NEWS FROM COMPUTER SCIENCE AND ENGINEERING

Written by the Beta Team of CSESoc
Produced by Wen Di LIM
Edited by CSESoc EXEC,
and Samuel POLLOCK.

βETA

Free as in speech and our awesome BBQs.

Beta Issue # 81
Theta edition
Week 01 Session 2, 2013

OPINION

Importance of Software Engineering workshops, and why you should take them

The workshops allow for a structured development to occur in small groups over three semesters, requirements mining, formal proving, prototype development; this simulates a work environment for many students who have never had experience in scaled development. The close-knit teamwork environment, supervised by an older Software Engineering student, allows students to learn from each other as well as see the work and process involved in developing a system from scratch.

These workshops flow from vague client idea, sourced requirements development, formal proving, to development of a working prototype. This allows for many students to begin to understand workplace practises and to develop essential teamwork skills that many courses cannot facilitate.

The well-knit nature of SENG1031, SENG2011 and SENG2021 offers a complete roadmap of a systems development. This is useful in a startup, well-established company, or even to apply to other course work and personal work. These three courses are a very good chance to undergo structured development with mentors and lecturers who work very closely with each group, and should not be missed.

SENG1031, the first of the series, develops presentation skills, diagram construction and formal skills needed to construct a formal description of functional and non-functional requirements. Interaction between mentors and groups provide a regular stream of feedback that many courses lack.

SENG2011, utilises skills that are developed during COMP2111 for formal reasoning of systems, this is an intermediate step in the development process. Taking a requirements specification provided to the group, not necessarily from your SENG1031 project, and formally proving using invariants and constraints that a system could viably exist. Formal proofs of software systems are an essential step needed to bring Computer Science and Software Engineering into the STEM discipline.

"You don't see a Civil Engineer building a bridge and then patching it when it falls down; the Engineer proves it can exist first" - Ken Robinson

This important quote shows both the attitude of many programmers today as well as the distance that must be covered to bring Computer Science and Software Engineering closer to STEM practises. This is the most important aspect of the workshops that sadly does not get carried out by many.

CONTINUED ON NEXT PAGE.

ANNOUNCEMENT

Welcome Back!

Some people say another semester, another beginning. I say "Why were my winter holidays so short?" There is nothing for it, it is back to the grindstone. Another semester of fun, another semester of hard work - and may it be the best semester of all!

Welcome back all you fellow CSE people! Here in Beta, we hope that we'll be able to bring you another semester of news, entertainment and be the medium for your voice!

To start off the semester we have some opinions on Unra, the current SENG program, and part one of the Microsoft Tech talk. Crosswords and puzzlers are continuing, and of course there is always the rare overheard and jokes section.

Speaking of rare, as in yummy medium rare steak, Beta is always hungry for new articles. So if anyone has something that they would like to say, or an idea that they feel should be included in Beta, feel free to contact me at csesoc.beta.head@unsw.edu.au!

I wish you all a very awesome semester ahead, and good luck!

WEN DI LIM - BETA HEAD

OPINION, UNIVERSITY

Importance of Software Engineering workshops - contd.

When I undertook SENG2021, this was the final stage of the three step process. It takes the formally reasoned proof from SENG2011 and the requirements specification developed in SENG1031. Development is based on the translation of formal proofs to a programming language. The translation results in a system where there are no unexpected results and where every result is proved by invariants and sets of constraints on the system.

The SENG courses heavily train you in project management, diagram construction, and most importantly requirements gathering. The workshops train you as a Software Developer, Testing and as a project manager. There are very few courses where all of these are trained at the same time in a practical development.

Why you should take them now and not later

Reason 1
With the rapidly changing Software Engineering degree, these courses are quickly being changed to reflect the new organisation. SENG2021 has already changed and is no longer the third leg of the series. Current students are no longer taking their previous semesters work and developing their first prototype. I would advise asking current SENG2021 students what their impression of the course is.

The fostering of the agile development process in SENG2021, and the next SENG workshop, are an unneeded constant reiteration of a development process that results in code that may not work, without extensive patching. It's important to be familiar with more than just the agile development method, which has become a recent trend; the SENG courses traditionally sought to introduce a different development method to many students.

Reason 2
Many employers are looking for projects to talk about with you in interviews. If you're not particularly involved with open source, programming competitions, etc. then the development process involved in SENG courses is a very good talking point with many employers.

