
CodeGear
Join us for <Dev Days of Summer> 2024!
August 12-16 | 19-23, 2024
Two full weeks of live and on-demand
developer content, all for free!
RAD
C++
UE
Visual Assist
Yellowfin
Sponsored by
embarcadero

Use SVG images in Delphi

Modernize your UIs

Patrick Prémartin

MVP Embarcadero, writer, editor,
freelance developer and trainer,
Twitch streamer.

Bio, links and contacts on
https://vasur.fr/gravatar

Agenda

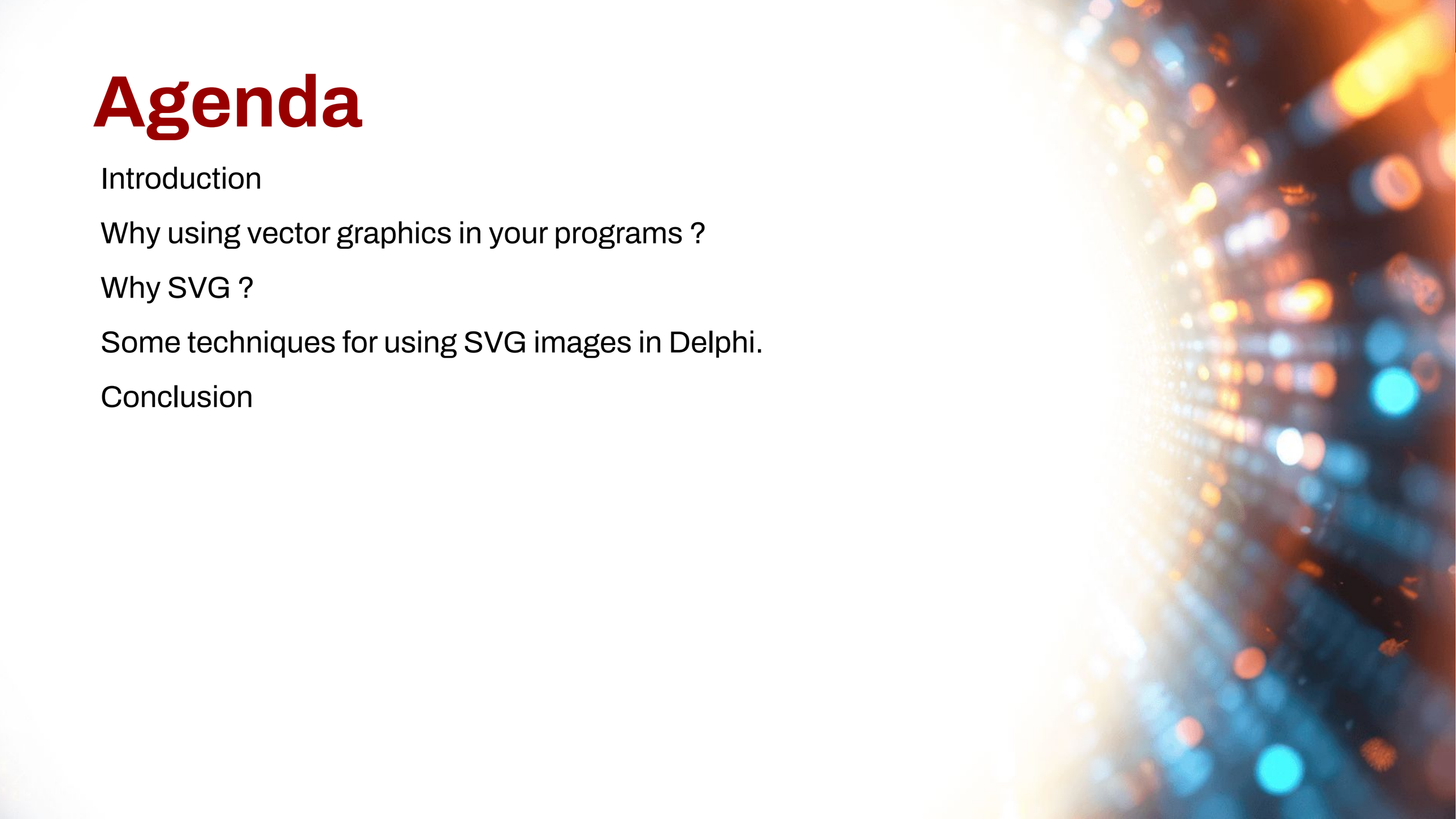
Introduction

Why using vector graphics in your programs ?

