Centro Universitário do Estado do Pará
Curso de Engenharia de Computação
Disciplina: Programação 2
Professor: Pedro Girotto

Projeto de Programação: Sistema de Combate JRPG

1. Introdução

Role-Playing Game, também conhecido como RPG, é um tipo de jogo em que os jogadores assumem papéis de personagens e criam narrativas colaborativamente. O progresso de um jogo se dá de acordo com um sistema de regras predeterminado, dentro das quais os jogadores podem improvisar livremente.

Jogos de RPG japoneses (comumente conhecidos como JRPG) são jogos que possuem características distintas dos RPGs ocidentais.

O desenvolvimento de jogos do gênero RPG é fortemente influenciado por RPGs de mesa, como Dungeons and Dragons. Antes da disseminação do Famicom ou de outros consoles de jogos, os jogos de computador eram bastante populares no Japão. Durante a década de 1980, os RPGs japoneses começaram a aparecer no mercado.

Em 1985, Yuji Hori, um designer de jogos japonês, criou um jogo chamado Dragon Quest que pegou pedaços de vários RPGs como *Wizardry*, *Ultima* e *Black Onyx*. A coisa revolucionária que Dragon Quest fez foi tornar o gênero RPG mais acessível aos jogadores. Dragon Quest é conhecido como um RPG leve porque não exige que o jogador desenhe seus próprios mapas, memorize nomes específicos de feitiços ou comandos. Dragon Quest também simplificou as estatísticas implementando HP, MP, EXP e níveis em vez de sistemas complexos vistos em RPGs anteriores. Dragon Quest não exigia que o jogador aprendesse muitas regras complicadas, mas oferecia uma história intrigante com jogabilidade intuitiva.

2. Projeto

O objetivo do projeto é desenvolver um sistema de combate em turno que será aplicado em um jogo de JRPG.

Ao iniciar o aplicativo, o jogador se depara com um menu inicial. Contendo a opção de *começar o jogo*, *contar a história* que envolve o jogo e *fechar jogo*.

Ao escolher a opção *história*, deve ser mostrado um texto contando uma breve história do seu mundo de fantasia e o principal motivo do personagem querer enfrentar o desafio.

A opção *começar jogo* inicializa o jogo, que segue o fluxograma da Figura 1. O jogador criará seu personagem que tentará vencer três combates, onde a cada vitória de um combate dará uma recompensa para o jogador. O jogo vai se tornando mais difícil à medida que vencemos os combates.

Se as três lutas forem vencidas, chegaremos em uma tela de vitória, e se perdemos um combate, uma tela de derrota é mostrada e voltaremos para o início do jogo.

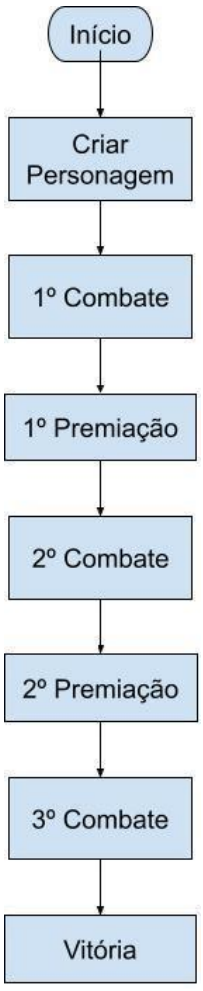

Figura 1 - Fluxograma do jogo.

2.1. Criação de Personagem

O personagem do jogo contém os seguintes atributos:

- Status Básicos:
  - Nome;
  - P.V. (Ponto de Vida);
- Atributos:
  - Força;
  - Constituição;
  - Agilidade;
  - Destreza;
- Equipamentos:
  - Arma;
  - Armadura.

A força influencia no dano da arma pesada. Constituição influência na quantidade de P.V. e na defesa. Agilidade influência na ordem dos turnos durante o combate. Destreza influencia no dano da arma leve.

As armas possuem os seguintes atributos:

- Arma:
  - Categoria;
  - Constante dano.

A categoria de uma arma pode ser dois: pesada e leve, na qual influencia no valor do dano.

O dano de uma arma pesada é calculado pela soma da rolagem de um dado de 12 lados (d12) mais 1,5 do valor da força e mais um valor constante (constante dano).

O dano de uma arma leve é calculada pela soma da rolagem de dois dados de 6 lados (d6) mais a rolagem de um dado de 4 lados (d4) mais o valor da destreza mais o valor de uma constante (constante dano).

As armaduras possuem os seguintes atributos:

- Armadura:
  - Defesa.

A defesa da armadura é calculada a partir de um número constante mais 1,5 do valor da constituição.

O ponto de vida é calculado pela soma da rolagem de três dados de 6 lados (d6) mais o valor da constituição.

