

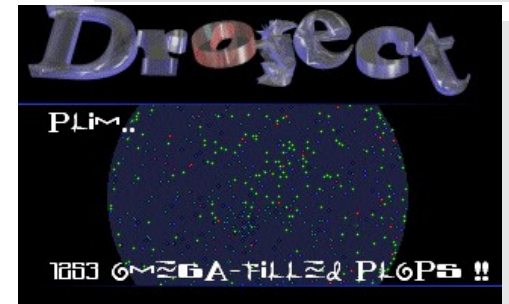
My Speccy Journey

Jari Komppa 2020

<http://iki.fi/sol>

Who am I to talk?

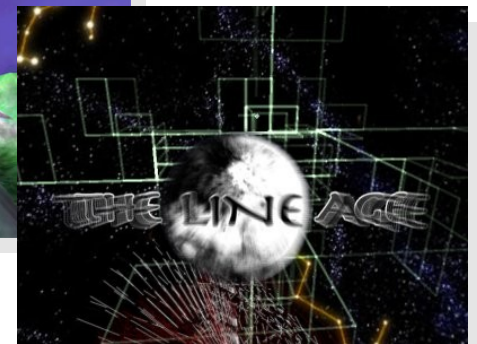
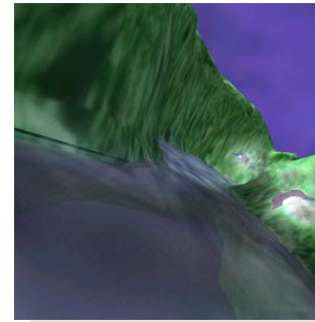
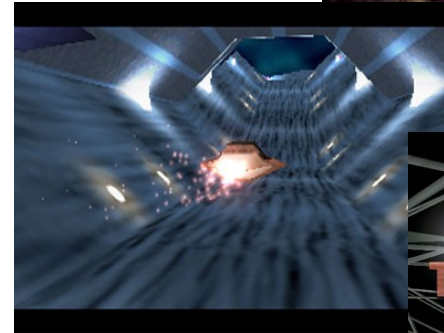
- Sol of Hysteria (1992-1994)
 - Hysteric Overload 1992
 - Silence 1994
 - Droject 1995 (unreleased at the time)



