

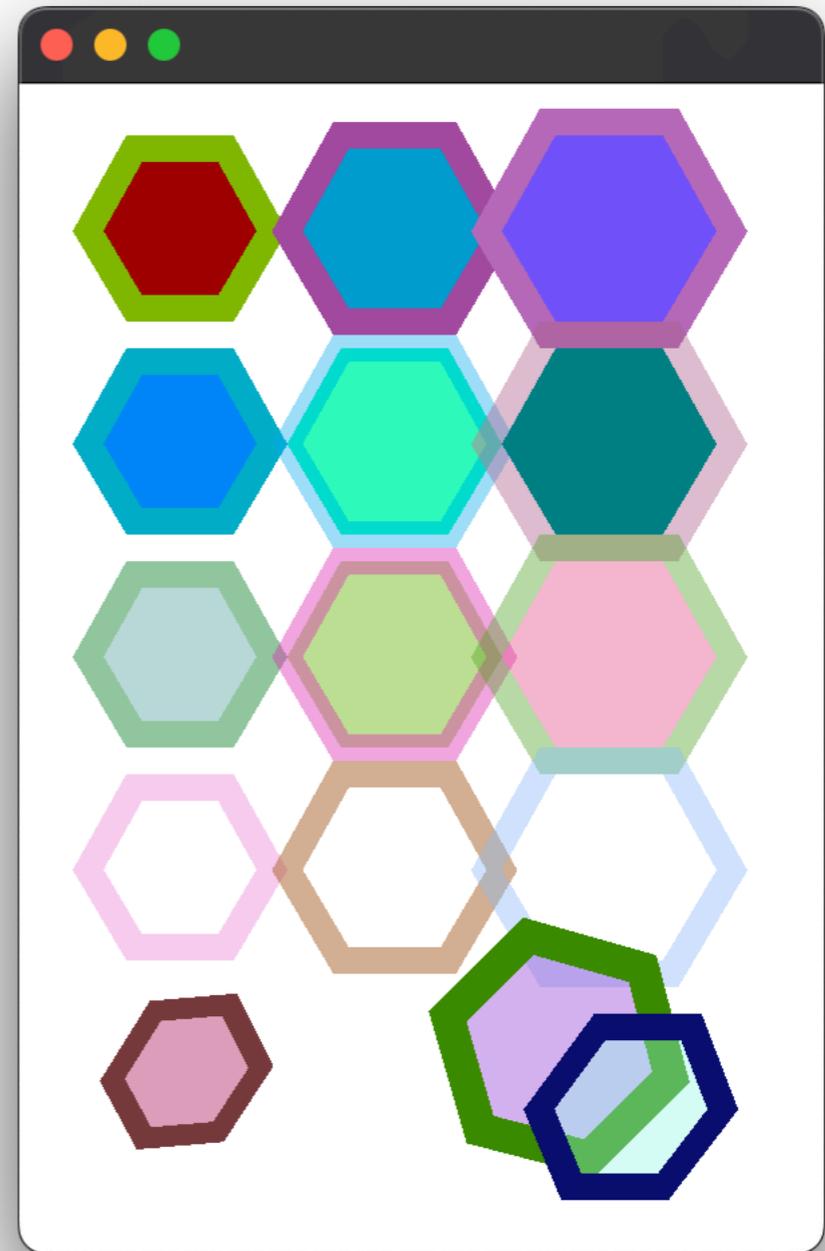
News on Bloc for *Pharo*

Pablo Tesone
Martín Dias

Thales
Pharo Consortium

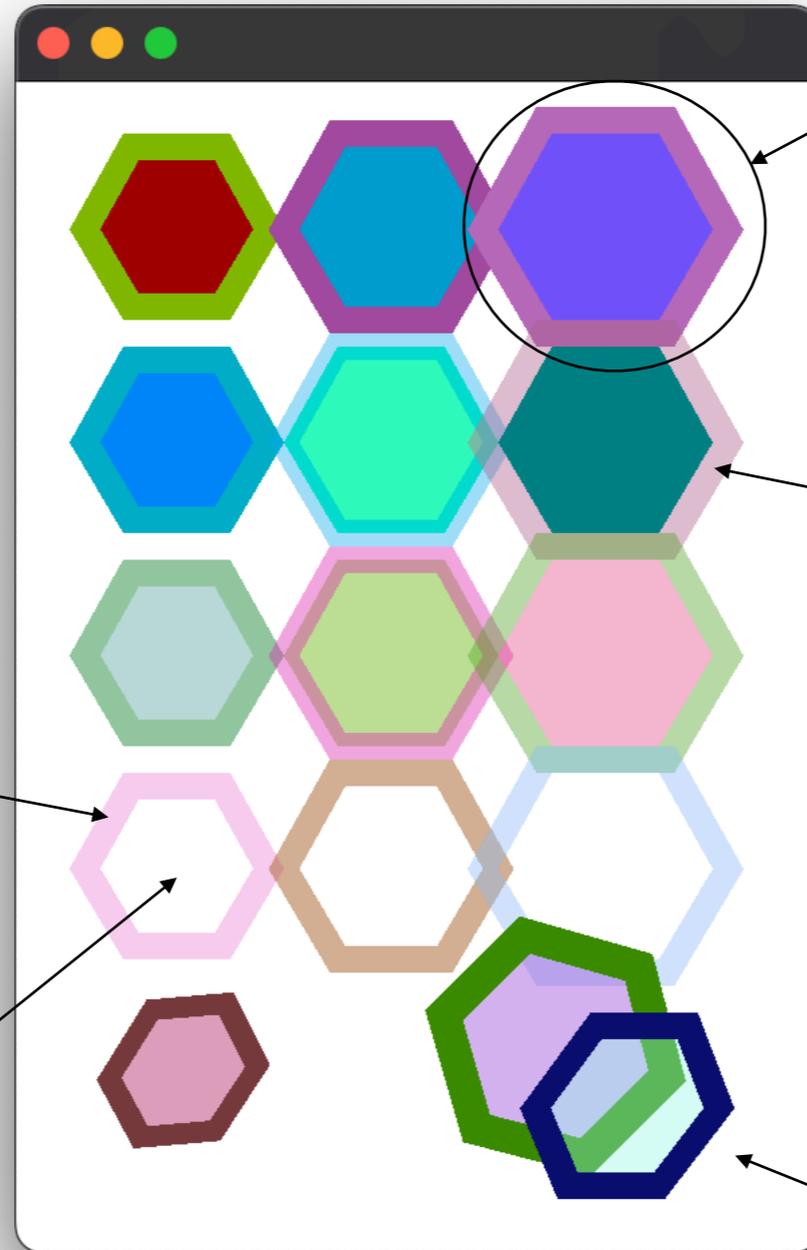
What is Bloc?

**a low-level UI
infrastructure &
framework for Pharo**



Element's visual properties

a BIElement



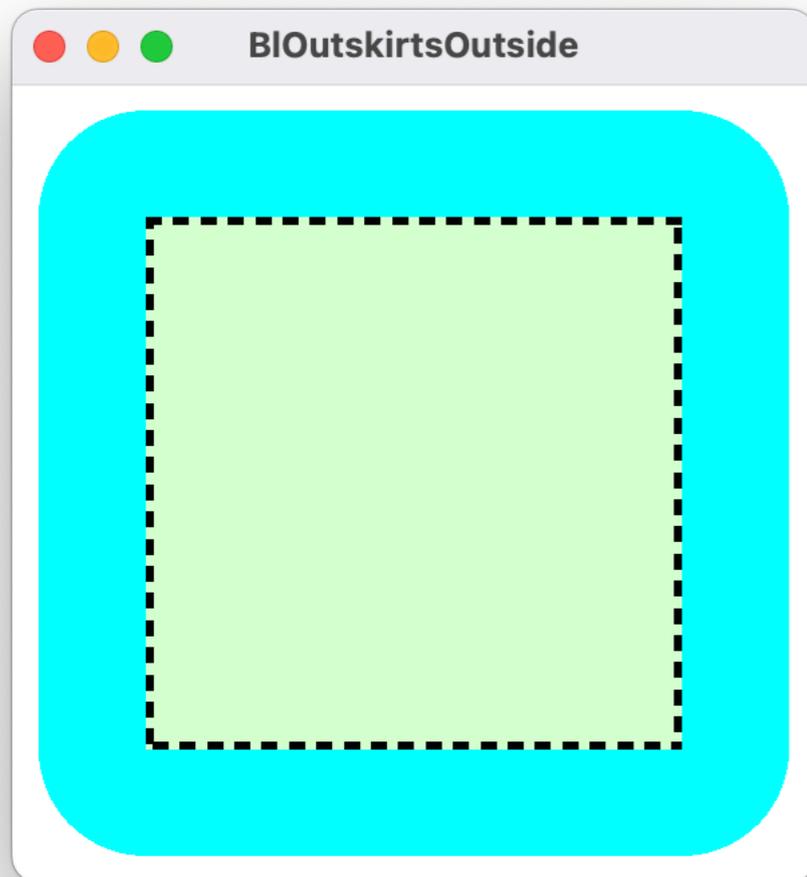
border

geometry
(polygon)

