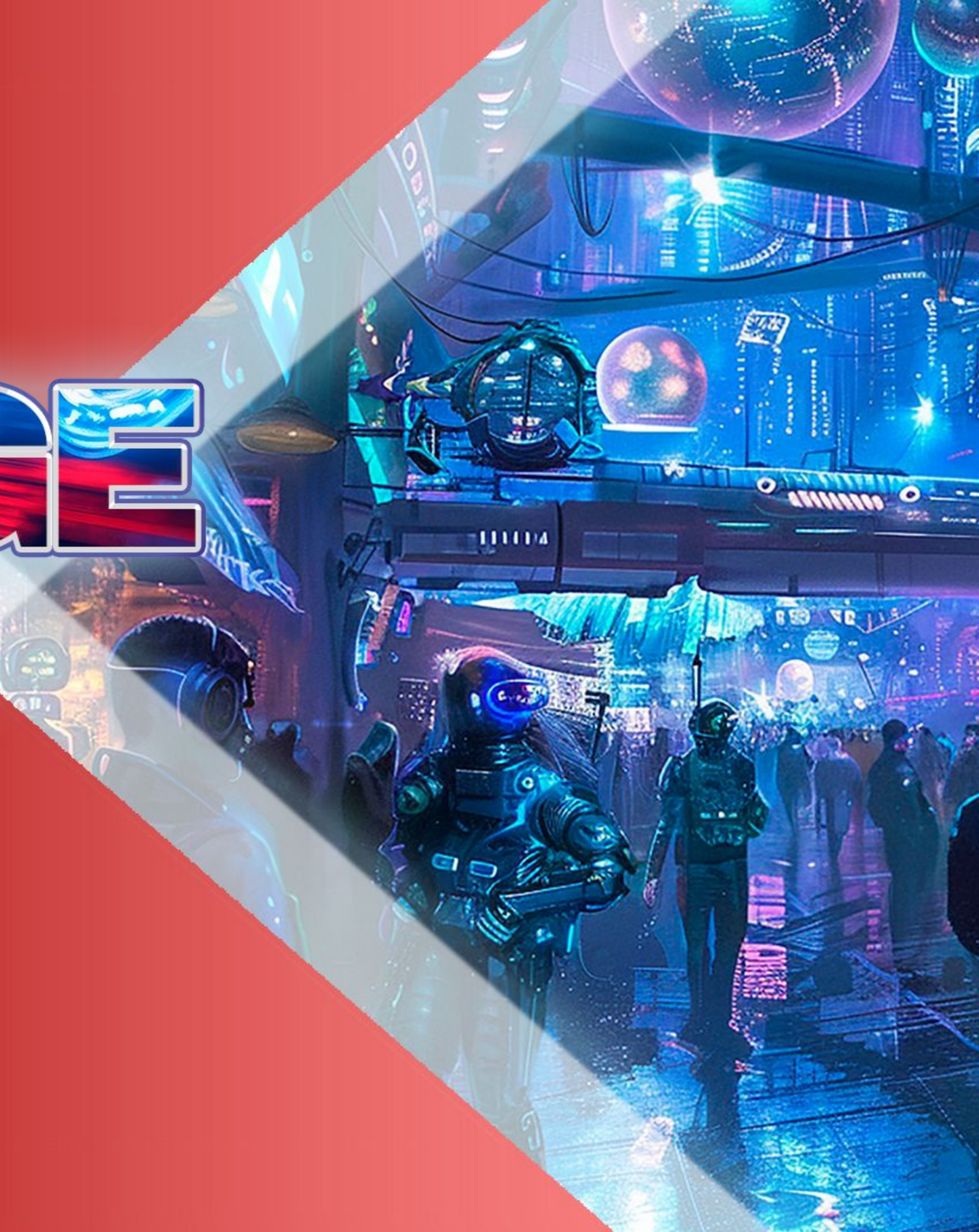
CODERAGE 2025



December 1-5 / 8-12

10 am -4 pm (CST)

embarcadero®

The Christmas Game Evolution

Patrick Prémartin
MVP Embarcadero, freelance developer and trainer

Agenda

- Introduction
- The video game
- The game engine
- The graphics
- The music and sounds
- The game scenes
- Conclusion

Introduction

Introduction

Once again, I invite you to dive into the world of video game programming in Pascal with Delphi and FireMonkey for Windows, Mac, iOS, Android, and Linux(*).

The reasons for choosing this topic remain the same:

- making games means asking questions that we don't always ask ourselves when coding management applications.
- making games means taking on unusual challenges that can help us in our daily work on all types of projects.

(*) Linux is available with Enterprise, Architect, and Academic licenses.

Who am I ?

MVP Embarcadero, freelance developer,
Delphi and web trainer, content creator.

To contact me:
https://vasur.fr/about

Regularly on Twitch:
https://www.twitch.tv/patrickpremartin

Open source projects:
https://codeberg.org/PatrickPremartin
https://github.com/DeveloppeurPascal

Patrick PREMARTIN
(on the right in the photo)

Some of my websites

Catalog of books on Delphi and Pascal
https://delphi-books.com

Blog on Pascal development in Delphi
https://developpeur-pascal.fr

Self-learning Delphi via VOD
https://apprendre-delphi.fr

Utilities for Delphi and C++Builder developers
https://getitnow.embarcadero.com/vendor/patrick-premartin/

Patrick PREMARTIN
(on the right in the photo)

Introduction

For the CodeRage 2018 conference, I created “Christmas Game,” a very simple video game.

