

Visualizing Objects in Pharo

Presentation BA thesis
Software Composition Group

Eve Mendoza Quiros
Supervisor: Claudio Corrodi

25th April 2017

Object inspector in

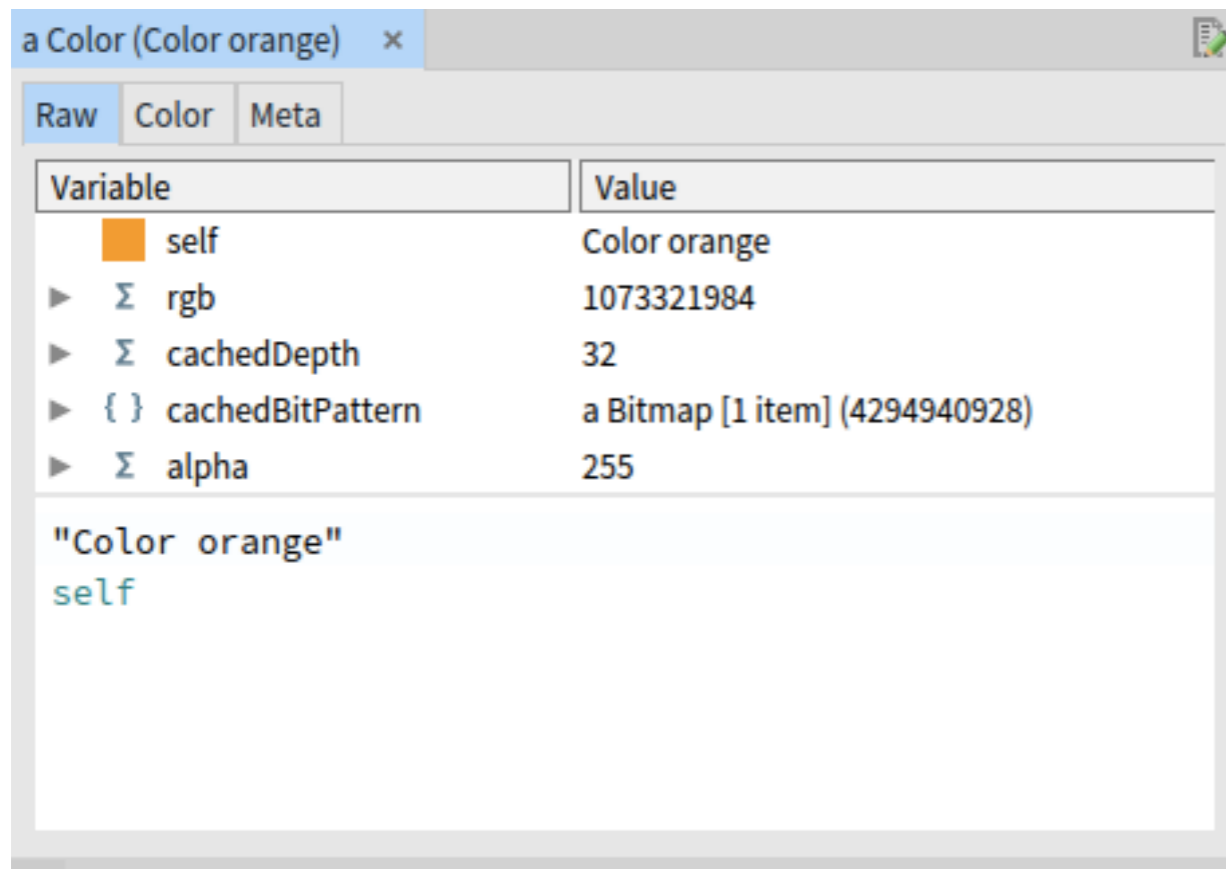


Relations to other objects shown as lists or trees

The screenshot shows the Pharo Object Inspector window for a `LinkedList` object containing four items: orange, red, blue, and green. The window is titled "a LinkedList [4 items] (Color orange Color red Color blue Color green)". It has three tabs: "Items", "Raw", and "Meta". The "Items" tab is selected, displaying a tree view of the object's internal structure. The tree shows the `self` object pointing to the `LinkedList` object, which has a `firstLink` attribute pointing to a `ValueLink` object for the color orange. This `ValueLink` object has a `self` attribute pointing to itself and a `nextLink` attribute pointing to another `ValueLink` object for the color red. This pattern repeats for the blue and green links, with the final `nextLink` attribute pointing to `nil`. The `value` attribute of each `ValueLink` object is shown as a colored square (green, blue, red, orange) and labeled with the corresponding color name.

