EKON

SKIA

- Grafische Brillanz für Ihre Anwendungen -

Dr. Veikko Krypczyk, 2024

http://larinet.com

…about us?

- LARInet (Learn Read Implement)
- IT-Dienstleister
  - Development (App/Web)
  - Delphi, WinUI 3, .NET MAUI, Blazor
  - Design
  - Konzepte
  - Workshops, Seminare

Wer?

Elena Bochkor
Veikko Krypczyk

Wo?

Erfurt

Kontakt:

https://larinet.com

Ihr erfahrt heute…

- Was ist Skia?
- Gründe für die Integration von Skia in Delphi
- Technische Features von Skia in Delphi
- Einsatz von Skia in VCL und FireMonkey (Beispiele)
- Leistungsanalyse und Vergleich
- Fazit und Ausblick

Quelle: https://pixabay.com/de/photos/buch-agenda-tisch-notizen-3043275/

Skia Grafik-Engine

Skia ist eine Open-Source-Grafikbibliothek, die von Google entwickelt wurde. Sie wird in vielen wichtigen Projekten verwendet, darunter Google Chrome, Android, Flutter und Firefox.

Ziel von Skia ist es, eine einheitliche API für 2D-Grafikoperationen bereitzustellen, die auf mehreren Plattformen funktioniert.

Hauptziel: Einfache **plattformübergreifende Grafikentwicklung, ohne auf systemnahe Grafik-APIs** wie DirectX, Vulkan oder Metal **zurückgreifen zu müssen.**

Geschichte und Entwicklung von Skia

2005: Google erwirbt das Skia-Projekt, um die Qualität und Leistung ihrer Anwendungen zu verbessern.

Skia wurde kontinuierlich weiterentwickelt und in viele Google-Produkte integriert.

Skia unterstützt die neuesten Rendering-Technologien wie GPU-Beschleunigung, Multithreading und moderne Grafikstandards.

Skia abstrahiert die Komplexität von niedrigstufigen Grafik-APIs und optimiert Zeichenvorgänge automatisch.

2005: Google übernimmt Skia.

2008: Integration in Android

2010: Integration in Chrome.

2018: Verbreitung durch Flutter

Heute: Kontinuierliche Weiterentwicklung

Verbreitete Anwendungen von Skia

- *Google Chrome*: Verwendet Skia für das Rendering von Webseiten und UI-Elementen.
- *Android*: Skia ist die zentrale Grafik-Engine für Android-Apps und das System-UI.
- *Flutter*: Nutzt Skia für plattformübergreifendes Rendering auf mobilen und Desktop-Plattformen.
- *Firefox und andere*: Viele Browser und Anwendungen verwenden Skia für effiziente 2D-Darstellungen.

=> Skia ist in vielen alltäglichen Technologien präsent und sorgt für schnelle und effiziente Grafiken.

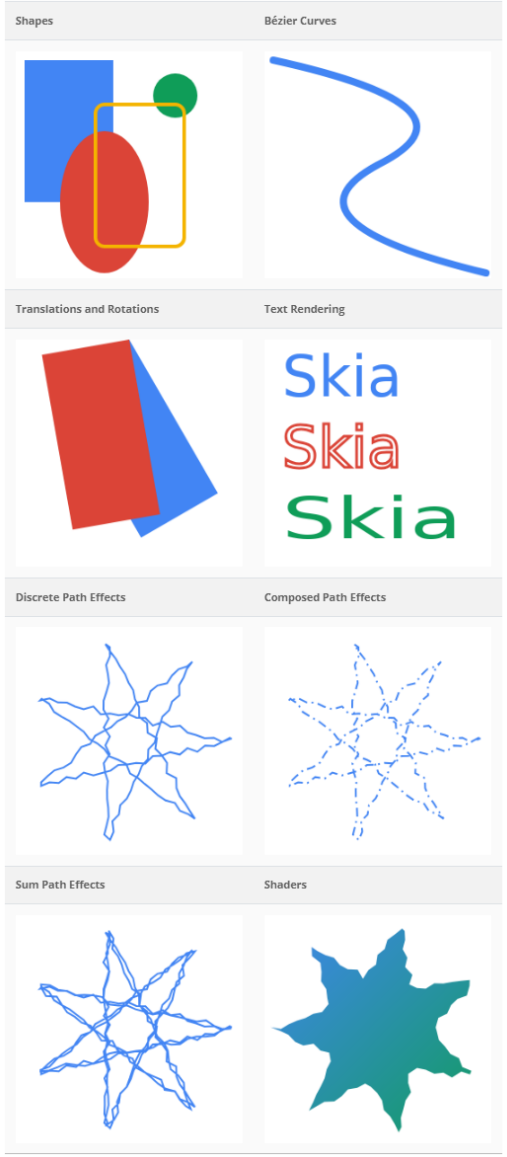
Technische Vorteile von Skia

- *Plattformübergreifend*: Funktioniert auf Windows, macOS, Android, iOS, Linux und vielen weiteren Plattformen.
- *Leistungsoptimiert*: Skia nutzt moderne GPU-Techniken und verbessert so die Rendering-Leistung erheblich.
- *Umfangreiche API:* Unterstützung für Text-Rendering, Vektorgrafiken, Bildmanipulation und PDF-Generierung.

