Teaching computing on civil engineering degrees

Alan Bloodworth

University of Warwick

Synopsis

- This talk is about *computer programming*
- Personal reflection
- Experience in teaching computer programming at UG
- What is happening in schools
- Implications for the future

Personal reflection

ZX Spectrum

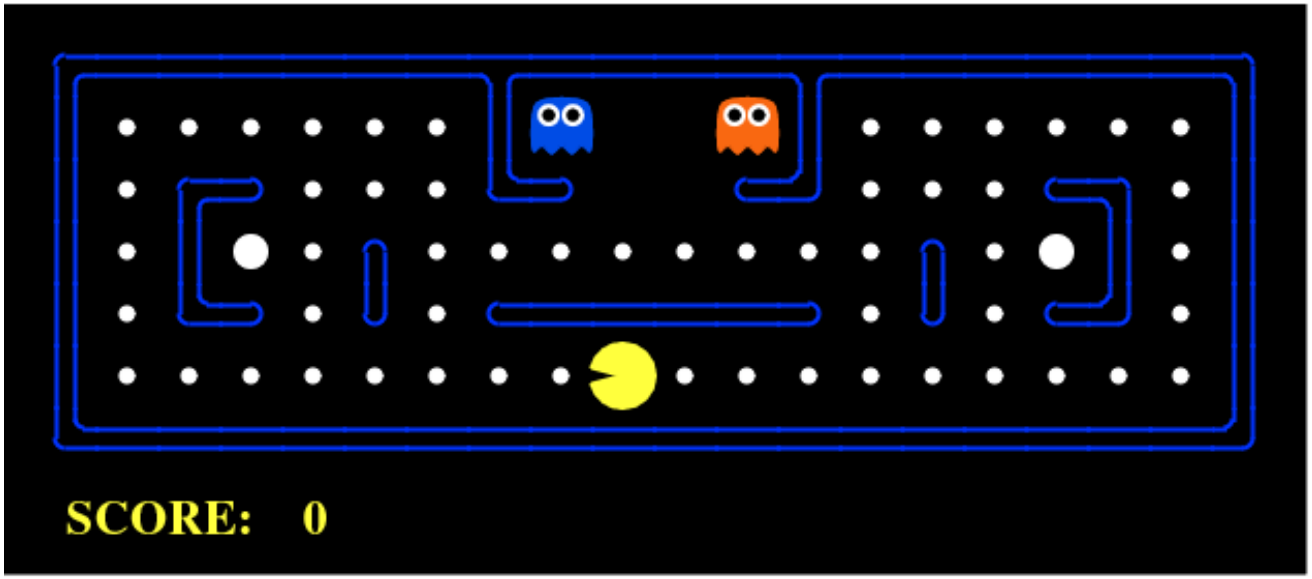
Texas TI99/4A

Amstrad CPC 6128

Personal reflection

- INPUT(…)
- LET…
- DATA(….)
- IF…..THEN……ELSE
- FOR……TO…….NEXT
- WHILE…….WEND
- DEF FN
- ON……..GOSUB
- PRINT(…)