Variable	Value
{ } self	a LinkedList [4 items] (Color orange Color red Color blue Color green)
▼ firstLink	ValueLink(Color orange)
self	ValueLink(Color orange)
▼ nextLink	ValueLink(Color red)
self	ValueLink(Color red)
▼ nextLink	ValueLink(Color blue)
self	ValueLink(Color blue)
▼ nextLink	ValueLink(Color green)
self	ValueLink(Color green)
▼ nextLink	nil
self	nil
▶ value	Color green
▶ value	Color blue
▶ value	Color red
▶ value	Color orange

Focus on single object



The screenshot shows a debugger window titled "a Color (Color orange)". It has three tabs: "Raw", "Color", and "Meta". The "Raw" tab is selected. Below the tabs is a table with two columns: "Variable" and "Value".

Variable	Value
self	Color orange
▶ Σ rgb	1073321984
▶ Σ cachedDepth	32
▶ { } cachedBitPattern	a Bitmap [1 item] (4294940928)
▶ Σ alpha	255

Below the table, the text "Color orange" is displayed in a light blue font, followed by "self" in a dark blue font.

Focus on single object

a Color (Color orange) x

Raw Color Meta

Variable	Value
self	Color orange
▶ Σ rgb	1073321984
▶ Σ cachedDepth	32
▶ { } cachedBitPattern	a Bitmap [1 item] (4294940928)
▶ Σ alpha	255

"Color orange"
self

a RTView x

View Raw Elements Meta

Color orange
alpha : 255
cachedBitPattern : a Bitmap of length 1
cachedDepth : 32
rgb : 1073321984

```
graph TD; A["Color orange  
alpha : 255  
cachedBitPattern : a Bitmap of length 1  
cachedDepth : 32  
rgb : 1073321984"] -- alpha --> B["255"]; A -- cachedDepth --> C["32"];
```

