




TRABALHO PRÁTICO 1 – TYPING TEST

Programação de Sistemas Robóticos 2024/2025



Pedro Afonso 104206
Rafael Morgado 104277

OBJETIVO DO TRABALHO

- Este projeto visa criar um programa similar a um *typing test* com dois modos: um modo para digitar letras individuais e outro para palavras, ambas geradas de forma aleatória.
- Cada modo tem limites definidos, como tempo máximo ou número de inputs permitidos, permitindo avaliar a velocidade e precisão de digitação de forma simples e eficaz.

-utm, --use_time_mode : Ativa o modo limite de tempo.

-mv, --max_value MAX_VALUE : Define o limite de tempo (segundos) ou o número máximo de inputs.

-uw, --use_words: Ativa o modo de inserção de palavras em vez de caracteres únicos.

-u, --user USER : Especifica o nome de utilizador (caso não seja fornecido o nome **User** será o padrão).

FUNCIONAMENTO DO TESTE

- O utilizador deve pressionar qualquer tecla para iniciar o desafio, dando início a uma contagem decrescente:

```
Bem-vindo, Rafa! Pressione qualquer tecla para começar o desafio...  
Starting the test in 3 seconds
```

- De seguida, dependendo do modo escolhido, o programa exibirá uma letra, ou palavra aleatória:

```
Digite a palavra: robotics
```

```
Digite a letra: y
```

FUNCIONAMENTO DO TESTE

- No modo de caracteres únicos, após cada input, o programa exibe uma mensagem indicando se a letra foi corretamente inserido, no caso de erro exibe a letra esperada.

```
Digite a letra: j  
Seu input: j Correto!
```

```
Digite a letra: a  
Seu input: e Incorreto! Esperado: 'a'
```

- No modo de palavras segue o mesmo princípio de funcionamento só que se o input estiver incorreto o programa exibirá os caracteres corretos da palavra a verde e os incorretos a vermelho, destacando assim os erros.

```
Digite a palavra: robotics  
robotics Correto!
```

```
Digite a palavra: industries  
Indusrties Incorreto! Esperado: 'industries'  
Seu input: Indusrties
```

CONDIÇÕES DE FINALIZAÇÃO

- Expiração do tempo máximo definido

Tempo máximo atingido!

- Limite máximo de inputs atingido

Número máximo de inputs atingido!

- Tecla de interrupção do programa pressionada (SPACE)

Teste interrompido!

ESTATÍSTICAS GERADAS

- No final são apresentadas várias estatísticas sobre a nossa performance durante o teste

Input esperado



Input recebido



Duração



```
'inputs': [Input(requested='this', received='this', duration=2.732403516769409),  
            Input(requested='evolve', received='elvove', duration=3.585782051086426),  
            Input(requested='various', received='various', duration=6.371495008468628),  
            Input(requested='humans', received='humans', duration=2.68612003326416),  
            Input(requested='landscape', received='landsacpe', duration=7.16799783706665)]
```

ESTATÍSTICAS GERADAS

Número de inputs corretos

Número de inputs

Duração total do teste

Final do teste

Início do teste

Duração média de resposta

Duração média de resposta correta

Duração média de resposta incorreta

Precisão (hits/total inputs)(%)

```
'number_of_hits': 3,  
'number_of_types': 5,  
'test_duration': 22.546558,  
'test_end': '2024-10-09 19:11:10',  
'test_start': '2024-10-09 19:10:48',  
'type_average_duration': 4.508759689331055,  
'type_hit_average_duration': 3.930006186167399,  
'type_miss_average_duration': 5.376889944076538
```

```
{'accuracy_percent': 60.0,
```

```
41 # configuração do alarme de tempo (quando for atingido o tempo máximo de teste)
42 signal.signal(signal.SIGALRM, handle_timeout)
43
44 # Parsing dos argumentos
45 args = parse_arguments()
46
47 # Inicializa o sistema de teste e o sistema de contagem regressiva
48 user = args.user
49 welcome_message(user)
50 start_test_timer()
51
52 # Definindo o tempo de teste (caso esteja no modo de tempo)
53 if args.use_time_mode:
54     signal.alarm(args.max_value) # Define o alarme de tempo
55
56 # Início do teste
57 test_start = datetime.now()
58
59 while True:
60     if args.use_words:
61         received = input()
62         print(fore.RED + "Texto interrompido!")
63         break
64         if check_letter_accuracy(requested, received):
65             number_of_hits += 1
66             inputs.append(input(requested, received, duration))
67
68             current_inputs += 1
69
70             # Checando se atingiram o número máximo de inputs
71             if not args.use_time_mode and current_inputs == args.max_value:
72                 print(fore.RED + "Número máximo de inputs atingido!")
73                 break
74
75             # Calculando e exibindo as estatísticas
76             collect_statistics(inputs, number_of_hits, test_start, current_inputs, user)
77
78 if __name__ == '__main__':
79     main()
```

Obrigado pela atenção!

```
80
81 # Exibindo o resultado do teste
82 print(fore.GREEN + "Resultado do teste:")
83
84 def should_end_test(args, start_time, current_inputs, end_time):
85     """Verifica se as condições de término do teste foram atingidas."""
86     if args.use_time_mode and time() == end_time:
87         print(fore.RED + "Tempo máximo atingido!")
88         return True
89     if current_inputs == args.max_value:
90         print(fore.RED + "Número máximo de inputs atingido!")
91         return True
92     return False
93
94 # Funções para medir o tempo
95 def tic():
96     """Retorna o tempo de início."""
97     return time()
98
99 def toc(start_time):
```