The more SENG courses you complete the more project development experience you can include on your resume. Past projects have included Transport ticketing, Ticket and Event Management, Warehouse and Inventory management, and Hotel management systems. These systems are complex enough to provide a challenge but not overly complex to make development difficult.

Reason 3
The workshop builds skills that you will find useful for developing a deeper understanding of concepts from other courses. Entity Relationship diagrams, State Transitions, Requirements gathering, are all helpful when undertaking core courses, especially when you're first being introduced to many of these concepts.

These concepts are learned and developed through interaction with the lecturers, your group and most of all, the mentor assigned to your group. The longer you put off learning in depth, the less you can use the skills in your university career.

Reason 4
The closely monitored nature of SENG workshops doesn't allow for a student to fall behind. SENG mentors are generally available around the clock for advice about anything, and more accessible and easy to talk to than your lecturers.

Mentors are from the cohort above, fresh from their own semester of the same workshop; they are the best source of advice you could have as a new student or even as a final year student for SENG advice.

As funding cuts have hit the Software Engineering department particularly hard, software engineering students have been particularly vocal about this, and mentors may slowly be phased out. So if you sign up for SENG now, there is a higher chance that you'll be getting a mentor that can guide you through a different and more structured development process.

MICHAEL CHAN

Opinion

UNRA is Not a Recursive Acronym

Reverse Computing Analogies

Hello World!
Pronounce it as you would “unrar”, and be sure to follow up with “not to be confused with unrar - the rar decompression program”. I intend to use this space to share some of my more amusing computer-thoughts. They may be in the form of short stories, philosophical ramblings, opinions or just facts that I think every eta-reading computing student ought to know.

Analogies are a useful tool for teaching foreign concepts, and they get used all the time in computing. Your Computing 1 tutor may have compared memory to a street with a row of houses to explain pointers. Some poor misguided (but annoying) soul probably tried to tell you how monads are like burritos (if anyone tries this on you, just walk away and find the post on A Neighborhood of Infinity titled “You could have invented monads”). We use analogies because people are usually quite familiar with rows of houses or burritos, but not so much with pointers or monads. But what about people who are familiar with computers?

Conversation Stack
Next time you talk to someone, pay attention to how the topic of conversation changes. Usually, somebody goes off on a tangent to a slightly different topic, it gets discussed for a while, and then the previous topic is resumed. One of the tangential topics could itself generate further new topics. To help you understand this concept, think of a conversation as a function call stack. Stack frames represent topics of conversation, function calls represent digressions, and function returns represent a topic ending and the parent topic resuming. When a function is called, a new frame is pushed onto the stack. When a function returns, the current frame is popped from the stack, and so the previous frame is now at the top of the stack again.

Face Hashing
Do you ever find that after meeting someone for the first time, you can’t recall their face? It’s not a big deal though, because even though you can’t picture them, next time you see them you’ll still be able to recognize them. Why bother memorizing a face? It‘s not like you expect to have to draw them from memory, so any time their face is useful to know is probably a subset of the times you can see them. Why bother storing all the data required to encode the image of someone’s face? It’s inefficient. Just compute the hash of the image of their face, and store that. Whenever you see someone you want to recognize, just take the hash of their face and compare it to all your stored face digests to see if it’s someone you know. So next time you feel like being casually racist, don’t say “All asians look the same”, but instead “My hashing function generates values with a high rate of collisions when given asian faces as input”.

Milk Fault
(Read the wikipedia article on “Paging”.)
You’re making a cup of coffee. You boil water, put coffee and sugar into a mug, and pour in the water, only to realize that you are out of milk. But worry not, for you live close to a 7/11, so you run to the store, buy some milk and come home to finish making your cup of coffee. Meanwhile, your computer goes to resolve a virtual address to a physical address, but oh no - the page containing that address is not currently in memory. A “page fault” occurs. The required page is copied from storage into memory, and execution can resume.

Stephen Sherrat

News in Brief

Teenage Mutant Ninja Burger
A Chinese man has been caught trying to smuggle a baby turtle disguised as a hamburger. The man, known only as Mr Li, was caught when his bag passed through an X-ray machine, and the attendants noticed what were described as “odd protrusions” coming out of the burger. Mr Li said he baked up this odd plan because he wanted to travel with his “beloved turtle”. He has since agreed to let the turtle stay with a friend.