In it, I showed how to use standard components that come with FireMonkey to create animations and interact with players. The game was coded in less than two hours and could be played on Windows, Mac, iOS, and Android.

It has been recompiled for Windows and macOS using Delphi 13 Florence. You can download it from its new website.

https://christmasgame.gamolf.fr

Introduction

This year, I'm repeating the exercise, combining:

- elements provided with Delphi 13 Florence and Delphi Community Edition
- open source projects, including the video game starter kit launched last year
- software developed to simplify my life and yours

The game created to illustrate this session is called “Christmas Game Evolution”(*).

You can play it on Windows and macOS with the “CodeRage 2025 version” of the game. It can be downloaded with its source code from its website.

https://christmasgameevolution.gamolf.fr

(*) You escaped “Christmas Max Game Ultra Pro Edition ++”. Too complicated to pronounce...

Introduction

This PDF, the list of links, the replay of the presentation, additional videos, and some behind-the-scenes videos will be available after CodeRage 2025 from this code repository:

https://apprendre-delphi.fr/cr2025-pdf-download

The video game

Why this game ?

For the past two months, I've been playing Plants vs. Zombies 2 on my iPad.

It's an “old” “free” game that offers different activities based on the same principle: zombies attack a garden. The player must grow plants to defend themselves. Between advertisements and paid offers, the game incorporates addictive mechanics to encourage players to stay or come back.

Why this game ?

Plants vs. Zombies 2 inspired me to create something based on the same concept.

With the CodeRage 2025 conference approaching, it was the perfect opportunity to develop a new winter-themed game centered around Christmas, gifts, and toys.

Christmas Game Evolution offers several activities: a clicker (as in 2018), a match-3 game, an inventory of items to collect, and a system of goals.
The player can no longer loose.
They must achieve the best score possible.
They access the activities through the goals to be achieved.
Other activities may be added later.

The CodeRage 2025 version of the game

The version I am going to show you was developed and refined on Twitch during a week of live coding sessions in November 2025.

It does not offer all the activities. Some are limited. The game does not support saving player progress.

A more complete version for the general public will be developed and made available for purchase on the usual app stores.

As with the 2018 game, I will leave the CodeRage version available for download from the game's website.

Christmas Game Evolution

Before getting into the heart of the game, I'd like to show you a short demo dated November 24, 2025.

It's silent, so don't worry if you don't hear anything.

https://christmasgameevolution.gamolf.fr/videos-demos.html

Christmas Game Evolution

Apart from marketing and storyline, video games essentially depend on three things:

- the game engine
- the graphics
- the sounds (background music and sound effects)

“Christmas Game Evolution” is no exception to this rule.

The game engine

The game engine

At the 2024 <Dev Days of Summer> conference, I presented the brand new open source project “Gamolf FMX Game Starter Kit.”

You can watch a replay of this presentation on Embarcadero Technologies' YouTube channel or on this page:
http://apprendre-delphi.fr/ddos2024-makegames

A presentation in French was given a few days later:
https://apprendre-delphi.fr/ddos2024-makegames-fr

Other videos about the starter kit or the development of video games based on it can also be viewed on YouTube and PeerTube.

The game engine

“Gamolf FMX Game Starter Kit” is a set of native features for creating video games as FireMonkey projects in Delphi.

The starter kit supports scene management, button management and other user interface elements, background music and sound effects, text translation, and more.

It uses several other more generic or video game-oriented open source projects.

For more information, visit its website.

http://fmxgamestarterkit.developpeur-pascal.fr

The game engine

To use the starter kit when creating a video game, there was the method I used, which involved creating a wizard and referencing it in the IDE, or providing a ZIP file of the preconfigured project itself.

I opted for the third solution. A new code repository has been created. You can use it by cloning it, using it as a template for your repositories, or using it as a basis for copying and pasting.

A downloadable ZIP version(*) is also available.
https://codeberg.org/DeveloppeurPascal/Game-Template-For-Delphi-FireMonkey

(*) The ZIP file should be available on GetIt when this presentation is released.

The game engine

Video games based on the starter kit also depend on the following projects:

- “About Dialog Delphi Component” serves as my standard dialog box, displaying version information, license, and project descriptions. It is activated by pressing F1 in all my games.
  https://dialogueapropos.developpeur-pascal.fr

- “CilTseg For Delphi” is the Delphi client for the CilTseg license and version manager API. Currently in limited public distribution, I use it in all my softwares, even though it is not always activated.
  https://ciltseg.olfsoftware.fr

The game engine

Video games based on the starter kit also depend on the following projects:

- “Delphi Game Engine” contains management routines for game controllers, manipulating user interface elements (real controls or simple drawings), visual help bars, components, score management, sound and music management, etc. https://delphigameengine.developpeur-pascal.fr

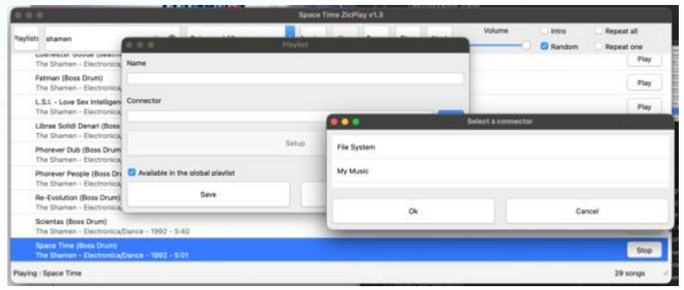
- My libraries (“toolbox”) for Pascal developers, where you will find basic routines such as (very light) encryption, parameter file management, data manipulation, extensions or enhancements for RTL, VCL, FireMonkey, or Skia. https://librairies.developpeur-pascal.fr

The game engine

These code repositories, libraries, and components can of course be used individually, whether in a video game or not.