<http://iki.fi/sol>

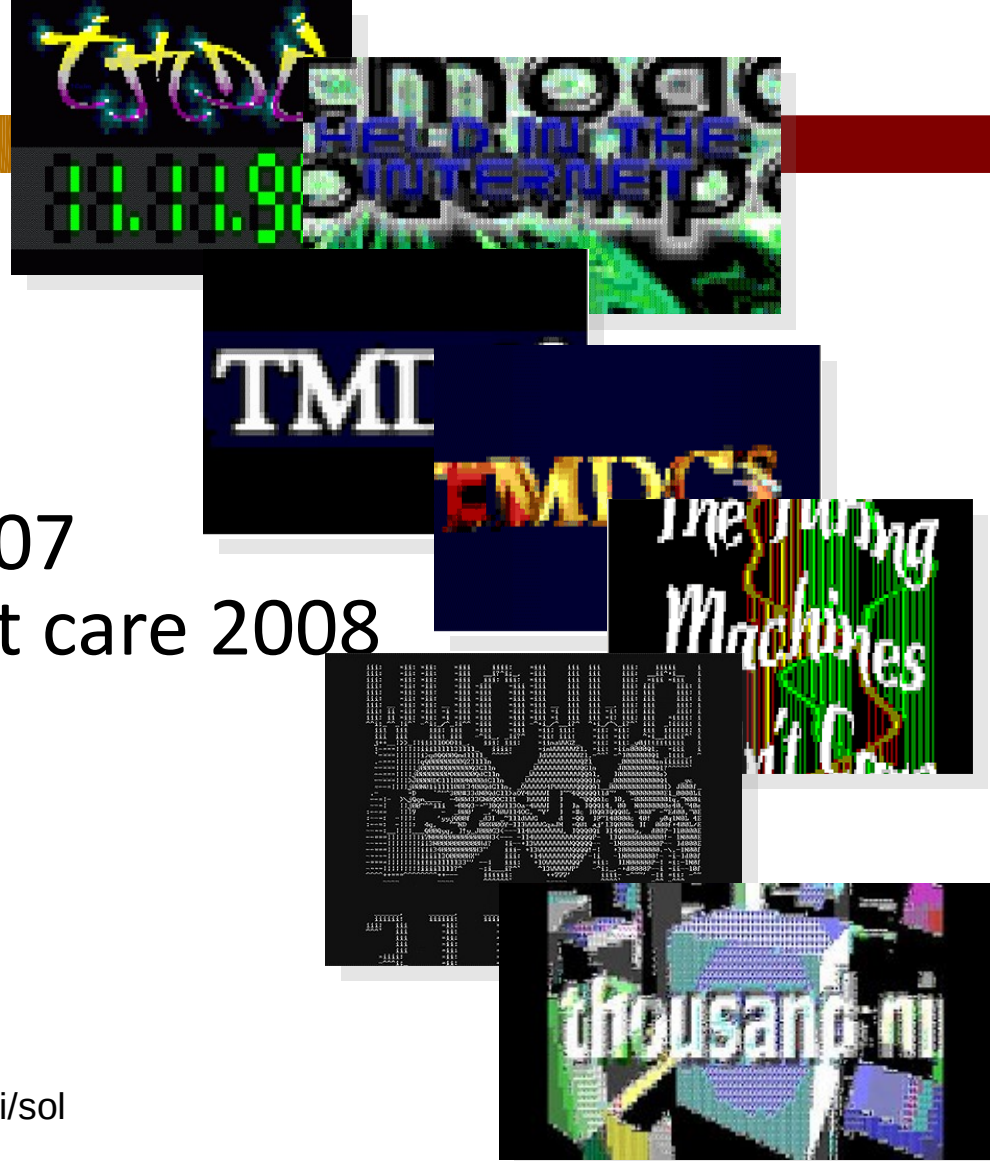
Who am I to talk?

- Sol of Trauma (1996-...)
 - Dee music disks 1996-1997
 - Mindtrap 1997
 - Gateways 1998
 - Traumatique 1999
 - Stuff I whacked together when I was bored 2000
 - The Line Age 2005



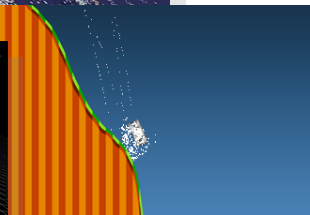
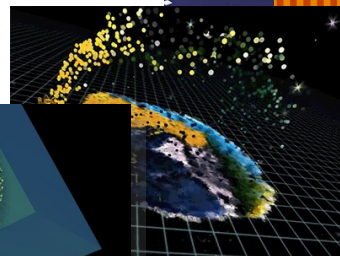
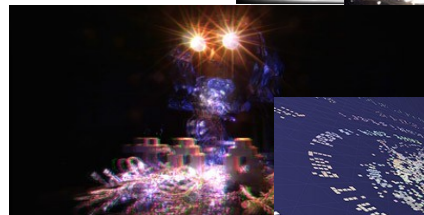
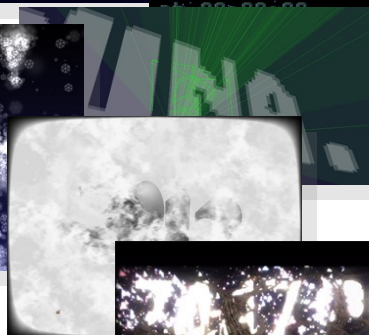
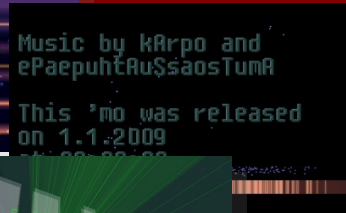
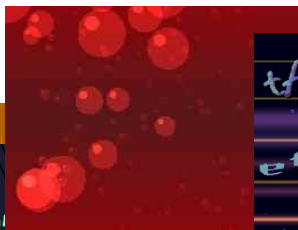
Who am I to talk?

