The dream of a lifetime
Shaping how our children learn computing

Simon Peyton Jones,
Microsoft Research and Computing at School
Aims

The National Curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles of computer science, including logic, algorithms, data representation, and communication
- 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.
Informatics education:
Europe cannot afford to miss the boat

Report of the joint
Informatics Europe & ACM Europe Working Group
on Informatics Education
April 2013

Informatics Europe:
 Walter Gander (chair), ETH Zurich, Switzerland
 Antoine Petit, Inria & ENS Cachan, France
 Olav Sand, National輪 Norway

Teaching computer science in France
Tomorrow can’t wait

Report of the
Académie des Sciences
(French Academy of Sciences)
May 2013
What we want
“Education should prepare young people for jobs that do not yet exist, using technologies that have not yet been invented, to solve problems of which we are not yet aware.”

Richard Riley
Disciplines

Skills
Disciplines

Ideas, knowledge, principles, techniques, methods

Maths, science, history, English

Skills

Artefacts, devices, programs, products, organisations, business

Presentation skills, metalwork, textiles, food technology, teamwork
ICT
Information and Communication Technology

Spreadsheets, databases, Powerpoint, web, internet, audio, video, e-safety

This was the situation in the UK during the 2000’s
Too much focus on technology
Not enough on ideas
Technology
Read
Consume
Use
Magic

Ideas
Write
Create
Understand
Knowledge
Discipline

Computer science

Skills

Digital skills

So what goes here?
Look!
No computers

http://csunplugged.org/sorting-networks
Follow the arrows to generate a sentence
Vision

Computer science is a foundational subject discipline, like maths and natural science, that every child should learn from primary school onwards.

Careful positioning

- *Ideas*, not *technology*
  Not even primarily about computers
- *Every child*, not just geeks
- *Educational* not *instrumental*:
  Not just a vocational/economic imperative
- *Discipline*, not *skill*
  In particular, not just coding
Computational thinking (Jeannette Wing)

Computational thinking is the process of *recognising* aspects of information and computation in the world that surrounds us, and *applying* tools and techniques from computing to understand and reason about both natural and artificial systems and processes.

- Computational thinking is something **people** do, not something **computers** do.
- Computational thinking is ubiquitous; it is useful in every profession, and in daily life.
Vision

Computer science is a foundational subject discipline, like maths and natural science, that every child should learn from primary school onwards.

Careful positioning

- *Ideas, not technology*
  Not even primarily about computers
- *Every child, not just geeks*
- *Educational not instrumental:*
  Not just a vocational/economic imperative
- *Discipline, not skill*
  In particular, not just coding
Every child

• Understand the digital world

• Understand the natural world

• Gain skills for almost any job
Vision

Computer science is a foundational subject discipline, like maths and natural science, that every child should learn from primary school onwards.

Careful positioning

• *Ideas, not technology*
  Not even primarily about computers
• *Every child, not just geeks*
• *Educational not instrumental:*
  Not just a vocational/economic imperative
• *Discipline, not skill*
  In particular, not just coding
1,000,000 more jobs than students by 2020

$500 billion opportunity

1.4 million computing jobs

400,000 computer science students

Computer science is a top paying college degree and computer programming jobs are growing at 2X the national average.
Computer science is a foundational subject discipline, like maths and natural science, that every child should learn from primary school onwards.

Vision

Careful positioning

- *Ideas*, not *technology*
  Not even primarily about computers
- *Every child*, not just geeks
- *Educational* not *instrumental*:
  Not just a vocational/economic imperative
- *Discipline*, not *skill*
  In particular, not just coding
The Telegraph

Teaching our children to code: a quiet revolution

The next wave of the digital revolution arrives next year, with every child in the UK being taught computer programming. But is Britain ready?
Vision

Computer science is a foundational subject discipline, like maths and natural science, that every child should learn from primary school onwards.

