

# **Vector Graphics in Cuis**

**A pixel independent Zoomable User Interface**

**Juan Vuletich - March 31st, 2021**

# Morphic

**Morphic is the UI framework in Self, Squeak and Cuis**

# Advantages of Morphic

## Morphic gives you:

- *Complete* malleability
- Runs on *any* platform
- Looks and behaves *the same* on any platform
- Simpler *and* more flexible and general than regular window managers and UI toolkits

# **Non-disadvantages of Morphic**

**Morphic is different from other desktop UI toolkits because:**

- It doesn't attempt to offer a native look and feel
- Will not use features that are too platform specific
- Lets you focus on your application and your specific needs for the UI
- Doesn't force you to think in the terms of the platform or UI toolkit provider

**These are explicit choices to be made!**

# Morphic 2 Limitations

State of the art graphics, 20 years ago.  
Now, 20 years old graphics.

- Limited drawing API. Can't do SVG
- Limited support for TrueType
- No AntiAliasing
- Not a Zoomable UI
- Bad support for High DPI screens

Demo how Morphic 3 addresses all these issues!





# Morphic 2 programming complexity

Writing new morphs and morphic applications is hard because...

- It uses global coordinates only
- Integer, pixel oriented coordinates only
- Any scaling needs to be done in Morph code (at the application level)
- #containsPoint: Geometry needs to be implemented twice
- #bounds, #fullBounds: In many cases, geometry needs to be done thrice!
- Limited drawing API: Curves, for example, done at the application level

Show in code how Morhic 3 addresses  
all these issues!

# Morphic 3 implementation

## Own Vector Graphics Engine

- Doesn't use Cairo, Skia, or platform graphic APIs
- Doesn't use FreeType or platform text APIs
- Vector Graphics Engine written in Smalltalk
- Automatically translated into C and a VM Plugin
- Includes standard Vector Graphics primitives
- Top quality, sub-pixel precision anti-aliased rasterization

[Browse code to show the general structure of Morphic 3 and VectorEngine](#)

# Vector Graphics Engine

## Vector Graphics API and implementation

- Parameters:
  - Stroke width and Color (or none)
  - Fill Color (or none)
- A Shape is an open or closed sequence of:
  - Move
  - Line
  - Quadratic Bézier curve
  - Cubic Bézier curve
  - Partial or complete Circle or Ellipse
- Shape is blended on existing background using:
  - Stroke
  - Fill
  - Both
- Additional Shape operations for Text:
  - Primitives that iterate over a sequence of Byte Characters / UTF-8 / UTF-32
  - An optimized internal representation of TrueType fonts
  - Drawing Quadratic Bézier curves for each glyph

[Browse code for all this](#)



# **Vector Graphics based User Interface**

**A rather old idea (remember NeWS and Display Postscript)  
Whose time has finally come**

# **It is now possible to build GUIs that**

- Have the rich content and graphic quality of a good printed page
- Are scalable to whatever zoom level the user prefers
- Are interactive and react quickly to user actions
- Look good on any kind of Display and pixel resolution

# Cuis Smalltalk

## An Open Source Smalltalk system

- Simple enough to continue evolving instead of becoming legacy software
- Without becoming simplistic: a complete and portable Smalltalk system
- Running on the OpenSmalltalk virtual machine
- A practical system, used for teaching, Satellite image processing, research in signal, image and audio processing, research in programming languages, and many other areas of application
- <https://github.com/Cuis-Smalltalk/Cuis-Smalltalk-Dev>

# Morphic 3

## A Vector Graphics redesign of Morphic, for Cuis Smalltalk

- All coordinates are Floating Point (they don't specify pixels)
- All coordinates are relative to a Morph's local Coordinate System
- Coordinate Systems can be scaled and rotated (resolution independence)
- Vector Graphics primitives and art preferred to pixel oriented ones
- Rasterization done with a new Vector Graphics engine

# Documentation

## Chapter 7 of “The Cuis Book”

- <https://cuis-smalltalk.github.io/TheCuisBook/>
- <https://github.com/Cuis-Smalltalk/TheCuisBook/releases/download/20201230/TheCuisBook.pdf>
- Book written by Hilaire Fernandes with Ken Dickey and Juan Vuletich

# Vector Graphics Engine

## “Prefiltering Antialiasing for General Vector Graphics”

- Existing rasterizer such as Skia or Cairo are based on pixel coverage antialiasing. This are conventional Computer Graphics algorithms, where each pixel is considered a square area on the display.
- I developed a new technique based on Signal Processing theory, where each pixel is considered a sample in a signal.
- This allows much greater flexibility in the design of the antialiasing filter.
- The result is natural looking results without pixellation at any pixel resolution.
- Read about it: [https://www.researchgate.net/publication/267152327\\_Prefiltering\\_Antialiasing\\_for\\_General\\_Vector\\_Graphics](https://www.researchgate.net/publication/267152327_Prefiltering_Antialiasing_for_General_Vector_Graphics)