Introducing TU100 'My Digital Life': Ubiquitous computing in a distance learning environment

Mike Richards
Computing Department,
Faculty of Maths, Computing and Technology,
The Open University,
Walton Hall,
Milton Keynes.
MK7 6AA
m.richards@open.ac.uk

John Woodthorpe
Department of Communications and Systems,
Faculty of Maths, Computing and Technology,
The Open University,
Walton Hall,
Milton Keynes.
MK7 6AA
j.woodthorpe@open.ac.uk

ABSTRACT

In this paper we describe the Open University's progress towards delivering an introduction to ubiquitous computing within a distance-learning environment. Our work is strongly influenced by the philosophy of learning-through-play and we have taken technologies originally designed for children's education and adapted them for adult learners, many of whom will have no formal experience of computer science or information technology.

We will introduce two novel technologies; Sense, a drag-and-drop programming language based on Scratch; and the SenseBoard, an inexpensive hardware device that can be connected to the student's computer, through which they can sense their environment and display outputs.

This paper is not intended as a detailed discussion of individual technologies (they will follow in time), rather it should serve as an introduction to the Open University's method of teaching and how we hope to continue to recruit new computer scientists and engineers using novel technologies.

Author Keywords

Ubiquitous computing, distance education, adult learners, programming

ACM Classification Keywords

K.3.2 - Computer and Information Science Education

General Terms

Experimentation, Human Factors, Languages.

UbiComp 2009, Sep 30 – Oct 3, 2009, Orlando, Florida, USA.

DISTANCE LEARNING AT THE OPEN UNIVERSITY

With more than 250,000 active students, the Open University (OU) is Britain's largest university and is currently celebrating its 40th Anniversary. The university offers independently audited qualifications ranging from introductory certificates, through to bachelors and postgraduate degrees and doctorates as well as a range of recognized professional certifications.

All OU undergraduate students study at a distance. Most courses use printed self-study materials that are ~~regularly~~ assessed at regular intervals through the course and at the end ~~and independently graded~~. The development of new courses may take a number of years; with materials going through a large number of quality assurance procedures in terms of readability, consistency and accessibility to ~~disabled~~ all students. This course development process is extremely expensive and may cost several million Pounds.

Each student belongs to a 'tutorial group' that is run by a part-time, fully-trained 'associate lecturer'; who in turn is supported by central academic staff at the university's main campus in

Milton Keynes. This scalable approach allows the OU to teach very large cohorts of students (some courses may have in excess of 10,000 students during each presentation), whilst ensuring individual students can receive personalized professional support.

Unlike most universities, the OU does not require any previous educational achievements as a condition of entry; instead offering a range of introductory courses teaching a basic grounding in subjects as well as key educational skills which will aid further study. OU student results compare favourably with comparable students from conventional universities.

The OU has always been at the forefront of innovation; not only was it Britain's first distance education institution; but it went on to pioneer the use of educational television and radio programming through a long-standing relationship with the BBC. During the 1990s the OU trialed many of the Internet technologies; such as electronic delivery of materials, computer conferencing and computer marking, which are now commonplace in other universities [1].

MODERNISING THE OPEN UNIVERSITY CURRICULUM

~~A recent reorganization within the University resulted in the merger of two major faculties – that of Technology and the Faculty of Mathematics and Computing. During the reorganization the opportunity was taken to assess the curriculum that had (inevitably) developed some degree of overlap.~~

~~Both of the prior faculties offered an introductory course in the broad area of computer science and technology.~~ The OU has two, extremely popular entry level courses, attracting approximately 4,000 students apiece each year. ~~The Technology faculty offered~~ T175 *'Network Living'* ~~–~~is an introduction to the technologies and role of networks in modern society, whilst Computing offered M150 *'Data, Computing and Information'* is a 'traditional' computing course ~~– a course~~ concentratinging on ~~how data is~~ acqui~~red~~sition, ~~processed~~ processing and ~~used~~use of data.

In mid-2008, a decision was taken to replace the two courses with a single course reflecting the rapid technological and societal developments that had taken place in the last five years~~Both courses had proved extremely popular; attracting approximately 4,000 students apiece over a number of years. However, it was clear that a replacement course would be needed to reflect the rapid changes in the area~~.