=> Skia bietet eine flexible und leistungsstarke Lösung für alle gängigen Grafikoperationen.

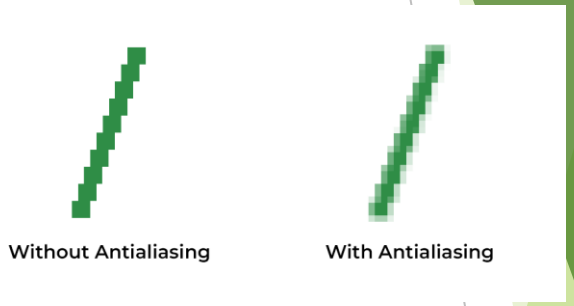
SKIA-FEATURES

- 2D DRAWING: Shapes, paths and texts
- SVG: Rendering and creation
- Image-Formate: BMP, GIF, ICO, JPG, PNG, WBMP, WEBP and raw images
- Aimationen: Lottie, Telegram Stickers, Animated GIF and Animated WebP



Weitere SKIA-Features

- *Antialiasing (Kantenglättung):* ist die Verminderung von unerwünschten Effekten, die durch das begrenzte Pixelraster oder durch den Treppeneffekt bei der Erzeugung einer Computergrafik entstehen können.
- *Erweitere Schrifteigenschaften:* Schriftstärke, benutzerdefinierte Schriftart, Zeilenanzahl, Zeilenabstand, Blocksatz, …
- PDF: Dokumente generieren



SKIA-Integration in Delphi

- Skia4Delphi ist eine plattform-übergreifende 2D-Grafik-API für Delphi und C++Builder, die auf der Skia Graphics Library von Google basiert.
- Kann zum Rendern von Bildern auf mobilen und Desktop-Modellen verwendet werden und ist mit den Frameworks (FMX und VCL) und den Zielplattformen kompatibel.

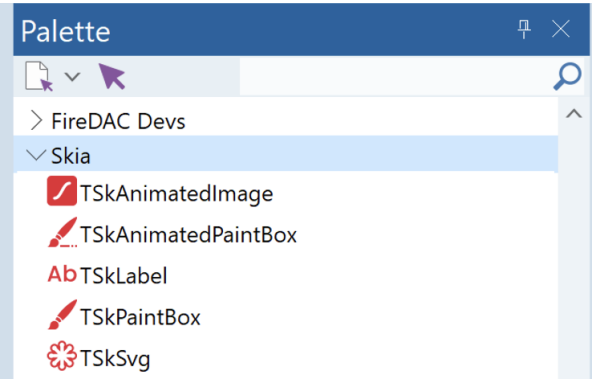
Feature set	Console	VCL	FMX	Description
Skia API	✓	✓	✓	Access to the pure Skia library, through a single unit: System.Skia.pas (or System.Skia.hpp)
Controls		✓	✓	TSkAnimatedImage, TSkLabel, TSkPaintBox and TSkSvg
App render			✓	Replacement of FMX graphics engine by Skia

SKIA in Delphi – Haupt Features

Feature	Details
2D draws	Shapes, paths and texts
SVG	Render and creation
Image decoders	BMP, GIF, ICO, JPG, PNG, WBMP, WEBP and raw images
Image encoders	PNG, JPG and WEBP
Animations player	Lottie, Telegram Stickers, animated GIF and animated WEBP
Anti-aliasing	High draw quality, no jagged edges
Font	Font weight, families fallbacks, custom font (without installation), ligature
Text	Multiple styles, max lines, line spacing, justified text, text outline, gradient, decorations
Right-To-Left languages	Rendering of texts in Persian, Arabic, Hebrew, etc
PDF	Generation of vectorized PDF
Unicode	Graphemes parser
Filters	Color, mask and image filters
Clippings	Support for many advanced clipping operations such as paths and shaders
Gradients	Linear, radial, sweep and conical gradients
Shader	Creation of shaders to execute specific draws directly on the GPU, through a single shader language (SkSL)