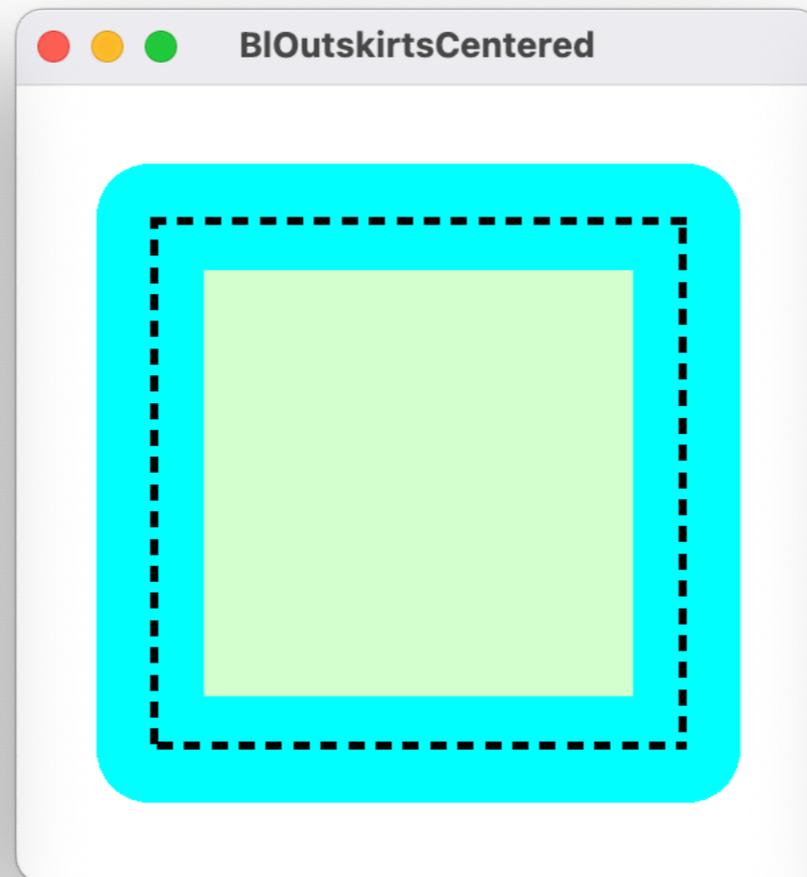
transformation
(a matrix with
skew and translation)

background

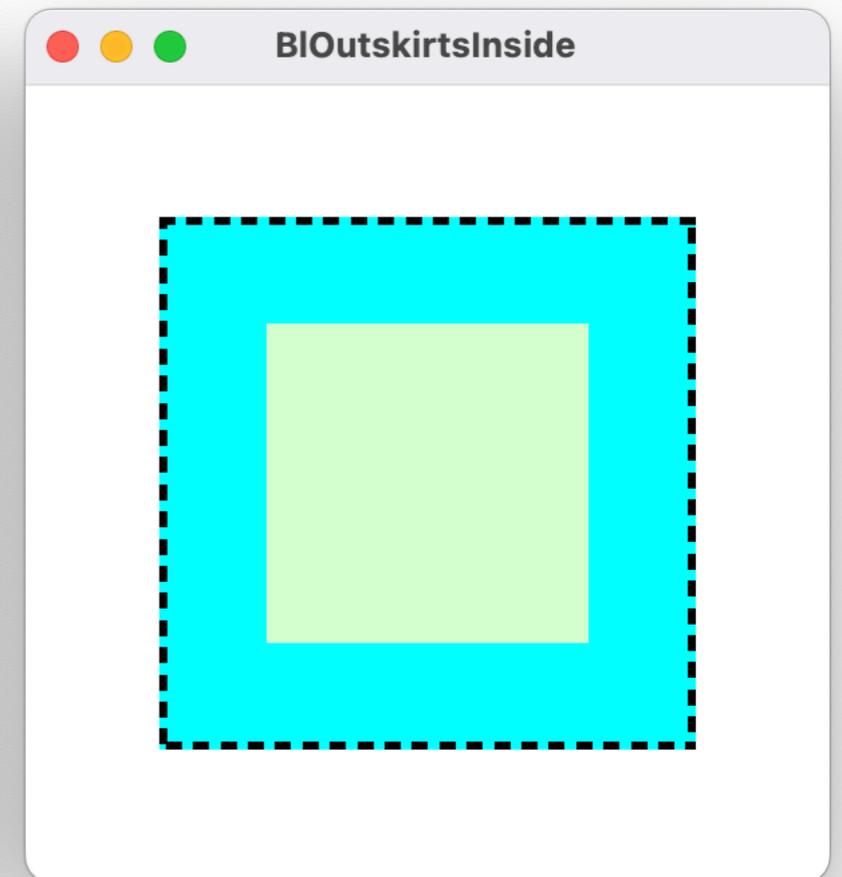
Element's outskirts



aBIElement
outskirts:
 BIOutskirts outside

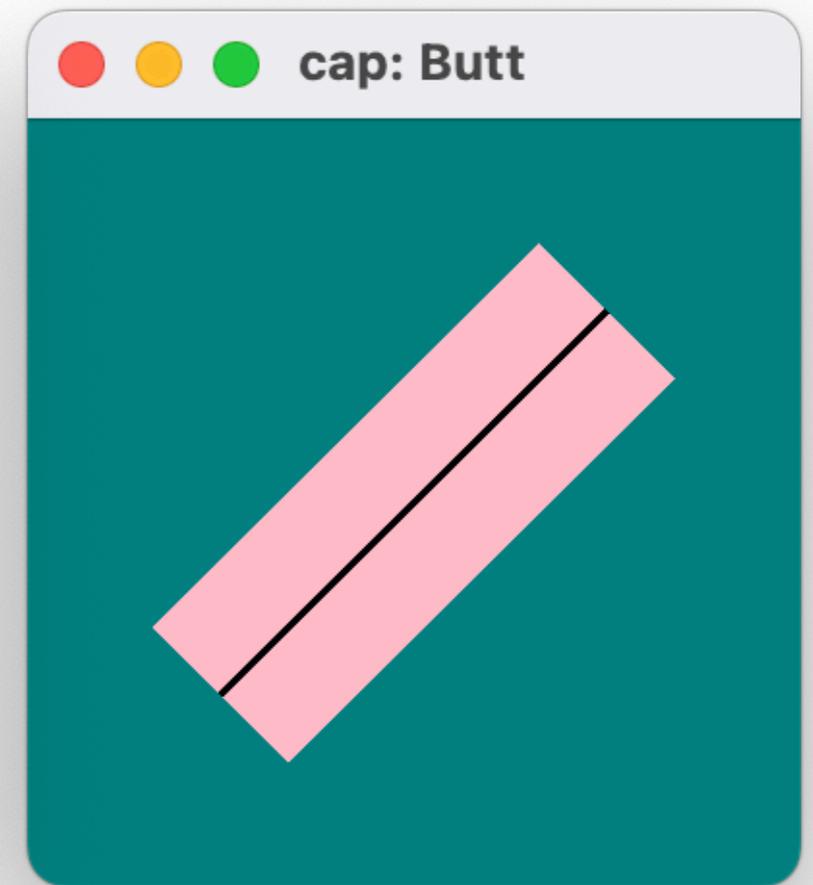
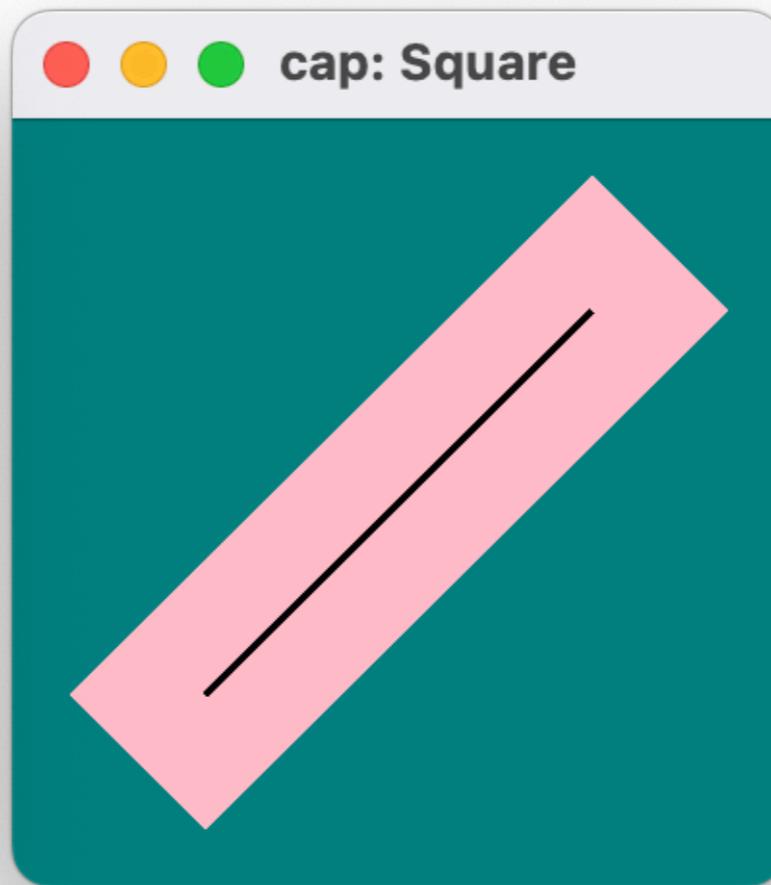
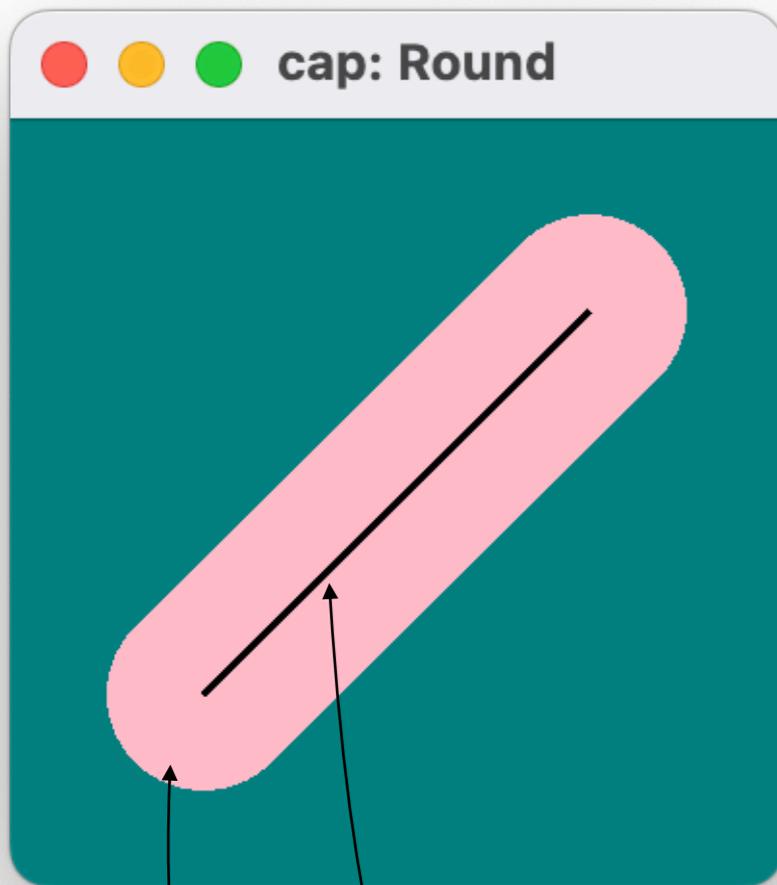


