SVG graphics for the Schneider Joyce

In computer graphics, a distinction is made between two types of images: Bitmap and vector graphics. The former store the color of each individual pixel in a rectangular grid, the latter store a description of the image objects such as lines, circles, squares, etc. If you want to enlarge or reduce (scale) the images, this can only be done without loss with vector graphics, as the image objects can be recalculated to the desired size. With bitmap images, you are familiar with the “steps” in slanted lines, which are particularly visible when enlarged - more or less clearly depending on the enlargement algorithm.

vector graphics

bitmap graphics

The internet is full of images; most of them are bitmap graphics. But there are also many vector graphics, and the standard format for vector graphics on the Internet is called SVG (**S**calable **V**ector **G**raphics). This is essentially an XML-based format in which tags such as <line>, <circle>, <rect> or <path> can be found for the basic geometric shapes. The tags are described in more detail by attributes; these specify, for example, start and end points, radius, width, height, line thickness, etc. The most complicated attribute is probably the “d” attribute in <path>: It contains simple graphic commands such as MoveTo or LineTo, but also more complex ones such as elliptical arcs, Bézier curves or Catmull-Rom curves. This can be used to draw a large variety of shapes.

Example of an SVG file:

```
<svg width="300" height="200" viewbox="0 0 300 200">
  <desc>Flagge Dänemarks</desc>
  <rect x="0" y="0" width="300" height="200" fill="#E31836" />
  <path d="m80,0 h40 v80 h180 v40h-180 v80 h-40 v-80 h-80 v-40 h80z" fill="white" />
</svg>
```

SVG graphics for download can be found here:

Frei:
https://freesvg.org
https://publicdomainvectors.org
https://freesvgclipart.com/
https://www.vecteezy.com/
https://www.freevector.com/
https://www.123freevectors.com
https://www.vector4free.com/

Commercial:
https://www.shutterstock.com

You can also create SVG vector graphics yourself: the open source program Inkscape is recommended. Inkscape can also convert bitmap images into vector graphics (although this does not always work perfectly and requires some trial and error).

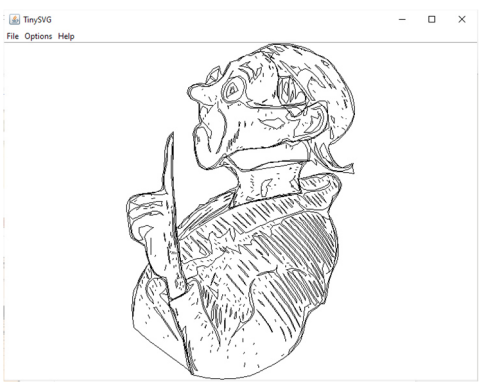
SVG is a complex format that leaves nothing to be desired in terms of display options. The complexity of the format raises the question of how SVG graphics can be displayed on an 8-bit computer from the 1980s with limited graphics options and limited main memory. The direct reading and interpretation of SVG files is hardly possible due to the memory requirements alone.

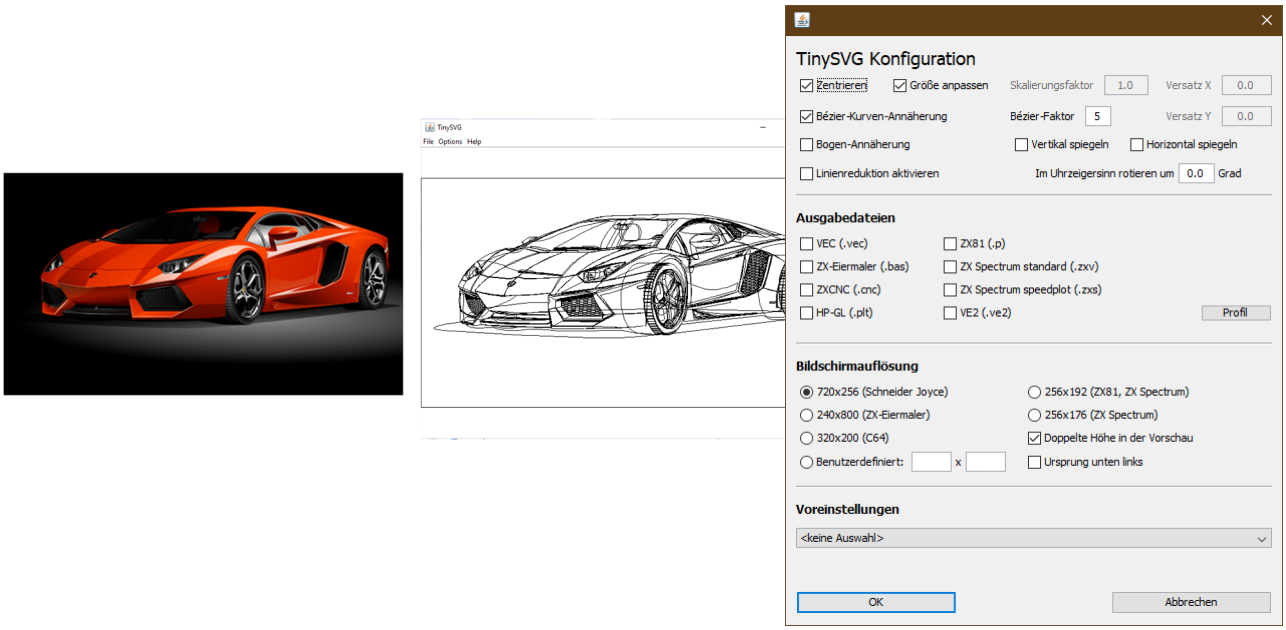
My approach is therefore two-stage: first, the SVG graphic is converted on the PC into a simple vector format containing only graphic commands that can be interpreted on the Joyce. All SVG specialties that cannot be displayed on a monochrome screen are omitted. A Joyce program then reads the simplified vector format and draws the graphic.