SKIA-Controls in Delphi

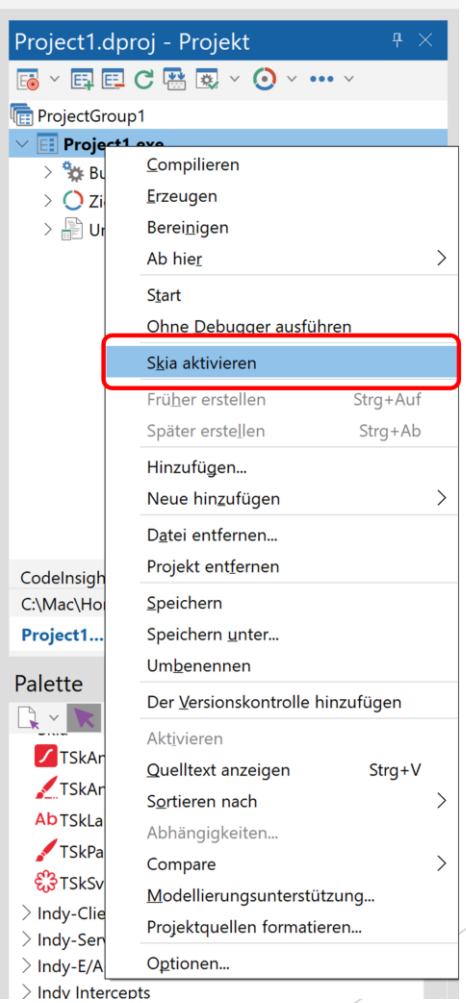
- TSkAnimatedImage
- TSkLabel
- TSkSVG
- TSkPaintBoxm & TSkAnimatedPaintBox



SKIA in Delphi-Anwendung aktivieren

- Bevor Sie eine Skia-Funktion verwenden, müssen Sie Ihr Projekt für die Verwendung von Skia aktivieren.

- Öffnen Sie dazu Ihr Projekt in RAD Studio, klicken Sie mit der rechten Maustaste auf Ihr Projekt und klicken Sie auf *Skia aktivieren*



TSkAnimatedImage

- *TSkAnimatedImage* kann animierte Bilder, einschließlich Vektoranimationen, auf sehr einfache Weise laden und rendern.
- unterstützten Formate:

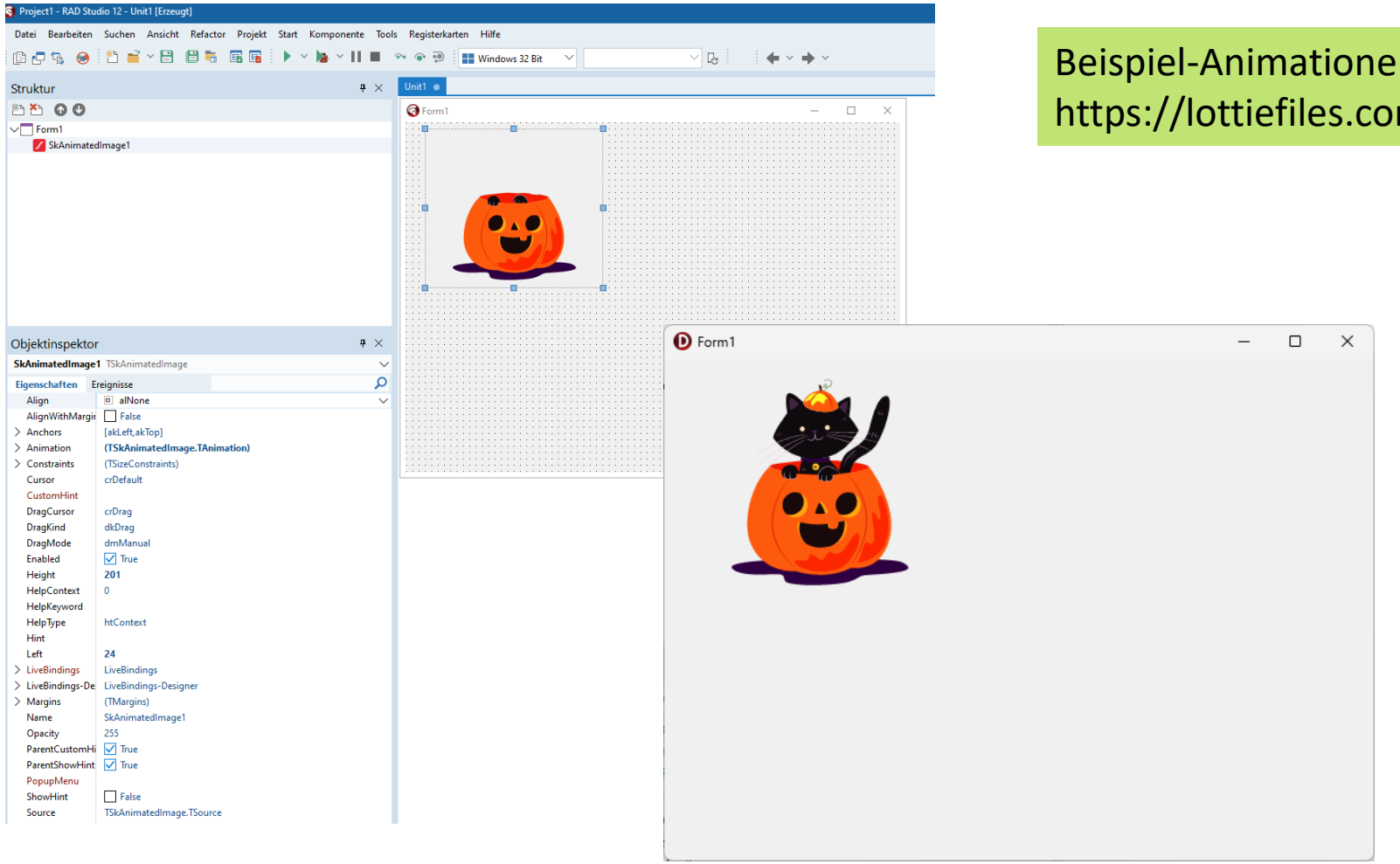
Format	Extension
Lottie file	.json, .lottie
Telegram Sticker	.tgs
Animated GIF	.gif
Animated WebP	.webp

