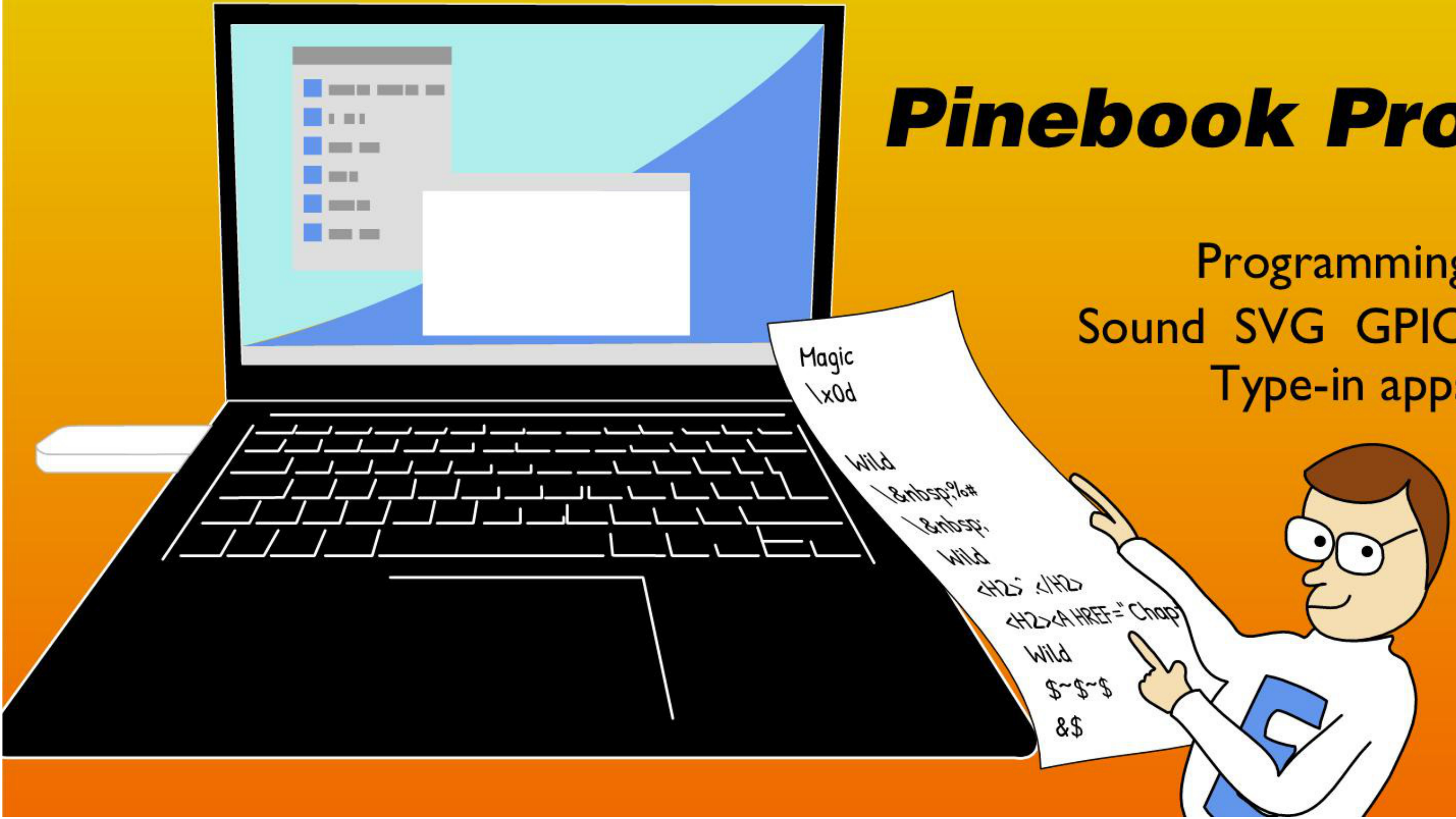
DRAG'NDROP

Volume 12 Issue 4
Summer 2023
£4.25

RISC OS**Pi** and all RISC OS 5 machines



Pinebook Pro

Programming
Sound SVG GPIO
Type-in apps

DRAG
’N
DROP

Produced on RISC OS computers

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

Norman Lawrence (PineBook Pro)
Richard Baraniak (GPIO)
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

Contents

Welcome to the summer edition of *Drag ’N Drop*. I hope you’re enjoying the pleasant weather now that polar jet stream is moving back home. The Midlands Mega Meet virtual show has been and gone and I am hopeful the London group will find a new venue so the in-person October show can go ahead.

In this issue we take a look at the PineBook Pro. If you’re on a budget look at the PiHard range of RISC OS fast machines. For the good old Raspberry Pi see the article on connecting and programming devices. EdiScript is a really useful application to find and replace a list of items in text files.