- Textmode productions
 - TMDC invitations 1996-2007
 - The turing machines didn't care 2008
 - Litterae Finis 2012
 - Light 2017



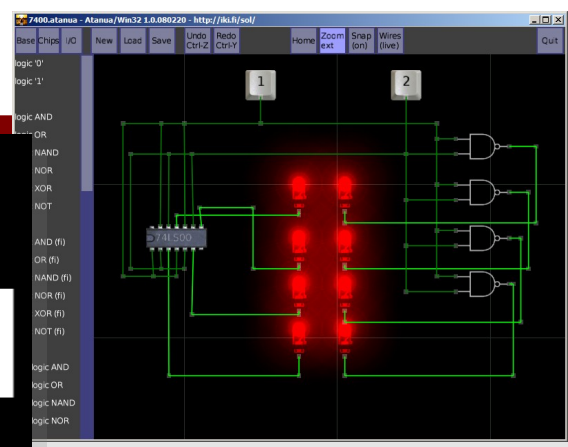
Who am I to talk?

- ToiMISTokEiju of tAAat
- New year demos since 2006



<http://iki.fi/sol>

Who am I to talk?



- Other notable stuff.
 - Atanua logic simulator
 - SoLoud audio engine
 - Extensive website



<http://iki.fi/sol>

Enter the Speccy

- I was born 1975
- ZX Spectrum was released 1982
- I was 7, in school
- Mostly for games..
- (and basics of BASIC)