TSkAnimatedImage: Beispiel Lotti-Animation

- Das Lottie-Animationsformat ist ein leichtgewichtiges, vektorbasiertes Format, das Animationen in Apps und Websites ermöglicht.
- Es basiert auf JSON, wodurch Animationen in sehr kleinen Dateigrößen gespeichert werden, was im Vergleich zu GIFs deutlich effizienter ist.
- Da es vektorbasierte Grafiken nutzt, können Lottie-Animationen verlustfrei skaliert werden, was sie ideal für unterschiedliche Bildschirmgrößen macht.
- Sie werden häufig für UI-Animationen und interaktive Grafiken verwendet und bieten eine plattformübergreifende Unterstützung.

```
var
    LAnimatedImage: TSkAnimatedImage absolute Result;
    begin
      LAnimatedImage := TSkAnimatedImage.Create(nil);
      LAnimatedImage.Align := alClient;
      LAnimatedImage.LoadFromFile(AssetsPath + 'check.json');
    end);
```

TSkAnimatedImage: Beispiel Lotti-Animation

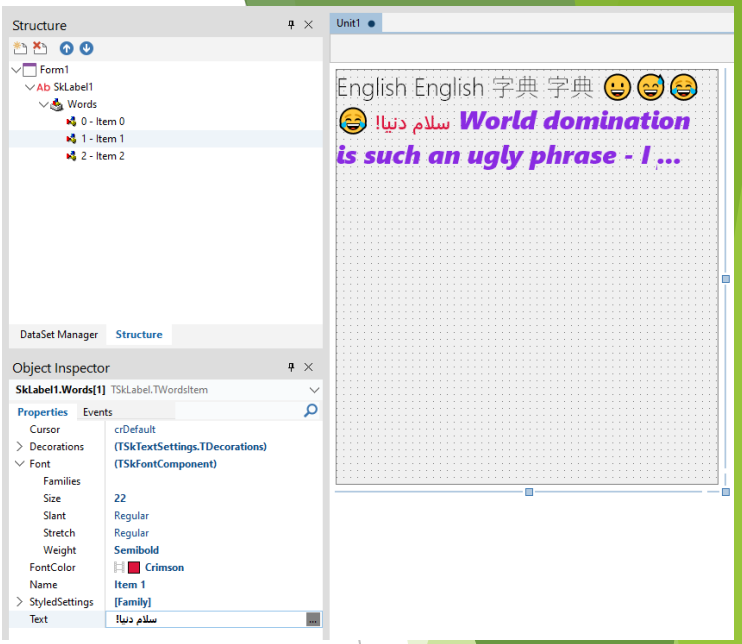


Beispiel-Animationen:
https://lottiefiles.com/

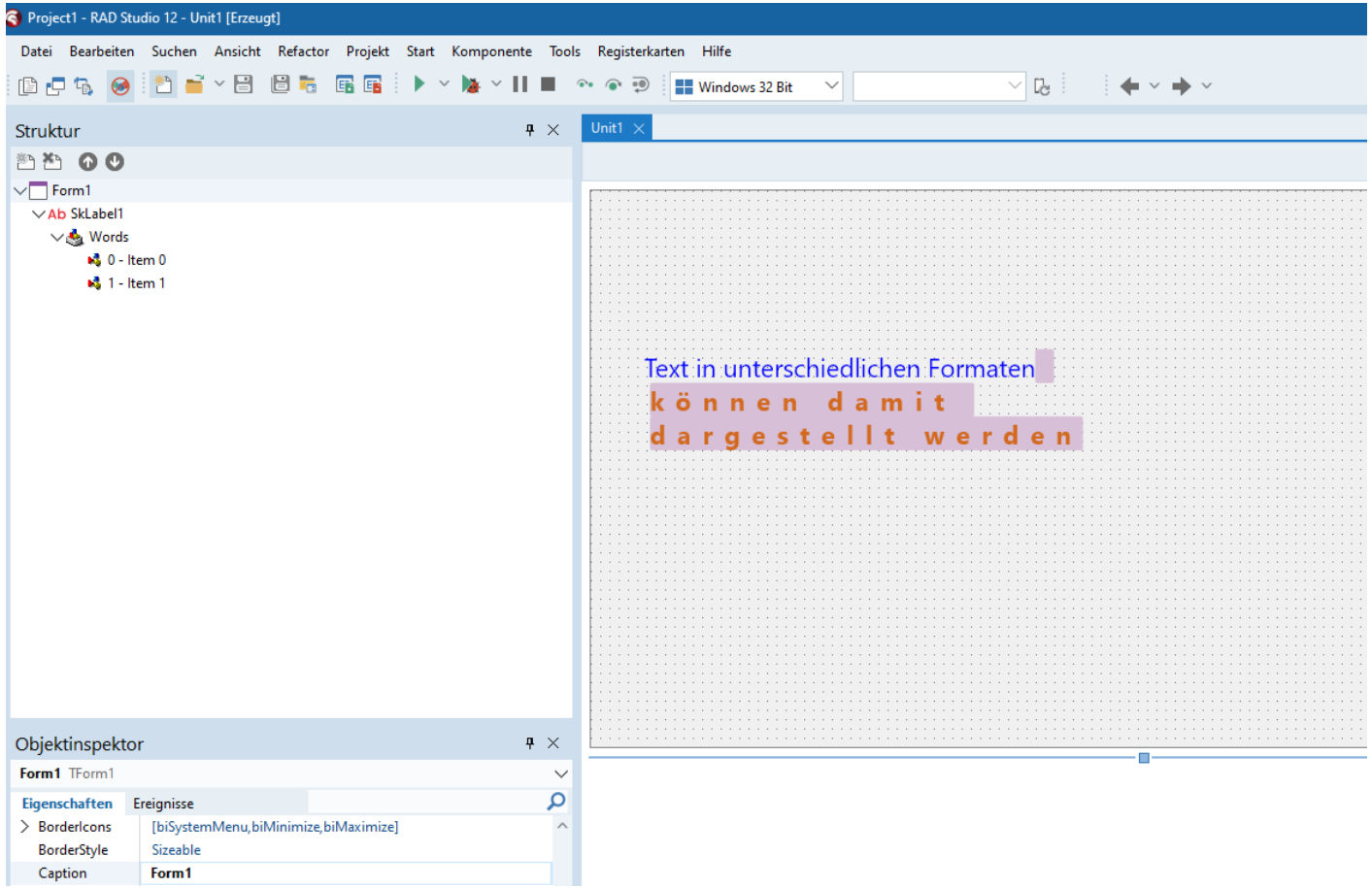
Beispiel: VCL/Lotti

TSkLabel

- Schriftfamilien
- Schriftstärke
- Schriftneigung
- mehrere Stile im Text
- von rechts nach links
- horizontale Ausrichtung im Blocksatz
- benutzerdefinierte Schriftarten
- Hintergrundfarben in Textteilen
- Begrenzung der maximalen Zeilenanzahl
- Automatische Größenanpassung (Breite und Höhe)
- Erweiterte Verzierungen (wie wellenförmige Unterstreichung, Überstreichung, gestrichelte Linie usw.)



