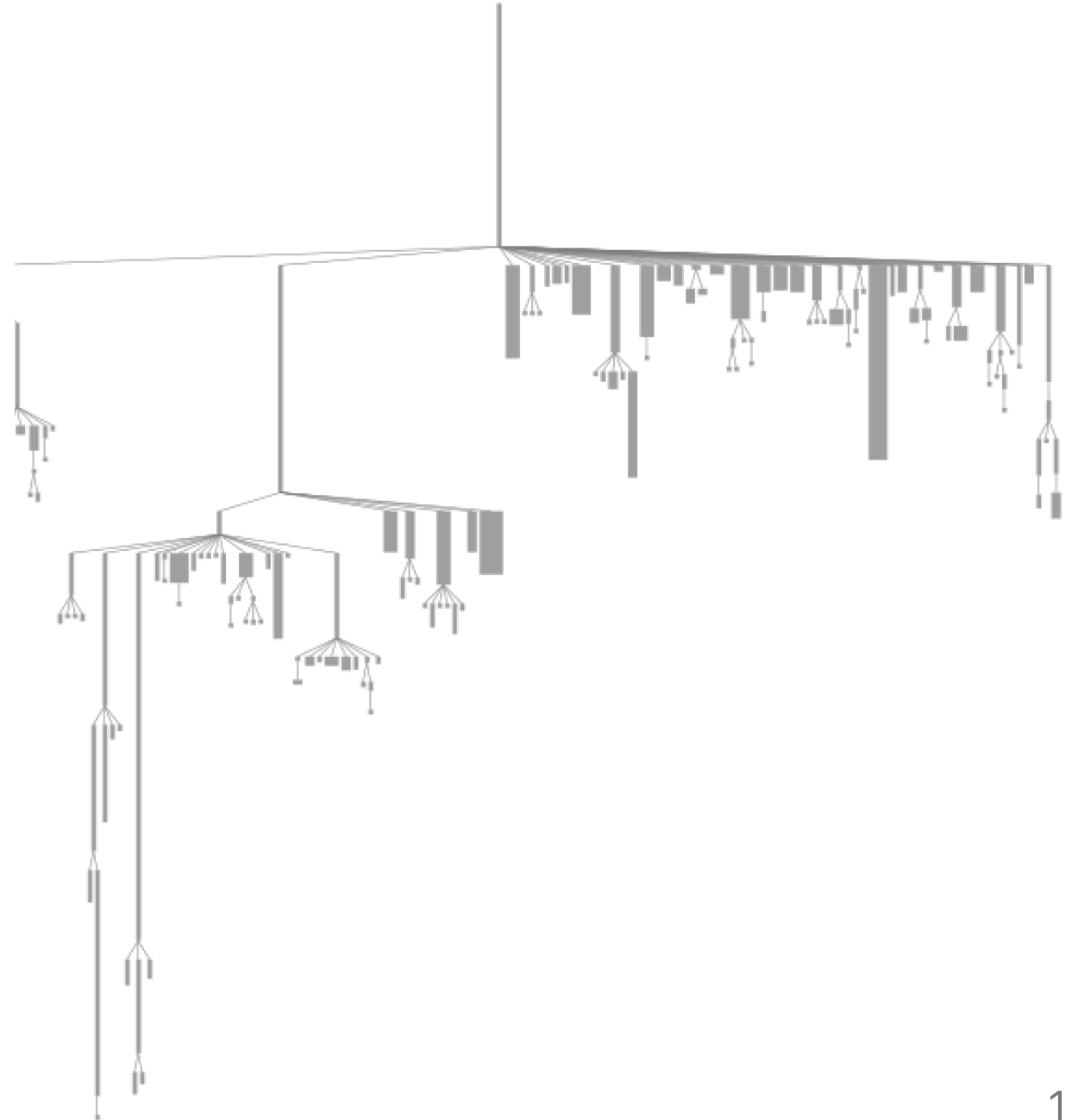


# Roassal



# Roassal

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

# Examples

# Les formes

- Rectangle `RSBox`
- Cercle `RSCircle`
- Ligne `RSLine`
- Texte `RSLabel`
- etc.