aBIElement
outskirts:
 BIOutskirts centered



aBIElement
outskirts:
 BIOutskirts inside

Border's cap



```
vertices := { 50@150. 150@50 }.
```

```
referenceLine := (BlPolylineGeometry vertices: vertices) asElement.
```

```
capLine := (BlPolylineGeometry vertices: vertices) asElement.
```

```
capLine border: (BlBorder builder
```

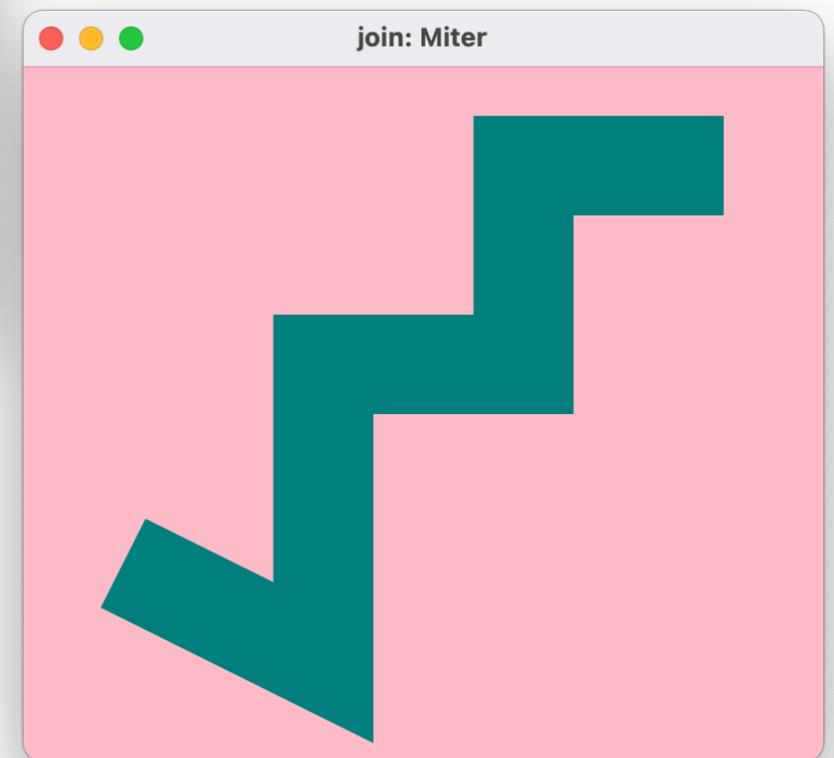
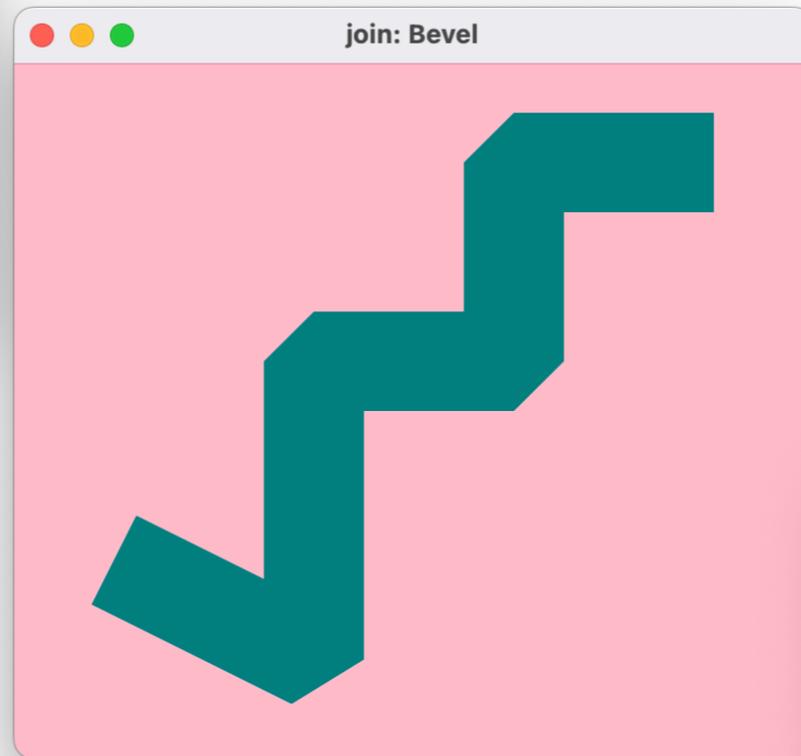
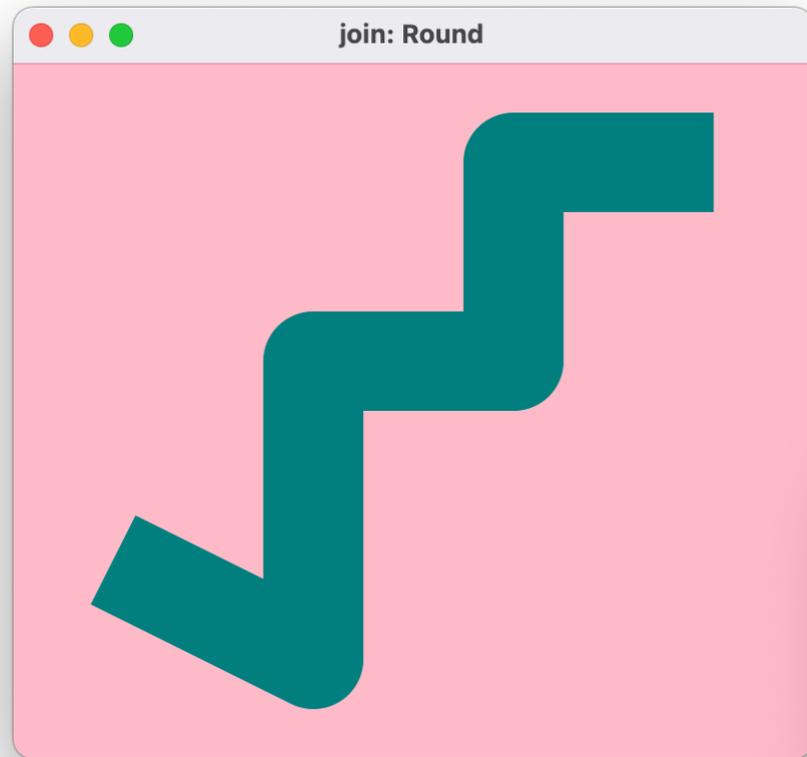
```
  paint: Color pink;
```

```
  width: 50;
```

