Starting with Ubicomp: Using the SenseBoard to Introduce Computing

Mike Richards
Centre for Research in Computing
The Open University
Milton Keynes, UK
+44 (0)1908 654023
m.richards@open.ac.uk

Marian Petre
Centre for Research in Computing
The Open University
Milton Keynes, UK
+44 (0) 1908 653373
m.petre@open.ac.uk

Arosha K. Bandara
Centre for Research in Computing
The Open University
Milton Keynes, UK
+44 (0) 1908 653545
a.k.bandara@open.ac.uk

ABSTRACT

In this paper, we describe a new undergraduate module for novice students conducted entirely through distance learning: *My Digital Life* (TU100). The module has been designed to lower the barriers to creating programs that interact with the world; TU100's materials have been designed to excite, encourage, reassure and support learners who explore the novel topic of ubiquitous computing through playful experimentation. It introduces the fundamentals of computing by giving students the capability for programming a device, the SenseBoard, which has built-in input/output and sensors. Programming is done in Sense, an extension of Scratch, which scaffolds programming and reduces the syntax burden. TU100 has taken inspiration from childhood learning and commercial product design to produce compelling, yet academically rigorous study materials.

Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]: *Computer science education, Curriculum, Literacy*; D.3.m PROGRAMMING LANGUAGES; D.2.6 [Programming Environments]: Interactive environments.

General Terms

Experimentation, Human Factors, Languages.

Keywords

Computer Science education, distance education, ubiquitous computing, SenseBoard, Sense, programming

1. MOTIVATION

The world of computing has changed: it is increasingly ubiquitous. Computers are becoming part of numerous manufactured objects that populate our everyday lives. Ten years ago there was no Facebook or Twitter. Fifteen years ago broadband access to the internet in the home was an unimaginable luxury. Just twenty years ago people still had to buy their books, music and feature films from a high street shop. The digital revolution might still be young, but it has arguably brought about the biggest change in our lifestyles in the last two hundred years. The ubiquitous computer of tomorrow promises, or perhaps threatens, to change the world beyond recognition.

Yet our students still often come to us with a vision of 'computing' limited to working with end-user applications such as Microsoft Office.

Our response is *My Digital Life* (TU100), an introductory module designed to align their understanding of the computing discipline with their experience of ubiquitous computing in the world.

TU100 makes powerful concepts and capabilities available to learners as their first academic experience of the discipline, in order to engage their attention while introducing fundamentals. Students are given hands-on experience of designing, building and programming the small, ubiquitous computers that will become increasingly common over the next decade. TU100 is designed in the tradition of efforts such as Alice [1, 4, 5] and Scratch [7, 8, 17], which prioritise hands-on demonstration of what computing can do, use scaffolding and libraries to give even beginners access to powerful programming elements, use programming to generate visible effects of interest to students, and hence captivate the imagination of learners in order to widen engagement in the discipline. TU100 considers the 'big picture', too, and gives students a conceptual framework for understanding and considering the profound technological, economic, political and ethical changes brought about by information technology.

The paper provides a broad introduction to TU100, outlining some of the challenges and experiences that shaped it, describing the key elements of the module, and summarizing early evidence of impact with students.

2. OVERVIEW OF *'MY DIGITAL LIFE'*

My Digital Life (TU100) is an undergraduate module for novice students conducted entirely through distance learning. (An introduction to the Open University and its model of distance education is given in the Appendix.) Beginning in October 2011, the module will be presented twice a year, with an anticipated cohort of some **5,250 students** per year. TU100 comprises 1/6 of the credits needed for an undergraduate degree. The workload is approximately 20 hours per week for a period of 30 weeks. Most OU students are part-time learners, and TU100 will be the only module they will be studying.

TU100 has been designed as a single point of entry for students wishing to complete a degree in computing or information technology, possibly in combination with another subject such as mathematics or business. Our intention was to provide a module for anyone who wishes to know about and use modern technology, whatever their own specialist subject.

3. CHALLENGES AND LESSONS

TU100 has been designed to address a number of explicit pedagogic challenges: improving the learning experience by reducing programming barriers, introducing fundamentals of ubicomp while also developing core academic skills, and realizing the OU philosophy through playful learning.