In the Name of Love
A 15 year old girl has been arrested and marched out of a school camp in the US because a camp official had falsely accused her and a boy of sexually provocative behaviour. The girl and boy took time during a counsel-supervised activity at the camp called “court time” to sneak behind the arts and craft building to share a kiss, where she was seen and reported by another camp official. She is now suing for emotional distress and humiliation after being publicly humiliated and expelled the next morning. I guess the staff member can kiss her job goodbye.

Facebook Addiction of Another Kind
Alice Springs police are investigating a man who apparently used his Facebook account to conduct drug deals. The man’s Facebook page also contained photos in which he was showing off how much money he had made. Another photo also appeared to show bags of marijuana and he has often asked his friends if anyone was “looking for bags”. Many of his friends have told him he was a “dope” and have asked him to remove his account.

ChromeCasting Call
Google have recently released Chromecast, a new inexpensive device which allows the user to stream media from their phone, tablet or chromebook to the TV. About the size and shape of a USB drive, Chromecast plugs into the TV via a standard HDMI port. It costs just $35 USD and connects to your home WiFi system. Many pundits have noted the similarity between this and Apple TV, but only time will tell whether or not it is a bite of the same apple or something new.

Patrick Chung

CAREERS, TECH TALK

Tech Talk - Microsoft Engineer Mark Staveley

Part 1: What working at Microsoft is really like

Editor's Note: Last Wednesday, Microsoft Engineer Mark Staveley gave a tech talk about working at Microsoft and interviewing for positions at the top tech companies. The following are excerpts from the talk about life at Microsoft.

Some background on Mark
Mark has a degree in computer science. His research interest is mainly in Parallel Programming and he has worked on two of Canada's largest Supercomputers. He has completed a PhD and is currently working at Microsoft on the Xbox One team.

Mark started the talk by stressing the main message that he want to bring home is that when looking for a career you should find your passions. You should find a job that you would do for free. It means that you love what you are doing so much that you're also more likely to push the boundaries in that area.

Why work at Microsoft?
Microsoft is a very flexible space with many projects going on at any one time that are pushing the boundaries of what is technologically possible. For example, one of his projects involved combining the mathematical power of Matlab with Excel in an intuitive way.

What are some examples of Innovation at Microsoft?
Example #1 - Azura
When Mark started at Microsoft, Azura was in its infancy. At the time, the high performance team focused on "big machines" in the "big machine room". Mark's team changed the approach so that computing clusters could grow dynamically between local clusters and clusters in the cloud. At any time you decide the processing power of your local cluster is insufficient, you can easily connect to Azura and say "Hey, I want another 1000 machines", and that's it.

Example #2 - Xbox One Optimisation
The Xbox One has a virtualised Operating System. Mark's team was responsible for extracting the maximum amount of performance that they could possible get out of the system, given the restricted number of cores and memory on the system. Mark profiled various games, watching how the system reacted to various tasks, to peg down how efficient the system was. The profiling allowed tasks to be distributed across the cores in different ways to boost performance.

Example #3 - Windows 8 and Xbox One Virtulisation
Microsoft has taken technology from Azura, server tools and past operating system to build a three layer Operating System for WIndows 8 and Xbox One. There is now a "hyperviser" (or a virtualisation layer), which manages the application layer, which in turn manages the game system. Thus you are able to suspend applications rather than shutting them down. This makes loading times and switching between applications significantly faster. In comparison, the Xbox 360 has a much longer loading time between applications as everything is sequential.

Do you have memorable experiences?
Experience #1 - TOP500 Challenge
The TOP500 Challenge is a ranking of computers done worldwide every six months.

XBOX ONE

Two years ago Microsoft wanted to see how well they would do in the challenge and Mark was flown over to Tokyo. His team broke the Petaflop barrier at 1.19 Petaflops (1 Petaflop is one quadrillion floating point operations per second). Such speeds allowed their computer to complete tasks such as modelling Tsunamis before they happened.

Experience #2 - Halo 4
Studio 343 was having an issue rendering video for Halo 4 as their machines were not able to do all the calculations fast enough. Mark was brought in and he built a custom cluster for them. Studio 343 said thank you by organising a surprise visit from Master Chief during one of his meetings. When Master Chief asked for him personally, it was amazing moment. There was also a Warthog parked outside for people to take pictures with.

