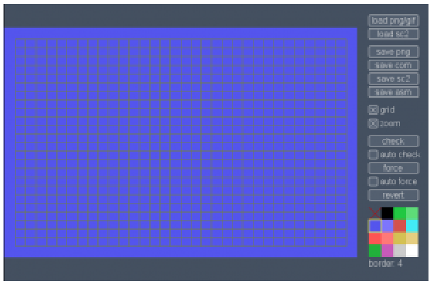
# .Markku Reunanen.

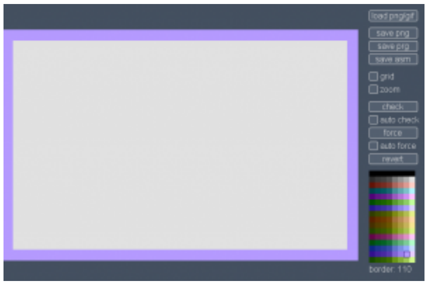
Etsi

- Blogi
- ¡Tequila!
- Tietotekniikka
- Projects
- Tietoja

---

## Pixel Polizei





*"Systeemin vartija se poliisin on työ / Poliisi hippejä pampulla lyö"*

## News

- 8.7.2017: What, updates? Yes! Fixed BG colors should work now with Plus/4 multicolor. Thanks to Ptoing for the bug report. In addition, the latest source-only features are now part of the binary distribution.
- 6.5.2016: Source-only: You can set your own error color with ERRORCOLOR in prefs.txt
- 31.5.2016: Source-only: Loading transparent images doesn't create a mess anymore. Fix kindly submitted by Dr. TerroZ.
- 25.5.2016: Another quick binary release with:
  - HRS loader and writer for Le Oric
  - ColecoVision screen 2 support
  - Name table supported on the MSX when loading SC2 images
  - Improved MZ-800 palette and viewers
- 23.5.2016: A brand new binary release with the following additions:
  - Sharp MZ-800 mode 1/2 (16/32k)
  - Sharp MZ-700
  - Panasonic JR-200
  - MSX screen 3
  - Smarter multicolor for the C-64 and Plus/4
- 17.5.2016: Aaand the first release is out!

## What?

*Pixel Polizei* is a tool for the retro/scene/retroscene artist. It lets you automatically check whether an image conforms to the color limits of a particular oldschool platform and save the result in certain native formats. The underlying idea is that you can use whatever paint program you happen to prefer for creating the images and observe – almost real-time – if you've made a mistake somewhere. One more fundamental bit is that the tool is multiplatform unlike many others.

Pixel Polizei is *not* a paint program or a generic image format converter, even though the aim is to implement at least the most typical file types for the supported platforms. Neither is it meant to be used as a converter for random family photos. You can use it for that purpose too, but the end results won't be as good as with dedicated tools. For converting various old formats to modern images check out the Recoil project.

# Usage

There's not much to it: *load* an image, *check* whether it's ok and save it in another format. In the case of errors you can either fix them in your favorite paint program and retry or *force* the colors to comply. When a piccy is loaded it is automatically converted to the native colorset without dithering, so do that kind of magic elsewhere if needed. Furthermore, remove any borders that you might have or the image will be scaled incorrectly.

If the loaded png or gif is not of the native resolution, it will be down- or upscaled as necessary, which might or might not produce the desired result. Therefore it's generally better to do the scaling elsewhere. C-64 multicolor and CPC mode 0 have very flat pixels, so drawing a 320×200 image will probably be easier; every second column will just be discarded.

Checking and forcing can be done automatically. In that case Polizei works according to its name and constantly polls whether you've saved the file anew. If there is an updated version, it's loaded and checked automatically. This way you can quickly find and fix color clashes – keep the Polizei window always visible on another screen or in some corner.

The *zoom* toggle lets you look at the image closer, and a *grid* is available for you as well to better see the character block locations. If the platform has a selectable border color, just click on the color selector.

Check out the "palettes" directory to find the exact colors used on each platform. Drawing with those colors ensures that there's no complications when importing the image.

## Keyboard shortcuts

- c – check
- f – force
- g – grid on/off
- r – revert
- z – zoom on/off

In some modes you need to press shift when clicking the color selector to set the background color.

## Settings

Some settings can be controlled through *prefs.txt* which should reside in the same directory as the program itself. The following options can be used:

- **ZOOM=x** where x is 1 or more. Note that with ZOOM=1 pixels are never stretched to mimick a real aspect ratio. The larger the value the slower it gets. This is all about UI zooming, not the zoom function of Polizei.
- **GRID=x**, set to 1 to show the grid when starting the program.
- **PATH=x** where x is the absolute path of the directory where your images can be found. Leave blank to go with the default. Do not use quotation marks.