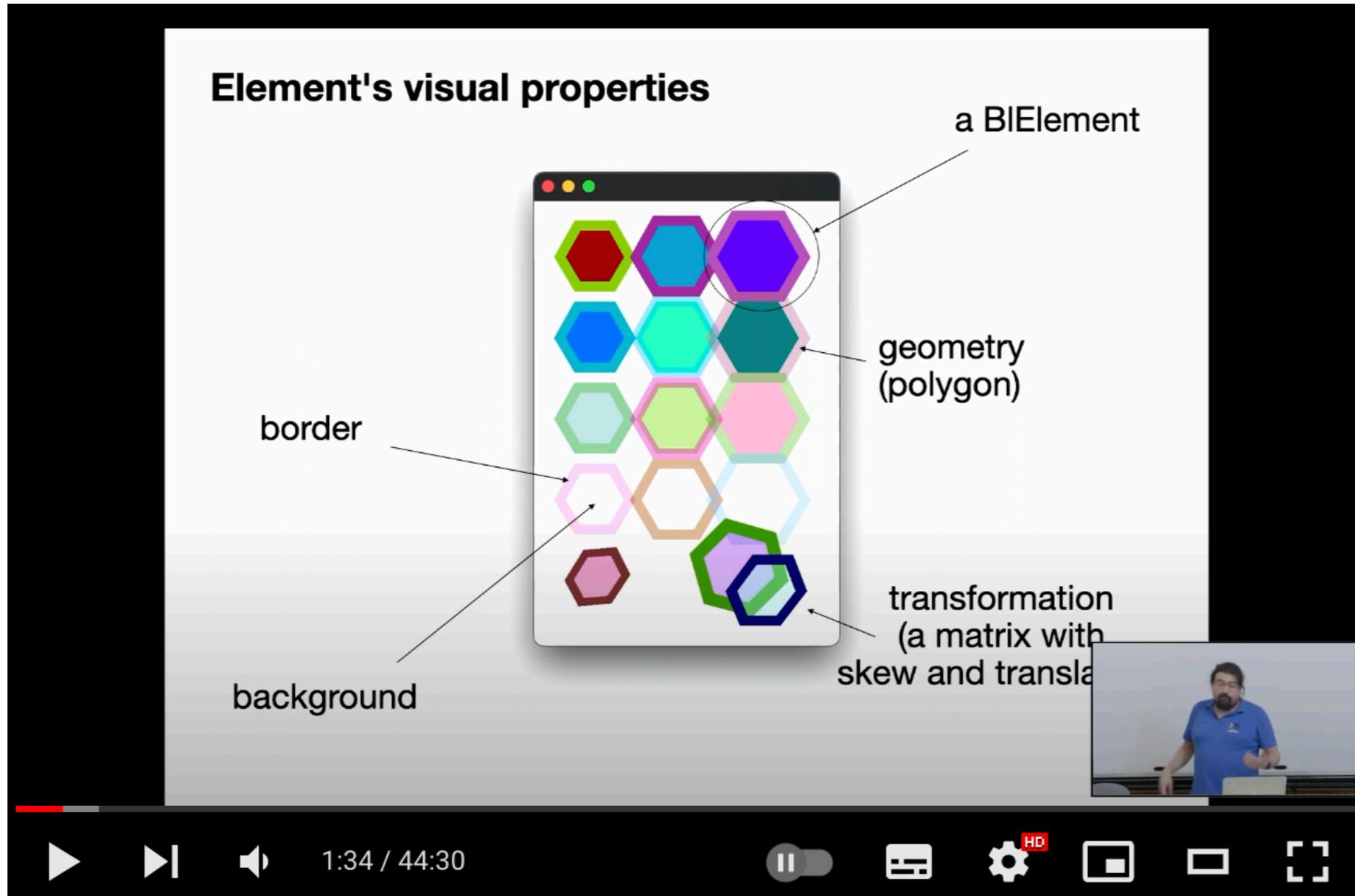
```
  lineCap: BlStrokeLineCap round;
```

```
  build)
```

Border's join



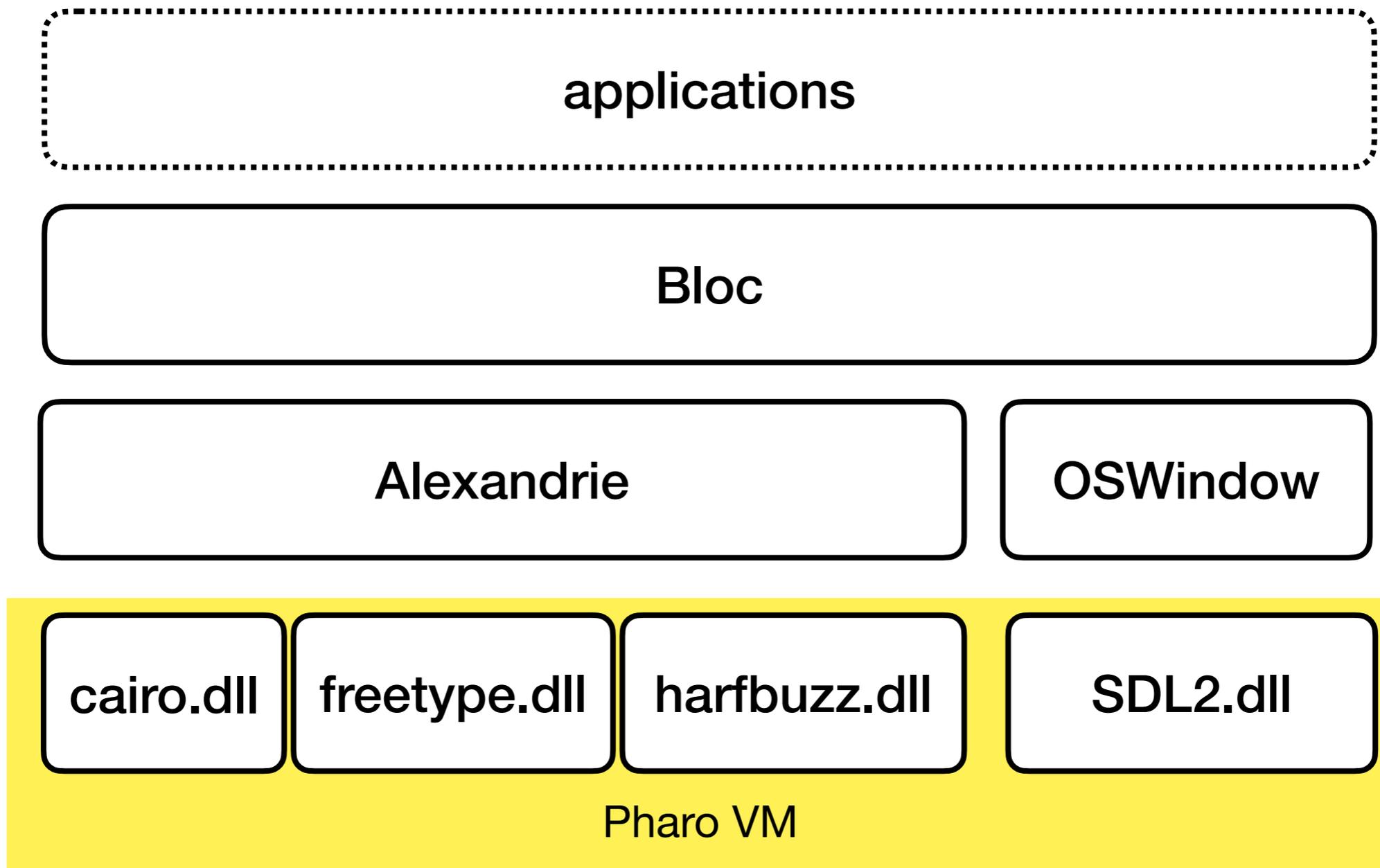
What is Bloc? For API tasting, please watch!



Bloc for Pharo - ESUG 2022



What is Bloc? Architecture



News!

10 things you may
not know of **Bloc**
in **2023**

Bloc and Alexandrie

New pre-release: **Bloc v2.0.0-alpha**

New commits: 874

Closed issues: 65

Closed PR: 43

Total tests: 1287

Alexandrie counts with
222 FFI call bindings.

~100 more than Athens
(we are ready to create an
Athens backend)

Bloc: <https://github.com/pharo-graphics/Bloc/releases/tag/v2.0.0-alpha>

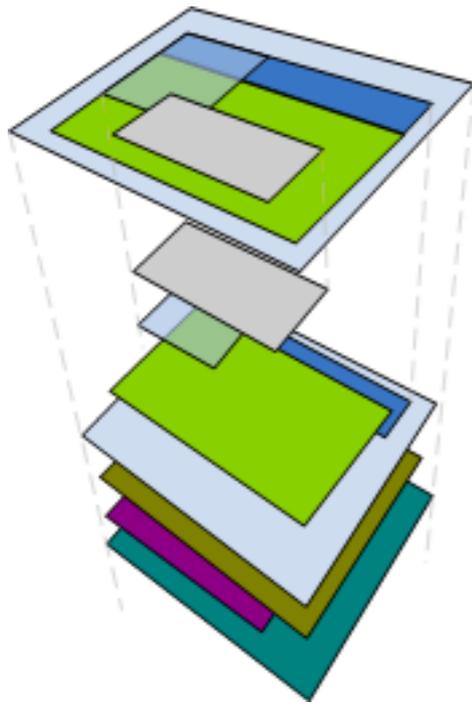
Alexandrie: <https://github.com/pharo-graphics/Alexandrie/releases/tag/v2.0.0-alpha>

New projects using Bloc

News
#2

Compositing Layers

Avoid rasterizing again elements that didn't change



Only a simple implementation for the moment:

- Any `BLElement` that answers true to `wantsSeparateCompositingLayer` is **cached** on a cairo surface (yeah, no GPU yet).
- Such element will be rasterized again only after a visual property changed (except a position change).
- It must be enabled **explicitly** by sending `beInSeparateCompositionLayer` to a `BLElement`.

Compositing Layers

An example



These 3 text elements received
`beInSeparateCompositionLayer`



The 3 layers are
rasterized only once
(already transformed)

Compositing Layers

Benchmarks show it can help

100 static circles of 200x200 pixels
that move down.

Three flavors:

- 👉 Dashed Border Circles
- 👉 Radial Gradient Circles
- 👉 Translucent Circles

None buffered (layers off) vs. All buffered (layers on)
fps = Frames Per Second
avg ms = Average ms per Frame

Bloc Alexandrie - Dashed Border Circles



layers:	off	on
fps	75	125
avg ms	13	8
ratio		1.7X

Bloc Alexandrie - Radial Gradient Circles w/ 2 stops



layers:	off	on
fps	14	120
avg ms	71	8
ratio		8.6X

Bloc Alexandrie - Translucent Opacity Circles



layers:	off	on
fps	28	97
avg ms	35	10
ratio		3.4X

Harfbuzz ffi bindings

Bloc-Alexandrie now has
FFI bindings to this C library

Why? 