TSkLabel



Beispiel: FMX/Label

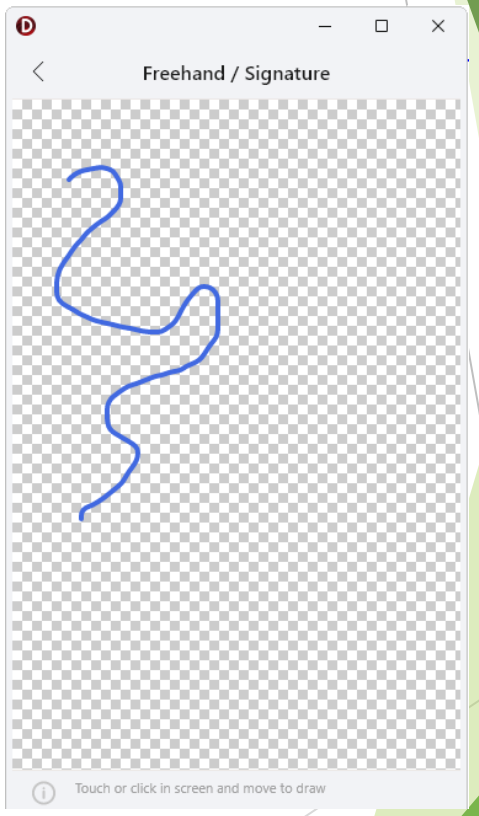
TSkSVG

- *TSkSvg* ist das Steuerelement zum einfachen Anzeigen von SVG.
- Fügen Sie das Steuerelement zum Formular hinzu und ein Doppelklick darauf öffnet das Fenster zum Laden der SVG-Datei.
- Mit der Eigenschaft *Svg.OverrideColor* ist es möglich, die Anzeigefarbe zu ändern.

```
var
  LSvg: TSkSvg;
begin
  LSvg := TSkSvg.Create(Self);
  LSvg.Svg.Source := TFile.ReadAllText('Samples\Demo\Assets\panda.svg');
  LSvg.Parent := Self;
end;
```

TSkPaintBox & TSkAnimatedPaintBox

- *TSkPaintBox* ist das Steuerelement zum Zeichnen mit Skia-APIs direkt auf dem Bildschirm mit dem *OnDraw*-Ereignis
- *TSkAnimatedPaintBox* ist eine Variante des *TSkPaintBox-Steuerelements*, mit der die Dauer einer Animation festgelegt und der Fortschritt dieser Animation mithilfe von Skia-APIs über das Ereignis *OnAnimationDraw* gezeichnet werden kann.



Neues Bildformart

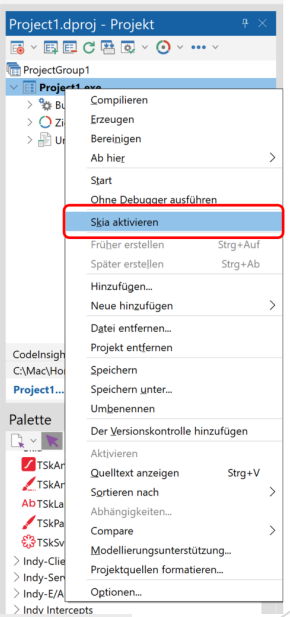
- Nachdem Skia im Projekt aktiviert wurde, werden neue Bildcodecs, die Skia unterstützt, automatisch im Framework registriert.
- Dann können die Steuerlemente *TBitmap* von FMX oder *TPicture* von VCL das Bildformat **WebP** direkt laden und speichern.
- Dabei handelt es sich um ein modernes Bildformat, das 30 % mehr komprimiert als JPG und PNG – und das bei gleicher Qualität.

App Render

- Optional ist es möglich, das Standard-Canvas von FMX durch ein Skia-basiertes Canvas zu ersetzen.
- Es werden alle **FMX-Steuerelemente** automatisch mit einer Skia gezeichnet.
- Damit ist es möglich, die Qualität und Leistung der Zeichnungen für **FMX** (primär Mobilgeräte) zu verbessern.
- Um App-Rendering mit Skia zu aktivieren, setzen Sie einfach die globale Variable *FMX.Skia.GlobalUseSkia* im Initialisierungsabschnitt auf „*true*“.

```
uses
  System.StartUpCopy,
  FMX.Forms,
  FMX.Skia,
  Unit1 in 'Unit1.pas' {Form1};

{$R *.res}
begin
  GlobalUseSkia := True;
  Application.Initialize;
  ...
```



Zusammenfassung: Kernfeatures Skia in Delphi

- Textstyling
- Transformation von Grafiken
- Text-Rendering, PDF-Erstellung
- Grafik-Primitive und Grafik-Pfade

