



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!



Visual Assist



Sponsored by



Utiliser des images SVG en Delphi

Modernisez vos interfaces



Patrick Prémartin

MVP Embarcadero, auteur,
éditeur, développeur et formateur
freelance, streameur sur Twitch.

Bio, liens et contacts sur
<https://vasur.fr/gravatar>

Agenda

Introduction

Pourquoi utiliser des images vectorielles dans vos programmes ?

Pourquoi SVG ?

Des techniques d'utilisation d'images SVG dans Delphi

Conclusion

Introduction

Avec l'arrivée de Skia&Skia4Delphi préinstallés depuis les versions 12 Athens et la mise à jour de la Community Edition cet été il me semblait important de rappeler que nous pouvons remplacer les bitmaps par du vectoriel dans tous nos projets Delphi à l'occasion de leur mise à niveau.

Bien sûr ça peut prendre du temps selon les projets et le nombre d'images utilisées mais à vous de voir si vous préférez le faire progressivement dès maintenant ou en urgence dans 5 à 10 ans.

Pourquoi utiliser des images vectorielles dans vos programmes ?

Les images bitmap ont une taille fixe. Le changement d'échelle est compliqué à gérer sans dégradation de l'image.

Ce changement d'échelle est une réalité. Il suffit d'activer le zoom d'écran ou laisser les logiciels s'adapter automatiquement à la résolution des écrans de vos utilisateurs.

Pourquoi utiliser des images vectorielles dans vos programmes ?

Les images, icônes, interfaces utilisateurs anciennes peuvent devenir troubles, baver ou avoir des contours moches dans certaines circonstances.

Le look est un point important dans l'appréciation des logiciels. Il faut en tenir compte.

C'est l'une des raisons du succès du relookage par style de beaucoup de logiciels, en plus de les adapter au thème de l'OS pour améliorer la vie et la vue des utilisateurs !

Pourquoi utiliser des images vectorielles dans vos programmes ?

Contrairement aux images bitmap les images vectorielles se redessinent proportionnellement à leur taille d'affichage (idéalement en pixels physiques plutôt qu'en pixels logiques).

Elles sont toujours impeccables (à condition de ne pas être trop détaillées et affichées dans une taille trop petite).

Il est possible de remplacer les icônes d'interfaces dans Delphi par des images vectorielles en VCL comme en FireMonkey.

Pourquoi SVG ?

Les fichiers SVG sont basés sur les spécifications “Scalable Vector Graphics” éditées par le W3C en 2001.

Ce format est ouvert, en clair, décrit dans du XML, librement générable et utilisable contrairement à d'autres formats vectoriels propriétaires (par exemple celui d'Adobe Illustrator).

Un peu de lecture :