Why SVG ?

Some techniques for using SVG images in Delphi.

Conclusion



Introduction

With the introduction of Skia&Skia4Delphi preinstalled since versions 12 Athens and the Community Edition update this summer, I thought it important to remind you that you can replace bitmaps with vectors in all your Delphi projects when you upgrade them.

Of course, this may take some time, depending on the project and the number of images used, but it's up to you to decide whether you'd prefer to do it gradually now or as a matter of urgency in 5 to 10 years' time.

Why using vector graphics in your programs ?

Bitmap images have a fixed size. Scaling is complicated to manage without degrading the image.

This change of scale is a reality. Simply activate the screen zoom or let the software automatically adapt to the resolution of your users' screens.

Why using vector graphics in your programs ?

Images, icons and old user interfaces can become blurred, drool or have ugly outlines in certain circumstances.

Looks are an important factor in software appreciation. It should be taken into account.

This is one of the reasons for the success of re-styling many software programs, in addition to adapting them to the OS theme to improve the life and view of users!

Why using vector graphics in your programs ?

Unlike bitmap images, vector images are redrawn in proportion to their display size (ideally in physical pixels rather than logical pixels).

They are always impeccable (provided they are not too detailed and displayed at a too small size).

Interface icons can be replaced in Delphi by vector images in both VCL and FireMonkey.

Why SVG ?

SVG files are based on the "Scalable Vector Graphics" specifications published by the W3C in 2001.

This format is open, in plain text, described in XML, freely generatable and usable, unlike other proprietary vector formats (e.g. Adobe Illustrator).

Further reading: https://en.wikipedia.org/wiki/SVG

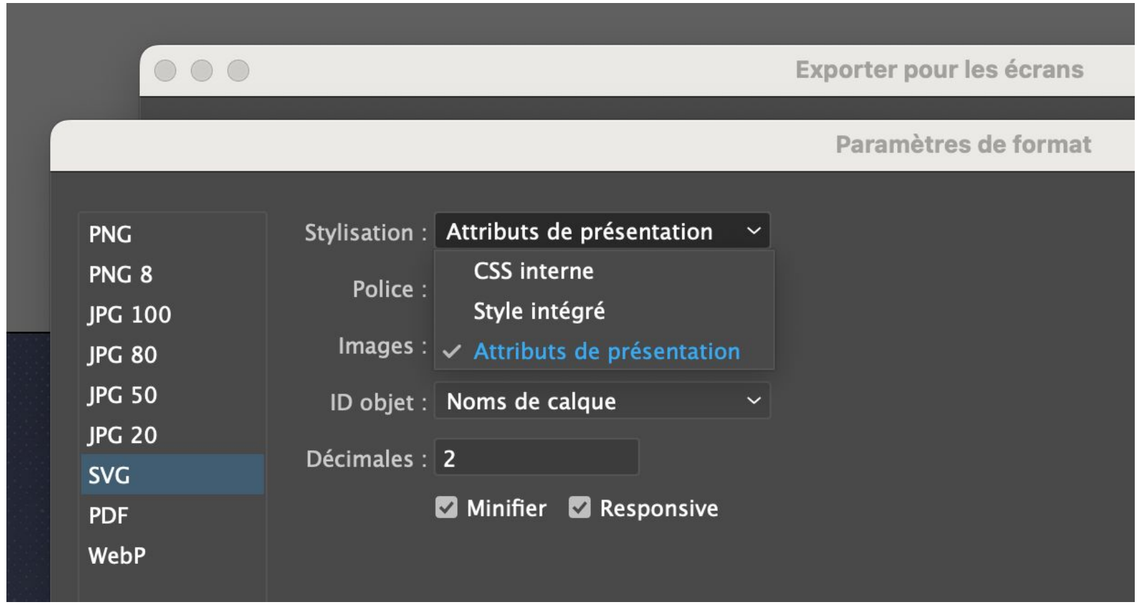
Why SVG ?

