EVALUATION OF THE SUITABILITY OF COMPUTER-PROGRAMMING TEACHING-APPLICATIONS FOR THE COMPUTER SCIENCE CURRICULUM IN ENGLAND

Theocharis Kyriacou, Joseph Brennan and Ed de Quincey
School of Computing and Mathematics, Keele University
Keele, ST5 5BG, Staffordshire
Shortage in IT professionals

- There is an undisputable increasing demand for Information Technology (IT) professionals (Lane-Fox, 2014).
- This demand is not anywhere near satisfied in terms of graduates in the sector.
- This shortage will become worse if there is no change in the attempts to motivate high school graduates to follow IT-related paths in their further study (Wellman, et al., 2009).
Programming in particular…

- Whilst computer programming is a core skill featuring in the national curriculum for computing (UK Department for Education, 2013) it appears that this is one of the main elements of computing that demotivates many students from considering an IT-related career (Bergin, et al., 2006).
Programming in particular…

- Traditional methods of teaching computer programming fail to excite the learners of today.
- Such introductions to computer programming today achieve almost the opposite.
- They are perceived as boring, uninteresting or even patronising (Kinnunen & Malmi, 2006).
- Also, they give the false impression that there is a very long and challenging road ahead to reach the stage of creating interesting visual applications (Esteves, et al., 2008).
Why should we (in HE) care?

- We inherit the problem!
- Students come to study at university with a fear of programming.
A solution…

- A significantly revised primary and secondary education curriculum for computer science has been released by the Department for Education and has taken effect from 2014 (UK Department for Education, 2013).
“Teachers' confidence varies to a large degree. It will depend on who is in your school. Teaching ICT is a statutory requirement, but there is no more official statutory programme of study. I think there may be a bit of a vacuum.” Nigel Hooton, ICT leader at St Peter's primary school in Romford, 2013.

“There is some alarm in other schools that they don't have this kind of skill.”, Darren Kelly, curriculum leader for ICT at Blatchington Mill secondary school in Hove, 2013.

“You do expect some framework and guidance to follow. We could get to 2014 and find that everything we've been working on is not what they want us to do”, Kelly, specialist ICT school, 2013.
...that caused other problems!

- This rapid change has made the existing problem even more prominent.

- Enormous pressure on teachers:
  - Lack of experience on the subject,
  - Lack of guidance as to how to teach it and
  - Lack of resources to teach it with (Tickle, 2012).

- The problem is particularly prominent when it comes to the teaching of computer programming (a completely new and challenging skill for some of the teachers).
Some initiatives to help

- Even though there have been some initiatives since 2014 to train teachers on the subject, these have been delivered in a ‘piecemeal’ fashion (Dredge, 2014).

- There is still a problem of a missing overarching framework with which to teach computer science.
Available tools/resources

- Web-based repositories, such as tesconnect\(^1\) and Computing at School (CAS)\(^2\) but resources provided are narrow and disjointed.
- Effectiveness of the majority of resources is not validated or tested adequately.
- No consistent support for any problems that may arise following their use.

\(^1\)https://www.tes.co.uk (Last accessed on 28/4/2015).
Available tools/resources

- A number of applications exist that can be used to support the learning of programming. For example SCRATCH, Kodable, HOPSCOTCH, Tynker, etc.

- Popular with learners and teachers and some schools are using them in order to teach aspects of the programming curriculum.

- Resources for them in CAS and TES in terms of presentation slides, examples and exercises.
Suitability of learning applications

- Evaluated 25 of the most popular teaching/learning programming applications.
- Against 26 computer programming requirements from:
  - the computing curriculum guidelines/specifications for key stages 1 to 5 and
  - the Computer Science GCSE and A-Level specifications of the three main examination boards (AQA, EDEXCEL and OCR).
Examples of the applications
Examples of the requirements

- Primary school (KS2):
  - Sequencing
  - Variables
  - Repetition
  - Selection

- Secondary school KS3-4 (GCSE):
  - Data Structures (Lists, tables, arrays)
  - String Manipulation
  - 2D Arrays
  - Arithmetic Operators
  - Procedures or Functions
  - Data Types
  - Etc.
Examples of the requirements

- Secondary school KS5 (A-Level):
  - Iteration (Both For and While/Repeat)
  - Selection (Both IF and CASE)
  - 3D Arrays
  - Linked Lists and trees
  - Stacks
  - Queues
  - Pointers
  - Etc.
Results

KS2

KS3-4

KS5
In conclusion

- There is not one well established tool or even a set of tools that can be used to teach programming to secondary education students (key stages 3 to 5).
- Except from very brief guidelines and requirements, there is no teaching and learning framework in place for using any existing tools and resources in order to teach programming.
- Programming books are a solution but…
So we need a framework

- A well-defined path/process (supported by resources) for teaching and learning programming that will cover all the requirements set out in the computing curriculum as well as those of the exam boards in the UK.
- ...that incorporates themes/metaphors that will serve as motivation/springboard for learning a general-purpose computer language.
- Recently submitted ESRC grant with the above aim.
References