Personal reflection



Berkeley.edu

Tools learnt from PacMan

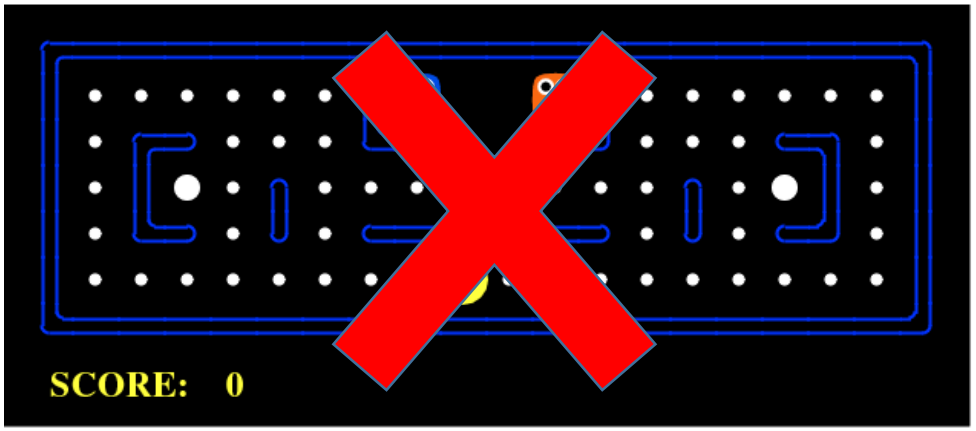
- Grid and your position is stored as an **array**
- Receive input from a device
- **If** the square in the direction you want to go is a wall (Flag = 1) **then** you can't go there
- As the monster, **for** each of the squares around you **then** check which one you can move into (that minimises the distance to quarry, *i.e*. you)
- If either you or the monster move, then **write** a blank square at old position and the relevant character at the new position

Teaching experiences

- Whilst at Southampton, in particular from 2010-2014 approx.
- Programme Leader for civil engineering
- Module co-ordinator for 1st year UG module including computer programming
- Personal tutor for some reluctant learners of computer programming

Further personal reflection…

Further personal reflection



Why were we teaching programming?

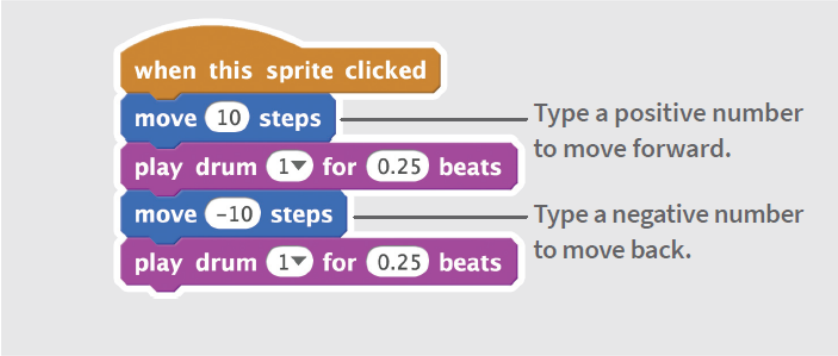
- Our vision was that students would be able to (*and want to*)…
  - Programme macros in Excel to undertake complex tasks
  - Write script files to execute other programs *e.g*. FE analysis programs, efficiently and repeatedly
  - Process ‘big data’
  - Interrogate instruments *e.g*. in lab or in the field, to read data and analyse/display it

The syllabus…

- A basic program (\Hello World").
- Base types (binary, decimal, HEX)
- Boolean logic AND, NOT, OR
- IF…THEN
- Iterative loops: FOR and WHILE.
- Data types: *e.g*. tuples, arrays, dictionaries.
- Code structures *e.g*. functions classes and modules.
- Solve a set of linear equations.
- Generate a graphical plot of a function.
- Use the Newton-Raphson method for root finding
- Demonstrate numerical differentiation and integration

The software

- Started with SCRATCH
  - But seen as a bit too ‘basic’



- Moved to Python
  - Much like BASIC…

Student response

- *What's it for???*
- Huge difficulty at first, verging on tears of frustration
- Much better after 4-6 weeks, all fine at the end – everyone could do it (12 week semester)

And did they carry on to use it later in projects etc.?

- Not really, no
- Seen as a ‘bolt-on’ to the curriculum
- Those that used programming later tended to be self-selecting, *i.e*. had previous experience anyway outside of formal study

Schools: a history lesson…

- National Curriculum in England introduced 1988
- Computer programming not part of National Curriculum in England, until 2014…

"ICT used to focus purely on computer literacy – teaching pupils, over and over again, how to word-process, how to work a spreadsheet, how to use programs already creaking into obsolescence; about as much use as teaching children to send a telex or travel in a zeppelin.