The “About Box” and my toolbox are included in all my software.

Video game-oriented repositories are also sometimes used for other projects. For example, I use the music management classes from “Delphi Game Engine” in my MP3 player ZicPlay, which you can find at https://zicplay.olfsoftware.fr

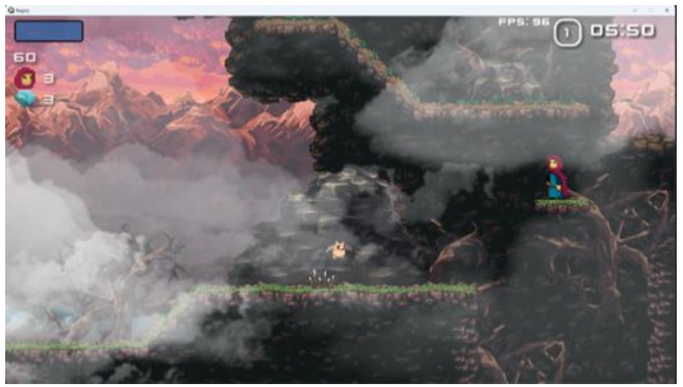


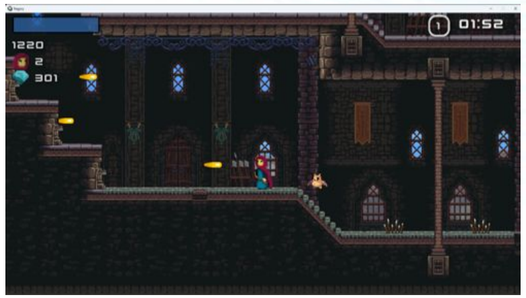
The game engine

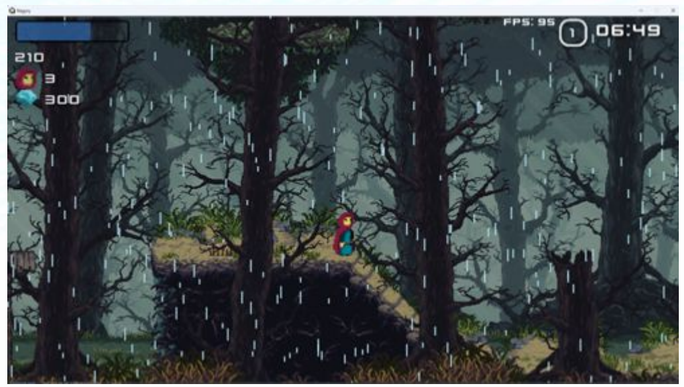
You can also use parts of my libraries in your projects, as Grégory Bersegeay (Embarcadero MVP) did to make “Nagory” playable with a game controller on Windows and Mac.

“Nagory” is a video game developed entirely in Delphi. Grégory spent almost two and a half years of his free time on it (developing the engine and game design). His game was released on December 2, 2025, for Windows, Mac, Linux, and Android.

https://gbesoft.fr/Nagory/







Reminders : user licenses

The source codes for everything I distribute are open on Codeberg or GitHub with an AGPL v3 or MIT license.

My libraries for Delphi and websites are distributed under the AGPLv3 license.

To support my work or if the AGPLv3 license is not compatible with your project, purchase a "developer license" at https://store.developpeur-pascal.fr

My software is distributed free of charge (as shareware).

To support my work or if you earn income from using my programs, you are welcome to purchase a license at https://store.olfsoftware.fr

Reminder: if it's free, sometimes you have to pay…

In general, whenever possible, support:

- developers of open source projects and “free” software,
- artists whose works you watch, listen to, or use,
- content creators who entertain you and share their passions and knowledge with you,
- …

They too need to eat, have a place to live, and time to devote to these unpaid activities.

The graphics

The graphics

For “Christmas Game Evolution,” only the text displayed in the game credits goes through a standard display component (a TText). Everything else is rendered graphically from bitmaps or SVG vector images (using Skia and Skia4Delphi).

For this, I use TImage, TRectangle, and TCircle components.

Since I didn't have any CPU-intensive animations to do, I didn't need to draw live on the canvas of the main form or the frames that serve as screens.

