

High School Computing Clubs: A pilot study

Andrew Bennett
School of Computing
University of Leeds
UK

andrewb@comp.leeds.ac.uk

Joanna Briggs
School of Computing
University of Leeds
UK

joannab@comp.leeds.ac.uk

Martyn Clark
School of Computing
University of Leeds
UK

martyn@comp.leeds.ac.uk

ABSTRACT

While classes in IT skills are endemic, high school students in the UK rarely experience computer science. We present a pilot of a scheme that aims to go some way towards addressing this. Specifically, computing clubs were run on high school premises by high school teachers using material prepared by the University of Leeds School of Computing and supported by volunteer undergraduate mentors. Feedback suggests that the clubs were highly successful in their objectives of broadening understanding of the idea of a computer and introducing the concept of a computer program. School students, their teachers and the undergraduate volunteers all report an enjoyable, purposeful experience.

Categories and Subject Descriptors

K.3 [Computers and education]: Computer and IS education

Keywords

Programming, high school, IT skills

1. INTRODUCTION

In the UK high school students rarely have the opportunity to experience computer science (CS). Comparatively few CS graduates enter the school teaching profession and anecdotal accounts by undergraduates of their computing experiences in schools are almost invariably unflattering. Indeed, even undergraduates admitted to university CS programmes with school-level computing qualifications tend to report minimal exposure to fundamental aspects of CS degree programmes such as modelling and programming. As a contribution towards taking real computing into schools we developed the idea of an extra-curricula activity. Here we report on a pilot scheme that ran in two high schools in the autumn of 2005.

Although we would argue that this type of initiative is desirable *per se*, the decision to commit resources to school-

level CS at this time is rooted in real concerns for the future of the discipline and profession. The UK IT workforce numbers some 1.2 million and, despite offshoring – which is forecast to claim 12% of IT jobs by 2010 – is forecast to grow at between 1.5 per cent and 2.2 percent per annum for the next decade [2, 1]. However, applications for admission to CS degree programmes are declining sharply – having peaked at over 27,000 in 2001, applications to study computing had fallen to less than 15,500 in 2004 [3]. Although this phenomenon is not well understood – and the effects of Y2K and the dot-com bubble cannot be discounted entirely – it is increasingly clear that high school students' appetites for computing are sated by their experience of 'IT skills' courses focusing primarily on the use of word-processing and spreadsheet applications. For example, one study found that school students see IT as being boring and difficult and that students felt that having studied the subject in school there was no need to take computing further [4].

At its simplest, therefore, the idea of the after school computing club is to take computer science into schools and to distinguish the discipline from IT skills courses. More specifically, we wanted to introduce school students to the idea of a computer program and to illustrate that computing is concerned with more than merely PCs. The following sections describe the structure and organisation of the scheme, the content of individual sessions and the results of an evaluation of the pilot. We conclude that the scheme has been broadly successful and we outline plans for future development.

2. ORGANISATION AND STRUCTURE

Computing clubs ran on school premises under the leadership of school ICT staff using resources prepared in the University of Leeds School of Computing and supported by undergraduate mentors. Local conditions were allowed to drive issues such as when and for how long the clubs met and the number and age of the students participating. In the event each pilot club met once a week beginning immediately the school day ended. Each ran for approximately one hour and together they recruited a total of some twenty-five 12-13 year olds.

For each club we recruited two volunteer mentors from undergraduate computing programmes. A bulletin board advertisement generated sufficient initial expressions of interest and this interest was developed in conversations with relevant staff. Undergraduates committed themselves to support all of the eight sessions at their allocated school and together with travelling time this implies a total commitment of two hours per week for eight weeks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE'06, June 26–28, 2006, Bologna, Italy.

Copyright 2006 ACM 1-59593-055-8/06/0006 ...\$5.00.

The pilots were planned as a sequence of eight club meetings that would take place in the school IT facility. A ‘work plan’ was produced by the University with the intention that this would be modified as required by the school teacher leading the club. By design, the early sessions were ‘tasters’; the work plans for these being more self-contained than later sessions.