The new course had to fit into an existing degree programme and to deliver certain key learning outcomes required for later study; most notably some experience of computer programming. However, the method of teaching could be novel.

Very early on we recognized that ubiquitous computing would allow us to teach all of our existing learning outcomes but provide a unique opportunity to distinguish our introductory offerings from those of any other university in Britain; where ubiquitous computing was taught at all, it was only ever taught at a higher level – usually in the final year of study.

Ubiquitous computing provided us with an opportunity to widen participation in computer science and engineering. If computers are going to be everywhere and indeed 'everyware', then we should attempt to teach the subject in a manner that would appeal to as large an audience as possible.

Course development would extend over two years with the first presentation scheduled for 2011. Creation of the materials would take approximately one year with six months given over to developmental testing by fellow academics, prospective associate lecturers and a number of existing OU students.

COURSE CONTENT

From the outset we chose to develop an approachable course that began when we gave it the name *'My Digital Life'*. The course team felt that giving students some sense of ownership would help them engage with the course materials – and so, the obvious place to start the six teaching blocks was with the students themselves and to gradually expand their view ~~of the world~~ until it ~~encompassed~~ encompasses the whole World; showing how ubicomp has already started to become a reality [2].

Block 1 - Myself

The course starts with a block concerned with our ~~selfish~~ needs to consume, process and publish information. It introduces students to the concepts of data and information and how these are created, distributed and stored. The concept of personalized information is central to this block; students will create their own online 'home' and fill it with appealing information that they want to share with others. In the process they will learn about the underlying structure of the Internet, the software that interacts with the Internet and the way data is encoded into machine-readable forms.

Block 2 - My stuff

The second block expands our view to encompass the devices we use on a day-to-day basis; from the familiar personal computer to smart phones, games consoles and television smart boxes. At their heart each of these devices is a computer. Uniquely, computers can change their behaviour depending on what software they are running.

The block begins by an exploration of the software and hardware used to create these devices and introduces students to the principles of computer programming. The second part of this block is concerned with how we interact with computers; as well as discussing the discipline of human-computer interaction (HCI), the unit explores how smart devices can find and filter useful information from the overwhelming amount of material in circulation. Students will learn how to find, rank and reference information and continue to build their literacy skills.

Block 3 – My place

Expanding outward~~s~~ again, this block is concerned with an individual user and how they interact with computers in a mobile environment. This block provides an in-depth discussion of ubiquitous computing, using familiar devices such as mobile telephones and GPS (Global Positioning System) devices, and exploring ~~exciting new~~ developments including location-based service~~s.~~

~~Students will continue to build your personal space on the Internet and to develop new software and hardware projects that will help their understanding of a fast-growing area. Shared online spreadsheets and other documents will be used in an exploration of the role of new technologies such as~~ s and Cloud computing. ~~However these different systems and approaches cannot exist in isolation and they will see how truly ubiquitous computing will only made possible by the adoption of standards that allow information to flow freely between individuals, networks and countries.~~ The block closes with a look at the role of standards ~~of~~ and contrasts open and closed source methods of working, and will provide some examples of how different business models and ways of working can co-exist in more than just the area of software development.

Block 4 – My friends

The second half of the course sees a greater emphasis on the social aspects of computer technology. This block discusses how computers allow us to make and maintain friendships no matter how great the distance between individuals. The block is devoted to those relationships we choose to cultivate. It is a discussion of social computing where information is freely shared on a consensual basis. It moves on to discuss the more advanced area of shared spaces that began with multi-user dungeons and is now epitomized by online virtual worlds such as Second Life. The block explores what information people share online and why they wish to share it. It closes with two explorations; the first being how social computing is changing the wider world – in the form of political campaigns fought online; the second being the explosive growth of online video games.

Block 5 – My society

Beyond the people we choose to share information with, there are those with whom we *must* engage online whether we choose to or not. This block is devoted to the growth of the electronic society and how such a society has positive and negative aspects. Case studies are used to show how electronic government can benefit healthcare and to examine the role of the database in enabling such technologies. This block also explores the role of the individual in e-society and how individual rights may conflict with those of the state. This introduces some of the legal aspects of computational technology – ranging from protection of personal data to the importance of copyright. It ends with a provocative examination of whether the freedoms created by modern technology are compatible with the security of individuals and the state. This will show you how to form arguments from conflicting evidence and produce your own, researched, opinion on a controversial topic.

