Harvey Mudd College
Computer Science 60
Fall 2010

Assignment 7
Sequential Logic GUI
Due. 11:59 p.m., Wed., 3 November 2010

This assignment involves implementing at least 2 points worth, **per person** (so a team of two implements 4 points worth) of the use-cases for the sequential logic GUI (graphical user interface) discussed in class. A baseline implementation of this GUI will be available as a zipped project directory at

http://www.cs.hmc.edu/courses/2010/fall/cs60/slog.zip

If your implementation of one or more of the features accepted, you may achieve fame as co-author of the system, if willing. You may also accrue extra credit for implementing more points than the minimum.

While not absolutely required, use of the Netbeans IDE is advised for this assignment. That will make it easier for you to exploit the baseline and easier for any contributions to be integrated into the baseline. In any case, you will submit a zipped directory of the project with your additions. It must be compilable by Java as is, without any other code added to it. Please do not make purely gratuitous changes that will make your code harder to test. Authorship of features must be noted in the corresponding source files. You should also include a README.txt file that indicates what you did (use-case name and a brief description).

Point Value (0 indicates already done)	**Use-Case Name**	**Brief Description**
0	Draw Node	User draws a node of selected type.
0	Move Node	User moves a node to a new position.
0	Delete Node	User deletes a node.
1	Change Node Name	User changes the name of a node.
1	Inject Value	User forces the current value of a node to be one of the two logic values.
2	Change Node Type	User changes the type of a Node (requires replacing and reconnecting).
1	Resize Node	User changes the size of a Node after initial construction.
2	Select Nodes	User selects multiple Nodes, for subsequent purposes, such as deleting all.
1	Delete Nodes in Selection	User deletes all Nodes in a previous Selection with a single action.

Point Value (0 indicates already done)	Use-Case Name	Brief Description
2	Copy Selection	User makes an off-screen copy of the Nodes in a Selection, including any Connectors connecting them.
2	Paste Selection	User pastes a previously-copied Selection at a designated point in the panel.
2	Move Selection	User moves the current Selection to a new position using the mouse.
2	Group Selection	The current Selection is made into a Group, which becomes movable, deletable, copyable as a sing unit.
2	Ungroup	A selected Group is ungrouped into its original nodes and connectors.
0	Draw Connector	User draws a Connector, with any desired articulation Points, between two Nodes.
1	Delete Point from Connector	User selects a Point within the Connector, which is then deleted.
1	Add Point to Connector	User adds a new Point within the Connector, for purposes of greater articulation.
2	Move Connector from one Node to Another	User moves either end of a Connector to another Node without destroying it.
2	Add Junction to Connector	A Junction is a node that provides the same value to all Connectors connected to it.
1	Clear All	Deletes all Nodes and Connectors from the current Circuit.
3	Undo/Redo	Provide an infrastructure so that any user command can be undone, or an undone command redone, an arbitrary number of levels.
3	Attach input stream	A stream of inputs is provided to an In Terminal, so that on each step, a new input is assimilated.
3	Attach output stream	The values of an Out Terminal are recorded as a stream, during Circuit simulation.
2	Load and save Circuit Description	Load and save Circuit descriptions as text using an appropriately designed S expression.

You may suggest other use cases, and implement them if approved.