3.1 Reducing the obstacle of programming

Despite a number of differing approaches, previous introductory programming modules at the OU have not proved to be entirely satisfactory; a sizeable proportion of students either failed the related assessment, or were disinclined to continue to study programming. Many of these students would withdraw from studying computing entirely. Student feedback revealed a number of causes:

- Almost all conventional languages require students to passively learn a large amount of material before they can begin writing and understanding even the simplest of programs;
- Students with no background in programming were frustrated by the pedantic syntax and cryptic naming conventions used in written programming languages;
- Novices have trouble understanding program flow and were often unable to debug their applications, a problem made worse by inadequate debuggers;
- Students who had previous experience with, or a natural aptitude for, programming were able to complete all the programming activities but found them trivial;
- Our chosen language, JavaScript (along with most other small, relatively simply languages) is not suited to developing interesting, media-rich applications that will engage novices.

Of course, TU100 is not our first attempt to mitigate the programming barrier. Between 1999 and 2006 the OU ran a successful Level 2 module that introduced programming using Smalltalk. The module's approach to Smalltalk was called LearningWorks [21]: which incorporated a large collection of small activities that built on one another. For introductory activities, students directly manipulated software objects using their mouse, instead of typing commands. As students' confidence and skills grew, typed commands were introduced alongside the graphical interface, until students were ready to begin using a conventional programming environment.

The module's success in teaching programming rested substantially on this notion of building from one thing to the next, from immediate, direct manipulation to alternative forms of interaction, from concrete activities to programming concepts.

Lesson 1: Segue between skills.

3.2 Ubicomp and critical thinking

As part of its social remit to widen participation in education, the OU does not set *any* prerequisites or admissions requirements for study at level one; our admissions are 'open' to anyone who wants to learn. Although the majority of beginner students have some formal qualifications, many have none, and most have no background in their chosen subjects. Therefore OU level-one modules must include some fundamental skills development, especially in the areas of numeracy, literacy and academic skills.

Any introduction to computing and information technology risks overwhelming newcomers with a colossal amount of disparate information. TU100 overcomes this by using a strong narrative thread which runs throughout the module: *our relationship with digital technologies*. It begins by considering personal experience of familiar technology and ends by discussing how technologies are transforming our world. It takes students on a journey from the origins of information technology through to the familiar computers of today, and on to tomorrow's radical technologies. The aim is to introduce computing concepts while also developing students' intellectual skills.

The beginning of the module explores fundamental yet familiar technological concepts such as the PC, the mobile telephone and conventional broadcast media. At the same time, students are eased into patterns of study that they will continue to use throughout TU100; they take notes, summarise short pieces of text and contribute to online surveys (such as their use of digital media) and to online discussion groups.

The remainder of the first half of the module introduces the concept of ubiquitous computing in the context of the previously-explored technologies. Students first experience ubicomp by reading Weiser's genesis paper [19]; not only developing their research skills, but also (hopefully) experiencing the excitement of seeing a fundamental technological change described for the first time. Once they are familiar with the concepts of ubicomp, students explore the enabling technologies of sensors and location-based services in more detail.

Once students have a sound understanding of technological concepts, they move on to the social implications of new technologies. The emphasis in the second half of the module switches toward the core academic skills of fact-finding and argumentation. Once again the discussion begins with familiar technologies such as email and social media.

The latter part of the module has an explicitly activist approach. Students are encouraged to express their own opinions on a range of contemporary issues, using a range of resources to summarise the issue, present a point of view, or develop counter-arguments. Topics as diverse as the possible effects of video games on childhood development, the role of social media in politics and the impact of new technologies in the developing world are discussed. Amongst other activities, students are asked to take part in a debate on a topical issue (such as the role of copyright in the digital age), to measure their internet connection speed to help map the roll-out of broadband internet access across Britain and to research and publicise organisations overcoming digital exclusion around the world.

Lesson 2: Use a clear narrative thread to guide the learning journey.

3.3 Playful learning

The OU philosophy embraces life-long learning, which means striving to realize education that is fun, engaging, and fulfilling, as well as effective in conventional terms of academic achievement. Our philosophy of 'openness' to learners means re-assessing assumptions and expectations about students (who may have no previous qualifications, or many), designing learning experiences that are hands-on and accessible, and emphasizing relevance and play. The design of TU100 was informed by previous research and experience.