Careful positioning

- *Ideas*, not *technology*
  Not even primarily about computers
- *Every child*, not just geeks
- *Educational* not *instrumental*:
  Not just a vocational/economic imperative
- *Discipline*, not *skill*
  In particular, not just coding
The UK journey
CAS is formed

CAS curriculum (35 pages)
Computer Science: A curriculum for schools

Computing at School Working Group

http://www.computingatschool.org.uk

endorsed by BCS, Microsoft, Google and Intellect

March 2012
Eric Schmidt (Google) tells us we should be educating our children in CS!

Shut Down or Restart Report by Royal Society

CAS is formed

CAS curriculum (35 pages)

Review of the National Curriculum in England
Shut down or restart

• "The current delivery of Computing education in many UK schools is highly unsatisfactory"
• "Computer Science is a rigorous academic discipline and needs to be recognised as such in schools"
• "Every child should have the opportunity to learn Computing at school"
Eric Schmidt (Google) tells us we should be educating our children in CS!

Shut Down or Restart Report by Royal Society

CAS is formed

New curriculum (35 pages)

BCS invited to create a working group to draft the new Computing curriculum

Review of the National Curriculum in England

New curriculum (2 pages) published

New curriculum launches
Computing
Programmes of study for Key Stages 1-4

Aims

The National Curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles of computer science, including logic, algorithms, data representation, and communication
- 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.
Eric Schmidt (Google) tells us we should be educating our children in CS!

Shut Down or Restart Report by Royal Society

BCS invited to create a working group to draft the new Computing curriculum

None of this would have happened without CAS

Review of the National Curriculum in England

New curriculum (2 pages) published

New curriculum launches

CAS is formed

CAS curriculum (35 pages)
Lessons: what worked for us

• A singular focus: CS as a foundational subject
• An educational message, not just an instrumental one
• A single voice, not competing special interests
• An independent, grass-roots group, not an employers group, not a teachers group, not a higher-ed group
• Support from professional bodies (eg Royal Soc): influences civil servants
• Support from industry leaders (eg Eric Schmidt speech): influences politicians
• Don't wait for policy change: just get on with it
• Luck: the Review of the National Curriculum was hugely serendipitous
Opportunity and risk

Two national-scale experiments at once

1. Establish computer science as a brand new subject at school
2. Government explicitly standing back, inviting others to lead on implementation

3,500 secondary schools
17,000 primary schools
200,000+ teachers

Virtually no qualified teachers
Teachers hungry for support
Computing at School
• **CAS is a vibrant grass roots movement:** teachers, professionals, academics..

• **CAS is a community of practice,** to support, encourage, equip, give vision to computing teachers

• **CAS is independent:** speaks for the subject, not for teachers, or academics, or companies, or govt.

Computing at School launched (2007/8)
20,000 members
Growing at 500/month

- About 3/4 teachers, both primary and secondary
- But not all! Developers, IT professionals, parents...
- UK-centric, but open to international members
Loose, decentralised organisation

- Ten Regional Centres (based in universities)
- 89 universities
- 170+ Hubs
- 400+ Master Teachers
- 450 Lead Schools
- Masses of training events
- Amazing termly magazine
- Online community
Partnership

- Now part of BCS
  - Credibility/legitimacy
  - Legal status
  - Good for BCS too

- Work in partnership with dozens of other groups of enthusiasts: Raspberry Pi, Code Club, code.org, cs4fn, Apps for Good, Sonic Pi, etc

- And publishers/suppliers: Codio, Codecademy, awarding organisations, Rising Stars, Hodder, etc
Funding

- DfE gives us our baseline funding, currently around £750k/yr.
  - Always vulnerable, but absolutely crucial
- Employers have been generous; typically project funding:
  - QuickStart (Microsoft)
  - Barefoot (BT)
  - Tenderfoot (Google)
Resources
Quickstart Computing

• Professional Development toolkit
• Help teachers figure out how to design, develop and deliver imaginative and creative computing in the classroom
• 40,000 packs went into schools
• 75,000 teachers trained, 60,000 more in-flight