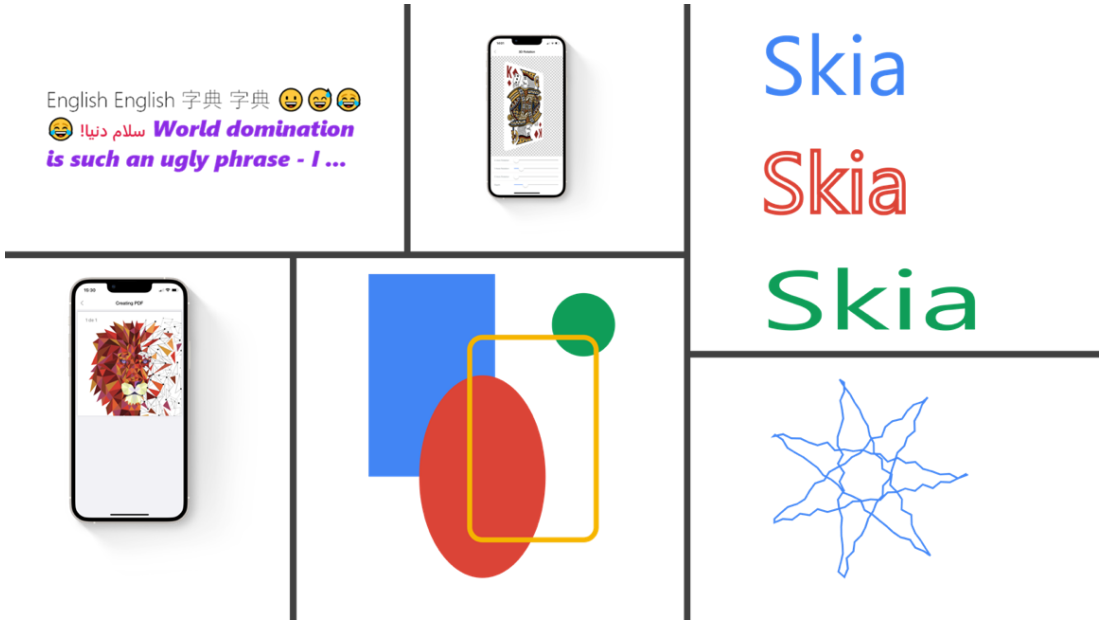
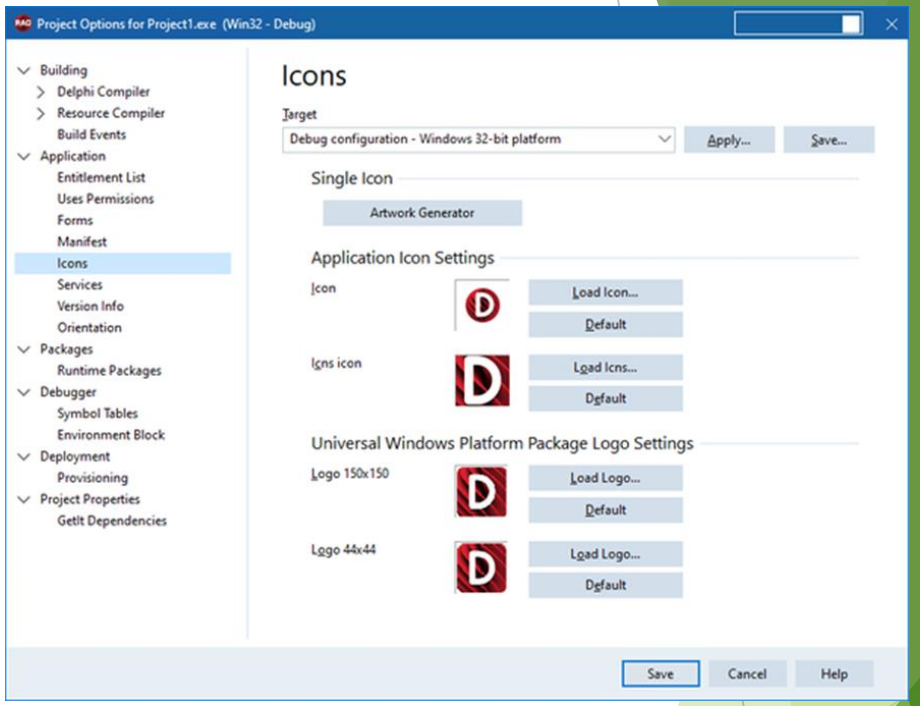


Image-Generator in Delphi (Skia basiert)

