

Space Shooter

Desafios Dinâmicos e Inteligência Artificial no Space Shooter

1st Ana Flávia de Souza Ribeiro

email@example.com

2st Luiz Victor Aldenucci

laldenucci@sga.pucminas.br

3st Rafaella Cristina de Sousa Sacramento

rafaella.sacramento@sga.pucminas.br

4st Samuel Ribeiro de Freitas

email@example.com

Abstract—Este projeto apresenta o desenvolvimento de um jogo eletrônico do tipo space shooter em um ambiente 2D, onde o jogador controla uma nave espacial enfrentando desafios que exigem navegação precisa, combate estratégico e interação dinâmica com o ambiente. O jogo emprega algoritmos e técnicas de inteligência artificial para aprimorar a jogabilidade e a complexidade dos desafios.

A movimentação da nave é controlada pelas setas direcionais, enquanto os disparos são ativados pela barra de espaço. A geração de consumíveis utiliza estruturas de grafos, aplicando algoritmos de busca em largura e profundidade para posicionar estrategicamente os itens coletáveis, maximizando o desafio. Além disso, a inteligência artificial é aplicada ao comportamento dos inimigos e bosses, incluindo o uso de algoritmos heurísticos, como a distância euclidiana, para guiar mísseis que perseguem o jogador e desviam de obstáculos.

Essas implementações não apenas aumentam a dificuldade progressiva do jogo, mas também oferecem uma experiência mais imersiva. O objetivo final é sobreviver aos desafios, derrotar os inimigos e alcançar a vitória, explorando o equilíbrio entre precisão, estratégia e tomada de decisões rápidas.

Index Terms—Artificial Intelligence, Space Shooter, Digital games, Parameter Manipulation

I. INTRODUÇÃO

Os jogos eletrônicos têm evoluído significativamente ao longo das últimas décadas, oferecendo experiências imersivas e desafiadoras aos jogadores. No gênero arcade, o estilo space shooter permanece popular devido à sua combinação de reflexos rápidos, navegação precisa e combate dinâmico. Este artigo apresenta o desenvolvimento de um jogo de aventura espacial em um ambiente 2D, onde o jogador assume o papel de um astronauta enfrentando uma invasão alienígena. O objetivo é sobreviver a um cenário repleto de obstáculos e inimigos, utilizando habilidades estratégicas e reflexos apurados.

A jogabilidade se destaca pela simplicidade viciante e pela mecânica de coleta de consumíveis que aprimoram o desempenho da nave. Esses elementos criam uma experiência envolvente e desafiadora, com dificuldade crescente para testar continuamente as habilidades do jogador. Além disso, foram aplicadas técnicas avançadas de inteligência artificial (IA) e estruturas de dados baseadas em grafos, permitindo dinâmicas de jogo inovadoras.

No desenvolvimento, utilizamos grafos para determinar o posicionamento estratégico de consumíveis, explorando

métodos de busca em largura e profundidade. Cada nó do grafo representa um ponto estratégico no cenário, sendo os consumíveis gerados nos locais mais desafiadores, aumentando a complexidade do jogo. A aplicação de IA se tornou um diferencial significativo: enquanto jogos que incorporam IA adaptativa são capazes de acompanhar e mimetizar a progressão do jogador [3], este trabalho implementou o algoritmo de busca A* com heurística de distância euclidiana para guiar os mísseis dos chefes. Esses mísseis não apenas perseguem o jogador de forma eficiente, mas também desviam de obstáculos, proporcionando uma jogabilidade mais dinâmica e estratégica.

O objetivo principal deste projeto foi integrar algoritmos de IA e estruturas de dados como grafos em um jogo arcade, explorando seu potencial para criar uma experiência interativa, desafiadora e envolvente para o jogador.

A. Descrição e Solução do Problema

Uma das **limitações** mais notáveis em jogos space shooter tradicionais é sua previsibilidade, o que frequentemente resulta em uma experiência repetitiva. Desafios e comportamentos de inimigos tendem a ser lineares e estáticos, o que reduz a complexidade e a capacidade de surpreender o jogador, prejudicando a sensação de imersão.

Além disso, a falta de sistemas de inteligência artificial adaptativa impede que o jogo acompanhe o progresso do jogador, criando desafios que sejam ao mesmo tempo acessíveis e estimulantes. Sem essa dinâmica, a experiência se torna monótona, afastando jogadores que buscam variação e estratégias mais desafiadoras.

