WINSTON CHURCHILL MEMORIAL TRUST

THE MERCERS' COMPANY

Developing a Computer Science Curriculum in England: Exploring Approaches in the USA

Michael Jones FRSA
Winston Churchill Fellow 2015
October 2015

Contents

Acknowledgments

The Winston Churchill Memorial Trust has made this report possible. Logistical, financial and moral support enabled me to undertake a four-week research visit exploring computer science education in the USA.

Northfleet Technology College and the University of Greenwich are thanked for providing the time that enabled the research to be undertaken.

Although a teacher myself, the generosity of teachers and those involved in education continues to amaze me. Time, resources, accommodation and encouragement was never out of supply in the USA. Particular thanks is provided to the named individuals in the itinerary

Itinerary

Denver Colorado:

Steve Smith and Lake International Middle School
Kate McDonnell Overland High School Aurora

Boulder Colorado:

Brian Huang SparkFun
Jacob Segil University of Colorado
Marcie Mason Skyline High School Longmont
Ann Root Centaurus High School Lafayette
Daniel Hernandez West View Middle School Longmont
Axel Reitzig Innovation Centre Longmont

New York:

Leigh Anne Delyser AfSENYC (Academy for Software Engineering New York)
Don Miller SEP (Software Engineering Pilot) New York
Jon Mannion BASE (Bronx Academy for Software Engineering)
Mike Zamansky Stuyvesant High School New York

Massachusetts:

Josh Sheldon App Inventor Massachusetts Institute of Technology
Cynthia Solomon CSAIL (Computer Science and Artificial Intelligence Laboratory) Massachusetts Institute of Technology
Ben Shapiro Tufts University and Malden High School

"What is happening now is an empirical question.
What can happen is a technical question.
But what will happen is a political question, depending on social choices."

Seymour Papert 1980

Aims

- To learn from the practice and experiences of professionals directly involved in the delivery of computer science in Middle and High Schools in the USA
- To investigate the support mechanisms applied by outside agencies in support of computer science education in Middle and High Schools in the USA

Objectives

- To develop a strategy which will enable computer science in England to be taught to best effect
- To create a set of resources, to include standalone units of work and a continuous programme, which will enable the strategy to be implemented
- To develop a means of assessing the impact of the strategy thereby enabling a cycle of revision and editing in response to increasing capability in pupil understanding
- To build the strategy and resources into online, free at point of entry programmes which will be used by trainee, new and retraining educational professionals
- As teacher and pupil capability grows during the next 5 years the English and American pedagogical systems will be revisited to extract best practice
- To develop a network to include industry and academic representatives from the USA and UK

Background

I am a Secondary CS teacher working with Northfleet Technology College based in North Kent. The school has a heavy technology bias stemming from the days of Specialist status.

The use of technology lies at the heart of the school. All pupils have access to a laptop – we are a mostly wireless school.

With technology at the core of what we do, the school is an early adopter of computing being one of the first Network of Excellence schools in England. Many aspects of the new curriculum were an established part of the school prior to 2012.

As part of my work with Northfleet, I work with the University of Greenwich in developing and delivering their PGCE programme for trainee CS teachers. We are now in the third year of our partnership. This exemplifies the school's and my approach to education which is seen as a lifelong and cross phase endeavour.

As a member of Computing at School I run a Hub, support other schools in their hubs and provide curriculum support to schools in the South East.

I have concerns about how the new computing curriculum is developing. Whilst welcoming the mandate for CS the 'political question' of financial and logistical support for the subject is not advanced. As a member of the CSTA (Computer Science Teachers Association of America) I have long observed from afar how the subject is advocated and supported. The generous help of the WCMT has allowed me the opportunity to explore at first hand the American system that informs the content of this report.

Michael Jones October 2015

jonesm@ntc.kent.sch.uk / mpjones@cstalk.org.uk

www.cstalk.org.uk

The English Context – Position Statement

The English approach to computer science education is a bold one admired by many education systems across the World. However, this bold move makes many assumptions predicated on support structures that either do not exist or lack cohesion to succeed on a national level.

The curriculum is based on a terse document. It is a document that does not tell teachers how to teach. It provides teachers with a series of commands. As a document of Statutory Guidance, schools will be inspected and assessed on this. Key Stages 1-3 contains curriculum direction. Key Stage 4 statutory direction does not specify the direction in which a pupil may be directed. The language, however, lends itself to a number of interpretations. These will be explored further below.

In September 2014 all pupils, around 7,000,000 became entitled to an education that includes computing. The new curriculum evolved from the demise of the National Strategy ICT subject area and the realization that a more appropriate digital education was required. The ICT curriculum fell into disrepute and was removed in 2012. There were a number of repercussions from this. Trainee teacher programmes offering ICT as a subject were no longer valid, schools were left in a position of not knowing whether to continue with a digital education (most continued albeit to a lesser extent) and job roles changed with many teachers moving away from ICT. The changing landscape of terminal qualifications for pupils further removed ICT from the curriculum. Michael Gove, speaking two years after his disapplication of ICT, reminded his audience that "we would introduce a new computing curriculum, ambitious, stretching and exciting - drawn up by industry experts, allowing teachers and schools more freedom, designed to equip every child with the computing skills they need to succeed in the 21st century".

The demand for suitably educated people to undertake the design and creation of the artifacts is high. There is an international shortage of computing professionals. A cursory search will illustrate this point. It is a point that governments are starting to take seriously. An in-depth investigation into this can be found here[1]

However, simply mandating a subject for study does not thereby lead to problem resolution. It can, and has, led to issues that are becoming entrenched.

The 2014 curriculum introduction rests on at least three assumptions:

- That there is a sufficient pool of competent and confident staff to deliver the new curriculum

- That there are contemporary, long-term, high quality curriculum plans available to take this new curriculum area forward
- That there is a consistency of support throughout England

ICT is not computing. Elements of ICT can be placed under the computing banner but there are many topic areas that do not fit and this is causing tension. Chief amongst these is programming. Expecting an ICT teacher to teach programming is a risky expectation. Part of the problem with the old ICT curriculum was the use of ill-equipped staff drawn from a range of other curriculum areas. Not all feel confident to deliver the newer aspects of the curriculum. If we are to assure students that they are receiving a credible curriculum offer then we have a duty to not parade ill at ease teachers in front of them.

There are not enough trained computing teachers. For anyone involved in education this statement is hardly news. Official statistics paint a picture of failure to achieve recruitment targets:

Computer Science Teacher Recruitment[2]			
Year	Target	Actual	% of Target
2012/2013	795	480	60
2013/2014	620	359	58
2014/2015	610	519	85
All years	2025	1358	67

The three years 2012-2015 comprise the history of recruitment into computing. Overall we have a shortfall of 1/3 of suitably trained graduates entering into a career in teaching computer science.

Computer Science Education as Disruptive Innovation: Political Will Required

The introduction of computer science in English education is an example of 'disruptive innovation' and warrants careful support in the fledgling years. The term is taken from Clayton Christensen's concept and adapted from his idea that "*An innovation that is disruptive allows a whole new population of consumers"*[3].

If we apply this business model to education it is clear that the new curriculum orders are disruptive and innovative. Disruptive in that they displaced the ICT orders, seek to develop new curriculum models and marked a major change in curriculum content from a business studies, use of applications approach to a more technical STEM orientated model. The new curriculum certainly allows a 'new population of consumers' access. This was the situation in the 1990s when the ICT curriculum was launched disrupting the design technology curriculum along the way. Anyone involved in delivery of ICT will know of the speech delivered by Michael Gove at the BETT exhibition in 2012[4]. With adjectives such as 'boring' and 'slaves to the interface' being applied to describe the state of ICT education by the Secretary of State for Education it is clear that major disruption would follow, and it did.

Opportunity for Innovation

Where is the "innovative"? The intention of the new curriculum is to place pupils in the position of creators, able to develop ideas and work with technologies – many of which are emergent in the fields of artificial intelligence and robotics. We are, to use a phrase previously applied, at the stage where we are turning 'bricklayers to architects'[5]. We have the opportunity and the legal requirement to shift the paradigm in technology education.

Innovation in education, however, isn't simply a case of saying 'let it happen'. Problems to resolve include identifying target groups, acquisition and preparation of teaching resources, assessment benchmarking, staff development, recruitment and future planning. None of these come for free. At a time when real budgets in schools are taking a downward path there is a real danger that the innovation will be compromised.

All of the above problems are disruptive and innovative by themselves. Do we target particular ability groups? If we do, what rationale can we use to support this? After all, computer science is a legal entitlement to all pupils from Key Stage 1 to Key Stage 3 with a requirement that Key Stage 4 “pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career.”[6] It would take a brave school to require all pupils to undertake a GCSE in computer science at this early stage in the curriculum development – after all, we are a little over 12 months into the mandatory entitlement. Reasons to provide the opportunity for a select band are often based on what sounds like logical reasoning: introduce the subject: build expertise, confidence and understanding of the syllabus requirements, allocate existing staff in a timetable where demands are already at the boundaries and, not every pupil will want to ‘progress to higher levels of study’. The reasons are sound and apropos for the here and now. At some stage, however, schools will need to break free from the restrictive offer and expand opportunity.