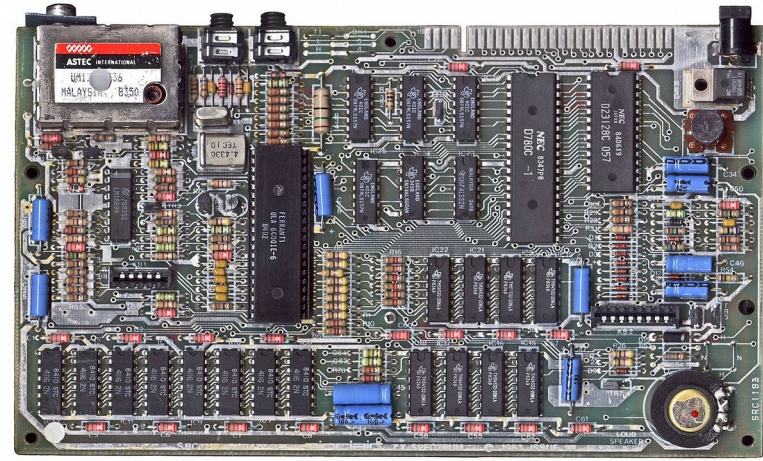
Enter the Speccy

- Fast forward to 2015
- I was 40
- Got inspired to learn more about the speccy



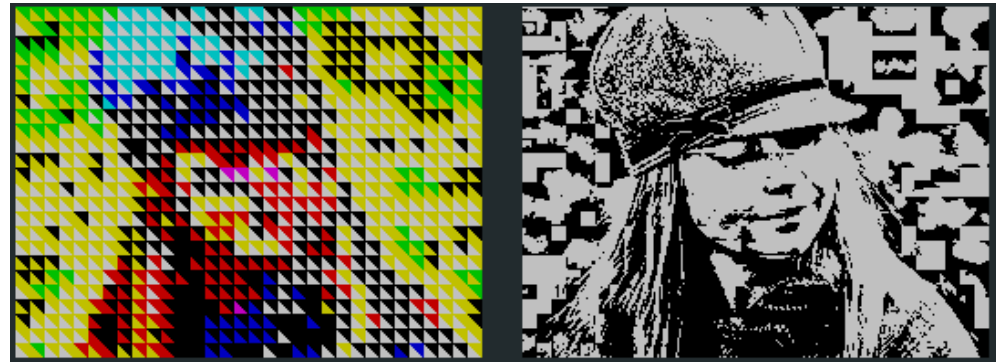
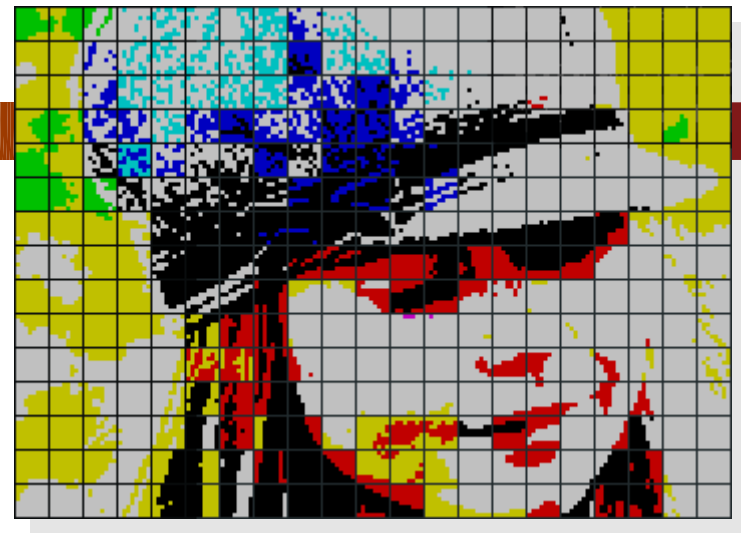
ZX Spectrum 48k Overview

- Cheap and simple design
 - Zilog Z80A 3.5MHz
 - (Or, often, a cheap clone)
 - 16KB ROM, 48KB RAM
 - 16K contended (“chip”),
32K uncontended (“fast”)
 - Piezo speaker
 - TV Output
 - ULA

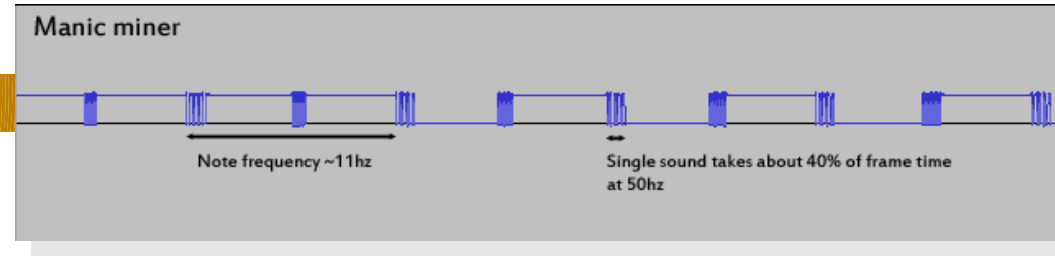


ZX Spectrum 48k Display

- Single graphics mode:
 - 256x192 non-linear bitmap
 - 32x24 linear color map
 - 8x8 cells with 2 colors
 - 15 color palette
 - Both colors must be dark or bright
 - Bright black is still black
 - Blink attribute
 - Border color



ZX Spectrum 48k



- Audio: Bit-banged piezo speaker
 - Takes 100% of CPU while producing sound
 - Most games had “chirps” as sound effects
- Data storage: C-Cassette audio
 - 1023-2046 bits per second (1364 on average)
 - Loading 32KB takes over 3 minutes

ZX Spectrum 48k Input Devices



- Rubber “dead flesh” keyboard
 - 2 layer membrane, 8x5 matrix
- No standard joystick out of the box
 - Multiple joystick standards
 - Mostly “Atari compatible” hardware
 - Different software interfaces..

