				Pass			Outstanding
Criterium	**%**	**0 Marks**	**1 Mark**	**2 Mark**	**3 marks**	**4 marks**	**5 marks**
Implementation Techniques	**60**						
Application stability	5	The application won't load at all	The application will load but might have required small adjustments by the marker to get it running.	The application loads but crashes frequently or has major bugs stopping it from working as intended	The application may crash occasionally or have small bugs	The application might have some bugs or errors but mostly works as intended	The application works flawlessly and all functionality interacts as expected
Application Usability	5	The application is unusable. Possibly because of stability issues.	The application performs erratically, it's hard to complete tasks using it.	The application has some confusing behaviour and is very inconsistent	The application is mostly usable but may sometimes be inconsistent or features are hidden from the user.	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.	The applications design is based on formal ux principles and there is evidence of user evaluation.
Range of extensions and modifications made	5	The code has not been modified from the original template.	Functionality does not substantially extend case-study templates.	Several modification have been made but may be repetitive or not deviate from the example extensions.	Good range of different features implemented, across enhanced versions of the provided extensions and / or ones of the student's invention.	Extensive range features implemented with some novel ideas to enhance provided extensions and / or original extensions.	Extensive range of highly original features implemented
Technical Complexity	5	No attempt made to make modifications to the code	Very basic coding techniques largely identical to case-study templates. Missing key techniques such as looping and arrays.	good use of many coding techniques covered during the course. Maybe doesn't utilise more complex techniques such as nested loops and complex conditionals	Very good use of the coding techniques shown in the course, Including object orientation.	Most of the more complex techniques shown has been utilised. Effective object Orientation to organise code logically.	Appropriate application of advanced programming concepts beyond what is expected at this level.
Variables	5	No variables added to the template	Variables added to the template	Variables have been added both locally and globally scoped. However, some globals could have been implemented as local variables.	Functions and methods generally use local variables instead of globals when appropriate	Significant effort was made to modularise the functions so they do not depend on globals when they do not need to do so.	All variables correctly scoped, using object orientation techniques. blocks. Students may show awareness of closures and ES6 variable creation.
Constructor functions	5	No new constructor functions have been created	Some attempt to create new constructors but the syntax is confused or incomplete.	Simple constructor functions have been used and structure of templates adhered to.	Constructors have been used for new tools / visualisations.	Constructors have been used extensively. This includes parameterised constructors.	Advanced object management e.g dynamically creating and destroying
Use of Object Orientation methodology	5	No attempt at object orientation	Simple object orientation has been applied but theory misunderstood or code doesn't work.	Some object orientation has been applied to the design of the code. Mostly replicating template structure.	Object orientation has been used for tools / visualisations. Other simple objects maybe created in the code.	Objects have been created for a range of purposes, possibly including containers.	Advanced object orientation. Beyond what has been shown in the module. This may include design patterns or ES6 techniques.
Object property use and scoping	5	No object properties created	The majority of object-related data stored externally to the object	Some object-related data stored as properties.	Object Related data stored as properties. Public and private considered	Object Related data stored as properties. Good use of public and private, and the this keyword.	All object-related data stored as properties or local variables. Complex use of variables in anonymous functions.
Object Method use	5	No object methods created	Most functionality in functions outside of constructors or global space.	Majority of object-related functionality implemented as methods. However, they may be simple or overly long.	Correctly constructed methods, using returns and parameters. private and public considered.	All object functionality as methods. Private and public used appropriately. methods well constructed.	Expertly constructed methods, ensuring readability and reuse within the object. advanced JS features maybe used.
Arrays	5	No arrays created	Some array use maybe incorrect or inappropriately used.	Simple arrays. Possibly only array literals used. Simple iteration of elements.	Generative arrays created and used. length property used correctly. Perhaps simple use of array methods.	Complex arrays, (arrays of objects) have been used.	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.
