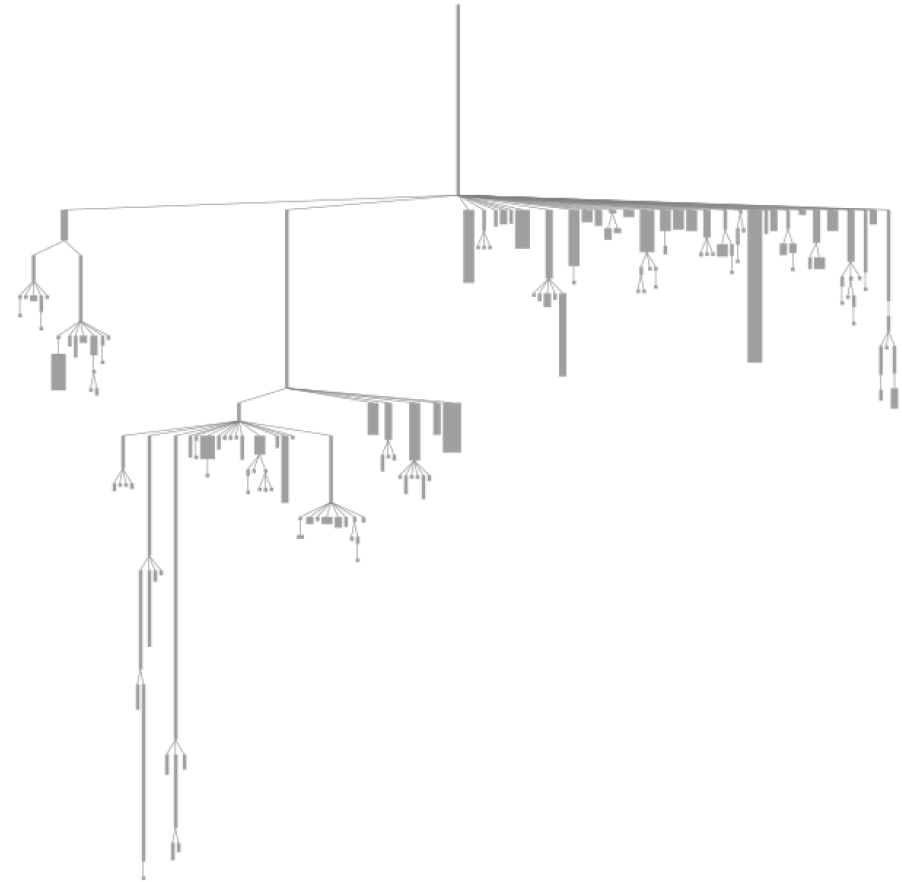


Roassal



Roassal

- Visualisation de données
- Un ensemble d'outils
- Dessiner des formes
- Disposer des formes
- Interactions avec les formes

Examples

Le canvas

- Contient et affiche les formes

```
canvas := RSCanvas new.  
canvas open
```

Les formes

- Sous classes de `RSShape`
 - Rectangle : `RSBox`
 - Cercle : `RSCircle`
 - Ligne : `RSLine`
 - Texte : `RSLabel`
 - etc.

