



DRAG'N'DROP

RISC OS Pi and all RISC OS 5 machines

July 2015

Volume 6 Issue 4

£3.50

Overlord

New RISC game



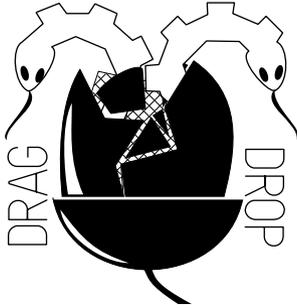
Programming Special

Python Primary School

Wimps Lua Postscript



Vol.1-6 index



Copyright © Drag 'N Drop 2015
Produced on RISC OS computers

This issue has been blessed with
contributions from the following people:

Anthony Bartram (Overlord)
Paul Dunnington (Python Primary School)
Jon Robinson (RISC OS programming)
Gavin Wraith (Spaghetti with Different
Sauces)
Christopher Dewhurst (everything else)

The views expressed in this magazine are
not necessarily those of the editor.

Alternative views are always welcome
and can be expressed by either writing an
article or a short editorial.

All articles and advertisements are
published in good faith. No materials in
this publication are meant to be
offensive or misleading. If you come
across something you believe is either of
the above please contact the editor using
the details below.

Contact Information
Editor: Christopher Dewhurst
Email: editor@dragdrop.co.uk
www.dragdrop.co.uk

EDITORIAL

Welcome to the summer
edition of *Drag 'N Drop*. You
could say that over the last 20
years the world has been
brainwashed into just pointing
and clicking on computers
running a certain American
operating system.

So it's great to see and hear
about the initiatives to
encourage children into
computer programming (at the
recent Recursion fair for
example) and the adults who
have have (re)discovered RISC
OS and are doing sterling work
in promoting the platform!

To that end we have an
extra large helping of
programming tutorials, as well
as an exclusive look inside the
exciting new RISC game,
Overlord.

Chris.

Christopher Dewhurst

At a glance...

Editorial 1

Beginner's Bit 2

News & Apps 3

Overlord 5

**Python Primary
School 9**

**Files of the World
(Postscript) 17**

Wimps 20

**Spaghetti with
Different Sauces 26**

C-Side 30

Index 32

Beginner's Bit



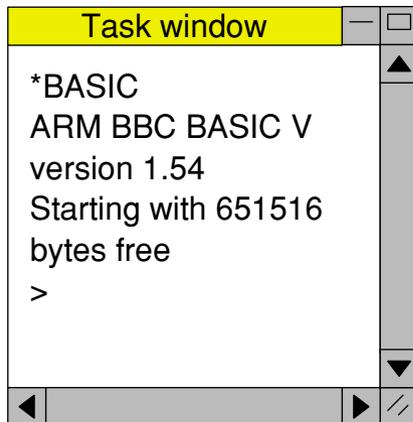
Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated



To get the BBC Basic prompt press F12 and type *BASIC and press Return. Type AUTO for automatic line numbering.

To return to the desktop type *QUIT.

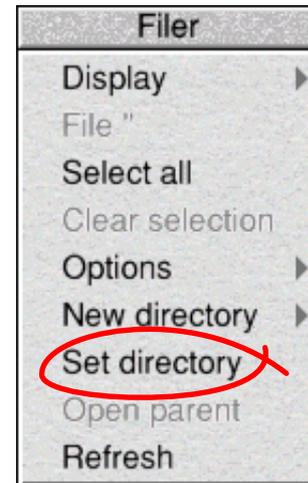
You can also type programs in a task window, hold down Ctrl and press F12. You can't use the cursor editing facility or change MODE, however.



Type SAVE "prog" when you are ready to save your program where "prog" is the program's name.



You can also program and run Basic programs from the desktop. Double-clicking on the filer icon runs it, holding down Shift and double clicking loads it into your text editor.



To set the current directory, click the filer menu and select "Set directory".

Or use the EasyCSD application listed in the Autumn 2014 issue.

