

Section 8 - Evaluation

The WJEC specification says:

- a) Produce an evaluation of the programming language used to create the solution.
- b) Compare the solution with similar commercially available systems.
- c) Identify the successful features of the system and make specific suggestions for improving less successful areas of the system.
- d) Describe the strengths and weaknesses of own performance in the design and prototyping of the solution.
- e) Describe changes of approach that would be adopted to solve a similar problem.

Things to consider doing:

- i) Put a heading, **8a Evaluation of the programming language used.**
- ii) You should make comments on your choice of programming language. How did you find it? What libraries did you use and how easy was it to use them? When you got stuck, what were the help facilities like? What IDE did you use? How did you find it? How clear was it? What options for setting up the IDE were available? What tools were available on the menu? What were the debugging tools like? What were the documentation and auto-documentation tools like? How did you find the error messages you got? When you didn't understand an error message, how easy was it to find out what it meant? What visualisation tools did you use? How did you find them? The trick here is to use lots of 'For example' whenever you discuss something. Give real, specific examples from your own experiences to back-up your assertions.
- iii) Put a heading, **8b Comparison of the system with commercially available systems.**
- iv) If you have done a project that is unique, your teacher will advise you what to do. Otherwise, can you identify similar problems and see how they were solved? For example, can you find a game with some input and output and see how they handled that? If you are writing an adventure or dungeons and dragons game, can you find the code for a working game and have a look at it? How did they store characters? how did they keep track of each character's attributes? How did they actually get two characters to 'fight'. You can learn a lot by comparing your solutions to similar problems and solutions.
- v) Put a heading, **8c Successful features and suggested improvements.**
- vi) You can use the results of your test plans, user testing and user feedback to help you identify the most successful parts of your solution as well as some ideas for the areas that could be improved.
- vii) Put a heading, **8d Strengths and weaknesses of my personal performance.**
- viii) You need to evaluate your own performance doing the project. Did you follow the all of the advice you were given in the 'Before the start ...' document? Were you organised? Did you back-up work properly and use version control? Did you use a template to write up every section of your work? Did you identify what information you needed and plan interviews appropriately? Did you record and write-up interviews well? How do you know? Did you select a 'good' project which was achievable? Did your prototypes really help you and inform the development of the final solution? How do you know? How did you manage time, to ensure that you both completed the project on time, and to ensure that the project wasn't so big that it used up far too much of your free time? How did you check the quality of your written work? Did you make use of proof-reading? Again, the trick here is to use lots of 'For example' whenever you discuss something. Give real, specific examples from your own experiences to back-up your assertions.
- ix) Put a heading, **8e Changes of approach for future projects.**

x) Thinking about your performance and how your project went from start to finish, what would you do differently and why? Be specific. Give examples.