Resumindo as características temos:

- Status Básicos:
    - P.V. = d6 + d6 + d6 + Cons.;
- Equipamentos:
    - Arma Pesada = K + d12 + 1,5*For.;
    - Arma Leve = K + d6 + d6 + d4 + Des.;
    - Armadura = K + 1,5*Cons.;

Ao começar a criação do personagem (Figura 2), o jogador deve dar um nome ao personagem, distribuir 15 pontos nos atributos e escolher uma armar e armadura. Onde o usuário poderá escolher uma das 3 armas iniciais e uma das 3 armaduras iniciais.

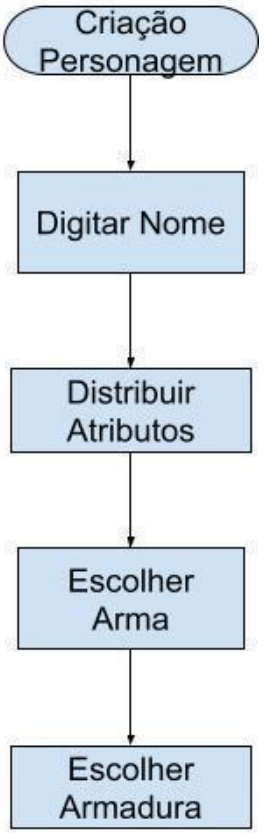

Figura 2 - Fluxograma da Criação do Personagem

2.2. Combate

No combate, o personagem lutará contra um adversário, que contém os seguintes atributos:

- Adversário:
    - Nome;
    - P.V. (Ponto de Vida);
    - Dano;

- Defesa;
- Agilidade.

O P.V. do adversário, defesa e agilidade são valores constantes decididos pelo programador. O dano do adversário pode ser um valor constante ou um valor aleatório, essa decisão e esses valores ficam a critério do desenvolvedor.

O combate envolve a sequência apresentada na Figura 3. Ao iniciar o combate, deve ser verificado quem é mais rápido (maior valor na agilidade), caso seja o jogador, ele terá o primeiro *round*, se não o primeiro *round* vai para o adversário. Após ambos fazerem suas jogadas, verificamos se alguém morreu (P.V. igual ou menor que zero), se o jogador morreu, o jogo termina e voltamos para o menu principal; se o adversário morreu, o jogador ganha sua recompensa e passamos para a próxima luta; se ninguém morreu, o jogador e o adversário jogam mais um *round*, e isso se repete até alguém morrer.