<http://iki.fi/sol>



ZX Spectrum 48k Memory Map

Start	End	Content
0x0000	0x3FFF	ROM
0x4000	0x57FF	Screen memory 256x192 pixels = 6144 bytes
0x5800	0x5AFF	Color data 32x24 = 768 bytes
0x5B00	0x5CCA	Reserved for basic thingies
0x5CCB	0xFF57	Available memory
0xFF58	0xFFFF	Reserved for basic thingies

16384B ROM
6144B display bitmap
768B color data
42240B "free"
32768B uncontended
(16K speccys had
9472B "free")
No point in returning
to BASIC, though..

<http://www.breakintoprogram.co.uk>

ZX Spectrum 48k Memory Map

Start	End	Content
0x0000	0x3FFF	ROM
0x4000	0x57FF	Screen memory 256x192 pixels = 6144 bytes
0x5800	0x5AFF	Color data 32x24 = 768 bytes
0x5B00	0x7FFF	Available contended memory
0x8000	0xFFFF	Available fast memory

- This is all memory you get:
All your code, data, variables, buffers, stack..

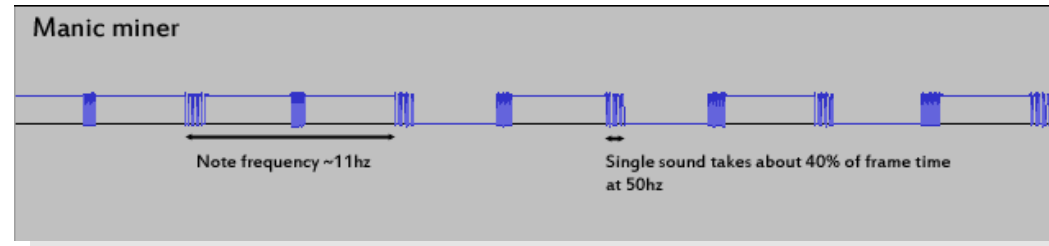
ZX Spectrum 48k Timing

- Single interrupt, triggers at retrace
 - Enable interrupts, HALT
- Default ISR talks to ROM
 - Custom ISR requires trickery..
 - ISR address taken from a 257B table randomly
 - Due to a hardware oddity
 - Destination low and high byte must be equal (eg, table of 0xFD:s will jump to 0xFDFD)

My 48k Development Toolchain

- Audio: Minimal, usually Shiru's BeepFX
- Graphics: my Image Spectrumizer
- Packaging: my Mackarel
- Code: SDCC (C and Z80 Assembly)

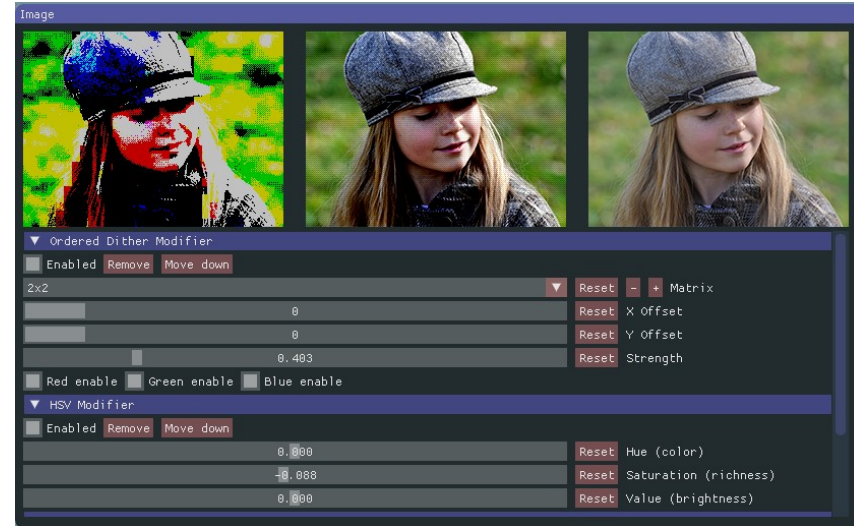
My 48k Development Toolchain



- Audio
 - Music largely out of the question
 - Still image or very minimal animations
 - Short chirps for real-time games
 - Move-based games may have longer effects taking full frame
- See also: BeepFX by Shiru
 - <http://shiru.undergrund.net/software.shtml>