https://fr.wikipedia.org/wiki/Scalable_Vector_Graphics

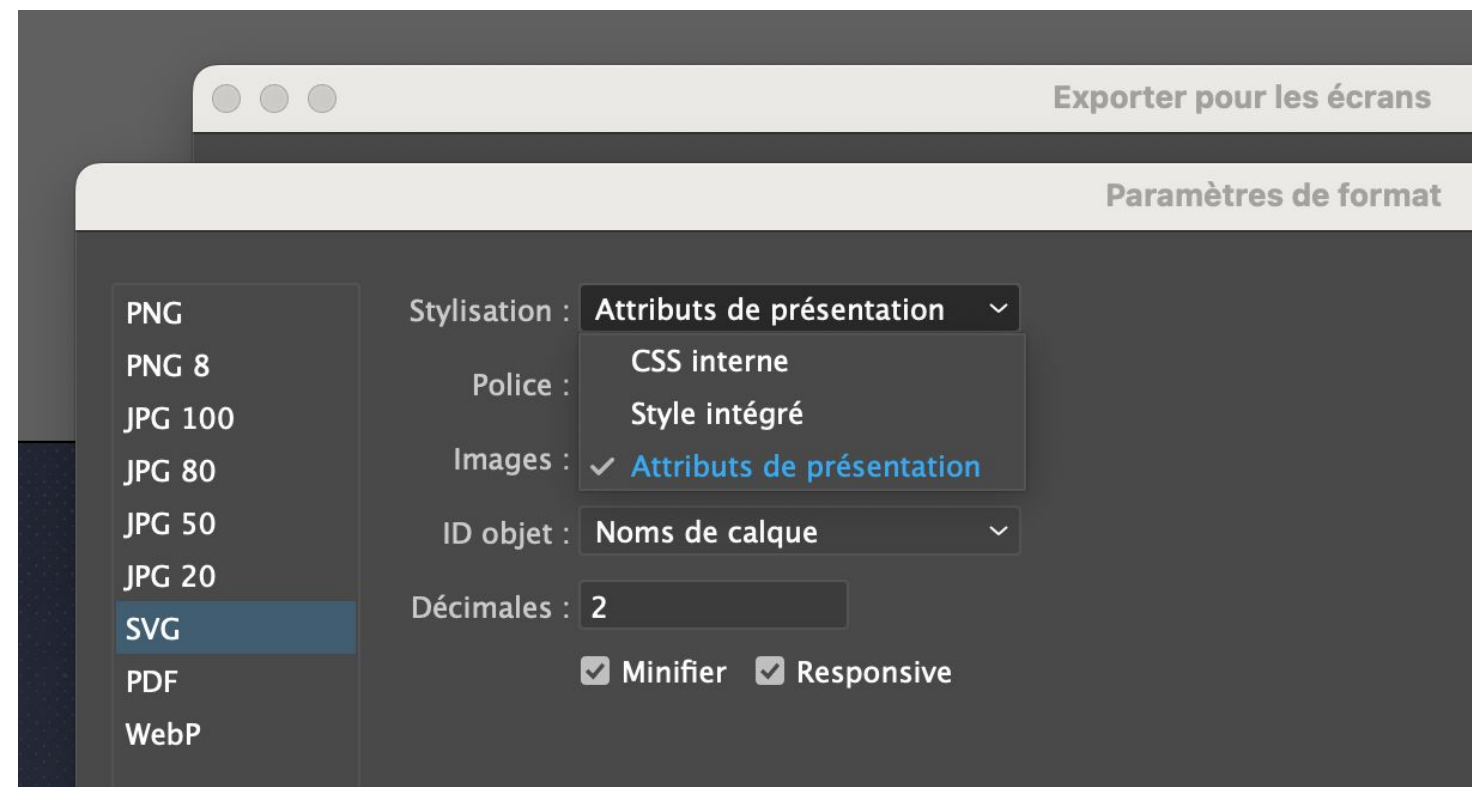
Pourquoi SVG ?

Comme beaucoup de choses en lien avec le web un truc simple au départ peut vite devenir compliqué. Il existe plusieurs versions de SVG.

Skia ne prend pas en charge les versions intégrant des CSS mais pas de soucis en mélangeant des formes, des courbes et des chemins vectoriels.

Pourquoi SVG ?

Si vous exportez des fichiers depuis Adobe Illustrator, paramétrez l'export en “attributs de format” au lieu de laisser “CSS interne” ou “style intégré”.



Ce problème ne devrait pas se poser depuis Inkscape ou Affinity.

Pourquoi SVG ?

Le format SVG est disponible dans la plupart des logiciels de dessin vectoriel.

De nombreux sites proposent des banques d'images vectorielles disponibles en SVG.

Aucune raison d'utiliser un autre format pour vos logiciels, surtout maintenant qu'il est pris en charge (presque) nativement par Delphi !

Pourquoi SVG ?

Si vous utilisez une version ancienne de Delphi et ne pouvez pas encore passer à une version plus récente vous avez d'autres options que celles qui suivent.

