

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO
PAULO**
Análise e Desenvolvimento de Sistemas

MATEUS CAMPOS SOUZA DE PAULA

SPACE SHOOTER
JOGO DESENVOLVIDO EM C++

CAMPOS DO JORDÃO
2025

Sumário

Sumário

1. Introdução	5
2. Metodologia.....	6
2.1. C++	7
2.2. Raylib	7
2.3. Descrição do Projeto.....	8
3. Resultados Obtidos	10
4. Conclusão.....	14
5. Referências Bibliográficas	15

Lista de Figuras

Figura 1 Diagrama de Classe.....	9
Figura 2 Seletor de Skin	11
Figura 3 Menu Principal.....	12
Figura 4 Jogo Rodando	13
Figura 5 Jogo Pausado	13
Figura 6 Jogo Perdido	14

Resumo

O presente trabalho descreve o desenvolvimento de um jogo digital original elaborado na linguagem C++ com uso da biblioteca Raylib. O objetivo principal do projeto consiste em demonstrar, na prática, a aplicação dos conceitos de Programação Orientada a Objetos por meio da construção de um software interativo funcional. O jogo desenvolvido pertence ao gênero *shoot 'em up* (jogo de ação no qual o jogador controla um personagem ou veículo que dispara continuamente contra inimigos), apresentando elementos característicos de *bullet hell* (subgênero marcado por grande quantidade de projéteis na tela) e incorporando um sistema de *power-ups* (itens que fornecem melhorias temporárias ao jogador). A estrutura modular implementada, associada a mecânicas de movimentação, disparo, geração de inimigos e progressão por ondas, permitiu consolidar aspectos fundamentais de encapsulamento, modularização e interação entre objetos. Assim, o projeto cumpre seu propósito pedagógico ao evidenciar como a orientação a objetos pode ser aplicada de maneira integrada ao desenvolvimento de jogos digitais.

Palavras-chave: Programação Orientada a Objetos; C++; Raylib; *shoot 'em up*; *bullet hell*; *power-ups*.

1. Introdução

O desenvolvimento deste projeto tem como finalidade aplicar, de maneira prática e contextualizada, os principais conceitos estudados na disciplina de Programação Orientada a Objetos. Essa abordagem permite compreender os pilares fundamentais do paradigma orientado a objetos por meio da construção de um sistema interativo real, no qual classes, objetos, atributos e métodos são utilizados de forma integrada. A implementação do jogo foi realizada em linguagem C++, que constitui um recurso adequado para esse tipo de aplicação por oferecer suporte simultâneo a mecanismos de abstração e a operações de

baixo nível. Conforme destaca Stroustrup (2013), o C++ foi desenvolvido com o propósito de combinar eficiência e abstração, possibilitando que sistemas complexos sejam construídos de maneira flexível e com alto desempenho, o que reforça sua pertinência para projetos que envolvem múltiplas entidades e interações dinâmicas.

Usando essa base como ponto de partida, este trabalho descreve o desenvolvimento de um jogo original do gênero *shoot 'em up*, pensado para ajudar na prática dos princípios de orientação a objetos. O foco do projeto não está na criação de uma experiência de jogo sofisticada ou graficamente elaborada, mas na oportunidade de demonstrar, por meio de um software funcional, a importância da modularização, da separação de responsabilidades e da reutilização de código. O jogo permite ao jogador controlar uma nave espacial em um ambiente de combate contínuo, enfrentando inimigos com padrões variados, coletando *power ups* e lidando com ondas de dificuldade progressiva, elementos que favorecem a necessidade de uma arquitetura orientada a objetos bem estruturada.

A justificativa do projeto encontra fundamento na importância da experiência prática para consolidar conteúdos estudados na disciplina. A criação de um jogo, mesmo que de baixa complexidade, demanda o emprego de múltiplos conceitos fundamentais da área de desenvolvimento de software, como encapsulamento, hierarquia de classes, modularização e integração de componentes independentes. O projeto, portanto, representa uma oportunidade de aplicar de maneira concreta os princípios teóricos da orientação a objetos, ao mesmo tempo em que explora um domínio motivador e amplamente utilizado na indústria de desenvolvimento de jogos.

