Welcome to what is now the second year of iCS! As in the previous issues, our aim is to let you find out about areas of computer science that go beyond what you can learn at school, as well as the opportunities that you will have as a computer scientist to engage in all sorts of innovative and creative work.

In this third issue, you will meet Stephen Gorton, who graduated in Computer Science at Leicester, received an MSc in Advanced Software Engineering at Leicester, and is now finishing his PhD - also at Leicester! Stephen has recently been employed by ATX Technologies to work on one of the areas that are critical for all kinds of businesses to survive in the fierce markets of today: the re-engineering of old software systems so that they can be more agile and compete with more modern systems. You can read more about this topical area in the article "Even software can become obese", which covers research that is being developed at Leicester in a partnership with ATX sponsored by the European Union under the Marie-Curie programme.

Indeed, IT systems play an essential role in supporting the need that companies have to create new products and services that they can quickly launch into the markets. Therefore, there is a big demand for computer scientists that are creative and able to innovate. A good degree in Computer Science will give you these skills and make you highly desirable by leading companies and businesses in all areas.

At Leicester, we are involved in EU-funded projects that do research in re-engineering; this means that, as a student, you have immediate access to the new methods and technologies as they are developed, and thus gain a competitive advantage in relation to students in most other universities. For instance, ATX helps us teach some of the more advanced topics in software engineering. We will keep informing you about the opportunities that you have to work closely with industry during your studies (for instance by taking a year out on an industrial placement), or you can write to us with specific questions.

Another area of research at Leicester that is being pursued in collaboration with industry is in supporting collaborative work through new service architectures. The article "Taking the headache out of scheduling a meeting" will let you take a peek at the innovative techniques that are being developed by a consortium of European universities, organisations, and companies to support teams of people collaborating with each other even if they are not physically co-located. The group at Leicester involved in this project includes two researchers who, like Stephen, did the MSc in Advanced Software Engineering with us.

Our popular Graphics Column brings you excerpts of a Q&A session with Nick Burton of Rare Ltd (part of Microsoft Games Studios) held when he visited Leicester recently to talk about graphics in computer games to our BSc students.

Finally, don’t forget that you can write to us at ics@mcs.le.ac.uk if you have any questions about the topics of this issue of iCS, or if you would like advice about your future studies or your future profession.

This issue of iCS was again put together by Roy Crole. He and I would like to thank our guest writers and also the members of staff at Leicester who have helped in the production.
Dear iCS

Dear iCS. Should the degree programme I choose be accredited by the British Computer Society? What are the Engineering and Science Councils?

Firstly, you need to know what the British Computer Society is. Established in 1957, the British Computer Society (BCS) is the professional association (sometimes known as a professional body) for people working in IT. The BCS was incorporated by Royal Charter in 1984. It aims to promote the study and practice of computing and to advance knowledge of and education in IT. The BCS is also a registered charity. It has around 60,000 members in over 100 countries.

Why is the BCS relevant to you when you choose your degree programme?

In the UK, the Engineering Council (ECUK) and the Science Council are also professional bodies. The ECUK’s mission is to “set and maintain realistic and internationally recognised standards of professional competence and ethics for engineers, technologists and technicians, and to license competent institutions to promote and uphold the standards.” The Science Council says “Our purpose is to provide a collective voice for science and scientists and to maintain standards across all the scientific disciplines.”

The BCS is licensed by ECUK to award Chartered Engineer status (CEng) and Incorporated Engineer status (IEng) to individuals; and more recently by the Science Council to award Chartered Scientist status (CSci) to individuals. This recognises that the individual has both engineering and scientific knowledge to very good standards, such as that provided by quality degree programmes. How does this work in practice? The BCS awards “accreditation” status to good degree programmes that have been assessed by the BCS. Thus if you pass such a degree then you automatically qualify, as an individual, for one or more of CEng or CSci.

