

DIOGO PEREIRA RIBEIRO  
EDUARDO TIMM BUSS  
ERICK RADMANN

Recriação do jogo Battle City utilizando Programação Orientada a Objetos

Orientador: Prof. Rafael Burlamaqui

Pelotas, dezembro de 2025.

## **1 DADOS DE IDENTIFICAÇÃO**

### **1.1 Nome do Projeto**

Recriação do jogo Battle City utilizando Programação Orientada a Objetos

### **1.2 Local de Realização**

CDTec UFPel

### **1.3 Responsável pelo Projeto**

Diogo Pereira Ribeiro, Eduardo Timm Buss e Erick Radmann

### **1.4 Professor Orientador**

Prof. Rafael Burlamaqui

## 2 SUMÁRIO EXECUTIVO

Este trabalho consiste no desenvolvimento de um jogo digital inspirado no clássico **Battle City**, originalmente lançado para o console NES. O projeto foi desenvolvido utilizando a linguagem **Java**, aplicando os conceitos de **Programação Orientada a Objetos (POO)** vistos em aula.

O principal objetivo do projeto é consolidar, na prática, os conceitos fundamentais de POO, tais como **encapsulamento, herança, polimorfismo e abstração**, por meio da implementação de um jogo completo e funcional. O jogo desenvolvido possui movimentação de tanques, disparo de projéteis, colisões, inimigos controlados pela lógica do sistema e um loop principal de execução.

Como contribuição, o projeto demonstra como a POO pode ser utilizada para organizar e estruturar sistemas mais complexos, facilitando a manutenção, expansão e compreensão do código-fonte.

### 3 HISTÓRICO E JUSTIFICATIVA

Jogos digitais têm sido amplamente utilizados como ferramenta de aprendizado na área de computação, especialmente no ensino de programação. Clássicos como Battle City tornaram-se referências por sua mecânica simples, porém eficaz, permitindo a implementação de conceitos fundamentais de desenvolvimento de jogos.

A Programação Orientada a Objetos surgiu como uma forma de organizar sistemas complexos a partir de entidades que representam elementos do mundo real. Em jogos, essa abordagem se mostra especialmente adequada, pois personagens, cenários, projéteis e regras podem ser naturalmente modelados como objetos.

A escolha do jogo Battle City como base para este projeto justifica-se por sua estrutura clara e por permitir a aplicação direta dos conceitos vistos em aula. Além disso, a recriação de um jogo clássico contribui para o aprendizado prático, estimulando a lógica, o raciocínio computacional e a organização do código.

## 4 OBJETIVOS E METAS

### Objetivo Geral

Desenvolver um jogo inspirado no Battle City utilizando a linguagem Java, aplicando corretamente os conceitos de Programação Orientada a Objetos.

### Objetivos Específicos

- Modelar as entidades do jogo utilizando classes e objetos;
- Aplicar herança para reaproveitamento de código entre entidades semelhantes;
- Utilizar encapsulamento para proteger os dados internos das classes;
- Implementar polimorfismo em comportamentos comuns entre objetos do jogo;
- Desenvolver um loop principal responsável pela execução do jogo;
- Permitir interação do usuário por meio de teclado.

## 5 METODOLOGIA

Inicialmente, foi realizada uma análise do jogo original Battle City, identificando seus principais elementos e mecânicas. Em seguida, foi feito o planejamento das classes necessárias, como tanque, inimigos, projéteis e mapa.

O desenvolvimento foi realizado de forma incremental, iniciando pela estrutura básica do projeto, seguida pela implementação do loop principal do jogo. Posteriormente, foram adicionadas funcionalidades como movimentação, colisões e disparos.

Durante o desenvolvimento, buscou-se aplicar boas práticas de POO, mantendo o código organizado, modular e de fácil entendimento.

## **6 CONCEITOS DE POO UTILIZADOS**

Nesta seção são descritos os conceitos de Programação Orientada a Objetos solicitados no trabalho e como eles foram aplicados no desenvolvimento do jogo Battle City.

