# Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Scope | Data Type | Range of values |
| games | Only used in the main routine | Integer (Stores the number of game the user wants to play) | 0 – 10 (Will decrease by one each time a game is played) |
| MASTER\_LIST  (Stores | Local (Only used in RandomWord function) | Constant List (consisting of only of words that can be used as an answer) | Never changes (only words that are pre-set) |
| answer | Local (Only used in game function) | String (consists of the word that is being guessed) | Will have one word from the master list |
| guessed | Local (Only used in game function) | String (consists of the number of letters that have been guessed) | Contains underscores (“\_”) spaces (for letter separation) and letters (that are correctly guessed) |
| lives | Local (Only used in game function) | Integer (Consists of the number of lives the user has) | 0 – 10 (decreases when the user incorrectly guessed) |
| letter\_guessed | Local (only used in game function) | String (consists of the letter that has been guessed [changes each time the user guesses a letter]) | Only letters that the user has guessed |

# Test Plan

|  |  |
| --- | --- |
| Test Point | Function in Code |
| 1 | Type verification in number of games (Making sure they user inputs a number and not letters or a decimal number) |
| 2 | Limiting the number of games to a max of 10 |
| 3 | Running the number of games the user wants |
| 4 | Making the correct number of spaces for guessed letters (i.e if the answer is 5 characters then making the guessing space \_ \_ \_ \_ \_) |
| 5 | Making sure the user has lives or else ending the game |
| 6 | Checking to see if the user has guessed the word |
| 7 | Checking to see if the user has entered any numbers when they are only supposed to be letters |
| 8 | Making sure the user only enters one letter at a time |
| 9 | Making sure the user has not guessed the same letter |
| 10 | Checking to see if the letter guessed is in the answer |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Point | input | Expected outcome | Actual Outcome |
| 1 | abc | Program telling me I need to enter only numbers |  |
| 1 | Abc123 | Program telling me I need to enter only numbers |  |
| 2 | 0 | Program ending |  |
| 1 | 5 | Program continuing |  |
| 2 | 0 | Program ending |  |
| 2 | 1 | Program running 1 game |  |
| 2 | 5 | Program running 5 games |  |
| 2 | 10 | Program running 10 games |  |
| 2 | 11 | Program saying the maximum number of games in 10 |  |
| 3 | 1 | Program running 1 game |  |
| 3 | 5 | Program running 5 games |  |
| 3 | 10 | Program running 10 games |  |
| 4 | *Word from random generated list* | Program making correct number of spaces for letters |  |
| 4 | *Word from random generated list* | Program making correct number of spaces for letters |  |
| 4 | *Word from random generated list* | Program making correct number of spaces for letters |  |
| 5 | 10 lives | Game ends when lives equal 0 (or word is guessed) |  |
| 6 | “*The Answer*” | The game ends when the correct word is guessed (or lives = 0) |  |
| 7 | a | Programs continues |  |
| 7 | 0a | Programs says I have entered a number |  |
| 7 | 2 | Programs says I have entered a number |  |
| 8 | f | Program continues |  |
| 8 | sj | Program says I have entered multiple letters |  |
| 8 | dsjh | Program says I have entered multiple letters |  |
| 9 | “*A letter that has already been guessed”* | Program continues |  |
| 9 | “*A letter that has not been guessed”* | Program tells me I have already guessed that letter |  |
| 10 | “*A letter in the Answer”* | Program tells me I have guessed correctly |  |
| 10 | “*A letter* ***NOT*** *in the Answer”* | Program tells me I have guessed incorrectly |  |

# Final Notes

There was no need for parameters for the modules because there is no need to pass something into a module only to return something from a module.