Curriculum Value

With a legal mandate and support network in place we have, perhaps, the foundation to be innovative and disruptive. The bold move has been taken to include computer science as a science in exit benchmarks. A GCSE can be either used a science for the purposes of calculating GCSE science grades or used as a stand alone GCSE. That surely is innovative? To take the stance there is a new science is brave and possibly disruptive. For some, this strains the concept of what a science is and disrupts the paradigm. A useful argument for why and how computer science is a science is put forward by Rosenbloom[8]

Teacher Recruitment

For innovation to be accepted and adopted we need to ensure that we have a workforce capable of delivery. The disapplication of ICT and development of a replacement curriculum has certainly disrupted recruitment patterns. Where are the teachers? Much has been done by NGOs such as the Master Teacher Scheme from CAS, training in programming by the PiXL Club, SLE development by the National College of Teaching and Learning and countless support groups. Without this the curriculum would not have advanced as far as it has in such a short time. However, preparing existing teachers will only have partial effect. Not all ICT teachers have the inclination to transfer to computer science. Perhaps we should be mindful that part of the fall from grace for ICT was the massive use of non-specialists to deliver the curriculum. Making non-specialists deliver a subject area will inevitably lead to a move towards delivery of the ‘easy’ topics and, at best, superficial coverage of the more difficult areas and, at worst, complete avoidance of taxing curricula. We need to recruit new teachers, fresh with contemporary curriculum understanding. New teacher training programmes require this. Where potential trainee teachers do not have a sufficient background they are normally required to undertake subject knowledge enhancement programmes (SKE) as a condition of moving to their year of teacher training. If we have applicants without the correct background is it wise to expect a short, often 8 weeks, programme will turn a non-specialist into a specialist? The final evidence of this is some time in the future. Assuming there are robust exit criteria for SKEs then an assumption could be reasonably made that the route to specialism has at least been started.

Getting the right people into the teacher training programmes provides an opportunity to drive the innovation forward, encourage disruptive teaching and produce the further generations of computer scientists. Are we, however, attracting the required number and calibre of potential teacher? For a good graduate, 2i in computer science or related degree area, there is a mandatory entitlement to a £20,000 bursary with the option to apply for a £5,000 scholarship administered by the British Computer Society. With a first class degree the bursary entitlement is £25,000. You might think that with such an inducement there would be a surplus of applicants. Quite the opposite. Recruitment for the 3rd cohort of computing initial teacher training is below target. The scholarship programme recruited into August 2015 in order to award the limited scholarships. This is not surprising. Economies are on the up. Digital vacancies worldwide are increasing with computer science graduates likely to be offered salaries far in excess of those provided by teaching. It would take a sagacious and philanthropic young graduate to not be tempted by large salaries, potential to rapidly gain experience across the computer science sector and generally be rewarded for their study and rapidly pay-off student debt.

So, is a computer science curriculum disruptive and innovative? Yes. It is making schools and training providers think deeply and innovatively about how the curriculum can be supported. It has replaced ICT as an option in many schools – the number of accredited ICT programmes for Key Stage 4 has shrunk. It is innovative – the subject provides new content area that will continue to evolve. The need to continually evolve is a direct reflection of computer science as applied in the World about us – hardware and software innovations are commonplace, web tools move forward at a fast pace.

Computer Science Education in the USA – Logistics, Gaining Agreement and Advocacy

Scope of Provision

Computer science education in the USA is complicated by the largely non-mandatory nature of the subject. Where States do allow computer science to count it does so, typically but not exclusively, as a mathematics or science credit rather than as a distinct subject. There are, at the time of writing, around a third of the States who allow computer science to be taken with differing levels of content counting towards exit grades. The picture is further clouded by the ability of individual education districts to set their own curricula. This may mean that whilst computer science is recognised by the State education board, it is not a guarantee of provision. That computer science as a subject is recognized does not automatically entail that it is mandatory.

Autonomy

In the USA there are in excess of 14,000[9] school districts. Each of these has a degree of autonomy over what must be taught and what may be taught. Each State has an education board which provides some guidance and some mandate to districts. Districts largely decide on their education policy. This is an exemplar par excellence on devolving power locally. Colorado for example has a total of 179 school districts. Numbers are uneven, with the largest district, Denver, having a school population of approximately 88,000[10] and Agate having a population of 33[11].

It is perhaps surprising that there is very little in the way of mandating at a Federal level. The picture in the USA is one where States do not have to adopt national standards. This is evidenced through the adherence to the new Common Core Standards[12]. The Standards are a set of attainment rubrics in English, arts and mathematics designed to ensure commonality across the nation.

In England, the National Curriculum for every child is decided and, assuming it passes into law, each school district and thereby state-funded school is required to follow it, almost all. Academies have the right to not teach the National Curriculum. This is not the place to discuss the finer points of what curriculum, if any, an academy should follow. Suffice to say, that if an academy does not follow the National Curriculum it will still need to show that pupils are making at least as good progress across subject areas as their non-academy peers.

Arkansas – a Case Study in Mandating Computer Science

So, how does computer science work in the USA, where it is offered? The most recent State to mandate for Computer Science is Arkansas[13]. Schools will be required to offer appropriate programmes of study. Some schools already offer programmes of study but until the Act was passed into law they were not required to. There is a catch with the Arkansas decision; capability in computer science will count towards mathematics outcomes not as a subject in its own right. This approach can prove unsettling for some teachers. Having your specialism count towards another does not provide an environment in which ownership of the subject can take place – the students could just as easily incorporate that curriculum strand into mathematics thereby providing an environment for stifling development of computer education. However, the idea warrants further consideration. In England, computer science can be counted as an additional science[14] or as a subject in its own right. Given the system used in England to calculate value and the quality of outcomes it is perhaps feasible to link computer science to mathematics outcomes.

Approaches to Teacher Shortages

Where Arkansas schools are not able to provide programmes, as a result of lack of qualified teachers, students will be able to complete courses online. Perhaps this is a means for England to adopt whilst we build our stock of competent teachers. The technology to create online programmes of study is well tested. The growth in MOOCs has shown this, CMS in the form of Moodle are available. The higher you move up the Key Stages the level of understanding required by teachers increases thereby shrinking the pool of available teachers. In such a situation using a pool of teachers to act as online guides begins to look like a viable means of delivering high level content. It is becoming commonplace for trainee teacher applicants to be directed towards SKEs. Canterbury Christ Church University, for example, run SKEs in computing and many other subjects. These have the express purpose of ensuring that potential trainees fill gaps they may have. A blend of monitored and assessed digital lessons are supported with a small number of face-to-face meetings and weekly online tutorials. This lends itself to scalability. Many schools in the USA use online course environments, eg code.org. The move in Arkansas makes recognition that the subject is important and that delivering it is unlikely, at least in the medium term, to be fully covered by traditional terrestrial methods and that online working is necessary and sponsored by the State[15].

NGOs in England and the USA

Just as the UK has a national, grass roots organization in the form of CAS, the USA benefits from the existence of CSTA. The similarities are many: they have NGO status (but both receive support from respective governments), are sponsored by commercial organisations, the membership is comprised of mostly working teachers and those with an interest in CS education, it is supported by a very small employed administrative and management base, they provide a forum for dissemination of good practice in the form of curriculum materials and discussion forums. Neither organization has been in existence for long. CSTA pre-dates CAS by approximately 6 years, being formed in 2004.

Advanced Principles – Changing Focus to Reflect Digital Need

Advanced Principles Computer Science can be viewed as broadly in line with the English A Level programme. The cohort taking the qualification matches the 6th Form profile in England. There is no real equivalent of GCSE – this is reflected in the large scale lack of mandate in the USA.

Just as A Levels have undergone an overhaul, AP CS has seen major changes. The current AP course in the USA is highly Java based and assessed mainly through formal examinations. There is very little scope to study the broader range of CS topics built around the "7 Big Ideas"[16]

The new AP programme, with first examination in 2017, sees a radical shift in focus. Assessment still retains a formal examination but included are 2 'through course assessments'. Consider these as coursework/controlled assessment. The reliance on Java is gone with students able to develop skills in a range of high level languages. At least part of the philosophy behind this is the belief that this will attract more students and therefore more schools.

Changes in AP CS are quite the opposite to those we are seeing in A Level. Whilst the English system moves towards a larger emphasis on formal examinations, typically 80% of final marks, the American system is placing greater value on how a student can work though large scale projects.

The value of AP CS is variable, as it stands. Some universities, eg the University of Colorado see AP as covering some content for the first year of an undergraduate programme whilst it was reported to me that some universities disregard AP success.

AP in practice is examined throughout this report.

Crosswalking – Mapping Skills Across the Curriculum

As with England, there is a continued push to establish a curriculum. Although not established, the CSTA have been advocating what are termed ‘crosswalks’[17]. Essentially, these are mapping exercises where standards in other subjects are matched to computer science competencies. The standards matched to are derived from STEM Cluster Topics, Common Core Standards and Partnership for 21st Century Skills.

A word on what subject areas are used:

> **STEM Cluster Topics**: derived from agreed technology areas
> **Common Core Standards**: derived from English language arts and mathematics
> **Partnership for 21st Century Skills**: broadly akin to work and communications skills including the use and application of technology.

CS Competency statements eg “Explain how sequence, selection, iteration and recursion are building blocks of algorithms” CT.L3A-03, can be accredited towards multiple competency identifiers in all three of the mapping areas. CT.L3A-03 is credited as being evidence for Communications and Academic Foundations to name 2 of 10. The mapping, in most of the CT statements is one way. The activity that satisfies the computing element may also contribute to other standards but there is very little in the way of reciprocity – a mathematics activity will not ordinarily satisfy a computer science competency.

