

## Trabalho Prático 1

**Prazo de Entrega:** 05 de Novembro de 2012, 23h 59min via Prático

**Valor:** 15 pontos

### Objetivo

O objetivo deste trabalho é simular um jogo de tabuleiro e a interação de um ou mais agentes (jogadores) sobre ele. O tabuleiro será composto por casas, identificadas por códigos únicos, e cada casa pode ter uma regra de navegação específica. As regras de navegação indicam a ação que um jogador deve tomar ao cair numa determinada casa do tabuleiro.

### Regras do Jogo

Ao receber os parâmetros de entrada, o programa deve ser capaz de interpretar a descrição do tabuleiro e processar as jogadas seguindo as regras de navegação estabelecidas. No arquivo de entrada serão descritas as jogadas, que são representadas por valores obtidos ao jogar o dado, que serão usadas para movimentar o agente de cada jogador no tabuleiro, sendo que o valor do dado deve refletir o número de movimentos para frente que o jogador deve fazer, a partir de sua posição inicial. A próxima casa na sequência do tabuleiro será dada pelo segundo identificador no arquivo de entrada na linha da casa atual. Caso o identificador da próxima casa seja 0, esta casa representa uma casa final do jogo, encerrando o jogo. Caso um jogador encontre uma casa final, nenhuma outra jogada deve ser avaliada. O jogo deve ser finalizado.

Após executar uma jogada, o programa deve avaliar a regra de navegação da casa, descrita na seção “Entrada”. Caso a regra de navegação da casa seja 0, nenhuma ação é necessária após a jogada. Caso seja identificada alguma regra diferente de 0, o jogador deve ser automaticamente direcionado a casa com o identificador informado pela regra. Ao ser direcionado a uma nova casa por uma regra de navegação, o jogador não deve avaliar a regra de navegação da casa subsequente e deve encerrar aquela jogada.

### Entrada

A entrada do programa será feita através de um arquivo de entrada, passado como parâmetros durante a execução utilizando a flag abaixo:

**-[iI]**<sup>1</sup> : arquivo de entrada com o mapeamento do tabuleiro e descrição das jogadas

O arquivo de entrada será composto de dois blocos de informação: um bloco que define o mapeamento do tabuleiro e outro bloco que define o conjunto de jogadas feitas pelo agente dos jogadores. O bloco de definição do tabuleiro sempre será iniciado pelo identificador “TABULEIRO”<sup>2</sup> seguido de uma quebra de linha. Em seguida, será informado um conjunto de 2 ou mais linhas com a descrição do tabuleiro. Cada linha terá 4 identificadores,

---

<sup>1</sup> Os parâmetros podem ser informados com letras maiúsculas ou minúsculas

<sup>2</sup> O identificador é *case sensitive*, ou seja, as letras maiúsculas são diferenciadas das letras minúsculas.

segundo o formato abaixo:

(Casa) (ProximaCasa) (RegraCasa) (Energia)

Os identificadores das casas no tabuleiro serão campos texto, sem a presença de caracteres especiais e acentos. Cada linha de descrição do tabuleiro serão separadas por uma quebra de linha. Os identificadores de cada linha representam:

Casa : identificador da casa descrita por esta linha no arquivo de entrada.

ProximaCasa : identificador da próxima casa do tabuleiro, a partir da casa descrita por esta linha.

RegraCasa : quando o jogador, após sua jogada, cair na casa descrita por esta linha, ele deverá ser automaticamente direcionado para a casa deste identificador<sup>3</sup>.

Energia<sup>4</sup> : Número inteiro que representa os pontos de vida que esta casa consome ou fornece ao jogador.

O bloco de descrição das jogadas será iniciado pelo identificador “JOGADAS”<sup>2</sup>, seguido de uma quebra de linha e um conjunto de 1 ou mais linhas no seguinte formato:

(Jogador) (NumeroDoDado)

Os identificadores dos jogadores serão campos texto, sem a presença de caracteres especiais e acentos. Cada linha de descrição das jogadas serão separadas por uma quebra de linha. Os identificadores de cada linha representam:

Jogador: campo texto que define o identificador de um jogador

NumeroDoDado: número inteiros maiores que zero, que representam um lançamento do dado para um jogador, que será usado para movimentar seu agente no tabuleiro.