It converts a Unicode string into properly formatted and positioned glyph output (based on font info that cairo+freetype ignore)

Harfbuzz ligatures example

Only Cairo 🙋

After coffee

A~~>B->>C

Harfbuzz + Cairo 🙋

After coffee

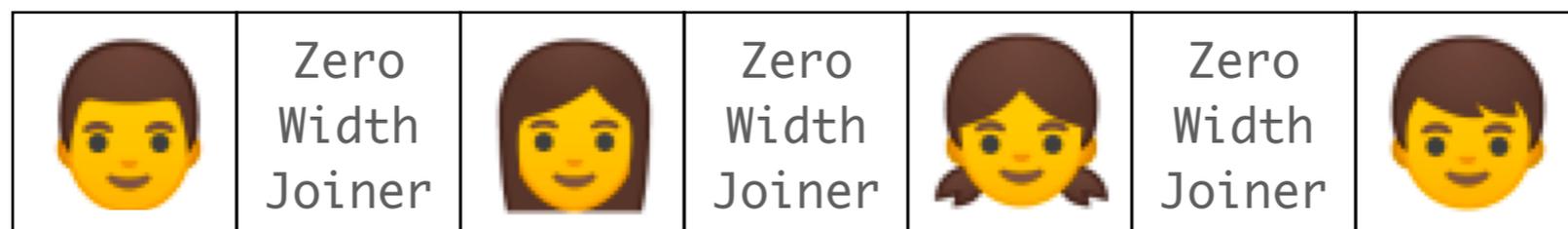
A~>B->>C

Source Sans Pro

Cascadia Code

Font files can provide a **ligature** glyph
for certain **sequences**

Harfbuzz emoji example

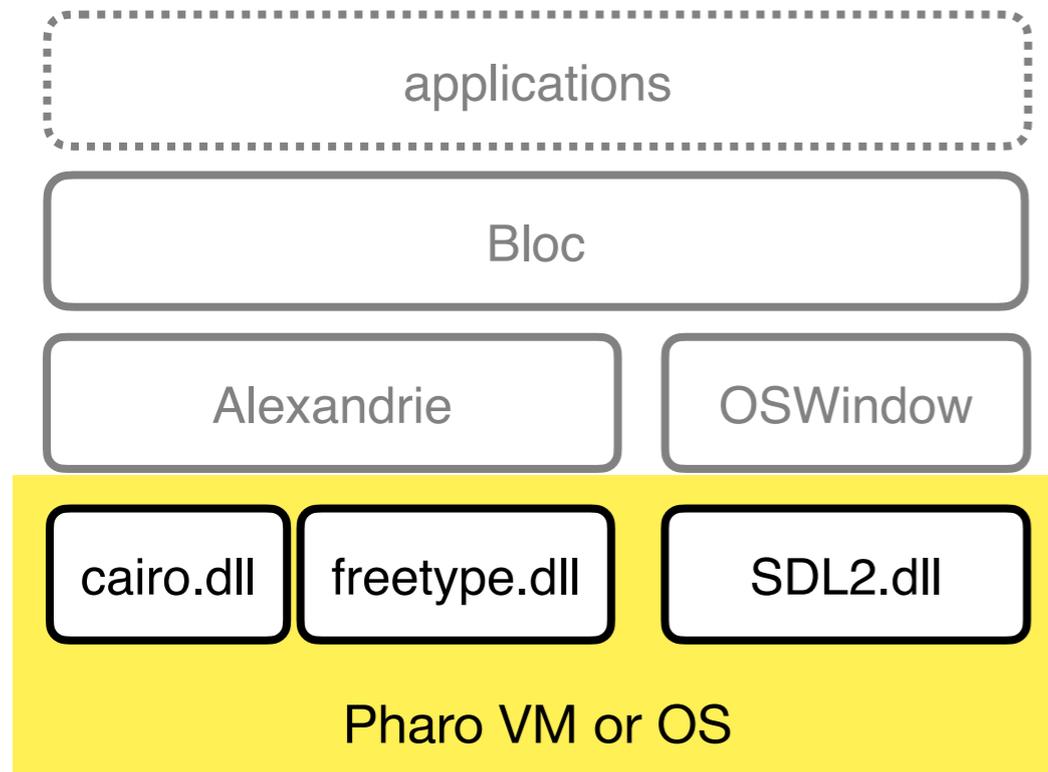


Zero
Width
Joiner

is a Unicode character to explicit "related with"

Dynamic libraries

Some work was required



- 👉 Win, Linux, Mac had different versions
- 👉 Harfbuzz was missing

Lib upgrade instructions were **too complex**
both for **users** and for defining **CI** jobs

Dynamic libraries

New VM version did it

Pharo VM v9.0.21 ~ Dec'22

cairo

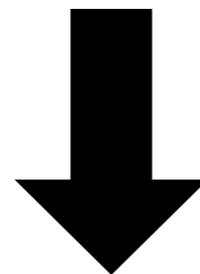
freetype

harfbuzz

SDL2

Now all platforms have:

- Cairo: 1.17.4
- Freetype: 2.12.1
- Harfbuzz: 5.3.1
- SDL: 2.24.1



Simplify Bloc
install instructions



Consistent CI results
on all platforms



Speed-up in
some cases

Dynamic libraries

MacOS had too old (slow) versions!

Frames per second
boosted between

1.8X and 4.4X

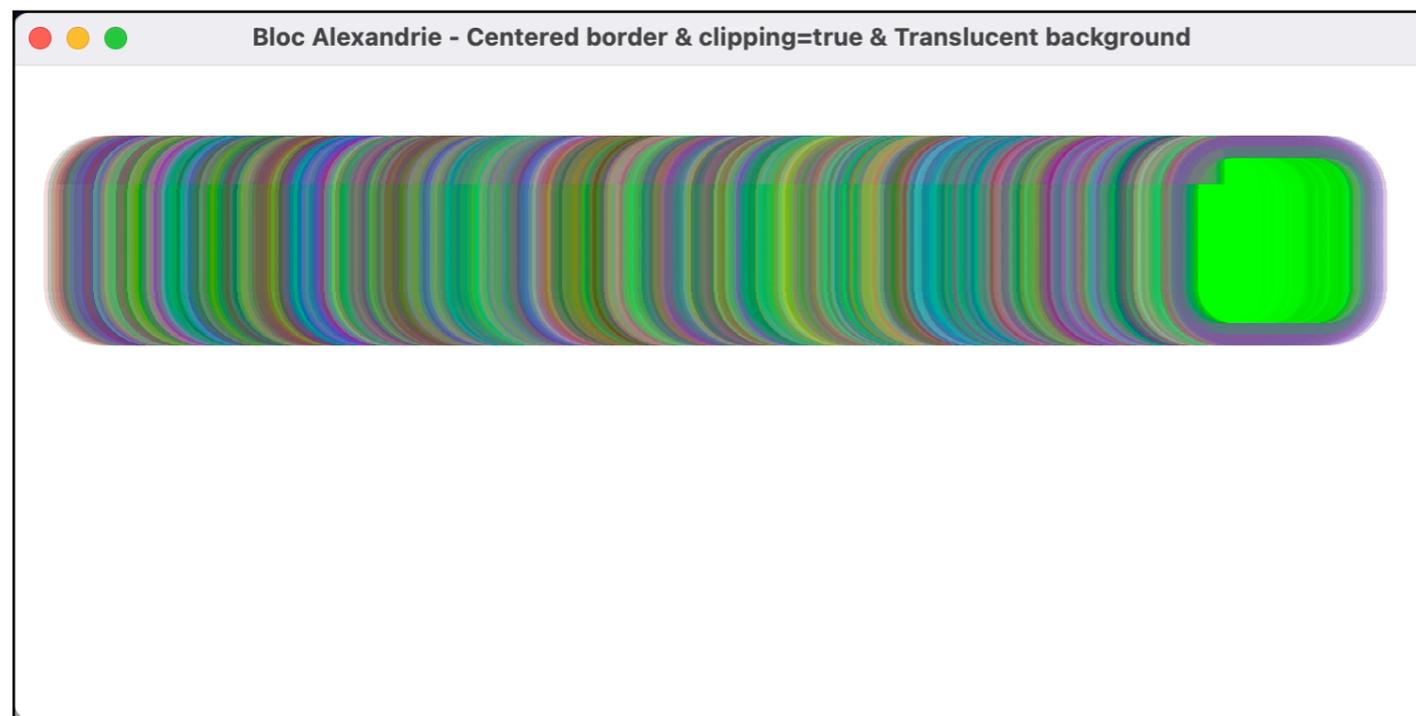


According to this benchmark
(18 variations).

300 rounded rectangles

200 x 200 pixels each

each one has a small child
(that is either clipped or not)



