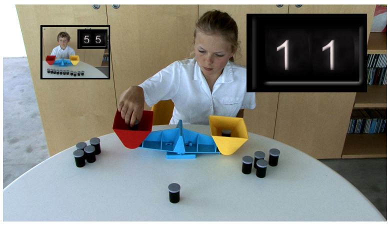
Canterbury Computer Science education research group

Tim Bell, 4 Feb 2011

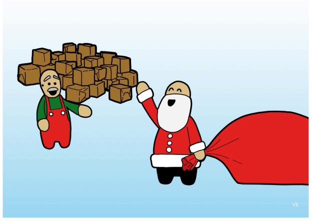
This is a brief summary of the activities of the University of Canterbury Computer Science education research group. The group is led by Tim Bell, with involvement from Mick Grimley (College of Education). Currently there are about 8 students doing research associated with the group; they are also involved in departmental outreach as well as education research and teaching.

Computer Science Unplugged



Overview: The CS Unplugged project is at the core of the CS Education group's work. CS Unplugged (csunplugged.org) provides a variety of resources that engage students in Computer Science activities without using a computer, instead interacting with ideas from Computer Science through magic tricks, games and puzzles [Bell 2009]. These activities generally have a strong kinaesthetic component and take a constructivist approach: students are given enough clues to enable them to work out principles for themselves. A constructivist approach is important for outreach because it demonstrates to the students that much of the knowledge is something they could have invented themselves; and of course, it is a lot more engaging that simply being told some impressive facts.

The CS Unplugged resources are available for free download; as well as activities with specific guides for the presenter, there are videos demonstrating the activities or providing challenges to students, and links to extensive related material for follow-up. Another format for the material is a one-hour show [Bell 2000], designed for a broad coverage of CS topics in a short time, rather than in- depth work by the students.



CS Unplugged is used in many situations around the world, generally relating to one-off events and visits, but increasingly as components of teaching programmes [Choi et al 2008, Bell et al. 2010]. It started around 1992 as a collection of ideas for outreach from universities to K-12 schools, but through its inclusion in the ACM K-12 curriculum in 2003 it started to be seen as the basis of a new approach for teaching CS in schools, either as the main material, or a supplement that gives students a break from being in a computer lab. It has been sponsored by Google since about 2006.

Translations: The CS Unplugged books have been translated into about 14 languages, and the videos have translated soundtracks and/or subtitles [Bell et al. 2008]. This is largely volunteer work from interested people in other countries, although the Chinese version that has just been released was a major re-write, designed for local culture, and for use by students rather than as a guide for teachers.

介绍
Introduction

通过前三章的学习，我们知道计算机是如何储存数字、字母和文本信息的。然而在多媒体系统中，计算机需要具备储存并显示图像的功能。

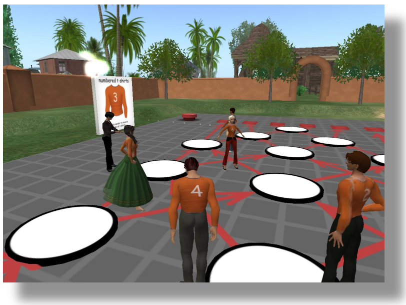
a 计算机需要储存或发送各种不同类型的图像，你能列举出多少种日常应用中的例子？

我们先来看看计算机是如何在屏幕上显示图像的。下图是鹦鹉木偶阿诺德的照片，照片看起来细腻平滑，可是……

……当我们放大它眼睛的部分，却发现细节过度其实并没有那么平滑……

Currently YingChun Han from Kunming Metallurgical College is visiting for 12 months (until April 2011), and is working on a paper about how Unplugged and related styles of teaching fit in to Chinese educational culture. Tim visits China for about a month each year, and uses this time to work with local contacts about CS education in that culture.

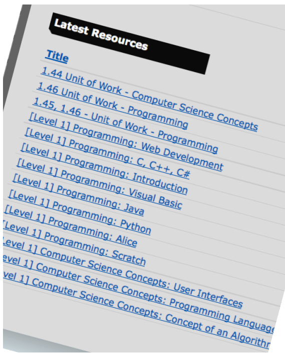
Virtual worlds



Peter Thompson (PhD student) is developing CS Unplugged activities in Second Life and Greenfoot, and evaluating their effectiveness for automated gathering of data for educational research [Thompson and Bell 2010]

Previously we have worked on providing CSUnplugged activities in virtual worlds for students with special needs [Bell et al. 2009, Marghitu et al. 2009]