Our experience in robotics for children [14, 16] and in studies of what children do on their computers after school [12] provided useful insights, not least into the impact on learning of motivational technologies such as robots and online social networks. Children in informal contexts learn by tinkering: examining and modifying existing artefacts to make new variants. They learn by trying things out. When they engage with a new environment, children go straight to the examples; they don't bother with tutorials, if they can have a conversation instead.

As summarized in [12]: "The environments that appear to be successful with children are those offering useful instructive examples as a springboard to things they actually want to do; that provide immediacy of results and effects, that provide a forum for sharing and publishing successes; and that offer ‘room for growth’ by considerable progression beyond the basics to more advanced concepts and tools.” Computing is a routine part of play for contemporary children; our challenge was to make play a routine part of learning computing.

Lesson 3: Enable tinkering.

4. KEY INGREDIENTS

TU100 takes on board the challenges and lessons, and strives to harness the excitement of ubiquitous computing as a means to motivate students and give them a concrete awareness of the potential of the discipline, while introducing them to the fundamentals of computing. The module has been designed to lower the barriers to creating programs that interact with the world; the SenseBoard, our ‘lab in a box’, has been designed to facilitate playful experimentation.

This section describes the key ingredients of the module:

- the *SenseBoard*, a programmable hardware device designed for ubiquitous applications;
- *Sense*, a media-rich programming environment that extends Scratch for ubiquitous applications using the SenseBoard;
- the *study materials*, a structured collection of texts and learning activities designed for distance learning.

4.1 The SenseBoard

The SenseBoard is a tethered device based around the Arduino microcontroller. The concept of a programmable hardware device for novices is not new; the PicoBoard [13] was developed precisely for this purpose and similar microcontroller boards have been used successfully in teaching [15]. However, the PicoBoard has no outputs (LEDs or motors) limiting its usefulness for designing ubiquitous devices and it is relatively expensive in the UK. An alternative was to use Phidgets [6] or Arduino [2, 3] kits, which are widely used in ubiquitous applications and electronics courses. Their intimidating ‘breadboard’ appearance and the need to understand basic circuits and to wire in connections and components precluded their use with novice students.

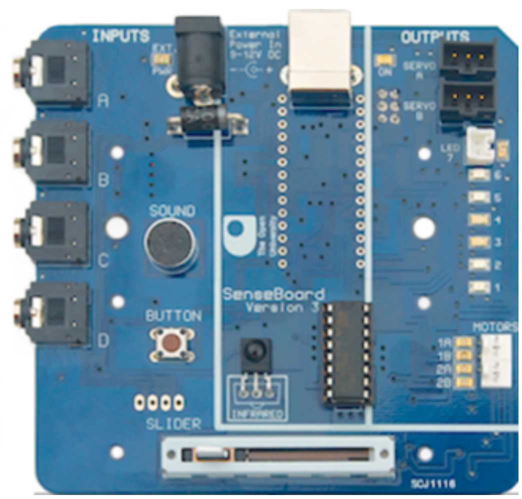

Figure 1: The SenseBoard

In contrast, the SenseBoard (Figure 1) provides a basic repertoire of ‘plug and play’ functionality. The design of the SenseBoard aimed to broadly replicate the functionality of the PicoBoard as well as adding output devices. The SenseBoard has a number of devices on board, including a slider, a pushbutton switch and a bank of 6 LEDs. It is supplied with the necessary USB cable, a set of plug-in sensors, a motor and a plug-in infrared LED.

The usefulness of an ‘experiment kit’ that allowed students to build their own ubicomp devices was unarguable, but it could only be justified if it would not place an undue financial burden on the university or the student. This necessitated developing a low-cost, yet powerful experiment kit. After several iterations of the design, the final cost of a boxed SenseBoard kit is approximately US$70 when ordered in bulk.

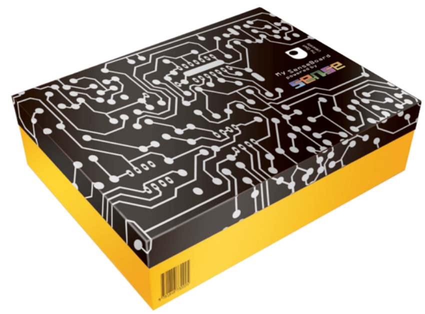

Figure 2: The SenseBoard box.