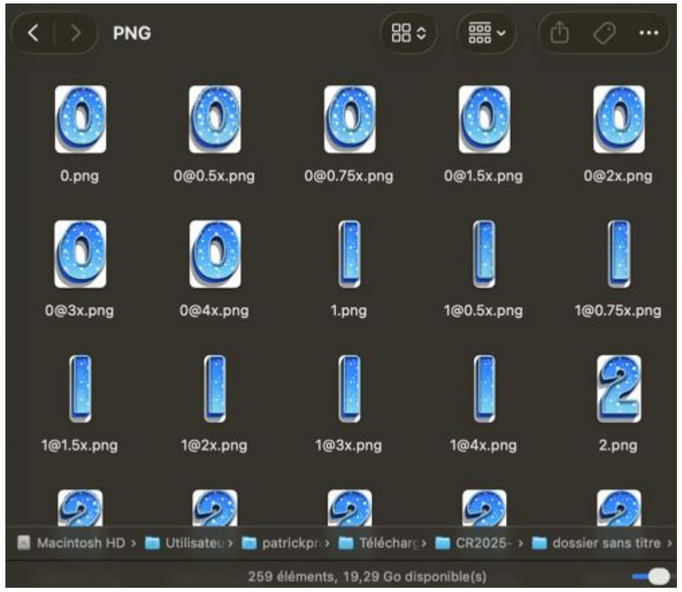
The graphics

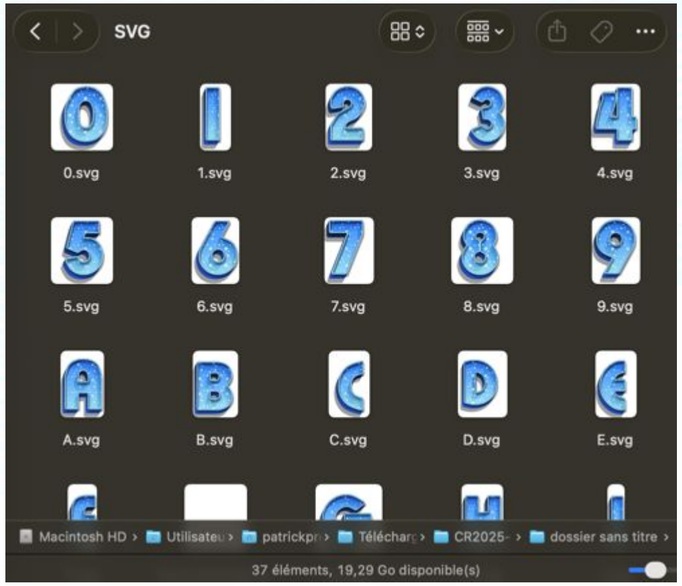
Since I am primarily addressing developers, I will assume that drawing tools are not your cup of tea as they are not mine.

Let's see how I started with Adobe Illustrator images from Adobe Stock to obtain text or drawings in the game screens.

1 - Open and cut the Illustrator file to obtain PNG and SVG files
2 - Bulk import the PNGs into image lists for FireMonkey
3 - Import the SVGs as character strings into the project
4 - Display graphic text
5 - Display graphic text on multiple lines
6 - Use SVGs to draw on the screen

Obtain PNG and SVG files from the original images

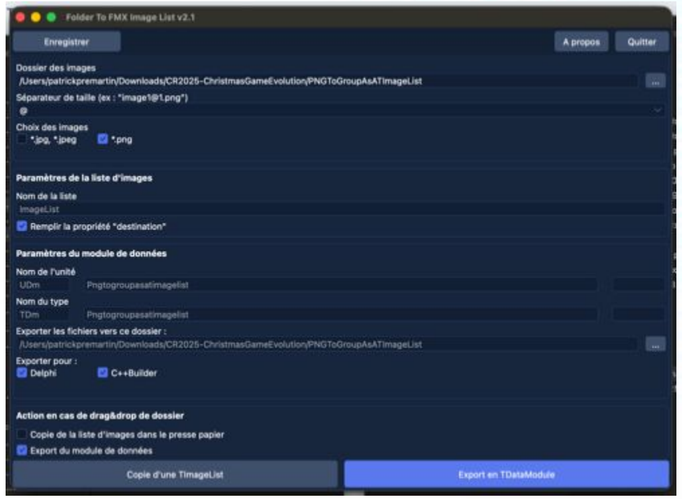


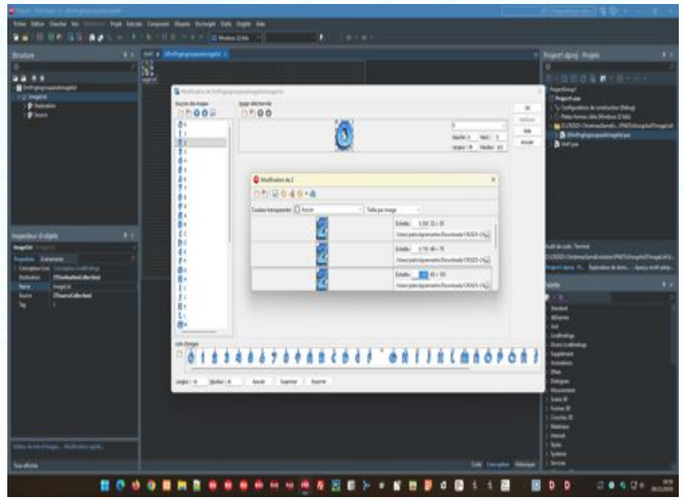


Switching from a series of images to a TImageList

In VCL, the TImageCollection component allows you to import files of different sizes from a folder. In FireMonkey, the TImageList component does not offer this feature. This makes the operation long and tedious depending on the volume of images.