Block 6 – My world

The final block takes the widest possible view of computing technology and demonstrates how it is changing the entire world at an unparalleled pace. We contrast those countries such as Dubai and Singapore that are wholeheartedly embracing the opportunities of computer technology with those parts of the world that face being left even further behind the most developed countries. Our discussion of this 'Digital Divide' is not just concerned with the divide between countries, but that found inside countries – between rich and poor, educated and less educated, old and young. We look at some projects and technologies that aim to close this gap. Finally, the course asks opinions; from technology experts, researchers, science fiction authors and from students, of what the future holds – will computer technology lead to a new utopia or to a dystopia?

TEACHING UBIQUITOUS COMPUTING

We suspected that ubiquitous computing would prove to be attractive to students – it was new, exciting and slightly quirky. Unlike longer-established areas of computing and technology, many of the ubiquitous technologies being developed are clearly 'rough around the edges' and there is the still the possibility that a newcomer can make a significant contribution to the field. But we faced two significant challenges;

1. most of our students will come to TU100 without any computer programming experience – let alone familiarity with ubiquitous technologies;
2. ~~1.~~because all of our students work at a distance, we *cannot* offer laboratory sessions where they could use the expensive electronics components used in most university ubiquitous computing courses.

Formatted: Bullets and Numbering

Whilst it would be possible to develop a completely theoretical course, perhaps one supported by video material, it would be a tragedy if our students could not get 'hands-on' experience of ubiquitous computing technologies.

Sense

Current first level students studying M150 use JavaScript for their programming exercises. Whilst this is a well-supported, relatively powerful, conventional language; it has proved to be an unsatisfactory choice. Whilst many students learn to program in JavaScript, a significant proportion of users either withdraws from the course or does not progress to more advanced programming courses citing JavaScript's pedantic syntax and relatively poor support for debugging. Student interviews and feedback from associate lecturers suggested that most students could design an algorithm to solve a problem, but lacked the confidence to turn that algorithm into an executable program.

The OU had previous experience through connections to the well-developed RoboFesta movement [~~2~~3], and in the development of T184 *'Robotics and the Meaning of Life'*, a robotics course for novices [~~3~~4][~~4~~5]. Both of these used the LEGO Mindstorms™ kit ~~which~~ that could be programmed using the drag-and-drop RCX Code environment. We had found that children and adults found drag-and-drop extremely intuitive to use and were able to build relatively complex programs with rich behaviours.

Formatted: Font: Italic

A decision was made to extend the Scratch [~~5~~6] language from the Lifelong Kindergarten Group at the MIT media Lab. Scratch is a media-rich programming environment which is especially notable for its clear programming structure;

individual program blocks – such as if-else statements, logical operators and variables can only be assembled in meaningful (not necessarily correct) manners, and as such helps remove one of the major frustrations of JavaScript – syntax and logic is implicit rather than explicit.

Whilst Scratch does not offer students an immediately useful language for their employment, it does allow them to learn all of the basic skills needed to succeed in any programming language~~,~~; it builds confidence and offers plenty of encouragement to study further – unlike most languages, with Scratch *'you don't need to know a lot to do a lot'*. Scratch has proved extremely popular with educators and students alike [~~6~~7][~~7~~8].

However, Scratch was not ideal for our purposes; it had been clearly designed for children ~~(~~and might feel patronizing ~~for~~ towards adult learners~~)~~, (although our initial tests with adult volunteers revealed a surprising number of them enjoyed the environment's toy-like appearance). ~~, and it~~Scratch also lacks~~ed~~ some of the richer programming concepts – such as lists~~,~~ - needed to satisfy ~~our~~ the learning outcomes needed for further study at the OU. Most seriously, Scratch lived in a sandbox – it could not talk to the outside world – if we were going to use it to teach ubiquitous computing, Scratch was going to need network support.

Fortunately it is possible to modify Scratch to suit our needs; ~~is~~ Scratch is programmed in Squeak and the underlying image file ~~is~~ has been made publicly available for modification. Over the last few months we have been building our own programming environment, *Sense*; which is more suited to adult learners exploring ubiquitous computing.