For the first step, I developed a Java program called **TinySVG**, which now also supports other platforms in addition to the Schneider Joyce. Currently, i.e. in version 0.15, not all SVG features are implemented yet, but many SVG files can already be converted well. You just have to try it out.

The second step is performed by the CP/M program **vecread**, which is currently available in version 1.2. TinySVG and vecread can be downloaded in the topic “Vektorgrafik für die Joyce” on joyceforum.de:
https://joyceforum.de/viewtopic.php?f=5&t=65

The following examples show SVG graphics as they are displayed in the Internet browser, in TinySVG and via vecread on the Joyce:

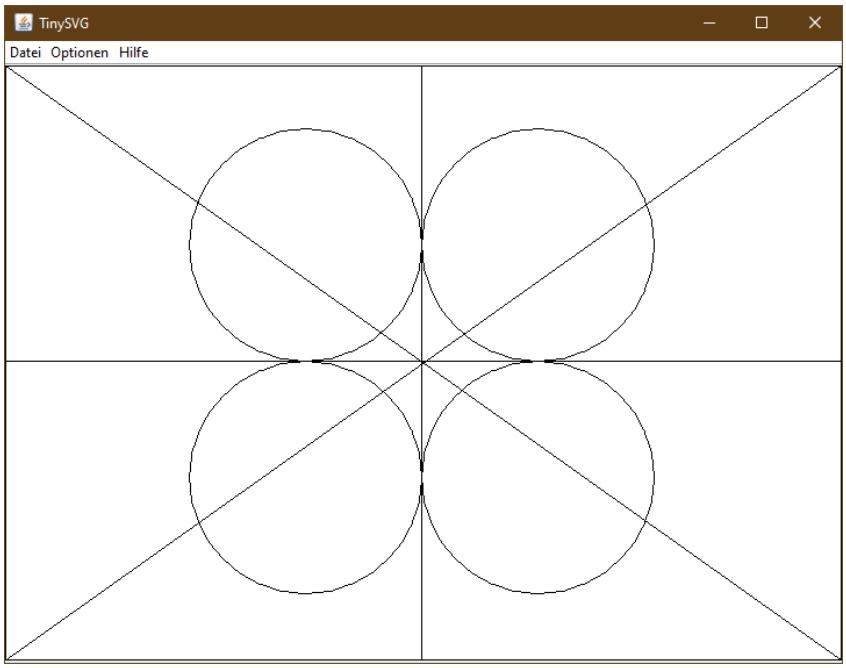




Instructions TinySVG (version: v0.27)

To run TinySVG, a Java runtime version 8 or higher must be installed on the PC. The program consists of a JAR file that can be started simply by double-clicking.

First, you should set the desired language if the menu does not already appear in the correct language: **Optionen > Deutsch** or **Optionen > Englisch** / **Options > German** or **Options > English**. The German menus are used in the following.
Then set the configuration via **Optionen > Konfiguration**.
The **Zentrieren** option should be activated; this will display the graphic in the middle of the preview window.



Größe anpassen should also be activated so that the graphic is resized to the full available height and width. The adjustment is made while retaining the aspect ratio, i.e. the graphic is not stretched or compressed.