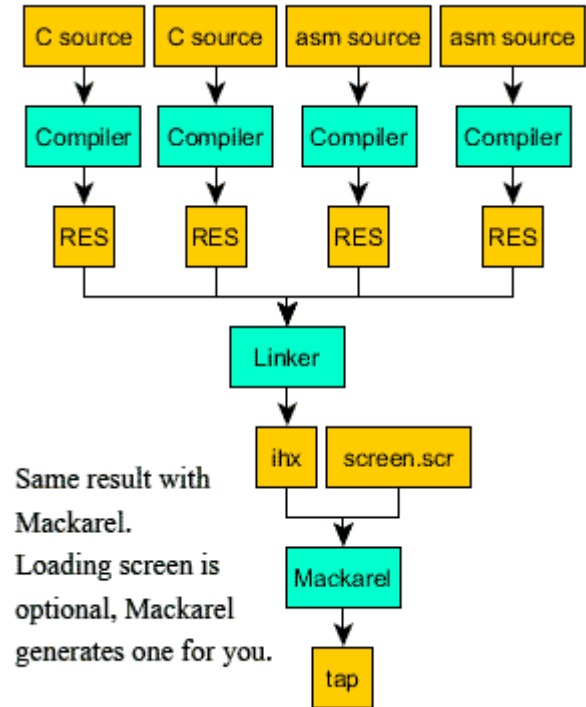
My 48k Development Toolchain

- Graphics
 - My Image Spectrumizer
 - Can edit graphics in photoshop, see changes after filters
- For sprite data, simpler tools that convert png:s to tables
 - Pre-shifted sprite data



My 48k Development Toolchain

- Packaging
 - Using my Mackarel tool
 - ZX7 Compressed application binary
 - Compressed loading image
 - Output as a .tap file
 - Result loads faster than commercial titles back then
 - .tap file can be loaded into emulator or turned into a waveform to load on real hardware



My 48k Development Toolchain

- SDCC
 - <http://sdcc.sourceforge.net/>
 - Creates “ok” code
 - Easily beaten by human in optimization
 - Generated code is readable, often good starting point for optimizations
 - SDCC’s assembler uses slightly different syntax than most z80 assemblers, and lacks features.

My 48k Development Toolchain

- Why C?
 - It's not assembler.
 - Generally easier and faster to write code
 - Complex algorithms can be developed on PC with debugger
- Why not C?
 - It's not assembler
 - General loss of control, code size, stack, absolute location of code..

My 48k Development Toolchain

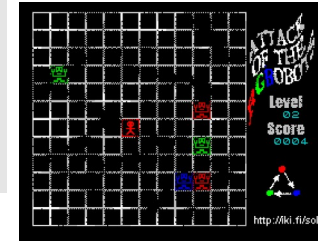
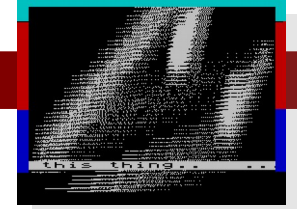
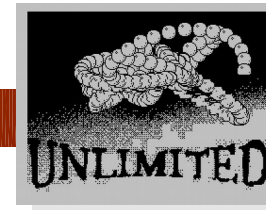
- SDCC has a lot of extensions
 - Variables can be declared at absolute addresses
 - Ports can be accessed without inline assembly
 - Code location, start address etc.
- No standard library though
 - No division, no multiplication, no malloc, no printf
 - Alternative is to use z88dk which has standard library and other doodads