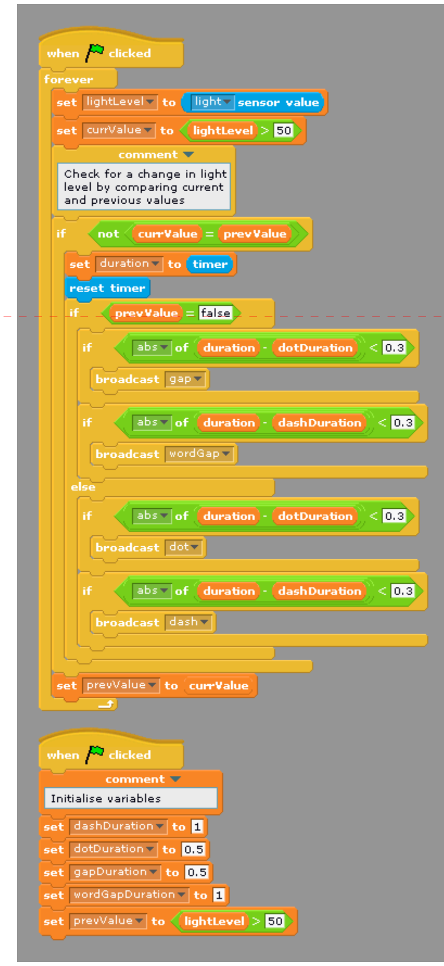


Figure 1. An example of a ~~sense~~ Sense program showing how program structure is made clear using different shaped blocks and colour.

Sense retains Scratch's look and feel, immediacy of operation and welcoming approach that encourages experimentation. Wherever possible we have tried to retain compatibility with Scratch by retaining the names and functionality of individual blocks; but we have added a number of new features such as support for protocols including TCP/IP and RSS, rich data structures and often-neglected commenting.

Formatted: Underline

Formatted: Font: Italic

Formatted: Normal, Space After: 0 pt

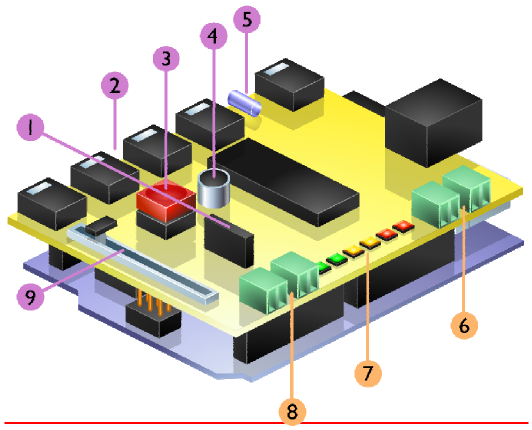
The SenseBoard

In the past, the OU has provided so-called 'Home Experiment Kits' (HEK) to students but had gradually abandoned their use because of the expense of ~~having~~ returning and refurbishing boards ~~returned~~ when they were no longer needed by students before they ~~and refurbished before they~~ could be dispatched to the next student cohort. We decided that this problem could be avoided if the TU100 HEK was made sufficiently cheap that it could be considered 'disposable' so far as the University was concerned. Before committing ourselves to a HEK we assessed a number of 'off the shelf' technologies already used in computer education.

One of the most popular products used in university ubiquitous computing ~~teaching~~ courses ~~are~~ is Phidgets [~~8~~9]; kits of more-or-less plug-and-play electronic components ~~which~~ offering almost unlimited potential for ~~designing~~ experimentation~~s~~. The high unit cost of Phidgets would have affected the profitability of TU100, but they were finally rejected after feedback from potential students who found Phidgets individual electronic components and breadboards to be extremely ~~off putting~~intimidating. Further concerns came~~, and~~ from members of the course team who were worried about ~~their~~ the Phidgets' durability and the possible high rate of returns.

An alternative was the PicoBoard, originally designed by the same team as Scratch, but now marketed by Pico [~~9~~10]. The PicoBoard is a small printed circuit board that can be connected to a computer through a USB port. The board contains a number of analogue inputs, a light sensor, a microphone, a push button and a slider. Although quite limited in its capabilities, the PicoBoard had a number of advantages over the Phidgets; it was cheap (approximately $50) and robust because of its surface mount construction. However, we felt it was lacking in that it did not offer any outputs such as motors or lights.