News and Applications

Stratford Summer Show

The second annual Recursion computer fair took place in sunny (and very hot) Stratford-upon-Avon on 4th July, which was a celebration of British computing with dozens of Raspberry Pis being shown off by local and national developers, schools and hobbyists running various operating systems - including RISC OS of course!

recursioncomputerfair.co.uk
Stratford-upon-Avon Sat 4th July
dragdrop.co.uk



New Game



A new game called Overlord has been released, a shoot-em-up, 3-D space game packed with exciting graphics, challenging missions, music and sound effects. Price is £9.99 from !Store any time or on CD from Ambiguous Contrasts at RISC OS shows. Exclusive to readers of Drag 'N Drop we take a peek inside the game with its author, Tony Bartram, on page ..

!Store is pronounced "plingstore" a shopping application which should already be installed in your Raspberry Pi in the Apps directory. If you don't have a copy, however, go to www.plingstore.org.uk/cgi-bin/plingstore/download.cgi

BBC Microbit

Billed as the most ambitious education initiative since the BBC Micro, the Microbit (as it has been initially dubbed) is a tiny device

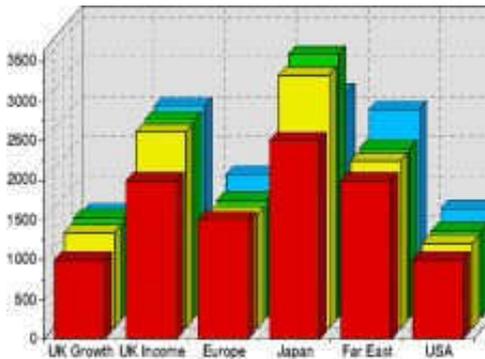
with an LED display which can be plugged into a computer like the Raspberry Pi and is designed to encourage young children into coding for computers.



You can read more at www.bbc.co.uk/mediacentre/mediapacks/makeitdigital/microbit

Chartwell 2015

Chartwell is business application suite enabling you to quickly create high quality graphs and charts with data imported from spreadsheets etc. and the resultant Draw file can be imported into DTP programs. Price is £9.99 from !Store.



Rhapsody 4 and PMS

R-Comp continue with their sterling work of refurbishing software to run on modern RISC OS hardware. PMS version 3.61 is available free from !Store.

Can You Feel The Love Tor

Words by Tim Rice
Music by Elton John



If you want to produce top quality sheet music on your RISC OS machine we have the perfect accompaniment in the form of the series which ran in Drag 'N Drop from 4i2 to 5i1, all on the back

issues memory stick available to buy from www.dragdrop.co.uk

Another musical scoring package from the 1990s is making a comeback: Rhapsody 4 is being updated and also will be available in !Store; details in the next issue we hope!

RISC OS 5.22

Version 5.22 of the OS was released by RISC OS Open in April for the Lyonix, Armini series, Panda and older Acorn hardware like the RiscPC. It is said that 5.22 was not stable enough on the Raspberry Pi and owners will have to wait until RISC OS 5.24 comes out to upgrade. However, for enthusiasts wishing to try out developmental versions an 'inbetween' RISC OS (version 5.23) is available from www.riscosopen.org/content/downloads/raspberry-pi.

Budget computer

The Ident-1 is a low cost machine based around the Raspberry Pi being developed by Ident Radio & Television. It's styled on the old BBC Micro so all you will need is

a screen to plug into. Prices and availability expected to be announced soon at www.ident-online.co.uk.

New Book

There is a lack of beginners books on BBC Basic published recently but that's all about to change thanks to an exciting new volume being written by Tom Williamson (incidentally also developing the Ident-1 computer), aimed at every version of BBC Basic from the BBC computer to the Raspberry Pi. Price and availability to be announced soon.



Writing Games for RISC

Over ten months I wrote a game called *Overlord* which was released in April 2015. In this article I'll outline how the game developed and what was required to make *Overlord* run on different contemporary RISC OS hardware.

Back in the '90s when I was half my age demos and games were definitely cool. When I wasn't working on university assignments, socialising or composing music late at night, I was coding games. Computers were fun, friendly and interactive.

