



# Utiliser des dessins vectoriels à la place d'images

Webinaire du 28 janvier 2020



# Au programme

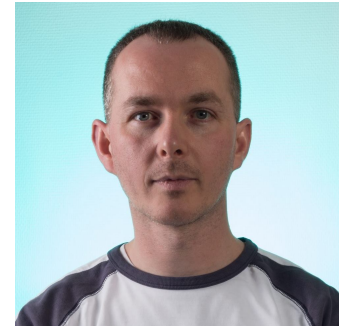
- Introduction
- Les résolutions d'écrans
- Les images dans Delphi
- Les dessins vectoriels
- Utilisation dans nos programmes
- Conclusion



# Présentation



# A propos de Patrick Prémartin



- Ancien utilisateur de Turbo Pascal, amoureux de Delphi depuis toujours, développeur web depuis NCSA Mosaic...
- Prestataire et formateur à travers sa société Olf Software.
- Editeur du blog <https://developpeur-pascal.fr/>
- MVP Embarcadero depuis octobre 2017.
- Auteur d'articles, de livres, webinariste et parfois même conférencier.



## A propos de Olf Software



- Depuis octobre 2001 la société de services Olf Software propose des prestations de conseil, développement et formation sur les technologies liées au web et au développement logiciel.
- La société édite également des livres, des sites Internet et des logiciels.
- Site web principal : <https://olfsoftware.fr>



# Formations Delphi



- Olf Software propose des formations en lien avec Delphi.
- Des formations inter entreprises sont organisées régulièrement et annoncées sur <https://se-former-a-delphi.fr/>
- Des formations intra entreprise personnalisées sont faisables sur demande partout en France et en Europe.
- Olf Software est un organisme de formation professionnelle enregistré en préfecture et référencé sur Datadock.  
Nos formations peuvent être prises en charge par les OPCA/OPCO.



# A propos de Barnsten



- La société Barnsten est basée aux Pays-Bas. Elle s'occupe de la distribution des produits Embarcadero pour la France et le Benelux.
- Barnsten assure également un premier niveau de support utilisateur et répond à toutes questions liées aux produits et licences. Quand c'est nécessaire vous serez redirigés vers des partenaires ou prestataires pouvant répondre à vos problématiques.
- Barnsten organise régulièrement des événements, des formations sur les produits Embarcadero et cette série de webinaires mensuels.



# Les versions de Delphi



- Plusieurs licences de Delphi existent (vendu seul ou dans RAD Studio):
  - Academic : pour les établissements d'enseignement et leurs étudiants
  - Community Edition : gratuit pour tous, en licence commerciale limitée
  - Professional, Enterprise, Architect : pour les entreprises, selon leurs besoins
- Pour toutes informations sur les licences et les tarifs contactez la société Barnsten sur <https://www.barnsten.com/fr/>



# Les webinaires précédents

- Je participe régulièrement à des webinaires. Une page leur est consacrée sur mon blog à l'adresse <https://vasur.fr/webinairesdelphi>
- Vous y trouverez les rediffusions en vidéos lorsqu'elles sont disponibles, un PDF de mes présentations, le lien vers les sources des exemples présentés et des informations complémentaires.
- S'y trouvent aussi les dates et liens d'inscriptions pour les prochains webinaires, conférences et formations.



## En bref

- Présentation de la société Barnsten  
<https://www.barnsten.com>
- Présentation de la société Olf Software  
<https://www.olfsoftware.fr>
- Présentation de Patrick Prémartin  
<https://www.linkedin.com/in/patrickpremartin/>
- Les différentes versions de Delphi  
<https://www.embarcadero.com/fr/products/delphi>
- Voir ou revoir les webinaires précédents  
<https://vasur.fr/webinairesdelphi>



# Introduction



# Introduction

- Historiquement les logiciels ne gèrent qu'une seule résolution pour les images qu'ils affichent.
- Avec l'arrivée des smartphones et tablettes les choses ont changé et nous avons dû nous adapter.
- C'est désormais aussi le cas avec les écrans classiques qui montent jusqu'à 8K et ont en réalité une double résolution.



# Introduction

- Notre problème c'est que les logiciels existants bavent sur des écrans récents et qu'il faut donc les mettre à niveau pour conserver un rendu visuel optimum.



# Les résolutions d'écran





# Les résolutions d'écran

- A nous de nous débrouiller pour limiter les dégâts visuels et faire en sorte que nos logiciels desktop ou mobiles s'affichent correctement.
- Les systèmes d'exploitation ont su évoluer pour gérer ces informations.
- Nos logiciels de développement également.