I developed “Folder To FMX Image List” to remedy this. Download it from GetIt or its website https://folder2fmximagelist.olfsoftware.fr

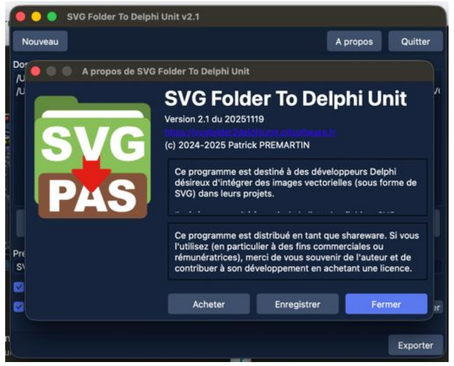


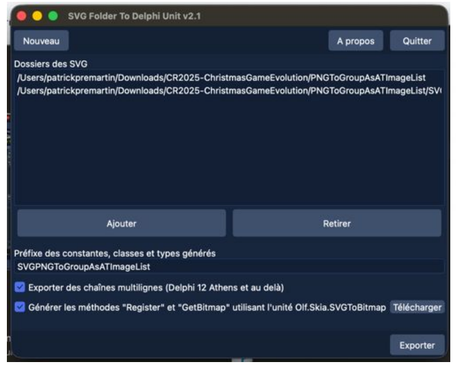


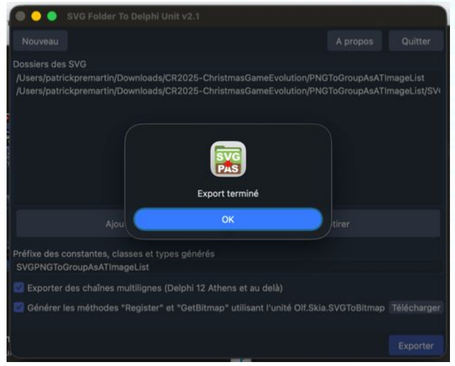
Converting vector images to source code

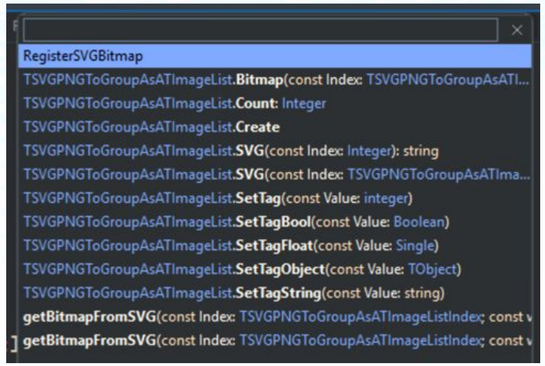
I demonstrated how to manipulate SVG in projects during the 2024 <Dev Days Of Summer>. View the presentation “Use SVG images in Delphi” at https://apprendre-delphi.fr/ddos2024-use-svg

In my software, I prefer to manipulate SVG via code. I developed “SVG Folder To Delphi Unit” to import these vector images directly into Pascal source code. Download it from GetIt or its website: https://svgfolder2delphiunit.olfsoftware.fr









Display text from a bitmap font

In my toolbox, you will find the Olf.FMX.TextImageFrame unit. It contains the TOlfFMXTextImageFrame class (a descendant of TFrame) which displays text from letters stored in a FireMonkey TImageList.

The class simply creates TGlyph components associated with the TImageList in a TLayout.

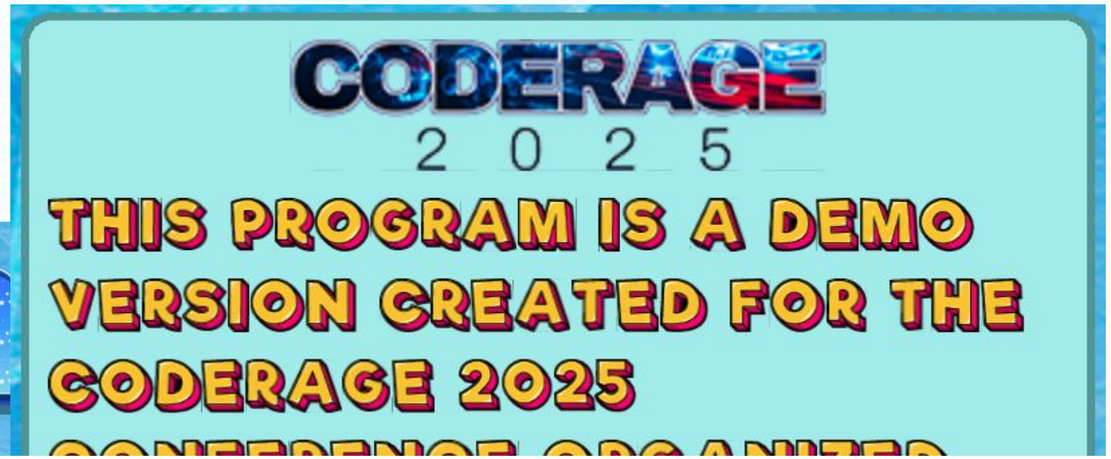
With this code in a TFrame, I can easily display the game title wherever I want in my project.

```
ftxt := TOlfFMXTextImageFrame.Create(self);
ftxt.Parent := self;
ftxt.AutoSize := true;
ftxt.Font := dmAdobeStock_296550198.ImageList;
ftxt.Text := CAboutGameTitle;
ftxt.HitTest := false;
ftxt.Align := TAlignLayout.Center;
```

Switching from single-line text to multi-line text

We know how to display text graphically.
We know how to stack components in a space that automatically adjusts in size.
So we know how to go from one line to many!

