Omnigram Explorer Technical Documentation

Version 1.0

Tim Taylor, 12 December 2014

Introduction

Omnigram Explorer (OE) is written in *Processing* (http://processing.org). It was developed using *Processing* version 2.2.1.

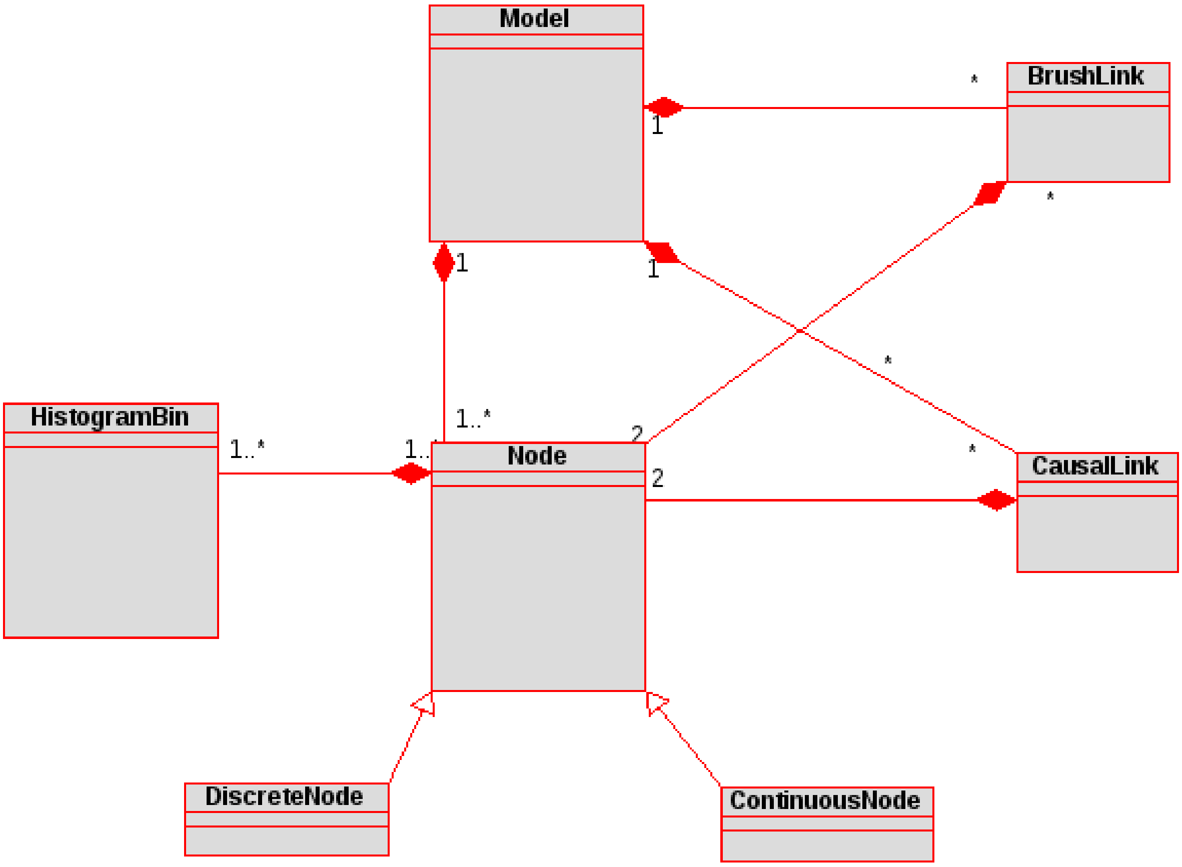
It can be run from within the *Processing* environment, or it can be compiled to standalone executables on Windows, Mac and Linux. *Processing* now has an option to compile to a Javascript target. However, this does not work straightforwardly with the OE code, at least with *Processing* version 2.2.1. This might be because OE makes use of some underlying Java libraries in addition to pure *Processing*. Perhaps future releases of *Processing* might do a better job at converting to Javascript.

Code Repository on GitHub

The OE code can be obtained from https://github.com/tim-taylor/omnigram

Code Structure

The source file `omnigram.pde` contains the top-level `setup()`, `draw()` and associated methods. The `setup()` method displays a file selection dialog for the user to select a Model Definition XML file. Once this is selected, the `modelFileSelected()` callback function instantiates a new `Model` object. The `Model` object is where all of the action is controlled from. A basic UML diagram of the OE code structure is shown below.



UML diagram of the major classes in Omnigram Explorer

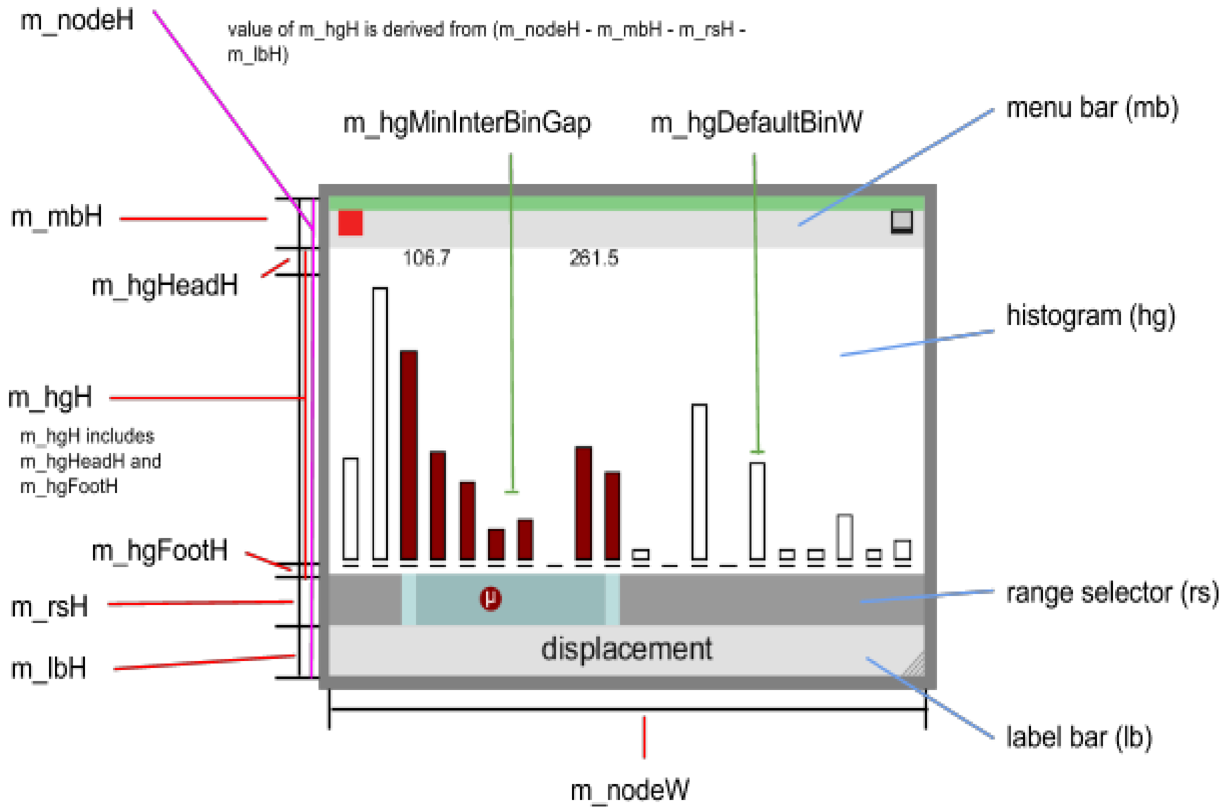
Screen Geometry

It is possible to zoom the OE display in and out (using the + and - keys), and this is implemented using the *Processing* `scale()` method. Note that this means that when checking for mouse interactions with the nodes and other UI components, we need to take the current scale factor into account rather than straightforwardly using the mouse's `x` and `y` coordinates. The `Node` class has methods `scaledMouseX()` and `scaledMouseY()` to do this.

Geometry of a Node

The `Node` class contains various member variables that control the size of the Node as displayed on screen.

The major variables and associated meanings for `Node` sizing are shown in the diagram below.



Details of the specification of size of the main components within a Node