**Overview:**

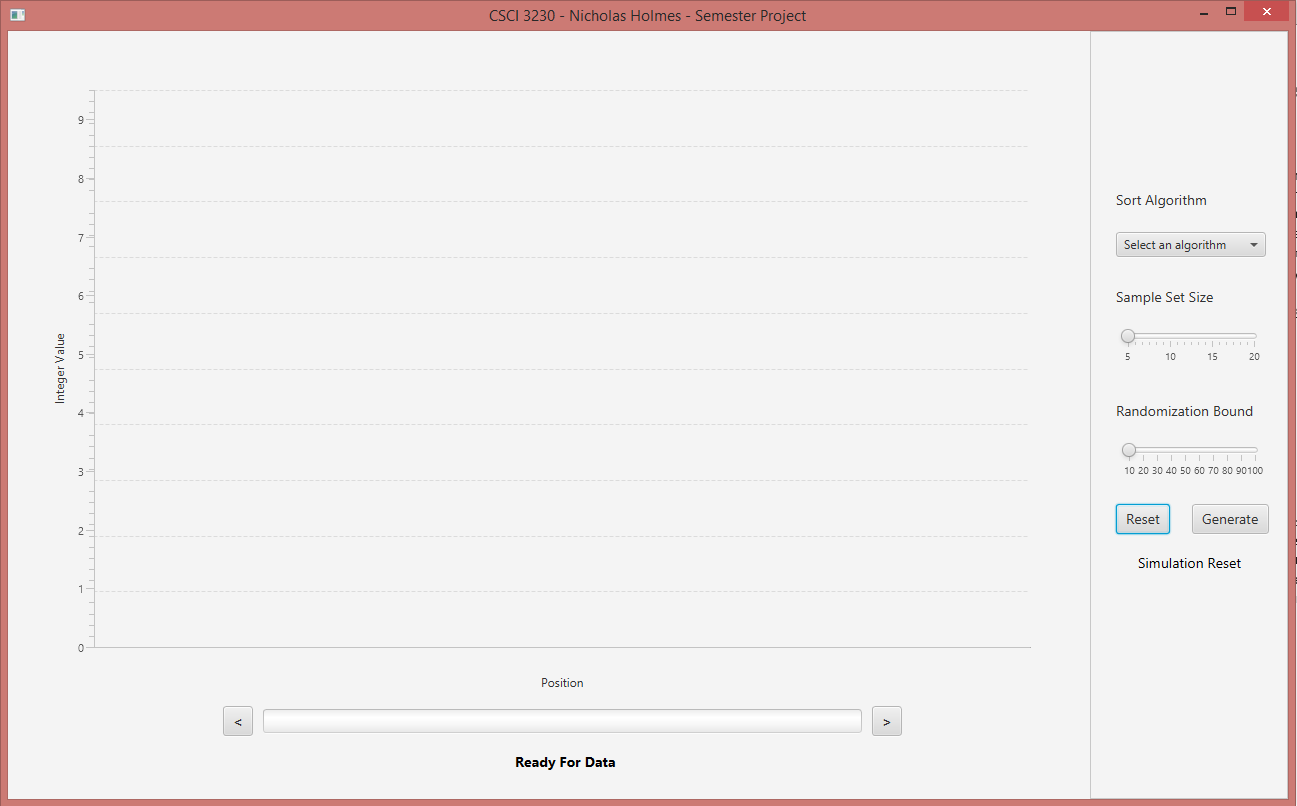
The goal of this project is to visually illustrate algorithms via a GUI. Given the requirements of the project, Java FX (as a part of Java SE8) was selected to interactively display a variety of sorting algorithms using a bar chart. To accomplish this an abstract programming model was created to allow the application to execute a sorting algorithm while simultaneously recording the states of the data while each step is executed. A Collection of these states (frames) indicates the movement of items in the data structure. These recorded data points can be displayed by a GUI in successive order.

**The following algorithms were selected for this project:**

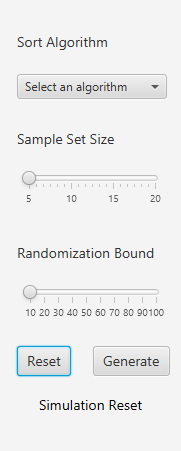
1. Selection Sort
2. Insertion Sort
3. Bubble Sort
4. Quick Sort
5. Merge Sort

**Instructions:**

To run this application, please extract the zip file to an empty directory and import the project into either an IntelliJ or Eclipse based IDE. The main entry point for this code is present in the directory ./src/base/Main.java. Running from this location will immediately start the application. This application was developed using the Java JDK version 1.8 and should be the only requirement for compatibility.