# Les résolutions d'écran

- Dans Delphi et C++Builder nous avons déjà la résolution du système d'exploitation pour les programmes VCL sous Windows et il se débrouillait pour les smartphones et tablettes.
- Depuis la version 10.3 Rio, Windows permettant de gérer une résolution par écran, Embarcadero a intégré ces API.
- Lorsque plusieurs écrans sont connectés sur le même ordinateur nous pouvons savoir comment afficher nos fiches sur chaque écran avec la résolution de chaque écran plutôt qu'une résolution globale.



# Les images dans Delphi



# Les images dans Delphi

- Jusque là nous avons des composants images permettant d'afficher un visuel dans une taille calculée lors de la création des fiches.
- Désormais nous avons des composants qui permettent de s'adapter lors de l'utilisation du programme mais ça a un coût.



# Les images dans Delphi

- Chaque image doit être prévue dans plusieurs tailles différentes pour permettre aux programmes de choisir la plus proche de celle qu'il doit afficher et prendre celle de dessus pour la réduire plutôt que celle de dessous pour l'agrandir.
- On part du principe qu'on a moins de perte visuelle pour l'utilisateur en diminuant la taille d'une image qu'en l'agrandissant.
- Bien entendu ce n'est pas vrai pour les photos, mais pour les icônes et dessins de boutons, en général, ça passe mieux comme ça (à condition de ne pas avoir un 1024x1024 pour afficher du 16x16).



# Les images dans Delphi

- Pour FireMonkey, le framework multiplateforme fourni par Embarcadero, tout est graphique et le multi résolutions été prévu dès le départ puisque la première version était destinée à tourner sous iOS.
- Le composant TImageList permet de stocker des images dans différentes tailles. Il gère aussi le découpage de feuilles de sprites ce qui permet de diminuer le nombre de fichiers à diffuser avec chaque programme.



# Les images dans Delphi

- Pour la VCL nous avons de nouveaux composants apparus avec la version 10.3 Rio : TImageCollection et TVirtualImageList.
- TImageCollection est à mettre sur un TDataModule pour stocker les images.
- TVirtualImageList remplace le TImageList historique. Il est à poser sur les fiches pour afficher les images dans la résolution de l'écran sur lequel chaque fiche apparaît.



# Les images dans Delphi

- L'inconvénient de cette solution c'est qu'elle oblige à créer une multitude d'images de tailles différentes pour un seul visuel.
- Ces fichiers sont importés dans les programmes sous forme de ressources et en gonflent inutilement la taille.
- Les utilisateurs n'ont pas besoin de tous les visuels embarqués, sauf à avoir des écrans de tailles différentes sur leurs postes de travail.



# Les images dans Delphi

- Le coût est vite calculé :
  - temps de travail à créer les images,
  - temps de travail à les importer dans les logiciels,
  - temps de travail à les maintenir en fonction de l'évolution du parc d'écrans,
  - taille des fichiers à diffuser donc coût de stockage et de bande passante puisque la plupart des logiciels se téléchargent sur Internet.
- Heureusement il existe des alternatives : les dessins vectoriels.



# Les dessins vectoriels



# Les dessins vectoriels

- Si vous avez connu les débuts de l'informatique moderne et graphique, vous devez déjà avoir entendu parlé ou vu des cliparts.
- Ces dessins étaient les premiers dessins vectoriels destinés à agrémenter les documents et présentations.



(c) Patrick Prémartin / [developpeur-pascal.fr](http://developpeur-pascal.fr)





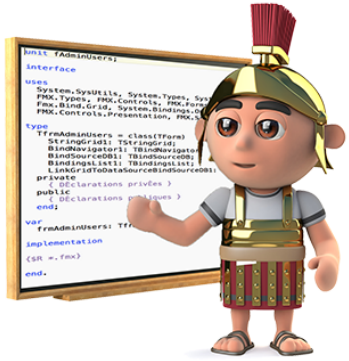
# Les dessins vectoriels

- L'avantage des dessins vectoriels, plus généralement utilisés en CAO, c'est qu'ils sont composés de vecteurs.
- Toutes les tailles et positions sont calculées relativement à l'emplacement sur lequel ils doivent s'afficher.
- Contrairement à une photo, un GIF, JPEG ou PNG, la déformation d'un dessin vectoriel n'entraîne pas de perte car il est redessiné, pas « tordu » ou « étiré ».