Like many things related to the web, something that starts out simple can quickly become complicated. There are several versions of SVG.

Skia doesn't support CSS-integrated versions, but there's no need to worry about mixing shapes, curves and vector paths.

Why SVG ?

If you're exporting files from Adobe Illustrator, set the export to "format attributes" instead of "internal CSS" or "embedded style".



This problem should not arise with Inkscape or Affinity.

Why SVG ?

The SVG format is available in most vector graphics software.

Many sites offer vector image banks available in SVG.

There's no reason to use any other format for your software, especially now that it's (almost) natively supported by Delphi!

Why SVG ?

If you're using an older version of Delphi and can't yet upgrade to a newer version, you have other options.

I made a review of what exists in January 2020. It's available in French (easily translated by your browser) at https://developpeur-pascal.fr/webinaire-du-28-janvier-2020-utiliser-des-dessins-vectoriels-a-la-place-dimages-de-differentes-resolutions.html

Some techniques for using SVG images in Delphi.

With that out of the way, let's get down to the nitty-gritty of code and components...

- TSVGIconImageList from Ethea: a descendant of TCustomImageList, an SVG file container that can use Skia, Image32 and other implementations. Works like a TImageList.

- FireMonkey's TPath for displaying vector paths

- TSkSVG from Skia4Delphi to display an SVG on screen

Some techniques for using SVG images in Delphi.

Let's move on to some code to support things not available in current implementations.

- SVGFolder2DelphiUnit: an open source utility that groups SVG files within a folder into a Delphi unit as an array of multiline strings with constants, enumerations and a class to access them.

Some techniques for using SVG images in Delphi.

- SVGToBitmap: an example of using the Olf.Skia.SVGToBitmap.pas unit from the DeveloppeurPascal/librairies GitHub repository to display pipe pieces (and other things).

- SVGToBitmapWithMargins: adding margins to the SVG bitmap export to frame drawings so that pieces can be pasted in the right place. This code is taken from the video game Ploumtris.

Some techniques for using SVG images in Delphi.

- SVGToBitmapList: presents the use of a list of SVG sources with an (optional) cache of generated bitmaps. A sort of TImageList or TImageCollection, very simplified and usable only by code (for the moment).

As a bonus, here's a project that creates an animation from a series of SVG images. Handy for video games or animating your screens with whatever you want. It's managed as a TFrame and will probably become a component one day.

Conclusion

As we've just seen, Delphi and the open source community offer classic solutions for displaying vector images in form design, but we can also do whatever we want with code.

Skia also offers to open other image formats as standard, transparently, in the usual display components.

Conclusion

This presentation, examples, replays and links are available on https://github.com/DeveloppeurPascal/DevDaysOfSummer2024-UseSVGImagesInDelphi

The unit Olf.Skia.SVGToBitmap is available from https://github.com/DeveloppeurPascal/librairies

SVG Folder 2 Delphi Unit is available from https://svgfolder2delphiunit.olfsoftware.fr/

Conclusion

TSVGIconImageList component is available from
https://github.com/EtheaDev/SVGIconImageList

And of course the brilliant Skia library and its integration into Delphi through the Skia4Delphi project.

https://skia.org

https://skia4delphi.org

Conclusion

The images used in this presentation come from :

- Kenney : https://kenney.nl
- Pictogrammers : https://pictogrammers.com
- Shaun Roselt :
  - https://github.com/shaunroselt/Delphi-Bootstrap-Icons
  - https://github.com/shaunroselt/Delphi-Feather-Icons
  - https://github.com/shaunroselt/Delphi-Font-Awesome-Icons

Conclusion

And finally, a few examples of how SVG is used in video games, the source codes of which can be seen on GitHub.

- Sporgloo : https://sporgloo.gamolf.fr
- Ploumtris : https://ploumtris.gamolf.fr
- And some games based on Gamolf FMX Game Starter Kit : https://github.com/DeveloppeurPascal?tab=repositories&q=gfgsk-game

Conclusion

I wish you all the best for your user interface makeovers!

Thank you for your attention.

See you soon(*)

(*) this is neither a threat nor a promise, just a strong probability.


CodeGear
<Dev Days of Summer> 2024!
Thanks for Watching
RAD
C++
UE
Visual Assist
Yellowfin