Some people think that Microsoft is a big and clunky company that isn't innovative. What do you think?
Microsoft is working very hard to remove this stereotype. Engineering management can get clunky but they've been working on ways to reduce it. For example in Mark's team there exists a "Night Court". If you see a feature that should really be fixed, you can come in at night and make a case before the division president/vice-president. So instead of just having the upper management handling and deciding everything, everyone can have a say. The input of engineers is highly valued.

What is research at Microsoft like?
Microsoft embraces research. There are startup teams at Microsoft which are essentially research teams. It is important, however, to find research areas that match your passions. Additionally Microsoft pays lot more compared to working as a research assistant.

How does Microsoft deal with the gaming industry competition?
The competition is a great motivation tool. For example, as a member of the Xbox One team, Mark wants to be proud of what he is working on. Mark and his team don't want Xbox One to be a piece of junk, so they strive to be the best engineers that they can be. That's the sort of attitude that Microsoft embraces.

CONTINUED ON THE NEXT PAGE.

CAREERS, TECH TALK

Tech Talk - Microsoft Engineer Mark Staveley - contd.

What sort of training do you get in the internship program?
Each intern is paired up with a mentor to help them learn the ropes at Microsoft. There are also courses, (sort of a mini university) that you can take to learn skills in areas such as programming standards, design patterns, and software quality assurance.

However, training takes time. As you learn you will have good days and bad days. When you're starting off, there may be some days where you just spin your wheels all day long and get nothing done. So you learn how to ask for help. But after a few weeks you start having more good days than bad days.

What sort of things can interns do at Microsoft?
Mark has an intern in her third year of university. His intern's project is about long haul based user simulation testing for the Xbox One. The purpose is to determine the stability of the Xbox One when switching between applications or games. She is working on building an automated testing system for this purpose and receiving the appropriate mentoring for it.

However there are certain visa restrictions for Australia and New Zealand, and hardware internships are just not possible at the moment. However, once you have worked at Microsoft it is much easier to transfer to other projects (under a revised visa).

How to Apply
To apply, either send an email to Veronica Finch (Vefinch@microsoft.com) or visit Microsoft.com/university. The website has lots of information about what the interview process is about, people's experiences and even a helpful guide on how to write your resume if you have yet to formulate one.

Microsoft is currently hiring interns for two main roles, Software Development Engineering and Software Development Engineer in Testing. All roles are in the headquarters, located in Redmond, Washington in the US. Be prepared to travel if you're applying to Microsoft as the Sydney office only houses non-technical roles such as marketing and sales. There are also some options in Europe. In addition to that, all travel expenses are fully reimbursed by Microsoft.

NB. This is part 1 of a two part article series on the tech talk given by Microsoft last Wednesday. Part 2 has interview tips and will be coming out in the next edition of Beta, so be sure to watch out for that!

THE BETA TEAM

COMPETITIONS

The Robocup Experience

RoboCup is an international robotics competition covering a vast array of challenges including soccer, rescue robotics, home assistance robots, and robots in the workplace. The Soccer leagues aim to develop robots good enough to beat the FIFA World Champions by 2050. UNSW has focused on the Standard Platform League, where standard robot hardware is used so that competing teams can focus on the ingenuity of their software solutions.

In this league the use of humanoid robots poses a significant challenge because of the inherently unstable nature of biped locomotion. It means the league has plenty of room for innovation to solve these difficult problems. Such problems range from computer vision to kinematics, high-level tactics to machine learning. Perhaps the most interesting (and also most frustrating) part of RoboCup is the rapid changes to the standardised robotic hardware used and the competition environment and rules. The idea behind these disruptive changes is to ensure the competing teams are constantly innovating such that the ultimate goal of beating the FIFA World Champion becomes more of a reality as the years go by.

The rUNSWift Team
This year at Eindhoven in The Netherlands "rUNSWift", placed 4th in the World. This year's core team included Beth Crane, Richard Hua, Jack Murray, Dan Padilha, Stephen Sherratt and myself, supported by Sean Harris and Alexander Whillas. There are no set roles per se, instead we work on the diversity of strengths of the team members to solve the problem at hand. These 'problems' could be calibrating the colour perception of the robot's cameras, optimizing the walking gait for speed and agility, or even the architectural design of the robotic software system.