2. Metodologia

A metodologia adotada para o desenvolvimento deste projeto foi a metodologia incremental, que consiste na construção do software em pequenas etapas sucessivas, cada uma delas adicionando novas funcionalidades ao sistema. Essa abordagem possibilitou que o jogo fosse estruturado gradualmente, permitindo testes contínuos, revisão constante do código e ajustes conforme as necessidades identificadas ao longo do processo. Por meio

desse método, foi possível garantir que cada etapa do desenvolvimento estivesse funcional antes da implementação do incremento seguinte, reduzindo a probabilidade de falhas estruturais e facilitando a integração entre os diferentes componentes do jogo. As ferramentas utilizadas para o desenvolvimento foram a linguagem C++, a biblioteca Raylib para renderização e controle de entrada e um ambiente de desenvolvimento integrado compatível com o projeto. A seguir, são apresentadas descrições detalhadas sobre cada tecnologia empregada.

2.1. C++

A linguagem C++ foi escolhida como base para a implementação do projeto por oferecer recursos adequados ao paradigma de Programação Orientada a Objetos, além de permitir controle eficiente sobre desempenho e gerenciamento de memória. Conforme discutem Paul Deitel e Harvey Deitel em *C++ How to Program* (2017), a linguagem foi projetada para possibilitar a construção de sistemas robustos a partir de abstrações bem definidas, ao mesmo tempo em que fornece ao programador acesso direto a mecanismos de baixo nível quando necessário. Essa combinação torna o C++ particularmente apropriado para aplicações que exigem simultaneamente estruturação modular e eficiência operacional, como é o caso de jogos digitais. No contexto deste trabalho, o C++ permitiu a criação de classes independentes para entidades como jogador, inimigos, projéteis e *power ups*, organizando o projeto de maneira coerente com os princípios da orientação a objetos.

2.2. Raylib

A biblioteca Raylib foi utilizada como ferramenta principal para a renderização gráfica bidimensional, gerenciamento de texturas, detecção de entrada do usuário e controle do loop principal do jogo. De acordo com sua documentação oficial, a Raylib foi desenvolvida com o objetivo de oferecer uma interface simples e acessível para a criação de jogos e aplicações gráficas, permitindo que programadores concentrem seus esforços na lógica do projeto sem a necessidade de lidar com estruturas complexas de baixo nível. A biblioteca disponibiliza funções diretas e bem organizadas para desenho de sprites, tratamento de colisões básicas, reprodução de sons e manipulação de vetores bidimensionais, o que a torna especialmente adequada para projetos educacionais e protótipos de jogos. No contexto deste trabalho, a Raylib foi

fundamental para implementar a movimentação da nave, o disparo de projéteis, o controle dos inimigos e a exibição de todos os elementos gráficos na tela.