Each work plan was related to the central theme of producing a tetris-style game to be played on a mobile phone. The game would be produced on school PCs running Microsoft Windows XP. Notepad was used for editing code that was compiled using Java 1.4.2 SDK and Java 2, Micro Edition (j2me) wireless toolkit (2.2 release). Clearly, necessary preparation for the schools included ensuring the required software was available.

2.1 Scheme of work

For the pilots the scheme of work was based on eight club meetings. The first two sessions covered block design and the following four sessions concerned the functionality of the game. Session seven was devoted to the design of a title page, an ‘about’ page and a rules page, and the final session focused on compiling the final game and making it available to play either on a Web site or, where possible, on the pupil’s own phone.

2.1.1 Block design

Session one began by introducing the game, known as I-block, and the concept of a block. Aspects of Java programming were introduced, particularly those concerned with files, the need for an editor and the process of compilation. A worksheet introduced the game as follows:

In I-Block pieces fall through space and during this time you can move them to the left and right and rotate them. The aim of the game is to use the differently shaped blocks to form complete rows. When you complete a row, that row disappears, the blocks above it fall down to replace it and you score points. The aim of the game is to score the greatest number of points. The blocks are of different shapes, sizes and colours, which complicates the game.

We provided code for a simple block and developed exercises for students to add blocks, set the colour of blocks, and to change the sequence in which blocks appear in the game. Sample code is shown below.

```
//DESIGN OF FIRST PIECE (piece number 0).
public static int [][]piece0_Top = {
{ BLUE , BLUE , BLUE },
{ NONE , NONE , NONE },
{ NONE , NONE , NONE }
};
public static int [][]piece0_Left = {
{ BLUE , NONE , NONE },
{ BLUE , NONE , NONE },
{ BLUE , NONE , NONE }
};
public static int [][]piece0_Bottom = {
{ NONE , NONE , NONE },
{ NONE , NONE , NONE },
{ BLUE , BLUE , BLUE }
```

```
};
public static int [][]piece0_Right = {
{ NONE , NONE , BLUE },
{ NONE , NONE , BLUE },
{ NONE , NONE , BLUE },
}
```

The activity of defining blocks continued into the second session.

2.1.2 Functionality

The following four sessions were concerned with the functionality of the game, in particular detecting and deleting complete rows, setting the speed of the game and associated levels, setting up a scoring system and then recording high scores. This was achieved by providing pupils with sample code that they entered into the editor and compiled. An example of code from a worksheet is provided below.

In comparison with the early sessions these activities involved significant elements where code merely had to be copied from worksheets or pasted from examples. Clearly, this offered less scope for individuality and creativity, but pupils learnt quickly about the consequences of entering code inaccurately. The main areas of functionality that students were able to customise included:

- Setting up a scoring system of their own based on the completion and deletion of rows.
- Deciding on a speed for the basic game and the increments as the game progressed through different levels.
- Formatting the high scores page and deciding how many highest scores to display.

In each instance a basic example of the code was given and students were advised how it might be modified. An example of the code is provided below.

```
public boolean checkHighScoreStudent()
{
if(gameScore>getHighScore(1))
{
storeScore(gameScore,1);
askName("What is your name?");
return true;
}
else
{
if(gameScore>getHighScore(2))
{
storeScore(gameScore,2);
askName("What is your name?");
return true;
}
else
return false;
}
}
```

2.1.3 Information pages

This session focused on the aesthetics of the game. During this session the pupils were able to format a title page for their game, including changing its name. They also produced an ‘about’ page where they were able to add their own name as a contributor to the development of the game. They could add a photo of themselves here if they wished. Finally, they wrote a set of rules for their game and designed a ‘rules’ page.

2.1.4 Final compilation and download

The final session involved compiling the final game and uploading it so that the individual pupils could play their game and share it with their friends and family. As a first step everyone was shown how to upload the game to a Web site. For those with compatible mobile phones, it was possible to download the game to their phones. However, the pupils were cautioned that this might be problematic and so to not expect to be able to play their game via a phone. By making the game available on a Web site everyone could maintain access to their work at school and, provided they had Internet access at home, there too.

