Assignment II

In this assignment, your task is to write a drawing application, using JavaScript and HTML5. It should use the HTML5 canvas element, and use object-oriented design for the JavaScript code. And you need to use JS objects instead of ES6 classes. A use case for your application could be for a teacher which wants to use your application instead of a regular whiteboard.

The application should be capable of the following:

1. (**40%**) it should be possible to add primitive drawing objects to the drawing, i.e.:
   - *circle*
   - *rectangle*
   - *line*
   - *text*
   - *pen (i.e. a freehand drawing)*

This should work similarly as in most drawing programs (MSPaint, Gimp etc.). The pen should be the default drawing object.

2. (**10%**) It should be able to manipulate various properties of the drawing objects, such as their color(s), linewidth, font etc.

3. (**10%**) the application should support undo and redo. It is sufficient that each object added to the drawing can be "undone" (and redone). A penstroke should be considered a single object, i.e. when that object is undone it should disappear completely.

4. (**10%**) all elements should be movable.

5. (**10%**) it should be possible to save and load a drawing (See below: an API is provided which makes this easier).

Also, the following will be considered in the final grade as well:

6. (**10%**) code quality (consistency in indentation, variable names, brace placements, whitespace usage etc., structure of the code (are there many global variables? Is the code split up into different files?)

7. (**10%**) usability: is it easy to use the application? Does it have sensible defaults?

Bonus points (which could bring the grade up to 12) will be awarded if the solution contains any of the following features:

- A more advanced undo/redo: such as when an object is moved, when the color is changed and more.

- Multiple move: The ability to move many objects simultaneously (i.e. select many objects first, then move them all together).

- The ability to group primitives (rect, circle, line etc.) together into a template, which can be saved and then added to a new whiteboard with ease (example: a teacher might want to create a "binary tree node" template, containing a circle with text inside, and two arrows pointing downwards from the circle)

- Math formulas, and symbols commonly used in math (such as Pi, the sum symbol etc.). You are free to experiment with the implementation of this, one possibility is to be able to write equations in some sort of a language which would then be converted to a drawing by the application (see example here).

Other features may come into consideration when bonus points are awarded. The question "how much do we have to implement to get 12?" can only be answered by: "It depends on the quality of the implementation".

A number of web-based drawing applications and tutorials are available online, including (but not limited to):

- An example from Opera
- SketchPad

You may find some of these to be helpful, but they are not necessarily doing exactly what we want. If you happen to stumble upon other examples you think might be useful, please allow other to enjoy your findings.
Using libraries to help with the implementation is allowed (such as **jQuery**, **Bootstrap** etc.)

How to save?

You should use *localStorage* which is a feature of HTML5 to store your templates. It is up to you to determine the best way to store these templates. Problems may arise on how the data should be parsed which is stored in *localStorage* and how it should be retrieved and translated into valid JavaScript objects again. But the methods used are entirely up to you.

https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

You can read the official document on how to operate *localStorage* from the link above.

Hand-in

Hand in a single archive containing all files necessary to run the application.