Outro **problema** reside na aleatoriedade descontrolada na geração de consumíveis e elementos estratégicos, o que pode resultar em cenários mal equilibrados ou desafiadores de forma inconsistente. Isso compromete a progressão no jogo e diminui o senso de recompensa e conquista do jogador.

Portanto, a solução proposta para os problemas mencionados é superar a previsibilidade e as limitações dos designs tradicionais, desenvolvendo um jogo mais equilibrado, imersivo e dinâmico, capaz de proporcionar uma experiência mais rica e inovadora [1].

B. Justificativa

O desenvolvimento deste jogo space shooter visa proporcionar uma experiência desafiadora e imersiva, combinando elementos clássicos de jogabilidade arcade com soluções modernas de Inteligência Artificial (IA). Os jogos são uma forma de entretenimento que envolvem ações e decisões orientadas por objetivos e desafios, todos baseados em regras. Além disso, eles promovem habilidades como reflexos rápidos, coordenação motora e estratégias de resolução de problemas [2].

Do ponto de vista técnico, o projeto busca explorar conceitos computacionais avançados, como grafos e algoritmos de busca, aplicados de maneira prática e interativa. A utilização de algoritmos como busca em largura e profundidade para a geração de consumíveis, bem como o A* com heurística para a movimentação de inimigos, oferece uma oportunidade para demonstrar como técnicas de IA podem aprimorar a jogabilidade e aumentar a dificuldade de forma dinâmica.

Além disso, a criação de um ambiente desafiador e adaptável, onde os jogadores enfrentam inimigos inteligentes e obstáculos bem planejados, contribui para a evolução dos jogos independentes e promove a inovação no design de jogos 2D. Este projeto também serve como exemplo acadêmico de como conceitos teóricos podem ser aplicados para resolver problemas práticos no desenvolvimento de jogos, tornando-o relevante para as áreas de computação, design e IA.

II. DESENVOLVIMENTO

O jogo foi desenvolvido utilizando a linguagem de programação Csharp, destacando-se por seu desempenho robusto e pela integração eficiente com o motor de jogos Unity, proporcionando uma experiência fluida. A escolha de Csharp permitiu a implementação de algoritmos avançados de inteligência artificial e buscas em grafos para a geração de consumíveis, além de facilitar a organização do projeto com práticas de programação orientada a objetos.

Para organizar e gerenciar o processo de desenvolvimento, foi adotada a metodologia Scrum, que favoreceu uma abordagem iterativa, ágil e incremental. Por meio de ciclos curtos de desenvolvimento, conhecidos como sprints, foi possível planejar, implementar e revisar funcionalidades regularmente, assegurando que os objetivos principais fossem atingidos dentro dos prazos estabelecidos.

A. Metodologia

O desenvolvimento do jogo seguiu uma arquitetura modular, onde diversas classes foram criadas para gerenciar funcionalidades específicas de maneira isolada e eficiente. A seguir, são descritos os principais módulos implementados:

- Gerenciamento de Fluxo de Jogo

- 1) **Classe Menu:** Controla a navegação entre diferentes cenas do jogo e fornece opções para iniciar (LoadScene) ou sair do jogo (SairJogo).
- 2) **Classe Play:** Centraliza a reprodução de áudio, conectando-se ao AudioManager para garantir que

os efeitos sonoros sejam consistentes durante a jogabilidade.

- Jogabilidade e Controle do Jogador

- 1) **Classe Player:** Gerencia a movimentação do jogador com base na entrada do teclado (Movimentacao) e permite disparos (Tiro), alternando o estado do flash visual associado a disparos (DesligarFlash). Lida com colisões (OnTriggerEnter2D) para verificar interações com inimigos, obstáculos e power-ups. Atualiza a barra de vida do jogador com o método UpdateHealthBar.

- Inimigos e Chefões

- 1) **Classe Boss:** Implementa o comportamento dos chefões, incluindo movimentos controlados (Move) e animações de entrada (EntryAnimation). Gerencia ataques com projéteis (Tiro), que são instanciados em um local definido (Instantiate), e obstáculos adicionais, como meteoros (SpawnMeteors). Utiliza a detecção de colisões (OnTriggerEnter2D) para interagir com projéteis do jogador e obstáculos no cenário, criando efeitos visuais (Instantiate(hitPrefab)) e controlando o estado dos meteoros gerados.
- 2) **Classe BossHealthBar:** Controla visualmente a barra de vida do boss com os métodos SetMaxHealth e SetHealth.
- 3) **Classe CriarBoss:** Gerencia a lógica de introdução dos chefões, alternando entre diferentes prefabs (bossPrefab1 e bossPrefab2) e exibindo uma contagem regressiva com textos (prepareTextMesh). Implementa métodos como ShowPrepareAndCountdown para criar momentos de antecipação antes do surgimento do boss.