- **ASPECT=x**, which can be one of PAL, NTSC or SQUARE. Used for stretching the pixels like the real machine would. Not relevant on all platforms.

# Supported Platforms

Next follows a list of the supported machines, graphics modes and file formats, coupled with platform-specific quirks (if any). New file formats and machines will most likely be added later.

## Amstrad CPC

The CPC is an exceptional 8-bit as you don't need to care about color clashes: any pixel can be of any color (16 colors in mode 0 and 4 in mode 1). It also has a palette of 27 jolly happy colors to choose from. I found myself wondering what the heck they were thinking when designing the bitmap layout like that.

- Supported modes: Mode 0 and 1, with or without overscan (160×200/192×272 and 320×200/384×272).
- Supported formats: bin and asm. When copying the files to a dsk image the needed start/load address is $6000 for ordinary images and $100 for overscan pics.

To display the image on an emulator or a real machine, you need to get the file to a floppy – most likely a DSK image. There's *ManageDSK* for Windows and iDSK for the rest of us. *iDSK* probably won't compile out of the box, but just keep adding *#include <stdio.h>* to the offending source files until it does. After that you should be able to do this:

```
iDSK mydisk.dsk -n
iDSK mydisk.dsk -i piccy.bin -t 1 -e 6000 -c 6000
```

And run the binary from BASIC with:

```
run"piccy
```

## ColecoVision

This console is so close to MSX (same CPU and VDP) that it was easy to add trivial support for it.

- Supported modes: screen 2.
- Supported formats: col and asm. MSX sc2 files can be loaded or saved, too, since it's the exact same mode.

COL is essentially a module dump that you can feed to any emulator or flash cart. I'm not sure if there's much use for this particular converter, but at least you can show pics on your old hardware – a nostalgic birthday gift for a Coleco fan maybe?

## Commodore 64

This old workhorse has been explored over and over again throughout the years. From a graphician's perspective there are dozens of curious hacks that let you display more colors and pixels than what the standard modes allow. Use the well-known Pepto palette to ensure a faithful conversion.

- Supported modes: hires and multicolor. At least basic FLI support will probably follow at some point, but as there's a plethora of hacked modes, don't expect all of them to make an appearance.
- Supported formats: (Advanced) Art Studio, Doodle, Koala, prg and asm.

There is automated logic which tries to choose the multicolor common "background" color wisely, but as it doesn't always work out, you can also select the color yourself (shift-click or rmb) and turn *fixed bg* on.

The resulting PRG files can be run directly by using VICE, 1541 Ultimate or SD2IEC. For some situations you may want to create a floppy image, which can be done with for example c1541 that comes with *VICE*:

```
c1541 -format mypic,0 d64 mypic.d64 -attach mypic.d64 -write piccy.prg piccy
```

## Commodore Plus/4

It wasn't a biggie to add support for the Plus/4 after getting the C-64 side working. A lot of colors available for you (121 shades) and practically square pixels in hires. Not that different otherwise to the C-64, except in multicolor you have 2 global colors and only 2 free colors per block.

- Supported modes: hires and multicolor.
- Supported formats: Multi Botticelli, prg and asm. Name Multi Botticelli files *m.something.prg* so that the original program finds them too.

There is automated logic which tries to choose the common "background" colors for multicolor pics, but you can also select the color yourself (shift-click and rmb) and turn *fixed bg* on.

Running the resulting PRG files is pretty much the same as on the C-64. See above.

## MSX

A plain MSX1 can do 256×192 pixels with 15 – not 16 – washed-out colors. PAL machines produce a really flat "widescreen" image, whereas the NTSC ones are more or less 4:3. MSX2 computers could display a lot more colors at better resolutions, but at least as of now they're not supported, as they're not that special from a conversion point of view.

- Supported modes: screen 2 and 3. S2 has otherwise free color placement except that a block of 8×1 pixels can only have two colors. S3 is blocky and has free pixel placement.
- Supported formats: sc2, sc3, com and asm. Name table supported with sc2, but sprites not.

The com files are normal MSX-DOS programs that you can copy to a floppy or an image for running them on a real machine or an emulator. The good old wrdsk will do the trick:

```
wrdsk myfloppy.dsk msxdos.sys command.com piccy.com
```

SC2 files are essentially screen memory dumps that can be directly loaded from BASIC like this:

```
bload"piccy.sc2",s
```

## Oric

Le Oric's graphiques are completely different to almost any other machine. It's a bit like teletext in some sense: you can either show pixels or change the paper/ink color. To make it even more complicated, any 6×1 block can also be shown in inverted colors and the bottom of the screen (3 character rows) is hardwired to text mode. Read this.

