floss UK

news@UK

The newsletter of FLOSS UK, the new name for the UK's oldest Open Systems User Group, UKUUG

Published electronically at `http://www.flossuk.org/Newsletter`

Volume 23, Number 2 **ISSN 0965-9412** **June 2014**

Contents

From the Secretariat

Jane Morrison

For those of you who couldn't make Brighton this year for the Spring conference we have now updated the web site with most of the speaker talks.

Check out **`http://www.flossuk.org/Events/Spring2014/Talks`** and see for yourself.

Most of the talks were also videoed – these will be able on the FLOSS UK web site very soon.

We are currently working on future events – see below – please put these in your diary now!

- Bar Camp – Saturday 7th June – Birmingham – more information on next pages
- Dynamic Language Conference – Saturday 13th September – Manchester
- UKUUG AGM – Thursday 25th September – London
- Un-conference – Saturday 25th October – London
- Intermediate and Advanced Perl tutorials - 11th, 12th, 13th and 14th November - London (see flyer enclosed)
- Spring 2015 – York – 24th, 25th and 26th March – call for papers available very soon!

Sponsoring Membership and sponsorship of events greatly assist us and allow us to keep event fees as low as possible.

Perhaps your company might be interested in Sponsoring Membership?

Silver Membership: Price: £1,000 p.a.

Benefits:

- Web link to Sponsors' web page
- Conferences: space or table-top at each conference including two personnel
- Insert/flyer with one newsletter each year
- Prestige of being a Sponsor Member of FLOSS UK

Gold Membership Price: £5,000 p.a.

Benefits:

- All of Silver Membership benefits, plus:
- Conferences: full access (up to 6 personnel) at any event
- space or table-top
- own brochure in delegate bags
- mention in conference publicity of involvement
- Insert or flyer with each newsletter
- Prestige of being a Gold a Sponsor Member of FLOSS UK

All AGM paperwork will be sent to all members in August, we may have two Council vacancies this year so if you are interested in joining Council let me know.

The next Newsletter will appear in September and has a copy date of 15th August. Any interesting articles from members will be very welcome – all submissions should be sent to: **`newsletter@ukuug.org`**.

Chairman's Report

Kimball Johnson

Events

It's been a quiet period since the Spring Conference, but we have more events planned, and some in the works.

Available for booking now is the ever popular Perl Training by David Cross, in November. Places are limited so book soon to avoid disappointment. We are working on other tutorials, more details will follow as they are available. In addition, in a few days there will be BarCamp Birmingham once again, on the 7th of June at The Studios on Cannon Street. If you are reading this in time, come along, please see the website for details and there will be a report of the event in the next Newsletter.

I would also like to encourage anyone with ideas for courses they would like to be run to email the office with their ideas.

Resignation of Chairman

It is with great regret that I have to announce my resignation from the post of FLOSS UK Chair. This will take effect from the Council Meeting prior to the AGM in September. I have enjoyed my time at the helm, but can no longer give the commitment that is required. I will remain on council for a time following the AGM, to provide a handover period for my replacement. I would like to take this opportunity to publicly thank the rest of Council, Jane and the other volunteers that keep FLOSS UK running.

Support for local user groups

The budget is still available for FLOSS UK to assist local user groups by helping them to obtain speakers for their events and assist with travel expenses. In addition we have a small budget for assisting with projects that would benefit the Free Software or Free Hardware communities in some way.

If you have an idea and wish some support, please contact Jane on `office@flossuk.org` and it will be discussed by Council.

Get Involved

FLOSS UK exists to serve its members and we are always on the lookout for people who are keen to get involved in any capacity, whether that be through volunteering to help with organising events, writing newsletter articles, or entirely new activities which haven't been tried before. If you would like to help out in any capacity please do get in touch via office@flossuk.org.

Birmingham Bar Camp

A Bar Camp is planned for Birmingham in June.

- *Date:* Saturday 7th June 2014 9am – 5pm
- *Venue:* The Studio Venue Company Ltd, 7 Cannon Street, Birmingham B2 5EP
- *Booking:* `http://barcampbirmingham.eventbrite.co.uk`

What is a bar camp? A bar camp is a conference where what happens is organised by the delegates on the day. The event organisers have to arrange something, the main one being the venue, but the rest is down to the delegates. So all the hassle of talk submissions, review and scheduling is taken away.

Typically At the start of the day everyone gets up in turn and says who they are, what their interests are and what they'd like to do. Based on this people write proposals on PostIt notes and stick these on a board.