• http://quickstartcomputing.org/
• Vision: repeat the "BBC Micro" impact. From the classroom to the kitchen table
• A year-long campaign across all of BBC broadcasting
• LEDs, buttons, gyro, etc
• A million devices, one for each 12-year old
• Multi-employer partnership.
• My earnest hope: not just a one-year wonder
Culture

- A community, not a service organisation
- Fundamentally grass roots; bottom-up not top-down
- Only two full time staff
- Think "open source community" and "gift economy"
- Passion, optimism
- There is no “them”; there is only us
Glue
Encouraging participation and respect

• Site is designed to feel like a common room, not like a service station
• Front page shows recent conversations, not a form to search for resources
• Daily digest of clickable links
• Names, faces and places; a map shows where members, hubs, master teachers are
• Walled garden: no pupils!
• Like/Unhelpful buttons on posts; too many "unhelpful" clicks and the post is hidden
• Low barrier to entry for contributing resources
• You would think “someone must have studied this; there must be guidelines, examples, dos and don'ts” ...but I have failed to find much

• Culture of respect is easy to lose, and hard to regain

• Curation, quality control vs the huge opportunity of the crowd eg StackOverflow.com
Lessons: what is working for us

- Don't wait for central intervention: just get on with it
- Inspire, equip, empower volunteers
- Spend most of the money at the leaves
- But seek project funding for targeted central interventions:
  - Develop excellent teaching materials
  - Develop excellent assessment [a very high-leverage opportunity]
- Challenges
  - Scale.  SCALE.
  - Variation from place to place
  - Over-dependence on individuals.
Research questions

Evidence-driven reflection on the pedagogy and assessment of computing at school
The opportunity

• **The laboratory**: thousands of teachers are teaching computer science and programming to hundreds of thousands of children.

• **The teachers** are eager but under-qualified; and hence unusually open to collaboration, partnership.

• **Low hanging fruit**, because so little study has happened in this area.

• Many questions....
Research context

What do we already know? Literature survey.

How can we make what we know accessible to teachers?

What can we apply from other subjects, or from generic edu research?
Programming

What language? For what purpose?

Scratch, Kodu, TouchDevelop, Greenfoot, Minecraft, Python, HTML, CSS, Javascript...

Programming as a vehicle for learning computational/informational thinking, rather than as an end in itself.

Debugging, explaining, predicting, not just writing code.

The role of pseudocode (eg Haggis)
Pedagogy and assessment

Testing what we want students to learn, not just what is easy to measure.

Plugged vs unplugged?

Which concepts in which order for which age groups?

How do you assess computational thinking?

Discovery, or worked-out examples?
Resources

What do teachers need? In their perception? Is their perception "right"?

How to build on the cornucopia of resources that are already available? Quality control. Paths through the forest, quality control, review/feedback.
Challenges: funding and capacity

• Even if we had the funding, do we have the research capacity? How could we nurture/develop greater capacity?

• Need: a spectrum of rigorous, well-articulated research proposals
  • Variety of scale: fund a PhD student, do a 1-year study, ... upwards ...

• If they go un-funded, we can make a noise about it. Without the proposals, we can’t.

Do you have evidence of strong proposals going un-funded? Liam Blackwell (EPSRC)
Focus

• On developmental projects that *generate new ideas* (rather than arms-length RCTs)

• On *active engagement with teachers as co-researchers*, rather than as experimental subjects.

• On *how to embody the "big vision" of the new curriculum* (CS as a foundational subject) rather than a narrow focus on programming.

• On *assessment* as well as on pedagogy

• This is an international issue
Non-focus

• New programming languages or programming environments for teaching kids to code
Is all this important?

• If we do nothing, *something* will happen anyway. But probably not something good
• Individual teachers, companies, and even government, are not going to address these questions.
• Only researchers can.
Key links collected here

http://community.computingatschool.org.uk/resources/3084