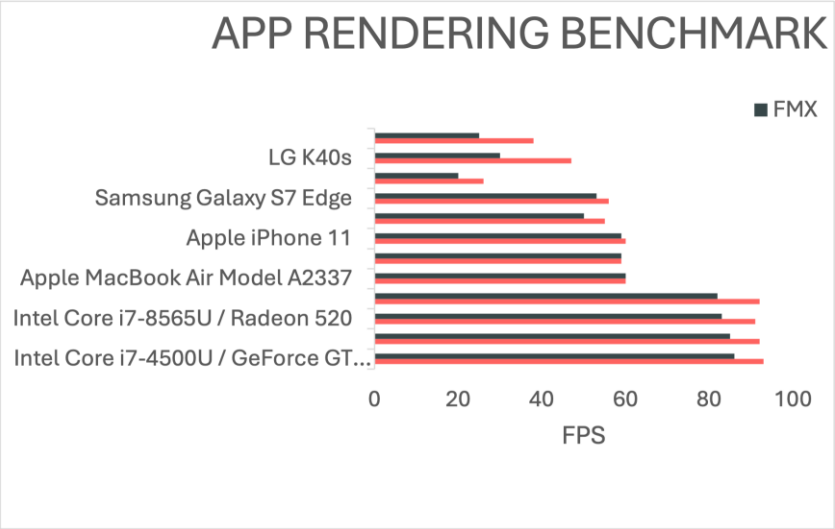
- Jede Plattform (Windows, Android, iOS, …) hat ihre eigenen Anforderungen an Bildelemente (Icons, Splash Screen usw.).
- Dabei müssen diese Bilder in unterschiedlichen Auflösungen erstellt und in den mit einer bestimmten Konvention für die jeweilige Plattform bereitgestellt werden.
- Der in Delphi 12 enthaltene **Multi Device Icon Generator** basiert auf Skia und ermöglicht es, dass man nach Auswahl einer Bildquelle und manuellen Anpassungen (Zoom, Hintergrund) automatisiert die notwendigen Bildelemente für alle relevanten Plattformen generieren kann

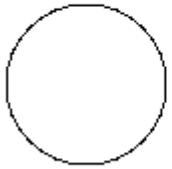
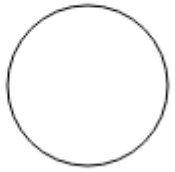


Leistungs- und Qualitätsvergleich

Device	Platform	FMX	Skia
Motorola Moto 3rd Generation	Android	25 fps	38 fps
LG K40s	Android	30 fps	47 fps
Samsung Galaxy A01 Core	Android	20 fps	26 fps
Samsung Galaxy S7 Edge	Android64	53 fps	56 fps
Samsung Galaxy S8 Plus	Android64	50 fps	55 fps
Apple iPhone 11	iOSDevice64	59 fps	60 fps
Apple iPhone 12	iOSDevice64	59 fps	59 fps
Apple MacBook Air Model A2337	OSXARM64	58 fps	30 fps *
Intel Core i7-8565U / Radeon 520	Win32	82 fps	92 fps
Intel Core i7-8565U / Radeon 520	Win64	83 fps	91 fps
Intel Core i7-4500U / GeForce GT 720M	Win32	85 fps	92 fps
Intel Core i7-4500U / GeForce GT 720M	Win64	86 fps	93 fps

Quelle: https://github.com/skia4delphi/skia4delphi?tab=readme-ov-file#fmx-render



FMX default	FMX with Skia render
	

Fazit & Ausblick

- Die Integration von Skia in RAD Studio bringt Vorteile im Bereich der grafischen Darstellung.
- Durch die plattformübergreifende Unterstützung, das hardwarebeschleunigte Rendering und die hohe Grafikqualität ermöglicht Skia die Erstellung moderner und grafisch reichhaltiger Anwendungen, welche mit einer guten Performance auf den Zielplattformen ausgeführt werden.
- Skia4Delphi ist einfach einzusetzen und kann für Applikationen mit einem Mausklick direkt in der IDE aktiviert werden.
- In diesem Fall wird das Standard-Rendering von **FMX** durch die leistungsfähigere Variante von Skia ersetzt.
- Für spezielle grafische Herausforderungen, wie Animationen (Lotti-Format), Textdarstellungen mit einem mehrfachen Styling oder flüssig darzustellenden Transformationen von Grafiken und geometrischen Primitiven stehen spezielle visuelle Controls zur Verfügung.
- Tests für eine Steigerung der Perfomance sind im Einzelfall notwendig, insbesondere auf mobilen Devices (keine VM nutzen)

Informationen/ Fragen

- Foliensatz

  https://larinet.com

- Fragen:

  v.krypczyk@larinet.com,

  https://larinet.com/

Vielen Dank für Eure Aufmerksamkeit!

Habt Ihr Fragen?

Gern sofort ☺

Allen Teilnehmenden noch eine schöne Zeit!

Quellen

- https://skia.googlesource.com/skia
- https://skia4delphi.org
- https://blogs.embarcadero.com/3-x-12-firemonkey-and-android-enhancements-in-delphi-12/#Skia_in_FireMonkey#
- https://github.com/skia4delphi/skia4delphi
- https://docwiki.embarcadero.com/RADStudio/Athens/en/Skia4Delphi
- Beispiele von Embarcadero [Ordner Samples]:
  - *FMX: Object Pascal\Multi-Device Samples\Skia4Delphi*
  - *VCL: Object Pascal\VCL\Skia4Delphi*