My 48k Development Toolchain

- Memory map considerations
 - Dude, where's my code?
 - Contended memory
 - Where's the stack?
 - Where's the data?
- Do we have overlaps? That would be bad..
 - Bane of C programs; less control
 - Mackarel helps, somewhat
 - Linker log analysis scripts

My 48k Releases

- Early experiments
 - Some demo effects, never released as a demo
- Games
 - Solargun, a shmup
 - RGBobots, a LD48 entry
 - Mazeract, a 3d maze puzzler
- Game engines
 - QT48k trivia engine
 - MuCho multiple choice adventure engine



After 48k Spectrum

- I got a 128k spectrum +2
- Fixed it so it runs
- Never really did anything with it...
 - Practical additions on top of 48k:
 - Double buffering
 - AY-3-8912 sound chip
 - Some additional banked memory
 - No internal speaker (audio through TV)
- What Next?



ZX Spectrum Next



<http://iki.fi/sol>

SpecNext, historical perspective



- 48k speccy was third “ZX” from Sinclair
- Widely cloned, officially and unofficially, with enhancements
- Official 128k sequels (thanks, Spanish tax laws)
- Wide range of expansions thanks to limited features and accessible expansion port
- ZX Spectrum Next is “best of all worlds”, with additional enhancements

SpecNext Hardware



- Z80 with additional instructions, 3.5MHz – 28MHz
- 2048KB memory (1792K directly usable)
- Graphics: multiple modes, 256 color palette modes, hardware sprites, layers..
- Video out: VGA, RGB, HDMI
- Audio: Three AY-3-8912 chips, DAC
- Media access: SD card, Wifi module
- Nice keyboard, two joystick ports, PS/2 mouse..

SpecNext Environment



- You're not alone anymore
 - Can also write programs that return to OS
 - “.dot” programs; small programs meant to be called from BASIC
 - Suddenly those BASIC-reserved things matter!
- Can still take over whole machine
 - “.nex” programs; large programs that usually require reset to exit
- Old formats still work: can make .tap files, or use various emulator snapshot formats

Memory Mapping



- Z80 can see 64k of memory, system has 2 megs, what gives?
 - Banking; a mess of history:
 - 128k speccy ram banking, 16k banks
 - Special “allram” mode originally made for CP/M
 - Separate mechanism for ROM paging...
 - Next banking system: simple to use and 8k at a time
 - ...but may get overridden by older methods

Memory Mapping



- What was that about 1792K?
 - Back to history: the 256K are taken by Multiface RAM, DivMMC RAM and various ROMs
 - In short, want disk access? Deal with it.
 - About 128K of those are relatively easy to access.. but more difficult than the 1792K.
- Thus, 1792K ought to be enough for everybody.

Feature Control



- 48k speccy had 1 port that was used for everything: reading keyboard, border color, audio output..
- Next has over 25 ports and about a hundred configuration registers. There's a bit more going on
 - Still manageable, and documentation is great

Emulating Specnext



- State of emulation:
 - No perfect emulators exist
 - Emus require a lot of processing power
 - Transferring files to emulators is nontrivial
 - Emulators have to emulate disk hardware
- Two emulators:
 - Cspect
 - ZesarUX
- I use neither

Developing on specnext



1. Copy file to SD card
2. Eject SD card (to avoid corruption)
3. Insert card in next
4. Reboot next (next notices card was gone)
5. Run
6. Eject SD card
7. Repeat

...Or, use NextSync. (or a wifi-enabled SD card)

NextSync



- There was no easy way to transfer files to the next using wifi.
 - Server approx. 300 lines of Python running on PC
 - Client is a dot command running on Next
1. Build on PC
 2. Run `.sync` on next
 3. Run program
 4. Repeat

Specnext dot commands



- Dot commands are kinda like DOS .com files
 - All code, no header
- Always loaded at specific address
- Might exit cleanly and be used from BASIC
- Max 8192 bytes (can be extended with some work)
- Access to OS, commandline parameters, etc
- Handy for small tools