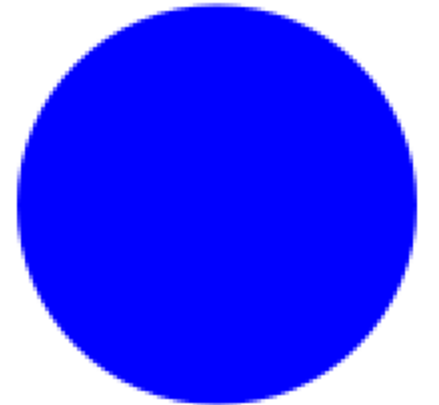
Les formes

- Rectangle

```
rect := RSBox new.
```

- Cercle

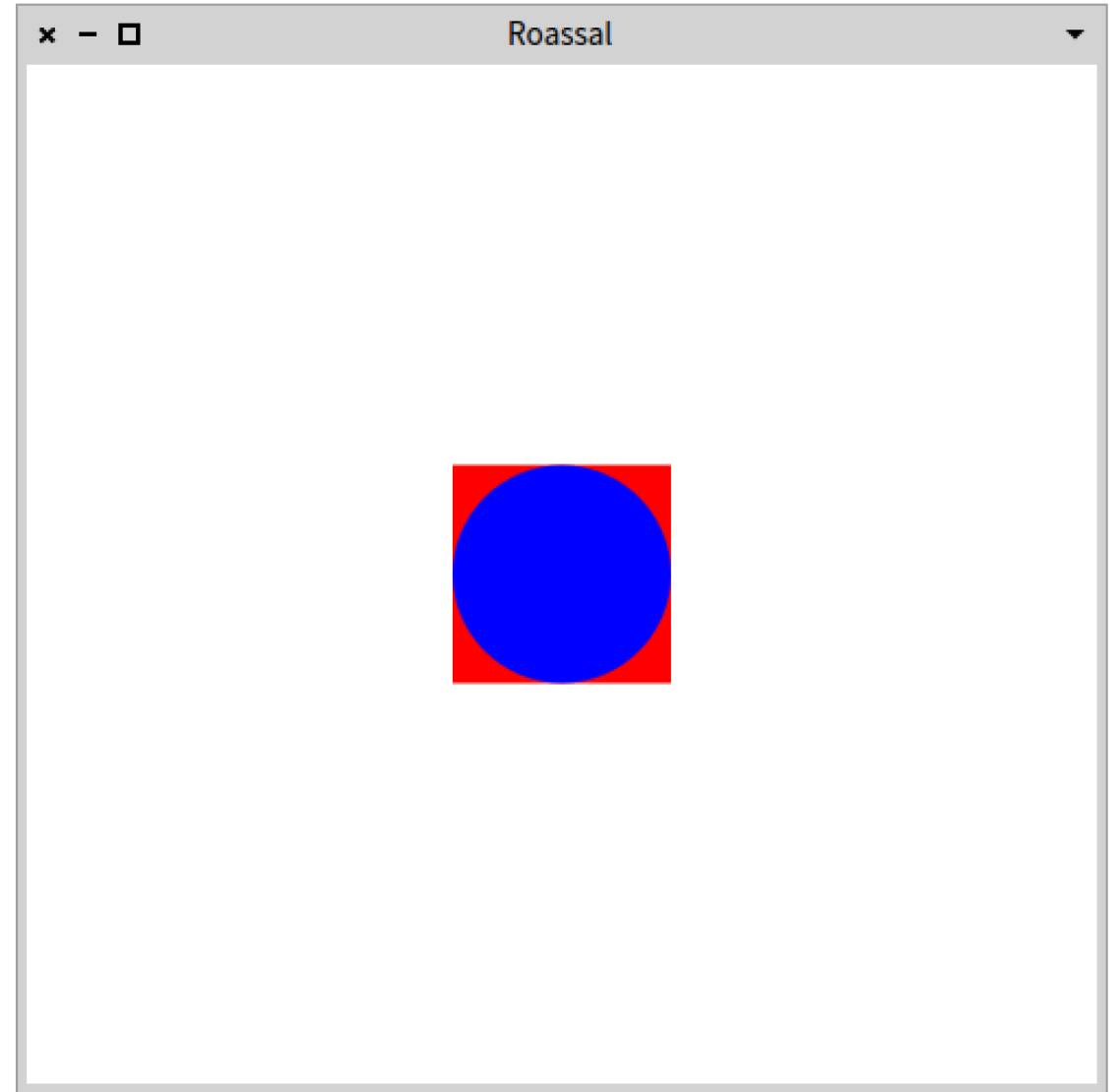
```
circle := RSCircle new.
```



Les formes dans le canvas

- Rectangle

```
canvas add: rect.  
canvas add: circle.  
canvas open
```

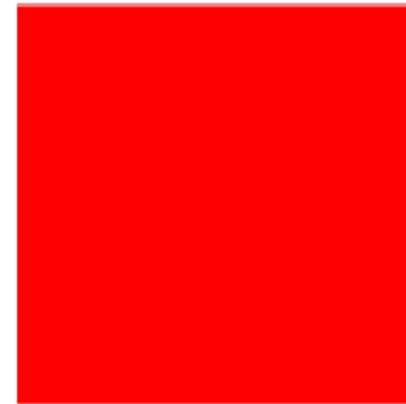


Moduler les formes

Changer la hauteur, largeur, couleur ...

```
rect size: 100;  
color: Color red.
```

```
rect height: 100;  
width: 50.
```



Les formes

- Associer un modèle aux formes
 - Une forme peut représenter un objet Pharo
 - Actions sur la forme en fonction de l'objet représenté

```
shape model: 1
```

Les formes - Exemple de la classe TOTO

Exemple associant un carré à la classe TOTO

Les formes - Exemple de la classe TOTO

Exemple associant un carré à chaque classe du package TOTO



Les formes - Exemple de la classe TOTO

