

Alunos: Airton Osório Frota Araújo RA: 2310114

Ryan Gustavo Escossio Prado RA: 2410345

RPG De Turnos

O jogo funciona como um RPG de turno baseado em um RPG de mesa, onde a ordem de iniciativa é decidida por um meio de rolagem de dados, adicionando o bônus do personagem que pode ser escolhido antes de começar o combate começar.

Depois da iniciativa ser decidida, o combate começa, o personagem rola um dado de 20 lados e adiciona o mesmo bônus escolhido para ordem do combate em seu ataque, se o ataque passar de 11 somado o personagem acerta o inimigo, assim rolando um dado de 8 lados, o resultado que sair é descontado da vida do oponente, a vida é escolhida antes do combate começar.

A estrutura de dados utilizada foi Lista Encadeada, pois a iniciativa se baseia no primeiro que ataca, após seu turno, é a vez do oponente, por isso a classe Nó consegue impedir que um dos personagens ataque 2 vezes seguidas, assim mantendo a aleatoriedade do combate e a ordem respeitada.

No jogo é possível escolher a vida e o bônus do personagem e do inimigo, para uma batalha perfeitamente justa em que a sorte decidirá é ideal que os números sejam iguais, o combate termina quando a vida de algum dos personagens chega a 0 ou menos.

Existem muitas partes do código mais focados em layout, feitos para rodar em .jar igual nos foi instruído.

As classes utilizadas são: Personagem, a qual contém os métodos get e set dos atributos nome, vida e ataque.

Nó: Garante que a ordem de turnos não seja alterada.

Lista Turno: Utiliza o Nó para organizar a lista de ordem.

O código dos personagens contém todos os gets e sets necessários.

Este é o código da lista de turno:

```
1 public class ListaTurno {
2     No inicio;
3
4     public void adicionar(Personagem p) {
5         No novoNo = new No(p);
6         if (inicio == null) {
7             inicio = novoNo;
8         } else {
9             No atual = inicio;
10            while (atual.proximo != null) {
11                atual = atual.proximo;
12            }
13            atual.proximo = novoNo;
14        }
15    }
16    public Personagem proximo() {
17        if (inicio == null) return null;
18        Personagem personagem = inicio.personagem;
19        inicio = inicio.proximo;
20        return personagem;
21    }
22
23    public boolean temProximo() {
24        return inicio != null;
25    }
26 }
27
```

Uma parte do código da classe principal, a qual define a batalha:

```
private void executarTurno() {
    if (turnos == null) return;

    Personagem atacante = turnos.proximo();
    Personagem alvo = (atacante == heroi) ? monstro : heroi;

    int resultadoDado = rolarDado(lados:20);
    int totalAtaque = resultadoDado + atacante.getAtaque();
    int dano = 0;

    String mensagemDado = "Dado: " + resultadoDado + " + Bônus: " + atacante.getAtaque() + " = " + tot

    if (totalAtaque >= 11) {
        dano = rolarDado(lados:8);
        alvo.receberDano(dano);
        mensagemDado += " | Acerto! Dano: " + dano;
    } else {
        mensagemDado += " | Errou!";
    }



    dadolabel.setText(mensagemDado);
    statusLabel.setText(atacante.getNome() + " atacou: " + alvo.getStatus());

    O necessário para iniciar o jogo é apenas abrir o .jar, denso suficiente para jogar.

    O painel de seleção {
        statusLabel.setText(alvo.getNome() + " foi derrotado!");
        atacarButton.setEnabled(b:false);
    }
}
```

RPG por Turnos - Estilo Mesa



Vida Herói:	0	Bônus Ataque Herói:	0	Vida Monstro:
0	Bônus Ataque Monstro:	0	<button>Iniciar Batalha</button>	



Configure os valores e clique em Iniciar

Como fica após o herói derrotar o monstro:

RPG por Turnos - Estilo Mesa



Monstro foi derrotado!

Monstro começa! (Iniciativa: 22 vs 14)
Dado: 16 + Bônus: 4 = 20 | Acerto! Dano: 3

Atacar

A parte de personagem foi feita pelo Ryan, já a lista e a classe nó foram feitas por Airton, já o método main foi feito pelos 2, com pesquisas para conseguir fazer o layout necessário.