The SenseBoard is made more attractive by consumer-quality packaging (Figure 2). Early anxieties were expressed that students unfamiliar with electronic components would find the SenseBoard unappealing or even frightening. Students might be afraid of breaking delicate components, or of receiving an electric shock during handling of the board. Some concerns could be addressed in the design of the SenseBoard itself, by using oversized, robust connectors and ensuring a high quality of finish closer to that of a consumer product than a bare electronics board.

Further reassurance was provided by packaging the board in a manner used for toys rather than that used for electronic components. The kits are delivered in an especially designed, robust cardboard box that can be reused for storing and organizing projects. Every part of the kit is individually boxed in clearly labeled and illustrated packaging so they can be identified immediately as and when they are needed.

The kit is further supported with a dedicated website that includes step-by-step set-up guides, diagnostic software, frequently asked question pages, online videos demonstrating how to set up the kit and what it can achieve as well as a telephone technical helpdesk.

4.2 Sense

For TU100 we chose to adopt and extend the Scratch language [17]. Scratch is a media-rich programming environment notable for its clear programming structure. Individual program blocks (e.g. if-else statements, logical operators and variables) can only be assembled in meaningful (not necessarily correct) ways, and this removes some of the major frustrations experienced by previous students. Scratch:

- Explicitly shows the flow of programs;
- Replaces cryptic naming conventions with simple, obvious names;
- Largely removes the need for syntax;
- Offers immediate feedback;
- Supports media-rich applications.

Although Scratch is not a language in industrial use, it provides students with all the basic skills needed to succeed in most common programming languages – without having to worry about syntax. It also builds students' confidence in their own abilities by allowing them to develop systems quickly. Scratch has proved extremely popular with educators and students alike [7, 8]. Although Scratch was clearly designed for children, our user tests demonstrated that Scratch appeals to adults as well; we were surprised by their enthusiasm.

Nevertheless, Scratch was not ideal for our purposes:

- It lacked some of the richer programming concepts (such as string comparisons) required by the computing curriculum;
- Scratch exists within a closed sandbox environment without the networked support which was necessary for teaching ubiquitous computing.

Sense [18] addresses these needs by adding a number of features:

- New operators for string and list manipulation, and file i/o;
- Constructs for reading data from internet data sources (such as RSS feeds);
- Ability to write data to an internet data repository (a web service hosted by the OU) and read it back as an RSS feed;
- Ability to read inputs from Sense board and control outputs;
- Ability to include in-line comments in programs – allowing students to develop the skill of documenting their code.

Students engage with Sense through a large number of activities, most of which last less than half an hour. The activities are designed to reinforce the learning found in the other module materials, and demonstrate how smart devices and ubiquitous technology can be used in authentic situations. Activities are tightly constrained and come with sample solutions. Students can choose to complete an activity without assistance, or use one of a number of methods of support:

- Step-by-step instructions on completing the activity;
- Partial programs: complex or tedious parts of the program are provided;
- Exploded programs: the program is broken into individual blocks, the student must assemble them in the correct order;
- Sample solutions: the complete program can be examined;
- Screencasts: narrated video showing how to complete an activity.

Every activity is supplied with a completed solution. If a student has failed to complete the current activity for any reason (most commonly lack of time rather than lack of skill) they can begin the next activity using the provided solution, rather than spending yet more time trying to catch up with their peers. All activities come with suggestions for extension and improvement, which will guide interested students to develop their skills further.

Two example activities are the weather clock and the whereabouts clock. As well as recording and displaying data, the SenseBoard and motors can be used to create large, highly visible, physical displays illustrating aspects of ubiquitous computing.

The weather clock reads and displays weather forecast data from a public RSS feed. The program parses the feed for particular weather data and uses that information to display the appropriate weather symbol on the Sense Stage. At the same time the motor rotates an indicator hand to a matching symbol on a clock-face like display which the student will have constructed from paper.

A similar idea underlies the whereabouts clock, inspired by the Harry Potter books and work conducted at Microsoft Research Cambridge [20]. This project relies on a common feed shared by a group of students which holds the status of each student; (at work, online, unavailable, etc.). Students change their status inside Sense by clicking onscreen buttons, and their status is immediately updated on the feed. Each user has their own whereabouts clock which periodically polls the RSS feed and indicates each person's status using a moving pointer.