All we need to do is split the text into words, create a graphic text for each word, and put everything in a TFlowLayout.



Display text from a vector font

Based on the same principle as text from bitmap images, displaying text from letters stored as SVG images is fairly simple: just draw one image per letter and place them in the right place in a global container.

For the menu button text, I created TImage objects in a TLayout.

What about the icons and game elements?

All other graphic elements in the game are drawn from SVGs in TRectangle.

The display on the screens is a stack of TLayout, TFlowLayout, and TVertScrollBox with the correct alignments to ensure that the game would be visible on as many devices as possible.

This game makes extensive use of TFrame and its inheritance for reused elements. This is simpler than creating components and maintaining them.

Where can you find graphics for your software?

Don't take anything from search engines or websites you like (nor dislike). All graphic elements are someone's intellectual property.

Many sites offer royalty-free or non-royalty-free graphic elements, either free or for a fee.

For my video games, I usually use Kenney.nl, Adobe Stock, Cartoon Smart, and Open Game Art. Humble Bundle also regularly offers attractive packages for 2D and 3D software.

If you have a budget, don't forget that illustrators and graphic designers can create images tailored to your projects and only your projects.

Music and sounds

Music and sounds

In a video game, there are two types of sounds: background music and sound effects.

In both cases, I use the TMediaPlayer component provided by Embarcadero.

In reality, I don't use it directly.

I go through the Gamolf.FMX.MusicLoop unit of “Delphi Game Engine” wherever I need to play music or sound.

Music and sounds with Delphi Game Engine

In the Gamolf.FMX.MusicLoop unit, you will find:

- The TMusicLoop class for playing MP3, WAV, and M4A files (depending on the OS and CODEC availability). It manages loop playback.

- The TSoundList class for referencing a series of tracks and playing them on demand. It allocates a pool of TMediaPlayer components to enable the playback of several sounds at once.

You can use this unit in your game projects, as in other softwares, to add sound to the user interface, add sound to input errors or dialog boxes, clicks, the end of tasks launched by the user, etc.

Music and sounds with the starter kit

The “Gamolf FMX Game Starter Kit” manages background music and sound effects as standard.

For music, simply fill in the CBackgroundMusicFileName constant in the uConsts unit. It will automatically start when the game loads and will play on a loop.

For sound effects, you need to customize the uSoundEffects unit.

Configuration parameters included as standard in uConfig manage activation and sound volume.

The musics in Christmas Game Evolution

In the CodeRage 2025 version of the game, I wanted to offer several background music tracks for the user to choose from.

This required modifying the uBackgroundMusic unit in the starter kit slightly, while maintaining compatibility with all existing softwares.

It is now possible to define the music to be played by code by assigning it to the variable TBackgroundMusic.Current.MusicFileName.

The starter kit takes care of the rest without having to modify anything else.

The musics in Christmas Game Evolution

In the game, I defined a type of music to play: 1, 2, or 3.

The change of MP3 file to play is placed in uCGEConfig(*) when changing the option from the settings screen.

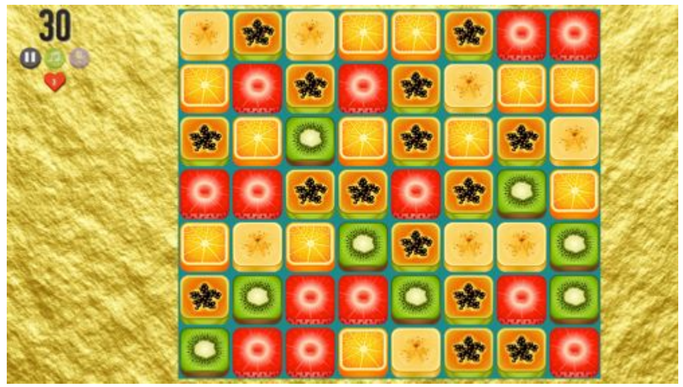
I also change the file name in the unit initialization so that the starter kit knows which music to play when the game starts.

(*) This has nothing to do with the excellent Castle Game Engine. I realized later that I was using the same initials.

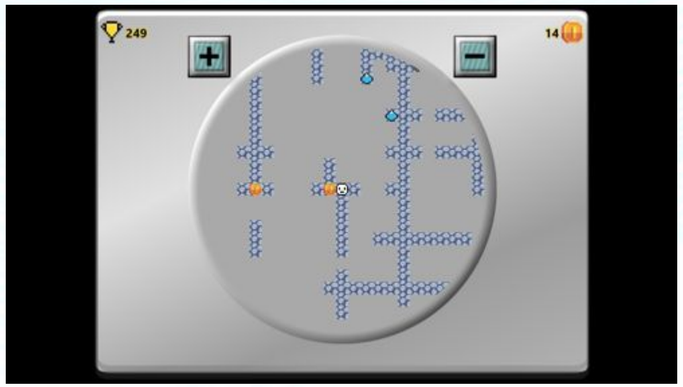
Sound effects with the starter kit

The CodeRage 2025 version of the game does not have sound effects. They will be added for the retail version.