2.3. Descrição do Projeto

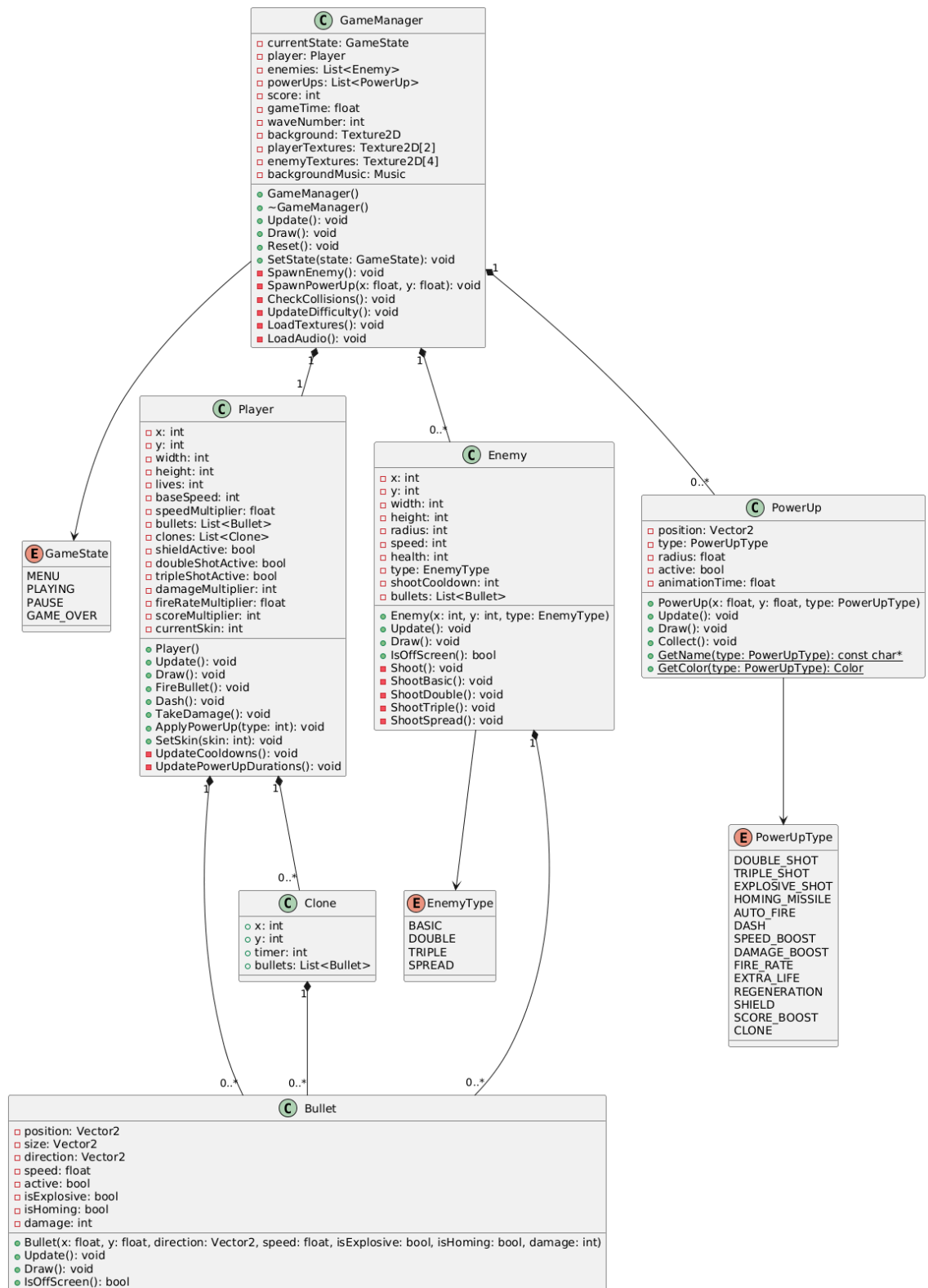
O projeto desenvolvido consiste em um jogo digital do gênero *shoot 'em up* vertical, no qual o jogador controla uma nave espacial em um ambiente bidimensional com perspectiva superior. O objetivo principal é sobreviver ao maior número possível de ondas de inimigos, que apresentam padrões distintos de movimentação e ataque. O jogo foi estruturado de forma modular, seguindo princípios de Programação Orientada a Objetos, permitindo que cada entidade fosse modelada como um componente independente e reutilizável.

A nave controlada pelo jogador possui capacidade de movimentação em oito direções e utiliza diferentes tipos de disparo, obtidos por meio de *power ups* distribuídos durante a partida. Esses itens também podem proporcionar melhorias temporárias de defesa, velocidade ou taxa de disparo, ampliando a variedade de estratégias disponíveis ao jogador. O sistema de inimigos inclui uma classe, configurada com padrões específicos de ataque, quantidade de projéteis emitidos e velocidade, permitindo criar graduações naturais de dificuldade.

O jogo implementa ainda um sistema de progressão baseado em ondas sucessivas, cujo nível de dificuldade aumenta de acordo com o número de inimigos e a complexidade de seus padrões. Cada elemento visual do jogo como nave, inimigos e projéteis é representado por sprites em estilo pixel art, renderizados utilizando os recursos gráficos da biblioteca Raylib. Além disso, o projeto incorpora um gerenciador central responsável por coordenar a lógica geral do jogo, incluindo atualização de estados, detecção de colisões, controle de pontuação e organização dos ciclos de execução.