5. STUDY MATERIALS

TU100 is divided into 29 week-long 'parts'. Each part is comprised of a core teaching text (delivered in print or online) as well as associated learning activities. The majority of materials have been developed especially for TU100, encompassing printed texts, electronic documents, video and DVDs. Approximately half of TU100's study material is delivered electronically. Generally, material discussing fundamental concepts (such as the history of computing, or network technologies) which are less likely to date has been delivered as print items, whilst rapidly changing topics have been designed for electronic delivery.

5.1 Printed documents

TU100's printed materials have been designed to appeal to new students, with widely-spaced text with wide margins that allow students to add their own notes. The materials are full colour, printed on heavy stock and illustrated throughout, in six softbound volumes. Early parts of the module contain prominent signposts informing students how long they should expect to spend studying a section. These signposts are gradually reduced as students learn the essential skills of managing their own studies. Marginal icons are used at points where students require additional resources (such as their SenseBoard, the OU library or one of the module DVDs), to help students organize their studies.

High-quality printed materials are comparatively expensive and less flexible than electronic delivery; however our experience is that many newcomers are attracted and reassured by books. Printed texts have a higher perceived value than online materials. Most students keep their study materials after completing a module, both in order to continue to refer to them and to share them with other people.

5.2 Online material

TU100 students have access to a dedicated website based around an online calendar telling them what should be studied and when. The calendar links to every resource needed to complete the module including electronic copies of the teaching materials, the activities, videos and the online conferencing system.

Electronic copies of all teaching materials are available which can be read online through a web browser, or studied offline in HTML, PDF or ePub (electronic book) format. Amazon's Kindle format will be supported at a future date.

Students submit assessment materials as word-processed documents through an assignment portal from where they are collected by their tutor. The document is marked and commented inside a word processor before being returned to the same portal and the student notified.

Students are also expected to complete self-assessment activities on the TU100 website. These activities take the form of multiple choice, single-word answer, single-sentence or drag-and-drop exercises which provide immediate feedback about the student's progress. Electronic self-assessment activities are improving continually, with increasingly complex activities being developed across the university.

5.3 Online conferencing

The OU has been an enthusiastic user of online conferencing and is a major contributor to the Moodle discussion software. TU100 students have access to a number of discussion fora. First, they share a discussion group with a group of about 20 students moderated by their personal tutor which is used to coordinate activities that require students to share programs, constructively criticize one another's work and complete group activities building a large project.

Second, students have access to a technical support forum where they can ask for assistance with some aspect of the TU100 software from fellow students, tutors and members of the module team. The module team will run occasional online conferences; these chats not only attempt to overcome any perceived isolation experienced by distance learners, but also allow the module team to address any concerns raised by students without waiting for end-of-module feedback. Many distance students find it important to 'engage with the experts': to interact with the people who create the modules.

Finally, students have a so-called 'café conference' where they can talk among themselves about almost any issue.

As well as these asynchronous conferences, TU100 provides tutors with Blackboard Inc.'s Elluminate *Live!*® conferencing software which allows them to create synchronous conferences. Tutors may, if they choose, run short online tutorials exploring an aspect of the module, a Sense masterclass or a question-and-answer session. The module team has provided a list of sample Elluminate tutorials, but tutors are encouraged to create and share their own tutorials with one another. Synchronous tutorials are especially popular with tutors who have widely dispersed students, such as those in rural Scotland and those with a large number of students serving in the armed forces.

Although TU100 will not be making formal use of social media such as Facebook or Twitter, we fully expect students to follow their predecessors in creating self-supported 'study groups' to offer mutual support. Whilst these groups cannot be moderated by the OU, students are made aware of the consequences of sharing assessment solutions with one another.

5.4 Broadcast-quality video

The OU has a long history of educational programming which has continued for TU100. Specially-commissioned video supporting teaching materials gives OU students unique access to individuals, companies and locations that they could not obtain for themselves or in conventional universities. Video helps make explicit otherwise complex technical or philosophical materials and serves to break up large amounts of teaching text. TU100's videos include access to Microsoft Research laboratories, Google, an interview with Alan Kay, a fieldtrip to Nepal exploring issues of the digital divide, clean power generation for cloud computing in Iceland and an interview with one of the founders of Wikileaks. TU100 videos are available in low resolution and high definition over the internet and in conventional PAL resolution on the module DVDs.

5.5 Further learning resources

As well as the core TU100 materials supplied directly to students and those found on the TU100 website itself, the module uses a number of other resources.