As the computing scene grew into a larger commercial affair some of its innocence and fun was lost. Gone was the tinkering, novel and idiosyncratic behaviour of computers. Welcome the grey square soulless, conformist box.

Then the Raspberry Pi sparked my interest and I discovered RISC OS. I found BBC Basic ran faster than on my ageing BBC Micro. There were sprite APIs, MP3 playback was available, all

easily accessible from BASIC. I felt this was hugely powerful.

"Why aren't people writing games for RISC OS", I thought? It looks ideal for demos and games.

It turned out that there were differences in modern hardware that needed to be accommodated by the code.

Also there was no single source of reference showing how to create a game for contemporary RISC OS systems.



Game Design

A popular technique used nowadays in professional software development is Rapid Prototyping.

That simply means repeatedly and quickly producing a working

prototype. There were 137 versions of *Overlord* produced and 8 versions that were put through beta testing.

Initially production-quality sound, music and graphics were not used. But the basic form of the game itself was tested i.e. a ship flying through a star field forever that could shoot aliens.

This kept it really simple and quick to produce prototypes and thus made it easy to quickly develop a proof of concept.

Then over time, play testers fed back comments and ideas were raised about what the game needed. What was dull and what was interesting was assessed. More elements were added and some things were removed or changed.

Chance Versus Skill

There is a psychological element to game design. The brain releases reward chemicals when a goal is attained. We are creatures of curiosity that want to



I put the numbers at the beginning of the filenames so they are at the top of the filer window and easy to find. Go back to the \$.Programming.Python directory and double click on !Python.

A task window appears and after a short while the Python treble chevron prompt appears.

You can resize the task window in the normal RISC OS way by dragging its bottom right corner.

```
Python 2.4.1 (#65, Aug 29 2005,
21:59:19) [C] on riscos
Type "help", "copyright",
"credits" or "license" for more
information.
>>>
```

Start by typing

```
license()
```

and press the Return key which gives a short history of the Python language and various other

I am a total beginner with the Python language and have only just got round to serious experimentation with the help of the *Programming the Raspberry Pi: Getting Started with Python* book by Simon Monk.

The URLs of where to download Python (version 2.4) for RISC OS are given at the end of this article. There's quite a bit of setting up to do before we can even write our first line of Python so pay attention class !

Open the Programming directory on your hard disc.

Create a new directory inside called Python. Open the zip you downloaded and drag the !Python directory into it. Copying may take some time, there are a lot of files.

Open the !Python directory by holding down a shift key and double click with the select button. Open the Lib directory.

To help me keep my programs in order I created two new directories inside Lib, one called **0Getting_Started** for the files from Simon Monk's book and one named **1Mine** where I am keeping my efforts.

Files of the World (2)

Postscript (or PS for short) is a powerful computer language which was developed in the 1980s, originally for the publishing industry to print high quality documents.



You don't actually have to have a printer to program in PostScript. All you need is the Edit to write the programs, Ghostscript and !PrintPDF, which on the Raspberry Pi come ready installed in the Printing directory.

!PrintPDF uses Ghostscript to compile the Postscript program to a PDF file for viewing and you can use !PrintPDF's export options to convert it to a Draw file.

In this installment I'll give a quick overview of the Postscript language with a few simple programs. If you want to go into greater depth I can recommend the *Postscript Language Tutorial and Cookbook*, also known as the "Blue Book". It's well written and easy to follow, available for free download at www-cdf.fnal.gov/offline/PostScript/BLUEBOOK.PDF



TUTORIAL
and
COOKBOOK

Load Edit so its icon is sitting on the icon bar. Click menu over Edit's icon and Create > PoScript. This opens up an Edit window with the file type pre-set to &ff5. The first line of a postscript file must always be:

```
%!PS
```

Which tells the computer that the following code is Postscript. Apart from this first line, everything after a percentage symbol (%) are ignored, like Basic's REM statement.

We're going to draw a line. In Postscript drawing consists of three parts: you tell it the various coordinates the line goes through, optionally set the line attributes (colour, width, fill) then tell Postscript to actually draw the line.

