

Desafio Desenvolvedor

Objetivo

Projetar e implementar um serviço web REST e um cliente que possa consumir tal serviço e que apresente resultados de todas as interações. O enfoque da avaliação é para backend, os critérios de interface não serão avaliados, podendo o cliente ser implementado até mesmo em linha de comando.

Descrição

Criar uma rodada de uma batalha medieval para um jogo de rpg em turnos. O duelo é entre um herói e um monstro, inicialmente devem existir 3 diferentes monstros e heróis, cada um conforme as características abaixo:

Herói	Pontos de Vida (PdV)	Força	Defesa	Agilidade	Fator de Dano (FdD)
Guerreiro	12	4	3	3	2d4
Bárbaro	13	6	1	3	2d6
Paladino	15	2	5	1	2d4

Monstro	Pontos de Vida (PdV)	Força	Defesa	Agilidade	Fator de Dano (FdD)
Morto-Vivo	25	4	0	1	2d4
Orc	20	6	2	2	1d8
Kobold	20	4	2	4	3d2

Ao iniciar uma batalha, o usuário deverá informar um nickname e poderá escolher o herói, o oponente (monstro) será escolhido de forma aleatória pelo sistema. Cada usuário poderá iniciar uma batalha por vez, evitando assim que um usuário tenha várias batalhas não finalizadas.

Abaixo está a lógica que deve ser implementada para a batalha.

Primeiro passo, iniciativa:

Rola-se um dado de 10 faces (1d10) e soma-se ao fator de agilidade. O maior valor inicia o ataque. Caso dê empate, rola-se novamente até que um dos lados ganhe.

Segundo passo, ataque:

Para cada personagem (heróis e monstros), rola-se o ataque.

Rola-se um dado de 10 faces (1d10) + agilidade + força, o resultado deverá ser comparado com o cálculo da defesa que é obtido rolando um dado de 10 faces (1d10) + fator de agilidade + fator de defesa.

Se o valor do atacante for maior que do defensor, o atacante é o vencedor e o dano é calculado, caso o valor do defensor seja maior que o atacante, o defensor conseguiu defender-se do ataque e nenhum dano foi sofrido. Caso dê empate, o defensor ganha e nenhum dano é sofrido.

Cálculo do dano:

Rola-se o dado de dano de acordo com o fator de dano do personagem (2d4, 2d6 etc) + fator de força. O PdV do adversário deve ser subtraído desse valor. Caso o PdV do personagem chegue a zero ou menos, a luta é encerrada. Caso os dois personagens ainda possuam pontos de vida, a luta volta ao primeiro passo até que um dos personagens zere o PdV.

A regra XdY (ex. 3d10) indica X = número de dados e Y = número de faces. Ex: rola-se 3 dados de 10 faces (3d10) obtendo os resultados 2, 3, 8, o resultado final é 13.

Ranking:

Após finalizar uma batalha em que o herói foi vencedor, deverá ser calculado uma pontuação da vitória, sendo calculada como 100 - total de turnos. Esse valor deverá ser somado à pontuação do usuário e servirá para classificar o mesmo em um ranking.

(Extra) Armazenamento da Batalha:

Os detalhes das batalhas (iniciativas, ataques, danos, etc..) podem ser persistidos no banco de dados para consulta posterior como um histórico do jogador. Essa atividade é opcional.

Tecnologia

Deve ser utilizado o stack solicitado no email que acompanha esse desafio.

Observações

É preciso criar um repositório no github com seu projeto e enviar o link para nós. Não é preciso terminar o projeto inteiro para publicar suas modificações. É interessante apresentar a evolução do trabalho com commits contínuos.

Forneça também as instruções (README) para instalar e executar sua solução. Temos ambiente Linux com MongoDB 3.6, Redis 5, Java 8, PHP 5.6, Node 10 e Python 3.7.

Quando julgar finalizado, mesmo que não consiga concluir todas as etapas, basta enviar um email de confirmação.

Quanto mais informações tivermos sobre as decisões que tomou para o desenvolvimento, melhor.