

The Ultimate Sprite Searcher

Introduction.....2
Your CPC's memory.....5
What is a Sprite.....6
Sprites and how they are stored in memory.....7
Which Version of TUSS should I use ?.....8
TUSS instructions.....9
Searching for sprites.....11
The Clipboard.....12
Trouble-Shooting.....13
How To use the Sprite in you own code.....14
By-products of TUSS.....15
Dictionary of terms.....16

TUSS © Sentinel Software by Richard Wildey 1993.

41 Enmore Gardens, East Sheen, London, SW14 8RF.

The Ultimate Sprite Searcher.

Introduction.

The Ultimate Sprite Searcher started life as a labour of love, a utility for my own personal use but gradually after several all night programming stints TUSS started to grow into a saleable item. Which is what you have here, the final version. This manual is designed for people with little or no serious CPC knowledge so those in the know please bear with me if you find this manual a little simple at times. You may even like to RUN "DISC" and skip straight to page 7.

It must be pointed out that to pass off any graphics ripped from a program as your own would be a blatant infringement of copyright laws, and by no means does Sentinel condone such acts.

The Ultimate Sprite Searcher copyright 1993 R. Wildey.

Sentinel Software,
41 Enmore Gardens,
East Sheen,
London,
SW14 8RF.

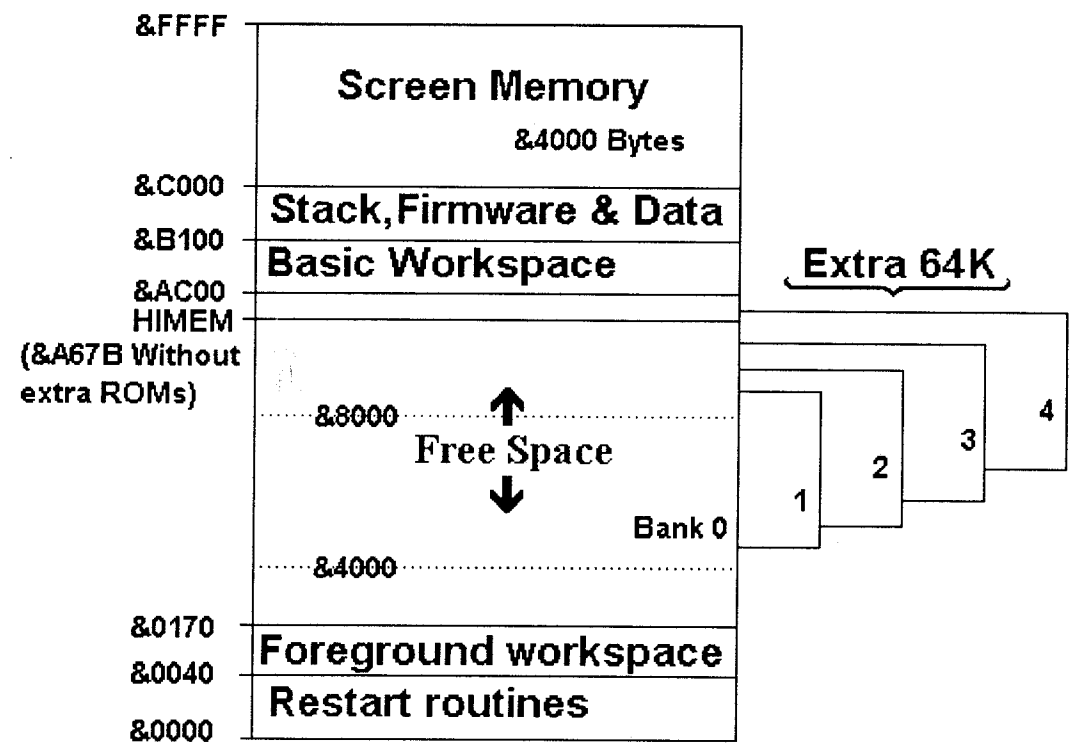
Before using this program a basic knowledge of your CPC's memory, sprites and how they are stored in the computers memory is needed.

Your CPC's memory.

Every CPC has 64K of memory (128K owners will be covered later) this 64K is made up of &FFFF (65535) bytes, of these only roughly 42K can be used to store data reliably under normal circumstances. &40 to approximately &A67B. Above this is used for the CPC Firmware the various routines that make a programmers life that much easier. &C000-&FFFF is the screen memory, any data stored here would be displayed on the screen.