Bézier-Kurven-Annäherung converts Bézier curves contained in <path> tags into line segments. The option should be activated, otherwise Bézier curves are replaced by straight lines, which can lead to interesting effects, but is generally undesirable. The **Bézier-Faktor** changes the number of line segments created. The smaller the value, the more lines are generated and the smoother the curve becomes. However, the file will then be larger and drawing the graphic will take longer. The default value of 10 usually works quite well.

The **Bogen-Annäherung** works in a similar way, but for circular or elliptical arcs. This option should also be activated.

TinySVG can reduce the number of lines by removing unnecessary lines. These are, on the one hand, connected lines with a length of only one pixel. If two lines are connected and have the same gradient angle, they are combined into one line. The **Linienreduktion aktivieren** option should remain switched on.

If required, the graphic can be mirrored; to do this, you can activate the options **Vertikal spiegeln** or **Horizontal spiegeln**. It is also possible to rotate the graphic. The rotation angle is specified in degrees, if necessary with a decimal point (not decimal comma). The **Im Uhrzeigersinn rotieren um** setting is responsible for this.

Under **Ausgabedateien**, you can select which vector files TinySVG should generate. **VEC (.vec)** or even better **VE2 (.ve2)** should be activated for the Schneider Joyce. These are formats that the current version of the vecread program can read. However, you can also create HP-GL files with **HP-GL (.plt)**, for example, which can then be displayed with the **HP-GL interpreter**.

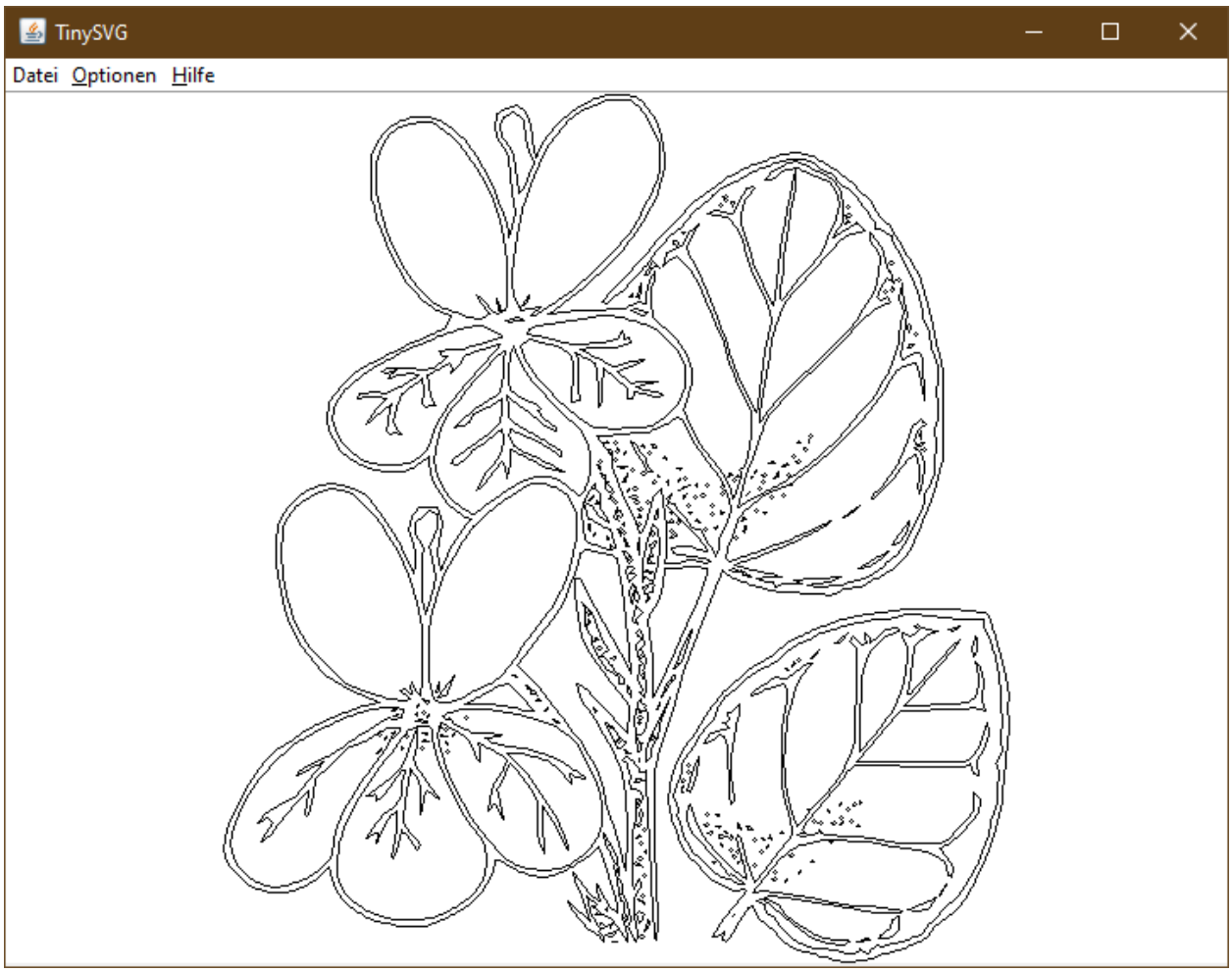
Via **Bildschirmauflösung** the resolution of the output vectors is determined. The appropriate option for the Joyce is clear: **720x256 (Schneider Joyce)**. In addition, a tick should be placed next to **Doppelte Höhe in der Vorschau**; the preview in TinySVG then displays the image in a resolution of 720x512. As the pixel aspect ratio (PAR) on the Joyce is almost 1:2, you will see an undistorted image.