NCEA resource development and evaluation



Latest Resources

Title

1.44 Unit of Work - Computer Science Concepts

1.46 Unit of Work - Programming

1.45, 1.46 - Unit of Work - Programming

[Level 1] Programming: Web Development

[Level 1] Programming: C, C++, C#

[Level 1] Programming: Introduction

[Level 1] Programming: Visual Basic

[Level 1] Programming: Java

[Level 1] Programming: Python

[Level 1] Programming: Alice

[Level 1] Programming: Scratch

[Level 1] Computer Science Concepts: User Interfaces

evel 1] Computer Science Concepts: Programming Languag

vel 1] Computer Science Concepts: Concept of an Algorithr

Starting in 2011, NCEA will include standards that teach Computer Science [Bell, Andreae and Lambert 2010]. Sumant Murugesh (research assistant) is developing a guide to resources for teaching Computer Science for new NCEA achievement standards [Murugesh, Bell and McGrath 2010], funded by a MoE contract and with support from NZ CS departments. The material is available at http://nzacditt.org.nz/project/programming-and-cs

Brett Ward (honours student) has worked on using Alice and Scratch to do "real" computer science [Bell et al. 2010], and Linda Pettigrew (summer scholar) has implemented several pairs of algorithms in Scratch so students can compare their speeds (e.g. quicksort vs. selection sort) without having to implement them. This supports the algorithms part of AS 1.44. This is to be developed into a larger range of resources to support students exploring the complexity of algorithms without having to implement them.

Samuel Williams (honours student): as a course project, developed posters comparing programming languages, as a starting point for students doing the languages part of AS 1.44.

COSC433 is a 400-level course where students learn about CS education, and develop resources, mainly aimed at the new NCEA standards. The paper was first taught in 2010 with Lynn Lambert [Bell and Lambert 2011]

CS Field Guide

Tim Bell is working with Peter Denning and Rick Snodgrass, who have an NSF grant to develop an extra-curricular system where students can work on Computer Science topics, akin to the Boy Scouts badge system, where there are levels to work through, and mentoring and general activities that support the specific goals.

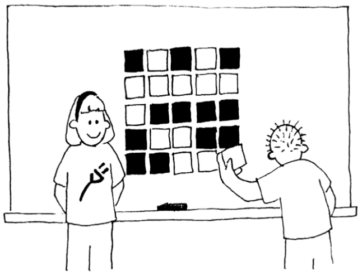
Steve Dunford (summer scholar) is developing a report on badge systems (e.g. Scouts, St Johns, Martial arts) and how to adapt them to a CS based structure.

Non-programming outreach activities

Tim is working with people from the UK (Paul Curzon, Quintin Cutts), Israel (Bruria Haberman) and Lithuania (Valentina Dagiene) on a paper to compare and contrast various approaches to outreach: CS Unplugged, CS4FN, Bebras, CS Inside etc. They all share resources, but have different ways of delivering the material. All are "successful" in that they have had tens or hundreds of thousands of students using them. We are interested in identifying the key elements that make them successful.

Repositories of Learning objects

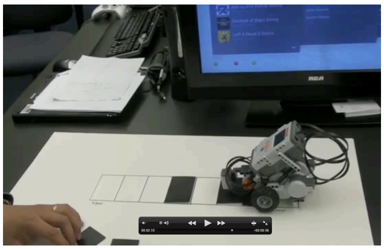
Working with developers of sites with teaching resources to find a way to make the range of material easy to navigate, and to make sensible re-use of the resources for different contexts [Curzon et al. 2009].



Other work

Over the years we've published various papers about novel approaches to teaching Computer Science at university, including digital recordings of lectures [Bell et al. 2001], podcasts [Bell et al. 2007], and a novel way to relate sorting algorithms [Bell and Aspvall 2011].

Future work



- Develop a student version of the CS Unplugged material, aimed at supporting the emerging curricula (including NCEA, AP, and UK curriculum) [Bell et al. 2010]
- Develop new materials for NCEA
- Evaluate the effectiveness of the new NCEA CS curriculum, including effect on teachers, student attitudes, and student achievement.
- Integrate CS Unplugged activities with programming (Scratch, Alice, Greenfoot, Kodu) and Robotics
- Investigating Computational Thinking and CS Unplugged [NRC 2010]
- Story-telling approaches to teaching about Computer Science [Bianco, Bell and Tinazzi 2009]; see also our recent video "Santa's Dirty Socks"
- Establishing how best to design new educational experiences based on the "Unplugged" idea [Nishida et al. 2009]
- Using Unplugged in the context of Science Fair projects and programming competitions [Voigt 2009]

References

[Bell 2000] Tim Bell. A low-cost high-impact computer science show for family audiences. In Australasian Computer Science Conference 2000 (ACSC 2000), 31 January – 3 February 2000, Canberra, Australia, pages 10–16, 2000.

[Bell et al. 2001] Tim Bell, Andy Cockburn, Bruce McKenzie, and John Vargo. Flexible delivery damaging to learning? Lessons from the Canterbury digital lectures project. In Proceedings EdMedia 2001, pages 117–122, 2001.

