## Enunciado do Projeto

Laboratório de Algoritmia I

Laboratórios de Informática II

Ano letivo 2019/20

Last Update: 04/17/2020 19:01:13

## Plágio

Ao colocar o código num repositório público, é sempre possível que alguém possa copiar o código de outro projeto. De acordo com o código de ética da Universidade do Minho, tal ação é considerada plágio. Caso haja dúvida em relação a plágio, a equipa docente irá utilizar informação sobre os *commits* para decidir quem produziu o código e quem o copiou. Caso a equipa docente decida que houve plágio, os grupos que sejam considerados autores de plágio terão **ZERO** na componente de projeto.

# Constituição dos grupos

- Os grupos deverão ser de 3 elementos, pertencendo todos ao mesmo turno prático.
- A constituição dos grupos deverá ser comunicada ao docente nas aulas da semana de 2 e 6 de Março através de uma folha com os números dos alunos e os nomes completos.
- O projeto será avaliado nas aulas da semana de 4 a 8 de Maio.

## Descrição

Pretende-se implementar o jogo Rastros. Para descobrir as regras do jogo, clique <u>aqui</u>. O seu programa deverá permitir que dois jogadores, partilhando o mesmo teclado, o possam jogar. Os comandos deverão ser introduzidos através de um interpretador de comandos.

## Interpretador de comandos

O seu programa deverá funcionar como um interpretador de comandos. Esta estratégia costuma ser chamada REPL (Read, Evaluate, Print, Loop). O seu interpretador deverá responder aos seguintes comandos:

#### coordenada

Tal como nas coordenadas de um tabuleiro de Xadrez: uma letra (a coluna) seguida de um algarismo (a linha). Neste caso, o programa deverá efetuar essa jogada desde que ela seja válida.

gr nome do ficheiro

O programa deverá gravar o estado do tabuleiro num ficheiro

ler nome\_do\_ficheiro

O programa deverá ler o estado do tabuleiro a partir do ficheiro

movs

O programa deverá imprimir a lista dos movimentos do jogo

jog

O seu programa deverá escolher a melhor jogada para o jogador atual pos numero\_da\_jogada

O seu programa deverá permitir visualizar uma posição anterior através do seu número.

Q

O programa deverá acabar

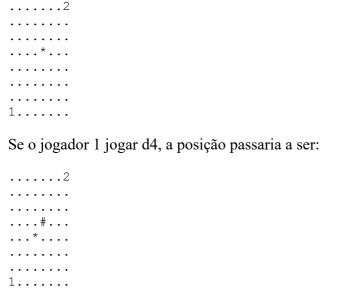
## Formato do ficheiro

O formato do ficheiro deve ser o seguinte:

- O estado do tabuleiro
- Uma linha em branco
- A lista de movimentos

#### Formato do estado do tabuleiro

O estado do tabuleiro deverá ser de oito linhas. Cada casa vazia contém um ponto, uma casa de uma jogada antiga um cardinal e a casa atual um asterisco. Para ajudar a reconhecer as posições vencedoras, o canto inferior esquerdo contém um 1 (posição vencedora para o jogador 1) enquanto que o canto superior direito contém um 2. Assim, a configuração inicial deve ser a seguinte:



#### Lista de movimentos

A lista de movimentos deverá ser dada por tantas linhas como jogadas. Cada linha começa com o número da jogada (sempre com dois algarismos) seguida dos movimentos de cada jogador. Assim se os jogadores fizeram os seguintes movimentos: d4, e4, e3, a lista de movimentos seria a seguinte:

```
01: d4 e4 02: e3
```

### **Exemplo completo**

Segue um exemplo completo do estado do ficheiro onde foram executados os seguintes comandos:

```
d4
e4
e3
f4
f3
gravar pos
```

A seguir a estes comandos, este deveria ser o conteúdo do ficheiro **pos**:

03: f3

# Visualização de posições anteriores

A visualização de posições anteriores será feita conforme o número da jogada, sendo 0 a posição inicial. Usando o exemplo completo da secção anterior, o comando **pos 2** deveria mostrar o seguinte tabuleiro:



## Bot

A título de bonificação, os grupos que quiserem deverão submeter o seu programa para um torneios. Cada programa deverá:

- 1. Ler o estado do jogo a partir de um ficheiro. O nome desse ficheiro deverá ser passado como parâmetro ao programa;
- 2. Efetuar a melhor jogada no menor tempo possível;
- 3. Gravar o estado do jogo num ficheiro. O nome desse ficheiro deverá ser passado como parâmetro ao programa.

Assim, se o programa for invocado da seguinte forma:

```
./bot jog01 jog02
```

Então ele deverá ler o estado do ficheiro jog01, jogar e gravar o estado no ficheiro jog02.

## **Torneio**

O torneio funcionará com as seguintes regras:

- Para participar no torneio, será necessário:
  - Criar o bot
  - O bot deverá respeitar o formato do ficheiro
- Caso um bot não faça uma jogada válida, ele perde automáticamente o jogo
- Caso um bot não faça uma jogada em 1 segundo de CPU, ele perde automáticamente o jogo
- Serão sorteados aleatóriamente jogos entre dois jogadores
- Cada jogo sera eliminatório, isto é, só o vencedor passará à fase seguinte

A avaliação do torneiro é feita da seguinte forma:

# Designação Avaliação Entrar no torneio 0.5 pontos oitavos de final 0.25 pontos quartos de final 0.25 pontos meias finais 0.25 pontos finais 0.25 pontos campeão 0.5 pontos

# Entrega

Para além do código, é obrigatório existir um ficheiro no projeto CLion com o nome grupo.txt com uma linha por cada elemento do grupo com o número seguido do nome completo. A entrega será através da exportação do projeto no CLion para um arquivo ZIP. O nome do arquivo deverá ter o seguinte formato:

<nome da UC>PL<número do turno prático>G<numéro do grupo com dois algarismos>.zip

O número nos casos dos grupos com um só algarismo é obtido colocando um zero como padding. Assim:

la1PL2G03.zip

Será o projeto do grupo 3 do turno PL2 de Laboratório de Algoritmia I li2PL4G09.zip

Será o projeto do grupo 9 do turno PL4 de Laboratórios de Informática II

Esse arquivo deverá ser submetido no blackboard até ao dia 2 de Maio. O projeto só será aceite se se respeitarem estas regras.