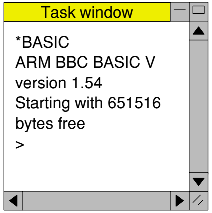
Chris Dewhurst, Editor

Entering listings

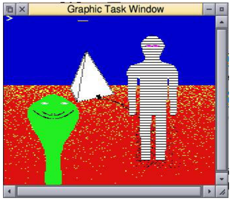
You can type in programs in three ways:
1. Use a text editor like Edit. Press the Menu mouse button over Edit on the icon bar and choose Create>BASIC. For other languages type in at the bottom. Save it by pressing F3.

2. Press F12 and type BASIC and press return. Change the screen mode with MODE n where n is a number e.g. MODE 7.
Type AUTO for automatic line numbering. Press Escape to stop and type SAVE "myprog" followed by Return to store *myprog* on hard disc. Type RUN to execute the program. To return to the desktop type *QUIT. Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated.

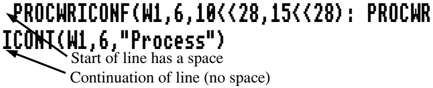
3. You can also type programs in a *task window*, Menu over the Raspberry icon on the right side of the icon bar and choose Tasl window, or press Ctrl+F12. You won't be able to change screen mode or use the cursor keys, though. However, GraphTask from armclub.org.uk/free/ allows you to type in and run Basic programs that use simple graphics (not sprites) in a window on the desktop.



```
*BASIC
ARM BBC BASIC V
version 1.54
Starting with 651516
bytes free
>
```



Program lines aren't numbered. Each line starts with space so you know when to press Return. To find out the line number you are on press f5 in Edit.

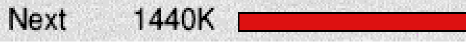


```
 PROCWRICONF(W1,6,10<<28,15<<28): PROCWR
ICONT(W1,6,"Process")
```

To run Basic programs from the desktop, double-clicking with select on the filer icon runs it. Holding down Shift and double clicking loads it back into !Edit.

More memory for applications

You may need to reserve more memory for a program. Adjust-click on the Raspberry icon and under *Application tasks* click and drag the *Next* slide bar out to the right.



What does 'currently selected directory' mean?

Articles may tell you to set the CSD (currently selected directory). Click menu over filer window and choose *Set directory ^W*. It's where the computer stores the file when you type SAVE "myprog".

How do I open an Applcation Directory?

Application directories begin with a ! called 'pling', for example 'pling boot'. To open (without running) it, hold down the shift key and double click select to open the directory.

Blank screen when running games listings

Check you have the Anymode module installed, download it from www.pistar.co.uk/anymode. It goes in !Boot.Choices.Boot. Predesk.

Open the !Boot application directory, in the root directory of the SD Card, that is SDFS::RISCOSPi. $.!Boot. Locate the *Loader* file and with Shift held down double click it to open it. Create a text file in Edit with the following line (press Return at the end):

```
disable_mode_changes
```

Save it inside Loader as CMDLINE/ TXT and restart your machine.

‘Screen mode not available’

Check you have Anymode (see above). The other issue may be Aemulor which can interfere with screen modes. Menu over iconbar > Quit > Emulator too.

Sounds are strange

Some listings need the free RDSP module installed. Download it from www.amcog-games.co.uk/rdsp.htm where you’ll find instructions on how to install it.

WIMP library

Many programs in *Drag ’N Drop* are multi-tasking (running in a window on the desktop). They use a set of standard procedures to create and deal with windows, icons and menus.

Rather than publish them with every listing they are collected here. Most of them are taken from *The Application Tutorial and Listings Book* available from Drag ’N Drop Publications. If you’re interested in writing desktop applications then you should consider buying this book.

```
DEF FNMKWINDOW
READ $T,X,Y,W,H
FOR I=0 TO 84 STEP 4
READ A$
I!B=EVALA$
NEXT
T+= LEN $T+1
SYS "Wimp_CreateWindow",,B TO X
=X
```

Basic’s DATA pointer is assumed to be at a line giving window title, position, size, colours, flags comes before this is called. Memory blocks T and B must have been set up. Makes a window returning handle in X.

```
DEF PROCMKICON(H,X,Y,W,D,F,A$,V)
$U=A$ : RESTORE +1
DATA H,X,Y,X+W,Y+D,F, U,V,LEN A$+1
FOR I=0 TO 32 STEP 4
READ B$ : I!B = EVAL B$
NEXT : U+= LEN A$+1
SYS "Wimp_CreateIcon",,B TO I
ENDPROC
```

Make an icon, handle is returned in I. H=window handle, X,Y=bottom left, W,D=dimensions, F=flags, A$=text and V validation string (1 if none)

```
DEF PROCRDICON(W,H)
!B=W : B!4=H
SYS "Wimp_GetIconState",,B
A$=$(B!28) : X = B!24
ENDPROC
```