Modifiez la taille de la forme en fonction du nombre de méthodes pour la hauteur et du nombre d'attributs pour la largeur.

Les formes - Exemple de la classe TOTO

Modifiez la taille de la forme en fonction du nombre de méthodes pour la hauteur et du nombre d'attributs pour la largeur.



Les formes - Exemple de la classe TOTO

Modifiez la couleur en fonction du nombre de ligne de code.



Les formes - Exemple des classes du package TITl

Affichez les classes du package TITl avec une taille différentes selon le nombre de méthodes en hauteur, le nombre d'attributs en largeur et une couleur différente selon le nombre de lignes de code.

Les layouts

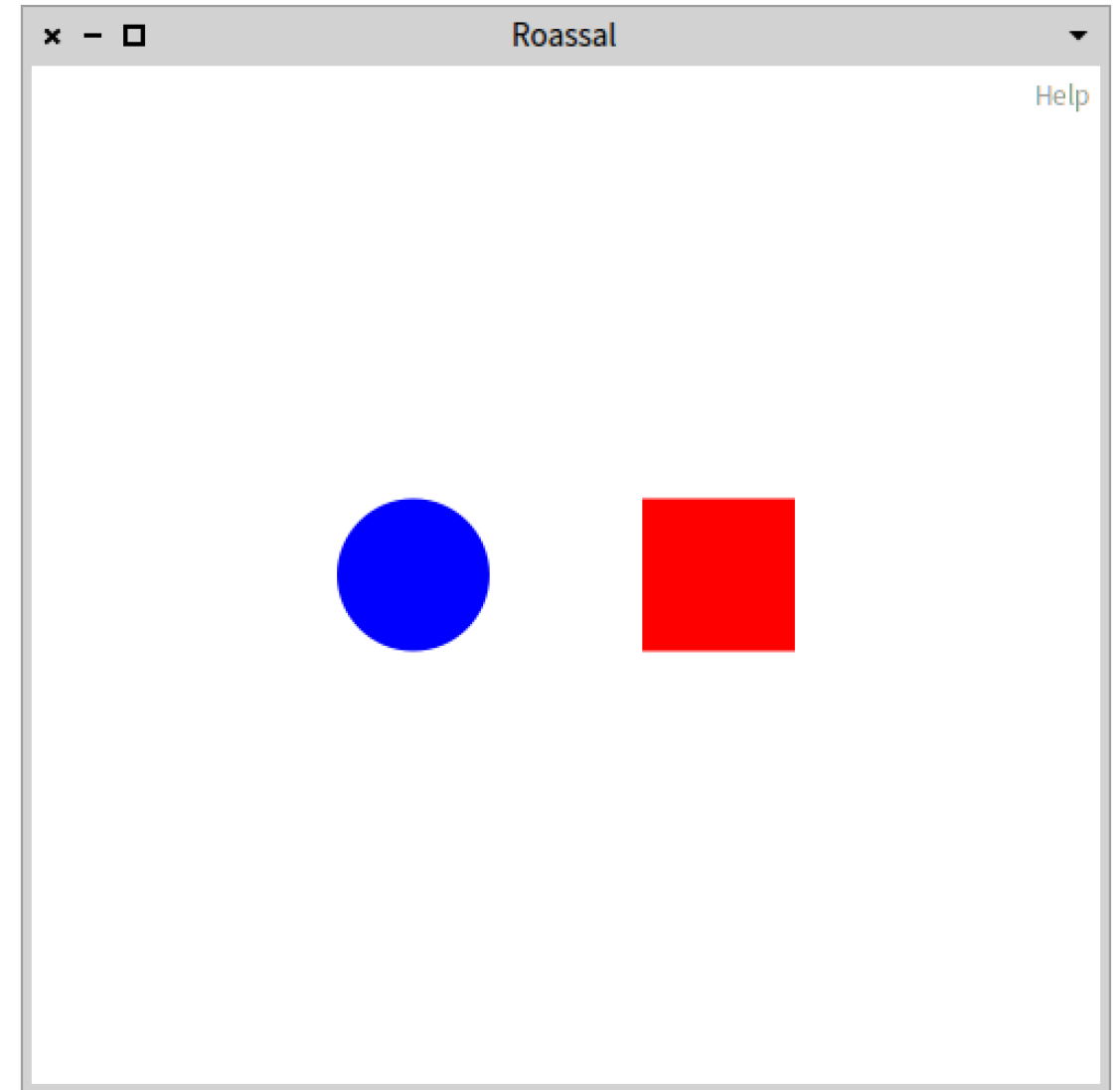
- Permettent de gérer la disposition des objets sur le canvas
 - Disposition horizontale
 - Disposition verticale
 - Disposition arborescente
 - etc.