Crosswalking in England?

The English curriculum lends itself towards a similar approach. We have three main strands: computer science, information technology and digital literacy. What we do not have is a mapping of the strands across other curriculum areas. This is not a radical or new idea. Many schools followed the route of mapping ICT into subject areas with a requirement that those subject areas be responsible for ensuring that capability was developed.

Defining what is Unique in CS

Crosswalking is not a total solution, some aspects of computing, eg networking and programming, do not lend themselves readily to cross-curricula deployment unless there is a further push in developing necessary skills in non-specialist teachers. For example, data logging as an activity is a science activity that has elements of ICT capability in the guise of using hardware and software to record speed, light, temperature and so forth. This is essentially a plug and play activity: unbox the data logging equipment, plug it into a computer, deploy the sensors and press record. A more computer science focused take on this would be to get the pupils to create a program that interfaced with hardware that recorded and reported back the data. An already crowded science curriculum is unlikely to have time to allocate programming as a part of the data logging skillset.

Could we credit language competency as part of the computing curriculum in an English setting? In the USA, computer science is predominantly recognised as contributing to science and mathematics competencies when it is taught. This is not exclusive the case. A number of States, notably Texas, are using computer science competencies as credit towards foreign language capability credits.

This is an interesting take on computer science. Accepting that programming requires proficiency with the syntax of the particular language being learnt: knowing when particular structures must be applied and could be applied are comparable to formal and informal language. For example, Python requires a high degree of formality down to indentation of code. Java on the other hand requires the use of keywords and structures, eg correct use of braces, but allows complete freedom over indentation style. It is not too far-fetched to claim that a student who can code in Java, Python and a C variant is multi-lingual. However, is it enough to satisfy elements of MFL competency? Allowing it to formally count may not be such a bad idea.

What can we take from this for use in England?

Mapping across competencies is certainly possible. A key competency for Key Stage 3 mathematics is “calculate and solve problems involving: perimeters of 2D shapes (including circles), areas of circles and composite shapes”. This is a fundamental of programming as is developing awareness of number data types undertaken through calculation of shape properties.

Mathematics as a Central Paradigm

In England we have the opportunity for two-way support. Key Stage 3 computing has a requirement that pupils “understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal] ”.

Understanding and working with binary is a fundamental for any pupil taking a GCSE computing course. Alongside the need to be able to demonstrate how to interpret and manipulate binary numbers, students also need the binary understanding to comprehend how computer systems work. Binary further serves as the gateway to working with hexadecimal at Key Stage 4 and Octal at Key Stage 5. Binary competency is not a requirement for the mathematics National Curriculum Orders yet the core competency of working with binary is as near to mathematical thinking as you are likely to get. Addressing the skills and understanding of number systems early will provide the foundation for the Key Stage 4 Order that "All pupils must have the opportunity to study aspects of information technology and computer science". Crosswalking the skills and techniques at an early stage, even before Key Stage 3, will enable the subject to mature.

Computer science teachers currently find that they are in position of being mathematics teachers for at least some aspects of their role. Maybe it is time for a more cross-curricula approach to delivery of topics across subjects. To undertake this it would require a formal acknowledgement and acceptance from the Department for Education that a subject competency can be achieved outside of traditional scope. Who undertakes the mapping and monitoring required for this is a school level decision that has resourcing and skill level implications.

Computer Science or Computer Engineering?

Case Study Westview Middle School

Programme Structure

Daniel Hernandez introduced me to the concept of Summer School at Westview Middle School. Pupils are offered the opportunity to undertake a range of activities during the summer vacation. At Westview this covered a weeklong programme of CS. In the mornings pupils had the opportunity to explore robotics using the Vex system. In the afternoons pupils developed their coding skills in RobotC. Some pupils took advantage of this and stayed for the whole day sessions. This provided them with experience in creating physical artifacts and digital representations. Both are activities that required pupils to apply problem-solving skills including abstraction, decomposition and algorithmic thinking

Method

I had the opportunity to visit the school at the start and towards the end of the summer school. This enabled me to witness the growth that can occur over a short period of time (albeit intensive). Students left with skills equal to those seen in many older students.
In providing this facility pupils acquired in the region of 30 hours intensive CS teaching and experience. Teaching is supported through a mix of direct teaching and independent learning and online materials. Daniel has supplied a link to his support site[36].

Funding

Whilst unusual in its scope and breadth, the provision at Westview points to what can be achieved with support and political will. The sessions are partially self-funding via parental contribution but receive support from the District.

Curriculum

The curriculum at Westview is comprehensive in its scope. All pupils in Grade 6 undertake a computing program. After this it becomes an elective. The key to the success is in the breadth of what is undertaken. Pupils are provided with access to a range of technologies and are expected to apply an engineering approach to their activities. The learning has a highly synoptic feel. Pupils are expected to prototype artifacts – both physical and digital. Time is given to consideration of suitability and review/redesign.

In the range of showcases the school exhibits the combined physical and digital products. It is as likely that if a part becomes unusable the pupils will recreate it using 3D printing technology.

Transfer to England

- Builds a synoptic approach to computer science
- Underpins late Key Stage 3 and Key Stage 4 skills
- Removes the software/hardware division
- Requires greater focus on skills and pedagogical approaches earlier in the teacher preparation cycle

Case Study Centaurus High School

Partnership Working

In partnership with Sparkfun, Brian Huang, and the University of Colorado, Jacob Segil, a group of (English equivalent) Key Stage 4 students worked on computing problems incorporating hardware and software. They were led by AnnRoot. The tasks students engaged in were as much engineering focused as computing. Students were required to develop an understanding of the physics of materials and make decisions about what to abandon and when to abandon it when the decision was made that an approach was not viable.

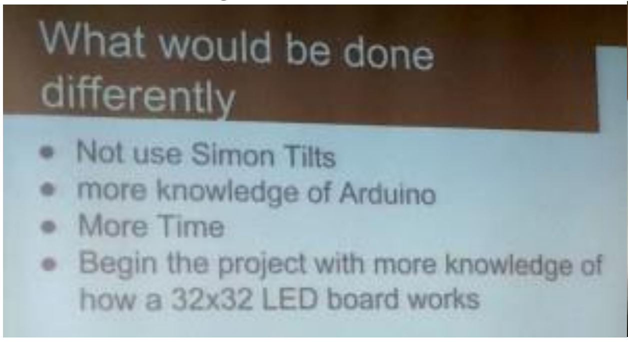
Focus and Problem Resolution

No matter what the project, a piece of hardware named 'Simon Tilts' was designated as part of the project requirement. Essentially a motion sensor, the students attempted to incorporate this into projects such as a 32×32 LED grid game and hand pong (a take on the perennial favourite). Many students found that the hardware component would not work for them. The most significant point here is that they persevered and they made the decisions to adopt a different approach by rejecting the motion sensor approach. This was not taken to signal a failure. Most acknowledged that the hardware might have viability but that this would require further work. In a system where we put much focus on getting it right it is refreshing to see opportunity for getting it wrong without penalty.

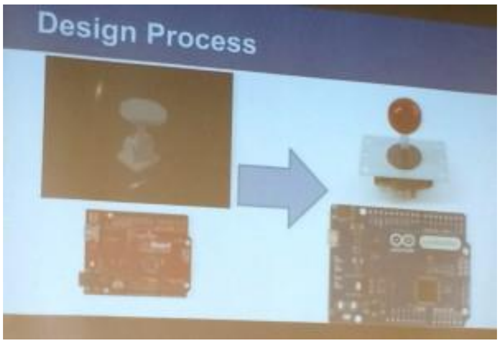
Projects

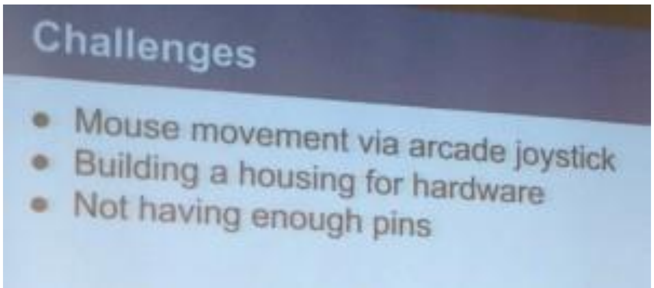
The projects: armed with a basic set of hardware, notably Arduino boards, breadboards and a range of add-on shields the students worked in groups, with a few exception of lone pioneers, to decide on and build their solutions. I was witness to "**Space Dots**" – a game based around a 32 X 32 LED grid. The objective was to capture positive dots and avoid negative dots through tilting the board. Abandonment of the motion sensor meant that an alternative, button controlled approach was taken. The last part of the project process was to present findings to the group.

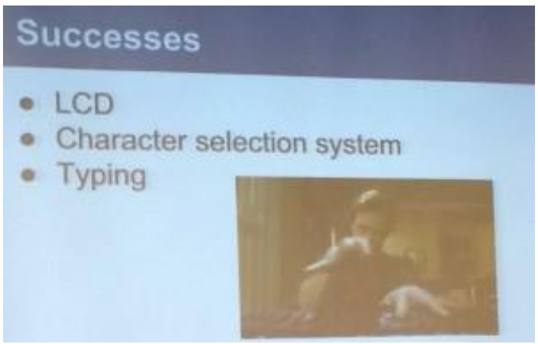
Summation was good with clear identification of processes:



Archangel Joystick is an intriguing project. The essence of this was to use a joystick to type. The position of the joystick within a predetermined grid allowed the user to select a range of characters. The characters are then output to an LED screen. This is no mean feat. The communication between the board, joystick and LED screen is complex with a need to acquire some level of understanding of character encoding.







