

Trabalho Prático – Jogo de Damas

Disciplina Fundamentos de Programação (CK0211) – Semestre 2019.1

Prof. Miguel Franklin

Desenvolver um programa em linguagem Python representando o jogo de Damas, em dimensão 10x10, com dois modos de execução: “USUÁRIO CONTRA USUÁRIO” e “OFFLINE”. No modo USUÁRIO CONTRA USUÁRIO, dois usuários poderão jogar entre si, entrando alternadamente suas jogadas. No modo OFFLINE, o usuário deverá informar um arquivo texto de entrada contendo as jogadas e o computador deverá apresentar a situação final do tabuleiro após a execução das jogadas contidas no arquivo. Caso o usuário ou o arquivo entre um movimento inválido, o programa deve informar esta condição de erro e não efetuar o movimento.

```

A B C D E F G H I J
+---+---+---+---+---+
0|#|o|#|o|#|o|#|o|#|o|0
+---+---+---+---+---+
1|o|#|o|#|o|#|o|#|o|#|1
+---+---+---+---+---+
2|#|o|#|o|#|o|#|o|#|o|2
+---+---+---+---+---+
3| |#| |#| |#| |#| |#|3
+---+---+---+---+---+
4|#| |#| |#| |#| |#| |4
+---+---+---+---+---+
5| |#| |#| |#| |#| |#|5
+---+---+---+---+---+
6|#| |#| |#| |#| |#| |6
+---+---+---+---+---+
7|@|#|@|#|@|#|@|#|@|#|7
+---+---+---+---+---+
8|#|@|#|@|#|@|#|@|#|@|8
+---+---+---+---+---+
9|@|#|@|#|@|#|@|#|@|#|9
+---+---+---+---+---+
A B C D E F G H I J

```

Figura 1

Requisitos Gerais

G.1. Caso o usuário execute o programa informando diretamente um nome de arquivo a ser lido, o programa deverá executar no modo OFFLINE e ler o arquivo cujo nome foi informado pelo usuário. Exemplo: quando o usuário, no Terminal, chama: `> python damas.py entrada.txt`, o programa `damas.py` deverá abrir e ler o arquivo chamado `“entrada.txt”`.

G.2. Caso o usuário execute o programa sem informar nenhum nome de arquivo como parâmetro, o programa deverá executar em modo USUÁRIO CONTRA USUÁRIO.

Requisitos – MODO USUÁRIO CONTRA USUÁRIO:

U.1. O programa deve apresentar o tabuleiro inicial, conforme exemplo da Figura 1.

U.2. O computador deve ser informado sobre qual usuário deve fazer a jogada: o usuário de cima (“o”), entrando a letra “C”, ou o usuário de baixo (“@”), entrando a letra “B”.

U.3. Na vez de cada usuário, uma jogada deverá ser entrada. Cada jogada é composta da posição inicial e da posição final de uma peça, no modo `“<COLUNA_INICIAL><LINHA_INICIAL>--<COLUNA_FINAL><LINHA_FINAL>”`, conforme mapeado no exemplo da Figura 1, sem nenhum espaço entre os caracteres da jogada. São

exemplos de jogadas: “B0--C1”, “I7--H6”.

- U.4. As peças podem ser de dois tipos: normais (“@” ou “o”) ou damas (“&” ou “O”). As peças normais só podem caminhar na diagonal, um salto por vez, sempre para a frente. Uma dama pode caminhar na diagonal, um ou mais saltos por vez, para frente ou para trás, desde que o caminho esteja livre.
- U.5. Uma peça normal se torna dama quando esta chega ao outro extremo do tabuleiro. Isto é, para o jogador de cima, quando a peça atinge a linha 9, e para o jogador de baixo, quando atinge a linha 0.
- U.6. Uma peça normal pode “comer” uma peça adversária saltando para a posição seguinte à peça adversária, que deve ser retirada do tabuleiro e contabilizada para o usuário que fez a jogada.
- U.7. Apenas na ocasião de “comer” uma peça adversária, uma peça normal pode fazer um movimento

para trás (“comer para trás”).