With no obvious 'off the shelf' option, the ~~Using the PicoBoard as a starting point, the~~ OU commissioned Kre8 Ltd, a company with experience of designing electronic toys to design a low cost sensor board for TU100. The eventual design placed inputs and outputs on a single surface-mount shield that could be plugged into a near-standard Arduino board. The SenseBoard offers a wide range of inputs and outputs for a very low cost (approximately $70 including the Arduino).



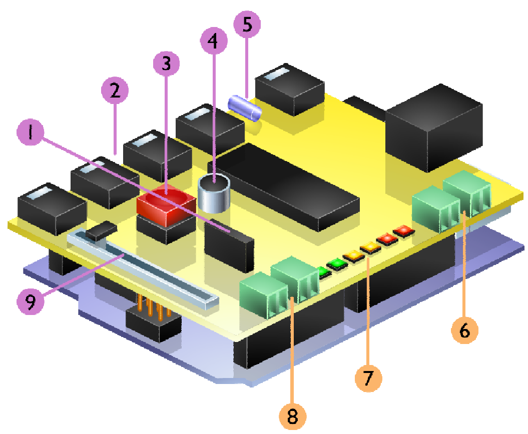


Figure 2. Schematic of the Version 1 SenseBoard (June 2009) mounted on its Arduino host. (Legend below)

The SenseBoard will be accompanied by a number of plug-in sensors such as a thermistor and a motion detector, a pair of motors, a battery pack (needed when using the motors), a USB cable and a magnetic board which allows the SenseBoard and all the associated equipment to be carried from room to room or stored without worrying about stray cables or losing items.

~~An example of a very early project that can be constructed by a novice student in a very short period of time will be to build a simple weather station incorporating a thermistor and a light meter. Initially the student can display the results inside the Sense window on their computer, but after a few more activities they will be able to use Sense to format data into an RSS feed and publish their own results to the Web. They will then be encouraged to look at other students' data and plot their results using technologies such as spreadsheets, online mapping tools or perhaps displaying other results on their SenseBoard using the LEDs or using the motor to drive a pointer.~~

Table 1. Key for the SenseBoard diagram (Figure 2.)

Key	
1	IR detector (remote control)
2	Analogue inputs 1 – 4 (temperature, motion, pressure etc.)
3	Push button
4	Microphone
5	Edge IR emitter / detector (barcodes etc.)
6	External motor 1
7	Programmable LEDs
8	External motor 2
9	Slider

An example of a very early project that can be constructed by a novice student in a very short period of time will be to build a simple weather station incorporating a thermistor and a light meter. Initially the student can display the results inside the Sense window on their computer, but after a few more activities they will be able to use Sense to format data into an RSS feed and publish their own results to the Web. They will then be encouraged to look at other students' data and plot their results using technologies such as spreadsheets, online mapping tools or perhaps displaying other results on their SenseBoard using the LEDs or using the motor to drive a pointer.

The aim of the course is not to exhaust the potential of Sense or the SenseBoard, but to demonstrate sufficient of its capabilities that students will be able to develop and share their own projects. Indeed when the student finishes their studies we would be delighted to know that they were continuing to use the board.

Course materials

As well as the SenseBoard, students will receive their course materials as a series of full-colour bound booklets and either one or two DVDs holding all of the course software (including the Sense environment), audio and video interviews, example projects and guided walkthroughs of key tasks. ~~We aim to produce~~Students will be introduced to their SenseBoards through a number of introductory video 'builds' (much in the style of the popular BBC children's television series *'Blue Peter'*)~~ where students are introduced to their SenseBoard.,~~ We will show~~n how~~ them how to plug ~~it~~ their board in to their computer, test it and then begin programming before moving on to their own projects. All of ~~these~~ the bundled materials will be supported by a course Website, containing links to ~~other~~ Open University resources – such as technical support, the Library and ~~the part of the u~~University's assignment handling~~ that handles assignments~~ system. Students will each have their own personal email box, online file storage area, study calendar, blog and journal. They will converse with their fellows using a range of technologies including instant messaging and small (20 person) forums, each run by an associate lecturer.