Read icon W in window H text in A$ and flags in X.

```
DEF PROCWRICONT(W,H,B$)
PROCRDICON(W,H)
B!8=0 : B!12=0
$(B!28)=B$
SYS "Wimp_SetIconState",,B
ENDPROC
```

Updates text B$ in icon H in window W.

```
 DEF PROCWRICONF(W,H,X,Y)
 !B=W : B!4=H : B!8=X : B!12=Y
 SYS "Wimp_SetIconState",,B
 ENDPROC
```

Updates icon W's flag in Window H, X is the EOR word and Y the clear word, ie flag = (flag AND NOT X) EOR Y.

```
 DEF PROCMKMENU(A)
 READ $T
 FOR I=0 TO 24 STEP 4
 READ A$ : I!A=EVAL A$
 NEXT : T+= LEN $T+1
 ENDPROC
```

DATA line before this is called with menu header details. Sets up menu header at memory address A.

```
 DEF PROCMKENTRY(W,X,Y,H,F,A$,V)
 $U=A$
 RESTORE+1
 DATA Y,H,F,U,V,LEN $U
 FOR I=0 TO 20 STEP 4
 READ B$
 I!(W+X) = EVAL B$
 NEXT : U+= LEN $U + 1
 ENDPROC
```

Basic DATA line before this is called giving menu item details. W=header address, X=offset (multiple of 24), Y=work flags, H=submenu pointer (–1 if none). F, A$ and V as for PROCMKICON. Add entry for menu already set up.

```
 DEF PROCMKSPRITE
 READ A$,W,H,M,PW,NC
 SYS "OS_SpriteOp",&10F,S,A$,-(NC<>0),W,
H,M
 SYS "OS_SpriteOp",&125,S,A$,-1 TO ,,,,C
 IF NC FOR X=0 TO NC*8 STEP 8:READ C!X:
NEXT
 READ J$: J=EVAL("&"+J$): K=PW
 FOR Y=H-1 TO 0 STEP -1: FOR X=0 TO W-1
 SYS "OS_SpriteOp",&12A,S,A$,X,Y,J
 J=J>>(32 DIV PW): K-=1 : IF K=0 K=PW: R
EAD J$: J=EVAL("&"+J$)
 NEXT: NEXT
 ENDPROC
```

Create a sprite. DATA pointer must be be at a line giving sprite's name, width, height, mode, pixels per word (eg 32 for 2-colour) and number of colours before PROCMKSPRITE called. If number of colours >0 the following words give RGB palette entries. Shape DATA then given as words (without &) in compacted format. Intended for use with 2-, 4- and 16-colour sprites.

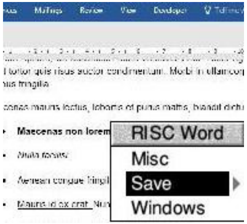
```
 DEF FNPROGINFO
 RESTORE +1
 DATA About this program,0,0,500,200,X,Y
,X+W,Y+H,0,0,-1,&84001012,&1000207,&C010
3,0,-H,W,0,&13D,0,1,0,T,0,0,0
 WX=FNMKWINDOW: $V="R2"
 DATA Name,Purpose,Author,Version
 DATA Application Demo,Drag N Drop,n.m (
dd-mm-yyyy)
 FOR IC=0 TO 7
 READ A$:LOCAL DATA
 IF IC<4 PROCMKICON(X,0,-50-IC*50,130,50
,&97000301,A$,0)
 IF IC>3 PROCMKICON(X,140,-50-(IC MOD4)*
50,350,50,&9700010D,A$,V)
 RESTORE DATA
 NEXT
 =WX
```

Create and return handle of 'About this program' window. Adust 3rd line of data as needed. Needs FNMKWINDOW and FNMKICON.

News and Apps

Were you April Fooled?

In the Spring edition of *Drag ’N Drop* we carried a news story about RISCWord which could read Microsoft Word files. Of course, that’s entirely fictional. (Some of you were duped although you *could* have tried rearrange the letters of the author’s name!)



So is there any RISC OS software for reading the so-called industry ’standard’ .doc files? Fireworkz (free version) can read very basic Rich Text Format (RTF) which is saved by Word. EZWriter (commercial, last updated 10 years ago) can load and save ancient (1990s) Word docs. As far as we are aware there is not (yet) any RISC OS software for handling the latest Word 2016 format.

However, there are websites which convert Word to PDF files such as www.ilovepdf.com which you can then view on RISC OS. You’ll need a capable browser like Iris.

Mamie Fletcher’s House

We reported the release of this game in 2021 which sees you roaming a haunted house armed with just an old fashioned camera, snapping it at the ghosties who perish in the camera’s flash. The author has updated the game it now sells at a budget price of £2.99. Your money now goes directly to support RISC OS Developments thanks to the author relinquishing his royalties. Mamie is available to purchase on !Store.