The extra 64K that 6128 owners have is split up into 4 extra banks each of &4000 (17K) bytes long. When a new bank is "paged in" it replaces all the memory at &4000 to &8000. Bank 0 is normal RAM (The default). The extra banks (ie. not bank 0) all survive a "soft reset" so you can load the game, reset the computer with [CONTROL], [SHIFT] & [ESC] or with the multiface button, then load in TUSS and still view the extra banks.

CPC Memory Map.



If you are somewhat lost here don't panic - the long and short of it is that the graphics you are looking for will be stored in memory somewhere in the free space as this is where your game will be, somewhere between &0040 and HIMEM. If you don't have a ROMboard your point of HIMEM (HIGhest usable byte in MEMory) will probably be &A67B. You can find your HIMEM address by typing, from BASIC:

```
PRINT "&";HEX$(HIMEM)
```

If a game is 128K only or a multi-load on a 64K machine but not on a 128K it is advisable to search the extra banks (&4000-&8000 each) as this is very often used for sprite data.

What is a Sprite ?

Sprites are the graphics that form what you see on the screen, this can be anything to characters you control to plants that make up the background scenery. Take the classic game Pacman for example. There are many objects in this game but only a few sprites in comparison. Firstly we have Pacman himself and the many ghosts that chase him. Pacman will probably have a number of "frames" of animation, causing his mouth to open and close as he eats, a minimum of two.

Though there are many ghosts this will just be one sprite repeated many times in different colours. The maze Pacman must make his way around will not be one large sprite but made up from a number of smaller ones consisting of straight lines, corners and junctions - much like the ASCII characters 145-159. This process of making large areas of graphics is a commonly used technic, another good example is a wall in a game. It may contain fifty bricks but most probably this will be a sprite of three or four bricks repeated a number of times, this is used in Prince of Persia.

Any character in a game that moves will no doubt have several frames of animation as explained, each of these will need to be a separate sprite - example Inzogold. In the above Pacman example one sprite will be needed with Pac's mouth open and another closed. Larger characters may well be made up of a number of sections such as a sprite for the head and body and a separate one for the legs as used in Andy Capp. This is done to save memory and speed up the movements but unfortunately means that a bit of extra work will be required to extract the sprites.

Sprites and how they are stored.

Not only are there these methods of printing the sprite but they can be stored in two primary ways, vertically or horizontally. This is a bit more complicated so read carefully. When sprites are originally drawn they must be "grabbed" and put into the code. This is done by reading each byte occupied by the sprite on the screen and putting the value into memory. How the programmer chooses to grab the sprite is his/her prerogative. This can be done by reading the top left pixel and reading left to right till the end of line is reached and then stepping down a pixel to repeat the whole process to form a horizontal sprite. The other method is exactly same except instead of reading left to right the pixels are read from top to bottom, then moved right to start again till the complete sprite is read storing a vertical sprite in memory. Out of the two methods horizontal is far more common. Vertical sprites are generally only used for large fonts.

Some MODE 1 games (Spectrum ports mainly) have monochrome sprites - those that are only one colour. So to save on memory the sprite is grabbed in MODE 2 and displayed in double width in MODE 1. (Rockstar Ate My Hamster/Nigel Mansell) I have only ever seen this method used with horizontal sprites.

A third method is used to display graphics, though not strictly speaking sprites. This is to store a whole screen in memory, this method is rarely used in games but you sometimes find it used for menu screens and in demos.

Which version of TUSS ?

On the disc are six different versions of TUSS, it is your job to decide which will suit you best.

RUN "DISC" to get to the main menu, this gives you the option of installing the multiface versions or running one of the two stand- alone versions.

1. TUSS 128K
2. TUSS 64K
3. MULTIFACE TUSS 128K
4. MULTIFACE TUSS 64K

Options 1 and 2 are the two stand alone versions, no extra hardware is needed.

Options 3 and 4 are versions for multiface users only.

If you own an unexpanded 464 or 664 chose the 64K versions 6128 owners or those with a RAM pack may use either, most commonly you would want to use the enhanced 128K version though this corrupts banks 3 & 4 so if you are trying to grab sprites from a 128K game that store sprites in either of these banks you should use the 64K version.

Each version has it's own advantages:

TUSS 128K. The stand alone version is the full version with nothing missed out.

TUSS 64K. The same as above though with no clipboard options, but does not use the extra banks.

Multiface TUSS 128K. This has no Load option but has the advantage of retaining all the colours used in the game make searching much easier.

Multiface TUSS 64K. As above but without any clipboard options. Again this does not use the extra banks.

The other two versions, not listed in the menu, are the same as the stand-alone version but located higher in memory to give Hackit users a choice.

Using TUSS with Hackit.

