**Working with Roassal and Sample Visualization**

In this document I will be explaining about my understanding of Roassal. I will be covering various basic functionalities Roassal offers and how we can use it create a visualization using the data we generate from twitter.

After installing Pharo, run it with Moose image. Execute the following code.

Gofer new smalltalkhubUser: 'ObjectProfile' project: 'Roassal'; package: 'ConfigurationOfRoassal'; load. (Smalltalk at: #ConfigurationOfRoassal) load

This will install Roassal into Pharo. This will download many packages offered by Roassal.

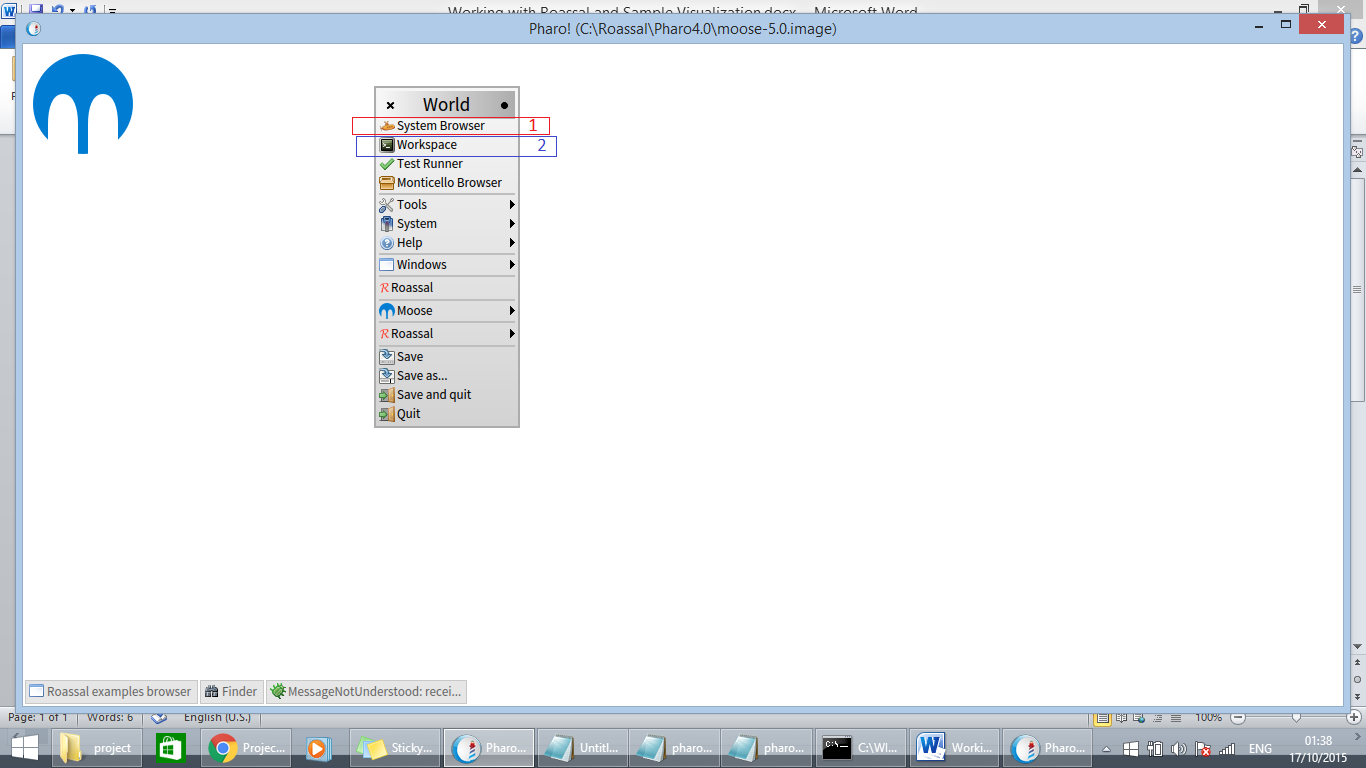


Image 1.

