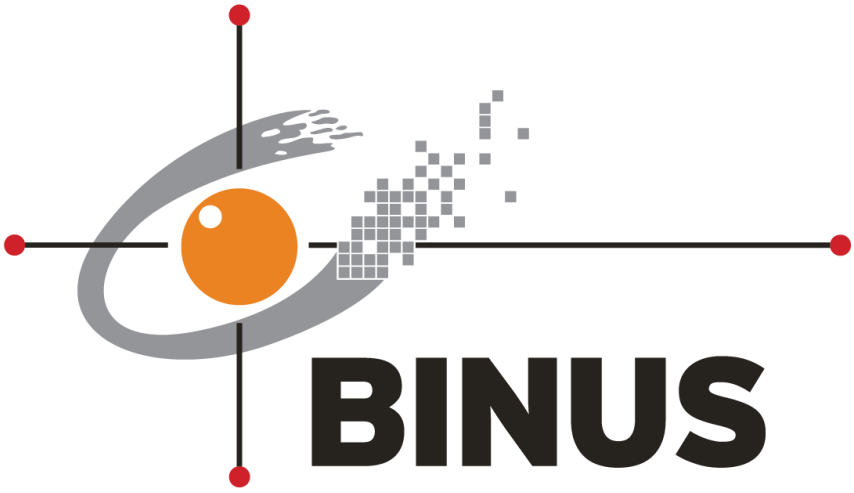
Algorithm and Programming

COMP6047001



BINUS
UNIVERSITY
INTERNATIONAL

Final Project

By : Christopher Owen / 2502019180 / L1BC

Project Report: LightningType

I. Overview

i. About

LightningType is a speed typing test application, which can measure the user's typing speed in the form of Words per Minute (WPM). The user is given 120 characters to type with the least amount of time.

ii. Background and inspiration

The COVID-19 pandemic has hit us for almost 2 years and forces people to do most regular activities at home. With that, people start to find new hobbies that can accompany them while being at home. One of those hobbies in the technology world is building a custom mechanical keyboard. I have not had the chance to build one, but I already have watched so many related contents.

All custom mechanical keyboard builders will test their keyboard using an application or website called speed typing test. There are many websites which provide those features that also can train people to increase their typing speed. Therefore, I, as an enthusiast and also a computer science student, tried to make a speed typing test program using python.

II. Specifications

i. Objective

The user needs to type with the shortest amount of time according to the text given.

ii. Content

Files: Text.txt, TypingTestOff.py (other files in folder are trial and error)

1. Text.txt

15 lines string of words that will be randomized in order to vary the user's test.

```
since other this during play point good develop against thing plan move for own long but ask school begin much order out new little increase
long take tell we as these general turn only word all by another help one lead if think might program early since follow through another can
run most new very use only the ask begin these even first little program late too early out real she last part use must also there under all
last run small group year can same help write may would if might out word many late but while never long go must we such interest because as
nation much all both during order well these may word eye look right want we place just group off be those people line since end school line
of about at increase program open which any that become some back then hand would good only nation right call stand help plan things problem
right against such also play must she form that but what last man end because however both here any late never like those get another should
after off run know own find about high at then line how say head lead good program eye problem but know general hold point at feel about way
number word can they mean fact day call place we few course set write that govern turn open own turn other would right say to should come we
never public take a open possible through most show ask come you who end how find another with make out consider what which find and this of
public the when because back take form time begin word know work that but then right off even child consider of other large eye lead without
place develop go need must long under program great world for time play order make too at any need open into order high call want line right
can of each all make set but need for course part a or much form now man again these program those high move thing use look much even own up
find ask show late say too when up good in program stand little small also tell at with develop it govern go here do before tell like change
when will this become nation another give will also system state over hand show time so line still be work it other under any this back mean
```

2. TypingTestOff.py

All programs: modules, layouts, and functions.

-
```python
import curses
from curses import wrapper
import time
import random

