

Trabalho #02 - Jogo *Java of Empires*

Data de entrega: 24/11/2017 (até 23h55), via moodle.

O trabalho poderá ser individual ou em grupo de até 3 alunos.

O jogo *Java of Empires* apresenta uma arena bidimensional de 10×10 posições onde os personagens se movem para poderem atacar seus inimigos. Existem cinco tipos de personagens e a hierarquia de poder se dá na seguinte ordem, do menor para o maior: aldeão, arqueiro, guerreiro, cavaleiro e navio. Cada personagem criado no jogo possuirá uma identificação única (que deverá ser gerada automaticamente) e sua posição inicial (x,y) será definida de forma aleatória, não devendo ser uma posição já ocupada na arena por outro personagem.

Os personagens se movem em 4 direções (norte \uparrow , sul \downarrow , leste \rightarrow , oeste \leftarrow) de acordo com suas respectivas quantidades de passos:

- Aldeão: de uma em uma posição,
- Arqueiro: de duas em duas posições,
- Guerreiro: de três em três posições,
- Cavaleiro: de quatro em quatro posições,
- Navio: de cinco em cinco posições.

Ao se mover, o personagem não poderá ir além dos limites da arena. A frente de cada personagem estará apontada para a direção de seu último movimento. Ao se mover, caso o personagem invada a posição de outro personagem mais forte que ele, o mesmo é excluído do jogo. Caso contrário, o movimento será inválido.

Os personagens se movem pela arena buscando posições de ataque aos oponentes. A maneira como cada personagem ataca se dá de acordo com uma hierarquia de força:

- O aldeão ataca à sua frente com alcance de uma posição, podendo abater (excluir) somente outros aldeões.
- O arqueiro ataca à sua frente com alcance de 5 posições e pode abater apenas aldeões e outros arqueiros. Caso outros personagens mais fortes estejam na direção do ataque, os mesmos não serão abatidos.
- O guerreiro ataca em duas direções, à frente e à direita, com alcance de 3 posições, podendo abater aldeões, arqueiros e outros guerreiros. Por exemplo, se a frente do guerreiro estiver voltada à norte, ele poderá abater inimigos nas direções norte e leste. Se a frente estiver para oeste, abaterá à oeste e norte e assim sucessivamente.
- O cavaleiro ataca em três direções, à frente, à direita e à esquerda, com alcance de 5 posições, podendo abater aldeões, arqueiros, guerreiros e outros cavaleiros.

- O navio ataca em quatro direções, norte, sul, leste e oeste com alcance máximo, abatendo quaisquer outros personagens localizados na direção de sua linha e na direção de sua coluna.

Caso um personagem ataque outro que seja mais forte, o ataque não terá efeito sobre o mesmo.

Com base no detalhamento anterior, faça:

1. Descreva o diagrama UML das classes do jogo (gerar o arquivo pdf do diagrama)
2. Desenvolva um aplicativo Java com um menu interativo que permita ao usuário executar o jogo com no máximo 30 personagens:
 - (a) Criar 10 personagens
 - Deverão ser criados 10 personagens (2 de cada tipo)
 - A posição (x, y) de cada personagem deverá ser escolhida de forma aleatória, bem como a direção de sua frente.
 - (b) Incluir personagem (deve-se informar o tipo e a posição (x,y) do personagem)
 - (c) Movimentar personagem (deve-se indicar o id do personagem e a direção do deslocamento (norte, sul, leste ou oeste). A frente do personagem ficará voltada para a direção do movimento)
 - (d) Remover personagem (deve-se informar o identificador do personagem que será excluído do jogo)
 - (e) Atacar (deve-se informar o id do personagem que irá atacar)
 - (f) Gravar jogo no arquivo
 - (g) Ler jogo do arquivo
 - (h) Imprimir arena (imprime o identificador de todos os personagens em um mapa, como no exemplo abaixo:
 - O identificador do personagem na arena será composto pelo id do personagem concatenado com sua sigla (**A** para aldeão, **Q** para arqueiro, **C** para cavaleiro, **G** para guerreiro e **N** para navio) e concatenado com a sigla da direção de sua frente (**N** para norte, **S** para sul, **L** para leste e **O** para oeste). Por exemplo, a posição (2,4) no mapa está ocupada pelo personagem “10GO” que consiste de um guerreiro com id 10 e frente para oeste. Na posição (7,3) está “5CS” simbolizando um cavaleiro com id 5 e frente para sul.)

	0	1	2	3	4	5	6	7	8	9
0								2AL		
1			3QO							
2					10GO					
3		7CL								9QS
4						1NS				
5										
6	4GN									
7				5CS						
8									8NN	
9							6AO			

- (i) Sair do programa

Avaliação:

O trabalho será avaliado em função da:

- Correção (o aplicativo cumpre com as exigências);
- Documentação (o aplicativo está devidamente comentado);
- Paradigma orientado a objetos (o aplicativo está seguindo os princípios da programação OO);
- Modularidade (o aplicativo está bem estruturado onde necessário, com métodos (funções) parametrizados);
- Robustez (o aplicativo não trava em tempo de execução).

Detalhamento de itens a serem avaliados e seus respectivos pontos descontados:

Item	Pontos
Respeitar o encapsulamento de dados	-5
Uso de modificadores de acesso adequados (private e public)	-5
Criar getters e setters que forem necessários	-5
Criar métodos construtores parametrizados	-5
Fazer sobrecarga de pelo menos um método (qualquer um)	-3
Pelo menos um atributo deve ser final	-4
Fazer uso da palavra reservada this	-2
Pelo menos um atributo deve ser static	-4
Criar relacionamento entre classes (Agregação ou Composição)	-2
Fazer uso de classe abstrata	-5
Fazer uso do conceito de herança e polimorfismo	-10
Não utilizar modificador protected	-5
O aplicativo não deve travar em tempo de execução	-5
Apresenta diagrama UML	-5