# Les dessins vectoriels

- Dans le domaine d'Internet où la bande passante et les temps d'affichage sont importants, les images d'illustration ont rapidement été remplacées par des dessins vectoriels car ils s'affichaient plus vite et sur toutes les tailles d'écran sans problème.
- C'est pour ça qu'est apparu le format SVG, un standard du W3C, pour intégrer des dessins vectoriels dans les pages web.
- SVG est une norme et ne nécessite pas de payer de licence pour l'utiliser, il s'est vite répandu en dehors du web dans les logiciels de dessin, les documents bureautiques et les PDF.



# Les dessins vectoriels

- Comme les dessins vectoriels sont dessinés en fonction de la surface d'affichage ils n'ont pas de problème de résolution d'écran et passent correctement sans effet de flou ni baver.
- Il est donc temps de s'en servir également dans nos programmes quand c'est possible pour limiter la taille des ressources à déployer.
- En revanche il faut quand même avoir conscience qu'ils sont dessinés à l'affichage, par conséquent ils sont plus lents à afficher qu'une simple image bitmap surtout s'ils sont complexes.



```
Unit AdminUsers;
interface
uses
  System.SysUtils, System.Types, Sys
  PMX.Types, PMX.Controls, PMX.Forms,
  PMX.Grid, System.Bindings,
  PMX.Controls.Presentation, PMX;
type
  TAdminUsers = class(TForm)
  StringGrid1: TStringGrid;
  BindNavigator1: TBindNavigator;
  BindSourceDB1: TBindSourceDB;
  BindMasterList1: TBindMasterList;
  LinkGridToDataSource1: TLinkGridToDataSource;
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;
var
  AdminUsers: TAdminUsers;
implementation
{$R *.res}
end.
```

# Utilisation dans nos programmes



# Utilisation dans nos programmes

- Avec la VCL aucun composant ne permet en standard d'afficher du SVG ou des dessins vectoriels. Nous devons le coder nous mêmes en dessinant sur le TCanvas d'un composant ou passer par une librairie externe.



# Utilisation dans nos programmes

- Avec FireMonkey nous avons le composant TPath qui permet de dessiner en vectoriel des formes simples comme complexes.
- Cette jolie voiture en cours de lavage est dessinée avec ce code :



```
M5,13L6.5,8.5H17.5L19,13M17.5,18A1.5,1.5 0 0,1 16,16.5A1.5,1.5 0 0,1 17.5,15A1.5,1.5 0,1 19,16.5A1.5,1.5 0 0,1 17.5,18M6.5,18A1.5,1.5 0 0,1 5,16.5A1.5,1.5 0 0,1 6.5,15A1.5,1.5 0 0,1 8,16.5A1.5,1.5 0 0,1 6.5,18M18.92,8C18.72,7.42 18.16,7 17.5,7H6.5C5.84,7 5.28,7.42 5.08,8L3,14V22A1,1 0 0,0 4,23H5A1,1 0 0,0 6,22V21H18V22A1,1 0 0,0 19,23H20A1,1 0 0,0 21,22V14M7,5A1.5,1.5 0 0,0 8.5,3.5C8.5,2.5 7,0.8 7,0.8C7,0.8 5.5,2.5 5.5,3.5A1.5,1.5 0 0,0 7,5M12,5A1.5,1.5 0 0,0 13.5,3.5C13.5,2.5 12,0.8 12,0.8C12,0.8 10.5,2.5 10.5,3.5A1.5,1.5 0 0,0 12,5M17,5A1.5,1.5 0 0,0 18.5,3.5C18.5,2.5 17,0.8 17,0.8C17,0.8 15.5,2.5 15.5,3.5A1.5,1.5 0 0,0 17,5Z
```

(oui, je sais, c'est un peu barbare mais c'est toujours mieux que du binaire)



# Utilisation dans nos programmes

- TPath gère un chemin. Il faut le voir comme [la tortue du Logo](#) qui se promène dans sa zone d'affichage. Chaque lettre est un ordre qui a des paramètres qui le suivent.
- Il ne permet pas de gérer le format SVG. On doit donc également faire appel à une librairie extérieure ou tout faire nous-mêmes comme avec la VCL.



