User’s documentation

This program allows users to visualize sorting algorithms, and compare between them. For each step of chosen algorithm, it generates a svg picture showing a current progress of sorting.

After launching the program, the user is shown most common sorting algorithms to choose from. Multiple can be selected.
Next, the user chooses the size of an array, writing a number from 1 to the maximum size of an integer to the row labeled “Size of array:”.
Under that, there is a possibility of using multithreaded sorting. If the user checks the box to use multithreaded sorting, a new option appears for choosing how many threads should be sorting the array.
Then, the number of sorting iterations is set, in the row “Algorithm iterations:”. This number indicates, how many arrays are sorted.
After that, by clicking the button “Select folder” in the row “Folder:”, the user chooses a folder to save generated files to. After choosing a folder, the selected path appears on the button.
Finally, after clicking the button “Generate”, generating pictures starts. In case there is an error in input parameters, an error windows shows announcing the error. If everything is correct, the status of current process is shown on a label below the button.
In case of single threaded sorting, for every iteration, a new folder is created in selected folder, called AlgorithmVisualization extended by number of iteration in current run of the program. This folder contains svg images of sorted array, every image has the name of sorting algorithm extended by number of frame.
If the user chose to sort on multiple threads, fewer images are generated, only showing the array after the threads finished sorting, and the process of merging blocks sorted by threads.
The content of every image is an array visualized by a column graph, in which the height of a column corresponds to the size of element in an array. Each array is sorted ascending.
Below the graph, there always is the name of algorithm and current time, measured in ticks.
For multithreaded sorting, there also is number of threads sorting the array, and number of comparisons and accesses to the array for single threaded.
In addition to svg files, for each sorting round is generated one text file, summarizing speed of given algorithm on an array of given length.

After generating svg files, we can make a timelapse of sorting by converting them to png, and then using some tool to make a video from these pictures, such as ffmpeg with command:

```
ffmpeg -framerate 30 -i *%01d.png -f avi -vcodec msmpeg4v2 -q:v 2 -y out.avi
```

where “*” is a name of prefered sorting algorithm.