Relatório - Trabalho Final de POO

Gabriel Moura, Luis Rasch, Renan Pinho

14 de março de 2024

Universidade Federal de Pelotas
Colegiado dos Cursos de Computação
Centro de Desenvolvimento Tecnológico – CDTec

Professor: Felipe de Souza Marques
Disciplina: Programação Orientada a Objetos

1 Introdução

Este relatório apresenta o desenvolvimento de um jogo como parte do trabalho prático para a disciplina de Programação Orientada a Objetos. O objetivo deste projeto foi aplicar os conceitos aprendidos durante o curso para criar um jogo interativo e funcional. O jogo desenvolvido é uma adaptação dos jogos RPG de mesa, entitulado "Dungeon Clash", consistindo em uma aventura linear com enredo e sistemas de batalha e progressão de personagens. O jogo foi implementado na linguagem de programação Java. Este relatório irá abordar cada fase do desenvolvimento do jogo, desde a concepção inicial da ideia até a sua implementação em código, as estruturas de classes e objetos utilizadas, as funcionalidades implementadas e os desafios enfrentados ao longo do processo.

2 Funcionalidades Implementadas

Criação de times: Sistema que permite criar um time principal de até três personagens com três classes possíveis: Mago, Guerreiro e Arqueiro.

Sistema de Batalha: Implementação de um sistema de combate baseado em turnos, onde tanto jogadores quanto inimigos utilizam habilidades específicas de suas classes.

Progressão da história: O jogador passa por uma campanha linear guiada por um arquivo de texto, que determina partes da história e inimigos das batalhas.

Progressão de personagens: Conforme o jogo avança e os inimigos são derrotados, os personagens ganham pontos de experiência que, em uma quantidade determinada, elevam o personagem para um próximo nível, aumentando sua força, agilidade ou inteligência.

Interface baseada em texto: Toda a interação do jogo é feita por uma interface baseada em texto, onde o usuário digita os comandos que deseja realizar.

2.1 Dificuldades Encontradas

Tivemos uma dificuldade considerável durante o desenvolvimento desse trabalho. Primeiramente, tivemos problemas no entendimento das especificidades do trabalho. Uma das principais dificuldades encontradas foi a complexidade na implementação do sistema de combate. Por fim, houveram vários outros desafios técnicos menores e problemas com o gerenciamento de tempo.

3 Passos para compilar e rodar o programa

A compilação pode ser feita através de uma IDE ou com um compilador Java via terminal. Para rodar o programa, basta abrir o arquivo compilado.

4 Exemplos de Utilização

Figura 1:

```
 --- Informações das Equipes ---

Heróis:
ID: 1
Nome: Luis Rasch
Classe: Arqueiro
PV: 2.5
PM: 3.0
Nível: 1
Tempo de Espera: 0
--------------------------------------
ID: 2
Nome: Gabriel Moura
Classe: Guerreiro
PV: 4.5
PM: 1.3333334
Nível: 1
Tempo de Espera: 0
--------------------------------------
ID: 3
Nome: Renan Pinho
Classe: Mago
PV: 2.0
PM: 3.6666667
Nível: 1
Tempo de Espera: 0
--------------------------------------
```

Figura 2:

```
--------------------------------------

Inimigos:
ID: 5
Nome: Morcego2
Classe: Monstro
PV: 4.5
PM: 0.33333334
Nível: 1
Tempo de Espera: 4
--------------------------------------

 --- Turno 3 ---
É a vez de Gabriel Moura atacar!
Escolha uma habilidade para Gabriel Moura:
Habilidades disponíveis para Gabriel Moura:
Socar (PM Necessário: 0.0, Dano: 3.0)
Golpe de Espada (PM Necessário: 0.0, Dano: 5.0)
Espada Flamejante (PM Necessário: 3.0, Dano: 3.0)
Digite o nome da habilidade: Golpe de espada
Escolha um alvo:
1. Morcego2
Digite o número correspondente ao alvo: 1

Morcego2 morreu.
```

Figura 3:

5 Diagrama de Classes

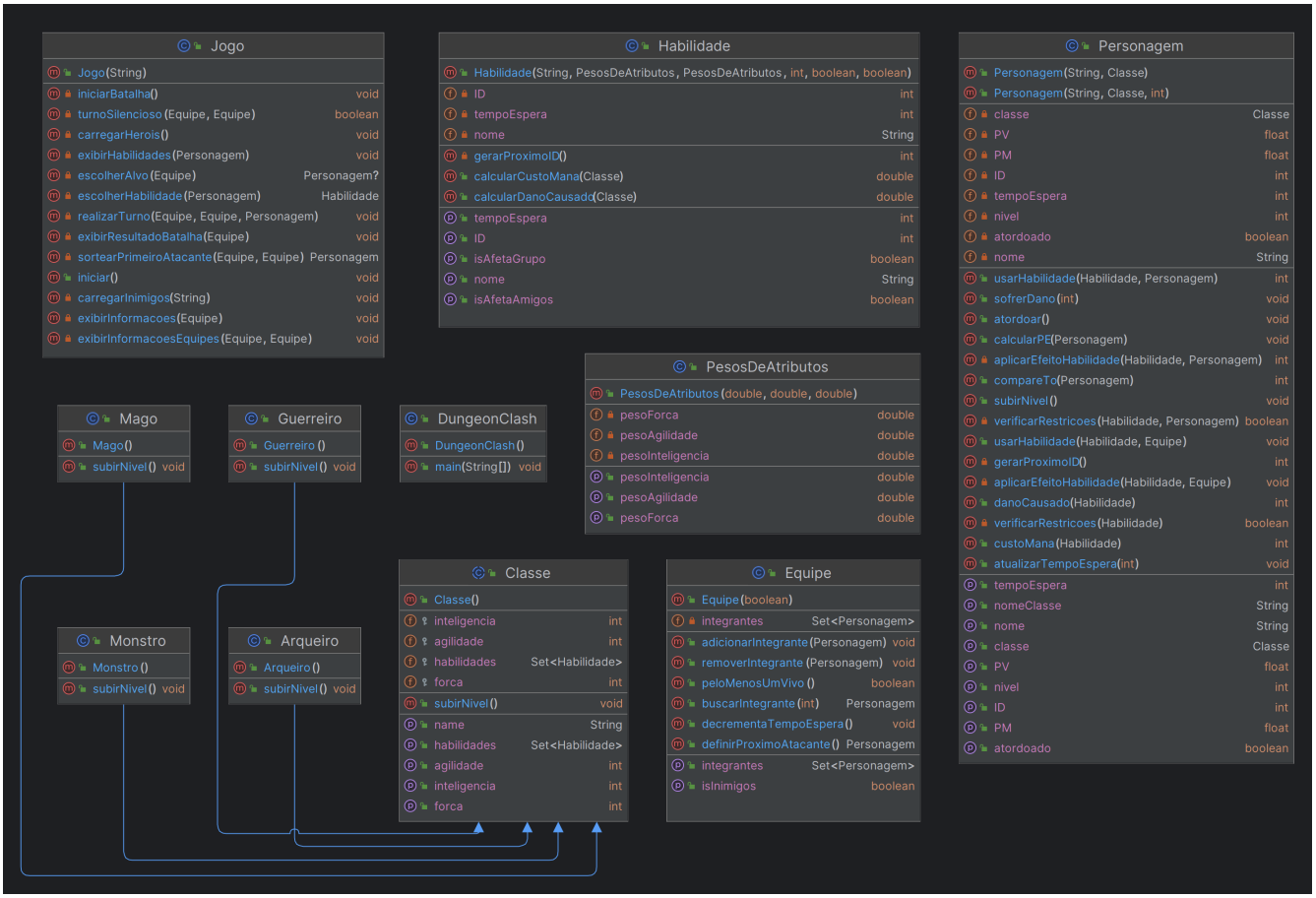

Figura 4: Diagrama arquitetural feito para esboçar o projeto.

6 Aplicações dos Conceitos de Orientação a Objetos

Construtores: Todas as classes, com exceção da classe principal "DungeonClash.java", possuem métodos construtores para inicializar um objeto de cada classe.

Herança: As classes Guerreiro, Arqueiro, Mago e Monstro são herdeiras da classe base "Classe", possuindo um relacionamento de especialização.

Encapsulamento: O princípio de encapsulamento pode ser observado nos atributos das classes que estendem "Classe". Características como força, habilidade e inteligência estão encapsuladas, ou seja, foram declaradas como "private", o que significa que só podem ser acessadas diretamente dentro da própria classe.

Polimorfismo: Diferentes classes possuem métodos com a mesma assinatura, mas que realizam coisas diferentes. Por exmeplo, o método "subirNivel()"possui a mesma assinatura nas classes "Guerreiro", "Mago"e "Arqueiro", mas suas implementações são diferentes.

Métodos: O jogador passa por uma campanha linear guiada por um arquivo de texto, que determina partes da história e inimigos das batalhas.

Exceções: O projeto faz uso de exceções para lidar com erros durante a execução do programa. Por exemplo, as exceções IOException são tratadas ao ler o arquivo e as exceções NumberFormatException são tratadas ao converter strings em números.

Coleções e Iteração: Há o uso de coleções "List"e "Set". Algumas classes utilizam coleções "HashSet"que permitem que múltiplos objetos possam ser gerenciados e iterados.

7 Conclusão

Com esse trabalho, exploramos e aplicamos todos os conceitos de orientação a objetos trabalhados ao longo do semestre, em um contexto prático e criativo. Apesar dos obstáculos encontrados ao longo do processo, obtivemos vários êxitos no desenvolvimento deste trabalho. Reconhecemos, no entanto, que sempre há espaço para melhorias e refinamentos, e esperamos continuar explorando e aprimorando nosso projeto.

8 Sugestões de Trabalhos Futuros

Para futuros projetos que visem aplicar os conceitos de Programação Orientada a Objetos, sugerimos a criação de um sistema de criação de jogos de tabuleiro digitais. Este sistema poderia oferecer uma interface intuitiva para os usuários projetarem e implementarem seus próprios jogos de tabuleiro personalizados, com suporte a uma variedade de mecânicas de jogo, componentes e regras. Isso envolveria a concepção e implementação de classes e objetos que representam elementos comuns de jogos de tabuleiro, como peças, tabuleiros, cartas e dados.