

✓ You finished this assignment

[Go to next item](#)

Grade received 80%

## Final project submission [002]

[Instructions](#)

1. Upload your completed project files as a zip. Please don't use any other compression file formats.

65 / 80 points

[final\\_OMJ.zip](#)

### Grading Rubric

How stable is your application?

0 points The application won't load at all.

1 point The application will load but might have required small adjustments by the marker to get it running.

2 points The application loads but crashes frequently or has major bugs stopping it from working as intended.

3 points The application may crash occasionally or have small bugs.

4 points The application might have some bugs or errors but mostly works as intended.

5 points The application works flawlessly and all functionality interacts as expected.

How usable is your application?

0 points The application is unusable, possibly because of stability issues.

1 point The application performs erratically, it's hard to complete tasks using it.

2 points The application has some confusing behaviour and is very inconsistent

3 points The application is mostly usable but may sometimes be inconsistent or features are hidden from the user.

4 points The application works consistently. It is easy to find the features and how they work. Perhaps including evidence the student has spoken to potential users.

5 points The applications design is based on formal ux principles and there is evidence of user evaluation.

How extensive are the modifications and extensions you have made to the app?

0 points The code has not been modified from the original template.

<input type="radio"/> <b>1 point</b>	Functionality does not substantially extend case-study templates.
<input type="radio"/> <b>2 points</b>	Several modification have been made but may be repetitive or not deviate from the example extensions.
<input type="radio"/> <b>3 points</b>	A good range of different features implemented, across enhanced versions of the provided extensions and / or ones of the student's invention.
<input type="radio"/> <b>4 points</b>	An extensive range features implemented with some novel ideas to enhance provided extensions and / or original extensions.
<input checked="" type="radio"/> <b>5 points</b>	An extensive range of highly original features implemented.

How challenging was the technical difficulty of your project?

<input type="radio"/> <b>0 points</b>	No attempt made to make modifications to the code.
<input type="radio"/> <b>1 point</b>	Very basic coding techniques, largely identical to case-study templates. Missing key techniques such as looping and arrays.
<input type="radio"/> <b>2 points</b>	Good use of many coding techniques covered during the course. Maybe doesn't utilise more complex techniques such as nested loops and complex conditionals.
<input type="radio"/> <b>3 points</b>	Very good use of the coding techniques shown in the course, Including object orientation.
<input type="radio"/> <b>4 points</b>	Most of the more complex techniques shown has been utilised. Effective object orientation to organise code logically.
<input checked="" type="radio"/> <b>5 points</b>	Appropriate application of advanced programming concepts beyond what is expected at this level.

Have you correctly declared variables and used variable scoping efficiently?

<input type="radio"/> <b>0 points</b>	No variables added to the template.
<input type="radio"/> <b>1 point</b>	Variables added to the template.
<input type="radio"/> <b>2 points</b>	Variables have been added both locally and globally scoped. However, some globals could have been implemented as local variables.
<input checked="" type="radio"/> <b>3 points</b>	Functions and methods generally use local variables instead of globals when appropriate
<input type="radio"/> <b>4 points</b>	Significant effort was made to modularise the functions so they do not depend on globals when they do not need to do so.
<input type="radio"/> <b>5 points</b>	All variables correctly scoped, using object orientation techniques. blocks. Students may show awareness of closures and ES6 variable creation.

Have you made good use of constructor functions in your project?

<input type="radio"/> <b>0 points</b>	No new constructor functions have been created.
---------------------------------------	---

<input type="radio"/> <b>1 point</b>	Some attempt to create new constructors but the syntax is confused or incomplete.
<input type="radio"/> <b>2 points</b>	Simple constructor functions have been used and structure of templates adhered to.
<input checked="" type="radio"/> <b>3 points</b>	Constructors have been used for new tools / visualisations.
<input type="radio"/> <b>4 points</b>	Constructors have been used extensively. This includes parameterised constructors.
<input type="radio"/> <b>5 points</b>	Advanced object management (e.g dynamically creating and destroying).

Have you made good use of object orientation methodology?	
<input type="radio"/> <b>0 points</b>	No attempt at object orientation.
<input type="radio"/> <b>1 point</b>	Simple object orientation has been applied but theory misunderstood or code doesn't work.
<input type="radio"/> <b>2 points</b>	Some object orientation has been applied to the design of the code. Mostly replicating template structure.
<input type="radio"/> <b>3 points</b>	Object orientation has been used for tools/visualisations. Other simple objects maybe created in the code.
<input checked="" type="radio"/> <b>4 points</b>	Objects have been created for a range of purposes, possibly including containers.
<input type="radio"/> <b>5 points</b>	Advanced object orientation. Beyond what has been shown in the module. This may include design patterns or ES6 techniques.