- U.8. Uma dama pode “comer” apenas uma peça de cada vez, que esteja em uma de suas diagonais. O caminho até a peça a ser comida deve estar livre e deve haver pelo menos um espaço livre após a peça comida. A dama não é obrigada a parar na casa logo após a peça comida, desde que ainda haja posições livres na mesma diagonal.
- U.9. O usuário que comer uma peça adversária deverá jogar novamente logo em seguida.
- U.10. O programa deverá verificar a validade da jogada entrada. Caso a jogada seja inválida, o programa deverá mostrar uma mensagem “Jogada Inválida”, e deverá solicitar uma nova jogada do mesmo jogador;
- U.11. São jogadas inválidas:
 - a. Quando o usuário entra as coordenadas de uma posição inicial que não tem nenhuma peça, ou tem uma peça que não é sua;
 - b. Quando o usuário indica um movimento de mais de um salto para peças normais (excetuando-se o movimento de “comer” uma peça);
 - c. Quando o usuário indica um movimento que não é em diagonal;
 - d. Quando o usuário indica um movimento onde a posição final da peça já está ocupada;
 - e. Quando o usuário indica um movimento de uma peça normal onde a posição final fica para trás;
 - f. Quando o usuário indica um movimento de comer peça que não é válido, segundo os requisitos listados acima.
- U.12. Quando um usuário se deparar com uma situação onde alguma das peças dele puder comer uma peça adversária, ele deverá obrigatoriamente realizar esta jogada de comer a peça adversária. Caso haja mais de uma possibilidade como esta em uma jogada, o usuário poderá escolher qual comerá primeiro.
- U.13. Quando todas as peças de um jogador são comidas, o programa deve contabilizar a vitória do oponente e perguntar se o usuário gostaria de recomeçar o jogo.

Requisitos – MODO OFFLINE:

- O.1. O programa deve ler o arquivo texto cujo nome foi informado no momento da chamada de execução do programa;
- O.2. No arquivo texto, haverá, na primeira linha, apenas o caractere “C” (cima) ou o caractere “B” (baixo), indicando qual jogador deve iniciar a partida;
- O.3. A partir da segunda linha, o arquivo texto deverá encontrar as jogadas, uma por linha, no mesmo modelo citado no item U.3;
- O.4. A cada jogada lida do arquivo, o programa deverá testar a validade da jogada, como descrito nos itens de requisitos do modo USUÁRIO CONTRA USUÁRIO;
- O.5. Caso o programa encontre, no arquivo de entrada, uma jogada inválida, o programa deverá mostrar a mensagem “Jogada inválida na linha <N> do arquivo de entrada.”, onde <N> deve ser substituído pelo número da linha no arquivo onde a jogada inválida foi encontrada, considerando que a primeira linha, que define o primeiro jogador, conta como a linha número 1. Em seguida, o usuário deverá retomar as jogadas a partir da linha seguinte;
- O.6. O programa deverá ler todas as linhas do arquivo, que deverá terminar com um caractere de retorno, isto é, após a última jogada listada no arquivo, deve haver um <Enter>;
- O.7. Após lido todo o arquivo de entrada e registradas todas as jogadas válidas, o programa deverá mostrar a situação final do tabuleiro, seguindo o exemplo da Figura 1, mostrando também quantas peças cada usuário comeu: “Cima = <X> / Baixo = <Y>”.
- O.8. Se, ao final, algum usuário tiver ganho o jogo, o programa deverá mostrar a mensagem: “O vencedor é o usuário [de CIMA] <ou> [de BAIXO].”

ATENÇÃO:

Nas primeiras linhas do código fonte do programa, deverá constar os nomes completos e números de matrícula dos componentes da equipe, em comentários.

Serão aceitas equipes de **ATÉ 3 (TRÊS) alunos**.

Critérios de Avaliação

A avaliação será realizada em duas fases:

1. Análise do código-fonte;
2. Análise da execução do programa (teste), nos modos ONLINE e OFFLINE.

O código-fonte será avaliado de acordo com os seguintes critérios qualitativos:

- i. Eficácia do programa em suprir todos os requisitos;
- ii. Eficiência do programa (otimização);
- iii. Organização do código (uso racional de subprogramas, estruturas, etc.);
- iv. Legibilidade do código (uso de endentação e semântica dos identificadores de variáveis);
- v. Documentação (comentários dentro do código fonte).

Apenas será admitida programação em linguagem **PYTHON**, sendo atribuída nota **ZERO** caso este requisito não seja respeitado.

Os trabalhos serão corrigidos no LINUX. Portanto, certifique-se que o trabalho feito no Windows também compila e roda no Linux.

Trabalhos cujos códigos fonte apresentem problemas que impeçam a sua execução terão nota **ZERO**. Não haverá, por parte da equipe de avaliação do trabalho, tentativa de correção do problema. Portanto, antes de fazer a entrega, certifique-se que o programa executa de forma limpa no LINUX.

Obviamente, funcionalidades adicionais às que foram solicitadas neste documento são bem vindas e serão gratificadas na nota. O código-fonte deve ser submetido na data fixada através do SIGAA.

Lembramos que todos os programas serão submetidos a análise léxica automática, que pode evidenciar cópia (plágio) de código. Nenhum tipo de cópia ou “inspiração” será tolerado. Caso ocorra, será atribuída nota **ZERO** às equipes com trabalhos semelhantes, e processo administrativo disciplinar poderá ser instaurado para apurar o plágio.

Prazo de Entrega: 21 de junho de 2019, até 23:59.

Upload no sistema SIGAA

Não serão aceitas entregas por e-mail (a não ser que haja algum problema com o SIGAA).