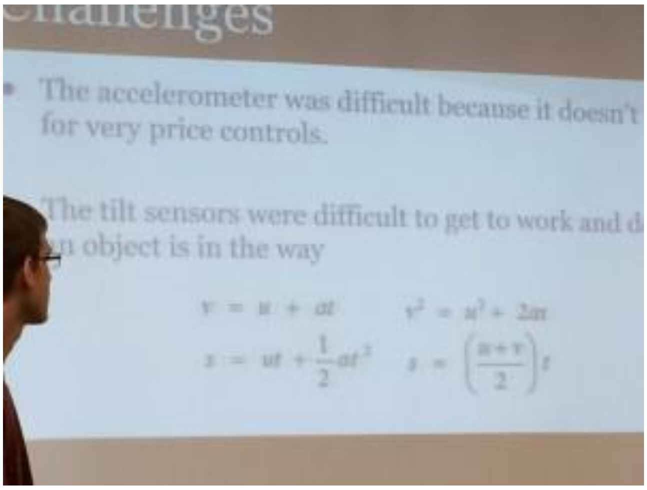
Dance Dance Revolution is based on the game of that name. In this version the players were initially required to tilt the board to select the correct LED as the song played. As with others, the motion sensor was abandoned. However, that is not the point. In order to code the game the team were required to encode the song. This requires identification of note and pitch. As the team said, “these were long lines of code”.

The projects were ambitious (not all mentioned here), technically demanding and innovative. The philosophy was very much about journey rather than arrival. To use one of the group’s words when commenting on success “kinda”. It was “kinda” right. In this environment “kinda” is success.

Engineering and Prototyping

Being able to take the daunting step of saying ‘no’ and either abandoning or reframing a solution figured highly in all of the projects. It was noticeable that none of the groups were shy about exposing problems and shortcomings. Few of the projects worked as the teams originally wanted but this was not taken as failure. All had developed significant hardware and software skills and all had realised that the process of getting the two to talk to each other is no mean feat. The skills acquired would put many undergraduates to shame.

The skills on show were highly engineering biased. Below is an image taken from the Hand Pong presentation.



In order to get the motion sensors to work required that the group apply high-level mathematics skills to the problem, dealing with motion, acceleration and position. See the above image for the type of mathematics capability required to progress.

If engineering is the ability to frame solutions that place and apply physical objects so that they have a degree of robustness then what I saw was Computer Engineering. Within this there was significant evidence of computer science but then again there was significant evidence of physics, mathematics and creativity.

Programming Environment

Much of the programing was undertaken using a beta version of codebender (codebender.cc). This is an online Arduino coding environment. Its premise is very much about sharing. Arduino sketches are accessible by the rest of the community. A great idea but suffering from bugs. Students report that code which runs in the Arduino environment away from the online environment does not always perform online. It is a beta, so bugs to remove but an excellent idea. Arduino being open source lends itself to Arduino programmers sharing their ideas. This is firmly within the strand of digital literacy in England but it jars against the reality of what students are required to do. Of necessity, Key Stage 4 GCSE students in England are required to undertake individual projects where the opportunity to share and work in groups is limited. This is an area that is limiting progress.

Key Messages

What can we learn from the students at Centaurus High School? Computing can be engaging, it can be creative and it can be a group activity. Computing has a place in an engineering syllabus.

Could we use the work at Centaurus as a model in England? It would be relatively straightforward to develop a curriculum. Even better, Sparkfun and the University of Colorado has one ready[18].

- **What would it require?**
  The students I saw are the equivalent of our GCSE age. They received credit for their work. In England the GCSE system is skewed towards individual working: 90 minute examination, 2 individual projects is not uncommon – the new GCSE syllabi increase the amount of examined outcome and reduce the project assessment to typically an 80:20 ration. This raises fundamental education questions about how pupils learn and how best to assess them. If an English curriculum is to be "*ambitious, stretching and exciting - drawn up by industry experts, allowing teachers and schools more freedom, designed to equip every child with the computing skills they need to succeed in the 21st century*" the new programmes of study at Key Stage 4 do not support this.

- **Modelling industry is fundamental**
  Working in computing in industry means working in a group with accountability and sharing key features. Individual success in examined situations is important but it does not fully mirror commerce and industry. Any software project will typically have many teams working towards a common goal. Budding computer scientists benefit from group working. Industry demands it.

- **Individual accountability**
  We could still leave the terminal 90 minute examination in place and relax the bonds on the project working. Of course, there are issues to resolve with team working. How do we allocate grades amongst team members especially if the teams are unequal? In my experience, team members are pretty ruthless when allocating grades based on contribution. In a programming project there is more to it than simply coding. A good coder is not automatically a great planner or tester. An individual can achieve success with his/her programming and lose marks based on the other aspects. If we want an examination system where we develop collaborative skills alongside technical competencies individual projects will not achieve that. However, if we want a system where we reward, group working is a viable approach.

- **Changes to assessment**
  At the moment GCSE syllabi have highly prescribed assessment scenarios. All students have to undertake the same task. Whilst making the assessment process relatively straightforward it does little to encourage creativity and may constrain ability with a subsequent impact on outcomes. How many computer scientists does it take to create a solution? More than one is the answer. Teams, when well managed and sympathetically constructed, are able to tackle much more complex problems than individuals. If I want to build a bridge it is going to take me significantly more time by myself than if I work with a team.

- **Balancing hardware and software knowledge**
  An aspect that is not developed anywhere near enough, and a potential source of tension, in England is development of hardware knowledge. There is a small focus on this in the examined element of GCSE programmes. However, students spend a large amount of time on development of software solutions. These are screen-based. There is a danger that

programming becomes a theoretical activity – okay for some but not for all. A glimpse into hardware knowledge can be seen in the Raspberry Pi scenarios released by some of the GCSE exam boards. Is this actually hardware focused though. It is really programming in a different operating system. Much can be achieved in a virtual environment.

- **Communication Skills**

  The students at Centaurus High School developed an in-depth knowledge of hardware and the tensions involved in getting the communications right. Putting aside the end solution aspect, what of the soft-skills arena in group working? In group working we need to communicate, we need to develop an awareness of others and be able to agree. As life skills these rank pretty high on the scale of needs to be met. Aside from these skills making us better citizens they also put a premium on employability.

- **Subject location**

  Trying to find computing as a separate subject on the Centaurus[19] site is not an easy task. At this school the subject falls under engineering with a heavy emphasis on robotics. Where you place a subject has an influence on how it is viewed. Students understanding that the purpose of software is to produce hardware outputs places computing well within the remit of engineering.

Transfer to England

- Would build problem solving skills
- Models industry working
- Places emphasis on engineering approach
- Encourages multi-partner working

Partnership Working

Innovation Centre Longmont Colorado: STEM and Multi-Agency Working in Action

Context

Partnership working is popular. The Innovation Centre at Longmont, Colorado[20] is an area of CS expertise with a slight difference. Students from Skyline School, Longmont have an opportunity to acquire a highly technical education. Funded by a Federal grant 'Race to the Top'[21], students can elect to study at the Innovation Centre. The article here[22] gives a flavour of what the Centre does. Skyline School is not a high performing school – hence the award of the Race to the Top grant.

Currently, the Innovation Centre can only accept students from Skyline High School. This is a product of the grant. When the grant completes there is opportunity to extend places to other High Schools in the Longmont District.

The curriculum, devised by Skyline, University of Colorado and the Innovation Centre offers a range of opportunities. It is part of the STEM initiative that figures largely. To get into the programme is no easy feat. Places are competitive and students must show an aptitude and willingness to study CS topics to a high level. This is even more so as the Centre has a range of commercial and public customers – real customers who require real solutions. Contracts undertaken are not simply classroom-based scenarios. During my visit students were testing and preparing a large number of laptops for district schools and a team of three were working on an App for a local company. This is not uncommon. One student took the time to explain a product they were making for the Sphero company. He and his team had prototyped the product using the Centre's 3D printing facilities. He was at pains to explain the engineering issues they had to understand and then overcome.

Qualification Route

Students undertake a formal course called 'Tech 1'. The focus of this is Apple products, their diagnosis and repair. A successful graduate of this course takes the Mac OS X Software and Hardware certification examinations. These are industry standard. This then opens vistas for students that would ordinarily be denied them. They may choose to move to working directly in the CS industry with Apple certification or move to undergraduate study. Notably, whilst success at the Innovation Centre does not guarantee acceptance at the University of Colorado it does improve the likelihood of acceptance. This is due in part to the close collaboration between the three partners. Any graduate of the Innovation Centre accepted by the University of Colorado will have already covered significant elements of the first year degree programme thus reducing the time they need to study on specific aspects. This provides opportunity for focused study on other areas. Tech 1 is not the only programme the Centre provides. Senior Engineering Design is available to students. The engineering in this instance is closely allied to CS.

Sustainability

The Innovation Centre is a good example of a sustainable project. The overall aim is to raise attainment of pupils in one particular school. This it appears to be doing. Through establishing commercial links with the local education district and commercial organisations the Centre stands a high chance of becoming self-financing. This provides sustainability, reduces reliance on central funding and opens up opportunities for students.

