|  |  |  |  |
| --- | --- | --- | --- |
| **Must** | **Should** | **Could** | **Won’t** |
| * Choose random word * Check if letter is in word * Detect win or loss * Work on a laptop * Be free | * Text as AAA rated * Allow operation via keyboard * Deactivate letter after selection * Use family symbolism * Store no data | * Keep win/loss record * Don’t repeat correctly guessed words | * Share to social media * Have multiplayer |

**Prescribed Requirements**

|  |  |
| --- | --- |
| **Requirement** | **Criteria** |
| Choose random word | Application will be run 10 times to ensure different words are chosen |
| Check if letter is in word | Display chosen word in terminal and then click on letters in the word |
| Detect win or loss | Display chosen word in terminal and then purposely win and lose the game |
| Work on a laptop | Run the game on a laptop |
| Be free | Assets are checked to ensure they are open-source or royalty free |

**Self-determined Requirements**

|  |  |
| --- | --- |
| **Requirement** | **Criteria** |
| Text as AAA rated | Use readability tools to ensure AAA rating |
| Allow operation via keyboard | Check all buttons can be accessed using a keyboard |
| Deactivate letter after selection | Check that all letter buttons can only be clicked once |
| Use familiar symbolism | Ask testers if the symbols used are familiar |
| Store no data | Ensure no files are written to storage |

**get\_word**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| word\_file | BEGIN get\_word  word\_list = READ word\_file  word = NULL  WHILE word <= 3  word = SELECT RANDOM from word\_list  ENDWHILE  RETURN word  END get\_word | word |

**Check letter**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| word  guessed\_word  guess  gallows | 1 BEGIN check\_letter  2 in\_word = FALSE  3 INPUT guess  4 FOR count = 0 to LEN(word)  5 IF guess == word[count]  6 guessed\_word[count] = guess  7 in\_word = TRUE  8 ENDIF  9 NEXT count  10 ENDFOR  11 IF in\_word == FALSE  12 gallows = gallows + 1  13 ENDIF  14 DISPLAY gallows  15 DISPLAY guessed\_word  16 END check\_letter | gallows  guessed\_word |

**Check letter**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line** | **word** | **guessed\_word** | **in\_word** | **guess** | **count** | **gallows** | **Conditions** | **Input/output** |
| 1 | DIG | \_ \_ \_ |  |  |  | 3 |  |  |
| 2 |  |  | F |  |  |  |  |  |
| 3 |  |  |  | I |  |  |  | guess ? I |
| 4 |  |  |  |  | 0 |  | 0 < 3 ? is T |  |
| 5 |  |  |  |  |  |  | I == D ? is F |  |
| 9 |  |  |  |  | 1 |  |  |  |
| 4 |  |  |  |  |  |  | 1 < 3 ? is T |  |
| 5 |  |  |  |  |  |  | I == I ? is T |  |
| 6 |  | \_ I \_ |  |  |  |  |  |  |
| 7 |  |  | T |  |  |  |  |  |
| 9 |  |  |  |  | 2 |  |  |  |
| 4 |  |  |  |  |  |  | 2 < 3 ? is T |  |
| 5 |  |  |  |  |  |  | I == G ? is F |  |
| 9 |  |  |  |  | 3 |  |  |  |
| 4 |  |  |  |  |  |  | 3 < 3 ? is F |  |
| 11 |  |  |  |  |  |  | T == F ? is F |  |
| 14 |  |  |  |  |  |  |  | gallows = 3 |
| 15 |  |  |  |  |  |  |  | guessed\_word = \_ I \_ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Input** | **Expected Output** | **Actual Output** | **Action** |
| Click letter 1st time | Click ‘E’ | ‘E’ accepted | ‘E’ accepted | N/A |
| Click letter 2nd time | Click ‘E’ | ‘E’ rejected | ‘E’ accepted | Disable button after 1st click |
| Click letter 2nd time | Click ‘E’ | ‘E’ rejected | ‘E’ rejected | N/A |
| Correct letter guess | Click ‘E’ | ‘E’ appears on guessed word | ‘E’ appears | N/A |
| Incorrect letter guess | Click ‘D’ | Gallows progress | Gallows progressed | N/A |
| Win condition | Guess word | Winner! | Nothing | Check winner test |
| Win condition | Guess word | Winner! | Winner! | N/A |
| Win condition | Fail to guess word | The word is ANGUS | Runtime error | Check parameters on method |
| Win condition | Fail to guess word | The word is ANGUS | The word is ANGUS | N/A |
| Random new word | Click ‘New Word’ | Number of letters change | Number of letters change | N/A |

**get\_word()**

|  |  |  |  |
| --- | --- | --- | --- |
| **Run #** | **Expected Output** | **Actual Output** | **Action** |
| 1 | Word > 3 letters | cameo | N/A |
| 2 | Word > 3 letters | zap | N/A |
| 3 | Word > 3 letters | frog | N/A |
| 4 | Word > 3 letters | at | Check letter size condition |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Concern** | **Data** | **Recommendation** |
| Letter ‘E’ button | Not disabling | Alpha Testing | Troubleshoot letter\_btn method |
| Win message | Not showing | Alpha Testing | Troubleshoot letter\_btn method |
| Loss message | Runtime Error | Alpha Testing | Troubleshoot letter\_btn method |
| UI | Button text too small | Beta Testing | Increase button text size |
| UI | Keyboard controls not working | Beta Testing | Investigate keyboard controls |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Impact** | **Rating** | **Justification** | **Recommendation** |
| Personal | Health and wellbeing | 5/5 | Users reported feeling relaxed | N/A |
| Personal | Quality of life | 4/5 | User reports a reduction in stress | N/A |
| Personal | Access to resources, services and opportunities | 2/5 | User can recall words, but doesn’t know their meaning | Include definition when the word is revealed. |
| Personal | Resilience | 4/5 | User reports having more energy the next day. | N/A |

**Prescribed Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Criteria** | **Rating** | **Recommendation** |
| Choose random word | Application will be run 10 times to ensure different words are chosen | 5/5 | N/A |
| Check if letter is in word | Display chosen word in terminal and then click on letters in the word | 5/5 | N/A |
| Detect win or loss | Display chosen word in terminal and then purposely win and lose the game | 5/5 | N/A |
| Work on a laptop | Run the game on a laptop | 5/5 | N/A |
| Be free | Assets are checked to ensure they are open-source or royalty free | 5/5 | N/A |

**Self-determined Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Criteria** | **Rating** | **Recommendation** |
| Text as AAA rated | Use readability tools to ensure AAA rating | 4/5 | Button text should be made bold to achieve AAA rating |
| Allow operation via keyboard | Check all buttons can be accessed using a keyboard | 0/5 | Investigate how to achieve keyboard control in PyQT6 |
| Deactivate letter after selection | Check that all letter buttons can only be clicked once | 5/5 | N/A |
| Use familiar symbolism | Ask testers if the symbols used are familiar | 5/5 | N/A |
| Store no data | Ensure no files are written to storage | 5/5 | N/A |