A integração entre C++ e Raylib permitiu que o jogo fosse construído de forma clara e estruturada, evidenciando conceitos fundamentais de orientação a objetos. Dessa maneira, o projeto cumpre seu propósito educacional ao demonstrar, na prática, a aplicação de técnicas de programação orientada a objetos em um contexto funcional e interativo.

Figura 1 Diagrama de Classe



3. Resultados Obtidos

A etapa de desenvolvimento resultou em um jogo funcional, capaz de demonstrar de maneira prática a aplicação dos conceitos de Programação Orientada a Objetos integrados à biblioteca Raylib. O produto final apresenta todos os elementos previstos no escopo inicial, incluindo o sistema de movimentação em múltiplas direções, os diferentes tipos de disparo, a geração progressiva de inimigos e a utilização de *power ups* que alteram temporariamente o desempenho do jogador.

Os resultados alcançados permitem observar, de forma concreta, como classes e objetos podem estruturar de maneira organizada elementos essenciais de um jogo digital. A interação entre as entidades Player, Enemy, Bullet e PowerUp evidencia o uso adequado de encapsulamento, modularização e responsabilidade distribuída entre componentes. Além disso, o sistema de progressão por ondas demonstra o funcionamento de ciclos contínuos de atualização, característicos de jogos desenvolvidos com bibliotecas multimídia.

A seguir, são apresentadas as capturas de tela que ilustram o funcionamento geral do jogo, bem como alguns de seus principais elementos visuais e mecânicos.