Transfer to England

- Lends itself to development in UTCs (University Technical College)
- Potential for self-funding programmes
- Targets the least accessible socio-economic groups
- Offers a route into Higher-Level Apprenticeships

Case Study SparkFun

Brian Huang – Education Engineer

Who are SparkFun?

The company background can be found here:
https://www.sparkfun.com. They are a company with their heart in CS education. It is unashamedly CS from an engineering approach, be this as evidenced by the 8 eye robot or the AVC (autonomous vehicle competition):

Left – Product Development Right – 2015 Autonomous Vehicle Competition

Much of the output is Arduino-based. SparkFun produce a range of components based around the Arduino platform. They have a dedicated group of engineers and equipment on site developing components. The company works with schools throughout the USA at a number of levels. At its most simplest, they act as shippers of CS Engineering components to schools on a large scale. On a more complex level they help to develop curricula with partner schools, other institutions and school districts. For example, the Centaurus project is example of such collaboration between Sparkfun, University of Colorado and Centaurus High School. On a much larger scale, the company works closely with New York City Education District in establishing the move towards computer science capability. Such activity signals close cooperation between the partners to create curricula, analyse results and reformat ideas for wider use.

What can SparkFun Offer Education?

Onsite, the company has a dedicated training facility used by school pupils and for dedicated training of staff. Training is carried out by a team of educators. This is not just lip service being paid to educational support. My opportunity to meet with Brian Huang was limited by his availability at the HQ as he is often on the road providing educational support.

Education facility

The facility and commitment to work with others in developing understanding is supported on-site. For approximately £350 teachers are provided with a 5 day training programme that includes £200 of equipment. Five days training at a net cost of £150.00 looks reasonable. A clue to the direction in which the company are looking to support education can be seen early on in day 1 of the programme. Slide 4[23] focuses on the 'Maker Movement' and what it means.

The focus is undeniably engineering, taking in electronics, hardware and software. By Day 2 the starting objective is "implement a Maker-centric classroom built around project-based-learning ". The session then develops into working with the building blocks for this to result in the creation of lesson resources to enable the objective to be met.

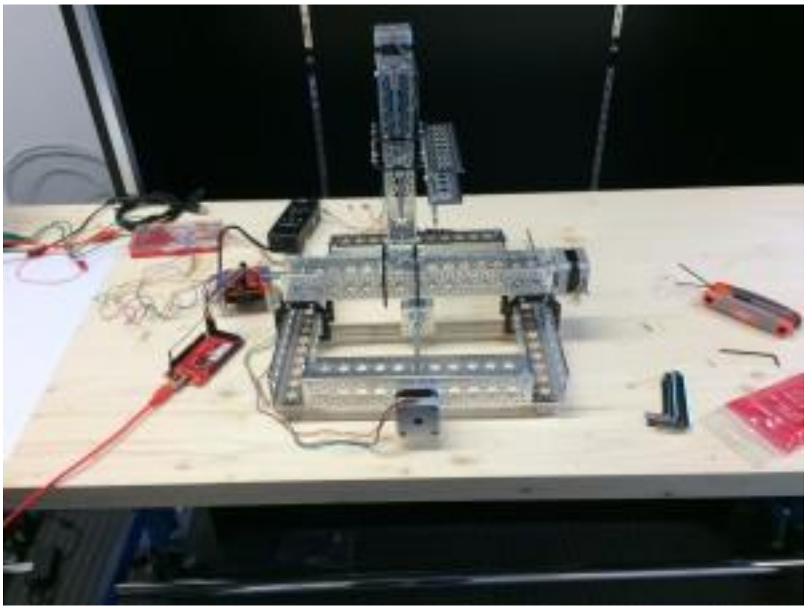
Creativity

Sparkfun is a place of work in progress. New ideas are prototyped on a daily basis. The creators, or 'makers' as they are called in the USA, have your archetypal engineers desk with soldering irons, electrical components and general 'let's see if this will work' furniture. Throughout the building you can see examples of past ideas. Some of which are writ large. For some time I have been watching the development of electronic wearables and conductive ink.

Conductive Ink Project: 12 feet x 2 feet

Each part of the above image is created with conductive ink. This is linked to an MP3 shield. The whole lot is programmed to play a range of sound files. Immediate ideas for art and music come to mind pointing to the cross walking approach. This takes us out of 'pure' CS and into the creative realm (an area referred to as STEAM in the USA). How much could we achieve in a cross-curricula project? Create interactive books, works of art that respond to touch, a work of art that is also a musical piece? The hardware can be repurposed being PCBs and associated electronic equipment. The ongoing outlay here being the conductive ink.

So, what is in development? Taking open source to the edges, SparkFun are prototyping an open source 3D printer:



Prototype 3D Printer

The intention behind this is that the whole kit acts as a complete project from the building to the programming to the printing of objects. The engineering involved in this for any student with the kit will be enormous – consider the problem solving required to program an object with 3 dimensions.

I missed the annual AVC (Autonomous Vehicle Competition). What an educational trip that could be for English students? Prepare your robot during the year, fly to Colorado and compete.

Transfer to England

- Arduino focus combines hardware and software working
- Provides a second text-based language to meet Key Stage 3 requirements
- Embeds links between industry and education
- Further encourages the 'maker community'
- Low-cost CPD: A typical 5 day teacher training programme onsite costs in the region of £350

High School Computer Science: Overland High School

Context

Kate McDonnell, as part of the engineering faculty, leads the computer science curriculum at Overland High School. This appears to be a recurring theme – CS placed within an engineering remit. This allows Kate to run extra curricula programmes in robotics. However, curriculum placement provides a general focus on creating for purpose aka engineering.

The school is relatively unique in that it offers AP computer science. Kate runs 2 AP CS classes. Perhaps more unique is the male to female ratio. Around a quarter of the AP students are female. This runs counter to all the research undertaken into the gender imbalance in computing whether this be at a school or professional level.

Curriculum

At Overland a four year progressive programme is offered as follows:

9th Grade (14 -15 YO) Introduction to game design and introduction to Java and Python
10th Grade (15 – 16 YO) AP Computer Science
11th Grade (16-17 YO) Mobile App Development, algorithms and data structures
12th Grade (17-18 YO) Visual Basic

Aside from 10th Grade, where the examination is externally created and marked, the other 3 years are assessed internally and accepted on the teacher's professional assessment. There are, of course, checks and balances in place.

Relevance to Further Study

There is some contention between AP scores and recognition by universities. As previously discussed, the AP programmes are given a terminal grade. A university may require a particular level eg Grade 5 is the highest level and may be required by a 'top level' university in much the same way that English universities require A* and A grade for entry to some university degrees.

It is by no means certain that a University will accept an AP grade, even a top flight Grade 5. This is an important point in that it has two serious effects. By not recognising the Grade, it will entail that the student has to take introductory CS courses at university that may be, and often are, a duplicate of High School AP learning. The possibility of not being recognised puts added pressure on recruitment to AP programmes. Just as in England, the US curriculum is crowded and it is often a case of 'what can I drop' that determines a course of study. Place myself in the mind of a 10th Grader "If I do not elect for AP CS I can still cover the content in my first undergraduate year. I will also have more leisure time or more time for other subjects eg AP mathematics and I'll save $90 on exam fees. No brainer".

Current Qualification Route

AP CS is currently Java based. There are two terminal examinations each worth 50%. First examination is multiple choice. Second examination is free writing and focuses on being able to interpret and write Java code. Aside from questions such as "is the best test of a programmer a closed examination room?" the focus on Java is very narrow. If a student doesn't gel with Java they are in for a year of pain. This is especially so if it is their initial foray into programming. Python, with its 'low notational overhead' (Jon Zelle) provides a more accessible entry into programming.

It has been recognised that AP CS is possible in need of a change. From the 2016 academic year a further course of study will be added to the AP CS stable as discussed earlier.

Comparing England and The USA

As the resurgence in interest in CS grows in England we may see an increase in the number of degrees programmes requiring A Level computing. At present, very few do. Indeed, to take an example, the University of Kent state that "no previous programming experience is required" for their Stage 1 programming course. This is a reflection that the vast majority of schools do not yet provide CS or programming. In the case of Kent, the language first visited upon undergraduates is Java. Schools, where programming is delivered, may not focus much, if at all on Java. With Python being the default language for most schools delivering components there would be a need at Kent, for example, to provide foundation level courses. Of course, it is more likely that pupils in England will be accepted onto degree programmes in CS if that have taken CS at GCSE and even less likely, A Level. Perhaps this is an area where the USA and England are diverging? In making CS compulsory we will produce more CS ready undergraduates – that's assuming we can agree on a curriculum and recruit enough teachers at KS4 and KS5.

Transfer to England

- Overland’s curriculum provides a model for Key Stage 4 and 5
- American experiences of the AP system are seeing a swing away from a predominantly examination focused system
- Reinforces the idea that computer science should be characterized as a branch of engineering

Being A Good Computer Scientist: Apps For Good In Massachusetts

Context – CS and Social Good

With the focus put on the computer science and narrower programming content of the curriculum it is easy to overlook the requirement that we "ensure all pupils are responsible, competent, confident and creative users of information and communication technology"[6]. An approach to integrate the concept in a computer science perspective is to turn it over to the students.

