

Software Modeling and Analysis

Fall 2018

S09 : Smalltalk: Software Visualization

Author : Sylvain Julmy

Exercise 1

Here is my code for exercise 1 :

```
1 |b|
2 b := RTSunburstBuilder new.
3 b layout sunburstWithRadius: 100.
4
5 testCls := Smalltalk allClasses select: [ :cls | '*Test' match: cls name].
6
7 b leafWeight: #numberOfLinesOfCode;
8   angularSpacing: 1;
9   radialSpacing: 5.
10
11 b shape
12   color: [ Color gray. ];
13   if: [ :cls |
14     testCls contains: [ :cls2 | cls name , 'Test' match: cls2 name ]
15   ] color: [ Color green. ].
16
17 b explore: Collection using: #subclasses.
18 b build.
19 b view.
```

Figure 1 show the output of the previous listing.

Exercise 2

Here is my code for exercise 2 :

```
1 |b|
2 b := RTTreeMapBuilder new.
3
4 b shape
5   if: [ :obj | obj isClass]
6   color: [ :cls |
7     (cls subclasses notEmpty) & ('*Array*' match: cls name)
8     ifTrue: [ Color green ]
9     ifFalse: [ Color veryLightGray ] ].
10
11 b leafWeight: #numberOfLinesOfCode;
12   explore: Collection using: #subclasses.
13 b build.
14 b view.
```

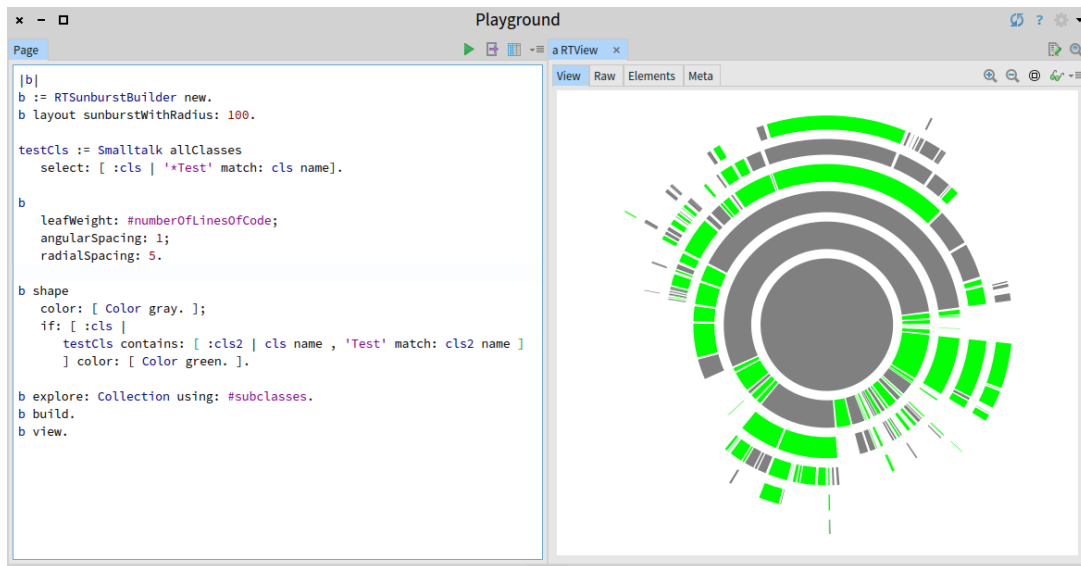


Figure 1: Output for exercise 1.

Figure 2 show the output of the previous listing.