A moderator may read out the proposals in turn to guage interest, and if sufficient the proposal will be put on a scheduling board. Delegates may adjust this to avoid clashes etc.

Experience shows that this format results in high quality sessions focussed on what delegates want.

There is no charge to attend. Everyone should be able to afford to attend and as it is a one-day event you will not need to incur accommodation costs, though there are several hotels nearby for anyone who wishes to stay over. Refreshments, tea, coffee and biscuits will be provided morning and afternoon, and lunch will also be provided.

Why attend?

There are lots of reasons to attend the FLOSS UK Bar Camp 2014 including:

- Keep abreast with new/emerging technologies
- Network with some of the people who are responsible for developing critical applications
- Become part of the UK Open Source community – build up informal relationships that can be invaluable in problem solving
- Benefit from the experience of delegates with similar interests

The 8 rules of BarCamp

- You DO talk about BarCamp
- You DO blog about BarCamp
- If you want to present, you must write your topic and name in a presentation slot
- Only three word intros
- As many presentations at a time as facilities allow for
- No pre-scheduled presentations, no tourist
- Presentations will go on as long as they have to or until they run into another presentation slot
- If this is your first BarCamp, you HAVE to present (OK, you don't really HAVE to but try to find someone to present with, or at least ask questions and be an interactive participant)

The LHS Bikeshed

Charles Yarnold

The LHS Bikeshed, for those who didn't fly with us in Brighton, is a 3 player theatrical gaming experience where your Captain and Control technician guide you on a mission to explore strange new worlds, seek out a training ground and not get killed horribly in the process (92% of teams don't make it home).

A labour of love by a plethora of members of the London hackspace, the bikeshed was first built as an item to take and show off at the UK Makerfaire held in Newcastle. As with all great ideas it was first formalised over a few beers and cemented when I took the plunge and bought an old caravan to be delivered to the hackspace.

We wanted to build the experience to take people on an adventure, but also to show the power of making and open source. Since its first outing we have taken the simulator to other mini maker faires, tech events, game dev conferences and restaurants on the seafront. (The sea breeze helped to add an extra element of movement to the sim.)

We have found that people are surprised then excited to find out that everything we make for the sim is free for anyone to download and use for their own projects. As of now we know of at least 2 other projects for people to make their own bikesheds based off our code base. Its great to educate people on the reason why its all free, experiencing the simulator is most of our audience's first interaction with opensource. We seem to leave people with a new found enthusiasm for it.

Now over 16 months on the simulator has evolved into a maze of network wires, leds, fans, relays and anything that comes to hand at the hackspace. We currently cram in 6 computers, 12 screens, 2 cheap android tablets and over 10 arduino like devices as of the last count.

But we can only reach a few people with the bikeshed due to the length of time and limited numbers that can play inside it at one time.

Thanks to FLOSS UK and their generous grant we are in the process of adding a brand new video camera system in. This consists of 5 720p network cameras places around the ship feeding back to a live video mixing system with the ability to broadcast locally to a TV outside of the ship, and/or to the internet for anyone to view. This has helped immensely with drawing in huge crowds to view the scifi TV show episode unfolding inside the caravan.

We are currently using snowmix to mix the live video and audio streams, snowmix is a project of Peter Maersk-Moller who is a member of the Copenhagen suborbitals group. It was created for use with their rocket missions, quite fitting for our use in technical terms as well as in spirit! It takes in gstreamer pipelines and allows you to control it via TCP and mix multiple video feeds, add overlays and images as well as text and much more. Then pipe it back out with gstreamer and onwards to anything you write. Its hugely flexible and we are only just starting to get started with it.

I also have whispers that snowmix and what we have started to build on it may be used to live stream and archive an upcoming multi stage conference. . . .

Coupled with snowmix is an android tablet that gives a control surface for the GM to direct the video system. This is written in processing for android, its proven to be quite a large draw for younger viewers, who doesn't like controlling their own TV show with lots of buttons! Top of our list to do once we are happy with the stable setup is to make a version to allow watchers to use and to show how easily android apps can be made with "gateway" languages like processing.

Our other future plans include an option for events in the game to automatically direct the camera angles to there the best action is, archiving of games to give to players as a souvenir of their flight along with a certificate of training. And many more things beside!

But FLOSS UK has kickstarted this whole new section of the experience so we can reach such a wider audience with what a bit of elbow grease and a whole lot of opensource can do. Thank you!