Our new curriculum teaches children computer science, information technology and digital literacy: teaching them how to code,and how to create their own programs; not just how to work a computer, but how a computer works and how to make it work for you."

Michael Gove, January 2014

Another quote from 2014…

- Bill Mitchell, Director of Education at BCS

"To me, the basic idea of computing is you have to get a computer to solve a problem: you have to come up with an algorithm, a set of instructions. If you can do that, it's a hugely valuable skill whenever you're working as a team for any kind of project," he says.

"Also, think about other subjects. When you learn physics, you think about physics. But when you learn computing, you are thinking about *thinking*. About how thinking works. You have to try to imagine how this computer is going to do something for you. There are lots of transferable skills."

The National Curriculum in England now

- Key Stage 1 (ages 5-7)
- Key Stage 2 (ages 7-11)
- Key Stage 3 (ages 11-14)
- Key Stage 4 (ages 14-16, GCSE)
- Post-16
- Computing/computer science defined as an academic *'core' subject* alongside science, maths, English, history/geography, modern language
  - So it has to be available to everyone at GCSE

Other nations in the UK

- Scotland
  - Curriculum of Excellence, since 2010
  - Experiences and outcomes
  - Technologies – Computing Science
- Wales
  - National Curriculum for Wales
  - No mention of computer science?
- Northern Ireland
  - Northern Ireland Curriculum
  - Digital Skills Framework, launched 2014 {Digital Skills}

{Digital Skills}

Digital skills are starting to penetrate every aspect of employment and, as a result, young people are increasingly more likely to need good digital skills. Being able to use digital devices, like an iPad or an Android tablet, is not going to be enough to gain some forms of employment in the future. In fact, being able to use devices will be the basic standard required in any future employment.

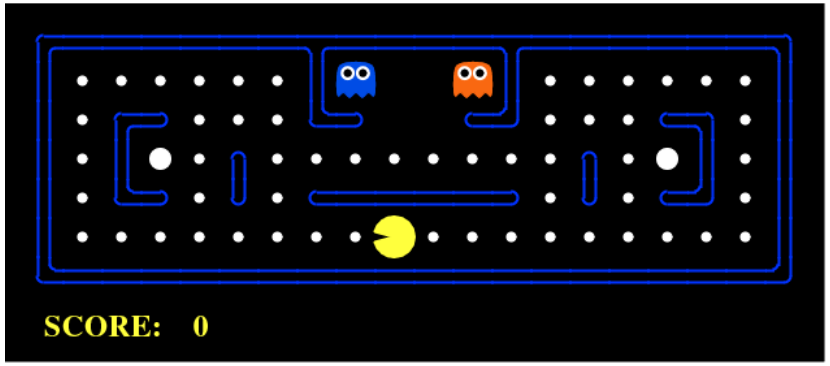
Increasingly young people will need to learn how to build things with digital devices. They will need to be able to programme or code and they will need to think logically and solve problems with different digital products. They will all need to become digital makers and workers.

Justin Edwards, CCEA Chief Executive Officer, Nov 2016

Questions/implications

- The world is changing so fast… VR, AI, autonomy, smart infrastructure, BIM, *etc*.
  - Civil engineers have a 2% probability of being replaced by a robot in the next 20 years (University of Oxford study reported by the BBC, Sept 2015, http://www.bbc.co.uk/news/technology-34066941)
  - The figure for Chartered and Certified Accountants is 95%
- We civil engineers have self-selected away from programming but now we can't avoid it
- All our (UK) intake will have done it beforehand

Questions/implications

- We can redesign curriculum to have much more hope of handling big data, using sensors in lab, developing decision tools/AI in all aspects of civil engineering
- And what about the transferrable skills, the thinking skills, ability to split a problem down into its component parts?
- It's creative – controlling and harnessing the computer, not the other way around
- Looking forward to a new and glorious future

Thank you