

TRABALHO PRÁTICO 2

Data de entrega: 26/11

Jogo do TOK

Você deve criar um programa em Java referente a um jogo que representa dois grupos da componente curricular de Resolução de Problemas (RP) competindo por um TOK (Total OK, que representa 100% de nota em uma atividade).

O Jogo do TOK é jogado em uma matriz 5 x 5, e cada jogadora controla 5 peças representando estudantes do seu grupo de RP. Além disso, há uma peça no centro do tabuleiro que representa o tão desejado TOK. O tabuleiro antes do começo do jogo deve estar da seguinte forma:

①	①	①	①	①
		TOK		
②	②	②	②	②

No exemplo acima, as peças estão sendo representadas por círculos numerados das jogadoras 1 e 2 e a peça no TOK está representada pelo seu texto, mas você pode usar o tipo de peça que quiser.

Todas as peças se movem somente em linha reta (horizontalmente e verticalmente), mas elas devem se mover o máximo de quadrados vazios que puderem até encontrarem um obstáculo (a borda do tabuleiro, outra peça ou o TOK). Os exemplos abaixo indicam os movimentos válidos para a peça da jogadora 1 destacada em verde.

	①	①	②	①
TOK	x	①		x
	②	x		①
②	②	②		

①	①		x	①
①		②		TOK
②				
	②	x	①	x
②		②	x	

Não existe captura de peças nesse jogo, então nenhuma peça desaparece ao longo da partida.

O esquema do jogo é o seguinte:

1. O jogo começa com a jogadora 2 (das peças de baixo) movendo uma das peças dela.
2. O próximo movimento é da jogadora 1 que move primeiro o TOK (que se movimenta da mesma forma que as outras peças) e depois uma das peças dela.
3. O movimento seguinte é da jogadora 2 que move primeiro o TOK e depois uma das peças dela.
4. O jogo segue dessa forma, alternando os turnos das jogadoras: primeiro move o TOK e depois move uma das peças daquela jogadora em questão.

O objetivo do jogo é mover o TOK para a linha inicial das suas peças (a primeira linha no caso da jogadora 1 e a última linha no caso da jogadora 2), forçar a oponente a mover o TOK para a linha inicial de suas peças (toda jogadora é obrigada a mover o TOK na vez dela) ou bloquear completamente o movimento do TOK na vez da oponente.

Ou seja, a jogadora 1 vence se:

1. Mover o TOK para a primeira linha.
2. Forçar a jogadora 2 a mover o TOK para a primeira linha.
3. Bloquear totalmente o movimento do TOK na vez da jogadora 2.

O contexto de vitória da jogadora 2 é análogo aos itens acima.

O jogo deve impedir movimentos inválidos das jogadoras e detectar automaticamente a condição de vitória. Uma mensagem deve ser mostrada no final do jogo, indicando a ganhadora.

Você está livre para fazer a interface gráfica com usuário (GUI) da forma que quiser, mas o tamanho mínimo da janela é 800 x 600 *pixels*.

Finalmente, o jogo deve possuir uma barra de menus contendo dois menus diferentes. O primeiro menu será “jogo” e terá as opções de “reiniciar” e “sair”. A opção de reiniciar volta o jogo ao estado inicial e pronto para jogar, e a opção de sair termina o programa. O segundo menu será “autores” e terá apenas a opção de “ver nomes”. A opção de ver nomes dos autores mostrará (você pode escolher a melhor forma de fazer isso) os nomes das pessoas autoras do jogo.

OBSERVAÇÃO 1: ao menos uma classe deve ser usada para guardar o estado do jogo (elementos do tabuleiro e suas posições, jogadora da vez, se a jogadora da vez deve mover o TOK ou uma peça comum etc.).

OBSERVAÇÃO 2: o jogo deve usar tratamento de exceções para implementar o sistema de jogadas inválidas.

OBSERVAÇÃO 3: depois da entrega, pode ser necessário apresentar o código-fonte do trabalho para o professor em uma das aulas ou em horário extraclasse combinado previamente.

RUBRICA DE AVALIAÇÃO

Verifique como seu trabalho será avaliado conforme os critérios estabelecidos abaixo. Se tiver dúvidas, entre em contato com o professor.

Critérios	Expectativa de desempenho para a entrega		
[10 pontos] Funcionamento geral do jogo	100% O jogo não tem erros de execução.	50% O jogo às vezes tem erros durante a execução, mas esses erros não encerram o programa.	0% O jogo apresenta erros que encerram o programa; ou o jogo não é jogável (não é possível fazer nada).
[10 pontos] Movimentos das peças	100% É possível selecionar uma peça e a mover para uma posição selecionada.	50% Há problemas na seleção de uma peça ou na escolha do destino (e.g., precisa clicar várias vezes para selecionar), mas é possível ainda assim mover uma peça para uma posição desejada.	0% Não é possível mover as peças; ou ocorrem erros quando uma peça é movida que levam o jogo a um estado indesejado.
[10 pontos] Movimentos inválidos	100% Movimentos inválidos não são permitidos e o controle é feito por meio de exceções.	50% Movimentos inválidos não são permitidos e o controle é feito por testes com variáveis comuns.	0% Movimentos inválidos podem ocorrer e não são detectados.
[10 pontos] Turnos	100% A troca de turnos acontece automaticamente, sem erros e controlada por uma classe que representa o estado do jogo.	50% A troca de turnos acontece automaticamente, mas não existe uma classe para o estado de jogo; ou é necessário indicar explicitamente de quem é a vez de jogar.	0% Não há qualquer controle sobre de quem é a vez de jogar; ou ocorrem erros no sistema de turnos.
[10 pontos] Fim de jogo	100% O fim de jogo é detectado automaticamente e a indicação de vitória é exibida.	50% O jogo às vezes não detecta corretamente o fim do jogo; ou o jogo não consegue saber quem ganhou exatamente.	0% O jogo não detecta condições de fim.
[10 pontos] Menus	100% Os menus implementam corretamente as funcionalidades de novo jogo, sair e mostrar pessoas autoras.	50% Os menus não implementam uma das funcionalidades previstas.	0% Não existem menus; ou os menus não implementam duas ou mais das funcionalidades previstas.
[20 pontos] Aplicação de POO	100% Uso de técnicas de encapsulamento e abstração adequadas para a solução do problema.	50% O código tem problemas de visibilidade de atributos e/ou métodos; ou o código usa somente atributos e/ou métodos estáticos; o código apresenta problemas sérios relacionados a orientação a objetos.	0% O jogo está em uma grande classe sem divisão adequada de responsabilidades em classes, objetos e métodos.
[10 pontos] Qualidade de código	100% O código segue convenções da linguagem e é muito claro e/ou apresenta comentários ¹ suficientes para entendimento completo.	50% O código segue convenções da linguagem, mas é pouco claro e apresenta poucos comentários (insuficientes para entendimento completo do algoritmo).	0% O código não segue convenções de Java; ou o código é obscuro e não possui nenhum comentário.

¹ Comentários automáticos gerados pela IDE adotada não são suficientes para explicar código obscuro.