Personal experience
To give you insight into what RoboCup @ UNSW is all about, allow me to share my perspective on this humbling experience. In 2012, I was in search of an interesting project to do. But as we all dearly love Brad Hall's daily email updates, I by sheer chance happened to read the one about doing RoboCup for thesis. At first the task seemed quite overwhelming because RoboCup was typically for more experienced 3rd or 4th years, but at that time I was still in 2nd year. But at the same time I was taking COMP3431 (Robotic Software Architecture) which gave me a broad context of the robotic world and that certainly helped me get a feel for the difficulties for the topics that lay ahead.

CONTINUED ON THE NEXT PAGE.

COMPETITIONS

The Robocup Experience - contd.

So my first go at RoboCup was to do a Special Project A working on simulating robot behaviours in a physics engine. I had to spend a few weeks analysing and deciding which topic I wanted to do and what my exact goal was. Then pretty much the rest of the semester was spent researching the ways to solve the missing gap that I was targeting. After that semester, I wanted to work more on RoboCup so I continued in 2013 by doing a Special Project B where I worked on computer vision. This gave me more dedicated time to RoboCup and it gave me the chance to dig deeper and work on something even more challenging.

The projects are inherently open-ended and challenging, so it is a far-cry from the coursework one would normally do at uni. As a consequence, I certainly had to learn whatever what needed to do the job: programming languages, computer vision algorithms, kinematics, etc.

But it's not all a one-man effort!

I was blessed to have an academic supervisor who had a lot of experience with RoboCup and mentoring the RoboCup team, so when times got a bit tough he and the team was always there for some guidance. To be honest, RoboCup is not for the faint-hearted. But as for all good things in life, it requires dedication in terms of time and effort. This year was certainly a humbling experience to work with the very dedicated and smart people of RoboCup and I highly recommend it to motivated people.

At the Competition
The competition is held in a giant hall or arena, with fields for each of the leagues and workstations for each team. The schedule is hectic; 2 or 3 'set up' days to calibrate the robots, fix any bugs and recover from jetlag; 3 days of pool play and knock outs; 1 day for the finals and the legendary SPL after party.

To ensure competitors get some sleep, the arena closes over night. Suddenly every hour between 8am and 11pm becomes even more crucial, thanks to the looming power-off deadline. The time is split between dealing with network issues, running robots to Aldebaran to fix broken parts, debugging or recoding skills, attending information sessions or referee meetings, and networking with other teams.

The arena is a buzz of activity, separate from the outside world. SPL has a number of fields, with teams stationed at tables around them. At any time of day you can see team members walking robots around the field, lying down coding, or running tests from the sideline. Posters decorate the walls around the arena, proudly displaying countries and affiliated institutes. Shirt swapping happens increasingly as the days wear on, with people competing to collect them all.

With honour and trophies at stake, emotions run high throughout the week. From up-and-coming newcomers carving out a name for themselves, to the stalwart been-around-from-the-start teams living up to their reputations (that's us!), everyone's in a frenzy to perfect and perform. Once competition starts tension is palpable, and the yells of success or frustration echo around the room.

The week is a blur of little sleep and constant problems demanding to be solved, and after the tensions of the finals fade away, the teams set to celebrating the efforts involved in full style (2013 featured a whole lot of free food and drinks, for instance).

Getting involved with RoboCup
The easiest way to start out with RoboCup is to talk to a team member or Brad Hall, and mention that youre curious. If you're interested there are actually several ways to join the RoboCup gig:

- Special Project A (COMP3901)
- Special Project B (COMP3902)
- Taste of Summer Research Scholarship
- Honours Thesis
- Volunteer in your spare time!

Typically the team members have joined 6-12 months before the competition in mid-year in order to familiarise with the robot hardware and the rUNSWift codebase. Some have returned multiple times because they like it that much! If you like robotics, self-directed research and a real tough challenge, RoboCup may be an interesting project for you.

CALVIN TAM, BETH CRANE

ENTERTAINMENT

Puzzlers

Congratulations to Malcolm Ryan for sending in his answers during the holidays, that shows dedication!

1. I am greater than God, and more evil than the devil. Poor people have me. Rich people want me. And if you eat me, you'll die. What am I?

2. In a two-child family, one child is a boy. What is the probability that the other child is a girl?
b) In a two-child family, the older child is a boy. What is the probability that the other child is a girl?

3. A woman shoots her husband. Then she holds him under water for over five minutes. Finally, she hangs him. But ten minutes later they both go out together and enjoy a wonderful dinner. How can this be?