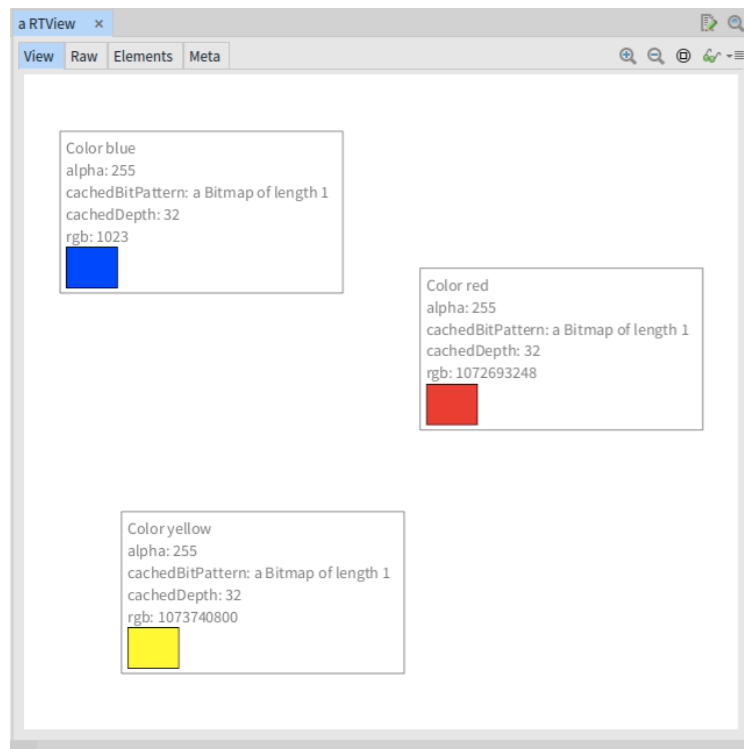
Demonstration

in

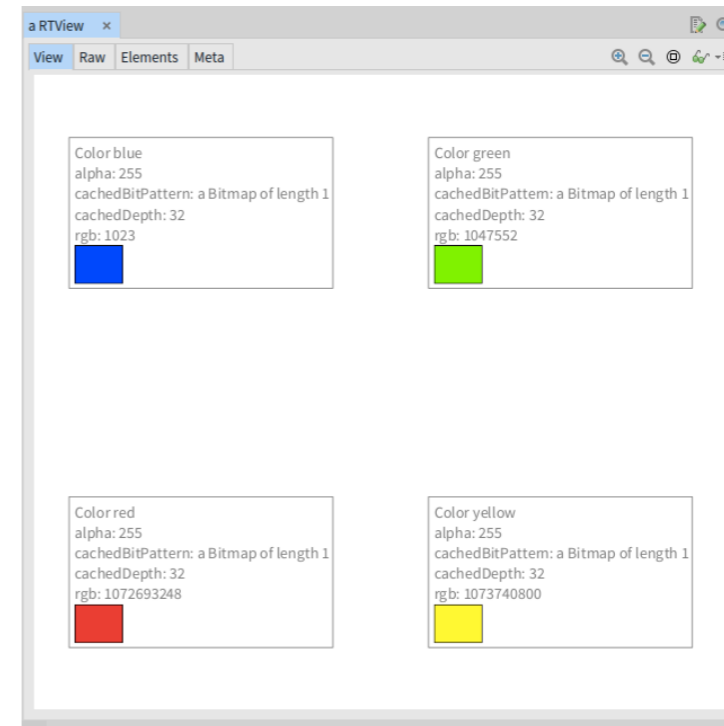


Scope of BA thesis

Persist Subgraphs



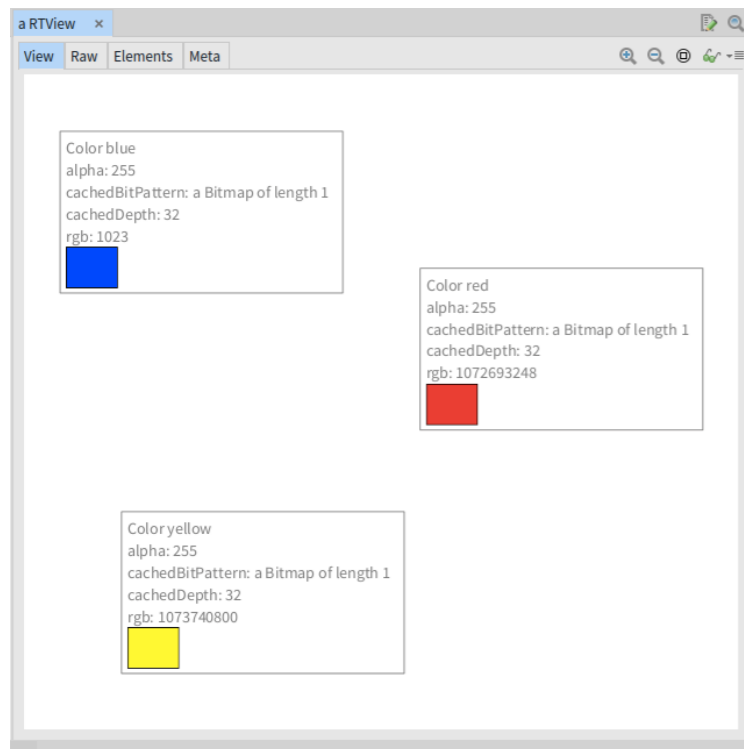
adding Color green



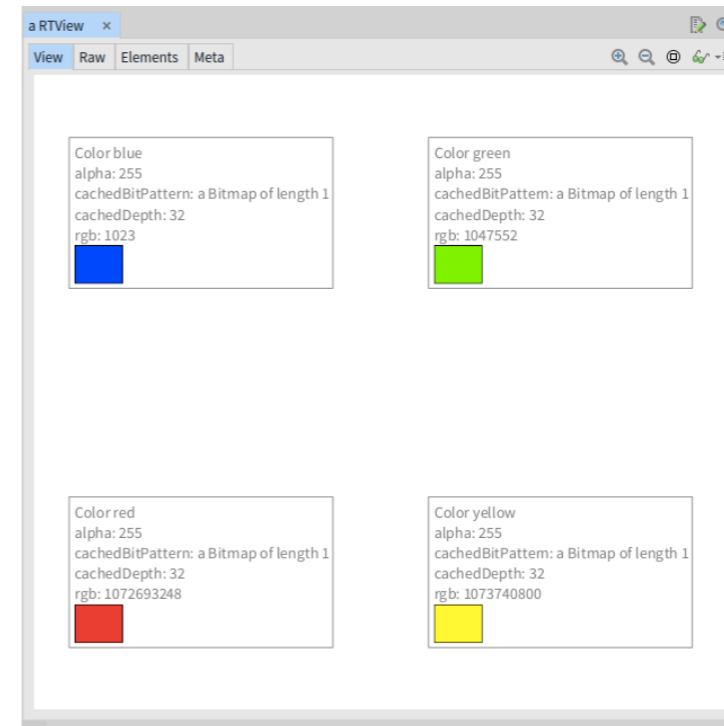
Currently

Scope of BA thesis

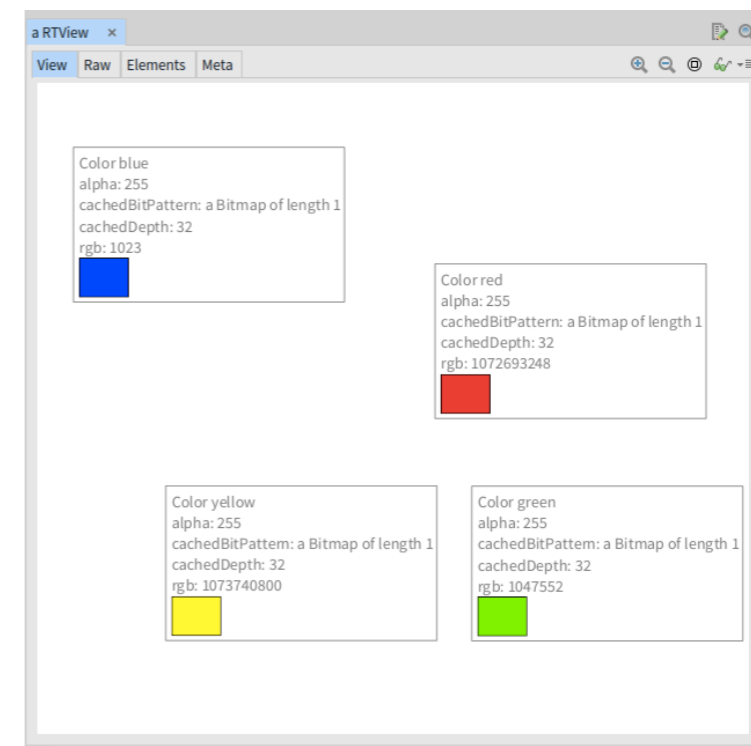
Persist Subgraphs



adding Color green



Currently



Goal

Custom node shapes for different objects

a CalendarMorph(213632512) ×

Raw Morph Meta

Variable	Value
self	a CalendarMorph(213632512)
bounds	(0@0) corner: (200@160)
owner	nil
submorphs	an Array [0 items] ()
fullBounds	nil
color	Color blue
extension	nil
borderWidth	1
borderColor	Color black

a CalendarMorph(213632512) ×

Raw Morph Meta

< April 2017 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Today: April 24 2017

Custom node shapes for different objects

Inspector window for `a CalendarMorph(213632512)`. The `Raw` tab is selected. The table below shows the object's variables and their values.

Variable	Value
<code>self</code>	<code>a CalendarMorph(213632512)</code>
<code>bounds</code>	<code>(0@0) corner: (200@160)</code>
<code>owner</code>	<code>nil</code>
<code>submorphs</code>	<code>an Array [0 items] ()</code>
<code>fullBounds</code>	<code>nil</code>
<code>color</code>	<code>Color blue</code>
<code>extension</code>	<code>nil</code>
<code>borderWidth</code>	<code>1</code>
<code>borderColor</code>	<code>Color black</code>

Goal

`a RTView` window showing a custom node shape for a `CalendarMorph` object. The node is a rectangle with a black border and blue fill, containing the following text:

```
a CalendarMorph(805767680)
announcer: nil
borderColor: Color black
