

Assignment 09 — 14/11/2018 – v1.0

Smalltalk: Software Visualization

Please submit this exercise by mail to sma@list.inf.unibe.ch before 21 November 2018, 10:15am, and **use the *file out* feature in the *System Browser* to export the Smalltalk classes you created.**

Note: For the following exercises you should use the pre-configured Moose 6.1 environments available in 32 bit flavor for [Linux](#), [Windows](#), and [macOS](#). Please choose the correct version for download in accordance with your current platform.

Exercise 1: Sunburst visualization with Roassal (2 Points)

Build a Sunburst visualization as shown in Figure 1 to analyze the test coverage of the `Collection` class hierarchy. Each tile represents a specific class, and the size of the tile should represent its number of lines of code. Moreover, tested classes (i.e., classes covered by tests) should be colored in green, while other classes should remain in grey.

Hint: You can assume that test classes (i.e., classes that test other classes) use a name which closely resembles the original name of the class they test; in general, they add only the postfix `Test` to the original class name (e.g., `ByteArray` will become to `ByteArrayTest`).



Figure 1: Sunburst visualization built with Roassal

Exercise 2: Treemap visualization with Roassal (2 Points)

Build a treemap visualization as shown in Figure 2 to gather an overview of classes that have subclasses, and that contain the string `Array` in their names. The size of the tiles must encode the number of methods of the class they represent. Furthermore, classes that contain the string `Array` in their names and that have subclasses should be colored in green, while all remaining classes should remain in grey.

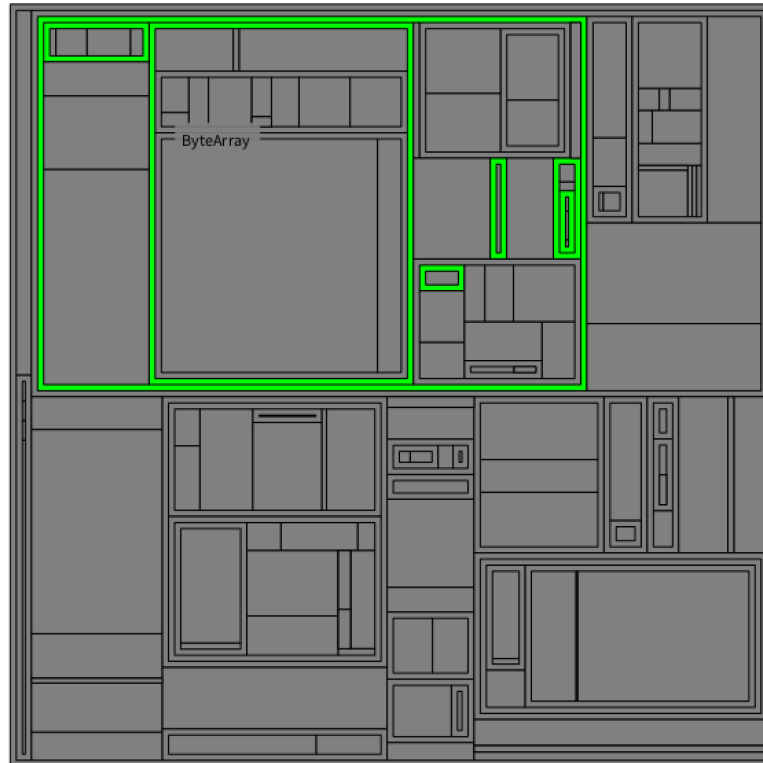


Figure 2: Treemap visualization built with Roassal

Exercise 3: Node-link visualization with Roassal (3 Points)

Node-link diagrams as shown in Figure 3 can provide valuable insights of the relationships amongst the classes in a system. In this exercise you have to create a visualization using the `Mondrian` builder to analyze the class dependencies between the `Collection` class hierarchy and the `RTLayout` class hierarchy. To this end, you have to:

- i) Visualize the classes of both hierarchies using circles (i.e., `RTEllipse`)
- ii) Use a different color to differentiate the classes of each hierarchy.
- iii) Add edges to depict the class hierarchy, while using the `RTClusterLayout`
- iv) Add blue Bézier edges to depict class dependencies

Provide the Roassal code to implement such a visualization, and use it to identify the classes in each hierarchy that have the highest number of dependencies.

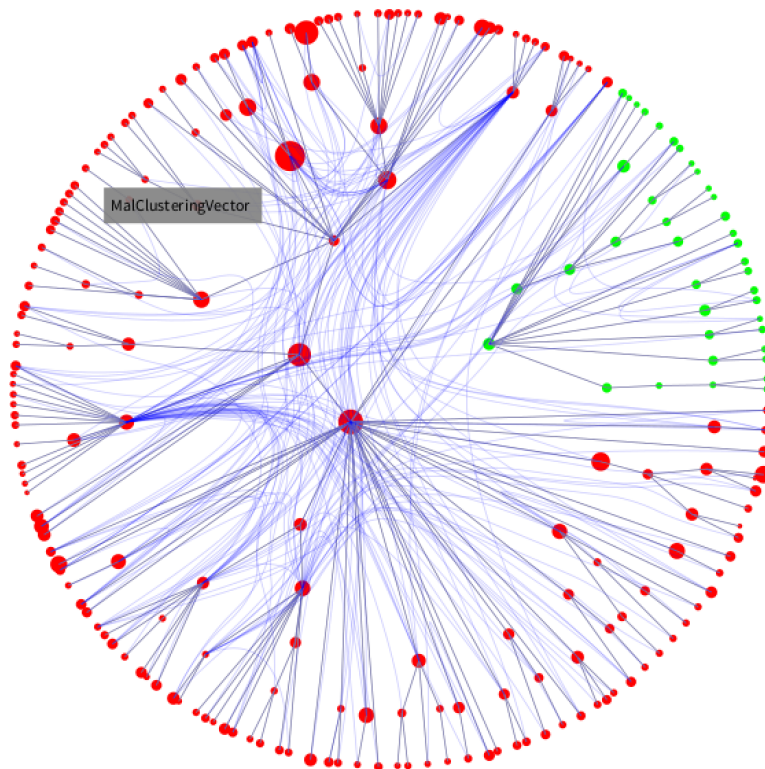


Figure 3: Node-link visualization built with Roassal

Exercise 4: Discussion (3 Points)

Comment on the strengths and limitations of each visualization you just created.