3. EVALUATION

To evaluate the pilot clubs feedback was sought through interviews with students, undergraduate mentors and teachers. Interviews were undertaken by the University’s project co-ordinator and took place during Session 7. Verbatim transcriptions from interview notes are provided in italics. In addition, we provide the reflections of the academic responsible.

3.1 The student experience

Five students were interviewed. Their reactions to the club were generally enthusiastic. In particular, both the idea of basing the exercise on a game and the idea of the game being played on a mobile phone were popular. Similarly, the support of undergraduate mentors was appreciated by students. Students were also able to make suggestions concerning potential improvements.

The students were enthusiastic about producing a game for their mobile phones:

The club is great – you were taught how to make a game and got to play it at the end. Playing the game and being able to download it to my phone were the best things.

Indeed, some students were sufficiently enthusiastic to have taken their programmes home for further development:

I worked on my game at home. I used a USB [device] to take it home and I played with it on my computer.

The level of difficulty was judged to be about right, though there was some opinion in favour of more time:

Not too complicated, not hard, not easy, about right.

A two-hour session would be better – I wouldn’t mind staying longer.

The undergraduate mentors were popular with students – and some preconceptions were challenged:

I liked the [university] people coming in and showing us how to make it because they know what they are doing.

Good to have the mentors – you could call them over and they helped straight away.

Before I met the mentors I thought that computer scientists have thick big square glasses, big bulging brains, use Linux machines or Linux PDAs, wear ties or have their top button done up and have back-combed hair. Now I think they are normal people but cleverer.

I expected serious nerds but they were normal so now I know what computing people are like.

Students were less enthusiastic about using Notepad and about some aspects of the work plans that relied upon copying or pasting pre-prepared code – although there are hints that such comments were made by more knowledgeable students:

It would be better to use jEdit or similar rather than Notepad.

Copying bits of code was boring – you should make it so that it is more enjoyable. Perhaps explaining more about what we were copying would help – i.e. what each function is doing.

In general, however, students were keen to continue the club – some even had ideas for future projects:

Next I would like a football game.

I would like another game for my phone.

If we do it again, I would like a 3D game like Motas by John Albert.

3.2 The mentor’s perspective

Undergraduate mentors were also generally positive about the clubs and all are keen to continue their involvement as the project develops.

The clubs have been extremely enjoyable, teaching kids this way is a brilliant concept.

When I was at school I learnt to program using Pascal – we had to work out the number of points available on a snooker table according to how many balls were left – a mobile phone-based game is much more inspiring and engaging.

The basic idea of a game, especially one which can be used on a mobile, as the basis for developing skills is great.

Interestingly, some mentors perceived a gender basis to students’ reactions to the activities. For example:

Girls tended to be keener on the design of the blocks and the result was a range of differently coloured blocks. Meanwhile the boys produced simple one-colour blocks and were keen to learn how to manipulate them with the program code.

Of course, the undergraduate mentors understood the limitations of the activity. Nevertheless, some mentors saw evidence that students were learning some elements of programming.

They didn't learn to program, rather to copy and paste, because the way that the session was presented meant that they had no idea what was happening in the background, although they could see the results of what they were doing. They did grasp the broad concepts such as procedures and compilers.

Two pupils in particular are budding programmers – they were always asking about what I study at university on a computing degree and were very on the ball with the club materials.

A couple of the pupils have been really on the ball. Its been amazing how they have grasped it.

Similarly, undergraduate mentors were impressed by the dedication of some of the students.

In one instance one of them stayed behind by half an hour with me to figure out why one part of the code didn't seem to be working because he was so determined to sort out the problem.

Like the students, undergraduate mentors highlighted the limitations of Notepad as an editor for programming. In addition, there was a feeling that students would benefit from more comments in the code.

Notepad has been very basic to use and gives no colour-coding to the text. Perhaps another editor would have been better such as gVim or similar. Not only would this have helped the pupils, but also us to spot the errors.

More comments through the program code would help the pupils to understand what that section of the code is doing. For those who are interested this would be great.