The course is assessed throughout. Six written assignments require students to submit a range of materials ranging from short answers through to researched essays and Sense programs. The course's final assessment takes the form of a short project. Obviously, we cannot expect students to submit their SenseBoards as part of their projects, so students will be graded on their programming code, their documentation and results and we will be asking them to contribute photographic and video evidence to support their conclusions.

OPEN SOURCE

The developers of TU100 have decided to release much of their work as open source projects.

1. Whilst much of our teaching material cannot be made freely available; some of TU100's content will be placed on the Open University's OpenLearn site [~~10~~11] for free re-use.
2. The Sense project will be released on a public Web site on an 'as-is' basis.
3. ~~2.~~3.The full specification, layout and component list of the SenseBoard as well as any necessary driver software will be published on a public Web site. Anyone will be able to make their own board without paying royalties, although the board's designers reserve the right to sell their own version of the SenseBoard.

~~FUTURE WORK~~ACCESSIBILITY

~~Although TU100 is a work-in-progress, discussions have already begun on revising the course. A~~The OU has along history of ensuring its course materials can be used by anyone who wishes to study its courses. ~~-significant concern has been the issues of accessibility raised by t~~The Sense environment and the SenseBoard ~~itself, which~~raise profound issues for~~ are especially profound~~ ~~for~~ students with visual impairments or limited motor skills~~ or those suffering from visual impairments~~.

~~-~~UK legislation requires the OU to make 'reasonable efforts' to allow disabled students to access course materials.

The most basic form of ~~accessibility~~provision ~~is~~ requires ~~any~~ disabled students to be supported by ~~an~~ able-bodied assistants. This is obviously unsatisfactory; not only do some students lack an able-bodied assistant, but also needing to ask for help reduces their independence. The ultimate form of ~~accessibility~~ provision is ~~that~~ when all materials are usable by all people no matter what type of requirements they have. This is almost always unobtainable, if only because of the great expense in designing appropriate materials. Inevitably some degree of compromise will be required and it is likely TU100 will become more accessible through time as new materials are developed.

We hope to include a certain degree of accessibility support within Sense before TU100 is released. The OU's Institute of Educational Technology (IET) has produced an initial report on Scratch using a range of different assistive technologies and computer settings. ~~They~~ IET identified a number of issues that can be summarized as follows:

- People who use a screen reader would not be able to access the application;
- People who do not use a mouse would not be able to operate the application easily;
- People with visual impairments or dyslexia may find some text difficult to read due to low contrast with the background;
- People who require large font size may have difficulty reading the buttons/text;
- People who use a screen magnifier may have difficulty reading buttons text;
- People who use voice recognition would not be able to operate the application easily;
- Hearing impaired people should not have any difficulties, apart from being aware when a sound is played in their program;
- People with a manual impairment may find some of buttons to be small targets;

These issues have been passed to the Sense developers so that a costing for a more accessible

Formatted: Bullets and Numbering

version of the application can be presented to the University.

The SenseBoard also presents issues of accessibility. The various connections and input devices obviously present problems for students lacking motor skills. Rather than relying customized components (an approach taken by LEGO with their MindStorms™ kits) which would increase the cost of every SenseBoard to a point where the course was no longer viable; the course team decided to build the board with over-sized components that are easy to grasp and highly robust. Individual items such as power and motor plugs have also been designed in such a way that they can only be connected in the correct manner.

We have also designed an onscreen SenseBoard~~simulator of the board~~; the push button and slider of which can be manipulated using the mouse, the SenseBoard's microphone is replaced with that on the computer, and where sensor inputs can be simulated using type-in numeric values.

Visually impaired students do not currently have any accessibility support on the board. However, the course team has entered discussion with the SenseBoard designers to develop a second version of the board. The new board will use speakers and vibration devices to replace the LEDs on the current board. Instead of sequences of light, outputs would be conveyed using musical notes or pulsed patterns of vibrations. We call this device the SenseBoard Touch.

This may be a separate board from the original SenseBoard automatically dispatched to any students registered as visually impaired. Alternatively,~~; or,~~ if ~~cheap enough~~the board can be manufactured at a low enough cost, we may issue the SenseBoard Touch to all students ~~, we may incorporate the functionality of the SenseBoard Touch on to the original SenseBoard~~ so that all students receive a more capable device.