# Utilisation dans nos programmes

- Afficher une image c'est bien, en VCL comme en FMX, mais ce serait quand même mieux de pouvoir remplacer toutes les images des différents composants avec des images vectorielles, à commencer par ces icônes de boutons baveuses.
- Pour ça, il y a de nombreuses solutions plus ou moins à jour, plus ou moins fonctionnelles et surtout plus ou moins coûteuses.
- Vous en retrouverez quelques unes sur la page <https://vasur.fr/svgdansdelphi>



```
Unit AdminUsers;
interface
uses
  System.SysUtils, System.Types, System.UITypes, System.Classes, System.Generics, FMX.Types, FMX.Controls, FMX.Forms, FMX.Graphics, FMX.Dialogs, FMX.Controls.Presentation, FMX.StdCtrls;
type
  TAdminUsers = class(TForm)
  StringGrid1: TStringGrid;
  BindNavigator1: TBindNavigator;
  BindSourceDB1: TBindSourceDB;
  BindNavigator1: TBindNavigator;
  LinkGridToDataSource1: TDataSource;
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;
var
  AdminUsers: TAdminUsers;
implementation
{$R *.res}
end.
```

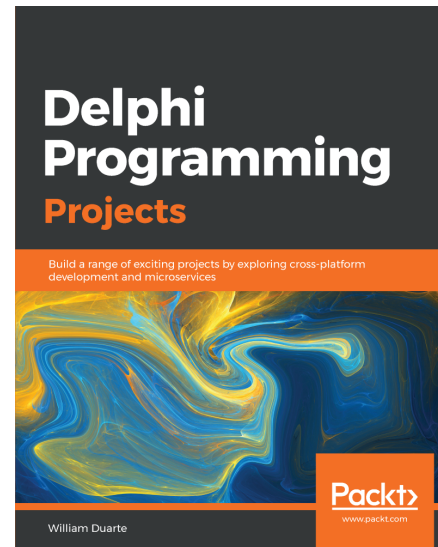
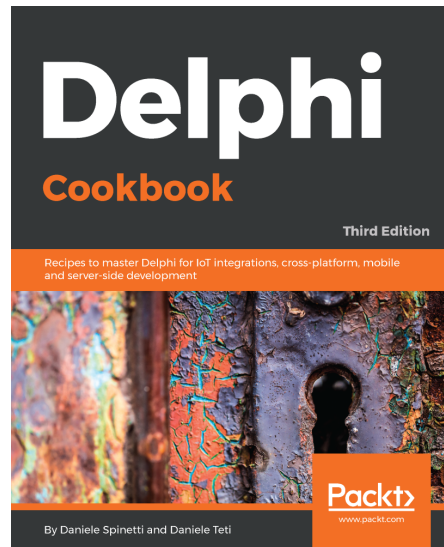
# Démo

FireMonkey et son TPath

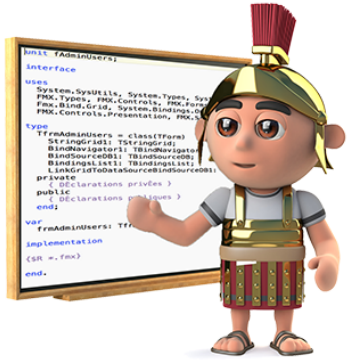


# Le composant TPath de FMX

- Découvrons ensemble comment utiliser le TPath de FireMonkey.
- D'autres exemples d'utilisations sont proposés dans ces livres :



(c) Patrick Prémartin / [developpeur-pascal.fr](http://developpeur-pascal.fr)



# Démo

La solution de RiverSoftAVG



# Les composants SVG de RiverSoftAVG

- Je vous propose de découvrir l'approche de RiverSoftAVG qui propose des composants FMX et VCL et a le gros avantage de référencer le format SVG en tant que format d'image standard.
- Tout ce qui manipule des images dans Delphi peut aussi manipuler des images au format SVG.
- Leurs composants sont payants sous forme de licence à vie avec renouvellement annuel pour bénéficier des mises à jour.
- Rendez-vous sur <http://www.riversoftavg.com/svg.htm> pour le détail.



```
Unit AdminUsers;
interface
uses
  System.SysUtils, System.Types, System.UITypes, System.Classes, System.Generics, FMX.Types, FMX.Controls, FMX.Forms, FMX.Graphics, FMX.Dialogs, FMX.Controls.Presentation, FMX.StdCtrls;
type
  TAdminUsers = class(TForm)
  StringGrid1: TStringGrid;
  BindNavigator1: TBindNavigator;
  BindSourceDB1: TBindSourceDB;
  BindMasterList1: TBindMasterList;
  LinkGridToDataSource1: TLinkGridToDataSource;
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;
var
  AdminUsers: TAdminUsers;
implementation
{$R *.res}
end.
```