```
These are the modules used in the program.
  - Curses module provides the curses library and enables the user to style and edit the terminal.
  - Wrapper will initialize the curses module and restore the terminal back after the program is finished.
  - Time module provides time-related functions.
  - Random module provides randomize functions for certain program.

-
```python
6
7   def start_screen(typetest):
8       typetest.clear()
9       typetest.border()
10      typetest.addstr(1, 25, "                      _______             _______  _______                         _______     _______  _______")
11      typetest.addstr(2, 25, "\\        /\\        / |         |         |         |       |      /\\         /\\      |               |     |       |
        ")
12      typetest.addstr(3, 25, " \\      /  \\      /  |_______  |         |         |       |     /  \\       /  \\     |_______        |     |       |
        ")
13      typetest.addstr(4, 25, "  \\    /    \\    /   |         |         |         |       |    /    \\     /    \\    |               |     |       |
        ")
14      typetest.addstr(5, 25, "   \\  /      \\  /    |         |         |         |       |   /      \\   /      \\   |               |     |       |
        ")
15      typetest.addstr(6, 25, "    \\/        \\/     |_______  |_______  |_______  |_______|  /        \\/        \\  |_______        |     |_______|
        ")
16      typetest.addstr(8, 28, "                  _______             _______                        _______  _______            _______  _______")
17      typetest.addstr(9, 28, "|            o  |           |       |      |     |\\     |  o  |\\     |  |           |      \\       /  |        |  |")
18      typetest.addstr(10, 28, "|            |  |    ___   |_______|      |     |  \\   |  |  |  \\   |  |    ___    |       \\     /   |        |  |_______")
19      typetest.addstr(11, 28, "|            |  |   |   |  |       |      |     |   \\  |  |  |   \\  |  |   |   |   |        \\   /    |________|  |")
20      typetest.addstr(12, 28, "|            |  |       |  |       |      |     |    \\ |  |  |    \\ |  |       |   |          |      |           |")
21      typetest.addstr(13, 28, "|_______     |  |_______|  |       |      |     |     \\|  |  |     \\|  |_______|   |          |      |           |_______")
22      typetest.addstr(15, 50, "Speed Typing Test will be measeured with Words Per Minute (WPM)")
23      typetest.addstr(17, 69, "Press ENTER to begin!")
24      typetest.refresh()
25      while True:
26          c = typetest.getch()
27          if c == ord('\n'):
28              break
29
```
start_screen function initializes the beginning screen when the program is run.
  - Line 8 : clear() will clear the terminal.
  - Line 9 : border() will print/display border to the terminal.
  - Line 10 – 23 : addstr() will print the given string with certain coordinates (row, column)
  - Line 24 : refresh() will refresh the terminal.
  - Line 25 – 28 :The while loop contains if the user press enter, then the program will get out of the while loop and proceed. Otherwise, it will not.

```
30
31  def type_result_screen(typetest, target, current, wpm=0, time=0):
32      typetest.border()
33      typetest.addstr(1, 28, "              ______            ______                    ______  ______          ______    ______", curses.
        A_BOLD)
34      typetest.addstr(2, 28, "|         o  |        |      |    |    |\\    | o  |\\    | |          |    \\    /  |      | |", curses.
        A_BOLD)
35      typetest.addstr(3, 28, "|         | |  __    |______|    |    | \\   | |  | \\   | |  __      |     \\  /   |      | |______",
        curses.A_BOLD)
36      typetest.addstr(4, 28, "|         | | |  | |      |    |    |  \\  | |  |  \\  | |  |  |      |      \\ /    |______|  |", curses.
        A_BOLD)
37      typetest.addstr(5, 28, "|         | |     | |      |    |    |   \\ |  |  |   \\ |  |     |    |        |    |         |", curses.
        A_BOLD)
38      typetest.addstr(6, 28, "|_______  |  |_____| |      |    |    |    \\|  |  |    \\|  |_____|    |        |    |         |______",
        curses.A_BOLD)
39      typetest.addstr(9, 12, target)
40      typetest.addstr(12, 50, f"WPM: ")
41      typetest.addstr(12, 55, f"{wpm}", curses.A_BOLD)
42      typetest.addstr(12, 100, f"Time: ")
43      typetest.addstr(12, 106, f"{round(time, 1)}", curses.A_BOLD)
44
45      for i, char in enumerate(current):
46          correct_char = target[i]
47          color = curses.color_pair(1)
48          if char != correct_char:
49              color = curses.color_pair(2)
50
51          typetest.addstr(9, 12+i, char, color)
52
```

- type_result_screen function will execute the following program after the start_screen function is done with some parameters.
  - Line 39 – 43 : The string of words, wpm, and time will be shown in a certain location (row, column) with the numbers bolded (curses.A_BOLD)
  - Line 45 – 51 : The for loop will scan every character that is inputted by the user with the string of words provided on the screen.
    If the inputted characters and the target characters match, it will show the color initialized with code 1. Otherwise, it will chow the color code 2.

```
53
54  def load_text():
55      with open("Text.txt", "r") as f:
56          lines = f.readlines()
57          return random.choice(lines).strip()
58
```