TUSS was originally designed for Hackit and is therefore totally compatible.

Interrupt your piece of software you wish to hack. Insert the TUSS disc type [D] , [RETURN] to initialize the Disc ROM,[L] to load and enter your filename, from below, in at corresponding load address. Now [CALL] TUSS with the same call address as the load.

Filename	Load/Exec addr	Length
TUSSMULT.128	&0040	&1280
TUSSMULT.64	&0040	&1100
TUSS0040.128	&0040	&1500
TUSS8B00.128	&8B00	&1500
TUSS0040.64	&0040	&1380
TUSS8B00.64	&8B00	&1380

Using TUSS with a Multiface.

Install TUSS from the main menu and press the reset button at the prompt. Now run your game and press the freeze button and TUSS should come on. To exit press the reset button again - Do not exit with [CTRL],[SHIFT],[ESC]. To remove TUSS from the multiface hold down any key and press the freeze button, you may have to do this repeatedly.

The Ultimate Sprite Searcher.

On loading you are presented with following options.

Drive:

Pressing [1] toggles between drives A and B

Load file:

Insert your disc you wish to load from and press a key. Then use the Cursor keys to move the cursor and [COPY] to load that file.

Search memory:

The main part of TUSS you are first given a list of ways to search as discussed in the first part where you can press [H],[V],[S] or [D] you are then given a screen with what will most probably be a collection of meaningless pixels. Using the cursor keys you can increase the width and height of the sprite and if you have chosen the correct memory address a sprite should appear before you.

Left and Right adjust the width of the sprite

Up and Down scrolls through the memory quickly

[SHIFT] & Left and Right increase/decrease the memory byte by byte

[SHIFT] & Up and Down adjust the height of the sprite.

[.] Clears the screen.

[SPACE] enter new address

These keys may seem odd but they will make sense when you use them.

Once you have found your sprite press [COPY] and you can now use left right up and down to draw a box around your sprite press [COPY] again and it will be saved.

Mode:

Change the Mode. MODE 0 allows 16 colours but a low resolution (160*200 Pixels), MODE 1 allows 4 colours Medium resolution (320*200 Pixels) MODE 2 Only 2 colours but High Resolution (640*200)

Set inks:

Set up the inks you wish to search for the sprites in by using the cursor keys to move left and right and up and down to change the ink.

Displayed are all the Hardware inks as given by your Multiface if you chose to take the inks from their "INKS" menu as well as the more common "Grey scale" colours.

Memory View:

Gives a graphic memory dump to allow you to find the sprite data quickly and easily.

Change Memory Address:

Enter a hexadecimal address. This is the address which you will start searching for sprites. Note that for a value such as &A00 - &0A00 must be entered

Change Bank:

Change to any of the 5 banks (0-4). remember that only code from &4000 to &8000 will be affected - 128K owners only.

Save Options.

This gives the following sub menu.

1. Save the sprite as ASCII
2. Save the sprite as Binary
3. Save the whole screen
4. Save as an Advanced Art Studio Window
5. Store to the clipboard (128K versions only)
6. Enter filename to save code under

Option 1 will save the sprite when grabbed as a series of Hexadecimal codes (separated by commas) ready to use in your own source code. This has been written with Arnor's CMAXAM in mind but should be compatible with most assemblers. You will be asked for a location for your 2K buffer. This a free space of &800 (2048) bytes that the computer needs. Select a place where no important data is held. So do not use &000 - &1500 as this is where TUSS is stored or If you are using any 128K version avoid the area from &4000 to &8000.

Option 2 saves the sprite (horizontally) out as a binary file.

Option 3 Will save the whole grabbed screen ready to load into you favourite art package such as *CG-Paint* or *CART Studio*. An Art Studio PAL file will also be saved out with this unless you do not specify a filename extension.

Option 4 Will save your sprite out as a Window file to be loaded into Advanced Art Studio. (Remember to have the extension .WIN) again a PAL file will be saved unless you do not specify a filename extension.

Option 5. Store to clipboard will allow you to place your grabbed sprite anywhere on the clipboard screen you desire for you to build a collage of sprites to save out how ever you wish. 128K only

Option 6. This allows you to enter any filename to save

How to recognise a sprite in memory.

Most hacking programs should give you the option of a Memory Map, Option 6 in TUSS, this displays a graphic representation of what is in the memory.

"Sprite Data" can be recognised by a number of blanks to give it a patchy look. Sometimes you may see diagonal lines or thick vertical lines this is probably also sprite data. Though at times it may only show up as a few blank dots.