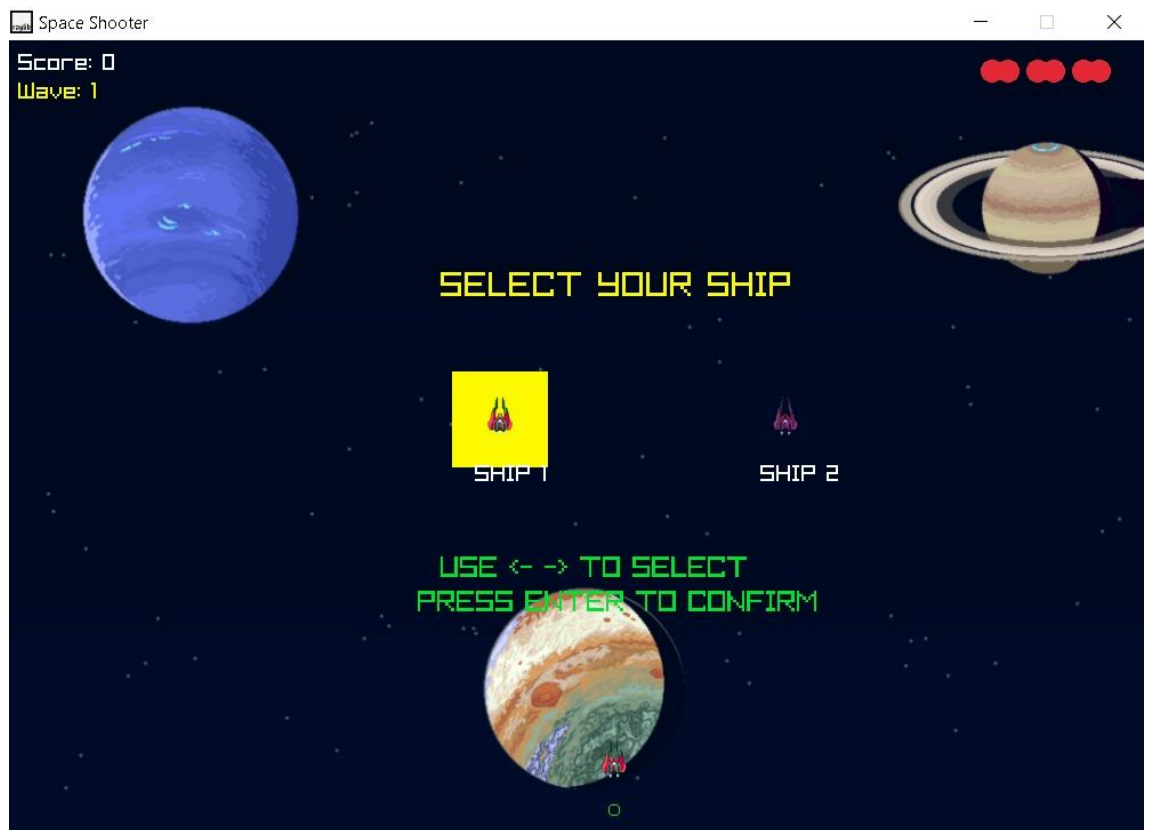


Figura 2 Seletor de Skin

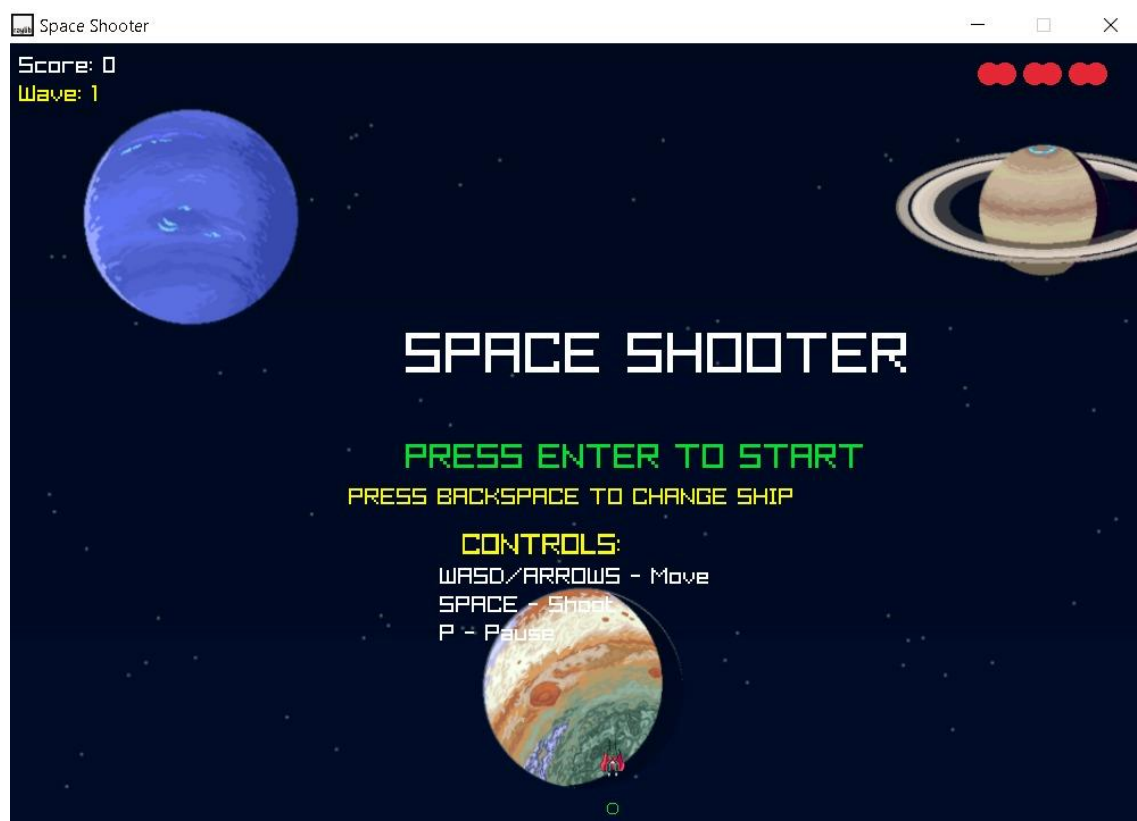


Figura 3 Menu Principal



Figura 4 Jogo Rodando

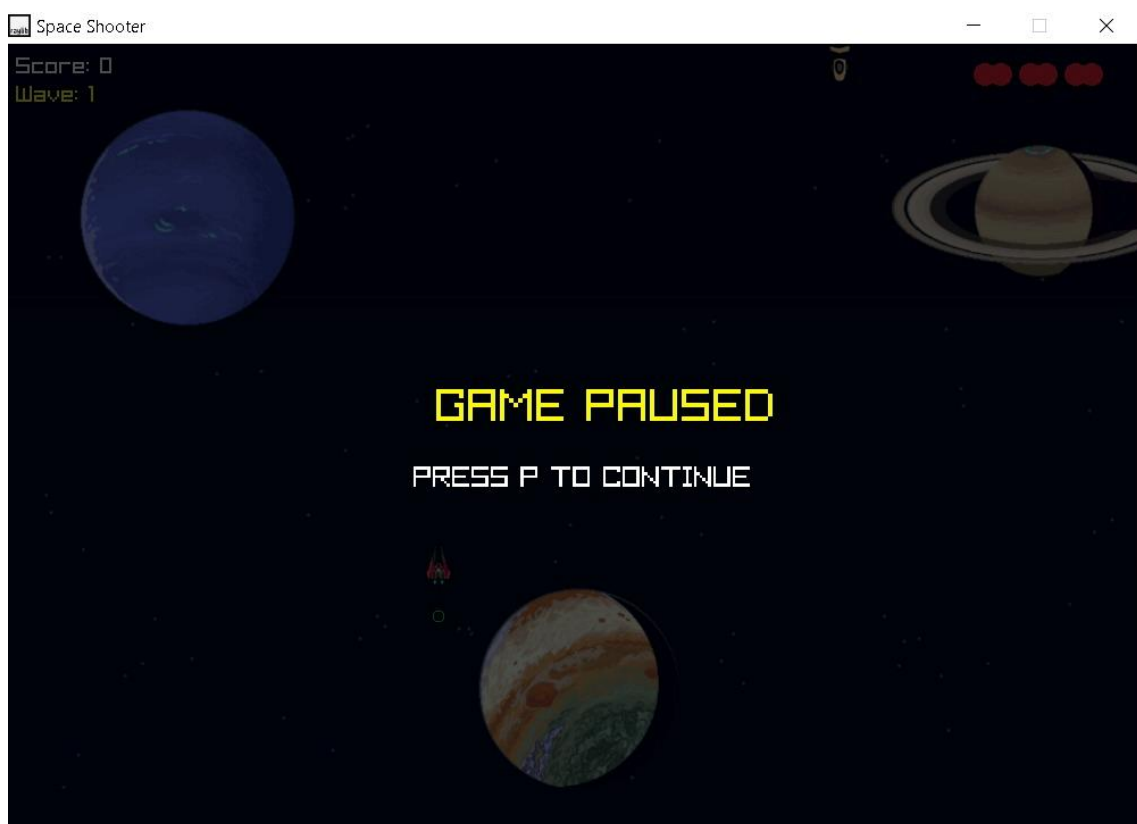


Figura 5 Jogo Pausado

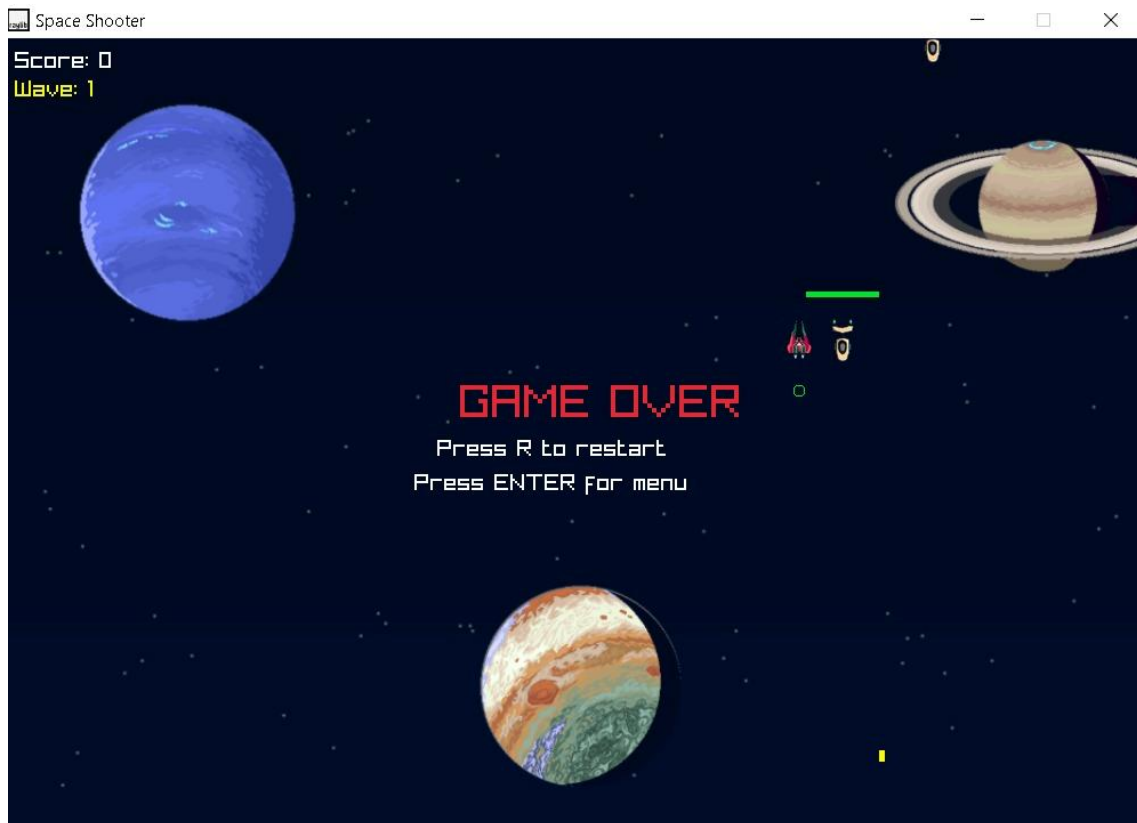


Figura 6 Jogo Perdido

4. Conclusão

O desenvolvimento do jogo *shoot 'em up* proposto neste trabalho permitiu a aplicação prática dos conceitos fundamentais de Programação Orientada a Objetos, conforme previsto nos objetivos da disciplina. A elaboração de um produto interativo completo exigiu a organização modular do código, a definição clara das responsabilidades de cada classe e a implementação de mecanismos de interação entre objetos, elementos essenciais para o entendimento e uso adequado desse paradigma de programação.

