Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação
Disciplina: Programação de computadores II
Professor: Reginaldo Santos

Neste trabalho, você deverá desenvolver um jogo de turnos na linguagem de programação **Java**, utilizando conceitos de programação orientada a objetos, como **abstração**, **encapsulamento**, **composição**, **herança** e **polimorfismo**. O jogo simulará batalhas entre heróis e monstros, com atributos para cada classe e mecânicas de combate *turn-based*. Além disso, você deverá modelar o jogo utilizando **diagramas UML** para representar as funcionalidades do sistema.

Requisitos do jogo de turno

1. Cada herói e monstro deverá ter, pelo menos, os seguintes atributos: nome, pontos de vida (HP), força de ataque, resistência a danos (defesa), destreza e velocidade. Terão no mínimo um construtor, métodos *getters* e *setters* para manipulação de atributos. Deve ter também no mínimo um método abstrato chamado *realizarAtaque()* para que as subclasses implementem o ataque. Uma possível hierarquia de herança seria: superclasse *Player* e subclasses *Hero* e *Monster*. Crie um método estático para gerar um *Player* aleatoriamente, podendo ser *Hero* ou *Monster*, e os valores de seus atributos serão gerados aleatoriamente dentro de um intervalo conveniente. É recomendável que se crie opções de heróis, pelo menos três: guerreiro, mago, arqueiro e furtivo. Cada um deve ter características próprias, como valores de atributos coerentes. Monstros diferentes fica por conta da criatividade da equipe.
2. Acomode o jogo dentro de uma classe *Game* que deverá armazenar o estado do jogo, nº de heróis, nº de monstros, *player* com maior/menor atributo específico no campo de batalha. Também é esperado que tenham métodos para iniciar e terminar o jogo, controle de fluxo, além de gerenciar os turnos e armazenar logs do jogo. Crie uma classe *Log* para atender este requisito. É importante que haja diferentes dificuldade no jogo, por exemplo, os níveis: fácil, médio e difícil, onde progressivamente apareçam monstros mais fortes a depender do nível.
3. É esperado que se acomode os turnos dentro de uma classe chamada *Turno*. Esta classe deve ser a responsável por realizar os turnos do jogo. Cada *player* deve realizar sua ação com base em seus atributos, tal como a velocidade que deve determinar a ordem de ataque de um jogador no mesmo turno. O resultado do ataque é determinado por um Enum.
4. O Enum deve modelar o resultado de um ataque com os valores: ERROU, ACERTOU, CRITICAL_HIT. Quando um *player* atacar outro, a destreza deverá ser levada em consideração. Uma destreza alta garante maior probabilidade de acertar o ataque.

5. A batalha deve ser simulada automaticamente e o jogo deve prover mecanismos para que seja possível visualizar o resultado do jogo e saber o que ocorreu ao longo da partida a partir dos logs.
6. Implemente uma inteligência artificial *naive* para os monstros, tais como: atacar o herói com menor HP ou defesa, atacar o mesmo herói, etc.
7. Realize um tratamento de exceção sempre que achar necessário.

O que deve ser entregue

A equipe deverá realizar uma apresentação sobre o trabalho desenvolvido no dia e horário estabelecidos pelo professor da disciplina. O objetivo da apresentação é demonstrar o processo de criação, desenvolvimento e funcionamento do jogo de turnos, além de evidenciar a aplicação dos conceitos de programação orientada a objetos (POO). A apresentação deverá ser objetiva e contar com a participação de todos os integrantes da equipe.

Aspectos esperados na apresentação: modelagem do jogo (diagramas UML e decisões de projeto), divisão de responsabilidades entre os integrantes da equipe, concepção do jogo via trechos do código-fonte que sejam relevantes para ilustrar, a lógica principal com o sistema de turnos, características de POO utilizadas, demonstração do jogo a partir de sua execução, seu funcionamento, possíveis limitações e sugestões de melhorias futuras.