|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 1 |
| **Purpose** | To see how the program will react when at startup a dictionary is missing |
| **Preconditions** | Change the name of English.zkj to something the program won’t recognize |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: “Error couldn’t find main dictionary” program wouldn’t run. Passed** |

note: Tests 1-7 were inspired through code review2

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 2 |
| **Purpose** | To see how the program will react when at startup a dictionary is missing |
| **Preconditions** | Change the name of zdict.zkj to something the program won’t recognize |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: “Error couldn’t find main dictionary” program wouldn’t run. Passed** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 3 |
| **Purpose** | To see how the program will react when at startup a dictionary is missing |
| **Preconditions** | Change the name of examples.zkj to something the program won’t recognize |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: Failed – program executed as normal without any obvious effects** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 4 |
| **Purpose** | To see how the program will react when at startup a dictionary is missing |
| **Preconditions** | Change the name of zdict.zks to something the program won’t recognize |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: “Couldn’t find stroke order…”Still ran, but without the functionality of seeing how to write out the kanji, i.e. stroke order. Anything to do with writing recognition disabled.** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 5 |
| **Purpose** | To see how the program will react when at startup a dictionary is replaced with something by the same name |
| **Preconditions** | Change the name of zdict.zkj to something the program won’t recognize, and change a text file to .zkj format The file replacing is only filled with normal punctuation and characters(i.e. alphabet) |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: “Couldn’t open main dictionary” Program didn’t execute** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary loading test 6 |
| **Purpose** | To see how the program will react when at startup a dictionary is replaced with something by the same name |
| **Preconditions** | Change the name of English.zkj to something the program won’t recognize, and change a text file to .zkj format The file replacing is only filled with non normal punctuation and characters(i.e. alphabet) ie ӓ |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: Ran as usual, no apparent problems** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary/What came with the program when it downloaded loading test 7 |
| **Purpose** | To see how the program will react when at startup an installed is replaced with something by the same name |
| **Preconditions** | Add a file with the same name as the similar.txt file, and see what happens. The file duplicating it is only filled with non normal punctuation and characters(i.e. alphabet) ie ӓ |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: Ran as usual, when looking at a kanji’s information, “similar” was disabled, i.e. nothing showed up** |

|  |  |
| --- | --- |
| **Test Case ID** | Dictionary/What came with the program when it downloaded loading test 8 |
| **Purpose** | To see how the program will react when at startup an installed is replaced with something by the same name |
| **Preconditions** | Add a file with the same name as the furigana.txt file, and see what happens. The file duplicating it is only filled with non normal punctuation and characters(i.e. alphabet) ie ӓ |
| **Inputs** | none |
| **Expected Outputs** | Program will throw an error message and refuse to execute |
| **Postconditions** | None |
| **Execution History** | **Try 1: Ran as usual, no apparent change** |

|  |  |
| --- | --- |
| **Test Case ID** | Kana Writing Test 2 |
| **Purpose** | To check the functionality of the drawing capabilities when seeking a specific letter or kanji, |
| **Preconditions** | In writing section, some letters are selected(in this case と　and　た) |
| **Inputs** | To, ta |
| **Expected Outputs** | Input accepted as 100% correct, no mistakes shown |
| **Postconditions** | None |
| **Execution History** | **Try 1: passed**  **\*Note: stroke order matters a lot; if the stroke order is wrong, it is likely that the correct symbol will not show up.** |

|  |  |
| --- | --- |
| **Test Case ID** | Kana Writing Test 3 |
| **Purpose** | To check the functionality of the drawing capabilities when seeking a specific letter or kanji, |
| **Preconditions** | In writing section, some letters are selected(in this case わ　and　を) |
| **Inputs** | Wa, wo |
| **Expected Outputs** | Input accepted as 100% correct, no mistakes shown |
| **Postconditions** | None |
| **Execution History** | **Try 1: passed**  **\*Note: stroke order matters a lot; if the stroke order is wrong, it is likely that the correct symbol will not show up.** |