Especificação do projeto 2

Programação Orientada à Objetos

O projeto deverá ser entregue no Moodle, em um único arquivo comprimido (.rar de preferência). Os códigos deverão seguir um mesmo estilo de código e deverão estar comentados.

1) Suponha que você está programando um jogo de estratégia estilo Age of Empires. No jogo, existem diversos tipos de objetos (divididos entre unidades e construções). Seu objetivo é modelar e programar as classes descritas abaixo usando orientação a objetos. **Não é necessário prever a execução do jogo**, apenas fazer a definição das classes e testá-las no método *main*. O importante é refletir sobre a modelagem do problema e fazer bom uso de conceitos de orientação a objetos em Java.

Todas unidades e construções deverão ter os seguintes atributos:

- Posição x e y (referente à posição da unidade ou das construções no mapa). As unidades podem ter sua posição alterada através do método *mover*, no entanto as construções não podem ter suas posições modificadas depois de sua construção.
- Imagem: não precisa guardar uma imagem, apenas uma String que guardará um nome fictício de um arquivo de imagem.
- Custo: quanto custa para construir esse objeto. O custo pode ser unidades de comida, madeira e/ou ouro.
- Estado: vivo ou morto.
- Pontos de vida: indica quantos pontos de dano o objeto pode receber antes de ser marcado como morto.

Outros atributos de alguns objetos (ver tabelas):

- Alcance: para utidades como arqueiro que atacam a distância.
- Armadura: um número de pontos de danos que é removido de cada ataque feito à unidade.
- Unidades que podem se mover terão o atributo velocidade: indica a velocidade com a qual a unidade pode se mover (double).
- atacar(Objeto unidadeAtacada): retira dos pontos vitais do objeto atacado o número de pontos de ataque do objeto atacante, menos o número de pontos da armadura do objeto atacado (caso o objeto tenha armadura). Em caso de objetos atacantes sem o atributo alcance, o objeto atacado somente poderá sofrer o ataque se estiver à 2 unidades de distância do objeto atacante (dica: pense em sobrescrita de métodos). Apenas objetos com ataque diferente de zero deverão ter esse método.

Todas unidades deverão ter os seguintes métodos:

- mover(String direção): modifica a posição da unidade, adicionando a velocidade da unidade à sua posição. Exemplo: se a posição atual é (10, 20) (ou seja, x=10 e y=20), e a direção é “Norte”, e a velocidade da unidade é 2 (por exemplo no caso do camponês), sua posição deve ser alterada para (10, 22), indicando que a unidade andou duas unidades para cima no mapa. Considere que x representa o eixo leste-oeste e y representa o eixo norte-sul.

Características das civilizações:

- Cada civilização pode ter uma certa quantidade inicial de comida, madeira e ouro, que são gastos a medida que a civilização cria objetos e recuperados através dos métodos dos objetos camponeses.
- Cada civilização deverá manter um array de unidades e um array de construções que estão “vivas”.
- Cada civilização deve começar com alguns camponeses e um centro da cidade.
- Quando a civilização não tiver nenhuma unidade ou construção, uma mensagem deverá ser impressa na tela e o atributo “extinta” da civilização deverá ser marcado como verdadeiro. A partir desse momento, criar qualquer tipo de unidade deverá ser impossível.
- As civilizações que deverão ser consideradas são: gregos e egípcios.
- Cada civilização tem um atributo que representa a população atual, que nunca pode ser maior que a capacidade da população (dado pelas casas e centros de cidade, ver descrição adiante).

Tabela de unidades a serem consideradas:

Tipo	Pontos vitais	Ataque	Custo	Alcance	Armadura	Velocidade
Camponeses	50	3	50C	0	0	2.0
Guerreiro	160	13	35C, 15O	0	2	1.8
Cavaleiro	180	12	70C, 80O	0	3	4.0
Arqueiro	45	5	40C, 20O	7	0	2.0
Sacerdote	25	0	125O	10	0	1.0
Elefante	600	18	170C, 40O	0	2	1.0
Falange	120	20	60C, 40O	0	7	1.2

*C indica comida, O indica ouro e M indica madeira.

Características especiais de algumas unidades:

- Sacerdotes possuem um método converteInimigo(Objeto objeto) que converte tanto unidades quanto construções inimigas para sua civilização.
- Camponeses possuem um método:
    - constroi(double x, double y, String tipoConstrucao) que cria uma construção do tipo

desejado.

  - colhe() que gera 1 unidade de comida para a civilização
  - corta() que gera 1 unidade de madeira para a civilização
  - mineira() que gera 1 unidade de ouro para a civilização
- Algumas unidades têm o atributo ataque: número de pontos de dano que a unidade pode fazer.

Tabela de construções a serem consideradas:

Tipo	Pontos vitais	Ataque	Custo	Alcance	Pode criar
Casa	75		30M		
Centro da cidade	600	10	200M	8	Campones
Templo	350		250M		Sacerdote
Quartel	350	0	125M		Guerreiro, Cavaleiro, Arqueiro, Elefante, Falange
Torre	200	20	70M	7	

Características especiais de algumas construções:

- Cada casa permite duas unidades de capacidade de população.
- Cada centro da cidade permite 10 unidades de capacidade de população.
- Algumas construções têm o método cria() que gera uma unidade nova do tipo indicado na tabela das construções. A unidade criada é colocada na mesma posição da construção

Características especiais das civilizações:

- Apenas egípcios podem gerar unidades do tipo Elefante
- Apenas gregos podem gerar unidades do tipo Falange

Avaliação

Os quesitos de avaliação serão:

- Bom uso dos conceitos de orientação a objetos, entre eles: herança, polimorfismo, sobrescrita e sobrecarga de métodos, atributos, métodos e modificadores de visibilidade (public, private e protected);
- Bom uso da linguagem Java;
- Comentários pertinentes no código;
- Identação e organização do código.

A entrega deverá consistir das classes criadas e de uma classe principal (com o método main) onde duas civilizações deverão ser instanciadas e alguns objetos deverão ser criados (via programa, sem interação com o usuário). Além disso, nesse mesmo trecho de código, todos os métodos e atributos especificados deverão ser testados, por exemplo:

- criar um objeto que tenha ataque à distância, tentar atacar uma unidade que está além do seu alcance, e mostrar que o ataque não gera efeito, além do caso contrário;
- verificar se o programa proíbe a criação de mais unidades do que a capacidade da população;
- verificar a conversão dos sacerdotes (uma unidade convertida deverá ser removida do array da civilização original e ser adicionada a civilização nova);
- verificar os métodos exclusivos do camponês.