### **6.1 Encapsulamento**

O encapsulamento foi aplicado por meio da definição de atributos privados nas classes do jogo, permitindo que o acesso e a modificação desses dados ocorra apenas através de métodos públicos. Isso garante maior segurança e controle sobre o estado interno dos objetos, evitando acessos indevidos.

### **6.2 Herança**

O conceito de herança foi utilizado para modelar entidades semelhantes do jogo. Classes mais específicas herdam atributos e comportamentos de classes mais genéricas, promovendo reutilização de código e facilitando a manutenção do sistema. Por exemplo, entidades como tanques do jogador e inimigos compartilham características comuns herdadas de uma classe base.

### **6.3 Polimorfismo**

O polimorfismo foi empregado permitindo que objetos de diferentes classes sejam tratados de forma uniforme através de referências do mesmo tipo base. Métodos com o mesmo nome apresentam comportamentos distintos conforme a classe concreta do objeto, especialmente em ações como movimentação, atualização e interação com o ambiente do jogo.

### **6.4 Composição**

A composição foi utilizada na construção de objetos mais complexos a partir de objetos mais simples. Um exemplo é a classe principal do jogo, que possui instâncias de outras classes como mapa, entidades e sistema de renderização, caracterizando uma relação do tipo "tem um".

## **6.5 Interfaces**

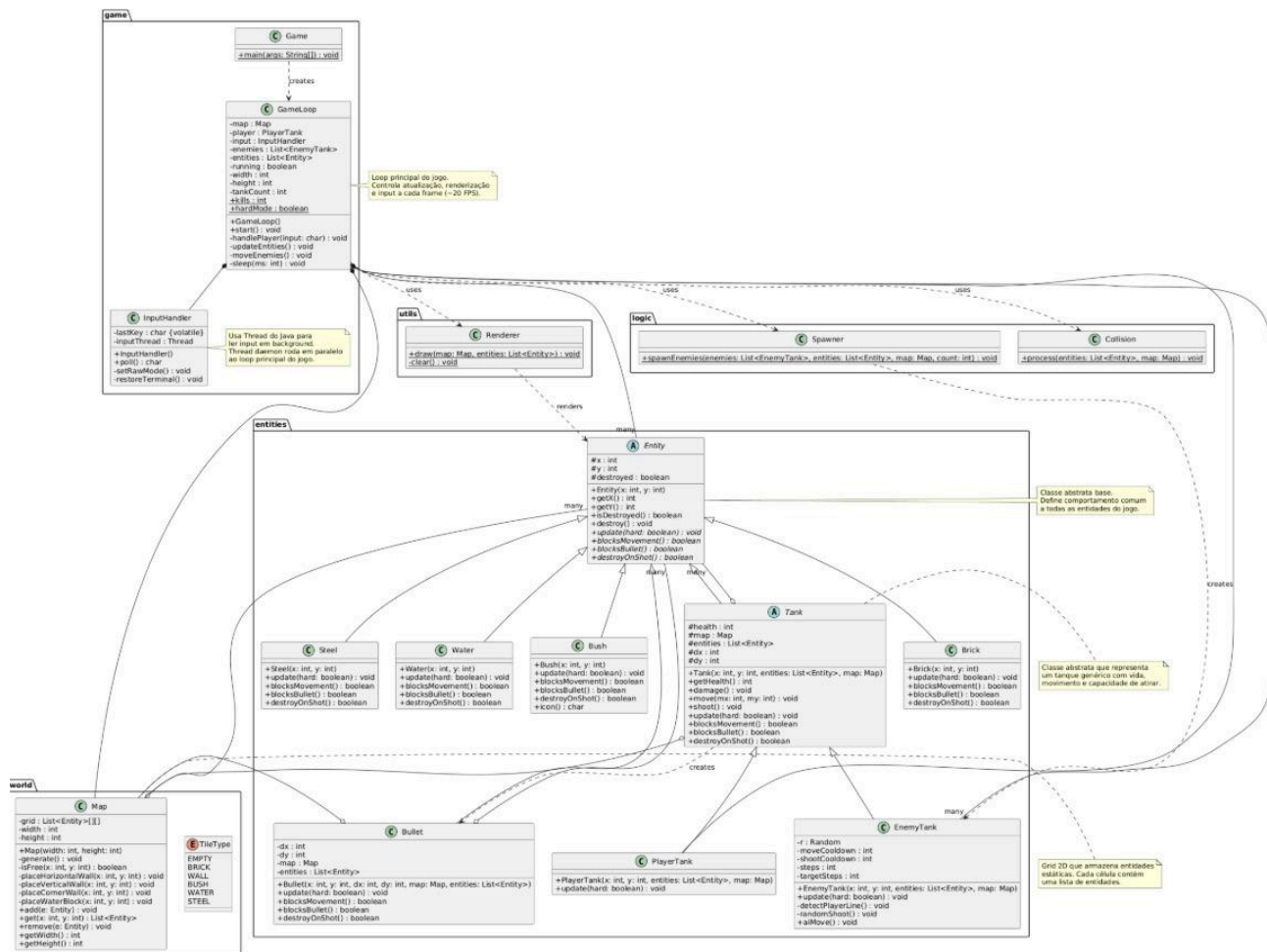
Foram utilizadas interfaces para definir contratos de comportamento que devem ser implementados por diferentes classes do jogo. Isso permite maior flexibilidade e desacoplamento, possibilitando que diferentes entidades implementem comportamentos comuns, como atualização de estado ou renderização.