- Geração de Objetos no Cenário

- 1) **Classe CoinManager:** Gera moedas em locais estratégicos utilizando busca em largura (BreadthFirstSearch) e profundidade (DepthFirstSearch). Implementa o método FindFarthestWaypoint para posicionar moedas no waypoint mais distante do jogador, aumentando a dificuldade. Controla a ativação da geração de moedas com SetGeracaoAtiva.
- 2) **Classe Gerar e GerarCoins:** Responsáveis por inicializar e atualizar a geração de inimigos e itens no cenário, com métodos como Gera e SpawnCoin.

- Movimentação e Obstáculos

- 1) **Classes BGinfinito e MovLinear:** Implementam a movimentação dos planos de fundo e elementos do jogo, criando um efeito de continuidade.
- 2) **Classe MovSenoidal:** Controla padrões de movimento baseados em ondas senoidais, adicionando imprevisibilidade aos obstáculos no cenário.

- Inteligência Artificial e Algoritmos

- 1) **IA dos Chefões:** Utiliza o algoritmo de busca A* com heurística euclidiana para guiar mísseis

que perseguem o Isso inclui desviar de obstáculos durante o trajeto, aumentando a complexidade e o nível de desafio.

- 2) **Gerenciamento de Moedas:** A lógica de busca no grafo permite que consumíveis sejam gerados de forma inteligente, melhorando a experiência do jogador ao incentivar movimentação estratégica pelo cenário.

- Pontuação e Progressão

- 1) **Classe Pontuacao:** Monitora e atualiza a pontuação do jogador com base no progresso e nas ações realizadas.

- 2) **Classe Recorde:** Salva e exibe os recordes alcançados pelos jogadores.

- Controle de Efeitos Visuais

- 1) **Classe ControleExplosao:** Gera efeitos de explosões no cenário com o método Create.

- 2) **Classe GirarTiro:** Controla a rotação visual dos projéteis disparados.

- Sistema de Colisões Colisões são gerenciadas em diversas classes, como **Coin**, **Boss**, e **inimigo**, por meio de métodos como `OnTriggerEnter` e `OnTriggerEnter2D`, que verificam as interações entre o jogador, inimigos, obstáculos e consumíveis.

III. RESULTADOS

Os resultados do desenvolvimento destacam a integração de algoritmos de inteligência artificial para aprimorar a experiência do jogo. O sistema de movimentação dos mísseis dos Bosses foi desenvolvido utilizando o algoritmo A* combinado com a heurística de distância euclidiana, permitindo que os mísseis rastreiem o jogador com eficiência enquanto desviam de obstáculos no caminho. Essa funcionalidade cria um desafio dinâmico e estratégico, forçando o jogador a se adaptar rapidamente durante as batalhas contra os Bosses. Além disso, a geração de meteoros e o sistema de colisões foram integrados para aumentar a complexidade, tornando cada confronto único e imersivo. A IA aplicada não apenas adiciona realismo, mas também eleva o nível de dificuldade, proporcionando uma experiência de jogo mais envolvente.

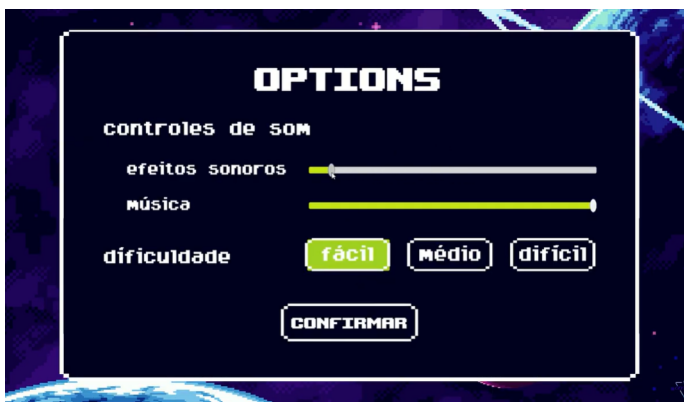


Figura 1: Tela inicial do jogo, onde o jogador pode selecionar

o nível de complexidade entre fácil, médio e difícil. A interface apresenta três botões para a escolha do nível, com uma breve explicação sobre cada um, permitindo ao jogador adaptar a dificuldade do jogo conforme sua experiência e preferência.