All settings under **Ausgabedateien** and **Bildschirmauflösung** can also be easily selected via the **Voreinstellungen**: The **Schneider Joyce** selection sets the recommended options.

All options are automatically saved in a file called TinySVG.properties, which is stored in the same directory as TinySVG.jar. This file is read the next time the program is started and all settings are as they were last set.

The file TinySVG.log is also created, which - as the file extension suggests - contains the log. You can control how detailed the log can be via **Optionen > Log-Schweregrad setzen**. The default setting is **Info**. More detailed logs with **Fine** or **Finer** are only useful if an error needs to be analyzed in more detail. However, the log files then become significantly larger and writing the log takes longer. You can also set the log severity to **Warning** or **Severe** if you prefer shorter logs.

Once all options have been set, you can open an SVG file via **Datei > Öffnen**. TinySVG remembers the path that you last opened. Once you have selected the desired file, the file is converted and appears in the preview window. The output files are also created immediately and saved in the directory where the SVG file is located.



If you want to experiment with the options after loading a file (e.g. with the Bézier factor), you can easily display the changes via **Datei > Neu laden**.
The program is closed with **Datei > Beenden**.

Instructions vecread

You can start the program on the Joyce simply by entering vecread, whereupon a menu appears which, among other things, prompts you to enter a file name. The file name must be entered in full, i.e. including the extension .ve2 or .vec. Alternatively, you can also call up the program with the desired file as a parameter, e.g. `vecread puffin.ve2`. Drawing of the graphic then begins immediately. A beep signals the end of the drawing process; pressing a key ends the program.

Bernd Bock, September 2022

Update TinySVG v0.26

- Determination of min./max. coordinates to achieve a better estimation of the required line segments for curves and arcs. They look much better now.
- VE2 configuration settings are now saved correctly.
- The “Schneider Joyce” preset now uses the newer (and more efficient) VE2 format by default instead of VEC.
- In the configuration there is now a “Profil” button under “Ausgabedateien”, in which you can specify which graphic objects should be generated. It replaces the VE2 details, which was only limited to VE2 files. By default, all graphic objects are selected, and most people will probably leave it that way; it is probably more of an “expert setting”.
- A few more bug fixes that I can't think of right now...

Update TinySVG v0.27

In SVG paths, horizontal and vertical lines are coded with H/V or h/v. This is normally followed by a single parameter that defines the end point: the X coordinate for horizontal lines or the Y coordinate for vertical lines. However, the specification also allows several coordinates to be specified, which actually makes no sense, as several adjacent horizontal lines can also be represented by a single horizontal line - you simply specify the last end point. The same applies to vertical lines

Nevertheless, there are SVG files that contain such nonsense, and the previous TinySVG versions did not display them correctly. Version 0.27 fixes this.

Update vecread 3.1

- The pause between images in slideshow mode is 4 seconds by default.
- Up to 100 files can now be displayed in the slideshow (if you get that many on the disk); previously it was 40.
- The pause can be changed via the command line. For example, if you want to have a pause of 10 seconds, call the program as follows: **vecread3 d 10**

Update vecread 4.1

The new version 4.1 of the VEC / VE2 display program has the same functions as its predecessor v3.1, but the interface is more convenient to use because it uses the FORM2 library. The slideshow is started with a “Start diashow” button; a single file is displayed with the “Ok” button and there is a “Cancel” button for canceling.

Bernd Bock

JOYCE-USER-AG e.V. - www.joyce.de - October 2023