Coordinates and line width are measured in points, 72 to the inch. Here is a program to draw a line five inches long:

```
%!PS
%start a new path
newpath
%move to start of line
72 72 moveto
% line 360 points (5 in) long
72 432 lineto
%Draw a line through the points
stroke
```

Save the program somewhere suitable on your hard disc. Call it



RISC OS Programming

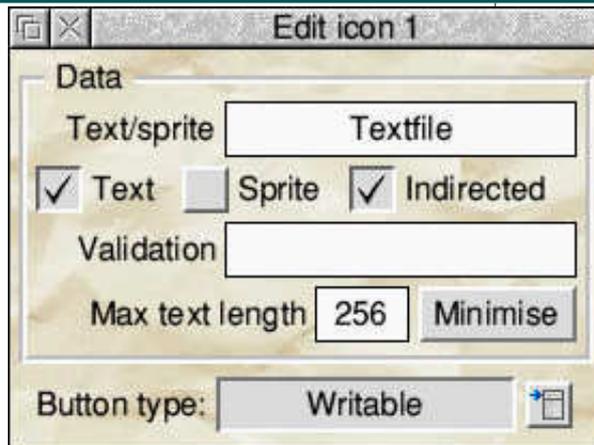
Part 7 – Determining the File Path

In the last instalment of this tutorial, we managed to read the window handle of the window where we dropped the file icon into the program. But we haven't managed so far to convert it into a proper file path, that can be used to save the output file.

Before we go any further, we need to do a bit of tidying up, to avoid getting into trouble later on. When the Templates file was originally created, the length of the writeable text was left at 8 characters, which will cause problems if the user tries to create a longer file name, or the program tries to update the text icon with the full file path.

To get around this, load the Templates file, inside the application, into !WinEd.

Double-click on the small Save icon, then double-click on the Textfile icon in the Save as window. Where it says Max text length in the Edit icon 1 window



change this to 256, and click on the Update button at the bottom.

Now go back to the main window, and save Templates.

If we go back and run the main program now, you will see that dragging the icon onto the icon bar produces a window handle of -2.

There's also the possibility that the user could pick up the file icon, then drop it again, without taking it out of the Save box.

We need to test for these two conditions, to make sure that the program doesn't do anything, if

the user drops the file icon on the icon bar, or inside the Save window.

If you haven't got the !RunImage file from last time it's listed at the end of this article with all the additions we will make in this instalment.

We need to expand PROC FileIconDropped, by PROCReport line with:

```
590IF DestinationHandle%<>-2 AND  
DestinationHandle%<>SaveHandle%  
HEN  
600 VDU 7  
610 ENDIF
```

(Stage7a)

Running this now produces a beep, when we drop the file icon on an ordinary window, but NOT on the icon bar, or back in the Save window.

Reading the icon

We can now expand this a bit by

```

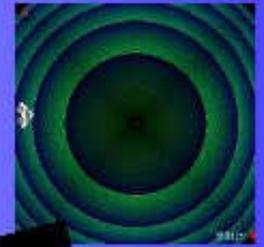
1980ENDPROC
1990
2000DEF PROCShowIconMenu(Menu%, x
%, y%)
2010SYS "Wimp_CreateMenu",, Menu%
, x%, y%
2020ENDPROC
2030
2040DEF PROCcloseIconClicked
2050REM> Called from main loop wh
en EventCode%=3
2060SYS"Wimp_CloseWindow",,Block%
2070ENDPROC
2080

```

Overlord

Only
£9.99
in iStore

The resistance have sent you on a mission to slip through the Overlord's defences and reach their home planets. Shoot asteroids to get shield energy, collect powerups and follow additional mission objectives.



Remember...your heroism brings hope to all oppressed by Overlord tyranny and that one day we will all be free...



Ambiguous
contrasts
Games

RISC OS 4&5 on Raspberry Pi 1&2
Pandaboard & Beagleboard systems
(vonix, ArmX6, VA SA RPCEmu.

www.amcoo-games.co.uk

Have YOU written a program for your RISC OS Pi?



Can YOU write an article to describe it?