At Malden High School, Massachusetts, a project was undertaken under the banner "Apps for Good". The project was very much free-flowing with a fairly broad remit. A group of sophomore students worked on creating programmed solutions that solve a social need. How that good is interpreted very much depends on what the students interpret this as. On the whole, the projects were very personalised and reflected a problem in need of a solution tied to their home lives. The students, mostly, worked in groups. So, we had devices that:

- Dispense pills
- Clean
- Monitor plant growth in a mini-greenhouse
- Irrigate plants
- Record tabletop football scores
- Control LEDs on a mask

A range of equipment and methods were in evidence. Students were encouraged to use at hand materials: tubing, pieces of wood, Lego parts, Perspex, expanded polystyrene and so on. Software was not dictated. Students used EV3 Lego code, App Inventor, Arduino to name but three platforms.

Managing Open-ended Programmes

That some of the interpretation is student defined marks a tension in providing open-ended approaches to student projects. Too much direction and you run the risk of turning students away and demotivating. This is a point the English examination boards could take on board when creating their project/controlled assessment briefs. Having loosely defined criteria makes the process of moderation more difficult. Difficult in that the skill set required by moderators needs to expand greatly. Difficult in that initial agreement of projects requires greater control to ensure the evidence will be robust, sufficiently complex and generally worthy of study at that the level which students are entered. We currently do this at A Level. Aside from guidelines that present a container approach to what the evidence should contain in terms of structure, the projects themselves are very much a case of what the student wants to do.

Were the social good projects at Malden as intended? Yes, ignoring the tabletop football and LED mask, the students had a focus that enabled them to explore how they might solve a problem near to them emotionally. Having a high proportion of minority students provided an opportunity to observe evidence. One of the two pill dispenser projects provides an exemplar case of non-verbal communication. Non-verbal in that the group were communicating in Creole, not a dialect I am any where near familiar with. The participants were handling the parts, pointing, assembling and disassembling to prototype their ideas. Watching without sound forces you to look at the non-verbal.

Student Cohort

Who are these students and why did they work on the project? The students take an elective. Elective means not compulsory and means it will not affect grades. Elective means a chance is provided to explore ideas and not be held penalised if the experiment goes wrong .

The students were largely 'low-ability'. This is important. In the USA, as with the UK, there is a tendency for a sharp intake of breath when deciding on which students take CS. Offering a 'difficult' elective to a low-ability group is not going to provide the most favourable conditions for success. Planning, monitoring and reviewing are constants. Were the students working in groups? Yes, all but one. Were they working towards a solution? Yes they were. The ideas were not fully formed and did not work quite as required ('required' here being defined by the students not the instructor) but they were developing solutions. Judging when it is okay to let students struggle and when to end this is ever the difficult part of education. Much of CS learning is about being prepared to accept that something is not working as initially required. As the student from Overland High School observed, "it kinda worked". The trick, if there is one, is being prepared to accept this and then move on to another approach. This is often where an educator needs to step in. This is not about whether the students are high or low functioning. It is about the teacher doing their part by being aware of where the student body is on a macro and micro level and applying this knowledge to drive learning forward. It is not about giving the solution it is about enabling the environment so that students arrive at the solution – otherwise known as scaffolding.

Qualifications Vs Skills

In England we have a mandatory curriculum. At Key Stage 4 there is a requirement to provide the opportunity to follow a programme of study. It's about numbers, if a student does not follow a GCSE, or GCSE level, programme, then it doesn't count. The cull in qualifications during the past 18 months was necessary and long overdue. We have an opportunity, at least in CS, to explore alternatives and give ideas a go. Ideas such as senior students working in groups to develop solutions to problems close to their experiences. Putting an assessment rubric on it is the easy part. Soft-skills are a vital aspect of being an economically viable citizen. The project I witnessed is supported by a Federal grant, developed in conjunction with a local university and held in a school identified as one that would benefit from such intervention. It is intervention and 'intervention' is no bad thing. If your purpose is to raise aspiration and thereby increase life chances intervention is a must. We have enough universities in England; we certainly have central government funding that can be developed to small-scale projects.

Transfer to England

- Key Stage 4 education has settled into a qualification focused economy controlled by external bankers in the form of examination boards. This leaves little room for alternative provision
- This programme avoids the niche approach to CS, where only the most able are provided the opportunity that will "allow them to progress to higher levels of study"[6]
- The SMSC (Spiritual, Social, Moral, Cultural) requirements of the English system are observably met and instantly assessable in this US developed approach

Labelling – Changing the Name, Changing the Focus, Improving Attainment

Computer Science Vs Software Engineering New York

New York City Context

New York City Department of Education is supporting an innovative computer science programme and branding it software engineering. A thread that returns is the use of 'engineering' when computer science is highlighted.

Amongst others, I have visited the Academy for Software Engineering (AfSE) and the Bronx Academy for Software Engineering (BASE). There are a number of characteristics common to the schools. I'll start with the organisational practicalities. Both schools are based in large multi-school buildings that were once single schools housing in the region of 2,000 students. Both are now mini-schools. Unlike English academies, where you may have several schools/communities headed by an assistant principal the idea is taken to the extreme here, each school is a separate entity staffed by its own teachers and headed by separate principals. This is a deliberate move away from large, difficult to monitor schools and into a more specialised focus with smaller student numbers, for example AfSE has around 500 students on roll across grades 9-11 – it started as a school in 2012. Next year this will extend to grades 9-12 with an increase in students. BASE has a smaller number, having been in existence for 2 years. Both schools occupy single floors of multi-storey buildings and both intend to add extra floors next year as they expand and other schools in their buildings close. Think Academy schools in England and you have a similar approach, or at least end, the new, smaller, highly specialised schools replace under-performing or no longer relevant schools.

Whilst having a software/CS/engineering approach the schools also provide a full curriculum covering the expected other subjects. The difference here is that CS is not an elective. Pupils enrol in the schools with the understanding that they will be studying a focused, computer science set of topics. Something must be working, both schools in widely geographically different areas are over-subscribed. Not all who apply to BASE or AfSEN are enrolled.

Accountability

How does this work? The schools are monitored and report to the New York City Department of Education[24]. They retain the ability to close underperforming schools as witnessed by the outgoing tenants in the two buildings I visited. There is large support for CS in New York. The city has its own employees for this. However, there is a major NGO involved in ensuring that CS is embedded. This organisation is CSNYC[25]. The team at CSNYC are supported through donations, some very large. Whilst New York City provides the buildings, infrastructure, staff and salaries to run the schools CSNYC provide what keeps the system going: professional development, advice, organisation and support. The extent of this can be seen on their site. Leigh Ann DeLeyser (Computer Science Programme Manager), is about as committed as you get. We met at both AfSE and BASE. As with many school districts, New York works closely with universities to develop curricula. In this area both Brown and the University of Berkeley have teamed with CSNYC and New York City to help embed and plan for the future.

Teacher Recruitment

Where are the teachers coming from to support this push in New York City? New York does not have a licence system for CS teachers. Let's take a lesson I was invited into at AfSE. Sean Stern is a recruit from industry. The lesson had an illuminating focus. Students were investigating data. More specifically, they were looking at how the same set of test data was interpreted by three different people. Not a computer was used. Perhaps the lesson can best be placed under the heading of 'fuzzy logic'. Students were considering the qualitative nature of data. All seemed to be perfectly capable of undertaking quantitative use of data but that was not really the point. In working in this way (from individual to small to large group working) students were getting a real lesson in not only how to judge what data to use but also how to make judgments on what data is relevant and why. So, the implications for them? This lesson feeds directly into a simulation that pupils are working on using Greenfoot. In this CS aspect they will be working with sets of data and making decisions on tests, number of simulations and significant outcomes. This looks suspiciously mathematical? Equally, you could include 'science' as well. The lesson exemplified scientific method with regard to testing. Towards the end of the lesson the pupils were asked to agree a set of criteria that should be used to assess the method and data used in a simulated experiment. The added aspect to this is that the students were later assessed by the teacher using the set of criteria developed by the pupils. Further details of the curriculum on offer to AfSE students can be found here[26]. A look through the curriculum ideas will provide an idea of the breadth and depth of what is taught.

Computer Science in the Bronx

At BASE I spent the day with Jon Mannion. In typical CS teacher format his specialism is history, or more specifically bi-lingual history. Jon has been in post for this academic year and is as committed to getting a CS curriculum in place in an what would easily qualify as an area where CS professionals are under-represented. As with AfSE, the pupils elect to join the school. Gender mix is low with 10% of the pupils being female.

Skills and Application

In Jon's classes pupils were using a range of tools to develop data handling skills in Python. Two tools being used are Codesters[27] an online programming environment (still in development but looking promising) and Tuvalabs[28] an excellent resource for data handling. The range of data sets is extensive to say the least. This resource has uses way beyond CS but was being used to support the Python coding work: grab the data from Tuvalabs and manipulate it using the statistical tools in Codesters using Python functions. Easy to spot mathematics, statistics, history, PSHE and CS in this. A fairly comprehensive CS lesson.

The Wider Picture

Schools have the ability to develop their own curricula, as they are doing. New York City and CSNYC are there to assist with this, and they do. Although not mandatory, New York City are encouraging schools to take part in the Software Engineering Project. Currently, there are around 3,000 pupils involved in this. A major objective of the programme is to target under-represented groups in New York. This is an explicit attempt to move CS away from its typical niche, 'geek' market.

What is the Software Engineering Project?