# Démo

La solution de TMS Software



# Les composants SVG de TMS Software

- C'est avec les mises à jour de décembre 2019 de ses suites [VCL UI Pack](#), [FNC Core](#) et [FNC UI Pack](#) que TMS Software a ajouté la prise en charge du format SVG dans ses composants.
- Le support du SVG a été ajouté dans les conteneurs d'images utilisés dans leurs librairies par leurs composants.
- Ils proposent également un descendant de `TImageCollection` ajoutant le SVG aux formats habituels. Il permet de fournir des images SVG aux composants `TVirtualImageList` fournis par Embarcadero.



# Conclusion



# Conclusion

- La question de la qualité visuelle de nos logiciels est une question d'actualité pour tout projet.
- Vous vous devez de vous la poser lorsque vous développez ou migrez vos logiciels vers Windows 10, macOS, Linux, iOS et Android.
- Delphi et C++Builder prennent en charge la résolution des écrans sur les plateformes sur lesquelles ils tournent.



# Conclusion

- La gestion des images de tailles différentes est en standard pour la VCL et FMX.
- Les dessins vectoriels permettent de s'affranchir de problèmes d'encombrement tout en garantissant un affichage pérenne quelle que soit l'évolution des écrans sur le long terme.
- En revanche ils alourdissent les temps de réponse si on ne gère pas de cache sous forme d'image classique dans les composants servant à les afficher.



# Conclusion

- Réfléchissez à ce qui est le mieux pour vous en tant que développeur et surtout pour vos utilisateurs qui auront une bonne ou mauvaise impression selon que vos fiches sont belles ou détériorées lorsqu'ils ont de bons écrans.
- Faites ce choix selon la qualité des interfaces utilisateurs désirées.



# Conclusion

- Comme pour les webinaires précédents, ce PDF et l'éventuel replay seront mis à disposition sur le blog <https://developpeur-pascal.fr>
- Vous y trouverez également sous peu une liste de bibliothèques gérant les SVG pour FMX et la VCL. J'ai fait une sélection non exhaustive par rapport à mes tests, les échos que j'en ai et la description faite par leurs éditeurs.
- N'oubliez pas en faisant votre choix qu'il vous faut quelque chose de pérenne et maintenu si vous l'intégrez à vos projets. Alors testez.



# Prochainement



# Prochainement

- Barnsten propose régulièrement des événements liés aux produits Embarcadero: formations inter entreprises, rencontres développeurs et meetups, conférences, présentations de produits.
- Si vous désirez participer ou organiser quelque chose chez vous contactez les sans hésiter sur <https://www.barnsten.com>



# Prochainement

- Le numéro 237 du [magazine Programmez!](#) proposera un article d'initiation à FMX Linux. Il sortira en fin de semaine ou début février.
- Si vous avez des questions sur FMX Linux et le compilateur Linux fournis avec Delphi Entreprise, envoyez les moi sur <https://www.developpeur-pascal.fr/contact/>

LINUX / IA / DENO / BASIC / BLAZOR / ESP / KOTLIN

# PROGRAMMEZ!

Le magazine des développeurs

02/2020 N°237

**DINO**  
Le nouveau back-end JavaScript

Le retour de **DELPHI** sur LINUX  
**DX**  
avec FMX LINUX

Tout savoir sur **PHP 7.4**

**Kotlin**  
pour les dévs Java

**Blazor :**  
une techno prometteuse

Le seul magazine écrit par et pour les développeurs

PHP / WINDOWS / SQL / THREADS / C++ / ROOT ME / LARAVEL / GIT

Printed in EU - Imprimé en UE - BELGIQUE 7 € - Canada 9,80 \$ CAN - SUISSE 13,10 FS - DOM Surf 7,50 € - TOM 1020 XPF - MARIQ 55 DH



# Prochainement

- Notre prochain webinaire sera annoncé sous peu. Il devrait avoir lieu le 25 février.
- Des ateliers et des sessions de formation inter entreprises sont envisagées.  
Contactez-moi sur <https://se-former-a-delphi.fr/contact/> si vous désirez être prévenu ou voulez une formation chez vous sur des sujets précis.



# Questions / réponses



(c) Patrick Prémartin / [developpeur-pascal.fr](http://developpeur-pascal.fr)