Archive address

We report on the rival [complementary surely – Ed] RISC OS magazine Archive for purposes of informing you it has moved offices [offices? What luxury – Ed] and can now be contacted at Archive Magazine, Beresford Business Centre, 2 Beresford Road, Coleraine, BT52 1GE. Their website remains at www.archivemag.org.uk.

RISC OS on a budget:
A PiRo computer

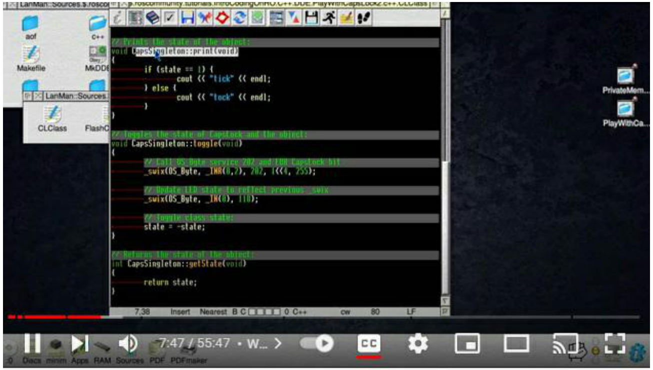
Hard and fast

Hardware developer RISCOSBits has released a couple of additions to their powerful but diminuitive RISC OS machine range. PiRO Noir and PiRO Qube are based on the Raspberry Pi Computer 4 module and feature state-of-the art ARMv8 architecture with options for solid state drives (SSDs) and NVMEs (Non Volatile Memory

Express) for super-fast RISC OS performance. Commercial software bundles are available. While away some time at pihard.co.uk/shop and marvel at the range of configurations of these mini marvels. Prices start at £129.

C++

Those who enjoy object programming (as opposed to functional programming) in languages like C and C++ may find video tutorials on debugging techniques within the Desktop Development Environment (DDE) on RISC OS. Visit Youtube and search for the 'RISC OS Community on GitHub' channel.



```
// Prints the state of the object:
void CapsSingleton::print(void)
{
	if (state == 1) {
		cout << "tick" << endl;
	} else {
		cout << "tock" << endl;
	}
}

// Toggles the state of CapsLock and the object:
void CapsSingleton::toggle(void)
{
	// Call OS_Byte service 202 and EOR CapsLock bit
	_swix(OS_Byte, _INR(0,2), 202, 1<<4, 255);

	// Update LED state to reflect previous _swix
	_swix(OS_Byte, _IN(0), 118);

	// Toggle class state:
	state = -state;
}

// Returns the state of the object:
int CapsSingleton::getState(void)
{
	return state;
}
```

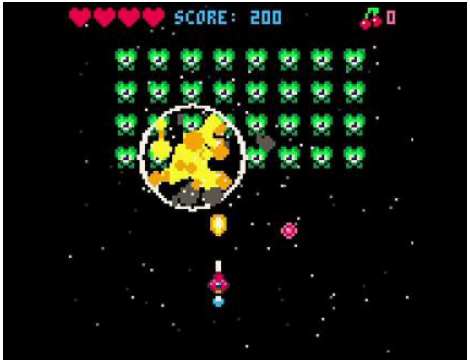
C++
MkDDE
Makefile
CLClass
PrivateMem
PlayWithCa
7,38 Insert Nearest B C 0 C++ cw 80 LF
7:47 / 55:47 • W...
CC
Discs RAM Sources PDF PDFmaker

Show me the money

RISC OS Developer Kevin Wells has been busy updating several useful little applications, Currency is now version 2.02 and Train Times is up to version 2.00. Both use real-time data fetched from the internet and useful for holiday planning. Visit kevsoft.co.uk to download these apps for free.

Iris 1.028

The capable and modern RISC OS web browser has received another update this time including a proper RISC OS dialogue box for printing web pages. Updates are free to existing users, just follow the link on Iris's home page. For new users, you fund Iris on an 'investment' basis: visit !Store, making a payment (£50.00 minimum is suggested) and you are furnished with login credentials to download Iris from a restricted page of RISC OS Development's website.



Cherry good fun

JVS Software has released a RISC OS port from 'Code the Classics' series. Cherry Bomb has been developed in Python, space-invaders style game with nine waves of aliens plus a final 'boss' screen. The titular weapon is a series of cherries that appear for collection and deployment. Cherry Bomb can be downloaded for free from !Store. You will need to have installed the Python-38 suite which can be obtained via Packman, see sites.google.com/sites/alansriscosstuff/packman if you don't yet have the package manager on your RISC OS machine.

Pinebook Pro

The Pinebook Pro from Pine64 is housed in a 12mm thick black magnesium alloy shell and weighting only 1.26kg it has a very contemporary look and feel.

The laptop features a 14inch IPS LCD display (1920x1080), a Hexa-Core Rockchip RK3399 processor clocked at 1800 MHz and 4GB of LPDDR4 RAM.