Once you have decided roughly where your sprites are press option 7 and enter the memory address you wish to start the search from. This should be approximately the start of the patchy area.

Check you have the correct mode (For games usually MODE 0, or MODE 1 if it is a spectrum port and MODE 2 if you are transferring graphics from another computer or viewing DTP files).

Now press 3 to search for the sprites, this can be any one of the four types discussed above, Horizontal, Vertical, Screen or Double width. The most common of which is horizontal, you should see a mass of coloured dots. Using the left and right cursor keys adjust the width of the sprite and not before long you should see the sprite appear. If you see something that resembles your sprite but is elongated and thinner it needs to be widened with the right cursor, if you see a number of your sprites but shortened and wider the width needs to be reduced with the left cursor. To give you an idea of the correct width think of the sprite you are extracting and how wide it is. The horizontal sprite will be displayed if the height is wrong but not the width, with vertical it is the opposite. With vertical sprites the most important thing to remember is how tall they are, if you are searching for a character in a game start at y=20 or so and work up to about 70 but beyond is pointless. Just remember the size of the sprite. When searching for a horizontal sprite but the sprite is in fact vertical you will see vertical lines.

Graphics displayed as a screen will be spotted easily, it will always be &4000 bytes long and will have a higher proportion of blanks than sprite data.

The Clipboard. (128K Owners only.)

The clipboard is just a blank screen which you can use to grab and place sprites around. This means you can grab a number of sprites from different place in memory or even completely different games and store the sprite on this screen where you can save the sprite out or the whole screen.

To use it select the "Store to clipboard option" from the save menu, and grab it as you would normally. Press [COPY] and you will be taken to a screen when you may position the outline of the sprite using the cursor keys. Pressing copy again will place the sprite.

To view the clipboard select bank 4 with a memory address of &4000 then search memory for a full Screen.

To clear the clipboard press [CLR] and you will hear a beep to acknowledge you have done so. Remember to do this the first time you use it.

Please note that using the clipboard corrupts banks 3 & 4 so any sprite data held there will be over written. As the clipboard is stored in bank 4 it survives a soft rest. This means you may arrange your sprites on the screen, reset, load another game (assuming it does not use bank 4), save files to the clipboard and your old sprites will still be there until you reset by physically turning your CPC off then on again or press [CLR] to clear the clipboard.

Trouble shooting Q&A.

C "I've loaded a file but it still says the length of the file at &1500 is &0000."

The problem is that the loaded file is ASCII, the file has loaded but ASCII files have no header so the length is undetermined.

C "I've selected LOAD FILE and the screen has just gone black."

Try pressing [C] for Cancel - you have a disc missing or read fail message.

C "When using the Multiface version I went to search for the sprites but every thing went black after I selected the method."

A very small number of games (I only know of one: Sgt. Seymour Robot Cop) can detect that the multiface button has been pressed and set all the inks to black, this can not be avoided so you will have to set the inks manually using option 5 SET INKS.

C "I've found the sprite but it seems to miss every other line."

You are in a too low a resolution if you are in MODE 2 try MODE 1 and try MODE 0 if you are in MODE 1

C "My clipboard has vertical lines going down it!"

You must clear the clipboard when you first use TUSS press [CLR] from the main menu.

C "Okay, Done that now how do I view the clipboard and grab sprites from it."

Select bank 4 (Option 8) and change the memory address to &4000 (Option 7) now Search Memory (Option 3) and select full screen ([S]).

C "I've installed TUSS into my multiface and now I want the Multiface menu back."

TUSS has been known to stay in memory for days with the computer turned off. Reset your machine and hold down any key as you press the freeze button, this may not work first time so keep doing it until it does.

C "When I wish to save as ASCII it saves but I just get stripy lines corrupting most of the sprite"

When using the 128K versions do not select a 2K buffer any where between &4000-&8000 unless it can be avoided. If you must depending on the size of the sprite you should be safe if you select it at &6000 onward.

How to use the sprites in your own code.

Included on the disc you will find a couple of assembly language source files (*.SRC) this includes a standard sprite routine and a demonstration of this "DEMO.SRC".

For BASIC users included is an RSX to print grabbed sprites (saved as Binary) to execute this enter

```
| SPRITE,width,height,screen address,data address.
```

```
| SHELP prints this to refresh your memory.
```

"RSXDEMO.BAS" is a demonstration of this.

When using the assembly language routine save the sprite out as ASCII the either merge it into your source code or use Maxam's read command to load it in.

If you are using the RSX save the sprite as Binary and load it in at any memory address you wish.