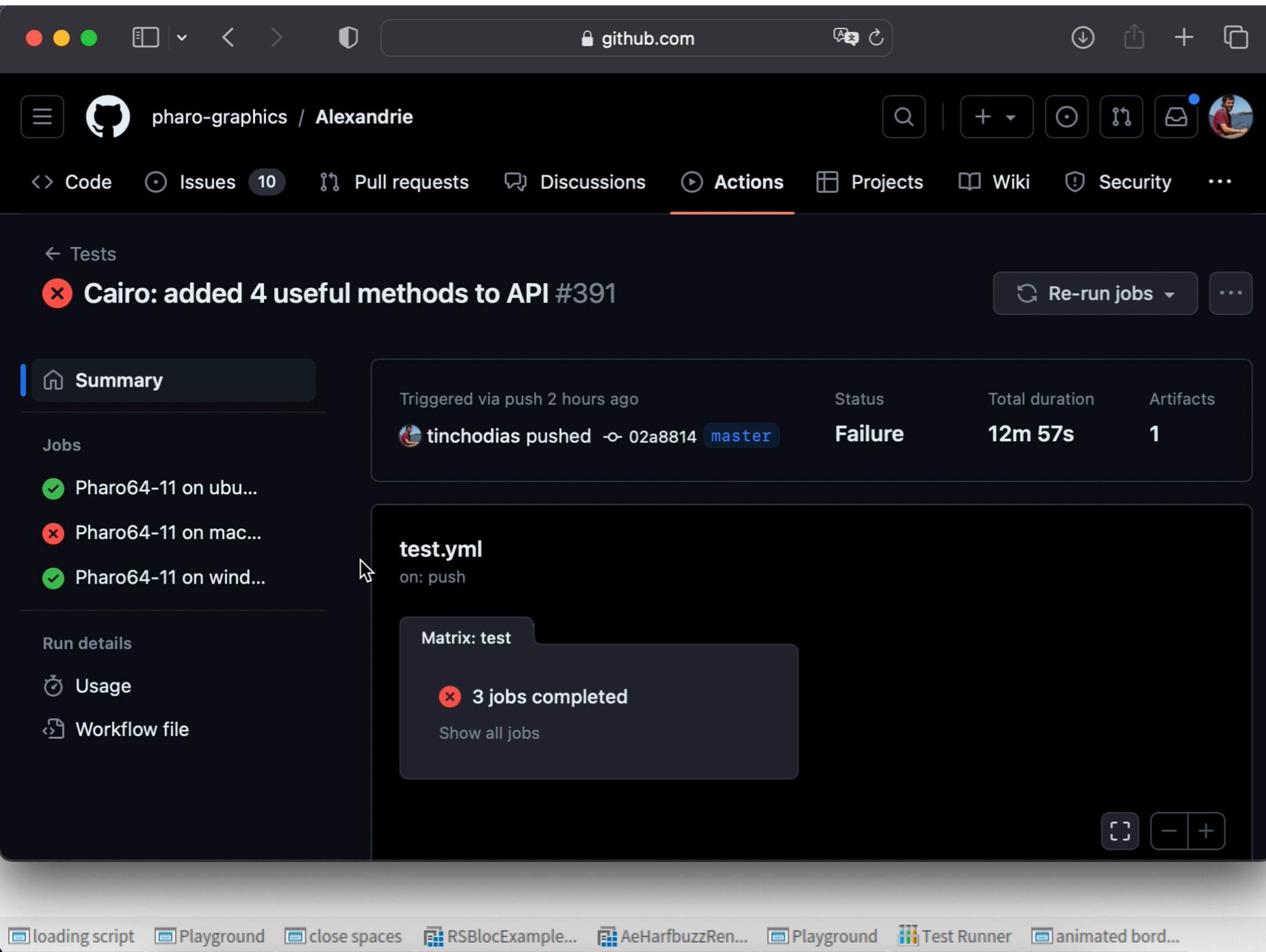
Outskirts	Background	Clip	Ratio
Centered	None	yes	2.3
Centered	Translucent	yes	3.6
Centered	Opaque	yes	2.2
Centered	None	no	2.5
Centered	Translucent	no	3.8
Centered	Opaque	no	2.3
Inside	None	yes	1.8
Inside	Translucent	yes	2.9
Inside	Opaque	yes	1.8
Inside	None	no	1.9
Inside	Translucent	no	3.1
Inside	Opaque	no	1.9
Outside	None	yes	4.2
Outside	Translucent	yes	4.4
Outside	Opaque	yes	2.9
Outside	None	no	4.3
Outside	Translucent	no	4.4
Outside	Opaque	no	3.1

CI exports regressions

Inspect by dropping a fuel file

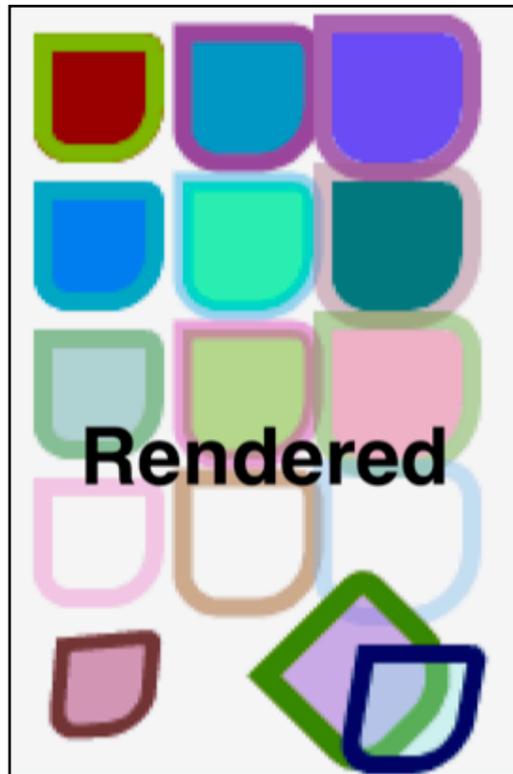
Use case:

- CI: Emoji regression test (randomly) fails
- CI: fuel-outputs the test's pixel comparison
- Download it from web
- Drop it into Pharo
- Discover somehow it is rendered in a single color!

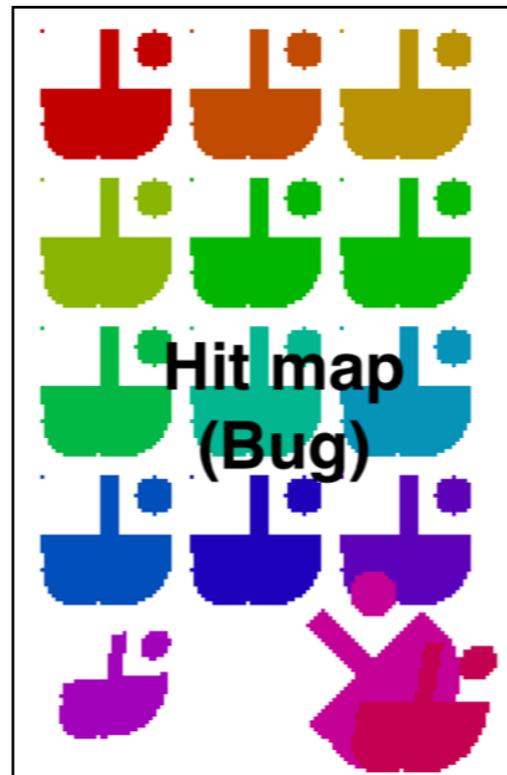


Rounded Rectangle

Bug fix: wrong hit testing



How you see
the test fixture



Mouse events were only
dispatched on the colored
regions



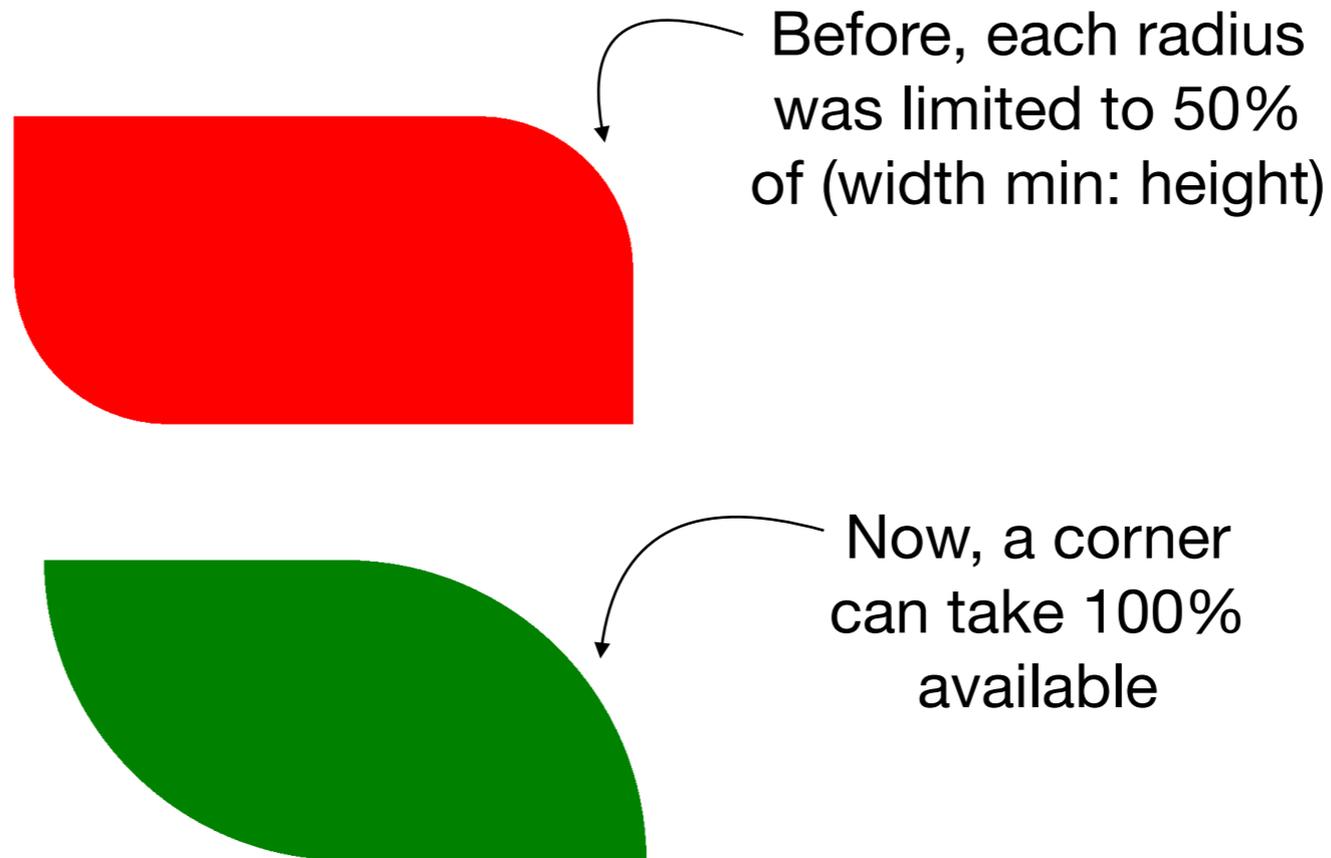
After the fix

Want to see more?