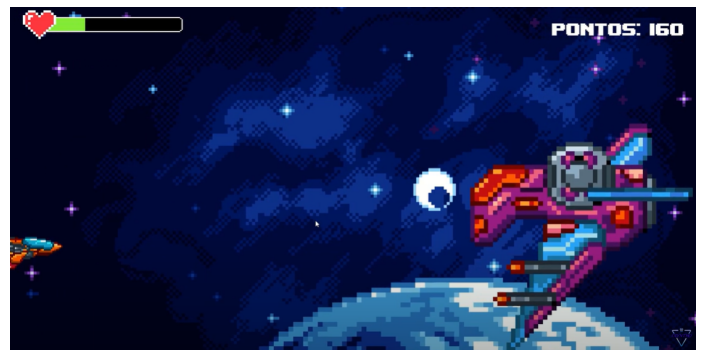


Figura 2: Confronto entre a nave do jogador e o boss, que utiliza inteligência artificial para disparar mísseis guiados. O boss está posicionado no centro da tela, enquanto a nave se movimenta ao redor para se esquivar dos mísseis. A IA do boss permite que os mísseis sigam a posição da nave e desviem de obstáculos, criando uma dinâmica de combate. A tela exibe uma intensa troca de tiros e estratégias de evasão.

O estado do Boss é gerenciado pela quantidade de pontos de vida restantes, e a derrota é desencadeada quando este valor chega a zero.

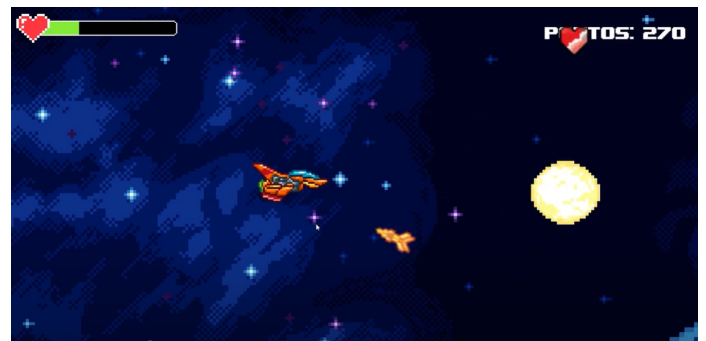


Figura 3: Explosão do boss após a nave do jogador ter causado dano suficiente ou atingido seus pontos vitais. A tela mostra a animação de uma grande explosão, com efeitos visuais de destruição e fragmentos sendo dispersos no espaço.

Esses resultados ressaltam a eficácia da abordagem adotada, garantindo que a jogabilidade seja tanto desafiadora quanto engajante, elevando a complexidade das batalhas e criando um ambiente dinâmico e interativo.

IV. CONCLUSÃO

O desenvolvimento deste jogo utilizando Csharp e Unity demonstra como ferramentas modernas podem ser aplicadas na criação de experiências interativas e desafiadoras. A implementação de técnicas de inteligência artificial não só otimizou a movimentação estratégica dos mísseis, como

também adicionou dinamismo ao jogo por meio da interação com obstáculos e elementos do ambiente.

Assim, foi possível alcançar o objetivo de oferecer uma experiência imersiva e envolvente para o jogador, destacando o potencial de aliar boas práticas de desenvolvimento a soluções inovadoras de IA. Este trabalho exemplifica como a tecnologia pode ser utilizada de maneira criativa e eficiente para desenvolver jogos que combinam entretenimento e desafio de forma equilibrada.

REFERENCES

- [1] SEGUNDO, Caetano Vieira Neto et al. Dynamic difficulty adjustment through parameter manipulation for Space Shooter game. Proceedings of SB games, 2016.
- [2] RIOS, Vanessa; SANTOS, Andressa Mota; DE SOUZA, Carlos Eduardo. Aprendizagem Baseada em Jogos Digitais: uma experiência realizada a partir Game Engine GDevelop 5. RE@ D-Revista de Educação a Distância e Elearning, p. e202428-e202428, 2024.
- [3] FILHO, José Ahirton Batista Lopes. Panorama e Perspectivas da Utilização dos Conceitos de Adaptabilidade em IA para Maior Imersão em Jogos Digitais.