For more information on sprites and assembly language sprite routines I strongly recommend one of the Pipeline Tutorials - TP 06 The world of Sprites this is available at 3 + an optional extra 3 if you wish the accompanying disc (or 1 if you provide the disc). This detailed guide, one in a series of many highly acclaimed by Amstrad Action, goes through all the principles of sprites and sprite routines including animation and movement, as well as using sprites in your own games and several collision detection algorithms and routines. You need not be an experienced programmer either, as as with all the tutorials they are written in a simple non-technical manner going right back to basics as to not put off the inexperienced yet provide the experienced with plenty of useful ideas and routines that could well prove invaluable.

This tutorial and more are available exclusively from us at Sentinel Software please write if you require further information on this or any other of our tutorials.

TUSS by-products.

Not only can TUSS be used to extract the sprites from CPC game it also has many other uses here are just a few.

Program development.

One of the principle reasons TUSS was designed was for program development the save as ASCII option was put in for the purpose. Draw your sprite as normal save it as a screen the with TUSS you can grab it and save it ready to use in you own programs. (See the section on using sprites in your own programmes or DEMO.SRC)

Retrieving Art Studio screens after crash.

Picture this, it's 2am you've been working on you CPC for five hours drawing a screen with Art Studio when suddenly you Romboard wobbles and crashes the machine or you reset it in a fit of rage. Argggg, time to start all over again. Well not any more. Boot up Tuss select bank 2 Memory Address &4000, Save options - Save the whole screen, Search Memory - full screen and press [COPY] twice and hey presto one screen retrieved!

Saving a Multiface screen as a normal screen.

Screens saved with the multiface are renowned for being incompatible with a number of art packages but doing the same as above with a couple of change you can save the file out as normal. Load the file using option 2, search as a full screen and save it out as a screen.

Viewing graphics files.

TUSS can be used to view most CPC graphic file formats, these include 17K screens, Art Studio windows, DTP CUTout files and Micro Design Icon files.

Graphics conversion.

Why stop at CPC graphics TUSS can quite easily grab other computers graphics with alarming success. The PC's BITMAP files (.BMP) can be converted. It works best with "Monochrome bitmaps". Firstly Convert the PC file using DOSCOPY, 2in1 or other such program and load it in. Then search for a sprite in MODE 2 horizontal or Double width. This works fine but for some reason the screen appears upside down - still can't be perfect, eh?

Another PC file format .CLP files - Windows clipboard files also transfer well again in MODE 2 select Horizontal or Double width. Windows icon files (.ICO) work too but are so small it is barely worth the effort.

As well as screens I have successfully converted actual PC game sprites by copying across the correct files and search though these in MODE 2. Again searching for horizontal or double width sprites.

Various Atari ST screen files have also been recognisable namely IFF and NEO files as well as sprite files from

Dictionary of Terms.

- &xxxx.....A 16 bit hexadecimal number, a number ranging from 0 to 65535 or &000 to &FFFF
- ASCII.....The American Standard Code for Information Interchange, an agreed set of code which give a number for each character.
- BASIC.....Beginners All-purpose Symbolic Instruction Code. The programming language supplied with the CPC.
- Bank.....The extra memory with a 128K machine is split into 4 extra lots of &4000 bytes each known as a bank
- Byte.....A single character.
- Data.....A string of bytes.
- Memory Map.....A graphical representation of the computer memory where an value greater than 0 is represented as a dot.
- MODE.....The CPC has three resolution modes. MODE 0 allows 16 colours with 160*200 Pixels, MODE 1 allows 4 colours with 320*200 Pixels and MODE 2 only 2 colours with 640*200 pixels.
- Multiface.....Romantic Robot's hacking and copying hardware add on.
- Multi-load.....A program that will load in a section of code after the previous has been used. Ie. Levels in a game.
- PAL.....Pallet file used to store the inks in Rainbird's Advanced Art Studio.
- Pixel.....The smallest element that goes into making a picture on a screen. ie a single dot.
- Hackit.....Siren Software's hacking hardware add-on.
- Hardware ink..The CPC has two sets of values for inks the commonly used grey scale (0-26) and the Hardware inks these are rarely used but are by the Multiface.
- Hardware Reset..When then machine is reset with the Hardware. Ie. a reset button on your multiface.
- RAM.....All the memory in you CPC you can store data in.
- Resolution.....The number of pixels possible to fit on a screen.
- Soft Reset.....Resetting the machine via the software, most commonly with [CONTROL],[SHIFT] & [ESC].
- Source Code....The uncompiled program code.
- Sprite.....A numerical representation of a graphic.
- TUSS.....The Ultimate Sprite Searcher.