Les layouts

- Disposition horizontale

`RSHorizontalLineLayout`

`RSHorizontalLineLayout` on: {circle, rect}



Les layouts - exemple du package TITl

- Disposition horizontale `RSHorizontalLineLayout`

```
RSHorizontalLineLayout on: {circle, rect}
```

Les liens

Le TreeLayout - exemple du package TITl

- Disposition horizontale `RSHorizontalLineLayout`

```
RSHorizontalLineLayout on: {circle, rect}
```

Events

- RSMouseEvent

Les interactions

- Draggable
- Popup

Ecrire le code qui va bien avec l'exemple du nom en popup

- highlight

Ecrire le code qui va bien avec l'exemple quand on clique sur un élément

Les interactions avec le canvas

- Draggable
Ecrire le code qui va bien
- Zoomable
Ecrire le code qui va bien

Exemple du package TITI

Ajouter un popup avec le nom de la classe quand on passe au dessus.

Ajouter le highlight

Script Demo

Exercice - À vous de jouer !

- Construire une visualisation qui décrit la hiérarchie du package TUTU.

Étapes à suivre :

1. Collecter les package d'une hiérarchie .
2. Pour chaque package, créer une forme associée. `RSBox`
3. Créer les liens entre les classes. `RSLineBuilder`
4. Appliquer un layout sur les formes obtenues. `RSTreeLayout`
5. Les ajouter dans un canvas. `RSCanvas`
6. Ouvrir le canvas.

Exercice - À vous de jouer !

- Modifier la visu précédente en ajoutant dans chaque noeud package ses enfants (classes et interfaces) sous forme de hiérarchie. Modifier la forme des enfants, avec en hauteur le nombre de méthodes, en largeur le nombre d'attributs, les abstract classes en rouge, les interfaces en jaune et les autres classes en bleu

Ressources

- Github (MIT)
 - <https://github.com/ObjectProfile/Roassal3>
- Documentation
 - [https://github.com/ObjectProfile/Roassal3Documentation:](https://github.com/ObjectProfile/Roassal3Documentation)
- Agile Visualization
 - <http://agilevisualization.com/>