If you are looking for an example of how to use the uSoundEffects unit to reference and play sounds in your video games, check out the source code for these:



Bidioo
https://bidioo.gamolf.fr



DAD48
https://dad48.gamolf.fr

Where can you find music and sounds for your games?

As with images, using files found on search engines is not recommended. If you do not have explicit permission to use a piece of music, do not do so. You would run the risk of having your software or videos rejected by distributors or of a complaint being filed by a rights holder.

For 20 years, I have been regularly purchasing royalty-free music packs that allow use in commercial projects. Feel free to do the same to create a small library that you can draw from when needed.

Note that this will not protect you from abusive DMCA claims or censored videos on YouTube. Audio rights management is complex and sometimes legally circumvented...

Where can you find music and sounds for your games?

To add sound to my video games, I draw on the licenses I own, but also from Humble Bundle, Open Game Art, Game Dev Market, and, more rarely, Fanatical.

I have also purchased licenses from Jamendo, such as for the music in the video game Pumpkin Killer, available at https://pumpkinkiller.gamolf.fr

The downside of these sites is the lack of exclusivity. Other products (software, video games, films, advertisements, etc.) may use the same music.

Where can you find music and sounds for your games?

If you have a budget and want a specific sound environment, you can also hire a professional composer.

Some are very easy to contact since they compose or play music live on Twitch, YouTube, or other platforms.

On Twitch, you will find hundreds of musicians of all nationalities and dozens of professional composers who accept one-off or regular assignments.

Among them, I can suggest at least three French composers: SarysFR, Loinduciel, and MisterMV.

Where can you find music and sounds for your games?

Xavier Dang (aka “Sarys”) is a composer, sound designer, and audio director.

He creates music for video games and other types of projects.

On Twitch, he composes music for fun or for his clients and plays video games.
https://www.twitch.tv/sarysfr/about

On YouTube, he shares excerpts from his compositions.
https://www.youtube.com/@Sarys

Together with MisterMv, they created the music for the video game “Noreya: The Gold Project.”
https://store.steampowered.com/app/1760330/Noreya_The_Gold_Project/

Where can you find music and sounds for your games?

Loinduciel is a musician.

On Twitch, he performs improvisations with piano or looper. He invents, composes, and mixes notes and instruments with disconcerting ease.
https://www.twitch.tv/loinduciel/about

On YouTube, he offers a selection of tracks created live on Twitch or at TwitchCon 2023, as well as piano ASMR and unreleased solo and group performances.
https://www.youtube.com/@loinduciel

Where can you find music and sounds for your games?

Loinduciel also composes on demand for various projects: video games, streams, charitable or commercial events, short films, and movies.

Feel free to send him your projects.

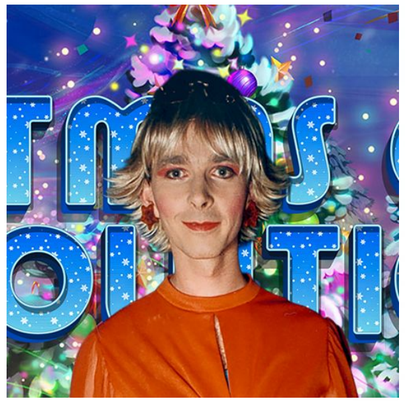
You may have seen him improvising on the word “Triskaidekaphobia” following a discussion I had with him after Ian Barker's presentation at the 2024 Brazilian conference.
https://makertube.net/w/3wEfN8Mmt8crUhqTkSkc31

He also composed the music for Ploumtris on his channel while I was coding on mine during the Kenney Jam 2023 game jam.
This game is available at https://ploumtris.gamolf.fr

The musics of Christmas Game Evolution

For the CodeRage 2025 version of the game, I wanted to innovate and offer you some original melodies on the theme of Christmas and the holiday season.

I called on three performers I like: Gladys from Tango Paris, Col Argol from the Pyrenees, and a friend from Hawaii who preferred to remain anonymous.



It was a great moment of humility and professionalism.
Replays of the recordings will be published with the PDF of this presentation.

The game scenes

The scenes in Gamolf FMX Game Starter Kit

Scenes are the screens in the game.

Each scene must descend from the T__SceneAncestor class.

Each scene must register to TScene.Current during its initialization.

Each scene corresponds to a scene type declared in uConsts.

Only one scene should be declared for a scene type, but you can reference several different types for the same scene.

Scene instances are created when they are first displayed. They are retained until the program exits.

The scenes in Gamolf FMX Game Starter Kit

Each scene is independent of the others.

Each scene has its own loading (ShowScene) and unloading (HideScene) methods.

To fill in the text on the screen, override the TranslateTexts() method, which will be called when the scene is displayed or when the user changes the language in the software.

Displaying a scene of type "None" hides the current scene and displays nothing in its place.

Displaying the "Exit" scene type triggers the program to close, regardless of where you are in it, without saving the current part.