Get £15 to appear in Drag 'N Drop!

dragdrop@dragdrop.co.uk

Spaghetti with Different Sauces

Each programming language has its own advantages and disadvantages. So different problems may require for their best solution the use of a different programming language.

Unfortunately, learning one requires an investment of time and mental energy, so many of the few who do write their own programs tend to stick with what they already know, regardless of whether it is the right tool for the job in hand.

This is particularly true of BASIC, because that is what most people learn first. A result of this is that because BASIC was devised a long time ago, lots of people get stuck in a time warp without exposure to more recent programming concepts.

This is a bee in my bonnet and one way of exercising it is to issue programming challenges: *can you write a program in BASIC to solve the following problem? or can you write a better program than the*

following, in any programming language, to do the same job?

Of course, much may depend on what better means: shorter, more readable, more efficient, and so on. These challenges have been the stimulus for lots of articles that I have written. This one started out that way, but ended up a bit differently.

Here is a small example of the first kind of challenge.

- Write a program that scans a text for ip addresses (four groups of four hexadecimal numbers separated by dots) and reports the frequencies with which they occur, in order of decreasing frequency. Useful for analysing router security logs, maybe?

Program 1 is a seven line solution in pure Lua (and so is platform independent). It presumes that the text is in a file whose pathname is the first commandline argument.

I type my Lua programs in

StrongEd in Lua mode and in Mode Choices selecting Font > Bitmap > Script, plus LineNos > Physical. If you are using any other text editor, don't type the line numbers.

```
1 local ip_pat =
  "(%xxxxxx%,%xxxxxx%,
  %xxxxxx%,%xxxxxx%)"
2 local L, f = { } --
  iplist, frequency
3 for line in io.lines
  (arg[1]) do
4   line:gsub (ip_pat,
  function (ip) if f[ip]
  then f[ip] += 1
5   else L[1 + #L] = ip;
  f[ip] = 1 end end )
  end -- if function) for
6 table.sort (L, function
  (i, j) return f[i] > f[j]
  end)
7 for _, a in ipairs (L) do
  print (a, f[a]) end -- for
```

The pattern **ip_pat** has a single capture, denoted by the parentheses. The symbol **%x** matches hexadecimal digits. The function **line:gsub** scans line for



Last time at the C-side we set up "GCC C", the free application for C programming. In this instalment we look at getting going with the commercial C package available from RISC OS Open.

It comes on the Nut Pi SD card, **AcornC/C++** is the folder you need to copy into a suitable place on your hard disc – into the Programming folder would be a sensible choice.

You may see AcornC/C++ referred to as the "Desktop Development Environment" (DDE) or even "Norcroft C compiler"; they are all the same thing.

Compared to GCC there's

thankfully very little else in the way of installation you need to do.

When you open the **Programming.AcornC/C++** folder you may see a message flash up "32 bit libraries installed". If you don't double click on **!SetPaths** inside AcornC/C++. The purpose of this is to let the computer know about the shared libraries – to use last time's analogy where C programs go to borrow bits of built-in code to save reinventing the wheel.

Press Ctrl+F12 to open a task window. The main 'star' command to invoke the compiler is ***cc** so at the asterisk prompt type two lower case C's and press return. This gives a list of help beginning with

```
Norcroft RISC OS ARM C vsn 5.70  
[12 Oct 2012]
```

```
Usage: cc [options] file1 file2  
... fileN
```

As we said in the last instalment, there are rules about where you must keep your C programs. Make **Programming** the Currently Selected Directory

(CSD) by menuing over the filer window and choosing Set directory. Or use the !EasyCSD application in issue 6i1 of *Drag 'N Drop*.

Create a subdirectory inside Programming for example called **c**. That's a directory with one letter, a lower case C. Also create another subdirectory called **o**. Save the Hello World source program as **c.hellow**.

The program is exactly the same as last time and the listing is repeated below. Make sure you include a blank line at the end.

```
/* c.hellow.  
  
The simple Hello World applica  
tion. */  
  
#include <stdio.h>  
  
int main (void)  
{  
    printf ("Hello World\n");  
    return 0;  
}
```

There's no "C filetype" so programs can be saved as ordinary text files.