- Supported modes: hires. 240×200 pixels with 8 colors. Setting the border color will only affect the last three text lines.
- Supported formats: tap, hrs/hir and asm. Apparently there exists something called hrs/hir, but nobody has taken the time to document the format.

The tap files can be run from basic with a *CLOAD""* or *CLOAD"PICCY"* depending on the emulator. Use the *tap2wav* utility to turn the emulator file into audio that can be played back to the cassette port on a real machine. I'll look into disk files later, but no promises yet.

HRS (HIR) files are simple screen memory dumps saved from BASIC. They can be displayed by switching to *HIRES* and then simply cloading the file as above. To save one on a real machine, try this:

```
CSAVE"PICCY.HRS",A#A000,E#BF3F
```

## Panasonic JR-200

Not many have ever even seen this failed Panasonic venture. See here for some technical discussion. From an artist’s standpoint it’s not unlike the Spectrum, except there’s no brightness bit and you can only have 320 different 8×8 character blocks. In addition, there’s blocky 64×48 pseudographics without attribute limits – both “modes” can be mixed freely.

- Supported modes: 256x192x8 hires.
- Supported formats: cjr and asm.

Panasonic emulators are scarce, but Virtual Panasonic JR-200 will get you started at least. CJR files can be loaded to the emulator or converted to wavs for playing back to the cassette port. See here again. If your goal is to convert Spectrum images to the JR-200, check you-know-where for an interactive tool aimed specifically for that. The images can be displayed as follows:

```
mload:a=usr($1000)
```

## Sharp MZ-700

The MZ-700 is a slightly revamped consumer version of Sharp’s old microcomputer series. 64k memory, color graphics and… wait, there’s only 80×50 pseudographics with a 40×25 color grid.

- Supported modes: 40×25 character approximation and 80×50 pseudographics.
- Supported formats: mzf and asm.

When approximating pixel graphics with characters you may gain a slight advantage by including the absurd 2nd character set in the conversion. Beware, however, that the output won’t work correctly on Japanese machines that used the 2nd set for something useful instead of various 2×2 character faces and Pacmen.

Pretty much any emulator supports MZF files, which you can load from the monitor by typing L. For a real machine take a look at mzf2wav.

## Sharp MZ-800

A lot more capable than its predecessor, the MZ-800 can display 320×200 pixels with four colors or 640×200 with two. Rather exceptionally the graphics are planar and thus pixel placement is free. Add another 16k of VRAM and you get 320x200x16 and 640x200x4.

- Supported modes: mode 1/2. 320x200x4 on a stock machine, 16 colors with 32k VRAM.
- Supported formats: mzf and asm.

Like with the MZ-700, emulators support MZF loading. In addition to the monitor method you can also choose option ‘C’ from the main menu. mzf2wav is again your friend when dealing with real machines.

## ZX Spectrum

The British miracle can display 256×192 pixels with 15 different colors, but in a block of 8×8 you can only have two different colors, which makes the machine a true giant when it comes to color clashes. Furthermore, you can’t have bright and dark colors in the same block.

- Supported modes: eh… there *is* just one.
- Supported formats: scr, tap and asm.

An scr file (also known as SCREEN$) is a simple memory dump of the Spectrum screen and supported by various tools and emulators. A tap file can be loaded by emulators or from a card reader such as *divIDE*. You can also convert .taps to audio using *tap2wav* and then play them back to a real Spectrum through the cassette port. *LOAD””* is all you have to type.

## Download

The latest stable binary is available here: ppolizei.zip. If you're after the latest features and bugs, check out *svn://kameli.net/marq/ppolizei*. Because the program is written in Processing you – unfortunately – need to install Java 7 or newer to run it. Mac users should visit *System Preferences – Security & Privacy* if you see something like "blah blah is damaged and can't be opened". Allow "any" software to run to correct the problem.

If all else fails, download Processing 2.x (3.x won't work at least as of yet) and run from source: copy the "ppolizei" folder from "src" to your sketchbook and open ppolizei (.pde if you're using the file selector). Press the play button and enjoy.

It shouldn't be too tricky to add new file formats or even totally new machines. Contributions are welcome, but let's stick to a 4-char tab to keep the source uniform. The code is released under the liberal WTFPL license.

## Credits

- Most code and design: Marq/Fit^L!T^Dkd (see also: PETSCII)
- Several templates and other help: Dr. TerrorZ/L!T (see also: Multipaint)
- C-64 palette: Philip "Pepto" Timmermann
- MZ-800 palette: Chaky

---