Don't forget to email your solutions to:
beta.puzzles@cse.unsw.edu.au

THE BETA TEAM

INTERVIEW, UNIVERSITY

OpenLearning as a Learning platform

Adam Brimo, a CSE graduate, tells us more about his current project, OpenLearning.

What is OpenLearing?
OpenLearning is an online learning platform that is based around the principles of student autonomy, diversity of learning materials, openness of resources and social interactivity.

While the platform supports student grading, automatically marked quizzes/tests and content distribution around a wiki, it's the comment and collaboration system, and the subtle features to keep students motivated that makes it special. The key difference with OpenLearning is that we have a student centred view of learning while Blackboard/Moodle strongly focus on the management and administration of education.

Who can use OpenLearning?
OpenLearning is licensed on the software-as-a-service model, which is similar to GitHub. We enable individuals and organisations to teach open online courses (i.e. anyone in the world can join) for free while charging fees to organisations that run private courses. I think the business model strikes the right balance between promoting and supporting open education while allowing us to build a sustainable business.

How did OpenLearning start?
Associate Professor Richard Buckland and a number of students at CSE developed an early version of OpenLearning for use in one of his courses. The ideas and concepts were well received however the University did not have the skills or resources to continue the project. So we decided to form a company and develop a new version of OpenLearning that could be used for a variety of courses.

Just over a year ago we were able to attract some seed funding so a number of CSE graduates including John Garlands, Ricky Setiawan, David Collien and myself could work full-time on the project.

With a solid team in place we were able to accelerate the pace of development and launch with Australia's first MOOC, UNSW Computing 1 by Richard Buckland in October last year. Since then OpenLearning has been used to teach over 22,000 students worldwide in over 80 public and private courses from a range of organisations.

What platform does OpenLearning use?
OpenLearning is a Python/Django web application that uses MongoDB as its database and RabbitMQ/Celery for managing events. OpenLearning is hosted on Rackspace and uses a load balancer to route traffic to a cluster of web servers. We also use a number of services to monitor performance, errors and analytics so that we can respond to problems as they arise. We try to use as many off-the-shelf libraries and services as possible to speed up development as even simple problems may turn out to be complicated once you start implementing a solution.

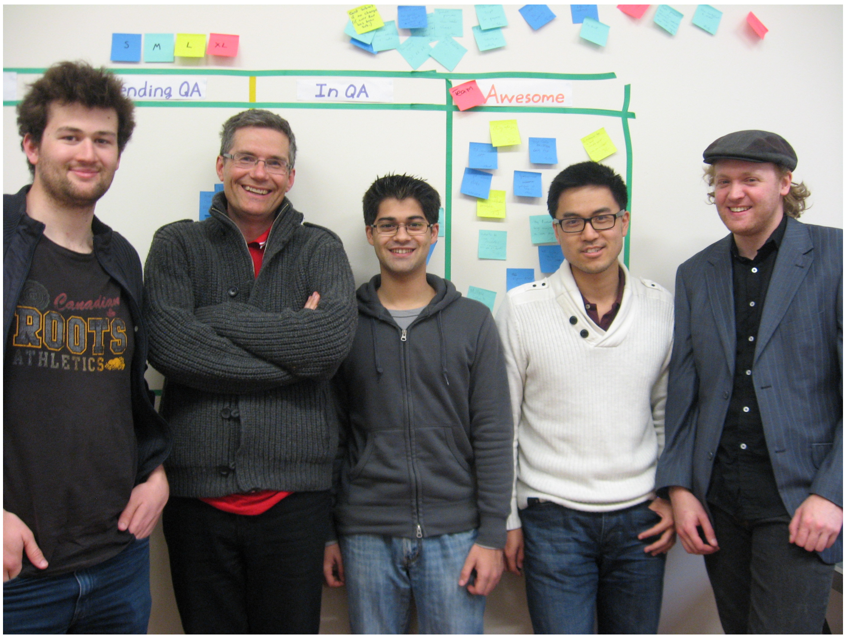
What do you hope to achieve with OpenLearning?
I hope to build a platform that students and teachers love, which is supported by a business that is sustainable in the long term. One of my key roles as a co-founder is to keep everyone focused and organized so that we can deliver the platform that the students and teachers want, when they need it. This is always a challenge with a small team and sometimes we have to make tradeoffs.