In your task window type

Drag 'N Drop Volumes 1-6 Index

This index is organised with a reference (e.g. application name) or article/listing/series title followed by the volume – issue number – page.

- 21st Century Computing (article) 4i2-5
- 24-bit colour 5i2-47
- 256 colours 4i2-10, 4i3-9, 5i2-47

- 55 BBC Micro Books 4i3-21, 4i4-17
- 76489 chip 1i2-22
- 7-segment LCD 5i3-6
- 8-bit roundup – New Releases from Retro Software (article) 3i3-19
- 8-bit roundup (article) 1i1-29
- 8-bit roundup (article) 1i2-22
- 8-bit roundup (article) 2i1-19
- 8-bit Roundup: Retro Software Roundup 3i4-15
- 8-bit: BBC ARM7TDI Upgrade 1i1-29
- 8-bit: BBC GoMMC (article) 2i1-23
- 8-bit: BBC RAM Disc (article) 2i1-19
- 8-bit: BBC Repairing your power supply 1i1-30
- 8-bit: BBC Retro Software Update (article) 2i1-25
- 8-bit: Game On (article) 2i4-17
- 8-bit: Game On 2 (article) 3i1-9
- 8-bit: Look Who?s Calling 2i2-19
- 8-bit: Repton – The Lost Realms 2i3-19
- 8-bit: Speak your mind (article) 1i2-22

- A simple thumbnailer (article) 6i1-47
- A Trip Round the Chocolate Factory (article) 1i3-20
- Acorn Electron World (article) 4i2-27
- Acorn speech system 1i2-22
- Acorn World in Pictures 1i1-20
- Adding document information to PDF files 3i4-6
- All Sorted 1 (article) 1i1-9
- All Sorted 2 (article) 1i1-22
- All Sorted 3 (article) 1i2-7
- All Sorted 4 (article) 1i3-10
- All Sorted 5 (article) 1i4-11
- All Sorted 6 – the Merge Sort (article) 2i1-4
- All Sorted 7 – Heaps of Data (article) 2i2-7
- All Sorted 8 – Quicker Sorting (article) 2i3-12
- All Sorted 9 – Strands & the Merge sort revisited (article) 2i4-6
- All Sorted 10 – Hacks & Tricks (article) 3i2-10
- All Sorted 11 – Hacks & Tricks (article) 3i3-9
- All Sorted 12 – Mashing it Together (article) 3i4-10
- All Sorted 13 – Timsort 4i4-33
- AltTab 0.42 1i1-3
- Anatomy of a Drawfile (article) 4i4-29, 5i2-36, 5i3-27
- APDL 1i3-2
- ArcSimp 1i1-3
- ARM7TDMI Coprocessor 1i1-29
- ARMini computer 2i3-2
- ARMini computer 3i3-3
- Armcode bit pattern demo (listing) 5i2-32
- ArmSort 4.08 3i4-4
- Artworks Viewer 2.14b 1i4-3
- awk 2i3-25

- Backdrops (Desktop Heroes article) 1i3-18
- BarFree 3i3-5
- Basalt 1.50 3i4-4
- Basic V for the Acorn Archimedes (book review) 6i3-6
- BBC Micro Games Retrospective (articles) 4i3-26, 4i4-11, 5i2-61
- BBC Playground 4i4-10
- BBC Microbit 6i4-3
- BBFlash 0.03 3i4-4
- Beagleboard 1i1-15
- Beagleboard-XM 1i4-2
- Beeblt 0.61 2i4-4
- Beginner's Tips 5i1-2, 5i2-2
- BgrndCtrl 1i2-5
- Birmingham 2009 1i2-20
- Bitcoin (article) 5i3-44, 6i1-36
- Bookmaker 2.15 3i2-3
- Bookmarking with PrintPDF (article) 1i4-35
- bubble sort 1i1-9

- C programming see C-Side
- C-Side (series) 6i1-61, 6i4-30
- CSV2DTA listing 6i3-22
- Calendars with Yahoo & Organizer (article) 2i1-32
- Calibra 3.23 3i2-3