CONCLUSION

TU100 is a~~n extreme~~ typically ambitious project for the Open University and has required us to develop a number of new technologies to address the particular needs of distance learning students. We have developed a programming environment for adult learners with no prior experience of computer programming, and a complementary piece of hardware that will allow them to begin experimenting with ubiquitous computing.

At a very early stage we realized that a significant number of students may be unable to use some or all of our technologies. biggest unanswered question remains how we can make ubiquitous computing devices that are truly useful to everyone. We are planning further developments of our hardware and software to not only accommodate, but to actively welcome those people who are normally excluded from education in general, and computing in particular.

~~We hope our experiences will prove useful to other educators and~~In order to facilitate the exchange of ideas between educators we will be releasing much of our work as open-source projects.

~~Our biggest unanswered question remains how we can make ubiquitous computing devices that are truly useful to everyone. We are planning further developments of our hardware and software to not only accommodate, but to actively welcome those people who are normally excluded from education in general, and computing in particular.~~

ACKNOWLEDGMENTS

We would like to thank everyone on the TU100 course team for all their work so far and for what they're going to do in the future. An especial thank-you to Dr. Chetz Colwell from IET who conducted the initial usability tests on our work. Also to Syntropy Ltd. for their work on the Sense environment. And to Kre8 Ltd., Garry Bulmer and Robert Seaton for the design and layout of the SenseBoard.

REFERENCES

1. Carswell, L., Thomas, P.G., Petre, M., Price, B., Richards, M. (1999) Understanding the 'Electronic' Student: Analysis of Functional requirements for Distributed Education.

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Journal of Asynchronous Learning Networks, vol. 3 issue 1 pp. 7-18 Sloan Consortium.

2.Bell, G., Dourish, P. (2007) Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. Personal and Ubiquitous Computing, Vol. 11, No. 2. (1 February 2007), pp. 133-143.

~~2.~~3.RoboFesta Europe (Britain) http://www.robofesta-europe.org/britain/

~~3.~~4.Price, B., Hirst, A., Johnson, J., Petre, M., Richards, M. (2002) Using robotics for teaching computing, science, and engineering at a distance. Proceedings of the 5th IASTED International Conference on Computers and Advanced Technology in Education (CATE), Cancun, Mexico pp. 154-159 IASTED.

~~4.~~5.Price, B., Richards, M., Petre, M., Hirst, A., Johnson, J. (2003) Developing Robotics e-teaching for Teamwork. The International Journal of Continuing Engineering Education and Lifelong Learning, vol. 13 issue 1-2 pp. 190-205 Inderscience.

~~5.~~6.Scratch at MIT.

http://scratch.mit.edu/

~~6.~~7. David J. Malan, Henry H. Leitner.(2007) Scratch for budding computer scientists, Proceedings of the 38th SIGCSE technical symposium on Computer science education, March 07-11, 2007, Covington, Kentucky, USA.

~~7.~~8.John H. Maloney , Kylie Peppler , Yasmin Kafai , Mitchel Resnick , Natalie Rusk, Programming by choice: urban youth learning programming with scratch, Proceedings of the 39th SIGCSE technical symposium on Computer science education, March 12-15, 2008, Portland, OR, USA.

~~8.~~9.Phidgets.

http://www.phidgets.com/

~~9.~~10.PicoBoard (ScratchBoard).

http://www.picocricket.com/picoboard.html

~~10.~~11.Open University OpenLearn.

http://openlearn.open.ac.uk/

Formatted: Font: 12 pt, No underline, Underline color: Auto, Font color: Auto

Formatted: Font: 12 pt, No underline, Font color: Auto

Formatted: Bullets and Numbering

Field Code Changed

Formatted: Font: 12 pt, Font color: Auto

Field Code Changed

Field Code Changed

Formatted: Font: 12 pt, No underline, Font color: Auto

Formatted: Font: 12 pt, Font color: Auto

Formatted: Font: 12 pt, Font color: Auto

Formatted: Font: 12 pt, No underline, Font color: Auto

Field Code Changed

Formatted: Font: 12 pt, Font color: Auto

Formatted: Font: 12 pt, Not Bold, Font color: Auto

Formatted: Font: 12 pt, Not Bold, Font color: Auto

Formatted: Font: 12 pt, Not Bold, Font color: Auto

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering