UML Editor Third Iteration
Design Summary
12/2/2016
Team: FiVe
Ryan Peterson
Eric Dougherty
Kelsey Fulton
Timothy Kettering
Matt McAnulty

############################
UML Editor Design Summary
############################

The focus on the overall design for the third iteration of the UML editor was continuing the development of the basic elements for an intuitive, consistent, and thought out user interface that works well with the software's design while finishing the software so that it has a high level of functionality. Like the last iteration, functionality was the top priority of this iteration. We continued to focus on the behavior and look of our software, but we also wanted to finish the software so that is fully functional upon submission date. We were hoping to combine the focus on behavior and look of the software from the first iteration with an emphasis on functionality in the second and this iteration.

##################################
Current Functionality/Behavior
##################################

At this point our UML editor opens full screen and displays a standard horizontal menu bar, a vertical toolbar on the left, and a large central staging area/work space in the center. Currently the user can create a box by clicking the box button on the toolbar. A box appears and can be moved by the user. To move the box the user can mouse down and hold over the box. Once this is done a grid will appear on the screen in the stage/workspace area. While still moused down (and the grid is present) the user can move the mouse to change the position of the box in the staging area. Releasing the mouse button will snap the box on the grid at the selected location, and the grid view will disappear. The user is able to add text to the box in four different sections. The sections are the class name, attribute, operation, and miscellaneous. Class name must have some type of input, but the other three do not need input. If they are not used, the sections are minimized.

Once a box is selected the user has the option to erase the elected box. To do so, the user must click the delete button on the vertical tool bar. The user also has access to one more button on the tool bar when two boxes are present in the workspace and one box is selected. The user has the option to add a relation. Once the user chooses to create a relation, a generalized relation appears, and the user must extend it to another class box. If the user selects the relation, ten more options appear on the tool bar. The user may change the relation type but selecting the type of the appropriate relation from the tool bar. The user may also change the type of line to dotted or back to solid by clicking the appropriate box on the tool bar. The user can reverse the direction of the relation by selecting the appropriate box on the tool bar as well. The user may change the relation from having one head to two by also selecting the

appropriate buttons on the tool bar. Finally, the user may erase a relation by selecting the delete button from the tool bar. If the user clicks on the relation, a text box opens to add text to the relation line. All the buttons on the tool bar have images that depict the functionality of the button. If the user selects in the workspace at any time, this deselects the relation and removes the unnecessary buttons from the tool bar.

##########################
User Interface Design
##########################

The UI design is selection focused. The idea here being that objects once rendered in the workspace area are selectable. Once an object is selected the toolbar will display actions that can be performed on that object. This is to streamline the interface for the user in an attempt to create a simplified, and intuitive experience. It also serves a practical use as well. It allows for future scalability of the toolbar interface to include more buttons/features on a single toolbar without worrying about making all the buttons fit into the usable area within the toolbar simultaneously. For example when a class box is selected then toolbar should show options unique to a class box. When a relation is selected, the tool bar shows options unique to a relation. The hope is that doing this will provide consistent UI experience while still being intuitive.

###################
Software Design
###################

The program is written using JavaFX. When launched, the program launches to full screen and creates a staging area for the application. It also creates an instantiation of the controller class. The controller class is responsible for displaying all the elements of the UML Editor application such as the toolbar, boxes, relations, the workspace, arrows, and text labels. The program also uses a CSS style sheet for its visual motif and physical layout. The toolbar, controller, and the workspace are all singletons in that there will only ever be one of each. The program uses the Model View Controller (MVC) design with the controller class functioning as the controller, and having access to the relation, box, toolbar and workspace classes.

######################
Future Design/Plans
######################

For future iterations of UML Editor the team would like to implement features such as adding multiplicity labels to the lines. We'd also like to create a help menu and possibly allow for different preferences.