# Hangman Project

We are going to make a hangman game where players can play against another player or the computer

# Success criteria

1. Show a menu
2. Get a choice from the user
3. Check if the choice made is a valid choice
4. If it is PVP then it should ask for an input and store it in a variable called ‘word’
5. It should check the input and see if it is only letters and more than one letter
6. It should then force the letters to be uppercase
7. If it is PVE then it should find a random word from a stored array of words
8. It should then display a façade using a different string on top of it that is the same length as the word.
9. It should then print for player 1 to leave the computer and for player 2 to come on
10. It should print the letters that have already been guessed and what letters have not been guessed [a to z array]
11. It should then take an input asking for a letter to be guessed
12. It should check if it is only letters and not numbers
13. If not then it should ask the player to re enter while displaying the previous guess, to not make confusion
14. It will then force the letters to be uppercase
15. It should check if the input is a single letter or a whole word
16. If the letter is not in the a to z array then it should also ask for the player to re enter their guess with the message “you have already guessed this word”
17. If it is not the whole word but it is more than one letter it should print that it is not the correct word and draw a piece of the hangman
18. If it is a single letter but the letter does not appear then it should draw a piece of the hangman. It should also remove the letter from the array that has the letters a to z
19. If it is the correct word then it should print the word by removing the façade and it should then print a message congratulating the player and stating that player 2 has won the game
20. If it is a correct letter then it should print the letter by removing the parts of the façade where the letter is in the string ‘word’ and then print that it was a correct letter
21. It should remove the letter from the a to z array so it cannot be guessed again
22. It should check if the hangman is complete and if it is then it should print that player 1 has won the game
23. It should check if the word is complete and if any of the façade is left. If not then it should print that player 2 has won the game
24. If 22 or 23 are true then it should ask them if they would like to play again
25. If they choose to it should show them the menu again
26. If they choose not to it should end the program
27. It should repeat from criteria 10

# Data structures

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Data | Variable type | Name [if applicable] |
| 1 | Menu as a string | String | displayMenu |
| 2 | Choice as a string or char | String/char | menuChoice |
| 4 | Word chosen by player 1 [if chosen] | String | guessableWord |
| 7 | Array or text file of words that can be chosen | Array/list/.txt | vComputerWords |
| 7 | Word chosen from vComputerWords for player to guess | String | guessableWord |
| 8 | Fake string with the same length as guessableWord | String | wordFacade |
| 10 | Array with letters a to z | Array | Letters |
| 11 | Guess made by player | String | playerGuess |
| 11 | If the game has been won [Boolean] | Boolean | player2Win |
| 17 | Counter for hangman | Integer | attempts |