SEP is intended to support students from grades 6-12. As students work through the programme they will be preparing for the new AP CS principles terminal examinations – not a requirement but it certainly moves them in the correct direction. The AP CS programme will have its first examination in 2017. The curriculum is fairly broad, especially when compared to the existing AP CS curriculum. The curriculum headings are as follows:

- Computer programming
- Robotics
- Web design and development
- Physical computing
- Mobile computing

The curriculum is much wider than the headings indicate. Students undertake long-term work experience (typically 6 weeks in which students are paid above a living wage) related to CS. There is a range of electives (options in England) that students can undertake. The curriculum offers a degree of freedom. Viewing the SEP[29] site will provide you with an idea of the range from 'standard' HTML through to mathematics and cryptography.

SEP is based on partnership working involving Federal Government, New York City Department of Education, large corporations such as Verizon and AT and T through to individual speakers.

Content and Logistics

The Pilot has an initial 3 year life-span from 2014-2017. This is significant. The third year of the Pilot marks the first examined year for the new AP CS examination series. Those students entered for this will have received a dedicated CS curriculum, based on the BJC format, for up to three years. During this time they will have received instruction in the curriculum areas and have produced the two artefacts (read portfolio for the English system).

Although not a mandatory part of the curriculum (I can find very little content that is mandatory) the BJC (Beauty and Joy of Computing: BJC[30]) course of study from Berkeley University of California is offered to SEP schools. This undergraduate programme has been modified for use by New York schools[31]. Curriclula can be found here.

What of the logistics and practicalities? All involved are provided with ongoing professional development. This is supported in two major ways: time given to teachers to prepare and gain curriculum knowledge throughout and payment for the time taken to develop their curriculum knowledge. In return, schools are recruiting the under-represented into CS. The Pilot is aligned to a core curriculum with electives to support areas of learning that provide a context: social, moral and ethical.

Learning From New York City

District and Higher Education Partnerships

There is a long-term commitment from all parties. Direct involvement in curriculum design and delivery is undertaken by universities. This theme is not uncommon. New York City is working with Berkeley and Brown University. Similarly, the University of Colorado supports the Boulder area schools in curriculum development. Does it make sense to involve universities in school curriculum? I believe it does. In partnership there is the opportunity to draw on expertise from all involved. This is not a top-down, pre-packaged route. The partnership working requires all agree on content, direction and pedagogy. Such partnership working allows for addressing of local needs with an emphasis on 'home growing' talent. It makes sense for universities – the students are the potential undergraduates of the future. It makes sense for the local districts who want to retain talent. It makes sense for schools where curricula are in need of change to reflect the needs of pupils.

Transfer to England

- Do we have such support in England? The answer to this is one of a mixed message. Financial support is there eg the £2,000 primer funding from CAS, via Central Government, to support schools with Master Teachers in supporting their local areas.
- The SEP programme addresses, more or less, the equivalent of Key Stages 4 and 5. This was also seen in the Colorado programmes.
- Master Teachers do an incredible amount of support work – register with the CAS site to receive notifications
- Is CAS involvement enough? There is concern about the recruitment of teachers into CS teaching
- PGCE and School Direct programmes exist but the numbers do not add up. Two universities based in the South East of England graduated 18 newly qualified teachers. Between them they serve in excess of 200 secondary schools
- This doesn't touch on the development of CS in primary schools
- Given the extent of the curriculum maybe it is time to take a more nationally driven programme supported by the teacher training and CS departments in universities have
- The QuickStart[32] initiative is a way forward.

- What of County support? For a range of reasons, political, financial, philosophical the education support arms of most counties in England disappeared at an alarming rate in the 2010-2012 period
- Now we have a largely new National Curriculum is it time for countywide support to be put in place?
- CAS hubs work well on a micro level helping to support schools within a relatively small area. Think CSNYC in support of the district. New York City Department of Education is supporting an innovative computer science/software engineering programme

Non-Specialist Computer Science: Stuyvesant High School

Context

Mike Zamansky heads the CS team at Stuyvesant High School. A little about the school, situated in the Financial District of New York City it is a short walk from the 9/11 Memorial Area. Being NYC, the students at the school are not drawn from the near geographical area but from the 5 boroughs. Stuyvesant is a large school: 11 stories and 3,300 pupils. That provides around 825 students per grade group (grades 9-12). That's a large school. This is in stark contrast to the other two other high schools with populations of 500.

Stuyvesant is seen as the best school performing school in New York City. It is a district controlled, state-funded school, and outperforms the best private schools. Indeed, the school is seen as one of the best in the country. There are various indices used in schools, just as in England. This school has a population where 34% are eligible for free lunch. That figure in England would attract a fair amount of interest and considerable PPG funding (Pupil Premium – an amount allocated for each student on a deprivation index).

Excellence and Economic Disadvantage

A high performing High School with a high proportion of economically disadvantaged pupils? Stuyvesant has a good, entrenched CS curriculum. Unlike other schools, Mike's is not a specialist school so CS has a fight on its hands to begin with. New York does not have a mandate for CS. At Stuyvesant there is a requirement for all pupils to undertake a Grade 10 introduction to CS class for a semester. This remains the only mandatory programme. That being said, the Grade 11 AP course, Java based, is over-subscribed. Around a third of Grade 11 students get to take the programme. This is a disappointment for some students as not all applicants get the opportunity. Needless to say, the majority of the students taking AP achieve Grade 5.

Curriculum – Pupils in Action

Beyond Grade 11 and AP, students take CS. This is where the real learning and progress occurs. Working in groups, not always the same groups and manipulated so that all students work with every peer in their group, students complete spectacular projects incorporating several tools. Here are just three:

> **Exercise and fitness project**: using a web front-end a user can input their regime, diet and general activity. The system will report back progress and recommendations

Summary in a flash (my name): frustrated by assignments which contained references to multi-page documents, a group of students have created a system which will read the documents, feedback the relevant information (the depth of this can be chosen) along with key words. The group could see the relevance of this and potential for them and others when they move to degree level study

3D Graphics: this is a class where all students develop ability with 3D graphics. Students create front ends that will allow objects, eg a torus, to be coloured. This is not easy, the mathematics involved is complex and the problem solving is highly developed.

Value of Group Working

In any of the group projects seen it would be highly unlikely that individuals could complete them by themselves. This is based on two factors, time provided (typically 1-2 weeks) and range of tasks involved. The two factors are significant and something we could learn from in England. Group working is a good idea; it provides a venue for cooperative skills, develops excellent employment 'soft-skills' and allows for the completion of highly complex projects. Simply put, it mirrors the CS commercial environment.

The work undertaken by students requires high levels of commitment, which may be being missed by specialist schools – more of that below. The commitment extends to private study working. CS is a natural subject for this. Students work together out of school but may, and often are, separated by many miles. Online tools such as Koding.com, Google sites and GitHub repositories are a natural venue for these complex CS projects. koding.com is an environment that allows for the incorporation of a range of tools to achieve a group purpose.

It is refreshing to see that students were proficient in languages other than Java. However, by far the greatest use of programming language used was Python. When asked about preference many students indicated that this really depended on what you wanted to achieve, a significant indicator that the students were judicious when choosing tools.

Driving Force

Mike Zamansky is credited with being the initial driving force behind New York City's push for a computer science curriculum. An article on this can be found here[33]. Mike and his team are about as passionate as you could get regarding their subject and idea of what education is. Mike heads the CSTUY[34] group and runs a highly informative blog[35]. The graduates from his classes inevitably follow a degree route and either work for high-end CS companies or start their own. Mike's teaching room is decorated with sweatshirts donated by alumni who have taken CS and gone forward to work in the industry. This drive starts early; many of the Y12 projects are put forward as commercial products with angel investor funding being found to take the ideas further.

Comparing Schools

So, what are the differences between the CSNYC supported schools, BASE and AfSE, and Stuyvesant. Entry to Stuyvesant is open to all but does require a selection test. As the most prestigious school in New York they are oversubscribed. This process, ignoring philosophical/socio-economic driven objections, finds the most able students. A culture of success and aspiration is already born. The CSNYC schools are non-selective. They are over-subscribed. Selection is based on a lottery system, much like non-selective schools in England. A question could be asked as to whether not having an inquisitive selection process sets up issues for the schools. Arguably, if you are going to specialise then it might be advisable to check that those opting having suitable aptitudes. This not only provides some future proofing for results but, perhaps more importantly, generates a means of avoiding disappointment for some pupils. It is one thing to claim a preference but quite another to be able to meet that preference.

Transfer to England

- The clear message is that for excellence to emerge partnership working is required
- Perhaps more difficult, the sustainability of excellence requires the commitment of talented teachers and support systems
- We have talent in England and it has a quasi management structure in the form of the new Regional CAS structure
- Economic background need not be a bar to excellence but it may require travel to focused schools

MIT, Scratch and Beyond – Addressing the National Curriculum at Key Stage 3 and Key Stage 4

Massachusetts Institute of Technology

Towards the end of my investigation I was lucky enough to spend time with Josh Sheldon and Cynthia Solomon of MIT. The American education system at all stages from primary through to postgraduate are intrigued by the situation in England. As they struggle to get CS recognized as a subject nationwide the English step to mandate for all is applauded.

Influence on English Teaching of Programming

MIT are at the forefront of software development for education. They are the originators of Scratch and latterly App Inventor. Both of these being blocks based programming platforms. App Inventor is in its second iteration and has raised great interest at MIT as a result of the uptake in its use. England has the largest use of App Inventor. It is no coincidence that the number of registered users in England has steadily increased with the inclusion of App Inventor tasks by two examination boards as part of the controlled assessment offering and the increase in numbers of pupils registering for GCSEs in CS.