The keyboard has good key travel, and a large touchpad. As supplied in 2020, my laptop came with Debian preinstalled in the 64GB eMMMC, built-in 802.11ac WiFi with Bluetooth 5.0, and an NVMe expansion option.

There are three USB ports – two type A ports (one USB2 and one USB3) and a type C. Under RISC OS, though, all ports are restricted to USB2 speeds.

Battery life is very good but obviously will depend on the number of external peripherals connected. I always have a wireless mouse connected as I have a strong aversion to touch pads, so I work on 8-9 hours battery life.

One of the great things about buying from Pine64 is that they stock a wide rage of accessories and spare parts for their products as well as wikis that are full of information about each product.

The forums provide lots of useful information both generally and also about specific SBCs and laptops. After perusing the Pine Store, I purchased an NVMe int erface adapter, a USB Adapter for eMMC modules, together with a 128 GB eMMC module.

Access to the internals of the

Welcome

Drag N Drop is a quarterly magazine delivered in PDF format for RISC OS 5 computers like the Raspberry Pi and Pinebook. If you are interested in coding, are new to the RISC OS scene or just a returning user, then read on. We also feature articles on the BBC Micro.

Drag 'N Drop for RISC OS costs just £ 4.25 per issue (or £ 5.50 with the programs 'typed in'). There are no subscription fees, just come back to this website to buy the next one. We feature articles on programming on RISC OS in Basic and machine code, reviews of products, games and tutorials on using RISC OS applications and more.

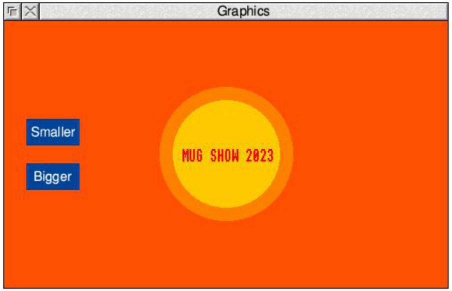
Shop

As well as RISC OS magazines you can buy other RISC OS publications from this site. Please select the product below and click Go for more information and to purchase!

MUG Mega Meet

This article is a summary of *Drag 'N Drop*'s presentation on writing a simple application at the summer 2023 MUG show.

We are going to write a simple application in BBC Basic which will plot a sun in a window on the desktop. It will let you click icons to change the sun's size.



BBC Basic is over 40 years old and some people think it's only for beginners learning to code but you can write very sophisticated apps using BBC Basic 5 on RISC OS.

The material in this article is based on the paper book The Application Tutorial and Listings Book (TATALB) available from www.dragdrop.co.uk.

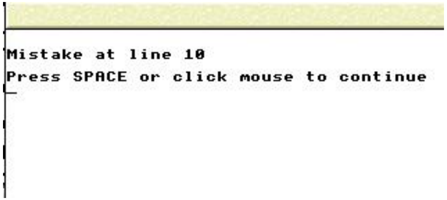
Firstly install Edit on the iconbar and menu over Edit's icon and choose Create>Basic. Next we need some standard procedures which can be typed in from the end of the article or downloaded from www.dragdrop.co.uk/free.htm

Don't worry about exactly how these procedures work, you can type in the them in any order and we'll fill in PROCPLOT later.

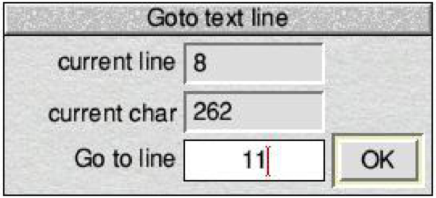
Now start typing some code. The first line is very important.

```
 ON ERROR SYS "Wimp_CloseDown":REPORT:PR
INT " at line ";ERL/10:END
```

If anything goes wrong, the app is stopped and the error and line number reported. The line number is divided by 10 so that in Edit you can press f5, type in the line number and go to the problem line. Other text editors may offer similar options to move the caret to a certain line number.



```
Mistake at line 10
Press SPACE or click mouse to continue
```

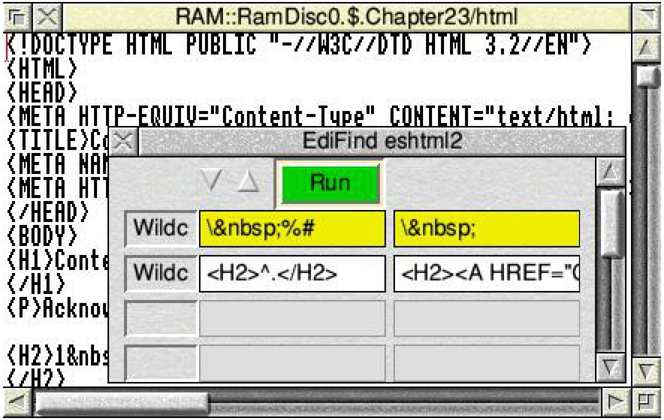


current line 8
current char 262
Go to line 11

Next some variables.

```
 DIM B 500: REM for window manager opera
tions
 DIM T 500: REM for window and icon text
```