The scenes in Christmas Game Evolution

There are 8 scenes in this video game:

- uSceneSplashScreen: the start screen displaying an image and a progress bar
- uSceneClicker: the game screen for the clicker activity
- uSceneCredits: the game credits screen
- uSceneGame: the main scene that gives access to activities
- uSceneGameOver: the end-of-game scene for the match-3 or clicker game when all lives are lost
- uSceneHome: the start menu, giving access to games, options, and credits
- uSceneInventory: the list of items in the player's inventory
- uSceneMatch3: the match-3 activity using several series of images
- uSceneOptions: game option settings with radio buttons and checkboxes

The scenes in Christmas Game Evolution

Scenes are designed in the form designer with a stack of TFrame and code.

A TVertScrollBox component is usually present to support the height of smartphones and tablets. The entire interface must be accessible at all times if you don't want to be rejected by Google, Apple, Amazon, or elsewhere.

The scenes in Christmas Game Evolution

In the CodeRage 2025 version, the splash screen does nothing.

The progress bar is only there to show that the application has not crashed.

You can skip to the next screen by pressing ESC if you don't want to wait.

The scenes in Christmas Game Evolution

The clicker activity screen works in the same way as the 2018 "Christmas Game": elements are randomly selected and fall from the top of the screen.

The images are SVGs in TRectangle component backgrounds.

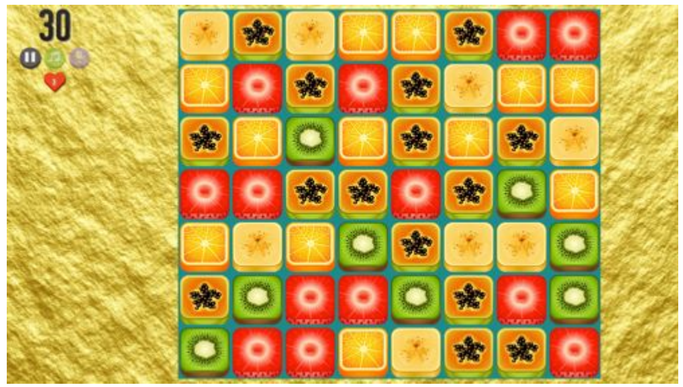
The number of lives, the score, and the number of Santas are handled by the game engine. Their display is processed in a TFrame common to several screens.

A change in the score, number of lives, or number of Santas in the current game properties triggers a message to be sent to subscribers of this type of operation, who refresh their screen area. I use System.Messaging for this (note that it is not thread safe).

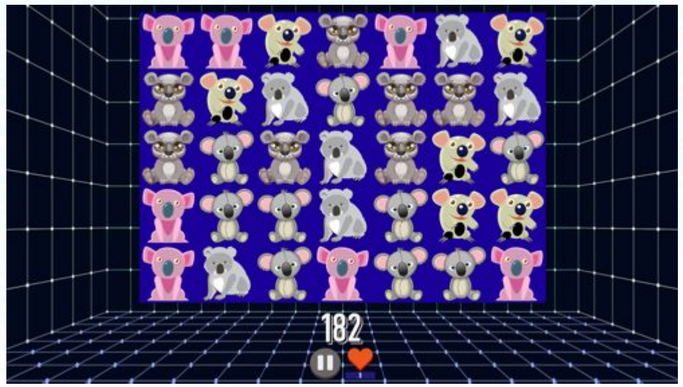
The scenes in Christmas Game Evolution

The match-3 activity is not coded in the game. It is a TFrame from a demo published in the "FMX Game Snippets" project on https://fmxgamesnippets.developpeur-pascal.fr

The video games Bidioo and Koala Match 3 also use it, but adapt how it works.



Bidioo
https://bidioo.gamolf.fr



Koala Match 3
https://koalamatch3.gamolf.fr

Conclusion

Conclusion

I hope you found this presentation interesting.

I wasn't able to go into detail about the project's source code. I hadn't planned enough time, nor did I anticipate that my "little demo project for CodeRage 2025" would actually turn into a monster...

If you have any questions about my libraries, the starter kit, the video game template, the source codes for the game in this presentation, or anything else, feel free to send them to me or join me to discuss them during a live stream on Twitch. I will try to answer in French or English.

Conclusion

The list of my video games is available at https://gamolf.fr

The list of my software is available at https://olfsoftware.fr/c/_1-logiciels--services.html

Each software, game, mobile app and library site now has a "dev corner" section at the bottom of the page.

There you will find links to software presentations, devlogs on blogs, source codes and developer documentation exported with DocInsight or PasDoc depending on the project, links to replays of coding sessions when they were developed on Twitch...

In short, enough to keep us busy for a few years and feed Tiobe. o:-)

Conclusion

Some of my software and libraries for Delphi or C++Builder are also available directly from GetIt: https://getitnow.embarcadero.com/vendor/patrick-premartin/

If you download open source projects from there, don't forget to copy them to your own directory tree before modifying them. This will prevent any unpleasant surprises in the event of automatic uninstallation of the concerned packages.

Conclusion

This PDF, the list of links, the replay of the presentation, additional videos, and some behind-the-scenes videos will be available after CodeRage 2025 from this code repository:

https://apprendre-delphi.fr/cr2025-pdf-download

Conclusion

Thank you for your attention (and thanks to DeepL for those slides translation from French to English).

See you soon somewhere online or IRL, who knows?

Thank You!