- Evaluate: BAHitTest inspectAll
- Browse: BIRoundedRectangleGeometry>>#containsPoint:alreadyInGeometryBoundsOf:

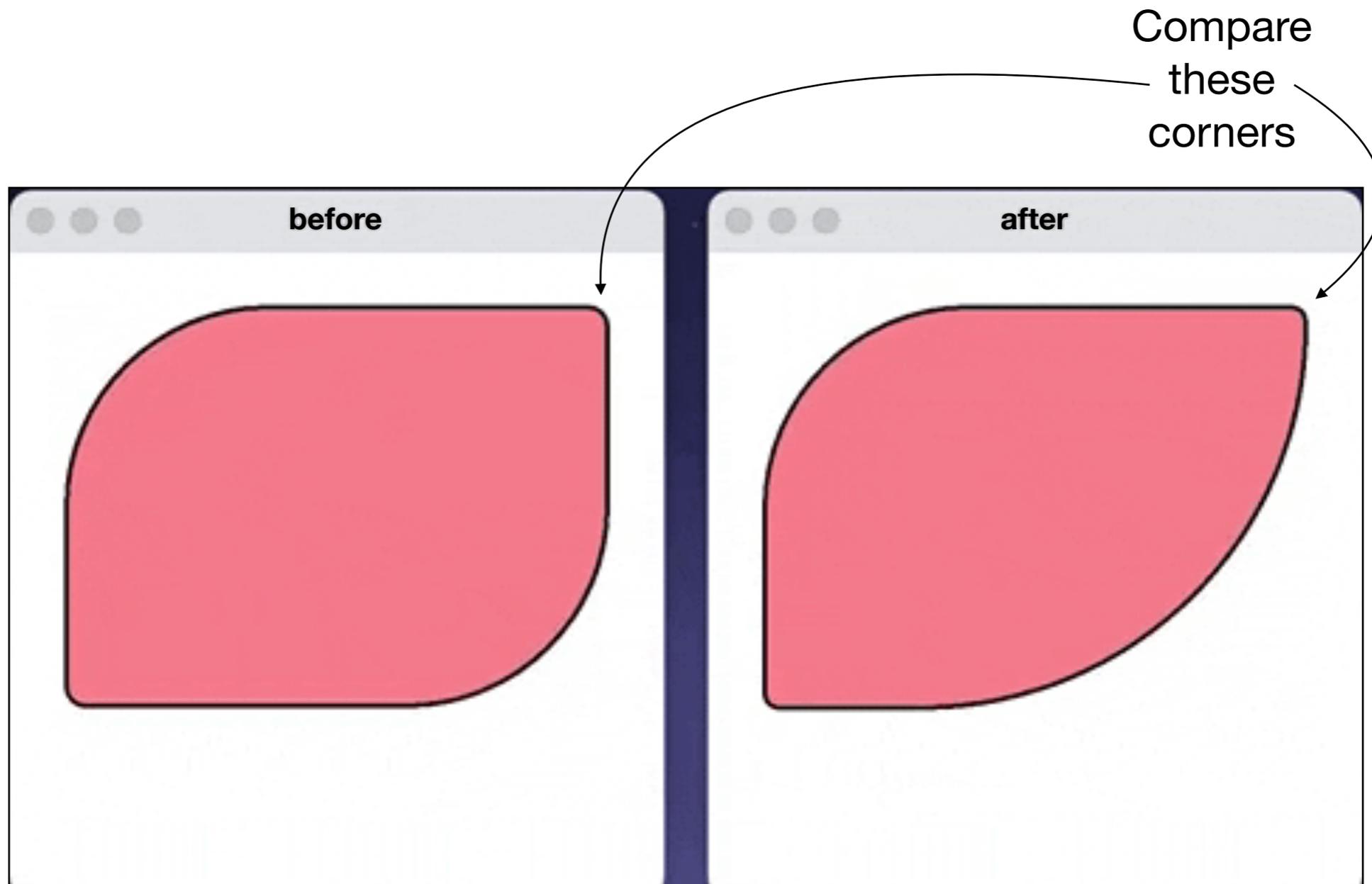
Rounded Rectangle

Enhancement: Give more flexibility on corner radii (as in CSS)



Rounded Rectangle

If two corners overlap, all corners are reduced proportionally

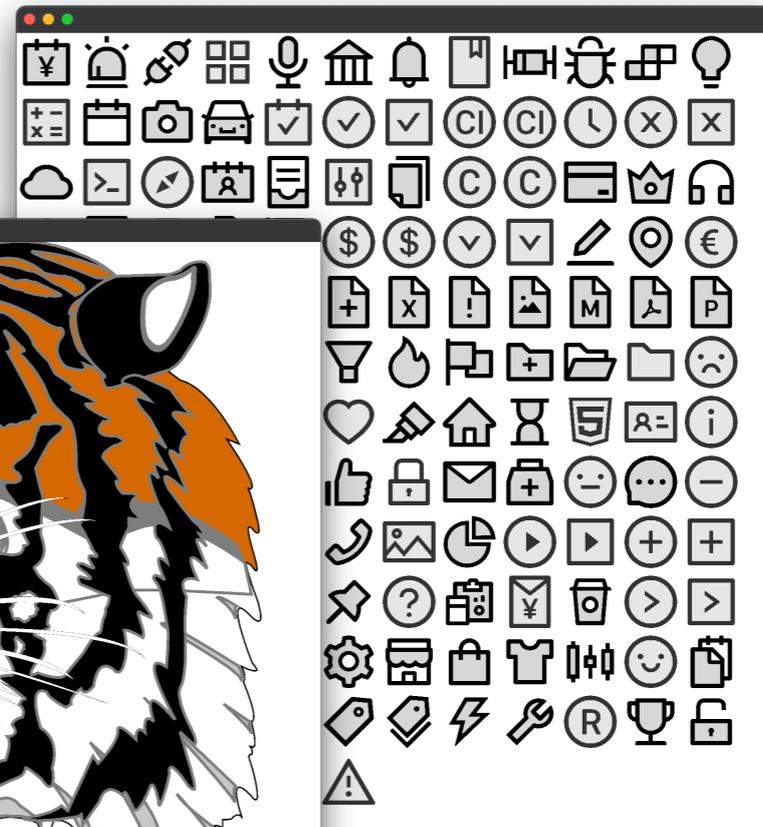
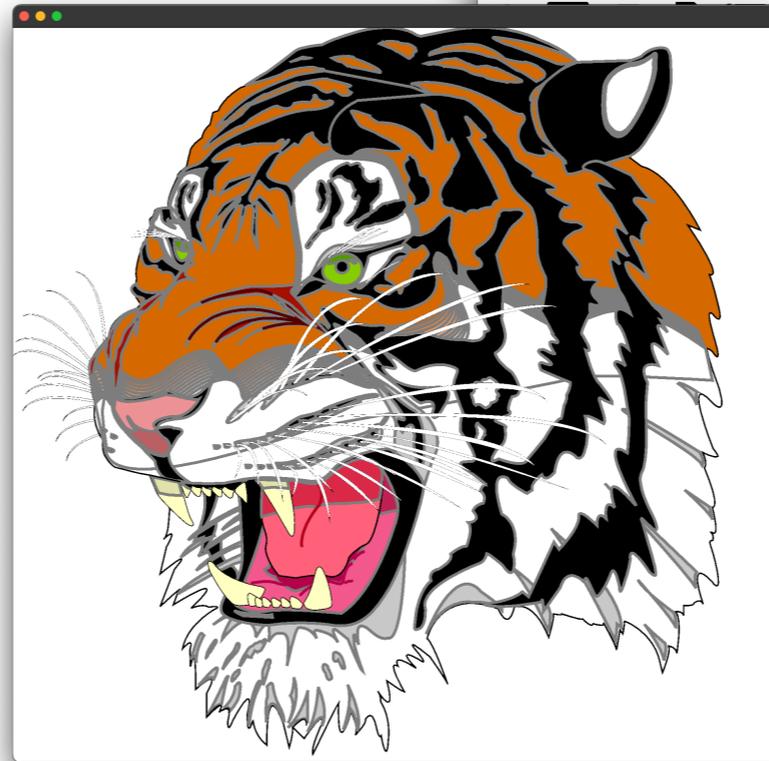