A última linha do arquivo de entrada será a palavra “FIM”<sup>2</sup>, seguido de uma quebra de linha, marcando o final da leitura do arquivo.

## **Saída**

A saída do programa deve ser direcionada a um arquivo de saída cujo caminho será fornecido através do flag abaixo:

-[o|O] : endereço do arquivo de saída

O arquivo de saída deverá imprimir o identificador de cada jogador seguido de um espaço e a casa final deste jogador, seguida de uma quebra de linha, após avaliadas as jogadas. Na linha seguinte, deverá ser impresso a mensagem “Pontos de vida (jogador) 5”, seguido de uma

---

<sup>3</sup> Ao ser aplicado esta regra de navegação, a regra de navegação da próxima casa não deve ser considerada.

<sup>4</sup> Verificar seção “Extra” para tratamento desta informação

quebra de linha, sendo que “(jogador)” representa o identificador do jogador<sup>5</sup>. A lista das casas finais dos jogadores deve ser ordenada segundo a primeira aparição dos jogadores no arquivo de entrada.

Após a saída das casas finais dos jogadores, caso um jogador termine as jogadas numa casa final do jogo, deve-se imprimir a mensagem “Ganhei”<sup>6</sup>, seguido do identificador do jogador e de uma quebra de linha.

### Exemplo de Entrada e Saída

Entrada	Saída
TABULEIRO 1 2 0 0 2 3 4 0 3 4 1 0 4 5 7 0 5 6 10 0 6 7 0 0 7 8 6 0 8 9 11 0 9 10 5 0 10 11 0 0 11 12 0 0 12 0 0 0 JOGADAS 1 3 2 1 1 2 2 2 1 4 2 1 1 1 2 1 1 2 2 3 1 3 2 3 FIM	1 12 Pontos de vida 1 5 2 5 Pontos de vida 2 5 Ganhei 1

### Premissas

1. Não existe ordem pré-definida da ocorrência do bloco de descrição do tabuleiro e das jogadas no arquivo de entrada
2. Não existe ordem pré-definida da ocorrência dos parâmetros de entrada e saída
3. Não existe limite do número de jogadores no jogo

---

<sup>5</sup> Esta informação é necessária para manter a consistência com a saída descrita na seção “Extra”, e será sempre obrigatória.

<sup>6</sup> *Case sensitive*

4. Todas as jogadas serão descritas por números inteiros maiores que zero
5. O agente de cada jogador começará sempre na primeira casa descrita no bloco de descrição do tabuleiro, independente do seu identificador
6. Todas as casas terão identificadores únicos
7. O identificador da próxima casa de uma linha do arquivo de entrada sempre terá valor 0 ou um identificador de uma casa existente
8. O identificador da regra de navegação de uma linha do arquivo de entrada sempre terá valor 0 ou um identificador de uma casa existente
9. Todos os arquivos de entrada terão um único bloco de descrição de tabuleiro e um único bloco de descrição de jogadas
10. Nenhuma informação virá após o identificador "FIM" do arquivo de entrada
11. As jogadas devem ser processadas na ordem que aparecem no arquivo de entrada

## Documentação

A documentação do trabalho deve conter, ao máximo, 5 folhas, incluindo elementos não-textuais como capa, sumário, figuras, tabelas e referências. A documentação deve ser organizada e mostrar claramente os seguintes tópicos:

- Descrição do problema: apresentar qual o problema a ser tratado
- Implementação: descrever o algoritmo utilizado bem como as principais estruturas de dados
- Análise de Complexidade: apresentar a análise de complexidade em função do tempo na notação  $O$
- Testes: apresentar o conjunto de testes que validam a solução
- Conclusão: apresentar os ganhos e dificuldades no trabalho de forma clara e detalhada
- Referências bibliográficas: referências utilizadas no desenvolvimento do trabalho

## Extra (2 pontos) - Obrigatório para CC, SI e MatComp

A implementação extra deste trabalho consiste na avaliação da lógica de ganho/perda de pontos de vida do agente durante o jogo.

Cada jogador deverá começar com 5 pontos de vida e, ao cair em uma casa, deve avaliar os pontos de Energia que a casa corrente possui. Caso o valor seja um número positivo, o jogador irá adicionar os pontos a seus pontos de vida. Caso o valor seja um número negativo, o jogador irá subtrair os pontos de seus pontos de vida. O valor 0 indica que não há perda ou ganho de energia.