References:

`https://github.com/lhsbikeshed/`

`http://lhsbikeshed.tumblr.com/` (the fewer the updates, the more we are working on new stuff!)

`https://london.hackspace.org.uk/`

Hack to the Future

Les Pounder

We live in exciting times. The Internet, the great leveller provides us with everything that we need to learn, relax and work. Children born in the last 10 years take this level of access for granted via their hermetically sealed devices. A generation are consumers, visiting the app store for their latest "fix". But it wasn't always this way. In the 1980s children had access to computers such as ZX80, Commodore 64 and the BBC Micro and had access to magazines that provided code and instructions for them to follow.

With open source software such as Python and Scratch we are slowly seeing a resurgence in the number of children, and adults, learning to code again. But can we make computing fun for everyone?

The answer is a resounding "Yes" and here is how.

Physical Games

Young children love to explore and investigate, so why not work with them to build their very own adventure? Using their toys and a few props you can build an obstacle course in your home or school. This course has to be explored using a robot, namely parents, siblings, or friends. To enable your robot to explore the world, it has to be programmed and this is where you need a pen and paper. Robots (generally) do what they are told, so using your pen and paper write down some instructions to control the robot, for example:

```
forward 1 step
turn right
forward 1 step
turn left
```

These instructions are in plain English, but you can see a similar syntax to the turtle module in Python:

```
turtle.forward(1)
turtle.right(90)
turtle.forward(1)
turtle.left(90)
```

Create your own language and explore the room, try to keep the game fun and not to focus too much on a precise language.

Scratch

For older children, why not introduce them to Scratch and create an interactive story featuring their favourite characters? I've made a simple story that you can use as an example, you can get a copy from my github repo [1]. In my story there is a dragon and a knight, who are in the spooky woods. The dragon is a friendly chap and asks our name, he then burps and alerts the knight, who asks us if there is anything behind him? If we answer yes, then he turns around, sees the dragon and then runs away.

This simple piece of code covers a lot of core programming skills. We have events, such as clicking on the green flag and a broadcast. We have a conditional statement for the knight that is controlled via the answer to a question. But most of all this is a fun introduction to learning more about computing.

Robotics with a Raspberry Pi

For children of a secondary school age there is nothing more engaging than robotics, it really helps to visualise the coding concepts that we take for granted. But how easy is it to build a robot? Well to be truthful it is quite easy to build a simple robot using a Raspberry Pi and a board from Pimoroni called [2] Pibrella. Pibrella is a fantastically cheap board that can really enhance anyones learning. It comes with it's own Python and Scratch library to control the many inputs and outputs on the board. The board is also a simple motor control board capable of powering at least two motors or servos. I recently wrote some code [3] to control two motors for a simple robot using Python and Scratch. I also took a few pictures [4] of it in action and the build process. All of the kit for this robot can be bought from Pimoroni [5].

But is learning to code for everyone?

Well no it isn't. Some children will only get to a certain stage in the process and then either lose interest or lack the instinct to take it further. This is OK though; it's not for everyone and rather than force the child to learn coding, why not help them explore the other creative uses of computing? Creating graphics and audio/video using open source software would be a great idea. Inkscape, GIMP, Audacity and Openshot are all great tools for this task. You can create some excellent open source content quite quickly.

The most important thing to remember is that it should be fun to learn and experiment. Children learn via experimentation so give them the space and time to explore and hack at their own pace.

[1] **https://github.com/lesp/FLOSSUK_Scratch**

[2] **http://pibrella.com/**

[3] **https://github.com/lesp/Pibrella_Robot**

[4] **https://flic.kr/s/aHsjXkbfzt**

[5] **http://pimoroni.com/**

Building a Mendel90 Part 3 – Intro to OpenSCAD

Bob Clough

Ok, so we can print! Yay! Of course, what we need to do now is the ability to create and prepare our own models for the 3D printer. For this we need a CAD package to create a 3D model, and some Slicing software to convert the model to GCODE control script that runs the printer.

As a programmer, I far prefer writing code to create a 3D model than using a classic visual CAD program. My tool of choice is OpenSCAD[1], a free software '3D Compiler', which allows us to create solid objects, join objects together, and carve other objects away from them. This allows us to create all sorts of levels of model, from a simple coat hook, up to a machine like the Mendel90 itself. Indeed the Mendel90 is released as an OpenSCAD project on GitHub[2].