The above image shows 2 basic functionalities that we will be using very often in our project viz. system browser and workspace.

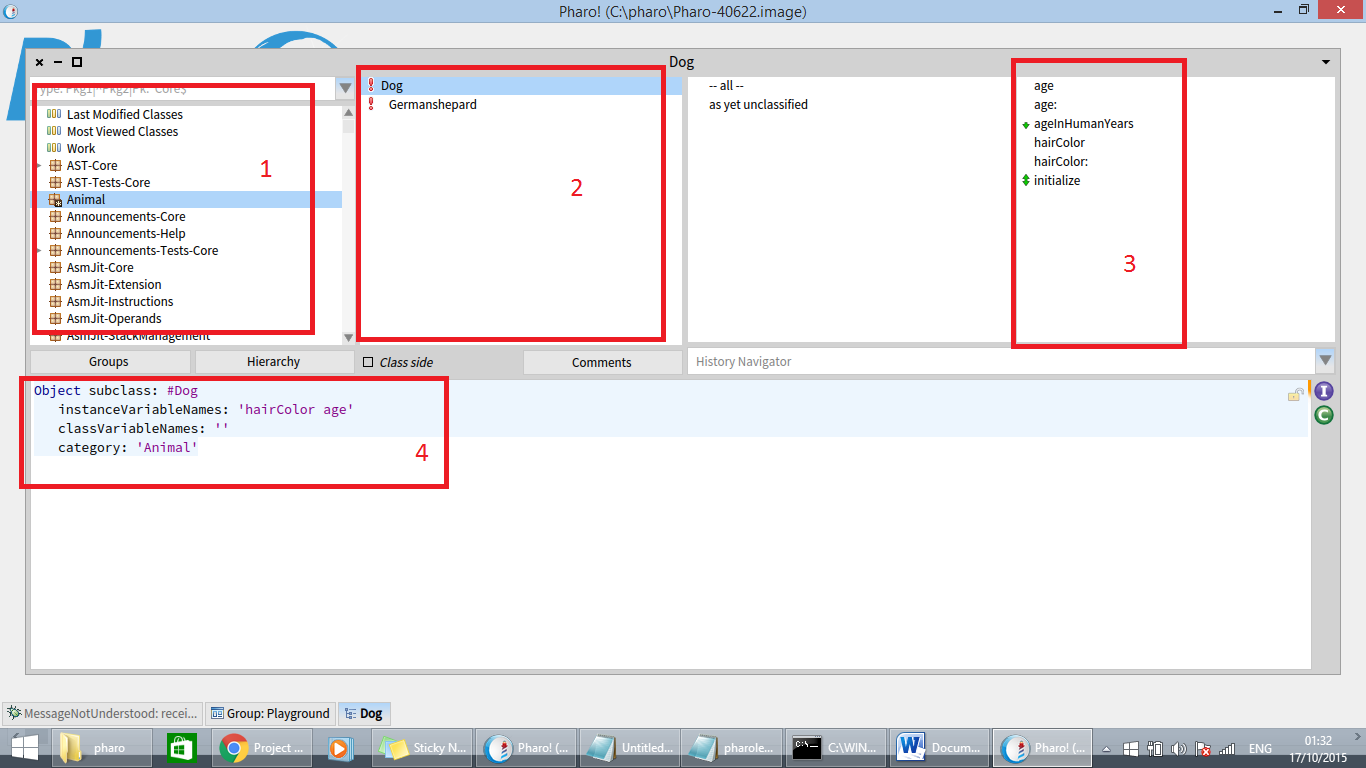


Image 2.

The above image shows various sections of system browser. I have created a simple package, class and functions based on tutorials offered by [Dimitris Chloupis](https://www.youtube.com/channel/UCJMkk_HLknuhO6kk6LLBHcg) found on web. I have shared the same on Git. The first block contains packages; the second have the classes while the third block defines functions. The fourth block is where we write our code defining classes and functions. Classes have 3 kind of variables viz. instance, local and class. Instance variables should be accessed using new keyword whereas class variables can be accessed directly. Local variables are declared within “|”.

“:=” is used as an assignment operator.

Pharo supports Inheritance.

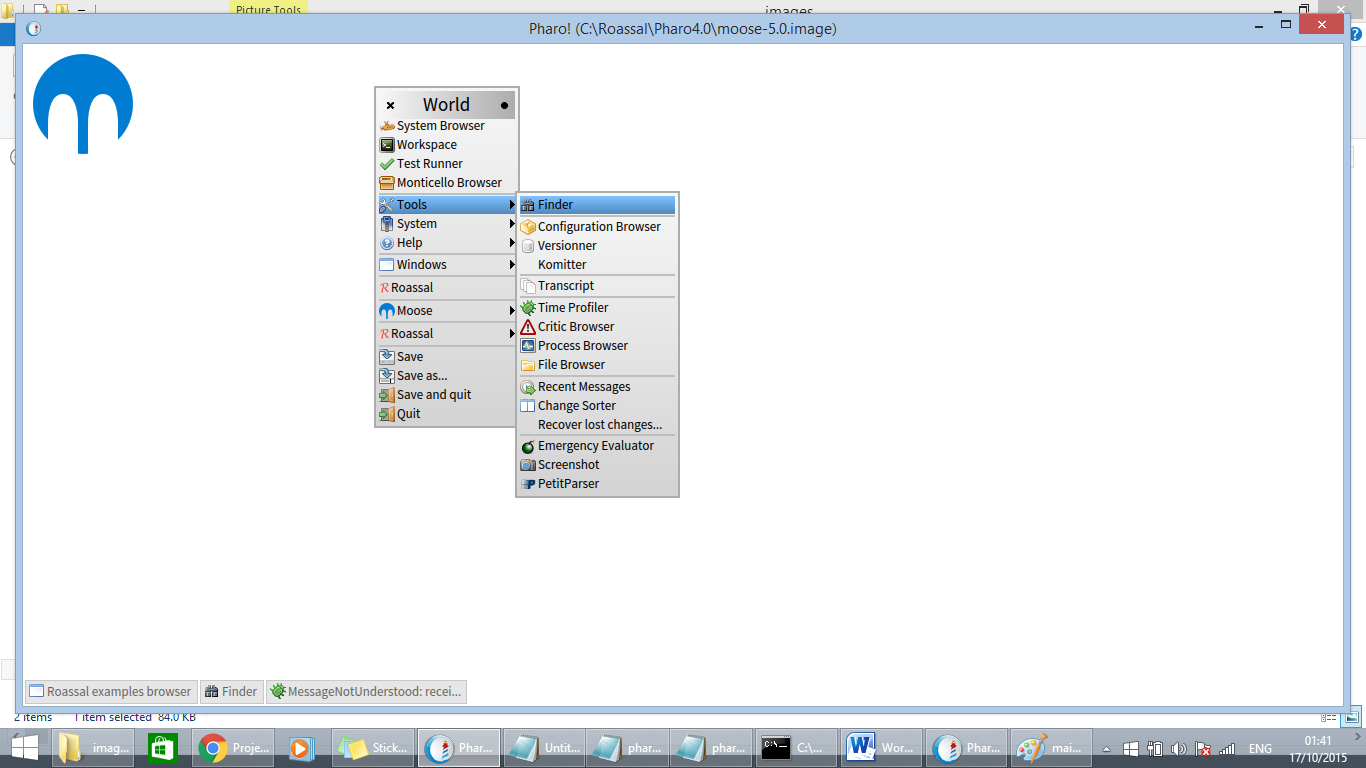


Image 3

The finder is a very useful functionality offered by Roassal which help in finding any class or function easily.

Pharo offers a feature where we can use C, JAVA and JavaScript libraries. We can import these libraries in case we face any short comes during our projects.

Pharo offers array and OrderedCollection to store set of data. OrderedCollection can grow in size. It is similar to list. Pharo also supports Dictionary, where instead of storing index it stores keys.

Example:

tab1 := Dictionary new.

tab1 at: #name put: 'anoop';

at: #lastname put: 'jatavallabha vijayakumar';

at: #address put: 'somewhere in a city';

at: #city put: 'somewhere in a country';

at: #country put: 'somewhere on earth'.

“;” acts as cascade.

Sample codes:

|  |
| --- |
| | b table |  b := RTMetricMap new.  table := RTTabTable new.  table input: 'Canada 0.3000  China 0.1000  Australia 0.5000  India 0.8000' usingDelimiters: String tab.  table convertColumn: 2 to: [ :t | Float readFrom: t ].  b allCountriesColor: Color white.  b countries: table values named: #first metric: #second.  b open |

(Reference ObjectProfile)

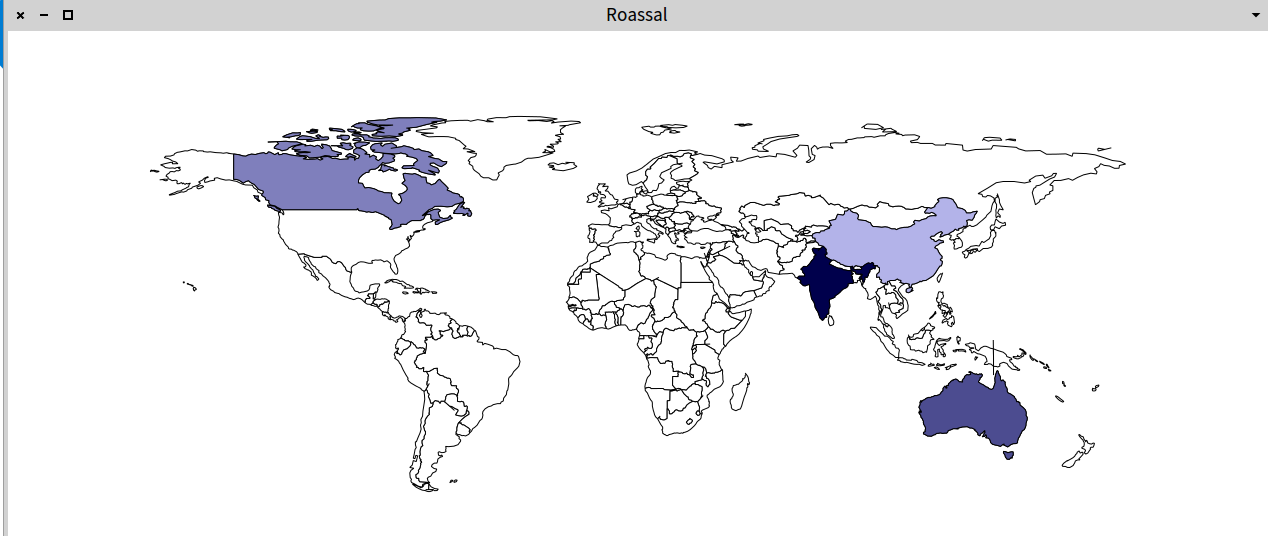


image 4

The above code uses a inbuilt class called RTMetricMap and generates a map. The color gradient is controlled by the values the country hold in the table. We can use this as an example and make our visualization depending on depth of public opinion. We can calculate and assign a value to each region based on public opinion on twitter. This will help in visualizing the data in hand.

References:

1. [Dimitris Chloupis](https://www.youtube.com/channel/UCJMkk_HLknuhO6kk6LLBHcg): <https://www.youtube.com/watch?v=Ol5ivaEATLQ&list=PLqbtQ7OkSta0ULYAd7Qdxof851ybh-_m_>
2. <http://objectprofile.com/ObjectProfile.html>