The uptake in use has been accompanied by a steady increase in the number of requests for help and guidance to the MIT team. The requests raise some degree of concern as this points to staff not being fully prepared to deliver, monitor and ultimately assess the pupil evidence.
Knowing that the use of App Inventor is growing has given rise to questions about the impact that the application may have on driving student capability forward. This provides an opportunity for English schools to become actively involved in driving the application development forward. What is needed is an agreed set of standards that can be used to monitor progress in this area.

Direct Influene on Key Stage 3

Taking a step back, App Inventor has links to Scratch. Both have been developed by MIT and both use a blocks or visual programming language approach. This lies at the heart of the Key Stage 3 curriculum where there is a requirement that pupils "use two or more programming languages, at least one of which is textual". This is at the least an acknowledgement that pupils are using blocks programming approaches. The inclusion of App Inventor based tasks by the examination boards points to value beyond Key Stage 3.

Aware that programming as part of a computer science curriculum is as much about judicious selection of tools, MIT teams are working on other blocks based language scenarios. Two witnessed are pencilcode[37] and gameblox[38]. These address particular programming scenarios and therefore allow for a more focused tool selection when problem solving. Pencilcode provides the opportunity to convert the blocks into web-based text languages such as CSS, JavaScript and HTML. This looks like a useful bridge in approaching the requirement for capability in at least one text language at Key Stage 3. Gameblox takes a more games-based approach to programming. This has the added benefit of being supported through an edX course.

Transfer to England

- Whilst widely used, the impact blocks-based programming environments is poorly monitored
- As the CS curriculum established itself it will be time to consider what value our approaches have and how we will measure this
- Closer ties to MIT and other organisations developing curriculum resources will help to establish a rigorous programme of research and development

Conclusions

Partnership working

- It is clear that great value is placed on partnership working in developing CS capability in the USA
- Partnership working requires political will in the form of financial and logistical support
- Recognition is given to students completing programmes
- Partnership working allows all involved to develop:
    - Schools benefit from the intellectual input brought by university departments through increased teacher capability and accelerated pupil attainment
    - University departments benefit through an enhanced understanding of pupils and pupil attainment
    - School districts benefit through enhancing the exit capabilities pupils acquire
- There is partnership working in England but this may be seen as in its infancy
    - We have CAS as the lead in networking people and organisations
    - How effective this is is as yet unclear given the infancy of the curriculum
- The lack of local authority involvement in part so England means that a crucial partner is missing
- Partnership working can help in reducing the effects of skilled teacher shortage

Educational Focus

- The English system has a narrow focus on programming skills
- The USA system tends towards an engineering approach where programming is one skill amongst many
- In the USA there is a move away from centering CS on programming
- Group working is actively encouraged using collaborative tools in the best exemplars in the USA
- The new AP model being created in the USA is a viable alternative to the English model

Building Sustainability

- The Key Stage 3 programming requirement requires attention to the depth of understanding and capability in teachers working with blocks programming languages in particular Scratch and App Inventor
- Crosswalking enables mapping of pupil attainment from CS into related areas in particular mathematics and science
- How particular technologies build pupil capability is a largely unknown and uninvestigated area.

Going Forward

Following my research in the USA a number of actions have been applied.

Partnership working

- Cross-authority working has been launched between two 6th Forms to enable sharing of expertise

  The Key Stage 5 students have already benefitted from use of the lead school online learning platform.

  This will be further extended through the use of online conferencing and inter-group working

- The University of Kent have expressed an interest in working to develop a Key Stage 3 curriculum model.

  First steps towards this will be undertaken in the 2015-2016 academic year.

- MIT have expressed an interest in developing the idea of partnership working through assessing the impact that App Inventor and related applications have on pupil attainment.

  The links are in place with a potential range of schools from which to draw the data.

  What is now required is support in collecting data, analysis and interpretation.

Addressing the Key Stage 3 Curriculum

- Computing at School have supported the creation of a local hub with a specific remit to focus on the use of blocks programming and the transition to text-based programming.

  The hub will cover the North Kent and South East London area and is open to anyone with an interest in developing this area of computing.

  The Hub launched in September 2015.

Addressing Professional Development and Teacher Recruitment

- Negotiations have been undertaken with SparkFun to explore how their CPD model may transfer to teachers in the English system.

  A model has been provided with costings. The next stage will be to explore sources of financial and logistical support to build an engineering focused core of teachers

- University of Greenwich trainees have devoted a portion of their training year to investigating the pedagogy of blocks based programming.

  This will assist in creating a cohort of trained teachers aware of the processes involved and tools available to ensure that core elements of the Key Stage 3 curriculum become embedded in Secondary schools in the South East.

- Key elements from my research, in particular a more engineering focused approach, to the curriculum has been applied to the trainees' programme.

  In particular, the curriculum includes open source hardware and software, robotics and group participation.

- Utilizing online approaches, a subject knowledge enhancement materials for Canterbury Christ Church University.

  This provides the initial steps for potential CS teachers drawn from a range of backgrounds but missing core CS knowledge.

Recommendations

Partnership working

- Partnership working should be given priority and include university, public and commercial organisations
- Results from partnership working should be reviewed and the results of the review process feed back into the support programme
- At the Key Stage 5 end of the scale, schools should consider skills sharing and inter-school collaboration to address the looming A Level teacher shortage

Educational Focus

- Commission specialist CPD to develop engineering based approaches using the SparkFun model to enable rapid building of core skills in local areas
- Move away from centering CS on programming
- Review GCSE and A Level approaches to the core research and group working ethos encountered in CS activities
- Consider the new AP model being created in the USA is a viable alternative to the English model

Building Sustainability

- Provision for specialist schools focusing on Computer Science should be encouraged with a further review of the success of the New York small school model
- Adopt a crosswalking model to enable mapping of pupil attainment from CS into related areas in particular mathematics and science
- Explore mathematics and computer science to develop joint PGCE programmes similar to the mathematics with physics programmes created to address the shortage of teachers in these areas
- Apply rigorous research to chart how and where success occurs and how this can be modeled across the country

References

1 Make or Break: The UK's Digital Future
http://www.publications.parliament.uk/pa/ld201415/ldselect/lddigital/111/111.pdf

2 Trainee teacher recruitment https://www.gov.uk/government/statistics/initial-teacher-training-trainee-number-census-2014-to-2015

3 Disruptive Innovation http://www.claytonchristensen.com/key-concepts/#

4 Michael Gove Bett 2012 https://www.gov.uk/government/speeches/michael-gove-speech-at-the-bett-show-2012

5 Bricklayers to Architects
http://www.csta.acm.org/Communications/sub/CSTAVoice_Files/csta_voice_05_2013.pdf

6 National Curriculum – Computing https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

7 Computing at School http://www.computingatschool.org.uk

8 Rosenbloom, P. S. (2013), On Computing The Fourth Great Scientific Domain, The MIT Press

9 USA School Districts https://www.census.gov/did/www/schooldistricts/

10 Denver School District https://en.wikipedia.org/wiki/Denver_Public_Schools

11 Colorado School Districts https://en.wikipedia.org/wiki/List_of_school_districts_in_Colorado

12 Common Core Standards http://www.corestandards.org/read-the-standards/

13 Arkansas Computer Science Mandate http://www.educationnews.org/technology/arkansass-hutchinson-signs-computer-science-bill/

14 Progress 8 Measures
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/456438/Progress_8_school_performance_measure_2015_updated_August_2015.pdf

15 Virtual Arkansas School http://computerscience.virtualarkansas.org

16 Advanced Principles Computer Science https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-curriculum-framework.pdf

17 Crosswalks
http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_Standards_Mapped_to_STEM_CC_P21_landscape_New.pdf

18 Centaurus group working curriculum
https://canvas.instructure.com/courses/876545/files#SparkFun%20Introduction%20to%20Engineering%20Design

19 Centaurus High School http://www.centaurushs.org

20 Innovation Centre Longmont Colorado http://innovation.svvsd.org

21 Race to the Top http://www2.ed.gov/programs/racetothetop/index.html

22 Innovation Centre Profile http://www.timescall.com/news/longmont-schools/ci_25006406/innovation-center-gives-skyline-high-students-new-space

23 SparkFun CPD https://www.schoology.com/course/299867764/materials/link/view/302230080

24 NYC Schools http://schools.nyc.gov/default.htm

25 CSYNC http://www.csnyc.org

26 AFSE https://sites.google.com/a/afsenyc.org/sy1415csprinciples/

27 Codesters as used at BASE http://www.codesters.org

28 Tuvalabs https://tuvalabs.com

29 Software Engineering Project New York http://sepnyc.org/2014/09/incoming-9th-graders-tinker-with-art-tech-and-the-fundamentals-of-computational-thinking/

30 Beauty and Joy of Computing http://bjc.berkeley.edu

31 High School version of the BJC syllabus http://bjc.edc.org/bjc-r/course/bjc4nyc_2015-2016.html

32 QuickStart http://www.quickstartcomputing.org
33 Mike Zamansky http://www.nytimes.com/2013/03/31/nyregion/software-engineering-school-was-teachers-idea-but-its-been-done-citys-way.html?_r=3
34 CSTUY https://cstuy.org
35 Mike Zamansky blog http://cestlaz.github.io
36 Daniel Hernandez https://sites.google.com/a/svvsd.org/sandboxhernandez/vex-robotics
37 Pencilcode https://pencilcode.net
38 Gameblox https://gameblox.org