TypingApp Testing

By: Amir Altakroori

Test plan:

We will divide testing to three levels which are ~~component's~~ level, logical level and deep test level. In the ~~component's~~ level we will test the UI components, timer reality and code's samples. In the logical level, first of all we will test the input process such as dealing with each words by itself, if the comparison process is a case sensitive and recognizing white spaces. Then we will test the results and measure how much it is accurate. In deep test level we will try to find bugs and errors.

Test cases:

* **~~Components~~ level:**
  + Buttons:
    - Upload button: pass
    - Practice button: pass
    - Exit button: pass
    - Back button: pass
    - End button: failed -> results log didn't appear and the program returned to the main menu directly.
  + List:
    - Language's lest: pass
  + Timer:
    - Time realistic: pass
    - Unique timer's start point
* **Logical level:**
  + Input process:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario | Test Step | Expected Result | Actual outcome | Status |
| Verify that the program can receive a correct words and compare it with the origin word | In training level, writing a single word that similar with the written word. | The program should receive the word and make the origin word color in green color which mean that word is correct. | The program receives the word and makes the origin word color in green color | passed |
| Verify that the program can receive an incorrect words and compare it with the origin word | In training level, writing a single word that not similar with the written word and it's length in average not exceeded the origin word's length. | The program should receive the word and make the origin word color in red color which mean that word is not correct. | The program receives the word and makes the origin word color in red color | passed |
| Verify that the program can recognize the deference between white spaces | In training level, writing a correct word. After writing the final character there will be 4 cases:   1. The origin word is located in the middle of text and it ends with blank space and we will type blank space. 2. The origin word is located in the middle of text and it ends with blank space and we will type new line. 3. The origin word is located in the end of text and it ends with new line and we will type new line. 4. The origin word is located in the end of text and it ends with new line and we will type blank space. | * In case 1:   The program should accept that character and go to the next word without scoring an error.   * In case 2:   The program should reject that character and go to the next word with scoring an error.   * In case 3:   The program should accept that character and go to the next line without scoring an error.   * In case 4:   The program should reject that character and go to the next line with scoring an error. | * In case 1:   The program accepts that character and goes to the next word without scoring an error.   * In case 2:   The program accepts that character and goes to the next word without scoring an error.   * In case 3:   The program accepts that character and goes to the next word without scoring an error.   * In case 4:   The program accepts that character and goes to the next word without scoring an error. | Case 1  Passed  Case 2  Failed  Case 3  Passed  Case 4  Failed |
| Verify that the program's comparison is case sensitive. | In training level, writing a single word with similar character but in deferent letter case. | The program should receive the word and make the origin word color in red color which mean that word is not correct. | The program receives the word and makes the origin word color in red color which mean that word is not correct. | passed |
| Find the training's input limitations | In training level, writing a huge amount of characters for a single word | The program should receive a huge characters of words | The program crashed after typing 92 character. | Failed |
| Test the case if the user finish the whole text before finishing the time | In the training, trying to finish the text in a correct way or not before consuming the time | The program should generate new text and update it's values | The program generates new text and updates it's values | pass |
| Verify the compatibility between fields in the same result's log | In the result's level, find the sum of list of errors filed and compare it with the number of errors | The results should be compatible and same. | There is an deference between two result | Failed |
| Verify the reality of the output | In training level, we will do the training and record the results manually. | The program's data should be compatible with out manual data. | The program's data are compatible with out manual data. | Failed in number of errors |

**Issues**:

**Environment:**

Operating system: Windows 10

IDE: NetBeans 8.2

* [Practice] program can't differentiate between blank spaces  
   **Steps to reproduce:**
  + Start a training mode
  + Start typing words and when you reach any middle of line write a similar origin word
  + After writing the final character press on enter instead of space.

**Expected result:**

Program should score a wrong input value and score it as a wrong answer.

**Actual result:**

The program accept the value and score it as an accepted answer.

* [Practice] There is no compatibility between results fields   
  **Steps to reproduce:**
  + Finish a training mode with some mistakes and accepted word
  + Find the summation of list of errors filed
  + Compare it with the numbers of errors

**Expected result:**

The results should be compatible and same.

**Actual result:**

The program's data are compatible with out manual data.