Specnext nex programs



- Nex programs are meant for big programs
 - No need to return to BASIC (but that IS possible)
- Nex loader is a dot program, sets next in a (more or less) known state, loads program into memory
- Nothing stops programs from loading more data off disk

Specnext development



- Memory map and planning still matters
 - Need operating system? Keep ROM around
 - Can be mapped in and out as needed, though
 - Where's your code? Where's your stack?
 - Swapping data in and out – pretty simple
 - Swapping code in and out – needs more care
- No contended memory anymore
 - Some memory banks are special (ULA mem etc)

Specnext development



- “Background” color if all layers invisible
- The ULA screen is still a thing
 - Timex enhanced ULA modes available too
- Lo-res 128x96x8 mode
- Sprite layer
- “Layer 2”: 256x192x8, 320x256x8, 640x256x4
- Tilemap layer (“text mode”)
- Separate palettes
- Combinations of above can be used simultaneously

Specnext development



- 256x192x8 is the “mode 13h” of next
- 48k of data. Remember, z80 sees 64k..
 - Prepare to map things around a lot
 - Each 8k bank is 32 lines
- 320x256x8 takes even more space, and rows and
- columns are swapped

Specnext development



- Palette: 256 colors out of 512
 - 8 by 8 by 8
 - Only 8 pure shades of grey :(
- All good old palette tricks are back
 - Cheap in/out fades, desaturate
 - Palette cycling
 - Crossfading between 16 color images (or 8 and 32)

Specnext development



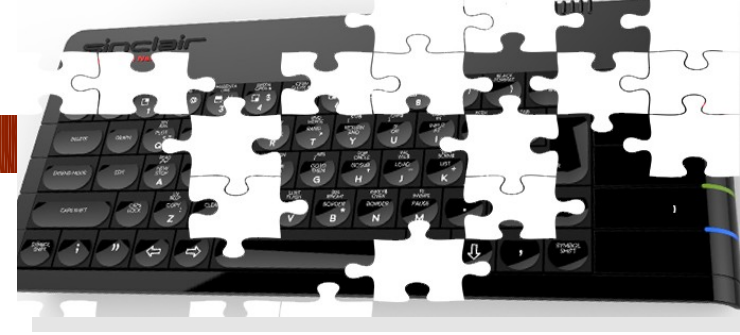
- ISR
 - The old system still works
 - Default ISR points at ROM
 - ..but now we can get rid of ROM
- Interrupt line can be changed
 - With some work, interrupt rate too

Specnext development



- A lot of potential for hardware trickery
 - Copper
 - Various graphics layers
 - Offset registers, page changes
 - The ULA layer may be crappy, but it's there

My Specnext Toolchain



- Doesn't really exist yet..
 - SDCC for code
 - Open questions; code overlays?
 - Graphics should be easier this time
 - Audio – can play AY reg dumps
 - In the long run, a VST solution?
 - Packaging; a “mackarel” for the next?
- NextSync was a big puzzle piece
- Can do other dot programs if they're needed

The Slide with the Links

- <http://iki.fi/sol> – my stuff
- <https://github.com/jarikomppa/speccy/> - my 48k stuff
- <https://github.com/jarikomppa/specnext/> - my specnext stuff
- <https://www.specnext.com/> - specnext portal
- <https://www.specnext.com/forum/> - specnext forum
- <https://wiki.specnext.dev/> - specnext development wiki
- <https://gitlab.com/thesmog358/tbblue/-/tree/master/docs> – hw docs
- <https://www.facebook.com/groups/specnext/> - facebook group
- <http://sdcc.sourceforge.net/> - sdcc
- <http://sdcc.sourceforge.net/doc/sdccman.pdf> – sdcc manual
- <https://github.com/z88dk/z88dk> – z88dk
- <https://z88dk.org/forum/> - z88dk forums
- <https://discordapp.com/invite/HdRvZVw> – specnext discord