P7

Sudoku on the Raspberry Pi

At the end of this laboratory, you should have had a lot of practice implementing simple C++ classes and manipulating pointers.

You should have implemented a fully working Sudoku solver on the Raspberry Pi implemented as a Qt4 application that accepts an input problem into a displayed grid and solves it using Knuth's *Dancing Links* algorithm.

There is a very helpful presentation from Rob Beezer on this topic.

Do not let yourself get stuck trying to understand the ideas here; ask for help in the lab.

Preparation time	:	3 hours
Lab time	:	3 hours

Items provided

Tools	:	None
Components	:	None
Equipment	:	DVI-D capable monitor
Software	:	Updated Raspberry Pi boot image Win32 disk imager

Items to bring

- Essentials. A full list is available on the Laboratory website at https://secure.ecs.soton.ac.uk/notes/ellabs/databook/essentials/
- Raspberry Pi
- Raspberry Pi power supply
- Raspberry Pi SD card
- Raspberry Pi HDMI to DVI-D cable
- Raspberry Pi keyboard
- R Pi mouse

Before you come to the lab, it is essential that you read through this document and complete ***all*** of the preparation work in section 2. If possible, prepare for the lab with your usual lab partner. Only preparation which is recorded in your laboratory logbook will contribute towards your mark for this exercise. There is no objection to several students working together on preparation, as long as all understand the results of that work. Before starting your preparation, read through all sections of these notes so that you are fully aware of what you will have to do in the lab.

Academic Integrity – *If you undertake the preparation jointly with other students, it is important that you acknowledge this fact in your logbook. Similarly, you may want to use sources from the internet or books to help answer some of the questions. Again, record any sources in your logbook.*

Aims, Learning Outcomes and Outline

This laboratory exercise aims to:

- Develop confidence in the use of C++ pointers.
- Develop the ability to convert a formal algorithm into a working program.

Having successfully completed the lab, you will be able to:

- Build a computational algorithm into a working graphical algorithm.

To prepare this lab you will need to familiarise yourself with the *Dancing Links* algorithm for Sudoku, using Knuth's original paper and other internet sources.

1 Preparation

Find out about Sudoku and the Dancing Links algorithm. In particular, by using web and other sources:

- Ensure you clearly understand how the *Dancing Links* algorithm works.
- Design appropriate C++ classes for the algorithm implementation..
- Design a clear Model-View-Controller separation of the algorithm and display..

Record what you have learnt in you logbook.

2 Laboratory Work

2.1 Subsection 1

On your Raspberry Pi, build a Qt4 program capable of displaying and accepting input into a Sudoku board. Use a **`QPainter`** inside a **`paintEvent()`** to draw the necessary lines.
.

2.2 Subsection 2

Implement the *Dancing Links* algorithm so that your program will, on demand, solve the Sudoku challenge.

---

3 Optional Additional Work

Marks will only be awarded for this section if you have already completed all of Section 3 to an excellent standard and with excellent understanding.

Re-implement using Qt5 and the Wayland compositor.

Revision History

April 15 2013	Denis Nicole(dan)	First version of this lab created