hanoi - łamigłówka

Ułóż stos na trzecim patyku przestrzegając następujących reguł
- przesuwaj jeden krążek z wierzchu stosu,
- nie kładź większego krążka na mniejszym

Shortest path – Beaver task



Bóbr chce dotrzeć do swojego domu najszybciej, jak to tylko możliwe. Na rysunku podano czasy przejścia między różnymi punktami w minutach. Bóbr wie, że najkrótszy czas dojścia do jego domu wynosi 14 minut, ale nie zna drogi, która prowadzi do celu tak szybko. Pomóż Bobrowi znaleźć tę drogę.

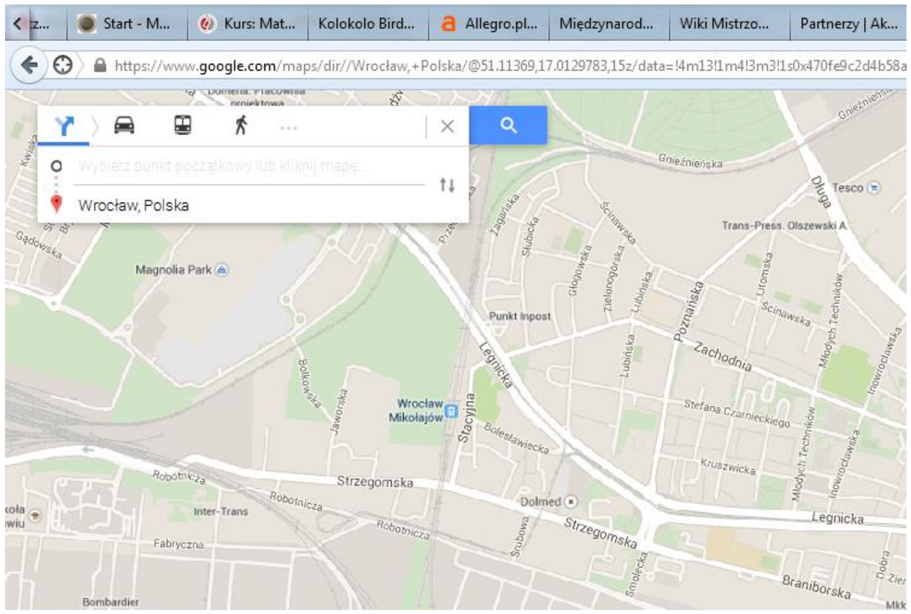
Kliknij na odpowiednich odcinkach.

Greedy algorithm – Dijksta’s Algorithm

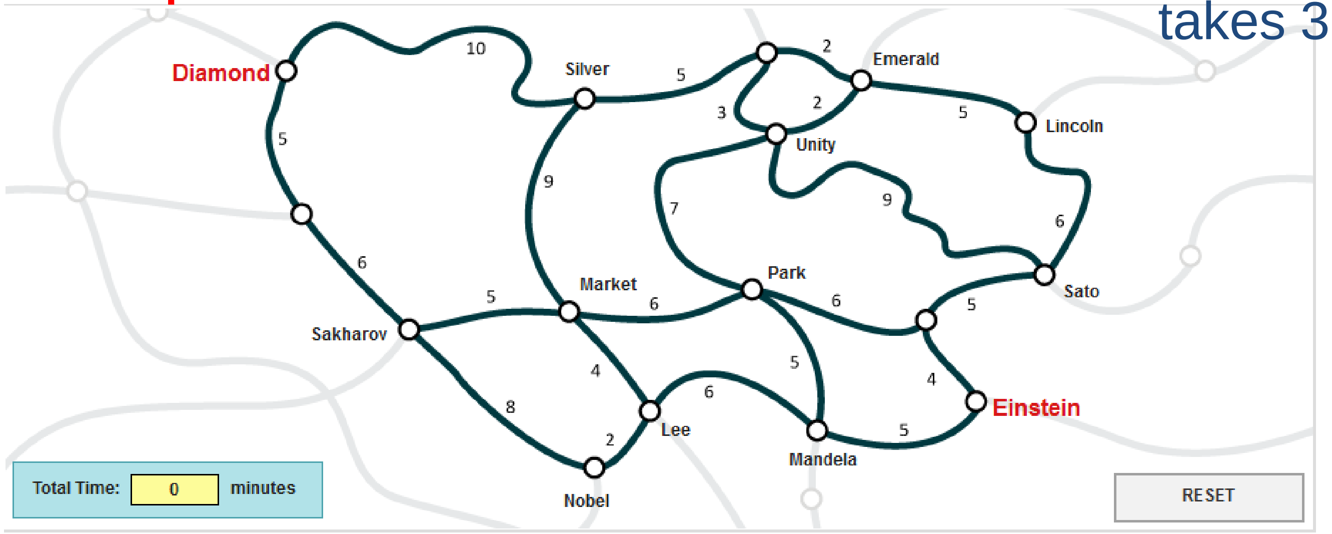
Shortest path

Kids are working with real situations– motivates them:

- Find your house and your school on the Google map
- Find your way to/from school
- Find shortest paths (distance and time) to/from school by different transportation means: on foot, by bicycle, by car
- Which is the shortest path (time/distance) to school?



Shortest path – PISA task



From Einstein to Diamond it takes 31 min – which way?

Concepts:

- graph models
- algorithm
- greedy approach
- shortest paths
- Dijkstra's algorithm

Typical approach, a greedy type: the nearest neighbor method.

It doesn't work !

However it works when you go from Diamond to Einstein !!!

Remember: Dijkstra's algorithm which a greedy method is optimal

Emerging Themes: Entitlement- computing curriculum for all?

- if learners are never introduced to Computing they will not know whether this is for them
- programming is difficult and it takes many years to learn to program so start early
- the ubiquitous nature of Computing: since so much of our lives is dependent upon Computing we need to develop the understanding and skills of Computing necessary to participate in society

Emerging Themes: New ways of thinking about curricula ?

- Should we try to fit the CS/ Informatics elements into existing curriculum structures or should we propose new structures and approaches?
  - Incorporating informal learning opportunities e.g. online, out of school, Code Club etc.
  - A more flexible, personalized curriculum ?

Emerging Themes: Curriculum content and balance

- computer science, IT, digital literacies and computational thinking
- social competences including project based learning and taking various roles in group projects

Emerging Themes: The importance of assessment?

$$\begin{tikzcd}
& \textbf{Curriculum} \arrow[dl, shift left] \arrow[dr, shift left] & \\
\textbf{Pedagogy} \arrow[ur, shift left] \arrow[rr, shift left] & & \textbf{Assessment} \arrow[ul, shift left] \arrow[ll, shift left]
\end{tikzcd}$$

- Typically a 3 way relationship between Curriculum, Pedagogy and Assessment
- In some countries assessment dominates and may constrain-
  - Creativity
  - Collaborative work

Emerging Themes: Curriculum change: risks and drivers

Drivers

- Economic benefits
- Learners' entitlement

Risks

- Failure due to inadequate teaching – Big challenges for teacher education
- Failure to understand how the conceptual structure of the subject determines pedagogically and cognitively coherent schemata of epistemic ascent (Winch 2013).