- load_text function will open the file and store in f
  - Line 56 : readlines() reads the targeted file by each line
  - Line 57 : return command will randomly choose one line and remove the enter (\n) with strip()

```
59
60  def wpm_time_type_settings(typetest):
61      target_text = load_text()
62      current_text = []
63      wpm = 0
64      typetest.nodelay(True)
65
66      while True:
67          if current_text == []:
68              start_time = time.time()
69          time_elapsed = max(time.time() - start_time, 1)
70          wpm = round((len(current_text) / (time_elapsed / 60)) / 5)
71
72          typetest.clear()
73          type_result_screen(typetest, target_text, current_text, wpm, time_elapsed)
74          typetest.refresh()
75
76          if "".join(current_text) == target_text:
77              typetest.nodelay(False)
78              break
79
80          try:
81              key = typetest.getkey()
82          except:
83              continue
84
85          if ord(key) == 27:
86              break
87
88          if key in ("KEY_BACKSPACE", '\b', "\x7f"):
89              if len(current_text) > 0:
90                  current_text.pop()
91          elif len(current_text) < len(target_text):
92              current_text.append(key)
93
```

- wpm_time_type_settings function will initialize all calculations and mechanics while the test is running.
  - Line 61 – 63 : Declares target_text with load_text() function, current_text with a list, and wpm with 0.
  - Line 64 : nodelay() will enable the time and wpm to keep counting even though the user is not typing.
  - Line 67 – 68 : In the while loop, if the current_text is empty, the wpm and time will not start until the user type.
  - Line 69 : time_elapsed is the time counter.
  - Line 70 : wpm provides the calculation for the words per minute.
  - Line 73 : type_result_screen will be called and given the parameters necessary for displaying
  - Line 76 – 78 : if the current_text turned into a string matches with target_text, the program will stop the timer get out of the while loop.
  - Line 80 – 83 : try getkey() which is waiting for the user to type something. If yes, it will proceed to the rest of the program. If no, the program will loop back to the while loop.
  - Line 85 – 86 : if the user hits the escape key (ASCII esc is 27), the program will immediately exit.
  - Line 88 – 92 : if the user hits the backspace key and the length of the current text is greater than 0, it will eliminate a character in the current_text (pop()).
    If the length of the current_text is still less than the target_text, the user can still input keys to the current_text as a list.

```python
94
95  def main(typetest):
96      curses.init_pair(1, curses.COLOR_GREEN, curses.COLOR_BLACK)
97      curses.init_pair(2, curses.COLOR_RED, curses.COLOR_BLACK)
98
99      start_screen(typetest)
100     while True:
101         wpm_time_type_settings(typetest)
102         typetest.addstr(14, 8, " ______   ______          ______   ______          ______                    ______  ______       ______")
103         typetest.addstr(15, 8, "|         |        |  |\\   |  |        |       |     /\\        |     |      |  |          /\\        |  o  |      |  |\\   |  |           |")
104         typetest.addstr(16, 8, "|         |        |  | \\  |  |   __   |_______|    /  \\       |     |      |  |         /  \\       |  |  |      |  | \\  |  |______     |")
105         typetest.addstr(17, 8, "|         |        |  |  \\ |  |   |  | |\\          /____\\      |     |      |  |        /____\\      |  |  |      |  |  \\ |          |  |")
106         typetest.addstr(18, 8, "|         |        |  |   \\|  |      | |  \\       /      \\     |     |      |  |       /      \\     |  |  |      |  |   \\|          |  |")
107         typetest.addstr(19, 8, "|______   |________|  |    \\|  |______|  |    \\   /        \\    |     |______|  |______ /        \\    |  |  |______|  |    \\|  ______|  o")
108         typetest.addstr(21, 50, "You have completed the test! Press any key to play again!")
109         typetest.addstr(22, 70, "Press ESC to exit")
110         typetest.refresh()
111
112         key = typetest.getkey()
113         if ord(key) == 27:
114             break
115
```

main function will declare some commands, call functions, and display the ending screen

- Line 96 – 97 : curses.init_pair() declares 2 colors that can be called using the number code
- Line 99 : call the start_screen function
- Line 101 – 110 : in the while loop, wpm_time_type_settings will be called and display the strings.
- Line 112 – 114 : declares key as getkey(), to wait for the user to type something. If the user hits the escape key, the program will exit. Otherwise, it will run the program again (play again).

```python
116
117 wrapper(main)
118
```

Call the main function with the wrapper function while initializing the curses module.