Have you made good use of object properties and correctly scoped them?	
<input type="radio"/> <b>0 points</b>	No object properties created.
<input type="radio"/> <b>1 point</b>	The majority of object-related data stored externally to the object.
<input type="radio"/> <b>2 points</b>	some object-related data stored as properties.
<input type="radio"/> <b>3 points</b>	Object-related data stored as properties. Public and private considered.
<input checked="" type="radio"/> <b>4 points</b>	Object-related data stored as properties. Good use of public and private, and the 'this' keyword.
<input type="radio"/> <b>5 points</b>	All object-related data stored as properties or local variables. Complex use of variables in anonymous functions.

Have you made good use of object methods?	
<input type="radio"/> <b>0 points</b>	No object methods created.
<input type="radio"/> <b>1 point</b>	Most functionality in functions outside of constructors or global space.

<input type="radio"/> <b>2 points</b>	Majority of object-related functionality implemented as methods. However, they may be simple or overly long.
<input type="radio"/> <b>3 points</b>	Well constructed methods, using returns and parameters. Private and public considered.
<input checked="" type="radio"/> <b>4 points</b>	All object functionality as methods. Private and public used appropriately. Methods well constructed.
<input type="radio"/> <b>5 points</b>	Expertly constructed methods, ensuring readability and reuse within the object. Advanced JS features maybe used.

Has your project incorporated arrays?

<input type="radio"/> <b>0 points</b>	No arrays created.
<input type="radio"/> <b>1 point</b>	Some array use maybe incorrect or inappropriately used.
<input type="radio"/> <b>2 points</b>	Simple arrays. Possibly only array literals used. Simple iteration of elements.
<input type="radio"/> <b>3 points</b>	Generative arrays created and used. length property used correctly. Perhaps simple use of array methods.
<input checked="" type="radio"/> <b>4 points</b>	Complex arrays (arrays of objects) have been used.
<input type="radio"/> <b>5 points</b>	Complex integration of array structures and / or other data structures such as trees, graphs, heaps and stacks. Advanced function use or enhancement of prototype object.

Has your project made use of loops?

<input type="radio"/> <b>0 points</b>	No loops used.
<input type="radio"/> <b>1 point</b>	Simple for loops. Perhaps incorrect or non-terminating.
<input type="radio"/> <b>2 points</b>	Correct use of single loops.
<input type="radio"/> <b>3 points</b>	More complex looping. Object access and method chaining with the iterator.
<input type="radio"/> <b>4 points</b>	Nested loops, with consideration for readability and modularity (i.e. use of additional functions).
<input checked="" type="radio"/> <b>5 points</b>	Advanced looping. Could include recursion or other techniques. Beyond what has been shown in the course.

Has your project made use of conditionals?

<input type="radio"/> <b>0 points</b>	No conditionals used.
<input type="radio"/> <b>1 point</b>	Simple, single condition. Perhaps incorrect or not functioning.

<input type="radio"/> <b>2 points</b>	Correct use of if and else. Single condition.
<input type="radio"/> <b>3 points</b>	Multiple conditions (i.e. collision detection).
<input type="radio"/> <b>4 points</b>	Complex structure of if, else if, and else.
<input checked="" type="radio"/> <b>5 points</b>	Advanced conditionals. Could include switch or ternary operator. Beyond what has been shown in the course.

Has your project been well commented?

<input type="radio"/> <b>0 points</b>	No comments added to the template code.
<input type="radio"/> <b>1 point</b>	Incomplete comments that don't improve readability of the code.
<input type="radio"/> <b>2 points</b>	Simple comments maybe not complete or clear what they mean.
<input checked="" type="radio"/> <b>3 points</b>	Good commenting, aids understanding and readability of the code. Maybe not as concise as needed.
<input type="radio"/> <b>4 points</b>	Excellent commenting. Makes tricky and important parts of the code clear and understandable.
<input type="radio"/> <b>5 points</b>	Extensive, clear, and concise comments throughout. Perhaps uses a consistent format or style.

Does your code consistently adhere to good coding style?

<input type="radio"/> <b>0 points</b>	What code has been written doesn't use consistent naming conventions or style.
<input type="radio"/> <b>1 point</b>	Code is badly used (e.g. long functions and poor variable name choices).
<input type="radio"/> <b>2 points</b>	Variable names are well chosen. Sometimes name choices become muddled or confusing between scopes.
<input checked="" type="radio"/> <b>3 points</b>	Good variable names with consistent style. Functions are good length and well considered with relevant input and output.
<input type="radio"/> <b>4 points</b>	Excellent choice of variable names, are consistent throughout the code. Functions have well considered parameters and their functionality is clear.
<input type="radio"/> <b>5 points</b>	Code is flawlessly written to a professional standard. There are no major linter errors and applies correct ES style and form.