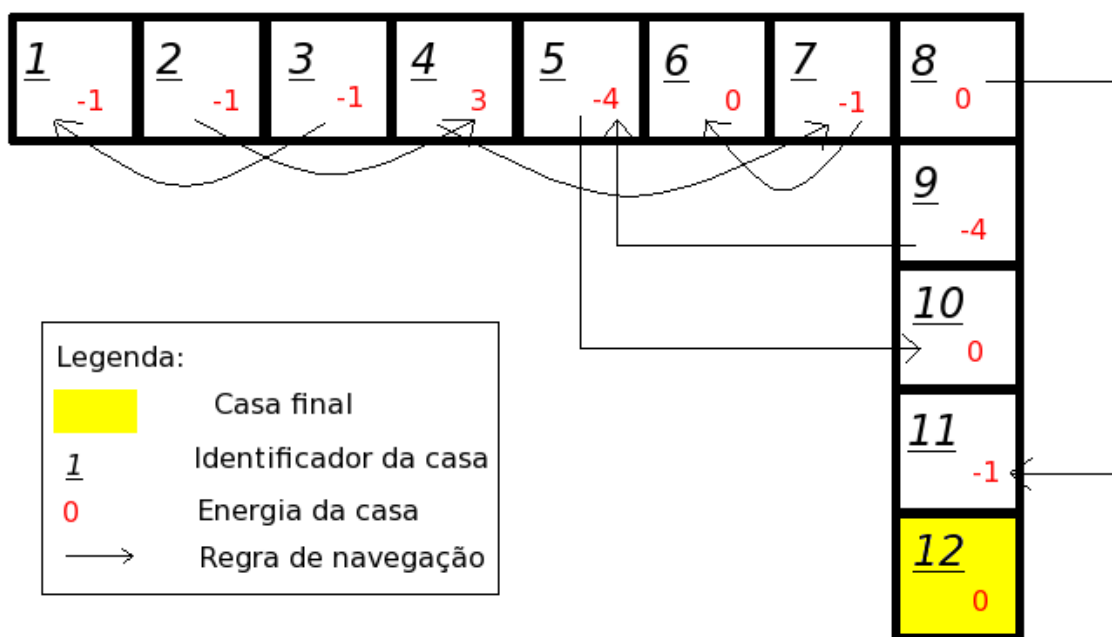
Caso o jogador, ao término da jogada, fique com um valor de pontos de vida menor ou igual a zero, o jogo terminará com a morte de seu agente e nenhuma outra jogada deste jogador deve ser processada. Neste caso, o jogador também não deve obedecer a regra de navegação da casa, permanecendo na casa em que morreu.

A saída do programa deve exibir, após informar a casa final que o jogador terminou o jogo, a mensagem "Pontos de vida" seguido de espaço, do identificador do jogador, espaço em branco e do valor de pontos de vida do jogador ao final da execução.

Em caso de morte, ao final da saída padrão, o jogador deve imprimir a mensagem “Morri”<sup>7</sup>, seguido do identificador do jogador e uma quebra de linha. Caso mais de um jogador morra durante o jogo, a saída deve exibir a lista de jogadores mortos ordenada pela primeira aparição de cada jogador no arquivo de entrada.

Caso o jogador morra numa casa final do jogo, o programa não deve considerar este jogador como o vencedor da partida, e o processamento das jogadas deve continuar.

Soluções sem esta lógica implementada podem desconsiderar o campo de Energia, presente no arquivo de entrada.



Entrada	Saída
---------	-------

<sup>7</sup> Case sensitive

TABULEIRO 1 2 0 -1 2 3 4 -1 3 4 1 -1 4 5 7 3 5 6 10 -4 6 7 0 0 7 8 6 -1 8 9 11 0 9 10 5 -4 10 11 0 0 11 12 0 -1 12 0 0 0 JOGADAS 1 3 2 1 1 2 2 2 1 4 2 1 1 1 2 1 1 2 2 3 1 3 2 3 FIM	1 9 Pontos de vida 1 0 2 9 Pontos de vida 2 -2 Morri 1 Morri 2
--	---

### Extra (1 ponto)

Será atribuído um ponto extra para documentações feitas utilizando a linguagem LaTeX.