You may well find that there are very good degree programmes in the UK that are not accredited by the BCS, and there may well be very good reasons for this. For example, a brand new programme must run for a few years before the BCS will access it. However, if some programmes are accredited, then this is indicative of the standards set. So the overall message is: do look out for BCS accreditation, and use it as one of your guides in choosing a course, but in the end choose a course that you will find interesting and stimulating!

http://www.bcs.org/server.php?show=nav.7065

Taking the headache out of scheduling a meeting

by Stephan Reiff-Marganiec

Imagine a very common situation: a project in a large company needs to hold a meeting to make decisions on the next steps ahead. The meeting requires the attendance of the project’s key people – all busy visiting customers around the world – as well as people with specific skills (say, a web designer). In order to get in touch with all these people, find out about their availability, choose a convenient time when they can all attend or make themselves represented, and select the web designer, a secretary will certainly be busy for the best part of several days...

The situation described above is an instance of what is called “collaborative work” – an area in which computer scientists are working to develop innovative technology that can support the work of teams. The challenges are multiple: teams bridge company boundaries, its members move around, people use mobile devices and work on many things at the same time, and so on.

The EU-funded research project inContext (www.in-context.eu) is developing a platform and techniques that make use of service-oriented computing to integrate existing tools (such as email systems, calendars, project schedulers) into a coherent system that can be used on any device, anywhere in the world, to make collaborative work more productive.

So far, the project has concentrated on the development of a Pervasive Collaboration Service Architecture (PCSA) that allows users to connect from a PC, a mobile phone or a PDA to the system and request services. The system automatically decides which services to offer based on the context of the requesting user and others involved in the activity: where are they? what are they doing? what have they done in similar situations before? Making such decisions is not easy: it involves methods and techniques that support data mining, the gathering and modelling of context information, and reasoning about models in order to derive new facts. The automation of the decision-making process involves sophisticated algorithms and methods.

Many scientific results have by now been produced, and the viability of the PCSA has been demonstrated through a prototype meeting scheduler: when secretaries tell the system that a meeting is required, it will automatically collect names of people who can represent those that cannot attend, find experts in specific areas, and suggest alternative times for the meeting. It will even send invitations to people on the device that they use: email, instant messages or SMS to a mobile phone.

See us in YouTube: www.youtube.com/watch?v=QF4dYzMrcl.
**The Graphics Column**

by Gavin Cox

In October 2007, the University of Leicester Computing Society (www.ulcompsoc.org.uk) was host to Nick Burton of Rare Ltd (part of Microsoft Games Studios, www.rareware.com), the creators of games such as Donkey Kong Country, GoldenEye, Perfect Dark and Kameo. Nick gave a talk to BSc students in Computer Science on graphics in computer games. He focused in particular on how graphics have evolved over recent years, and how real-time techniques can be used to produce a realistic experience.

Nick spoke about some of the techniques used in the games industry such as the use of pixel, vertex and geometry shading, along with lighting and texturing of complex surface types (such as fur). He also covered more character-specific techniques such as skeletal systems and inverse kinematics. These effects are built using a combination of C++, DirectX and HLSL (High Level Shader Language), which is how you can program the shaders mentioned above.

Nick also explained how the current Xbox 360’s hardware design makes certain effects possible that previous generations of Xbox, GameCube and Playstation 2 just could not support. In some cases, such effects will still be out of reach of most PC games developers due to the wide range of hardware that appears in the PC market and the costs involved with ensuring compatibility.

Students were able to see the debug version of Kameo and were shown all the effects applied in a game to generate the finished result. Obviously, the effects can also be seen in the retail version of the game, but it makes it harder to jump around and effectively impossible to turn the effects off or switch to a free roaming camera.

One of the highlights of the talk was when Nick showed how the game used a series of sprites to generate the large armies in the games! When the player gets close to them, they switch from sprites to full 3-Dimensional models in order to produce a more realistic looking effect without overloading the vertex buffers.

**Question Time with Nick Burton**

This is part of a Q&A session between students and Nick after the talk.

**Q:** From a Games Programmer point of view, do you think that it is better to do a Computer Science Degree or one of the newer Games Programming Degrees?

**A:** Difficult to say because there are good and bad versions of both degree flavours! At Rare, we don’t favour one in particular, but we do find in general that we get better applicants from traditional degrees. This could just be because traditional courses are now well established and have stood the test of time.