Looping	5	No loops used	Simple for loops. Perhaps incorrect or non-terminating.	Correct use of single loops.	More complex looping. Object access and method chaining with the iterator.	Nested loops, with consideration for readability and modularity i.e. use of additional functions	Advanced looping. Could include recursion or other techniques. Beyond what has been shown in the course.

				Pass			Outstanding
Criterium	**%**	**0 Marks**	**1 Mark**	**2 Mark**	**3 marks**	**4 marks**	**5 marks**
Conditionals	5	No conditionals used	Simple, single condition. Perhaps incorrect or not functioning.	Correct use of if and else. Single condition.	Multiple conditions i.e. collision detection.	Complex structure of if, else if, and else.	Advanced conditionals. Could include switch or ternary operator. Beyond what has been shown in the course.
Implementation Quality	**20**						
Commenting	5	No comments added to the template code.	Incomplete comments that don't improve readability of the code.	Simple comments maybe not complete or clear what they mean.	Good commenting, aids understanding and readability of the code. Maybe not as concise as needed.	Excellent commenting. Makes tricky and important parts of the code clear and understandable.	Extensive, clear, and concise comments throughout. Perhaps uses a consistent format or style.
Code Style and consistency	5	What code has been written doesn't use consistent naming conventions or style.	Code is badly used e.g. long functions and poor variable name choices.	Variable names are well chosen. Sometimes name choices become muddled or confusing between scopes.	Good variable names with consistent style. Functions are good length and well considered with relevant input and output.	Excellent choice of variable names, are consistent throughout the code. Functions have well considered parameters and their functionality is clear.	Code is flawlessly written to a professional standard. There are no major linter errors and applies correct ES style and form.
Code organisation	5	What code has been written isn't organised i.e. all new code is written in sketch.js without consideration for indentation.	Code is badly organised. Large blocks of commented out code maybe present.	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.	Code is well organised and separated between objects. is mostly indented and formatted correctly.	Excellent separation of code into units. Good consideration of code reuse. Code is indented and long running lines truncated as appropriate.	Perfect organisation of code. It is highly readable and there is no code redundancy.
Modularity	5	Code isn't separated into separate files.	An attempted at separating files. Not all code is correctly referenced in index.html	Code is separated into files. One per constructor.	Code is separated into files, one per constructor with additional objects in separate files also.	Code is separated into files, one per constructor with additional objects in seperate files also. Any helper functions are generically written to be used by other objects.	Advanced file loading. Consideration of performance and ordering.
Report	**20**						
Documentation of work done	4	No documentation provided.	Scant description of changes made to template.	Changes listed with little detail or technical description.	Good description of the changes made. Some discussion of technical challenge.	Descriptions are very clear and well annotated with diagrams. Detailed discussion of techniques used.	
Reflection on planning	4	No reflection provided.	Simple restating of plan with limited critique.	Reflection on plan and where it didn't work. Not well evidenced.	Good description of how the plan aided the project process and where improvements could have been made.	Excellent analysis of the plan including demonstrating how refinements would enhance overall project outcomes.	
Evaluation	4	No evaluation provided.	Limited reflection on experience of project. Simple statements on success and failure.	Personal reflections backed up by examples in the code and linked to original intentions for the project.	Personal reflections alongside evidence of simple testing process where needed to fix bugs or complex points of interaction.	Evidence of involved testing plan, either system or user. Personal reflections are thoughtful and balanced.	
Logs	4	No logs uploaded.	Some logs little detail.	All logs are likely to be present with details on progress through topic.	All logs have been completed with topic progress and reflections on previous logs targets.	Excellently maintained logs, considering workload and progress towards project goals.	
Quality of writing including spelling and grammar	4	Report is lacking in quantity. Very poor spelling and grammar. One or more report sections is over word limit.	Writing is mostly incomprehensible.	Mistakes make understanding the report difficult. Word choice is poor or incorrect.	A well written report. It is easy to read and understand the students achievements.	Flawlessly written. Excellent spelling and grammar. Appropriate academic tone is taken throughout.	