Features planned in future releases

Some features under development for the next major release are listed here. Items 1 and 2 have already been partially implemented in Version 5.2.

1. *Node and edge selection prompts in algorithms.* An algorithm will be able to ask the user to select a node or edge via a text query. The appropriate incantations will take the form
   - `getNode(`*`message`*`)`, where *message* is a prompt (string); a popup window asks the user for the id of a node; the function will return the actual node; there is an error dialog if no node with that id exists.
   - `getEdge(`*`message`*`)` – same as `getNode` except that the user is prompted for two node id's; an error dialog arises if either id is non-existent or the edge is non-existent.

2. *Sets of nodes and edges.* New types/classes `NodeSet` and `EdgeSet` will be added. Both will be concrete classes and will have conversion constructors from `NodeList` and `EdgeList`, respectively. Eventually lists, queues, stacks and priority queues of nodes and edges will all be concrete and all graph, node and edge methods that return containers will return one of these. For example, `getNodes()` will return a `NodeList` instead of a `List<Node>`.

   When combined with the previous enhancement we can add functions such as
   `getEdge(`*`prompt, set, message`*`)`,
   where *prompt* is as before, *set* is a set from which the edge must be selected (e.g., the set of edges incident on a node) and *message* is an error message to be given if the edge exists but is not a member of the given set.

3. *Mode-less graph editing.* In place of the GDR-like mechanism, where panel buttons are used to determine how the graph editor responds to mouse actions – a click might create a node, initiate an edge or select an object – the Ctrl and Shift keys can determine the "mode". So, for example, Ctrl-Click would create a node and Shift-Click would initiate an edge.

4. *Mapping attributes to actions.* In order to make animations more accessible to visually impaired users, there should be a mechanism that, under user control, specifies how Boolean attributes such as marking or highlighting are "displayed". Currently, the thickness of highlighted node borders and edges can be controlled in the Preferences panel. A more sophisticated mapping mechanism that incorporates sound as well as visuals is needed. The ultimate approach would allow mappings for arbitrary attributes defined by user or programmer.