Peer Review 1

Usability of code

* Appropriate parameterisation including defaults
* Encapsulation (private fields where appropriate)
* Useful methods including draw

The code has appropriate getters and setters for individual parts of the drawing e.g. GetBackground(). The architect class its own draw and setup functions which are called from the master.js file. The constructor has its own variables, most of which are setup initially in the class. They are encapsulated within the class and are modified with getters and setters. Methods such as background() and frameRate() have proper parameterisation where the values passed in are setup within the constructor which can be modified.

The code does not have extra methods built in apart from the included ones in the p5 library. Methods such as the zoom function could have been created as a separate function to increase usability.

**Development of original**

* Original code to be submitted as well as final version
* Work done in refactoring code to class
* Work done in useful parameterisation
* Work done in extending scope

Original code has been converted from processing to Javascript. Both the original code and the new code was submitted. The code has now been put into its own class file on which an object is created within the master.js file. The scope has been extended to change the foreground, background, thickness of the line and its speed. The developer has added in the option to change those options using jQuery. Within jQuery, they have used proper parameters passed in to the setter functions within the architect class. As there are not any other functions apart from the getters and setters there is no useful parameterisation done

**Quality of example**

* HTML page is valid
* Appropriate on-page instructions
* Appropriate on-page controls (form)

HTML page is not valid, checking for errors gives the following: character encoding was not declared, Self-closing syntax (“/>”) used on <br /> tag and <select> on line 61 has no closing statement.

HTML page contains a working example. The webpage has a good description of what the user can do to interact with it, by detailing what each range of choices does. The on-page controls are done through jQuery and not HTML within the master.js file where the user can select from pre-determined options.

There is no DOM as the developer has used jQuery instead to get the values of the input forms. **Quality of documentation**

* All methods and parameters explained (including constructor)
* Explanation of example
* Source of initial code acknowledged (including licence)

No parameters are explained including the constructor for either of the two JavaScript files. The developer has instead listed the possible options they can choose from the inputs given on the webpage. The person has given general explanation of what happens with the code, however, go into how the line is generated and the way the sketch is drawn on the canvas with calculations. There are screenshots included of the example itself.

The documentation does include a link to the original openprocessing link. License file has been included but has not been acknowledged within the documentation.

**Code quality: ESLint**

Apply rules from [eslint.org/docs/rules/](https://eslint.org/docs/rules/):

* Possible Errors
* Best Practices
* Variables
* Stylistic Issues
* ECMAScript 6

Eslint produces 2 types of errors. There are many errors when it comes to indentation e.g. Expected indentation 8 spaces but found 4. Another error that comes up is ‘missing semicolon’ a couple of times. For example on line 73, in architect.js, resizeCanvas(this.width, this.height) is missing a semicolon. These are both stylistic issues. Everything else produces no errors for the other categories.