Major Project Marking Guide

Code (25 marks) In each category below higher marks will be awarded to teams that demonstrate ~~correct metadata~~, appropriate use if comments, descriptive variable names and efficient code.	Marks
Some correct coding	0 - 5
Working program but does not meet basic requirements outlined in sprint	6 - 10
Meet basic requirements of initial sprint 1 (text based game)	11 - 15
Meets requirements of second sprint 2 (basic graphical rendering)	16 - 20 19 - no metadata
Meets requirements of third sprint 3 (graphical game)	21 - 25
Other project components (50 marks)	
Project Planning (evidence of planning of your project using tools in GitHub. Higher marks will be awarded to teams that show evidence of detailed planning and task allocation) basic use of planning tools	0 - 5 3
Design brief (higher marks will be awarded to those teams that 'identify the problem' and a description of their proposed solution including explicit outcomes v basic identification of the problem only	0 - 5 2
Storyboard ('screenshots' of the game at each important phase of play in logical order) none	0 - 5 0
Data flow diagram (should show processes, data flows, external entities and data stores) not a DFD - see course specs	0 - 5 2
Algorithm design (flowchart or pseudocode: higher marks will be awarded to those algorithms that show all terminators, processes, inputs and outputs, subprograms and decisions using correct symbols [flowchart] or syntax [pseudocode])	0 -15 12
Data dictionary (higher marks will be awarded for data dictionaries that give correct descriptions of all variables used in the program. Teams using classes need not include class attributes.) not a properly formatted data dictionary - see course specs and example below	0 - 10 5
User manual (higher marks will be awarded to manuals that describe the purpose of the game and give detailed and logical instructions for playing. The user manual should also include system requirements and licencing)	0 - 5 0

not found

24/50

the internal documentation (comments and variable names) were good but your comments should not say what you are doing (assume that the person reading your code knows basic python) but say how and why it is doing it. sadly no metadata or it would have been 20 for your code.

Data dictionary example:

Item	type	Description	example	validation
levelOne	int	difficulty level	1	0< n < 5

24 + 19 = 43/75 = 57%