What can students expect from OpenLearning?
Students can expect that we'll be focused on building a learning environment that promotes collaboration and community within their courses. We won't be able to implement every feature but I always aim to explain our product and business decisions clearly.

Some students are worried about OpenLearning's policy about retaining IP. Could you elaborate on that?
We don't retain any IP from students, staff or lecturers who use OpenLearning. Our agreement with CSE ensures that students are protected by the UNSW IP policy for all of the content they create or submit for their courses on OpenLearning.

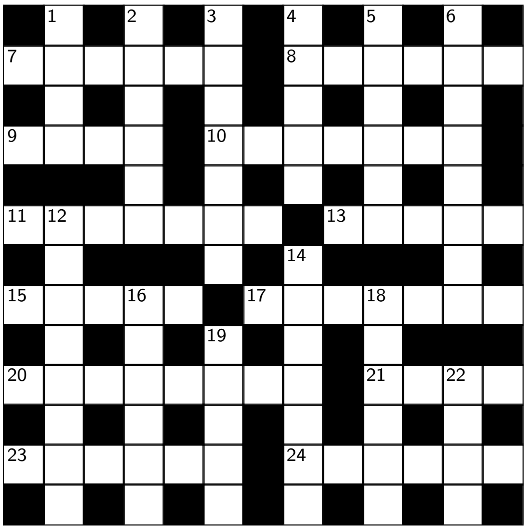
In short, we provide the platform, we don't own any content. Although UNSW courses are covered under an existing agreement, I'll review our public terms and conditions to make sure this is clear.

WEN DI LIM, ADAM BRIMO



ENTERTAINMENT

Crossword



Down

1. Flightless bird of New Zealand
2. Declare positively and strongly
3. Capital city of Algeria
4. Sudden involuntary muscle contractions
5. Tax haven nation bordering France
6. Large Canadian city, hosted the Olympics
12. First nation to officially adopt Christianity as state religion
14. Historical predecessor state to Germany
16. Pertaining to Norway
18. Central European nation, capital is Warsaw
19. A point of progress
22. Small nation on the Arabian Peninsula

Across

7. Name of iconic tower in French Capital
8. Logic programming language
9. Pacific island nation
10. A very short amount of time
11. Scandinavian nation, capital is Copenhagen
13. Asian peninsula, split into two nations, North and South
15. World's most populous nation
17. Capital city of Libya
20. Greek philosopher from Athens
21. Small landlocked Asian nation
23. A pillaging Scandinavian
24. Major ancient Greek tribe

OSWYN BRENT

CSE EVENTS & SOCIETIES

Upcoming Events

CSESoc Awesome BBQ
Tuesday, 12:30pm - 1:30pm
Physics Lawn
The usual BBQ is back! Looking forward to seeing you all there!

CSESoc Cardboard Night
Wednesday, August 7th, 6.00pm
k17 Seminar Room
$3/5 (Arc/Non-Arc) for Pizza and Drinks
Semester is back and kicking things off with a night of Card & Board Games, and as always, food and drinks! Join us to have loads of fun and meet new people!

Don't hesitate to bring any games you'd like to play on the night.

CSESoc PostGrad Drinks Night
Thursday, August 8th, 4:30pm
k17 Level 1 Kitchen
CSESoc Postgrad Drinks are on again next Thursday. Come and have some free afternoon tea whilst getting to know your fellow postgrad students. There will be a variety of snacks and soft drinks as well as some beer and wine for the more adventurous.

CSESoc LaserTag and Bowling
Week 3 (Exact date TBA), 5pm - late
Strike Entertainment Quarter
Missed those strikes and pew pews? CSESoc is going to Strike EQ again at week3 for more laser tag and bowling. $15 will get you unlimited laser tags, bowling and pool, so get your friends together and come shoot each other! OORAH!

Optiver Workshop: Importance of Speed in the System
Tuesday, August 13th, 4:30pm
k17 Seminar Room
High frequency trading is an exciting field where literally billions of dollars change hands every single day. One of the most important aspects of this is simulating real time communication. Speed in systems like this is of the utmost importance!

Optiver is an exciting key player in this field, and will be giving a presentation on the importance of speed in their systems and how it is achieved. Come along and learn, cakes will be provided!

VINCENT WONG - CSESOC SOCIAL HEAD

This Edition of Beta is Sponsored By...