The OpenSCAD UI is fairly simplistic, with a text editor box, console window for compile status, and a preview window of your 3D model. Personally I prefer not to use the built-in editor, because it lacks some basic functionality like tabbing multiple selected lines. Luckily, the editor can be turned off, allowing the user to use Vim, Emacs, or a GUI text editor if you're a bit strange! In this mode, whenever the file is saved, the compilation will be re-run in preview mode, and the model updated.

So, on to usage. At the very basic level, say we want a cuboid, 20mm wide, 10mm long, and 10mm high. Create a file called cube.scad containing the following snippet, and open it in OpenSCAD. Our cuboid will be rendered to the preview window.

```
cube([20,10,10]);
```

Lets put a hole through a cylinder! We'll use some new commands: cylinder, difference, rotate, and the center keyword.

```
difference() {
        cylinder(r=10/2, h=10, center=true);
        rotate([90,0,0]) cylinder(r=3/2, h=11, center=true);
}
```

The first new thing here is the difference command. It takes the first object it is given, and removes any subsequent objects from it. These objects can be primitives, like cubes and spheres, or combinations thereof.

Next we come across the cylinder. A cylinder is a primitive object specified using a radius and a height. Since I prefer working in diameters, I normally specify the radii as a fraction, so a 10mm diameter cylinder has a radius of 10/2.

The cylinder is specified with `center=true` as an argument. This causes the cylinder to spawn with its center point at the 0 point, instead of its bottom corner at 0.

`rotate` takes an object, and rotates it around the 0 point. In the above code snippet I have rotated it 90 degrees around the X axis. Since it is in the center this rotates it in-place, and lays it on its side.

Woo, we have a cylinder with a hole through it in the preview window[3]! Pressing F6 will do a full compile of the object, and then we can export it to an STL file, a file used for interchange of 3D models between different software packages. The STL can then be sliced in the software of your choosing (I prefer Cura[4], which I will discuss in a future article).

Hopefully this article has given a useful first glance at the OpenSCAD language. Full documentation is available online[5], covering topic such as module declaration, mathematical functions, the 2D subsystem and more.

References:

[1] **`https://github.com/openscad/openscad`**

[2] **`https://github.com/nophead/Mendel90`**

[3] **`http://hacman.org.uk/?attachment_id=382`**

[4] **`https://github.com/daid/Cura`**

[5] **`http://en.wikibooks.org/wiki/OpenSCAD_User_Manual`**

Python Pocket Reference 5th Edition

Mark Lutz

O'Reilly Media
ISBN: 978-1-4493-5701-6
266pp.
$14.99
Published: February 2014

reviewed by Andrew Richards

This is a handy book to have around when Python programming if you're out of range of the 'net, want to travel light and don't get on with eBooks, but I've not been enamoured of it enough to switch from my existing Python reference book.

This is a companion volume to the author's *Learning Python* and *Programming Python*, both of which are absolutely huge books. By contrast, this will fit in a jacket pocket, although at 266 pages it is a bit on the thick side.

Considering that this is a small book, it covers Python in impressive depth, although the dense descriptions do sometimes take a bit of time to digest. Reasonably clear section and topic headings help locate the desired information quickly. It includes coverage of which features are specific to particular versions of Python. Three-quarters of the book focus on Python's syntax and language features; one quarter looks at key parts of the standard library; there are also a few pages on hints, tips and Python idioms. It includes a decent index, and the Table of Contents is also handy when looking for more general topic areas.

After using it for a while, I find myself still preferring to reach for a larger reference book (David Beazley's *Python Essential Reference*) which although not very portable provides additional information including more examples and a more accessible writing style. The smaller title is however more up-to-date, including discussion of Python 3.4. If I needed something particularly lightweight myself, I think I'd rather go for the Python reference documentation on my eReader in epub form from **`python.org`**: not as concisely written, but more comprehensive, and I can update it regularly to the version of Python I'm using.

Overall then, good for a specific use case (low-cost portable reference when out of Internet range if you don't get on with eBooks), but otherwise consider if other options could suit you better.

Speaking JavaScript

Axel Rauschmeyer
O'Reilly Media
ISBN: 978-1-4493-6503-5
460pp.
£ 30.99
Published: March 2014

reviewed by Lindsay Marshall

JavaScript is, of course, the new Fortran. It's kind of weird in places, it's ubiquitous and there are dozens and dozens of books about it, and people keep feeling that they need to write more of them. JavaScript probably has more quirks than Fortran, so as people keep pushing it harder and harder to do more and more, there is a need to have material that does address the fundamentals in a comprehensive way. This book certainly does that.

