



UNIVERSIDADE
DE ÉVORA

Escola de Ciências e
Tecnologias



Logo do jogo dos dedos

Autores: João Santos n.º 51966, José Silva n.º 52673

Licenciatura Engenharia Informática
Programação 1 - Ano letivo 2022/2023

Índice

Introdução	2
Descrição do programa	3
Funções	7
Complicações	12
Conclusão	13

Introdução

A realização deste trabalho prático tem como objetivo adquirir os conhecimentos obtidos na cadeira de Programação 1 para desenvolver o “Jogo dos Dedos - Módulo 5” em linguagem C.

É um jogo para 2 jogadores que se joga com os dedos das duas mãos.

O jogo começa com os 2 jogadores com 1 dedo em cada mão. Depois são realizadas jogadas alternadas.

Em cada turno, o jogador pode escolher atacar o adversário ou, em certos casos, igualar os dedos das suas próprias mãos.

Uma mão com zero dedos não poderá atacar ou ser atacada.

Descrição do programa

Output:

Terminal:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos chico-esperto chico-esperto
```

Executar jogo chico-esperto vs chico-esperto

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos humano humano
```

Executar jogo humano vs humano

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos ao-calhas ao-calhas
```

Executar jogo ao-calhas vs ao-calhas

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos ao-calhas chico-esperto
```

Executar jogo ao-calhas vs chico-esperto (também funciona, *./dedos chico-esperto ao-calhas*)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos humano ao-calhas
```

Executar jogo humano vs ao-calhas (também funciona, *./dedos ao-calhas humano*)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$ ./dedos humano chico-esperto
```

Executar jogo humano vs chico-esperto (também funciona, *./dedos chico-esperto humano*)

```
(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
./dedos humano HUMANO
Insira corretamente os jogadores !
```

Após uma inserção errada do jogadores o programa não executa.

Tipos de jogadas:

```
(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
./dedos humano humano
1 🌀

Jogo dos dedos - humano vs humano

humano 1: 1, 1
humano 2: 1, 1

Vez do humano 1: ee
dedo da esquerda ataca dedo da esquerda, ficam 2 dedos

humano 1: 1, 1
humano 2: 2, 1

Vez do humano 2: dd
dedo da direita ataca dedo da direita, ficam 2 dedos

humano 1: 1, 2
humano 2: 2, 1

Vez do humano 1: ed
dedo da esquerda ataca dedo da direita, ficam 2 dedos

humano 1: 1, 2
humano 2: 2, 2

Vez do humano 2: de
2 dedos da direita atacam dedo da esquerda, ficam 3 dedos

humano 1: 3, 2
humano 2: 2, 2
```

Jogadas possíveis são: *ee* (esquerda-esquerda), *dd* (direita-direita), *ed* (esquerda-direita), *de* (direita-esquerda) .

```
humano 1: 3, 2
humano 2: 0, 2

Vez do humano 2: =
divide os 2 dedos pelas duas mãos

humano 1: 3, 2
humano 2: 1, 1
```

Também é possível igualar ao usar '='.

```
humano 1: 3, 2
humano 2: 1, 1

Vez do humano 1: .

(razejb@kali) - [~/C projects/helloworld/51966-52673.tgz]
$
```

Também é possível finalizar o jogo ao usar '.', levando à desistência do jogador.

```
Jogo dos dedos - humano vs ao-calhas

ao-calhas 1: 1, 1
humano 2: 1, 1

Vez do humano 2: EW
jogada inválida !

ao-calhas 1: 1, 1
humano 2: 1, 1

Vez do humano 2: █
```

Após uma inserção errada da jogada o programa realiza um novo input da mesma.

Funções

Jogos possíveis:

int humanoJoga()

int humano_aoCalhasJoga()

int humano_chicoEspertoJoga()

int aoCalhas_chicoEspertoJoga()

void aoCalhasJoga()

void chicoEspertoJoga()

Descrição:

Estas funções definem o jogo que irá acontecer após a inserção dos jogadores no terminal.