[Bell et al. 2007] T Bell, A Cockburn, A Wingkvist, and R Green. Podcasts as a supplement in tertiary education: an experiment with two computer science courses. In MoLTA 2007 (Conference on Mobile Learning Technologies and Applications), February 2007.

[Bell 2009] Tim Bell, Jason Alexander, Isaac Freeman, and Mick Grimley. Computer science unplugged: School students doing real computing without computers. The NZ Journal of applied computing and information technology, 13(1):20–29, 2009.

[Bell and Aspvall 2011] Tim Bell and Bengt Aspvall. Sorting algorithms as special cases of a priority queue sort. In SIGCSE, 2011.

[Bell and Lambert 2011] Tim Bell and Lynn Lambert. Teaching computer science majors about teaching computer science. In SIGCSE, 2011.

[Bell et al. 2008] Tim Bell, Ben T Wada, Susumu Kanemune, Xia Xie, WonGyu Lee, Choi SookKyoung, and Bengt Aspvall. Making computer science activities accessible for the languages and cultures of Japan, Korea, China and Sweden. In SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer Science Education, 2008.

[Bell et al. 2009] Tim Bell, D Marghitu, M Grimley, H Manabe, and Giovanni M. Bianco. Kinesthetic computer science activities in a virtual world. In SIGCSE 2009, 2009. Poster.

[Bell et al. 2010] Tim Bell, Arno Pasternak, Chris Stephenson, Allen Tucker, and Jan Vahrenhold. Implementing CS curricula in secondary education: An international perspective. In The 40th Annual Frontiers in Education (FIE) Conference, 2010.

[Bell, Andreae and Lambert 2010] Tim Bell, Peter Andreae, and Lynn Lambert. Computer Science in New Zealand high schools. In Tony Clear and John Hamer, editors, ACE '10: Proceedings of the 12th conference on Australasian Computing Education, volume 32 of Australian Computer Science Communications, pages 15– 22, Brisbane, Australia, January 2010. Australian Computer Society, Inc. invited paper.

[Bianco, Bell and Tinazzi 2009] Giovanni M. Bianco, Tim Bell, and Simonetta Tinazzi. Introducing powerful ideas to children: analogical mediation for computing concepts through storytelling. In SIGCSE 2009, 2009. Poster.

[Choi et al 2008] Sook Kyoung Choi, Tim Bell, Soo Jin Jun, and Won Gyu Lee. Designing offline computer science activities for the Korean elementary school curriculum. In ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education, page 338, New York, NY, USA, 2008. ACM.

[Curzon et al. 2009] Paul Curzon, Peter W. McOwan, Quintin I. Cutts, and Tim Bell. Enthusing and inspiring with reusable kinaesthetic activities. SIGCSE Bull., 41(3):94– 98, 2009.

[Marghitu et al. 2009] Daniela Marghitu, Tim Bell, Ben Kearns, Brett Ward, and Kim Stephen-Pope. Using virtual worlds to engage typical and special needs students in kinesthetic computer science activities: A computer science unplugged case study. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2009, pages 183–188, Honolulu, HI, USA, June 2009. AACE.

[Murugesh, Bell and McGrath 2010] Sumant Murugesh, Tim Bell, and Ann McGrath. A review of Computer Science resources to support NCEA. In Samuel Mann and Michael Veerhaart, editors, First annual conference of Computing and Information Technology Research and Education NZ (CITRENZ2010), pages 173–181, July 2010.

[Nishida et al. 2009] Tomohiro Nishida, Susumu Kanemune, Yukio Idosaka, Mitaro Namiki, Tim Bell, and Yasushi Kuno. A CS Unplugged design pattern.

In Sue Fitzgerald, Mark Guzdial, Gary Lewandowski, and Steven A. Wolfman, editors, SIGCSE, pages 231–235. ACM, 2009

[NRC 2010] Committee for the Workshops on Computational Thinking; National Research Council: Report of a Workshop on The Scope and Nature of Computational Thinking, , ISBN: 0-309-14958-4, 114 pages, 6 x 9, (2010), available from: http://www.nap.edu/catalog/12840.html

[Thompson and Bell 2010] David Thompson and Tim Bell. Virtual worlds: Evaluating CS education activities through automated monitoring. In SDPS-10, pages 1–7. Society for Design and Process Science, June 2010.

[Ward et al. 2010] Brett Ward, Tim Bell, Daniela Marghitu, and Lynn Lambert. Teaching computer science concepts in Scratch and Alice. The Journal of Computing Sciences in Colleges, 26(2):173–180, December 2010.

[Voigt 2009] Janina Voigt, Tim Bell, and Bengt Aspvall, Competition-style programming problems for Computer Science Unplugged activities. (2009) CEDETEL