The author carefully goes through all aspects of the language from its history to its future and looks at all the statements and features in detail. This is all done clearly and carefully with lots of nice examples, counter-examples and warnings of pitfalls. However, this it is not bedtime reading, not a book to be read at sitting. I expect your brain might give up if you tried to read it like that. It's a book to dip into when you want to understand the specifics of a particular language feature rather one for learning about the language. Pretty much for experts and people wanting to become experts.

As always, much of this material is on the net but I haven't seen it all gathered together in one place as nicely as it is in this book. It is certainly a book that I will use in the future and I recommend it to anyone who is working seriously with JavaScript.

Client-Server Web Apps with JavaScript and Java

Casimir Saternos
O'Reilly Media
ISBN: 978-1-4493-6933-0
260pp.
£ 22.99
Published: April 2014

reviewed by Lindsay Marshall

If you were to ask me about building a client-server web app I definitely would not recommend starting with Java, but this is just personal prejudice. If you are a Java programmer and come from a Java only background this might be the book for you, though I do have to say that it might be more beneficial if you were to expand your horizons and work with a few other languages and environments.

This book will take you through a myriad of different approaches and frameworks and ways to build client-server applications. I have to say it I found it very confusing. I simply can't work out who this is intended for. It does seem to be for some kind of Java only programmer who has been living in a bubble for a lot of years without any access to hearing about how people build web systems. There's a strange five page introduction to JavaScript that tells you about functions and not very much about the object system. And there are a lot of things like that throughout the book. Snippets that start to go somewhere but then tail off. Frequently with a hard to type `bit.ly` URL (which is fine on the web but not good in text).

I just don't know about this book at all. It feels over simple and over detailed at the same time. Its like a lot of summaries pushed into a book – maybe its a website somewhere. I honestly don't know if this is a useful book or not. It isn't useful for me, but if you are a Java programmer who has been living in a bubble, you should probably take a look at it if you want to start working on these new fangled client-server systems.

Contributors

Bob Clough is a Founder Member of Hackspace Manchester (`hacman.org.uk`) and Manchester's 3D Printer User Group (`3dpug.co.uk`).

Kimball Johnson is a Developer at Shadowcat Systems in Lancaster. He has been programming since a very early age, starting with BBC Micros, then MS DOS and Windows Systems, however was enlightened with a copy of Debian GNU/Linux Woody at university. From this developed an affinity to Systems Administration, but like all good admins he is lazy and so tries to automate as much of his job as possible, and now follows the way of the Infrastructure as Code.

Lindsay Marshall developed the Newcastle Connection distributed UNIX software and created the first Internet cemetery. He is a Senior Lecturer in the School of Computing Science at the Newcastle University. He also runs the RISKS digest website and the Bifurcated Rivets weblog.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Les Pounder works closely with North Western Linux and Free Software groups to promote the use of Open Source software as opposed to proprietary software. He is also the organiser of UCubed, a free Linux and open source event in Manchester, and an organiser of Barcamp Blackpool and Blackpool Geekup, and has been head of crew at Oggcamp. He writes for Linux Format magazine and contributes to Linux podcasts including Fullcircle, Ubuntu UK Podcast and Linux Outlaws.

Andrew Richards is an IT consultant specialising in Linux and Unix based email servers, general sysadmin and networking. He dabbles in C and Python and has written a number of open source add-ons and patches to qmail. In his free time he enjoys arty foreign films, badminton and cycling. Find him at `www.acrconsulting.co.uk`

Charles Yarnold is one of the founding members of the London Hackspace and co-creator of the LHSBikeshed. A trained theatre technician, Charles has worked lighting and sound for hundreds of productions, and has recently made the leap to full time "maker" and builds weird and wonderful gadgets for people and compaies alike. He has also appeared on TV as one of Sky's Gadget Geeks where he helped to build all manner of wonderful creations.

Contacts

Kimball Johnson
UKUUG Chairman
Preston

Gavin Atkinson
Council member
York

Tim Fletcher
Council member
Manchester

Holger Kraus
Council member
Leicester

Ian Norton
Council member
Rochdale

Stephen Quinney
Council member
Edinburgh

Quentin Wright
UKUUG Treasurer
Warwick

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Cambridge

Jane Morrison
UKUUG Secretariat
The Manor House
Buntingford
Herts
SG9 9AB
Tel: 01763 273475
Fax: 01763 273255
office@ukuug.org