# Les formes

- Rectangle

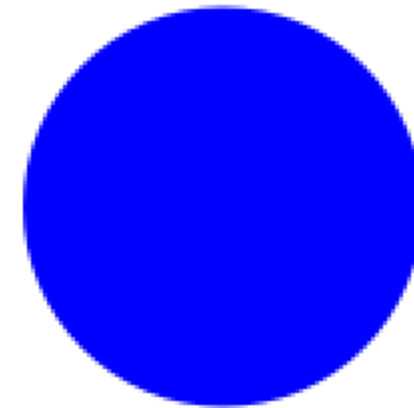
```
rect := RSBox new.  
rect size: 100;  
color: Color red.
```



# Les formes

- Cercle

```
circle := RSCircle new.  
circle size: 100;  
         color: Color blue.
```



# Les formes

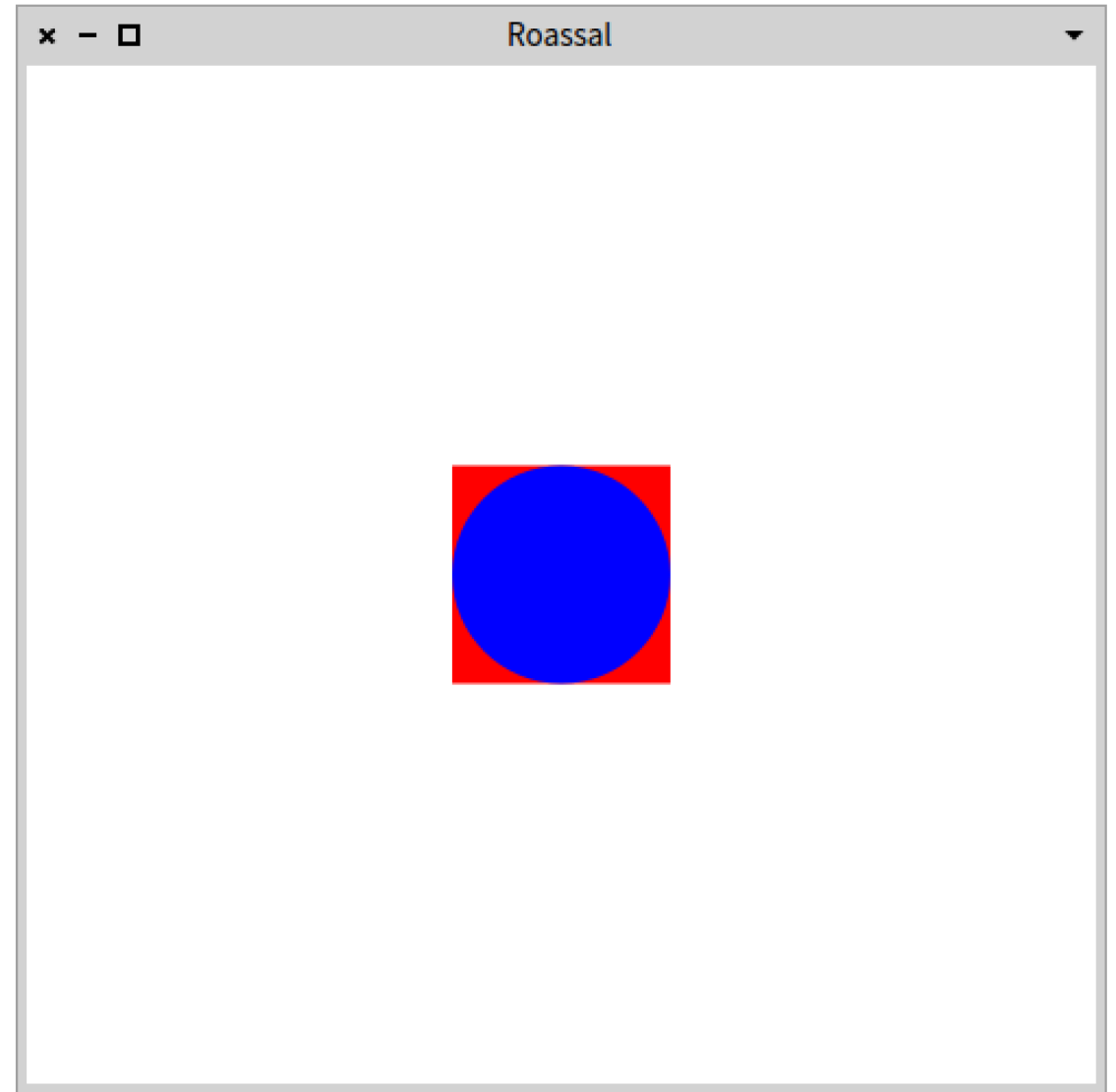
- Associer un modèle aux formes
  - Pour ....
  - ....

```
shape model: 1
```

# Le canvas

- Contient et affiche les formes

```
canvas := RSCanvas new.  
canvas add: rect.  
canvas add: circle.  
canvas open
```