3.3 The teachers' view

The school teachers leading the clubs were similarly enthusiastic and each has plans to continue and expand the activity. Like the students and the undergraduate mentors, the teachers identified problems with Notepad. Similarly, they shared the students' reservations about the amount of 'copy and paste' involved in some aspects of the scheme of work.

Using Notepad to edit the code produces only black text. Using a different package to edit the file which had some form of colour coding would be extremely helpful both to the pupils and the mentors.

The early sessions on block design were brilliant ... these early sessions built confidence because the pupils could quickly extend and develop the code and the fruits of their labours could quickly be seen. The other sessions have been less interesting and appealing to the pupils because they deal more with the background to the game and as a result are less creative. Largely these sessions have simply involved copying the code and so are less intuitive. Having said that, these sessions highlighted the difficulty in copying something complex so from that point of view these sessions were a real eye-opener.

The teachers were also able to offer suggestions that would streamline the running of the club. For example,

It would be helpful to be able to update the background code without affecting the look of the game, for example to have one file containing the code for the blocks along with perhaps the outline for the title page, about page and rules, and another for the background code. In this way the [individual students'] designs could be preserved and missing or incorrect code could easily be replaced.

3.4 The academic's view

Although only a small contribution to a significant problem, the feedback and analysis presented above confirms that the clubs have succeeded in engaging and enthusing school students. Alone, this will not overturn the trend of falling applications for CS programmes in universities but it is undeniable that these groups of high school students have had a different, and, in our view, more appropriate experience of computing.

Although not part of the objectives for the pilot, it is heartening to see that some of the mentors are interested in a career in the teaching profession. We do not claim, necessarily, that the mentoring experience is the sole explanation for such career aspirations but an increase in the number of CS graduates teaching in high schools can only benefit students' understanding of the discipline and, we believe, their enthusiasm for it.

Finally, the teachers' enthusiasm to continue the clubs with new cohorts of student is motivating and rewarding. By design, the materials can be reused with only minor revision. Thus support for the teachers' plans for extending the clubs requires only that we recruit sufficient undergraduate volunteers. With the positive experience of the pilot we have no hesitation in undertaking this endeavour.

4. CONCLUSION

We conclude that the pilots have been successful in the basic objective of taking *real* computer science into high schools. Of course, the exercises were relatively simple and we can make no claim to be solving the problem of falling applications to CS degree programmes. Nevertheless, as a result of the pilot clubs a small cohort of school students enjoyed an experience that otherwise would not have been available to them. Further, the experience as volunteer mentors has undoubtedly been of benefit to our undergraduates.

From the pilot, we have learnt a number of lessons that will help future development of the idea. Specifically, we

will adopt an alternative editor, seek to reduce the amount of copying/pasting of code, and restructure elements of the program in response to teacher comments.

We thank the school teachers, without whom the scheme would be impossible. It is gratifying to learn that they remain committed to the project and have a variety of ideas for its development and expansion. Examples of likely future developments include:

- Running the clubs again at the same schools but with different cohorts of students.
- Using the materials with students studying A-level computing (the main pre-university school level qualification in England).
- Running a version of the club for parents of school students.
- Extending the clubs to other schools.

If all of these possibilities are realised we may not be able to recruit sufficient undergraduate volunteers from within our own institution. Therefore, negotiations are underway to develop the possibility of recruiting mentors for other universities.

Meanwhile we will begin to work on a second set of exercises, perhaps based on a new project theme in order to offer a further experience for the students who participated in the pilot.

5. REFERENCES

- [1] British Computer Society. Offshoring: A challenge or opportunity for British IT professionals. Swindon, 2004.
- [2] E-skills UK and Gartner Consulting. IT insights: Trends and UK skills implications. London, 2004.
- [3] G. Lovegrove. BCS proposal to HEFCE: Widening participation preliminary work to increase the supply of IT students, 2005.
- [4] A. Mitchell. Computing science: What do pupils think? available at: http://www.ics.heacademy.ac.uk/student_retention/workshops/pres/Alison%20Mitchell.ppt. [accessed 10.1.06], 2005.