A utilização da linguagem C++ em conjunto com a biblioteca Raylib demonstrou a viabilidade de construir um jogo funcional com foco em desempenho, simplicidade estrutural e controle direto sobre os recursos gráficos e lógicos do sistema. O projeto resultante apresenta movimentação responsiva, diferentes padrões de ataque, múltiplos tipos de power ups e um sistema de progressão por ondas, evidenciando a integração entre componentes e o uso adequado de técnicas de atualização contínua.

O trabalho também contribuiu para o desenvolvimento de habilidades práticas relacionadas à manipulação de sprites, detecção de colisões, gerenciamento de estados e organização do loop principal de jogo. Embora o escopo tenha sido limitado para atender ao caráter didático da proposta, os resultados indicam que o projeto pode ser expandido futuramente por meio da adição de novos tipos de inimigos, sistemas de chefes, modos de jogo alternativos ou mecânicas mais complexas de progressão.

Conclui-se, portanto, que o objetivo principal de demonstrar a aplicação dos conceitos de Programação Orientada a Objetos em um projeto funcional foi plenamente alcançado. O jogo desenvolvido cumpre seu papel como atividade integradora, permitindo que o aluno pratique de forma concreta os conteúdos estudados e desenvolva competências essenciais para projetos maiores e mais elaborados no futuro.

5. Referências Bibliográficas

DEITEL, P. J.; DEITEL, H. M. *C++ How to Program*. 10. ed. Boston: Pearson, 2017.

RAYLIB. *Raylib: C library to enjoy videogames programming*. Disponível em: <https://www.raylib.com>. Acesso em: 26 de Agosto de 2025

STROUSTRUP, B. *The C++ Programming Language*. 4. ed. Boston: Addison-Wesley, 2013.