borderWidth: 1
bounds: (0@0) corner: (200@160)
color: Color blue
date: 24 April 2017
days: an OrderedCollection(CalendarDayMorph (26 March...
extension: nil
fullBounds: nil
owner: nil
submorphs: #()
touchPoints: a Dictionary()
```

Below the text is a calendar view for April 2017, with the date 24 circled in red. The calendar shows the following data:

< April 2017 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Today: April 24 2017

Inspector window for `a CalendarMorph(213632512)`. The `Morph` tab is selected, showing a preview of the calendar view. The calendar shows the following data:

< April 2017 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Today: April 24 2017

Outlook further work

Interactive visualisation

an OrderedCollection((Color r: 0.79 g: 0.035 b: 0.03 alpha: 1.0) (Color r: 0.485 g: 0.16 b: 0.387 alpha: 1.0) (Color r: 0.091 g: 0.454 b: 0.465 alpha: 1.0))

array: {(Color r: 0.79 g: 0.035 b: 0.03 alpha: 1.0). (...

firstIndex: 1

lastIndex: 3

Elements

(Color r: 0.79 g: 0.035 b: 0.03 alpha: 1.0)	(Color r: 0.485 g: 0.16 b: 0.387 alpha: 1.0)	(Color r: 0.091 g: 0.454 b: 0.465 alpha: 1.0)
---	--	---

Outlook further work

Interactive visualisation



Setup



- **Pharo**

Pharo is a pure object-oriented programming language and a powerful environment, focused on simplicity and immediate feedback.

- **Moose image 6.1**

Moose enables agile visualization. The Roassal engine enables rapid visualization creation for various types of data.



- **Roassal**

Roassal is a visualization engine, written in the Pharo and VisualWorks programming languages.

- **Trachel**

Trachel is a low-level API to draw primitive graphical elements.

Questions?