J'ai fait un tour de l'existant en janvier 2020. C'est disponible en français (facilement traduit par votre navigateur) sur <https://developpeur-pascal.fr/webinaire-du-28-janvier-2020-utiliser-des-dessins-vectoriels-a-la-place-dimages-de-differentes-resolutions.html>

Des techniques d'utilisation d'images SVG dans Delphi

Ces rappels étant faits, nous pouvons passer à un peu de pratique, du code et des composants...

- TSVGIconImageList de Ethea : un descendant de TCustomImageList conteneur de fichiers SVG pouvant utiliser Skia, Image32 et d'autres implémentations. Fonctionne comme un TImageList.
- TPath de FireMonkey pour afficher des chemins vectoriels
- TSkSVG de Skia4Delphi pour afficher un SVG à l'écran

Des techniques d'utilisation d'images SVG dans Delphi

Passons à un peu de code pour prendre en charge des choses non disponibles dans les implémentations actuelles.

- SVGFolder2DelphiUnit : un utilitaire open source regroupant les fichiers SVG d'un dossier dans une unité Delphi sous forme de tableau de chaînes multilignes avec constantes, énumérations et classe permettant d'y accéder.

Des techniques d'utilisation d'images SVG dans Delphi

- SVGToBitmap : un exemple d'utilisation de l'unité
Olf.Skia.SVGToBitmap.pas provenant du dépôt GitHub
DeveloppeurPascal/librairies pour afficher des morceaux de tuyaux.
- SVGToBitmapWithMargins : l'ajout de marges dans l'export bitmap de
SVG permettant de recadrer les dessins afin de coller les morceaux au
bon endroit. Ce code est extrait du jeu vidéo Ploumtris.

Des techniques d'utilisation d'images SVG dans Delphi

- SVGToBitmapList : présente l'utilisation d'une liste de sources SVG ayant un cache (facultatif) des bitmaps générés. Une sorte de TImageList ou de TImageCollection très simplifiée et utilisable uniquement par code (pour le moment).

En bonus je vous propose un projet créant une animation à partir d'une série d'images SVG. Pratique pour les jeux vidéo ou animer vos écrans avec ce que vous voulez. C'est géré sous forme de TFrame et deviendra probablement un composant un jour.

Conclusion

Comme nous venons de le voir Delphi et la communauté open source proposent des solutions classiques pour afficher des images vectorielles en conception de fiches mais pouvons aussi faire ce que nous voulons par code.

Skia propose aussi d'ouvrir en standard d'autres formats d'images, de façon transparente, dans les composants d'affichages habituels.

Conclusion

Cette présentation, les exemples, rediffusions et liens sont disponibles sur

<https://github.com/DeveloppeurPascal/DevDaysOfSummer2024-UseSVGImagesInDelphi>

La librairie Olf.Skia.SVGToBitmap est sur

<https://github.com/DeveloppeurPascal/librairies>

SVG Folder 2 Delphi Unit est téléchargeable depuis

<https://svgfolder2delphiunit.olfsoftware.fr/>

Conclusion

Le composant TSVGIconImageList est disponible sur
<https://github.com/EtheaDev/SVGIconImageList>

Et bien sûr la géniale librairie Skia et son intégration dans Delphi par le projet Skia4Delphi.

<https://skia.org>

<https://skia4delphi.org>

Conclusion

Les images utilisées dans cette présentation proviennent de :

- 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

Et pour finir quelques exemples d'utilisation de SVG dans des jeux vidéos dont les codes sources sont visibles sur GitHub.

- Sporgloo : <https://sporgloo.gamolf.fr>
- Ploumtris : <https://ploumtris.gamolf.fr>
- Et des jeux basés sur Gamolf FMX Game Starter Kit :
<https://github.com/DeveloppeurPascal?tab=repositories&q=gfgsk-game>

Conclusion

Je vous souhaite de bons relookages de vos interfaces utilisateurs !

Merci pour votre attention.

A bientôt(*)

(*) ceci n'est ni une menace, ni une promesse, juste une forte probabilité



<Dev Days of Summer> 2024!

Thanks for Watching

RAD



Visual Assist

Yellowfin