Has the code been well organised?

<input type="radio"/> <b>0 points</b>	What code has been written isn't organised (i.e. all new code is written in sketch.js without consideration for indentation).
<input type="radio"/> <b>1 point</b>	Code is badly organised. Large blocks of commented out code maybe present.
<input type="radio"/> <b>2 points</b>	Code has been well organised, although lack of object orientation or helper functions means that there is some code duplication. Indenting and separation will not be thoughtfully altered for readability and consistency.

<input type="radio"/> <b>3 points</b>	Code is well organised and separated between objects. is mostly indented and formatted correctly.
---------------------------------------	---

<input checked="" type="radio"/> <b>4 points</b>	Excellent separation of code into units. Good consideration of code reuse. Code is indented and long running lines truncated as appropriate.
--	--

<input type="radio"/> <b>5 points</b>	Perfect organisation of code. It is highly readable and there is no code redundancy.
---------------------------------------	--

Is your code modular?

<input type="radio"/> <b>0 points</b>	Code isn't separated into separate files.
---------------------------------------	---

<input type="radio"/> <b>1 point</b>	An attempted at separating files. Not all code is correctly referenced in index.html
--------------------------------------	--

<input type="radio"/> <b>2 points</b>	Code is separated into files. One per constructor.
---------------------------------------	--

<input type="radio"/> <b>3 points</b>	Code is separated into files, one per constructor with additional objects in separate files also.
---------------------------------------	---

<input checked="" type="radio"/> <b>4 points</b>	Code is separated into files, one per constructor with additional objects in separate files also. Any helper functions are generically written to be used by other objects.
--	---

<input type="radio"/> <b>5 points</b>	Advanced file loading. Consideration of performance and ordering.
---------------------------------------	---

2. List the modifications and extension that you have made to the template file (400 words).

3 / 4 points

For each extension your answer should include:

- What is the function of the extension?
- How does the code fit into the templates design?
- How have you structured the extension's code?

I have added the following extension:  
 Rectangle Tool: It draws a rectangle on the canvas. It has three modes. The first mode draws on top of anything on the canvas with no fill, second mode overwrites the canvas with no fill, third mode draws on the canvas with fill color. The code makes use of the populate method to list its three modes. The code is structured to dynamically work based on mode selected.

Bucket Fill: Fills an enclosed shape found on the canvas by using p5 pixels array depending on the position of the user mouse. The extension was added using the toolbox and the extension uses class constructors with all the methods defined in the constructor function.

Text Tool: Text tool allows writing on the canvas while changing the text font and the text size. It also allows writing to a specific pixel on the canvas. The code makes use of its draw function like other extensions to render the text; the constructor function in the class calls the method available within the class for its operation. It has no history option due to its action within the draw function. The noHistory has been added to toolbox and mouseReleased for proper handling.

Zoom Tool: The Zoom tool can be used to increase view width and view height of any selected area on the canvas. It also uses class as its code structure and all necessary functions required by toolbox are defined within the constructor function.

Layer History: This allows for easy versioning of the application, user can easily go back to previous state of their drawing.

application. The code has been fit in with the helper function of the template, and the code structure is related to other functions in the helper function constructor

Offline Capability: The application can now be used without losing the state of the application either on reload or system restarts, it automatically saves the state of the application in local storage. This is structured to work asynchronously so as not to interrupt user actions, the code is also implemented with helper functions

Canvas Image: This allows uploading an image as background of canvas and also resizing the image approximately to the image size. The code has been structured using class and it fits into the template just like color pallet is.

Eraser Tool: This can be used to clean/erase the canvas

#### Grading Rubric

How well have you documented your project?

<input type="radio"/> <b>0 points</b>	No documentation provided.
<input type="radio"/> <b>1 point</b>	Scant description of changes made to template.
<input type="radio"/> <b>2 points</b>	Changes listed with little detail or technical description.
<input checked="" type="radio"/> <b>3 points</b>	Good description of the changes made. Some discussion of technical challenge.
<input type="radio"/> <b>4 points</b>	Descriptions are very clear and well annotated with diagrams. Detailed discussion of techniques used.

3. Describe how effective your plan was in completing your project (250 words).

2 / 4 points

Your answer should include:

- How well did you stick to your schedule?
- Did you divide up the task and the time effectively?
- Did you have an unexpected difficulties or challenges?