III. Demonstration

i. Start Screen

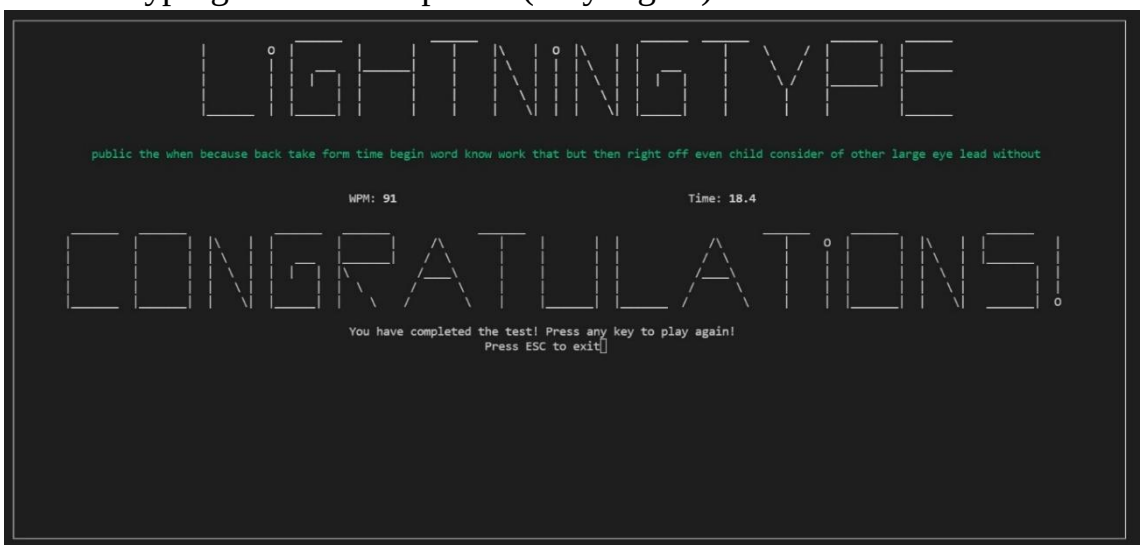
ii. Typing Screen

iii. Typing Screen Process

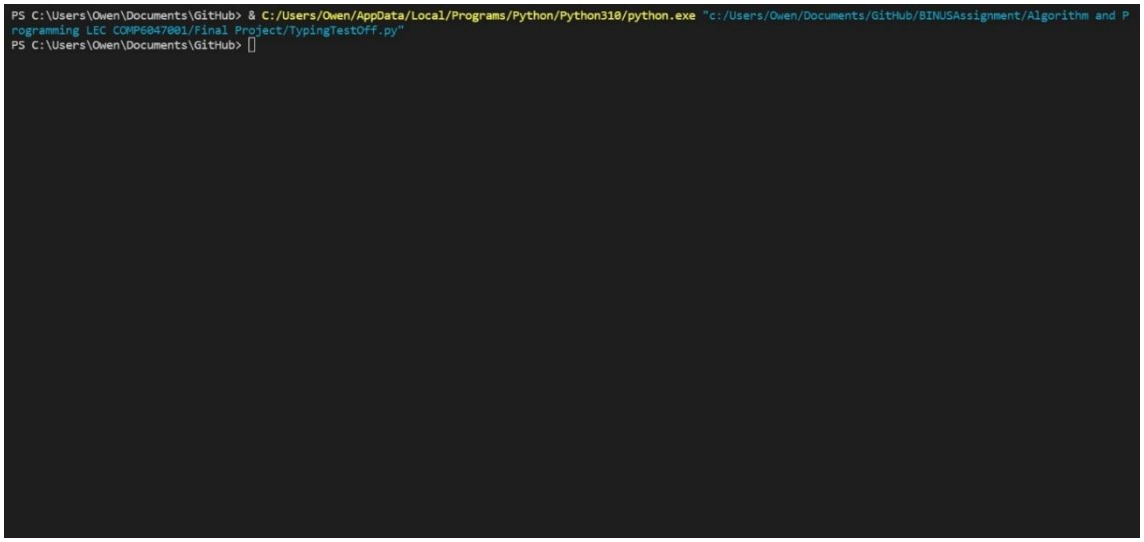
iv. Typing Screen Completed

v. Typing Screen (Play Again)

vi. Typing Screen Completed (Play Again)



vii. Exit Screen (to terminal)



IV. References

i. https://www.youtube.com/watch?v=NQ5i1kJAA6Y&t=512s
ii. https://www.youtube.com/watch?v=JBE4OwdqzQ8&t=444s
iii. https://www.youtube.com/watch?v=elfqLbiLAT4
iv. https://docs.python.org/3/library/curses.html#module-curses