5.5.1 Library resources

Students have remote access to the Open University library collection, and through it, to a wide range of journals, collections and citation lists. A number of activities in the module demonstrate how to use the library to find and download technical journal articles and to cite materials appropriately.

5.5.2 Technical support

If students encounter problems using Sense and the SenseBoard they are first guided to support pages found on the TU100 website which provide a set of diagnostic tests they can complete to identify and resolve issues. If these tests fail, students can telephone or email technical support staff who have been trained in setting up Sense and the SenseBoard.

In the event that technical support cannot resolve a software problem, the issue is referred to the module team so that a bug fix can be implemented. Faulty SenseBoards are replaced at no expense to the student and are returned to the board supplier so that any manufacturing defects can be identified and eliminated.

5.5.3 Employment resources

Rather than learning for the sake of it, the majority of OU students are learning to improve career prospects. In association with the OU employment group, TU100 provides two week-long online workshops where students can receive advice about improving their employability. Video interviews have been recorded with major employers including Google, Opera and CCP Interactive that give light-hearted, useful information about how to apply for technology jobs and what it is like to work in the field.

6. OPEN SOURCE LEARNING

The OU has been a strong proponent of sharing learning materials and technologies with individuals and other institutions. Sense is freely available as an open-source project. The SenseBoard specification is also freely available in the public domain. At present, the OU cannot supply SenseBoards to other users, but SenseBoards can be built either individually or mass manufactured without paying any licence fee to the OU.

TU100 video materials will be available for free reuse on iTunesU [10] and through YouTube [11]. Some TU100 materials are also available for free reuse through the OU's OpenLearn website [9].

7. DOES IT WORK?

We have tested the materials – and especially the SenseBoard and Sense – with a range of prospective users, from teenagers to retirees, from complete novices through hobbyists to computing professionals. So far, in terms of empowering novices to create interesting effects with low overheads, the outcome has been reliable: new users are able to produce a working program during their first session with the SenseBoard and Sense in under 20 minutes. They often start with the SenseBoard equivalent of 'Hello, world' – making the LEDs light up in a pretty pattern. And they don't stop with their first program. Novice and near-expert users alike tend to carry on tinkering. When they realize that the input from the sensors on their board can be aggregated with that from other students via the internet – wherever those students may be – they are struck by possibility: "Oh, so that means that I could…"

Early indications (less than one month into the first presentation) are that students have engaged enthusiastically with Sense and the SenseBoard. If anything, we underestimated their enthusiasm to begin using the novel technologies. Although we provided many introductory activities, it quickly became clear that many students were designing, creating and sharing their own projects. Before any Sense teaching had taken place, more than 200 projects had been announced on the student fora, many offered as downloads to other students. The most common projects were recreations of early arcade video games, perhaps reflecting the age of our students and the rich multimedia potential of Sense. Students requested and received advice about all aspects of game programming, including discussions of complex mathematics and how to increase performance.

A large number of student projects are under development. The most 'conventional' include video game controllers and simple weather stations, but other projects include a sound-controlled tea maker and a remote sensor for detecting paranormal activity. Each of these activities has been suggested by a student who has received encouragement and support from fellow students. We hope that this collaboration and community spirit continue.

We wanted to teach entry-level computing in a way that could compete with students' rich experiences of computing in the 'real world', e.g.: gaming, social networks, multi-media, web sites. So we mimic the characteristics of that *ad hoc* exposure: starting with 'cool ideas' that capture the imagination, reducing barriers, using commercial product design to package it compellingly. The result has attracted interest from educators at all levels, from policy-makers and from researchers. The typical reaction to the SenseBoard is: "Where can I get one?"

8. ACKNOWLEDGMENTS

We thank Garry Bulmer, Robert Seaton and Ian Cameron for their work developing the SenseBoard hardware and software. Our thanks also to the Scratch team at MIT for all of their encouragement and help in bringing Scratch's ethos to adult learners. Sense was developed for the OU by John Daniels and Dave Cleal of Syntropy Ltd. We also thank the hundreds of people around the world who gave their time during our interviews and location shoots, and the colleagues who contributed to TU100.

