

TRABALHO FINAL - PROGRAMAÇÃO ORIENTADA POR OBJETOS
JOGO DE RPG - TOGA DOS GOBLINS

Caio Moreira Marques

Luan Francisco dos Santos

Pedro César da Matta Silveira

Yasmin Vitoria de Oliveira Munhoz

DESCRIÇÃO DO PROJETO

.Este projeto é um RPG de turnos simples, desenvolvido em Unity com foco nos princípios da Programação Orientada a Objetos (POO). O jogador pode criar e evoluir seu personagem através de um sistema de progressão básico.

O jogo possui um sistema de combate básico, inspirado no balanceamento de Tormenta 20. A coleta de itens é feita ao eliminar inimigos ou completar missões. Um sistema de missões simples guia o jogador pela pequena narrativa. A aplicação da POO visa um código modular, reutilizável e de fácil manutenção.

Todos os dados do jogo, como informações do personagem, progresso das missões e inventário, são armazenados em arquivos JSON e carregados ao iniciar uma nova sessão, permitindo que o jogador retome a aventura de onde parou.

ARQUITETURA DA SOLUÇÃO

A solução é estruturada em camadas e módulos, promovendo separação de responsabilidades e fácil manutenção. Os principais componentes são:

1. Entidades (Entities)

Contém as classes que representam os elementos centrais do jogo, como Personagem, Inimigo, Classe, Raca, Item, Habilidade. Cada entidade encapsula dados e comportamentos específicos, facilitando a expansão e reutilização.

2. Gerenciadores (Managers)

Os gerenciadores, como GameManager e CombatManager, centralizam o controle do fluxo do jogo, estados globais, transições de cena, gerenciamento de combate e persistência de dados. O GameManager é responsável pelo ciclo de vida do jogo, carregamento de dados e controle de áreas, enquanto o CombatManager gerencia o combate em turnos e suas interações.

3. Sistema de Combate

O módulo SistemaCombate implementa a lógica de combate em turnos, incluindo ataques, uso de habilidades, poções e fuga. Utiliza eventos para atualizar a interface e o estado do jogo conforme as ações do jogador e do inimigo.

4. Fábricas e Helpers

A classe CharacterFactory é responsável por instanciar personagens, inimigos e NPCs no mundo ou em combate, garantindo que os objetos estejam configurados corretamente para cada contexto.

5. Persistência e Carregamento de Dados

O módulo DataLoader gerencia a serialização e desserialização de dados do jogo, como personagens, itens, áreas e missões, utilizando arquivos JSON. Isso permite fácil expansão e customização do conteúdo do jogo sem necessidade de recompilar o código.

6. Interface do Usuário (UI)

Os painéis de UI, como InventarioPanel, CombatUI e outros, são responsáveis pela interação do jogador com o inventário, combate, estatísticas e diálogos. A UI é atualizada dinamicamente conforme o estado do jogo.

7. World

O mundo do jogo é composto por áreas, cada uma contendo NPCs e inimigos. O gerenciamento de transição entre áreas e posicionamento do jogador é feito pelo GameManager e WorldExplorationManager.

DIAGRAMA DE CLASSES