Import SVG

Repair and improve code from BlocPac and Athens-SVG

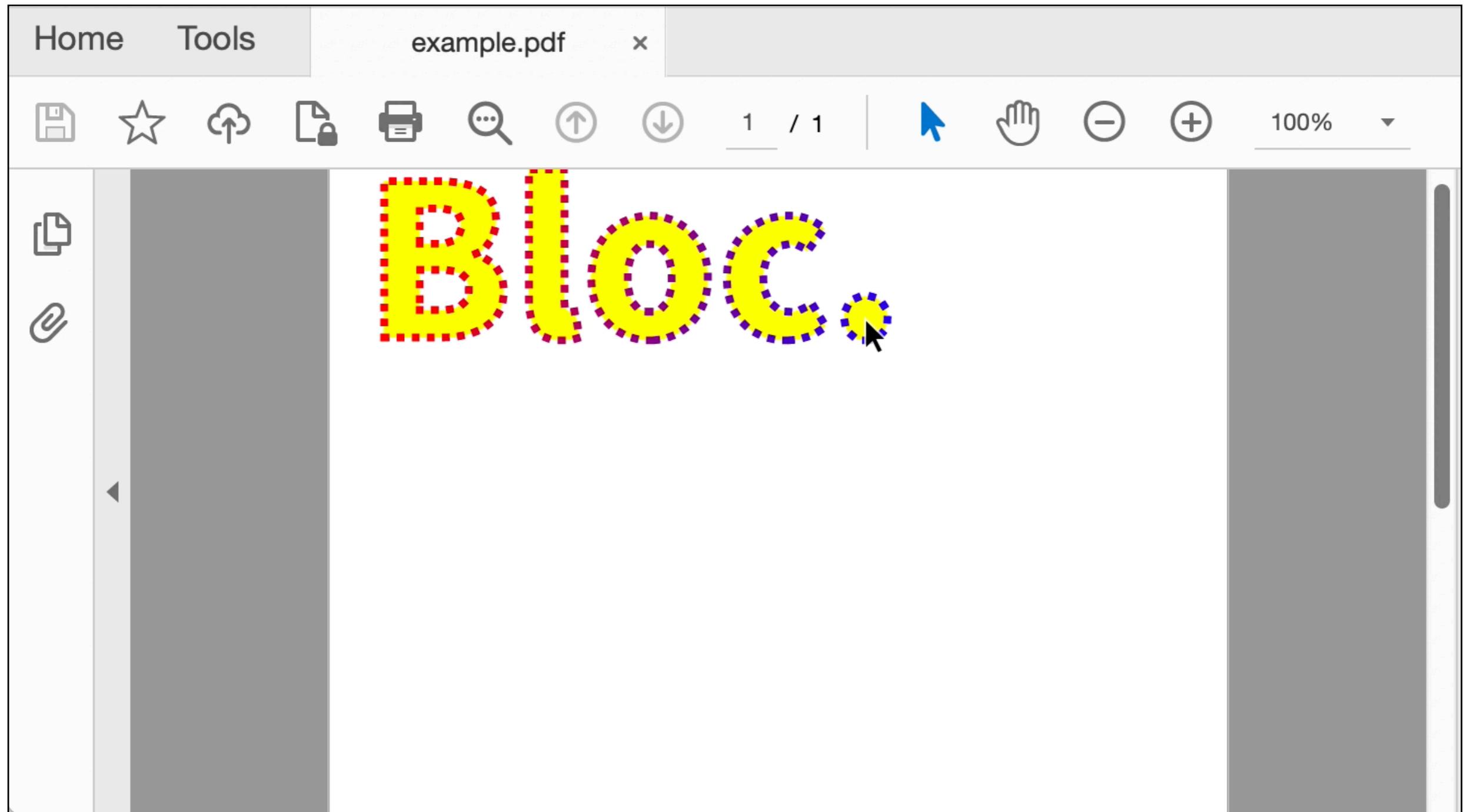
```
<?xml version="1.0" standalone="no"?>  
<svg width="1024" height="1024">  
  <path d="M512 64C264.6...z"/>  
  <path d="M232.3 15...z"/>  
</svg>
```

Bloc
importer



Export SVG and PDF

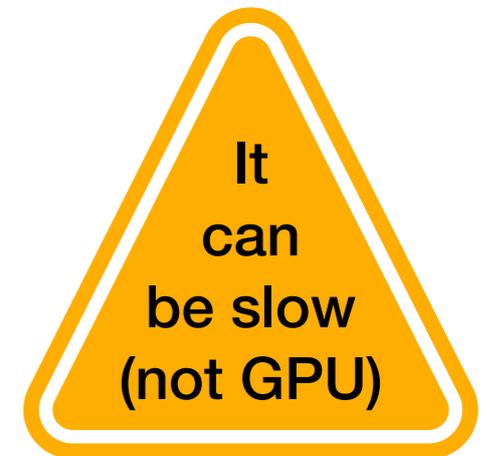
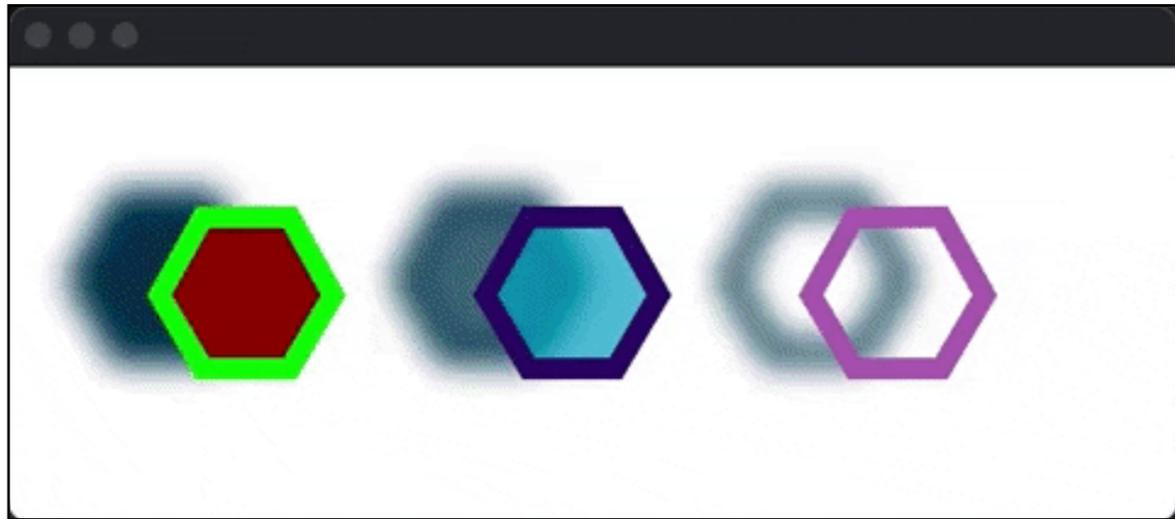
As vectors (not just pixels) via cairo API



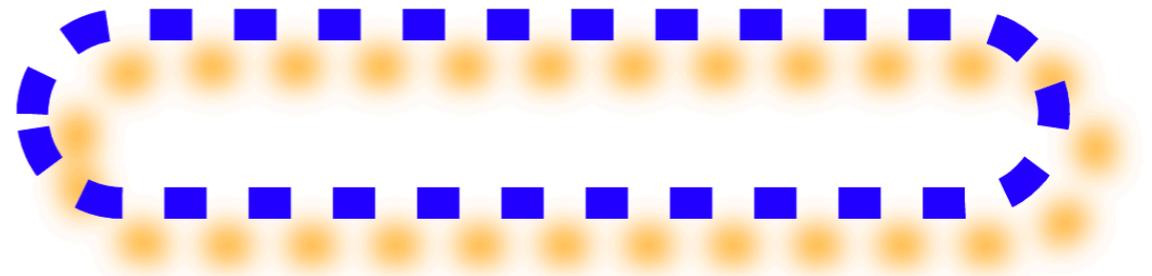
For more information, look at [AeCairoPDFSurface](#) and [BIPdfExporter](#)

Gaussian Shadow Effect

Cairo doesn't provide this effect, we implemented the algorithm.

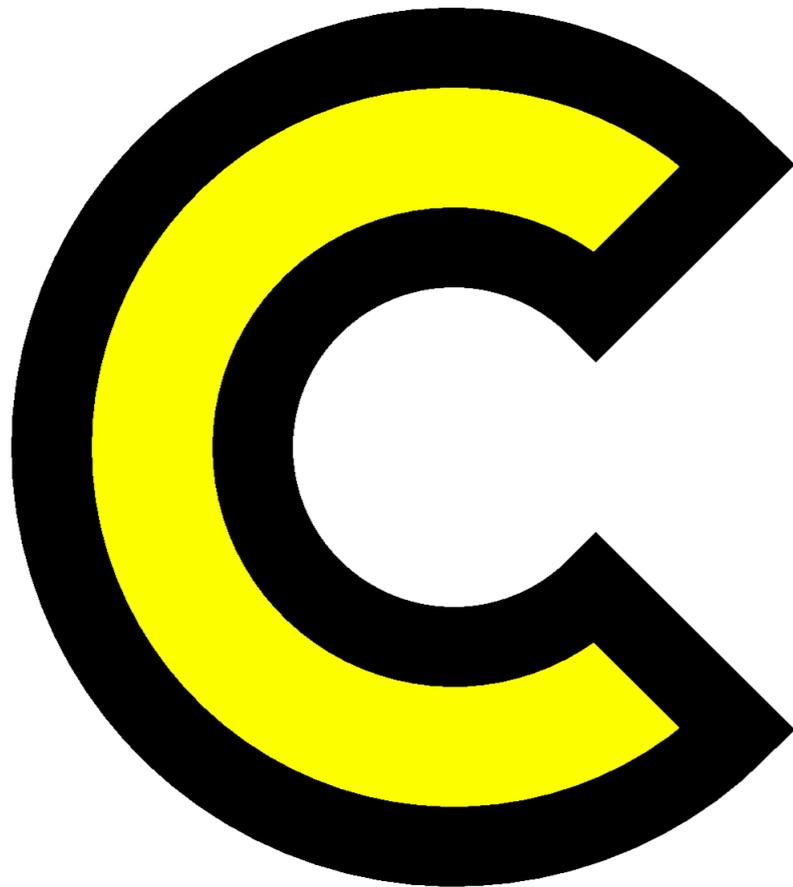


```
BlElement new
  geometry: (BlRoundedRectangleGeometry cornerRadius: 50);
  size: 500@100;
  border: (BlBorder builder
    paint: Color blue;
    width: 15;
    dashArray: #(20);
    build);
  effect: (BlGaussianShadowEffect
    color: Color orange
    width: 20
    offset: 20 asPoint);
  yourself
```



Border with gradients

Feature request in Bloc-Alexandrie



BEFORE:

an element's border could
only have a single color



NOW:

Borders can have a linear
or radial gradient as paint

**STAY
TUNED!**

Mailing-list: lse-openbloc@inria.fr
(subscribe at: <http://sympa.inria.fr>)

Discord: Pharo server / #bloc channel

Bloc: <https://github.com/pharo-graphics/Bloc>
Alexandrie: <https://github.com/pharo-graphics/Alexandrie>

News in Bloc for Pharo

Pablo Tesone
Martín Dias

ESUG 2023 - Lyon