|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayFloatingIcons | Prints characters in a single line format. The occurrences of the desired character to be printed can be manipulated using the spacing, and how long the characters will be printed can be manipulated using its max length | nMax – length of the characters to be printed  cIcon – the character to be printed every nSpacing  nSpacing – the nth character where the cIcon will be printed every time from the start or last print | Displays a single line of a specified format of character that is printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayHeaderUnderline | Prints an underline for a header with unique character design | None | Prints an underline |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayScanDivider | Prints and displays a divider at the bottom part of the current display and it notifies user that user needs to enter a character or value | None | Prints a bottom divider and scanning to user |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayUpperUniversalChoiceBox | Displays the upper portion/border of the choice box | None | Prints upper part of choice box |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayLowerUniversalChoiceBox | Displays the lower portion/border of the choice box | None | Prints lower part of choice box |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| getSingleChar | Gets the single character input of user and ensures that it takes up the newline character to avoid being stored to other variables while it is in the input buffer | cChar – the address of the character choice of the user | \*cChar – updated value of the character choice of user |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| getString | Gets the string input user of the user in accordance to the required string length | sString – the address where the input string will be stored  nLength – string length needed  \*cCancelChar – cancel chararacter for cancelation of string input in string format | \*sString – the updated string content |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| generateRandomNum | Generates a random number from 0 up to the given maximum limit | nMaxLim – max limit of range of values to be randomly generated | nRandNum – generated random number from the given range |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| delay | Pauses the program with the given number of seconds | nSeconds – time in seconds on how long the program should pause | Pauses program for nSeconds |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| getWordEntry | Gets the word entry for a specific chosen scenario (i.e. Add word, Play, etc.) until player enters a valid or input cancelation. | sChosenWord – where the entered string will be stored  bValid – the address of the Boolean indicator if string input is valid or an input cancelation  \*sEntryFor – indicator for what type of word input the string for | sChosenWord – updated string content |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| isUniqueLetter | Checks if generated random entry is unique and not yet used | gameboardRow – the current row of the gameboard being checked. (1D array)  nCols – current number of columns that have been initialized in the gameboard  nRandNum – the generated random number for the index of aEntries  aEntries – the list of entries | 1 – if unique and not yet used  0 – if not unique or entry is already in use |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| checkLetters Left | Checks the letters left in the current list of aEntries that are not yet used in the gameboard | aEntries – the list of entries  nElem – number of elements in aEntries | nLettersLeft – total number of letters left in the aEntries that are not yet used in the gameboard |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| printOuterGameboard | Prints the outer(upper and lower) borders of the gameboard display | nCol – number of columns for the current game session  cLeft – character to be printed on the left side of the border  cRight – character to be printed on the right side of the border | Prints the outer border of the gameboard |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| printGameboardDivi | Prints the dividers inside the gameboard | nCol – number of columns for the current game session | Prints gameboard divider |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| printGameboardContent | Prints the contents of the gameboard | aGameLetters – the array of contents/characters of the gameboard per row  nCol – number of columns for the current game session | Prints content of gameboard |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| printGameboard | Prints the whole gameboard display with current row indicator | aGameLetters – the array matrix of gameboard letters  nRow – number of rows for the current game session  nCol – number of columns for the current game session  nCurrentRow – current row index of the gameboard that the player is answering | Prints whole gameboard with current row indicator |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| checkGameboard | Checks if letter input of player exists in the current row of the gameboard | gameboardRow – the contents of the current row of gameboard. (1D array)  aEntries – the array/list of word entries  nElem – number of elements in aEntries  nIndex – address where the index of the current entry being guessed in the aEntries  nCurrCol – address where the column index of current letter being guessed in the gameboard  cChoice – letter input/choice of player  nCurrentRow - current row index of the gameboard that the player is answering  nCol - number of columns for the current game session | \*nIndex – updated value of index for aEntries  \*nCurrCol – updated value of column index for gameboard |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayMaintenanceContent | Displays the content for the choice box of action in the maintenance phase | nNum – number choice for indicated action for maintenance phase  sString – String contains the indicated action for maintenance phase (i.e. Add Word, etc) | Prints the choice content |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayMaintenanceChoices | Displays all the choices that can be done in the maintenance phase | None | Prints the maintenance phase choices |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| initializeEntries | Initializes all the strings to empty string and integers to zero in aEntries | aEntries – array/list of word entries | aEntries – initialized contents of to empty values |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| printViewBorders | Frame for the listed words | cHeaderOrFooter – character Boolean for determining whether to display the header or footer part | Prints the borders |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayAddWordHeader | Displays the add word header during the add word action | None | Prints the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayAddTriviaHeader | Displays the add trivia header during the add trivia action | None | Prints header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| addTriviaMenu | The main menu for the add trivia action. Executes the adding of new trivia to existing entry. Asks user which word to add trivia with. If word exists, it calls addTrivia() function. Otherwise it exits the function | aEntries – array/list of word entries  nElem – number of elements in aEntries | Updated clue contents and number of clues in one element/entry in aEntries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayModifyWordHeader | Displays the modify word header during the modify word action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| modifyWord | Modifies word of the chosen oneEntry. Asks user for modified word. If entered word exists, it asks user for another string input. Otherwise it modifies the current word to the entered string by user | oneEntry – address of the single entry to be modified  aEntries – array/list of word entries  nElem – number of elements in aEntries | oneEntry->answer – modified string/word entry |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayModifyCluesHeader | Displays the modify clues header during the modify clues action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayCluesChoices | Displays all the clues that can be modified in the chosen entry (oneEntry) | oneEntry – chosen word entry that will be modified  nNumClues – number of clues oneEntry has  toModify – Boolean if display would be for modification purposes or not | Displays all the clues of oneEntry |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayModifyTriviaChoices | Displays the options on which part of chosen clue that can be modified | None | Displays the choices for modification of a single clue |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| getClueModification | Gets the modified phrase for the chosen clue part (relation or relation value) | oneTrivia – address of the clue being modified  sClueToModify – the specific part of the clue being modified  sTypeOfClue – string that indicates whether the part of the being modified is the “Relation” or “Relation Value” | sClueToModify – modified phrase for the chosen part of the clue |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| modifyClue | Modifies the clue. Modification menu of the chosen clue. Askes user which part of the clue would be modified. Calls getCluModification() function if player chooses to modify either the relation or relation value of the clue | oneTrivia – address of the clue being modified | \*oneTrivia – modified phrases of the clue |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayModifyEntryHeader | Displays the modify entry header during the modify entry action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayModificationChoices | Displays the modification choices/actions for the chosen entry | None | Displays the modification choices for the entry |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| executeModification | Main menu for the modify entry. Asks user which word to modify. If entered word exists, it displays the modification choices of the entries and asks user which to modify. Exits the function if entered word is not found. Calls modifyEntry() function for clue modification and modifyWord() for word modification | aEntries – array/list of word entries  nElem – number of elements in aEntries | aEntries – modified one of its word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayDeleteWordHeader | Displays the delete word header during the delete word action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| clearLastEntry | Responsible for clearing the values and contents of the last entry | oneEntry – address of the last entry that is being deleted/cleared | \*oneEntry – cleared contents of the last entry |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| executeClueDeletion | Responsible for the deletion of the chosen clue to be deleted and updates the number of clues left. | aCluesList – array/list of clues in one entry  nNumClues – address where the number of clues would be updated  nDeleteIndex – the index of the clue to be deleted in aCluesList | aCluesList – updated contents/list of clues  \*nNumClues – updated count of clues left for the entry |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| deleteClue | Main menu for delete clue action. Asks user which word to be deleted a clue with and deletes the chosen clue by calling executeClueDeletion() function. If number of clues of the chosen word is at minimum capacity (1 clue), then it will exit the function | aEntries – array/list of word entries  nElem – number of elements in aEntries | aEntries – updated clue count and content in one of its elements or word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayViewWordsHeader | Displays the view word header during the view word action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayViewCluesHeader | Displays the view clues header during the view clues action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| viewClues | Main menu for view clues. Asks user which words to view clues with | aEntries – array/list of word entries  nElem – number of elements in aEntries | Displays the chosen word and its clues |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| swapEntries | Swaps the contents of two chosen entries | sDestination – address of the desired destination for the contents of \*sSource  sSource – address where the desired contents would be coming from and the content from \*sDestination will be placed | Swapped contents of \*sDestination and \*sSource |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayExportImportHeader | Displays the export/import header during the export/import action | None | Displays the header |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| import | Imports the entries from the file to the aEntries. If file doesn’t exist, it notifies the user that file doesn’t exist and exits the function. If there is a duplicate word in the current entry, then it asks the user if it should be overwritten or not. | fname – file name with the extesnion  aEntries – array/list of word entries  nElem – address that ccontains number of elements in aEntries | \*nElem – updated number of elements in aEntries  aEntries – updated contents/list/array of word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| intiBoard | Responsible if gameboard can be initialized with the current list of word entries. If it can be initialized then it assigns entries to be used for the gameboard. | gameboard – 2D array of characters containing the first letter of the word entry  nRows – number of rows to be initialized in gameboard  nCols – number of columns to be initialized in gameboard  aEntries – array/list of word entries  nElem – number of elements in aEntries | 0 – if gameboard not initialized  1 – if successfully initialized the contents of the gameboard  gameboard – updates the character contents of gameboard if successfully initialized  use member field – updates the row number the entry is being used |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| play | Facilitates the game play of the user | gameboard – 2D array of characters containing the first letter of the word entry  nRows – number of rows to be initialized in gameboard  nCols – number of columns to be initialized in gameboard  aEntries – array/list of word entries  nElem – number of elements in aEntries | Displays and executes the whole gameplay |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| maintenance | The maintenance phase. It contains all the maintenance actions. Calls specific functions for specific action of choice | None | Exports entries to file if player decides to export it |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| searchWord | Searches if entered word by user exists in the current list of word entries | key – the string of the word to look for  aEntries – array/list of word entries  nElem – number of elements in aEntries | -1 – if key is not found  Returns the index in aEntries if key is found |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| listWords | Lists all the words in aEntries (list/array of word entries) | aEntries – array/list of word entries  nElem – number of elements in aEntries | Displays the list of words |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| addWord | Asks user for new word to be added to aEntries. If entered word exists it exits the function, and if not it calls the addTrivia() function to add trivia to the new word | aEntries – array/list of word entries  nElem – address that ccontains number of elements in aEntries | \*nElem – updated value/count of elements in aEntries if new word is successfully added  aEntries – updated list of word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| addTrivia | Continuously asks user for phrases for relation and relation value member field of one entry until max capacity is reached or at least 1 trivia is successfully entered or user decides to cancels to add trivia | oneEntry – address of the single word entry to be added trivia with | \*oneEntry – updated clue contents and clue count |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| modifyEntry | Asks user which clue would be modified in the chosen entry | oneEntry – address of the single word entry to be modified clues with | \*oneEntry – updated clue contents |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| deleteWord | Asks user which word to delete. If word exists it deletes the word and updates the contents of aEntries and number of elements left. If not, it exits the function | aEntries – array/list of word entries  nElem – address that ccontains number of elements in aEntries | \*nElem – updated number of elements left when deletion is successful  aEntries – updated list of word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| viewWords | Displays all the entries in the aEntries with its corresponding in an alphabetical next-or-previous format by calling viewEntry() funciton. Asks user whether to view the next or previous clue or exit. | aEntries – array/list of word entries  nElem – number of elements in aEntries | Displays all the entries in alphabetical order |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| viewEntry | Displays one entry/word with its corresponding clues with help of displayAllClues() funciton | entry – the struct that contains the word and clues of an entry | Displays the word entry with its corresponding clues |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| displayAllClues | Displays all the clues of the entry | aList – list/array of clues of the current entry  nNumClues – number of clues in the current entry | Displays all the clues of the entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| export | Exports the contents of aEntries to a file. | fname – the file name with the extension  aEntries – array/list of word entries  nElem – number of elements in aEntries | Creates a file if file doesn’t exist yet, and writes the contents of aEntries there.  Overwrites a file if it exists |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
| sortEntries | Sorts the word entries in alphabetical order using selection sort | aEntries – array/list of word entries  nElem – number of elements in aEntries | aEntries – updated the order of the word entries |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Description** | **Input Parameter** | **Return Data** |
|  |  |  |  |