## **6.6 Threads**

O uso de threads ocorre no controle do loop principal do jogo. A execução contínua do jogo é realizada em uma thread separada, permitindo a atualização da lógica do jogo e a renderização de forma constante, sem bloquear a aplicação.



O diagrama de classes do projeto representa as principais entidades do jogo, como Game, Tank, Enemy, Bullet e Map, bem como suas relações de herança e associação.



## 8 PASSOS PARA COMPILAR E RODAR O PROGRAMA

Entrar na pasta: `cd src/`

Compilar todos os arquivos: `javac */*.java`

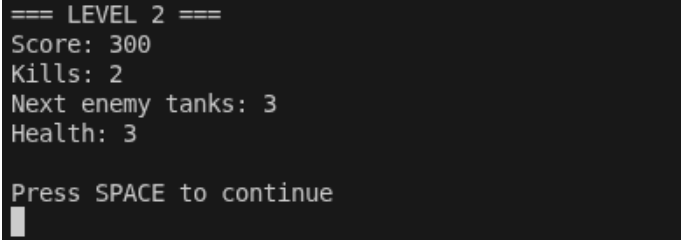
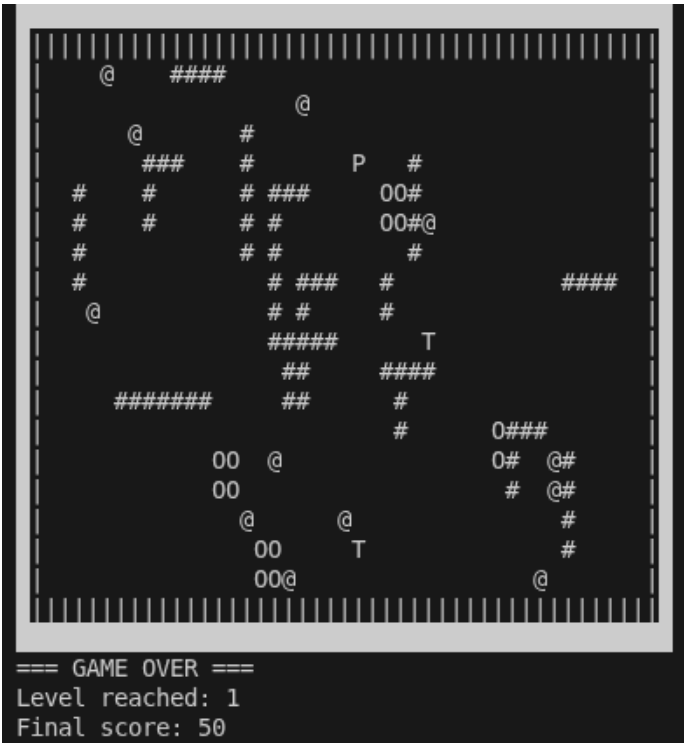
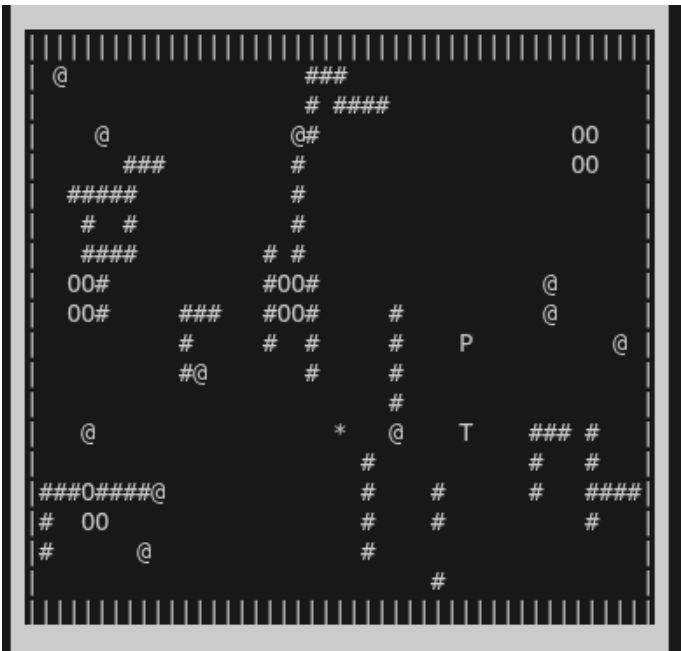
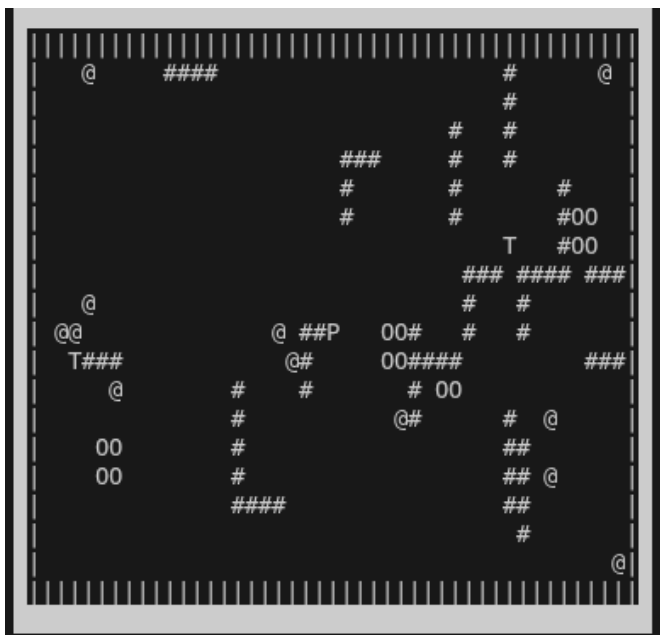
Executar a main: `java game/Game`

IDE: VS Code

Sistema Operacional: Linux

9 EXEMPLOS DE UTILIZAÇÃO

Nesta seção são apresentadas algumas telas do jogo em execução, demonstrando a movimentação do tanque, disparos e interação com inimigos.



=== GAME OVER ===  
Level reached: 1  
Final score: 50

## **10 DIFICULDADES ENCONTRADAS**

Durante o desenvolvimento do projeto, as principais dificuldades encontradas foram a implementação correta das colisões, o controle do loop principal do jogo e a organização das classes de forma coerente utilizando POO.

Essas dificuldades foram superadas por meio de testes, revisões de código e estudo adicional sobre desenvolvimento de jogos em Java.

## **11 CONCLUSÃO**

O desenvolvimento deste projeto permitiu consolidar, na prática, os conceitos de Programação Orientada a Objetos estudados em aula. A recriação do jogo Battle City mostrou-se uma abordagem eficiente para aplicar teoria e prática de forma integrada.

O projeto atingiu seus objetivos propostos, resultando em um jogo funcional e bem estruturado, além de proporcionar um aprendizado significativo sobre organização de código, reutilização e modelagem de sistemas.