EdiScript



RAM::RamDisc0.$.Chapter23/html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
<TITLE>Co
<META NAM
<META HTT
</HEAD>
<BODY>
<H1>Conte
</H1>
<P>Ackno

<H2>1&nbs
</H2>
```

EdiFind eshtml2

Wildc	\ %#	\
Wildc	<H2>^.</H2>	<H2><A HREF="(

EdiScript, or Eddie as he's known to his friends, is an application which enhances Edit's find/replace facility by running sequential searches at the touch of a button.

Edit is the free text editor which has come bundled with every version of RISC OS.

Type in the the application listing and, ensuring there are no errors, double click to run it. A window opens with three columns plus some control icons at the top.

If you already have Edit loaded on the iconbar it's a good idea to quit and rerun since Eddie works best with a fresh Edit.

The first column is the search type – normal, magic or wildcarded. The second is the find text and the third the replace text.

You'll notice some text already entered into the find/replace fields. This is Eddie's default parameters for converting Microsoft text files, replacing a carriage return with nothing, leaving just the line feed as the standard RISC OS end-of-line seperator, i.e.

To run the search, position the caret anywhere within the Edit window and click Run to look for the text highlighted in yellow on EdiScript's window. Eddie will always begin his search from the top of the file so the caret can be placed anywhere.

If the text is found, click the desired option in Edit's own window, such as End of file replace.

To add extra searches click on the leftmost column of the next empty line in the EdiScript window which will activate the greyed out boxes. Click again to cycle through the three search types (normal, magic, wild) then input your search/find text in the boxes.

Eddie's real power lies in running script files, which are searches put into plain text files and dragged to the EdiScript window.

Each search comprises three lines, the first is Magic, Wild or Normal. (In fact, Eddie just looks for M or W and assumes anything else is a Normal search term. So 'M', 'Magic', or even 'Magichars' are fine for Magic characters.) The second and third line are the find and replace text respectively.

For example:

```
Magic
\x0A
```

Scalable Vector Graphics

SVG or Scalable Vector Graphics is an industry standard image format used for web pages.

Unlike bitmap images (Jpegs, PNGs and the like) which become blocky when scaled up, vector graphics are only restricted by the resolution of the screen.

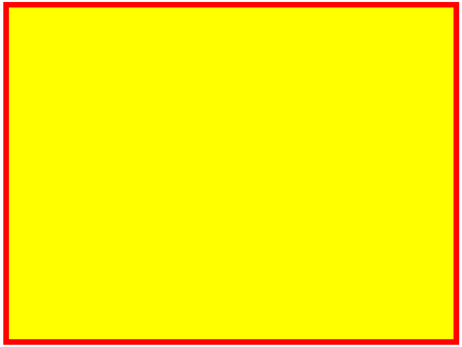
SVG is actually a mark up language like HTML – a series of tags in plain text to draw rectangles, circles and more complicated path objects.

In this article we’re going to learn how to write some simple SVG code and display it in Iris or Netsurf. Unfortunately Netsurf has limited support for SVGs although it has the option to export them as Drawfiles which Iris can't. We will stick to elementary SVG for now, however, so both browsers can be used.

First ensure the browser is installed on the icon bar. Next open your favourite text editor and type the following lines:

```
<svg>
<rect
x='5' y='5'
width='400' height='300'
style='stroke:red;
stroke-width:10;
fill:none'
/>
</svg>
```

Save it with a filename of your choice with filetype SVG (&aad), suffix /SVG if you think you’re going to be using it on non-RISC OS systems like Windows which rely on file extensions.



Drag the file to Netsurf and you should see a red rectangle appear. If you want to view the file in Iris you need to 'wrap' the SVG in HTML so insert <html> at the start and </html> at the end of the file.

The tag <svg> states that the following code is SVG. Like HTML, SVG doesn’t report errors, at best you’ll just get a blank page if you mistype anything.

Another tag is opened, rect to draw a rectangle. Each parameter is enclosed in single or double quotes (I have used single quotes here). Thus the bottom corner of the rectangle is (5,5) and is 400 × 300 units big.

The style is a string of codes giving the line or stroke colour in named colours or RGB notation (red and #ff0000 are the same). stroke-width:10 is the thickness of the line (10 units) and fill:none specifies no fill. Try changing this to fill:yellow for a yellow centre.

Confusingly, the rect tag is closed by a slash and a right angle bracket, />, not just an angle bracket as you might expect.

The closing tag for the whole of the

Last time in *Drag 'N Drop* we found some workrounds for older BBC Basic programs written for low-colour depth modes causing odd effects on newer hardware like the Raspberry Pi.

I am grateful to Tony Bartram who corrected me when I said all modes on the Raspberry Pi are 256 colours – there are actually 16 million colour modes available on the Pi, so the *minimum* colour depth is 256 on Pis.

USING SOUND AND COLOUR PROPERLY.

The 256 comes from the COLOUR n statement where n=0 to 255, many more colours can be displayed by using the extended COLOUR r,g,b.

With that sorted, we'll move onto SOUND. I'll be frank, sound in BBC Basic on RISC OS is *very* tricky. It's designed to allow the programmer finer control but playing tunes on the BBC Micro is childsplay by comparison.

For a start, there's no queueing system. The user has to write a routine of their own.

Playing chords isn't possible until you've fulfilled some formalities and the ENVELOPE statement is entirely useless. The system accepts it, as it does a positive number in place of the amplitude in the SOUND statement, but nothing happens.

Furthermore, on the BBC the duration parameter of the SOUND statement is measured in 1/20ths of a second. You can still make sounds on this basis but on RISC OS you have

This means programs using DATA statements giving a series of pitch and durations – which half-decent musical programs written for the BBC generally do – will fail.

What a boorach, as they say in these parts. But there are ways round these problems if we persist.

If you run a program like this on the BBC Micro:

