

## Breve resumo das implementações

Houve três principais alterações em relação ao trabalho 2. A criação do enumerador de vocação nos Itens e Personagens, concepção de terrenos, o estabelecimento de batalhas com threads e a implementação da classe Pair com generics.

Com a gênese do atributo “vocation” foi possível diversificar e especializar os itens para cada tipo de classe (Wizard, Knight e Thief). Para que isto fosse devidamente implementado, foi necessário transformar as classes Armor e Weapon em abstratas, criar um enum “Vocation” e criar classes como Robe, Mail, Sword, entre outras, cujas instâncias só podem ser relacionadas a um tipo específico de personagem.

Foram criados terrenos (Desert, City, Forest) em que cada personagem recebe vantagens ou penalidades de acordo com sua classe. Ao início da batalha é decidido qual será o terreno e são atribuídas as devidas consequências. Mas há casos em que não existem alterações (por exemplo, Knights não possuem vantagens nem recebem penalidades na cidade). Em função de tal alteração, geramos interfaces para cada terreno e implementamos nas classes necessárias. As interfaces contêm os seus respectivos métodos de ativação que são implementados por cada personagem de formas diferentes.

Para cada batalha foi criada uma thread, o que permite que as lutas aconteçam simultaneamente. Randomicamente, ao início da batalha são decididos o terreno e as duplas para o combate. Então começa a batalha e os pares começam a infligir dano uns aos outros, além de desencadear uma soma de MP nos Wizards (adiciona 50 de MP a cada chamada de “run()” da thread). Ao chegar em 100, os magos causam 25 de dano em todos os personagens do time oposto. A batalha acaba quando todos os combates terminam, resultando num total de 4 sobreviventes, de qualquer um dos times, ao final do processo. O time vencedor é determinado pela soma de HPs dos personagens sobreviventes do mesmo.

Em virtude das mudanças citadas acima, houve algumas outras alterações ao longo de todo o código, como a automatização do processo de setters dos atributos dos personagens, isto é, caso o usuário tente fazer uma modificação de atributo que ultrapasse o somatório de 100 pontos, em vez de esta ser simplesmente ignorada, a modificação será feita no atributo escolhido e penalidades serão aplicadas nos demais atributos.