9. REFERENCES

1. *Alice*, Carnegie-Mellon University: http://www.alice.org/ (accessed 28 August 2011)
2. *Arduino*: http://www.arduino.cc (accessed 28 August 2011)
3. Buechley, L., Eisenberg, M., Catchen, J., Crockett, A. (2008) *The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education*. CHI 2008, ACM Press, 423-432.
4. Conway, M., Audia, S., Burnette, T., Cosgrove, D., Christiansen, K, Deline, R., Durbin, J., Gossweiler, R., Kogi, S., Long, C., Mallory, B., Miale, S., Monkaitis, K., Patten, J., Pierce, J., Schochet, J., Staak, D., Stearns, B., Stoakley, R., Sturgill, C., Viega, J., White, J., Williams, G., and Pausch, R. (2000) *Alice: Lessons learned from building a 3D system for novices.* CHI 2000, ACM Press, 486-493.
5. Cooper, S., Dann, W., Pausch, R. (2003) *Teaching objects-first in introductory computer science*. SIGCSE, ACM Press, 191-195.
6. Greenberg, S., Fitchett, C. (2001) *Phidgets: Easy development of physical interfaces through physical widgets*. User Interface Software and Technology (UIST'01), ACM Press, 209-218.
7. Malan, D.J., Leitner H.H. (2007) *Scratch for budding computer scientists.* SIGCSE'07, ACM Press, 223-227.
8. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010) *The Scratch programming language and environment.* ACM Transactions on Computing Education, 10 (4): 1-15.
9. *OpenLearn at the Open University*: http://openlearn.open.ac.uk/ (accessed 28 August 2011)
10. *The Open University on iTunes*: http://open.edu/itunes/ (accessed 28 August 2011)
11. *The Open University on YouTube*: http://www.youtube.com/user/TheOpenUniversity (accessed 28 August 2011)
12. Petre, M., Blackwell, A. (2007) *Children as unwitting end-user programmers*. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'07), 239-242.
13. *PicoBoard:* http://www.picocricket.com/picoboard.html (accessed 28 August 2011)
14. Price, B., Hirst, A., Johnson, J., Petre, M., Richards, M. (2002) *Using robotics for teaching computing, science, and engineering at a distance*. 5th IASTED International Conference on Computers and Advanced Technology in Education (CATE), 54-159.
15. Richards, M., Woodthorpe, J. (2009) *Introducing TU100 'My Digital Life': Ubiquitous computing in a distance learning environment*. UbiComp 2009.
16. *RoboFesta*: http://www.robofesta-europe.org/britain/ (accessed 28 August 2011)
17. *Scratch at MIT*: http://scratch.mit.edu (accessed 28 August 2011)
18. *Sense at the Open University*: http://sense.open.ac.uk/ (accessed 28 August 2011)
19. Weiser, M. (1991) *The computer for the 21st century*, Scientific American, 3, 94-104.
20. *Whereabouts clock, Microsoft Research Cambridge:* http://research.microsoft.com/en-us/groups/sds/whereaboutsclock.pdf (accessed 28 August 2011)
21. Woodman, M., Griffiths, R., Macgregor, M., Holland, S. (1999) *OU LearningWorks: A Customized Programming Environment for Smalltalk Modules*. ICSE'99, 638-641.

Appendix: Introduction to the OU

With more than 250,000 active students, the Open University (OU) is the UK's largest university. Originally founded in the late 1960s to exploit the possibilities of broadcasting to bring higher education to those who had been unable to attend conventional universities, the OU has continued to be at the forefront of technological developments in education.

The OU offers independently-audited qualifications ranging from introductory certificates, through bachelors, postgraduate degrees and doctorates, as well as a range of recognized professional certifications. The OU was conceived as a distance education institution, and all OU undergraduate students continue to study at a distance.

Most modules use printed and online self-study materials that have been designed especially for distance students. The development of new modules may take several years; with materials undergoing rigorous quality assurance procedures in terms of readability, consistency and accessibility to disabled students – as well as developmental testing with target users, and critical reading by external experts. Quality comes at a price; developing a module of this calibre may cost several million pounds.

Students are assessed throughout their period of study, via self-assessment online materials and by periodic formal assessment which is sent to a part-time member of the university staff for marking. Modules may include a final project, which requires the student to use a number of skills learned in the module.

Every OU student belongs to a 'tutorial group' that is run by a part-time, fully-trained 'Associate Lecturer'; who in turn is supported by regional 'Staff Tutors' and 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 modules may have in excess of 10,000 students during each presentation), whilst ensuring that individual students can receive personalized, professional support and engage in a student community.