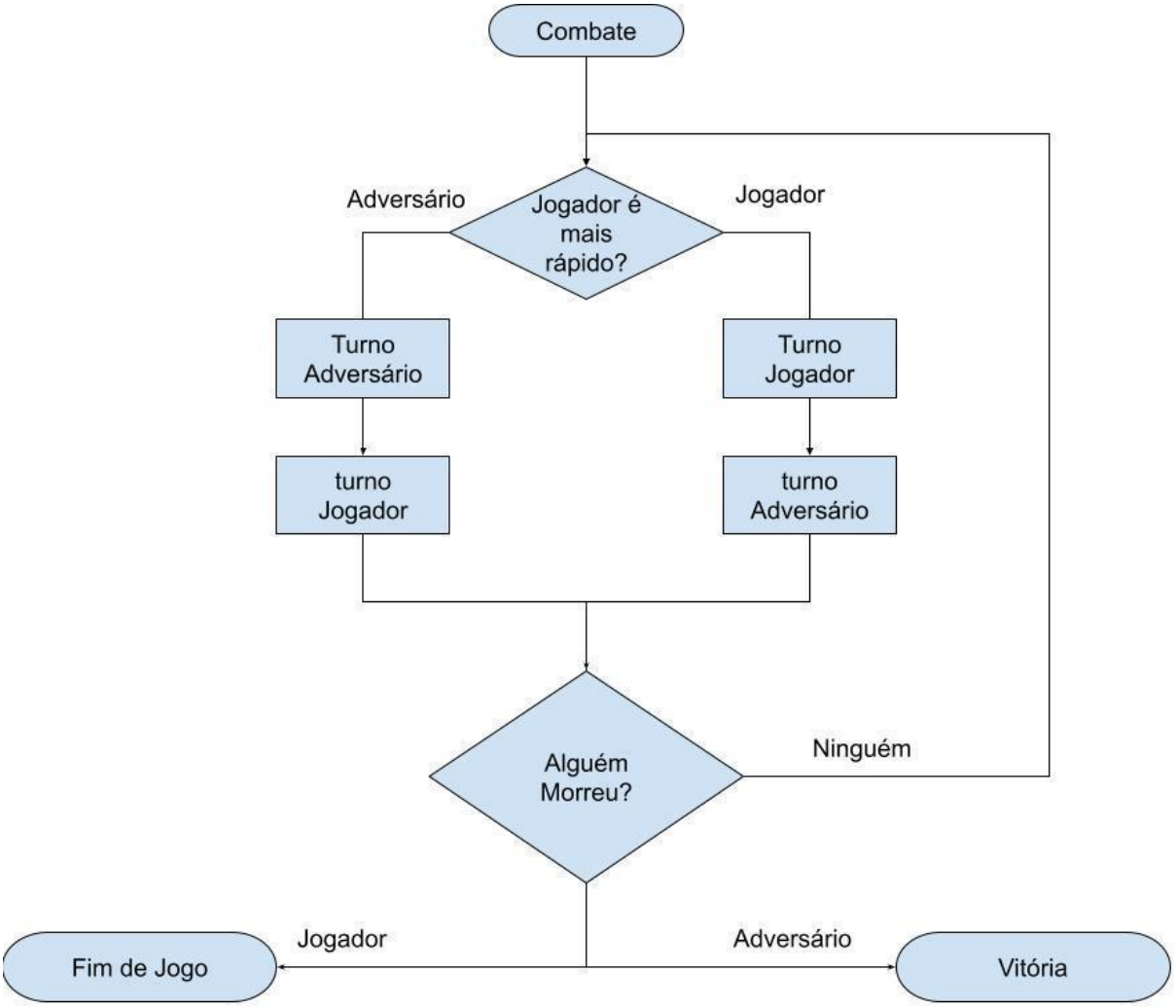

Figura 3 - Fluxograma do Combate.

O adversário deverá ser escolhido de forma aleatória para cada combate, fazendo o jogo ser aproveitado mais de uma vez.

2.3. Turno Jogador

No turno do jogador terão três ações disponíveis:

- Ações:
    - Atacar;
    - Defender;
    - Usar Poção.

Atacar faz o personagem causar dano ao adversário. O dano causado no adversário é calculado pegando o valor do dano da armada do jogador menos a armadura do adversário. Com o valor do dano calculado, subtrai-se do P.V. do adversário.

Defender dobra o valor da defesa do personagem por 1 *round*.

Usar Poção recupera os P.V. do jogador. Esse valor é calculado fazendo a soma da rolagem de três dados de 6 lados. O jogador só tem acesso a três poções por combate.

2.4. Turno Adversário

Mesma ideia do jogador, porém quem escolhe a ação é o computador. Um número aleatório entre 0 e 2 é gerado e a partir desse valor uma ação é selecionada.

- Ações:
    - 0 - Atacar;
    - 1 - Defender;
    - 2 - Usar Poção.

2.5. Premiação

Ao vencer cada combate, o jogador receberá uma premiação.

- Primeira Premiação:
    - Subir um level;
    - Mais P.V.;
    - Escolher uma nova arma.

Ao subir um level, o jogador ganha mais 5 pontos de atributos para serem distribuídos.

O novo valor do P.V. é feito pegando o antigo valor máximo de P.V. e acrescenta o valor da constituição.

O usuário poderá escolher uma das 3 armas apresentadas, essas novas armas devem ser mais fortes do que as armas iniciais.

- Segunda Premiação:
    - Subir dois leveis;
    - Mais P.V.
    - Escolher uma nova armadura.

Ao subir dois níveis, o jogador ganha mais 10 pontos de atributos para serem distribuídos.

O novo valor do P.V. é feito pegando o antigo valor máximo de P.V. e acrescenta o valor da constituição.

O usuário poderá escolher uma das 3 armaduras apresentadas, essas novas armaduras devem ser mais fortes do que as armaduras iniciais.

- Terceira Premiação:
    - Fim do jogo.

Na terceira premiação o jogador venceu o jogo, então uma tela de vitória é mostrada e a conclusão da história.

3. Requisitos do Projeto

Alguns requisitos serão obrigatórios na implementação do projeto.

- O projeto deve ser desenvolvido na linguagem de programação JAVA;
- Devem ser utilizadas *classes* para o jogador, arma, armadura, poção e adversário;
- Todas as classes devem ter construtores e encapsulamento.
- O jogo deve conter no mínimo seis adversários:
    - Três para o primeiro combate, onde um será escolhido aleatoriamente para participar da luta;
    - Dois para o segundo combate, onde um será escolhido aleatoriamente para participar da luta;
    - Um para o último combate.

Para pontuação extra, a equipe pode fazer as seguintes implementações:

- Text Adventure: Uma descrição do cenário é apresentada para o jogador onde ele pode fazer interações pré-definidas, cada interação gera resultados diferentes. Essas interações podem levar a novas salas, ao encontro com um adversário, tesouros, armadilhas e etc.
- Uma classe *Magia*, o seu funcionamento fica a critério da equipe mas a classe *jogador* deve ter um vetor de *Magia* que podem ser escolhidas durante o combate.