Jogadas jogador 1:

```
void verifica_eeJ1( )
```

```
void verifica_ddJ1( )
```

```
void verifica_deJ1( )
```

```
void verifica_edJ1( )
```

Descrição:

Estas funções realizam as jogadas do jogador 1, com principal funcionalidade atacar as mãos do adversário, alterando assim os valores das variáveis das mãos do jogador 2.

Frases jogador 1:

```
void frase_eeJ1( )
```

```
void frase_ddJ1( )
```

```
void frase_deJ1( )
```

```
void frase_edJ1( )
```

Descrição:

Estas funções realizam as frases das jogadas do jogador 1, com principal funcionalidade descrever a situação do jogo em tempo real, estas, criam uma melhor interação da pessoa com o programa.

Jogadas jogador 2:

void verifica_eeJ2()

void verifica_ddJ2()

void verifica_deJ2()

void verifica_edJ2()

Descrição:

Estas funções realizam as jogadas do jogador 2, com principal funcionalidade atacar as mãos do adversário, alterando assim os valores das variáveis das mãos do jogador 1.

Frases jogador 2:

void frase_eeJ1()

void frase_ddJ1()

void frase_deJ1()

void frase_edJ1()

Descrição:

Estas funções realizam as frases das jogadas do jogador 2, com principal funcionalidade descrever a situação do jogo em tempo real, estas, criam uma melhor interação da pessoa com o programa.

Jogada igualdade Jogador 1

void igualdade_J1()

void aoCalhas_igualdade_J1()

Jogada igualdade Jogador 2

void igualdade_J2()

void aoCalhas_igualdade_J2()

Descrição:

Estas funções realizam a jogada *igualar*. Estas, verificam se as mãos apresentam as condições necessárias para a jogada ser realizada.

Foi criada uma função específica para o ao-calhas, devido a um erro de apresentação no terminal, foi necessária a sua criação para a correção do erro.

Apresentação do jogo no terminal

```
void verificaVez_J1( )
```

```
void verificaVez_J2( )
```

Descrição:

Estas funções realizam a apresentação do jogo no terminal, os dedos dos jogadores, e os seus respectivos nomes.

Algumas funções e partes do código importantes

```
time_t t;
```

```
srand((unsigned)time(&t));
```

```
n = rand() % 5;
```

A função `time()` retorna a hora atual em formato de contagem de segundos e é chamada com a variável `t` como parâmetro.

A função `srand()` é usada para inicializar a semente para a função `rand()`.

A variável '`n`' irá receber valores aleatórios de 0 a 4 devido ao `rand()`.

```
sleep(1);
```

Esta função é usada para parar a execução do programa durante um determinado periodo de tempo, neste caso, 1 segundo.

Complicações

- Houve alguma dificuldade ao passar as jogadas entre as funções, e para tal foram implementados apontadores, isto para tornar as funções das jogadas de certa forma mais fáceis e rápidas de usar, facilitando assim a alteração dos valores das variáveis guardados em memória.
- O código de igualdade do ao-calhas no printf da jogada, ao ser compilado, aparentava erro e não aparecia no terminal, logo teve-se de alterar a função de maneira a corrigir esse erro.
- Também houve problemas com o espaçamento das palavras dos jogadores no terminal e, para não haver o espaço, criou-se um string *estetica* que tem o mesmo nome da string com o espaçamento mas, sem o espaçamento o que acabou por resolver o espaço no terminal.
- A função rand não estava a “randomizar” corretamente e para tal usou-se o srand: *time_t t;*

srand((unsigned)time(&t));

Conclusão

Neste trabalho tentámos transmitir os nossos conhecimentos adquiridos em aula com o objetivo de cumprir da melhor forma possível todos os objetivos enunciados.

Este trabalho foi importante para a nossa compreensão, lógica e escrita, de código em linguagem C, além de nos ter permitido desenvolver as nossas competências em programação.