Candidate number: as_1051882_034643.pdf

Summary

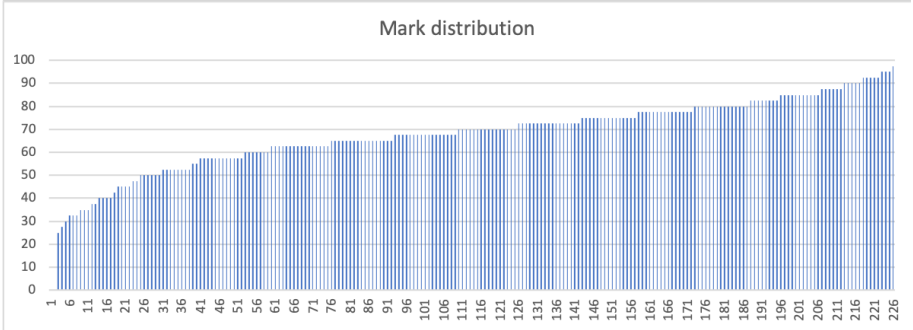
Mark /40	Total %
27	67.5

Feedback

" - Section 1 - create wordseacrh board - Highlight: The structure of the code Improvement: the use of proprocessing on the words.txt to get rid of capital letter etc alongside expanding the logic for the directions" " - Section 2 - check answers - Highlight: Again the structure of the work is really nice Improvement: expand upon the logic using the indexs to achieve the check which would also allow to check computational effecetivness. " " - Section 3 - User interface design - Highlight: the formatting of the matrix is really nice Improvement: -" " - Section 4 - Game play and project delivery - Highlight: The gameplay logic for the program Improvement: The inclusion of difficulty settings (letter frequency functionality) and the logging testing etc "

All class feedback

Overall this assignment was completed to a good standard and the flowcharts and pseudo code demonstrated a good understanding of problem solving with algorithm design. Creating the wordsearch board was the main strength for the majority of people and the detail in this section tended to be good with good use of looping and control statements. The main features that distinguished more advanced submissions were in the consideration of details such as how to handle intersections between words and adding difficulty levels. One feature very few people addressed was keeping a record of errors, for example making a record when words weren't able to be placed. This type of log information can help debug algorithms when there are errors. In summary, this assignment reflected a successful first step into programming. The mark distribution below shows the spread of marks for the entire class.



Mark breakdown: CA1 breakdown

Create word search boards	7 **/10**
Check answers	5 **/10**
User interface	10 **/10**
Project logic	5 **/10**
Total	27 **/40**

Marking criteria: CA1

Each of the four sections has a potential mark of 10 and the assignment as a whole has a potential mark of 40. Functionality marks will only be awarded if the pseudo code and flowchart are well defined and show the same logic.

- **Creating the word search board** [10 marks]
  - Is there a function called `create word search` that creates a letter matrix - 2 mark
  - Does the function return a matrix with the suitable dimentions containing randomised letters - 2 marks
  - Does the function use the words.txt file effectively - 2 marks
  - Does the board effectively embed words in all directions - 4 marks
- **Check answers against word search board** [10 marks]
  - Is there a function called check answer that could check an answer - 2 marks
  - Does the function use indexing correctly to check the corresponding board letters - 2 marks
  - Does the function use the words.txt content to check if the word is a correct word - 2 marks
  - Is the answer checking done in a computationally efficient manner to optimise game play - 4 marks
- **User interface** [10 marks]
  - Does the interface start make appropriate use of user inputs - 2 marks
  - Is the word search board well formatted - 2 marks
  - Is the user input managed through functions and filtered for the keywords - 4 marks
  - Is the time correctly recorded and returned to the user at the end of the game - 2 marks
- **Gameplay logic and project delivery** [10 marks]
  - Does the game logic enable gameplay from start to finish - 3 marks
  - Is the gameplay adjusted to make the game difficult by, for example, making randomness match letter frequency in natural language and embedded more words - 4 marks
  - Is the software reliable (through testing), fault tolerant (through effective exception handling and input validation), accountable when errors occur (through logging) - 3 marks