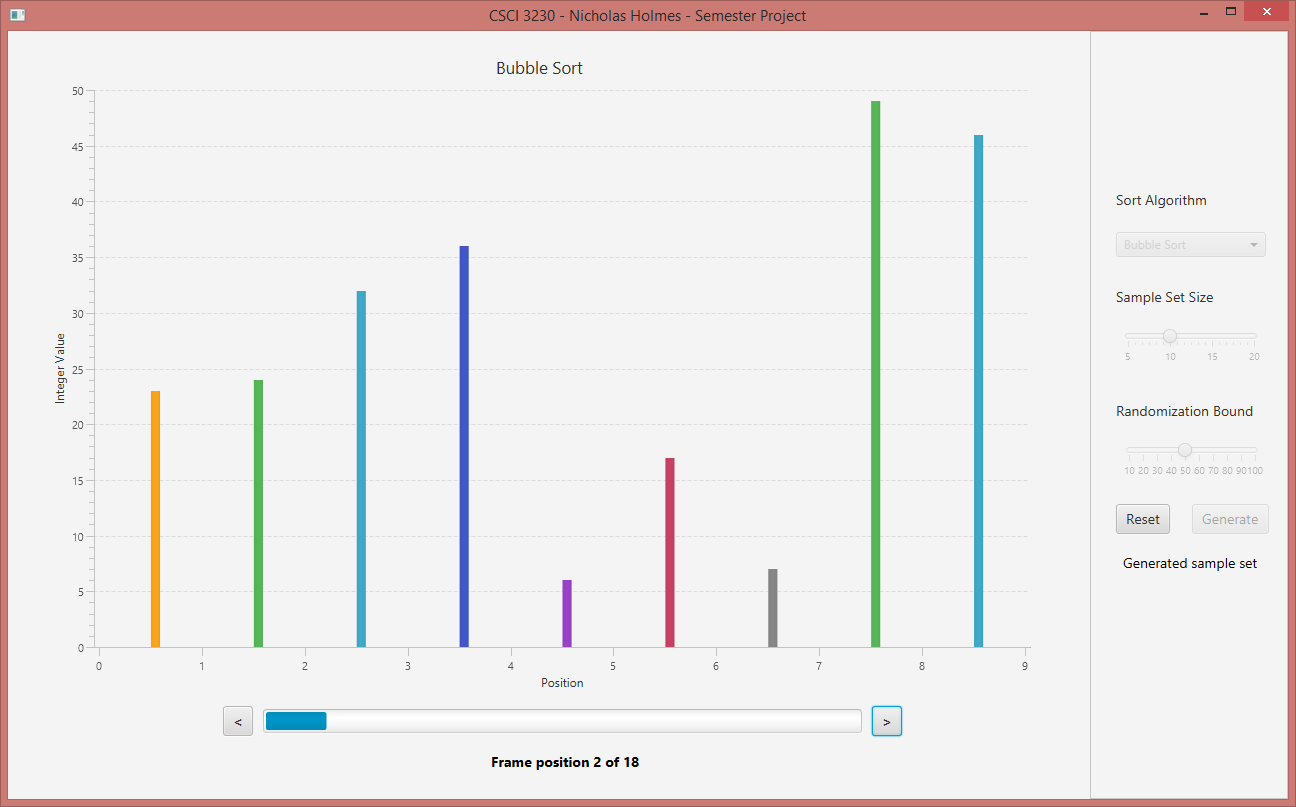
The screen shot above illustrates the initial state of the application after start up. A grid pane was used to split the GUI between the graph area on the left, and the options on the right.



The right pane, as shown in the close-up above shows the selectable options in detail. Use the sort algorithm drop down to select from one of the five selections. An algorithm is not selected by default, so a selection is required by the user to continue. If a selection is omitted a message will appear in the bottom area of the pane indicating that a selection is required.

The two sliders, sample set size and randomization bound, are available to adjust the size and complexity of the simulation. The minimum required settings are selected by default and can be used without adjustment. Sample set size increases the number of integer values in the total target set for sorting. Randomization bound allows you to select the upper bound of random numbers generated by the algorithm. A greater number will allow for a wider range of random integers.

The two buttons available are ‘Reset’ and ‘Generate’. Reset can be selected an any time to restart the simulation and revert all of the options and chart grid to the default state. Generate allows the user to capture the options selected and generated a simulation for evaluation.



The screen above illustrates the result of selecting a sample size of 10 and a random bound of 50. Once the generate button is clicked, all of the options in the right pane become disabled with the exception of the Reset button. A message is also displayed in the bottom of the right pane, indicating to the user that the generation of the simulation’s data is complete.

The type of algorithm is displayed as the title. The y-axis represents the integer value and the x-axis represents the position of elements in a collection. The area below displays a progress bar as well as previous and next buttons to advance to each frame. Click these buttons to observe each change resulting from the algorithm.

**Known Issues:**

* A bug related to Java FX causes the auto-ranging animation to fail to range the x and y axis on the initial step (frame 1). Click the next button causes the auto-ranging feature to refresh and display the correct results.
* Animation between frames temporarily causes the bar values for each position to rise up and show multiple colors. This seems to be the intended animation affect provided by JavaFX. An alternate fxml (./src/base/alternate.fxml) is provided for experimentation that simply disables all animations. It can replace gui.fxml to illustrate this.