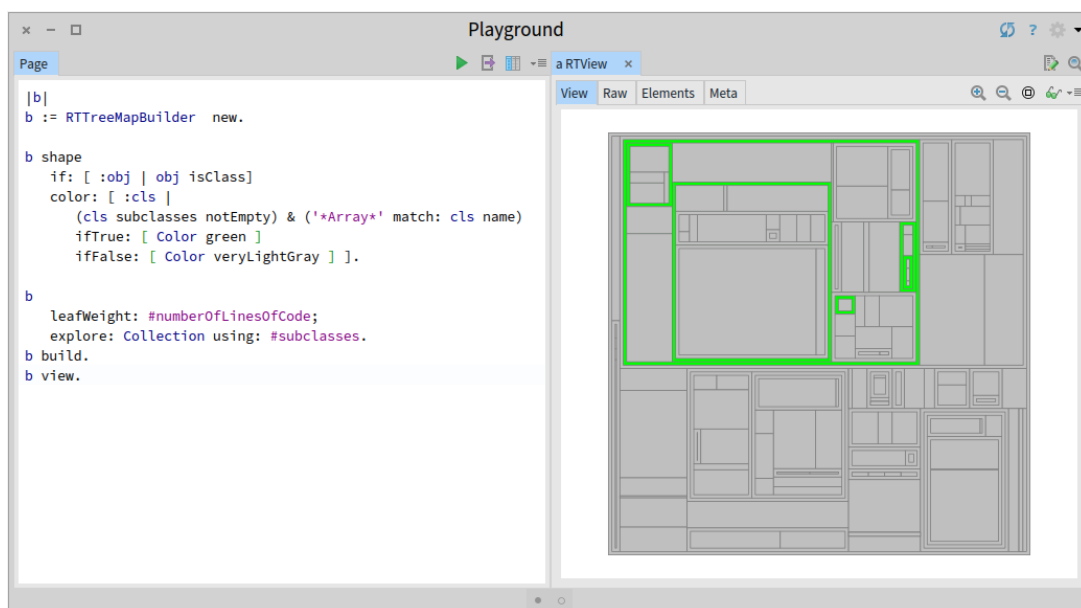


Figure 2: Output for exercise 2.

Exercise 3

Here is my code for exercise 3 :

```
1 |b|
2 classes := RTLayout withAllSubclasses , Collection withAllSubclasses.
3 b := RTMondrian new.
4 b shape circle
5   size: 8;
6   color: [ :cls |
7     (Collection withAllSubclasses contains: [:cls2 | cls = cls2 ])
8     ifTrue: [ Color red ]
9     ifFalse: [ Color green ] ].
10 b nodes: classes.
11 b edges connectFrom: #superclass.
12 b shape
13   bezierLineFollowing: #superclass;
14   color: (Color blue alpha: 0.2).
15
16 b normalizer
17   normalizeSize: #numberOfMethods using: #sqrt.
18
19 b edges
20   notUseInLayout;
21   connectToAll: #dependentClasses.
22
23 b layout cluster.
24 b build.
25 b view.
```

Figure 3 show the output of the previous listing.

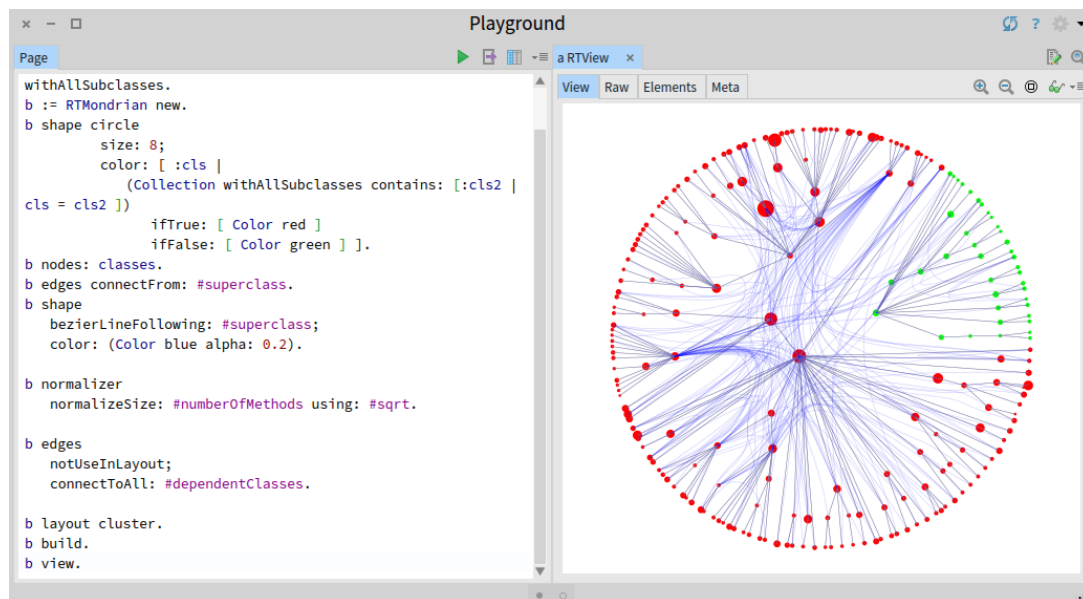


Figure 3: Output for exercise 3.

Exercise 4

Sunburst

Strenghts :

- Great for navigating heirarchical data.
- More effective at visualizing a big dataset.
- Doesn't lose the middle layers of hierarchy.

Limitations :

- Hard to estimate the exact value using arc length.
- Small, periphery arcs can be hard to see/analyze without interaction.

Treemap

Strenghts :

- The eyes visually aggregates rectangles in the same group, allowing to see patterns quickly.
- Similarity inside a group give relevant information.
- We can detect anomaly very quickly by seeing a divergent color inside a group.

Limitations :

- It's hard to show negative and zero value inside a treemap.
- Printing a treemap is quite hard and label would overlaps very easely.

Mondrian

Strenghts :

- We can quickly see dependencies relation and the high dependable class.
- The interconnection between the subclasses of a specific class.
- Can handle a large amount of data.

Limitations :

- We can't easely see precise dependencies if there is too many edges.
- We can't see dependencies between the leaves of the tree.