I encountered an unforeseen difficulties when trying to make the application compatible. I didn't stick to my initial schedule in the first four weeks(14-18) because the extension implementation were more challenging than expected. I agreed to completing one extension before moving to another. I examine the complexity of my first extension and time taken, then I used the result to divide the task and time effectively. This later works for me

#### Grading Rubric

How effectively have you reflected on your project plan?

<input type="radio"/> <b>0 points</b>	No reflection provided.
<input type="radio"/> <b>1 point</b>	Simple restating of plan with limited critique.
<input checked="" type="radio"/> <b>2 points</b>	Reflection on plan and where it didn't work. Not well evidenced.
<input type="radio"/> <b>3 points</b>	Good description of how the plan aided the project process and where improvements could have been made.

4 points

EXCELLENT ANALYSIS OF THE PLAN INCLUDING DEMONSTRATING HOW REQUIREMENTS WOULD ENHANCE OVERALL PROJECT OUTCOMES.

4. Evaluate the process of completing the project and how effective the final product is. (250 words)

4 / 4 points

Your answer should include:

**Self evaluation of the process of completing the project.**

- You may find it helpful to reflect on your plan and think about what you might do differently next time.

**Have you performed any system testing or user testing on your application?**

- What errors did you uncover?
- How did users respond to the application?
- How would you rectify these in a future version of the application?

Next time, I'll work on understanding more about algorithms and data structure before working on project like this, because I was forced to learn algorithms like flood fill, heap sort, trees, binary search and even array structure( having an object as elements of an array) I sent out a survey to colleagues and here's the [Survey Response](#). First impression was that the tools aren't useful because the tester do not understand how to use it. As a response to the feedback, I add description to each tool.

**Grading Rubric**

How effectively have you evaluated your project work?

<input type="radio"/> 0 points	No evaluation provided.
<input type="radio"/> 1 point	Limited reflection on experience of project. Simple statements on success and failure.
<input type="radio"/> 2 points	Personal reflections backed up by examples in the code and linked to original intentions for the project.
<input type="radio"/> 3 points	Personal reflections alongside evidence of simple testing process where needed to fix bugs or complex points of interaction.
<input checked="" type="radio"/> 4 points	Evidence of involved testing plan, either system or user. Personal reflections are thoughtful and balanced.

5. List any external sources that you have actively utilised in your project.

Your answer should include:

- any code you have used from external sources verbatim
- any code where you have taken inspiration from although adapted and refined for the project (such as pseudocode algorithms or code pens)
- any online help forums you have taken code from (i.e. StackOverflow or library documentation)
- any third-party libraries you have used.

You do not need to include everything you have read or that has helped you. Only where you have used or adapted code that appears in your project.

This can be a revised version the answer you provided for the midterm.

**There is no word limit for this answer.**

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)

[https://en.wikipedia.org/wiki/Flood\\_fill](https://en.wikipedia.org/wiki/Flood_fill)

<https://stackoverflow.com/questions/5623838/rgb-to-hex-and-hex-to-rgb>

<https://editor.p5js.org/micuat/sketches/xzRtK385>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)

6. Upload your progress log files as a PDF. Please don't use any other compression file formats.

6 / 8 points

[progress%20log%20omj.pdf](#)

**Grading Rubric**

Have you provided detailed logs of your project's progress?

0 points No logs uploaded.

1 point Some logs but little detail.

2 points All logs are likely to be present with details on progress through topic.

3 points All logs have been completed with topic progress and reflections on previous logs targets.

4 points Excellently maintained logs, considering workload and progress towards project goals.

Quality of written work including spelling and grammar.

0 points Report is lacking in quantity. Very poor spelling and grammar. One or more report sections is over word limit.

1 point Writing is mostly incomprehensible.

2 points Mistakes make understanding the report difficult. Word choice is poor or incorrect.

3 points A well written report. It is easy to read and understand the students achievements.

4 points Flawlessly written. Excellent spelling and grammar. Appropriate academic tone is taken throughout.

Final comments on your project from the marker:

You've made a very good range of extensions and well done using ES6 features. I found some difficult to follow without comments e.g. rectangleTool.js- have another look and see what you think. This may be personal preference, but I'd probably put history functions in a history (rather than 'helpers') file. It would be ideal to move more global variables into constructors/classes. Consider using fuller variable names. As an aside, consider whether you'd prefer: this.noHistory = true; or perhaps this.hasHistory = false; You do give good insights in your written answers. You should use the full word count to take answers further. Congratulations, you've done well. It may be useful to review your code with an eye to where you could improve structure and readability, for your development.