COMP 2659 Project
High Level Marking Guide

Note: the final project mark is subject to the course late policy as described in the course outline. To avoid a 20% late penalty, projects must be submitted by 11:59pm on Friday, April 11. Projects submitted later than 11:59pm on Sunday, April 13 will not be marked.

Basic Functionality Mark By Stages Completed

Completed	Letter Grade Range	Percentage Range
Less than Stage 2	F	0 – 19
end of Stage 2		20 – 29
end of Stage 3		30 – 39
end of Stage 4a		40 – 49
end of Stage 5	D	50 – 54
Stage 6 with inefficient redrawing	D+	55 – 59
end of Stage 6 with efficient, minimal redrawing (i.e. 4b)	C-	60 – 62
end of Stage 7	C	63 – 66
end of Stage 8	C+	67 – 69
end of Stage 9a	B- to B	70 – 76
end of Stage 9b	B+ to A-	77 – 84
some or all of Stage 10 note: stages 10a, 10b and 10c can each be attempted independently of the others. Each is worth approximately 5%	A to A+	85+

Quality Modifiers

A set of three "modifiers" will be applied to the basic functionality mark, subject to various quality assessments. For example, if the basic mark was 80 and a +5% modifier was applied, the bonus would be 4 (80 × 0.05 = 4). Note that after applying all bonuses, it is possible to score higher than 100 (e.g. earning three bonuses of 4 on a basic mark of 90 would give an overall project mark of 102).

- Design modifier
    - superior quality = up to +5% of basic mark
        - source code organized into various appropriate file modules/libraries (categorized by functionality, game-dependent versus game-independent code clearly separated, high-level versus low-level support libraries clearly separated)
        - appropriate use of header files
        - appropriate use of make file

        - modular decomposition (functions and modules are cohesive and each have one clear purpose; minimal coupling between functions and modules; each function delegates to sub-functions as appropriate; all code for each function fits on the screen at one time)
    - satisfactory quality = +0% of basic mark
        - mostly as above, with a small number of minor problems
    - inferior quality = up to -5% of basic mark
        - more than minor problems
- Code readability modifier
    - superior quality = up to +5% of basic mark
        - identifiers are self-documenting
        - named constants are used in place of hard-coded literals, where appropriate
        - source code is formatted for readability, including consistent and clear use of white space (e.g. indentation)
        - header and inline comments are used sparingly, but appropriately, to highlight important information and to clarify non-obvious details
        - stage 1 game specification document is complete, up-to-date, and accurately describes the final version of the project
    - satisfactory quality = +0% of basic mark
        - mostly as above, with a small number of minor problems
    - inferior quality = up to -5% of basic mark
        - more than minor problems
- Game professionalism modifier
    - superior quality = up to +5% of basic mark
        - game play is well designed and complete
        - no noticeable latency in event handling
        - graphics are of high quality, and animation is smooth and steady
        - music and sound effects are of high quality, and playback is smooth and steady
        - overall, the game has a finished and professional look and feel
        - attention to detail is apparent