```
10 FOR note=0 TO 12
20 SOUND 1,-10,52+note*4,10
30 NEXT
```

You hear a chromatic scale with each note lasting 1/2 second (ten 1/20ths of a second). But try this on RISC OS and you only hear the last note.

As we said earlier, there's no built-in queue on RISC OS so notes sound immediately rather than waiting for the first note to finish. Rather as if you had used the 'flush' option eg SOUND &11,–15,52,10 on every note.

Assistance comes in the form of an extra, fifth parameter to the SOUND statement. It makes the computer hold on for the specified number of centiseconds before playing the next note. Here is a revised version of the above for RISC OS:

```
10 wait=0
```

Maze and Means

Product: Amazeing
Supplier: Amcog Games
Price: £9.99

The latest offering from Amcog sees you on various missions in a scrolling maze, collecting energy pods, drums of chemicals and dispatching the marauding robots.

It's a "2.5D" game, a term I hadn't heard of until now – I know, I was too busy playing them in my youth to notice – which means the maze walls and your interaction with them look 3D but otherwise your character and robots move in plan view.

With Amcog you are guaranteed well thought out graphicss and high quality musical accompaniment. Amazeing with its atmospheric opening screen and chilling piano music, all composed by the game's author, is no exception.

Navigation on the intro screen is also easy with large, pointer-controlled buttons to start the game, select a level or quit to the desktop. Escape has an immediate effect bringing you back to the title page.

After clicking Start you are briefed on your mission. The aim might be to eliminate deadly fungus by picking up cans of chemicals or destroy a certain number of robots before you can progress to the next level.

You move a rather ghostly (but nicely animated) figure with simple control keys for up, down, left, right, fire and bomb. Some of the robots are quite cute looking but I was soon reminded that appearances aren't everything!

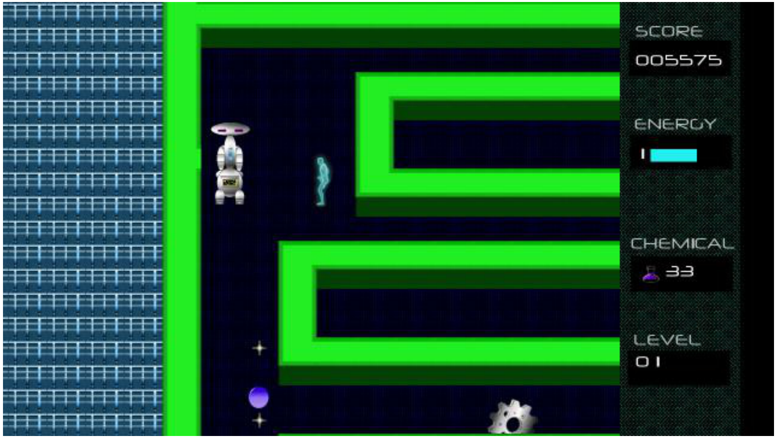
I liked it that robots lay trails of energy pods to collect and on some levels you can just kill as many robots as you want to boost your score. Pills appear at random which you can collect to speed you up or slow down your movement. All this adds to a varied gaming experience.

You can skip to any of eight different levels before commencing play without having to enter a code

I'm not sure I'm keen on this evasion of a challenge (by not having to complete a previous level to get to the next one). I suppose these days, with walk-throughs and codes being shared everywhere online, there maybe isn't much point.

Score and lives are displayed on the right, it took me a while to realise the 1, 2 or 3 on the left of the energy meter was the number of lives you have left!

● Amazeing is a good game, enjoyable to play and well worth the money.



Stop Motion

It seems to be the season for graphics character editors.

Last time we featured a tessellating sprite editor (see *Drag 'N Drop* 12i3) and here we have another unusual one whose potential makes you wonder why it hasn't been written before.

You're restricted to the monochrome 8×8 user-defined graphics (UDG) characters but larger shapes of any size and any number of frames can be built up, subject to screen mode and range of defineable characters.

You can see the result instantly as the UDGs are redefined 'on the fly' and the program cycles through each frame. The program works both on BBC Micros and RISC OS (!GraphTask is recommended).

When RUN, you'll be prompted for screen mode, size of an individual frame in characters and number of frames. For example, entering 4,2,2,2 will make the editor run in Mode 4 with two frames of 2×2 characters per frame.

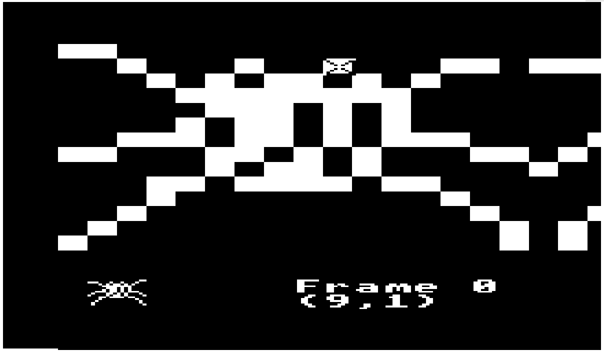
Use the cursor keys to move around the screen with space to set or unset a pixel.Frame number and co-ordinates within the frame are displayed. As you build up the shape the animated version is shown at the bottom left.

Pressing P will cause the pixel to be replicated across subsequent frames (but not the preceeding ones). This useful if you had, say, two frames and tweak pixels in the second frame rather than build it from scratch.

The frames are animated at 5cs which can be slowed down in steps to 50cs by tapping the F key.

When you have finished press Return and you'll be prompted for a filename then the character definitions will be SPOOLed to disc as DATA statements. To see how to use them, follow the two demonstrations.

The following program demonstrates animation of the spider character



Frame 0
(9,1)

SPIDER listing

