

Node focus requirements

Contexts

- Mouse + Keyboard
- Virtual reality
- Users on other instances of slicer when collaborating (m+k, VR)

Soft focus

- Soft focus on N-nodes at a time, each with any number of selected component type/index pairs
- Not saved with scene
- Outline visualization
- Triggered when hovering over objects in combo-boxes or SH, or in views

Hard focus

- Hard focus on N-nodes at a time, each with any number of selected component type/index pairs
- Hard focus is saved with scene
- Simple bracket visualization around selected objects RAS bounds (2D)
- Triggered when clicking on objects in combo-boxes or SH or in views

Display options

- Outline color
- Outline thickness
- Visible view IDs

Displayable manager

- Query all other displayable managers looking for actors that match the focused display node IDs
- Apply default focus visualization based on the type of actor
- Displayable managers are given the option of handling their own focus visualization, which may have more specialized requirements that the default implementation can't handle.

Tracking node focus

Current implementation involves setting node references on the singleton `vtkMRMLSelectionNode`, as well as component type/indexes when needed (markups/segmentations).

- Singleton selection node per context
- Convert singleton node to a displayable node
- Create new display node to manage display properties

Misc. questions

- Can modules/widgets control the visibility of outlines independent of user action
 - Ex.
 - Input/output combo boxes within a module highlighting input/output nodes with different outline colors
 - Tutorial highlighting elements within the scene independently
- How can users perform multi-selection
 - Already handled in Qt widgets
 - Should users be able to Shift+Left click to select multiple nodes in views?
 - Can users click and drag a box in views to select every visible node in the box?
-