# Les layouts

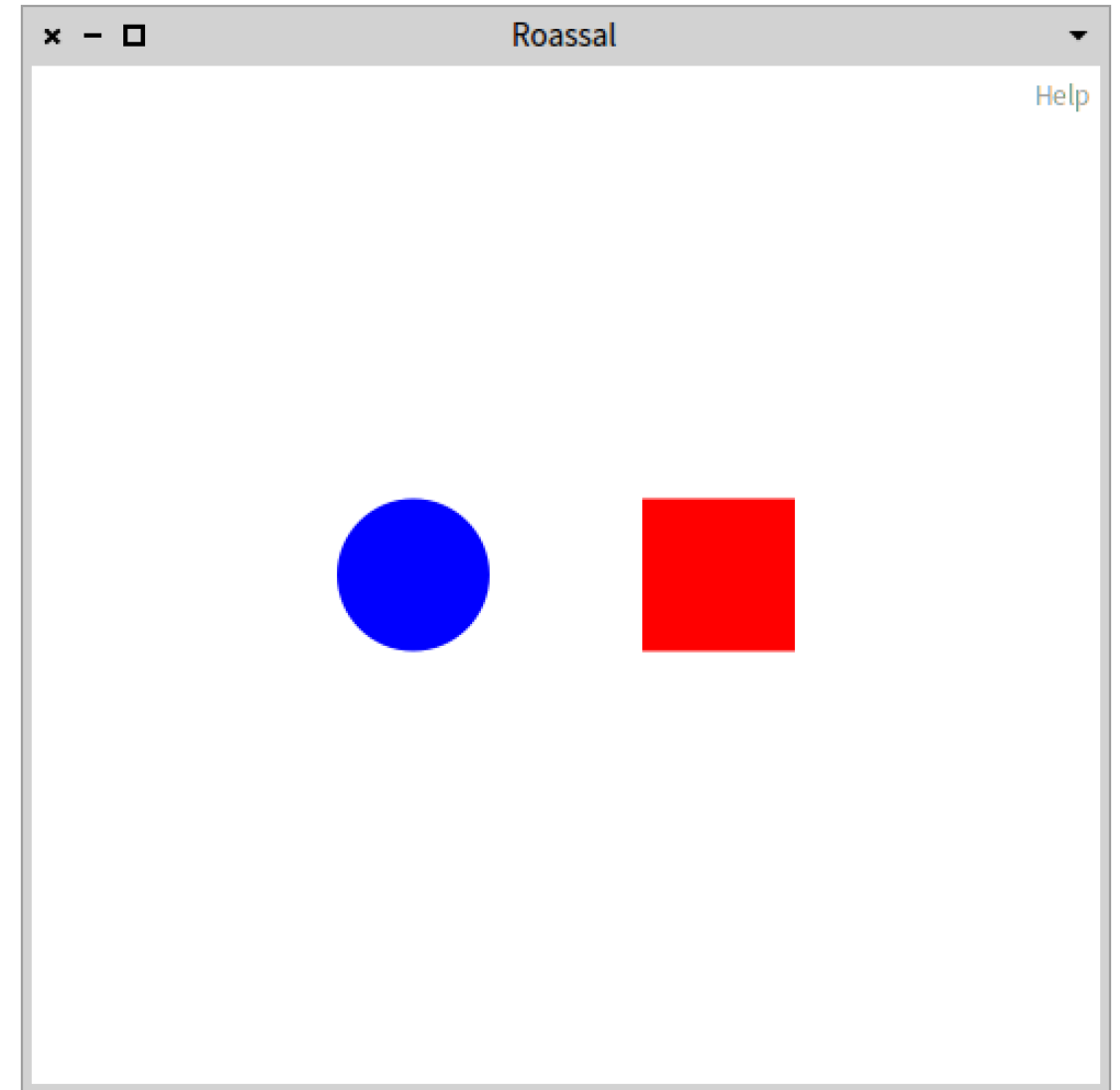
- Permettent de gérer la disposition des objets sur le canvas
  - Disposition horizontale
  - Disposition verticale
  - Disposition arborescente
  - etc.

# Les layouts

- Disposition horizontale

`RSHorizontalLineLayout`

`RSHorizontalLineLayout` on: {circle, rect}



# Les interactions

- Interactions avec les formes
- Interactions avec le canvas

# Script Demo

# Exercice - À vous de jouer !

- Construire une visualisation qui décrit la hiérarchie d'une classe.

Étapes à suivre :

1. Collecter les classes d'une hiérarchie de classe.
2. Pour chaque class, 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.

# Ressources

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