```
   10 MODE4
   20 VDU 23,1,0;0;0;0;
   30 FOR ch=0 TO 7:VDU23,128+ch
   40 FOR i=1 TO 8:READ j:VDU j
   50 NEXT:NEXT
   60 F%=0
   70 REPEAT
   80 VDU 31,10,10,F%+128,F%+129,8,8,10,
F%+130,F%+131
   90 F%=F% EOR 4
  100 T%=TIME:REPEAT UNTIL TIME>T%+5
  110 UNTIL0
  120 DATA 192,34,21,15,7,11,59,198
  130 DATA 0,70,168,240,80,80,92,163
  140 DATA 5,27,16,32,64,128,0,0
  150 DATA 160,216,4,2,1,1,0,0
  160 DATA 0,226,21,15,7,11,59,70
  170 DATA 3,68,168,240,80,80,92,162
  180 DATA 133,27,16,32,64,64,0,0
  190 DATA 161,216,4,2,2,1,0,0
```

G’day mates, I’m back in the Antipodes after flat out travelling to Europe and beyond.

Did you know the Eiffel Tower has to be repainted every 7 years and it’s had 7 different colours since it was built in 1887?

We’re going to extend the project we built in the Spring 2023 edition of *Drag ’N Drop*. We connected a breadboard to the Pi and got an LED (light emitting diode) working.

It needed no programming on RISC OS, just connecting it to the pins on the GPIO (General Purpose Input/Output port) was enough to light it up.

It’s time to get on to some GPIO programming but there are a couple of key things concerning Ground pins and Reserved pins of which you should be aware.

Reserved pins are used to perform communication between I2C and EEPROM. If you are new to the Raspberry Pi, you are advised not to connect anythingwith these pins, which are pins 27 and 28.

Ground pins are used to close the electric circuits. The ground pins help you to protect your board from burning and play an important part in a circuit. The ground pins are numbered 6, 9, 14, 17, 20, 25, 30, 34 and 39. Refer back to the last article for a pin diagram.

In some of your forays into GPIO you may well find that your Pi may feel a little bit warmer than it does in regular use.

The programs which follow represent the first in a collection of mine which may allow you to come to grips with using the GPIO pins found on the Raspberry Pi 3B+.

You may need to install some modules to drive the GPIO which can be found at www.tankstage.co.uk/software/GPIO.zip.

The non computing hardware consists of breadboards, jumper leads, resistors, press switches, clear and coloured LEDs and old fashioned vero boards.

Listing 1 is a version of a ‘Blink’ program written in Basic 5.There are some minor changes to the setup we used last time as shown in the diagram on the next page.

Ensuring you have turned off the power to your Pi, disconnect the female end of the jumper lead from pin 1 on the GPIO and connect it to pin 7. GPIO 4 (physical pin 7) is now connected to the LED anode, its' longer pin. The LED's shorter pin is connected to physical pin 6, an earth (ground) pin. Physical pin 7 is made to be write enabled using: