

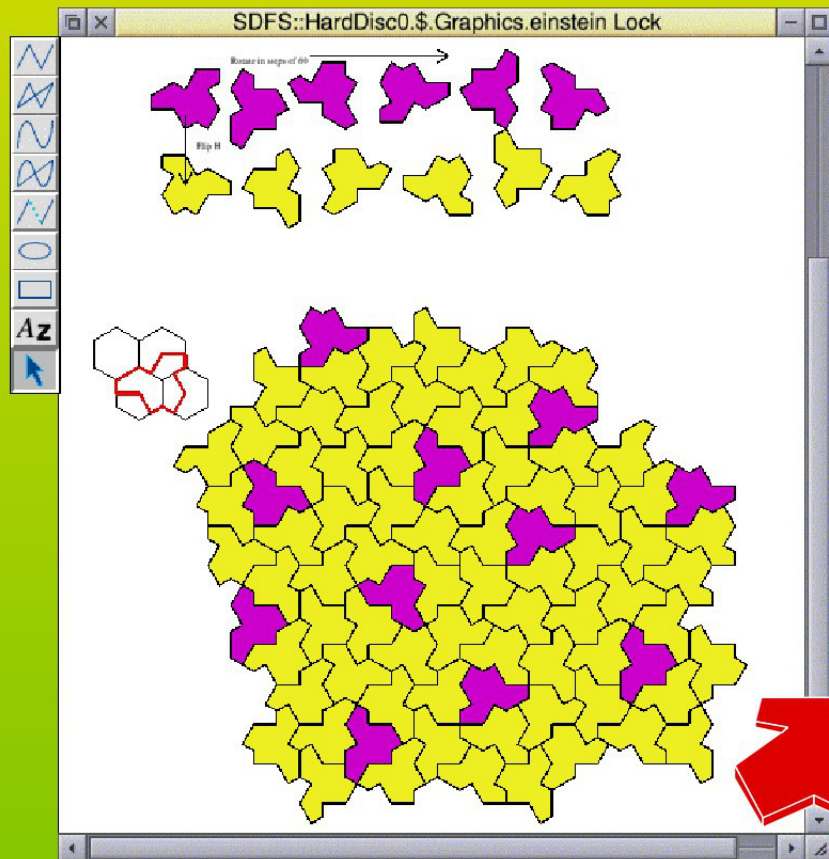
DRAG'N'DROP

Volume 12 Issue 3

Spring 2023

£4.25

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



Einstein Tile

3D Turbo Boost

Draw to JPEGs

Custom Icon Bars

Raspberry Pi GPIO

TM

RISCOS - For the Past, Present and Future

<http://mug.riscos.org/mega>



MUG Mega Meet 2023

Putting the mmm back in to summer!

Saturday 1st July 11.00 – 4.00



Copyright ©2023 Drag 'N Drop. All rights reserved.
Produced on RISC OS computers

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

Anthony Bartram (Anatomy of a Game)
Brian K Charradia (Risc OZ)
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	3
Beginner's Page and Wimp Library	4
News and Application Updates	7
3D Turbo Boost review	9
EinsteinTiles	10
Anatomy of a Game.....	13
Draw2Jpegs	18
Custom Icon Bars.....	23
Leaky Roof	26
8 bits	28
RISC OZ.....	30

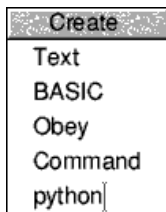
Editorial

Hi RISC OS users and welcome to another edition of *Drag 'N Drop*! I know that a number of you like to print your magazine so I hope you like the new, wider format which prints in landscape orientation on A5 (or enlarged to A4 landscape). Drag 'N Drop isn't able to be at the Wakefield (Bradford) show this year but if you are going then have a good time – and even if you aren't, enjoy reading this edition of the magazine.

Entering listings

You can type in programs in three ways:

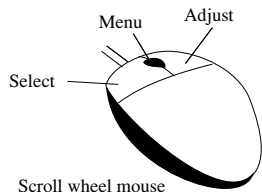
1. Use a text editor like Edit. Press the Menu mouse button over Edit on the icon bar and choose Create>BASIC. For other languages type in at the bottom. Save it by pressing F3.



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.



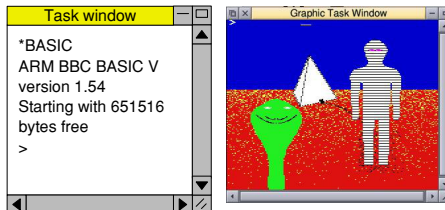
Type AUTO for automatic line numbering. Press Escape to stop and type SAVE "myprog" followed by Return to store myprog on hard disc. Type RUN to execute the program. To return to the desktop type *QUIT.



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.



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.

```
PROCWRCONF(W1,6,10<<28,15<<28): PROCWR  
TCONF(W1,6,"Process")
```

Start of line has a space
Continuation of line (no space)

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.



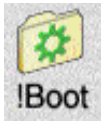
What does 'currently selected directory' mean?

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 Application 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.pi-star.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.

WIMP Library

```
DEF PROCWRICONT(W,H,B$)
PROCRCICON(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$,U)
$U=A$
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 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 FNMPROGINFO
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: $U="R2"
DATA Name,Purpose,Author,Version
DATA Application Demo,Drag N Drop,n.m (
dd-mm-yyyy)
FOR IC=0 TO 7
READ A$:LOCAL DATA
IF IC<4 PROCMKICON(X,0,-50-IC*50,130,50
,&97000301,A$,0)
IF IC>3 PROCMKICON(X,140,-50-(IC MOD 4)*
50,350,50,&9700010D,A$,U)
RESTORE DATA
NEXT
=WX
```

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

News and Apps

RISC OS Yorkshire

By the time you read this the RISC OS show oop north will be in full swing. Organised by the Wakefield RISC OS Computer Club (WROCC) it's taking place in the Cedar Court Hotel Bradford 10.30am–4.30pm on 22nd April. Prices a whole £2.00 on the door.

WROCC design competition

The user group is looking to update their logo and submissions are invited with a closing date of 22nd April (presumably the winner will be announced at the show then). Find out more at www.wrocc.org.uk.

Spark of an idea

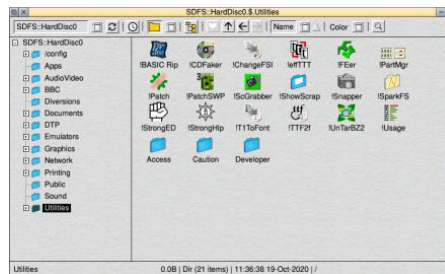
SparkFS has been bundled as read-only with RISC OS distros, usually the default app for opening Zip files from the Internet. If you wanted to create your own compressed files you had to buy the full version but now that is being bundled free with the HardDisc4

image too. SparkFS can handle a variety of compressed formats.

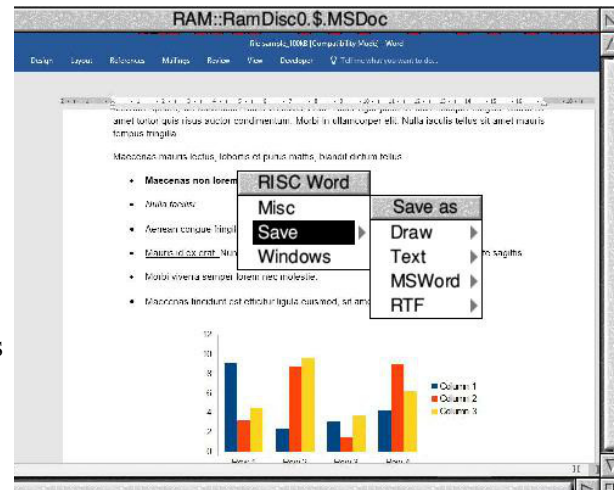
RISCWord

Development of a RISC OS version of Microsoft Word has begun by retired physics Prof Ali Lo who is looking for beta testers, visit tinyurl.com/3sn2rd9u to learn more.

CLFile 1.02



The drag and drop nature of the RISC OS filing system and integration with

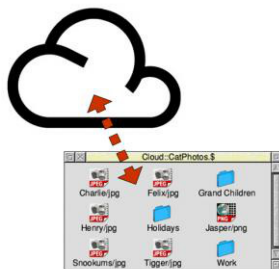


applications is renowned for its ease of use but this hasn't stopped Stefan Fröling developing an even better version of it. CLFile is available on a 'pre-order' scheme where for a one-off payment of €15.97 you get monthly beta updates. See riscoscloverleaf.com/product/new-filer-gui-pre-order to find out more.

CloudFS 0.34

Cloud storage is the practice of keeping

files on the internet (a data centre or server) instead of on your Raspberry Pi (local machine). The internet keeps backups on different servers so you don't need to worry about losing your work and you can access your files anywhere you have an internet connection. Elesar's CloudFS allows you to interact with the cloud just like the normal RISC OS filing system with a standard filer window. Version 0.34 has had some tweaks and is free to existing users or £28.80 for new users. Visit shop.elesar.co.uk > Software.



MkSprite 1.11

MkSprite is a package of two command-line driven utilities to



convert Drawfiles and Euclid files; it also merges sprites. Example files are included but not instructions – copy the contents of MkSprite-1/11/zip.Library to your !Boot.Library, double click Examples.test and the result is Examples.draw_s which you can open in Paint.



Midlands Mega Meet

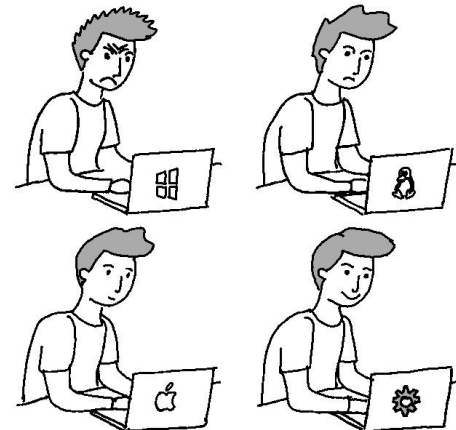
The RISC OS Midlands user group, traditionally known as the MUG, is holding a global meeting on 1st July 2023 and you don't even have to get out of bed to attend! As long as you have a device other than your RISC OS machine you can join on Zoom (for sadly there's no version of Zoom for RISC OS just yet). So there's no excuse

for not going. For more details visit

mug.riscos.org/mega/Virtual.html.

Next Issue

And speaking of summer, the next issue of *Drag 'N Drop* will be out at the beginning of August. Due to space in this issue of the magazine, Using Sound and Colour Properly will be back plus a couple of high quality and very useful type-in applications along with all the regulars. See you!



Be fast or be last

Product: 3D Turbo Boost

Supplier: Amcog Games

www.amcog-games.co.uk

Price £11.99 (Plingsotre)

RISC OS gameswriter Tony Bartram of Amcog has turned his hand to a 3D car racing game.

You've been selected to test drive a range of indestructible cars (so the story goes) and you race them against the

computer or another player.

The introductory screen lets you modify control keys, choose from five soundtracks, composed by Mr Bartram himself, and which one of four vehicles to drive. Think of beefy stunt cars from 70s and 80s TV series like Knight Rider and you get the idea.

The race begins on a screen with a 3D split view – from the other car's

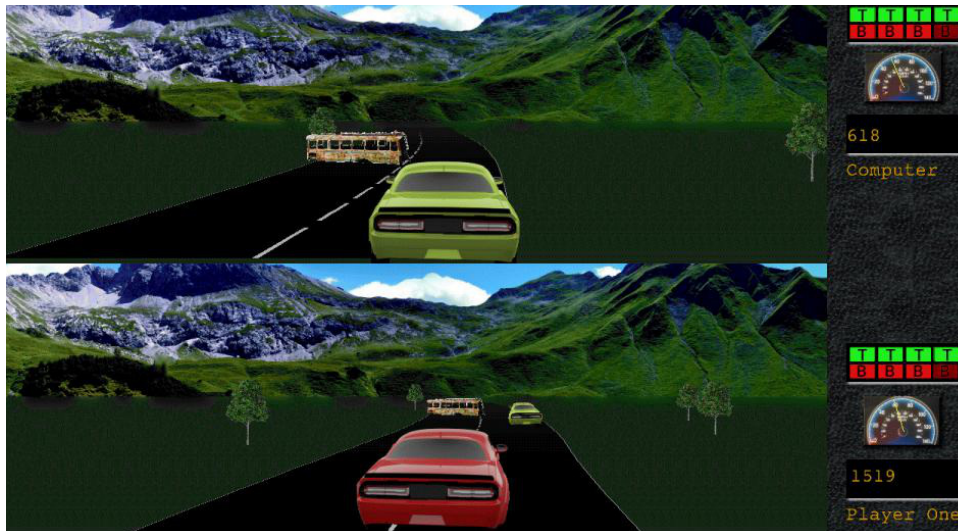
perspective at the top and yours at the bottom. There is some nice background scenery (mountains on the first level) although it doesn't move as you steer left and right to negotiate bends and obstacles blocking the road. Crash into a massive concrete bollard and you don't even sustain a scratch, just rev your engines back up and get on your way.

Once you hit an open stretch and get speed up you can apply one of the eponymous turbo boosts. You can slow down and steer round the blockage or leap over them using up one of four jumps. This makes six control keys – left, right, speed up, slow down, jump and apply turbo charger.

The car's speedometer and number of remaining boosts and jumps are clearly drawn on the right.

I liked the fact that when you catch up with the other car (or vice versa) it appears in your portion of the screen.

● Being an immortal driver in an indestructible car is great fun. When both cars try to occupy the same space on the road the ensuing kerfuffle is quite entertaining.



Einstein Tiles

You may have read in the press recently about the discovery of a 13-sided 'Einstein tile' by retired Yorkshire printing engineer, David Smith.

Einstein has nothing to do with he of General Relativity – in German 'einstein' means 'one stone' and mathematicians have been looking for such a 'stone' or tile for years. It can be used to cover a surface without creating a repeated pattern.

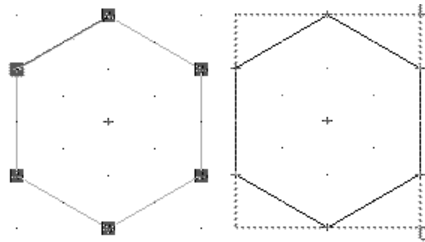
The technical name for this shape an 'aperiodic monotile'. An einstein tile is just a hexagon cut up and rearranged into another shape which is rotated in steps of 60 degrees and flipped to make 12 shapes to tessellate on the 2D plane.

In this article we will use Draw, which has come free with every version of RISC OS, to create the einstein tile, the 12 variations, and placing them to create a fascinating pattern. Be warned, it's addictive!

First of all load up Draw by select clicking on the Apps folder and double

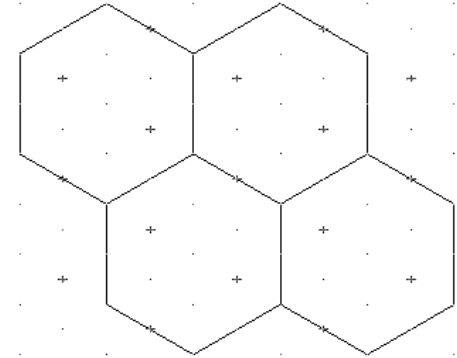
clicking !Draw. Select click the icon bar to open a new Draw document. We need an isometric grid so click the Menu mouse button > Grid. Using the Adjust button click on Show, Lock and Isometric. Leave the spacing as the default which is 1cm with 2 divisions (meaning objects snap to the nearest 1/2 cm or 5mm).

Now select the closed straight line tool (the one that looks like a wonky bow tie). Draw a hexagon, using the spots on the isometric grid to help.

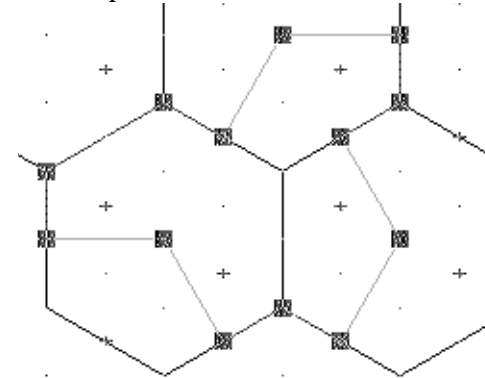


Now create three copies of the hexagon. Ensure it is selected (select click anywhere on it) and choose Menu > Select > Duplicate) or Ctrl+D. Arrange the four hexagons in honeycomb

pattern as shown below.



Select the close-line tool again. Construct the 13-sided tile with the control points shown below.



Anatomy of a Game

The latest title from AMCOG is a 3D car racing game. It's our first two player title and also includes the option of playing against the computer.

There were a number of challenges in producing a game like this and this article explores the kind of problems that were encountered and how these were solved. It also takes a look at how the graphics, sounds and music were made.

Game Design

Game design is perhaps the hardest part of developing a game. The game needs to be fun, so there should be some known entertaining aspect to the game e.g. being chased, solving puzzles, having a time limit, competing against an adversary, answering questions etc. This is with a view to triggering either the flight or flight response, a sense of nostalgia, endorphins, or entertain in some other way.

At the core of the game there should be some key distinctive focus that defines what the game is about and makes it fun. I went through multiple designs:

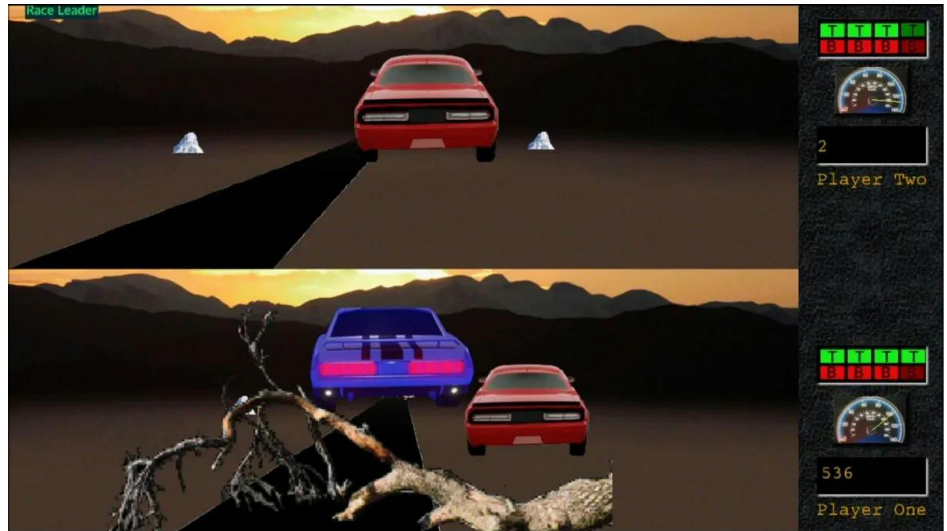
1. Collecting and using powerups e.g. nitrous oxide for speed, oil or smoke to thwart an adversary.
2. Running away from the law.

4. Participating in a race of 4 or more cars for racing positions
5. A car duel, where only 2 cars are competing.

For (1) I felt that one player could gain an advantage and then progressively disadvantage the other player.

This would seem to prevent an effective competition. So I decided that any power-ups would be non-collectable and fairly distributed.

I selected (4) as a design and thought



Draw to Jpegs

I wrote this application to convert a directory of drawfiles to Jpeg format for putting in HTML documents.

Users of non-RISC OS platforms can then view RISC OS drawfiles as part of a web page – rather than printing a Drawfile to PostScript/PDF and making the web browser download the file each time.

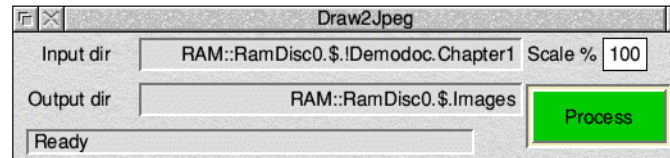
Draw2Jpgs, the application listed here, can be used in conjunction with DDF2Html (discussed later) to convert Impression Style files to HTML. It needs ChangeFSI to work behind the scenes, this is supplied in the ROOL distros of RISC OS so most users will have it already installed.

In Style, if choose File > Preferences and uncheck Single file document, Style will save the document as an application directory.

If you downloaded Style 2019 from Pingstore there are a couple of files saved in this format, !Tutorial for instance. Inside !Tutorial (hold down shift and double click the 'I' application

icon). There are subfolders called Chapter 1, Chapter 12 etc. Inside them are drawfiles called Story11, Story 13 and so on. These are numbered in order of creation rather than appearance, for example the Chapter 26 folder actually is the first chapter in the document.

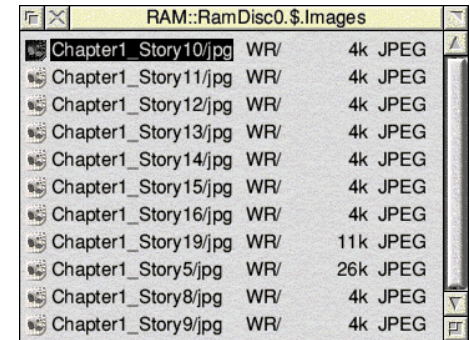
Before you do anything else, adjust the memory allocated to applications by adjust-clicking the switcher (raspberry) icon and drag the red Next bar in the top section. Drag it out to at least a couple of megabytes (2048K). If you don't the application will report an



error message and refuse to run.

Then type in the listing and save it as DRAW2JPGS. Double check you haven't made a typing slip and run it. All being well a window should appear like the above. Drag a chapter directory onto

the blank rectangle to the right of 'Input dir' field. Create a suitable directory (eg Images) and drag that into the 'Output dir'. Adjust the scale if desired. Click the large green Process button to begin.



Notice the button turns red, you are processing large numbers of drawings and want to stop for any reason, hold down the Menu button to halt conversion.

The Images directory will contain Jpegs with filenames beginning with the Chapter directory name and Story name separated by an underscore.

The above example is from the Demodoc in the Documents folder.

Custom Icon Bars

Last time we created our own set of Toolsprites and we'll round off by sprucing up the usual boring, grey icon bar.

The iconbar is actually a window with no furniture. Making our own iconbar window is done in four stages:

1. Add an iconbar-sized sprite icon to the iconbar window definition that is held inside the Wimp templates file
2. Design or obtain a sprite for the sprite icon
3. Write a relocatable module to instruct RISC OS to 'soft load' the new templates into the Resource Filing System
4. Add a small Obey file to your Boot sequence to activate the softload on bootup.

This all sounds intensely scary but don't panic, I'll walk you through it.

The computer has a default set of

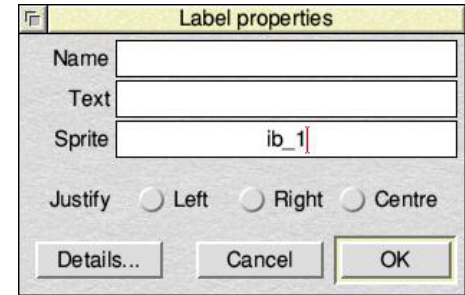
window templates including one for the iconbar. These are located in the Resources folder – menu click over Apps on the iconbar > Open '\$'. Navigate to Resources.Wimp and that's where the file Templates live.

You'll need the WinEd utility, go to www.riscosopen.org > Downloads > Miscellaneous > Bonus binaries. It's located in the BonusBinDev/zip download in HardDisc4.Utilities. Developer.

Double click the Templates file. In the pictorial list of windows double click iconbar and the grey iconbar rectangle will appear.

Adjust-click on WinEd's iconbar icon to bring up the icon palette. Drag the Label icon from the palette to the window. Double click it. In the Label properties window delete the Label text and type ib_1 in the Sprite box. Adjust-

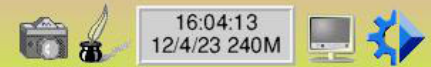
click on the radio button next to Right so that all justification is switched off.



Click Details and amend Width and Height to 2600 and 130 respectively. Then click Close then OK. Now click and hold the red frame. Drag it up and left so the edges of the frame disappear off the edges of the window.

In WinEd's main window save the templates to a suitable location on your hard disc and ensure the currently selected directory is set to where you've saved. I suggest a new folder called IconBar in the Utilities directory.

We now need to create a sprite called ib_1. Listing 1 will do the job, it creates



Leaky Roof

This game shows what you can do thanks to the easy sprite system on RISC OS and a few lines of code.

Originally from a Spectrum listing, the roof's leaking and you only have one bucket. Use the Z and X keys to move left and right. The rain lasts for thirty drips and speeds up – so catch as many as you can.

LEAKY Listing

```
REM Leaky Roof
REM From a ZX Spectrum Game
REM Drag N Drop Spring 2023
MODE 13: OFF
ON ERROR REPORT:PRINT" at ";ERL:END
PROCsprites
sc=0: PRINT "SCORE ";sc
GCOL 24:MOVE 0,375:PLOT 1,1200,0
FOR j=0 TO 1
FOR i=0 TO 19
```



```
PROCput(0,i*64,927-j*64)
NEXT:NEXT
bx=480
PROCput(2,bx,383)
REPEAT d=d+1
kx=RND(309)*4
PRINTTAB(25,0)"Drip ";d
ky=735
REPEAT PROCput(1,kx,ky)
IF ky=407 AND kx>bx-32 AND kx<bx+64:sc+
=1:PRINT TAB(0,0)"SCORE ";sc
k$=INKEY$(0) : dx=- (k$="x" AND bx<1200)
+(k$="z" AND bx>0)
IF dx PROCput(2,bx,383):bx+=dx*16:PROCp
ut(2,bx,383)
PROCdel(1+(30-d)/5)
PROCput(1,kx,ky)
ky=ky-8
UNTIL ky=399
UNTIL d=30
END
:
DEF PROCput(A,X,Y)
SYS "OS_SpriteOp",@122,S,STR$A,X,Y,3
ENDPROC
:
DEF PROCdel(T)
T1=TIME:REPEAT UNTIL TIME>T1+T
ENDPROC
:
DEF PROCsprites
X=1000:DIM S X,C 3
```


Tesellation Editor

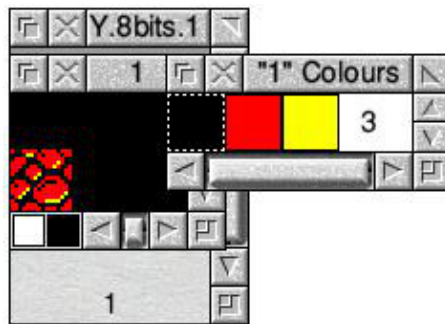
Over the years there have been more sprite editors for the BBC Micro and Electron than I've had hot dinners.

The sprite designer presented here offers one significant advantage over all the others I've seen and that is the ability to see the sprite tessellating with itself as you edit.

It runs equally happily on 8-bit Acorns and RISC OS machines, !GraphTask is recommended on the Raspberry Pi. If you are only going to be using a BBC Micro (or emulator) you don't need to type in lines 300 and lines 490-640.

When you run the program, input the graphics mode for the sprite which can be zero, 1, 2, 4 or 5 (plus 8 or 9 on RISC OS) followed by the sprite dimensions in pixels.

Move the X cursor round the editing area with the arrow keys. Keys 0-9 and A-F will colour in the square on the grid and it will also plot nine copies of the pixel at actual size on the right – the sprite plus its eight neighbours



When you have finished designing your sprite, press the Return key and enter a filename. On the BBC Micro a series of DATA statements will be spooled to

disc and on RISC OS the data will be saved as a standard sprite file which can be opened in Paint.

SPTEDIT listing

```

10 REM Simple Sprite Editor
20 REM (c) Drag N Drop Spring 2023
30 ON ERROR OSCLI"FX4,0":REPORT:PRINT
" at ";ERL:END
40 REPEAT INPUT "Mode",mode$
50 mode=INSTR("0124589",mode$)-1:UNTIL mode>0
60 INPUT"Width, Depth",w,d
70 MODE VALmode$
80 VDU23;8202;0;0;0;
90 DIM pix(w,d)
100 gx=VALMID$("2484824",mode+1,1):REM
graphics points per pixel X
110 pc=VALMID$("8428484",mode+1,1):REM
pixels per column
120 cc=w DIV pc : cr=1+((d-1) DIV 8)
130 xs=8*gx:ys=32 :REM size of cell gr
aphics units
140 rm=cc*pc*xs + gx*pc*cc + gx :REM r
ight margin
150 xc=0: yc=0: a$="0123456789ABCDEF"
160 *FX4,1
170 PROCcursor
180 REPEAT
190 k=GET
200 PROCcursor

```



Pesky here again from the Antipodes, mates, and pack your swag as we're heading to the uncharted territory of your computer – the GPIO port.

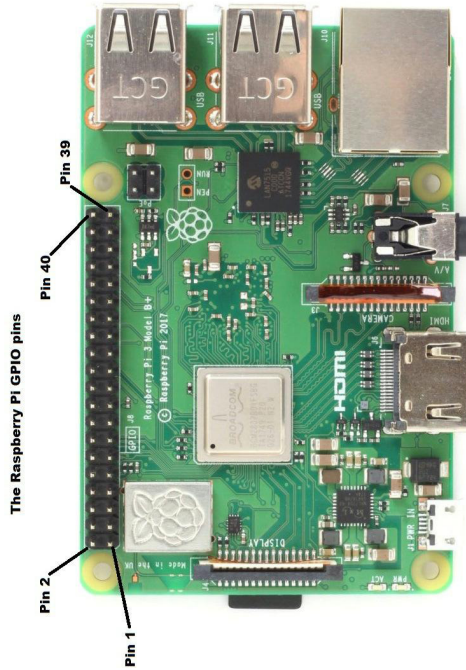
The BBC Basic 5 computer language available on a RISC OS based computer can be used to input data via its GPIO pins.

Using GPIO pins you can collect data from the outside world as well. Once you learn how to do that safely, without harming the Pi, other fun projects are waiting for you to do.

In recent versions of the Raspberry Pi the GPIO pins are positioned on the left hand side of the circuit board.

Take a squiz at the picture, pins are organised into a 2 × 20 rectangle with

pin number 1 positioned bottom. Pin 2 is on the left, pin 3 on the right above pin 1 and so on all the way up to pin 40.



Next is a 'pin map' table orientated in the same manner. The pins are numbered the same in middle two columns.

BCM 21	GPIO 21	pin 40	pin 39	GND	
BCM 20	GPIO 20	pin 38	pin 37	GPIO 26	BCM 26
BCM 16	GPIO 16	pin 36	pin 35	GPIO 19	BCM 19
GND		pin 34	pin 33	GPIO 13	BCM 13
BCM 12	GPIO 12	pin 32	pin 31	GPIO 6	BCM 6
GND		pin 30	pin 29	GPIO 5	BCM 5
BCM 1	ID_S C	pin 28	pin 27	ID_S D	BCM 0
BCM 7	GPIO 7	pin 26	pin 25	GND	
BCM 8	GPIO 8	pin 24	pin 23	GPIO 11	BCM 11
BCM 25	GPIO 25	pin 22	pin 21	GPIO 9	BCM 9
GND		pin 20	pin 19	GPIO 10	BCM 10
BCM 24	GPIO 24	pin 18	pin 17	GND	
BCM 23	GPIO 23	pin 16	pin 15	GPIO 22	BCM 22
GND		pin 14	pin 13	GPIO 27	BCM 27
BCM 18	GPIO 18	pin 12	pin 11	GPIO 17	BCM 17
BCM 15	GPIO 15	pin 10	pin 09	GND	
BCM 14	GPIO 14	pin 08	pin 07	GPIO 4	BCM 4
GND		pin 06	pin 05	GPIO 3	BCM 3
5v		pin 04	pin 03	GPIO 2	BCM 2
5v		pin 02	pin 01	3v3	