**Q:** Are there any skills that are particularly relevant to jobs in the games industry?

**A:** It depends on the discipline, but a general rule of thumb is: good core skills can be used in our field, but field-specific skills are more vocational. So, for a programmer, mathematics, software engineering, and so on, should always be the foundation, and other stuff comes as a bonus. Same for art: good drawing skills, an understanding of shape, form composition and so on are the foundation; other stuff is then a bonus.

**Q:** What is the best way to get into the games industry?

**A:** Do a good degree and have a couple of really nice bits of example software you have written!

**Q:** How does console programming differ from PC programming?

**A:** This is not an issue, don’t worry about it; really, don’t! Xbox 360 = PC = PS3 = Wii and so on. Students always ask this question but don’t know why as it makes no difference whatsoever! The only minor difference is that writing code for any closed platform (like the 360) means that you do not have deal with the compatibility issues such as on the PC.

**Q:** What is the best thing about your job?

**A:** Too many things to choose from, really! Seeing your game in the shops or talked about on the web, of course, is one. But also getting to play with really powerful computing hardware, being in a truly creative environment, knowing that millions of people will have seen your work, and having a really nice campus/office with great benefits.

**Q:** Which game have you enjoyed working on the most?

**A:** Kameo was the best project to work on as we were really pushing the boundaries at the time with exciting new hardware. It’s also a great intellectual property, full of possibilities. Jetpac was also good though because it was a pet project of mine and the original version was one of the first games I ever played: being given the chance to remake it in the way I wanted was a dream come true.

**Q:** If you were a pinata, what kind of pinata would you be?

**A:** A sour Macaraccoon. Watch the TV series episode Mad Mongo to see why.

**Q:** Is Halo 4 going to be a launch title for the 720 and are Rare making it?

**A:** Now, that would be telling!
Even software can become obese
by José Fiadeiro

To many people, software is something that allows us to use a computer to perform certain activities (like writing a piece of text). However, software doesn’t just sit inside the computer: it has a life of its own and evolves over time. This is what happens, for instance, when we are asked if we want to download an updated version of the application that we have just launched. The fact that software needs to be continually adapted in order to deliver the same level of satisfaction to the user (or even increase it), is known as Lehman’s first law of software evolution.

Large organisations such as banks use very complex software applications and evolving them is a much more challenging task. If one is not careful, complexity increases as software is evolved. This is known as Lehman’s second law of software evolution. Very often, layers of software keep being added without restructuring what was there already, or new applications are coarsely stitched to old ones without taking into account the global structure or architecture of the system. In modern terms, we could say that software becomes “obese”: for instance, programmers are often afraid to delete code even if it is no longer executed, and code can get replicated all over the application in marginally different versions.

As it lets “fat” accumulate, software becomes less and less efficient, more and more difficult to change, and lacking the levels of agility, flexibility, and responsiveness that companies require to address the fierce competition and market volatility that characterises business today. As with humans, one can make a surgical operation to remove the fat. In software engineering, this requires a careful analysis of the code, breaking it into meaningful chunks so that one can understand what is “fat” and what is “muscle”, and reorganise what is left so that the original functionality is preserved. These rejuvenation techniques (re-engineering in computer science speak) are based on graphs and require sophisticated mathematical operations: it’s a job for specialists, supported by clever software tools!

Again as with humans, if we don’t change our life style, fat will accumulate again. One of the methods that has been gaining popularity for keeping software fit and agile is the adoption of an “architecture”, i.e. a way of organising software as a structure of interconnected (smaller) components and restricting changes on software to reconfigurations that conform or preserve that structure. In other words, evolution has to adhere to a (strict) “regimen”, which allows organisations to plan and optimise the use of resources, as well as control the quality of their systems.

At Leicester, we are doing research aimed precisely at re-engineering “obese” (aka “legacy”) software into what are called service-oriented architectures (a subject for a future article), and on ways of supporting evolution within those architectures.

Contact Details

If you would like to write to us with questions, comments or other matters concerning iCS, please do so at ics@mcs.le.ac.uk. It may not be possible to reply to all correspondence, but whenever possible we will do so. To find out more about the Department of Computer Science please visit www.cs.le.ac.uk

Editor: Roy L. Crole Compilation: Staff in the Department of Computer Science University of Leicester, University Road, Leicester, LE1 7RH, UK.