

Eduarda Mendes Pinto Barbosa

A Inteligência Artificial em Jogos Digitais e sua Influência na Imersão e Experiência do Jogador

Belo Horizonte, MG
2025

Eduarda Mendes Pinto Barbosa

A Inteligência Artificial em Jogos Digitais e sua Influência na Imersão e Experiência do Jogador

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Orientador: Pedro Olmo Stancioli Vaz De Melo

Belo Horizonte
2025

Resumo

Este trabalho investiga o impacto de diferentes técnicas de Inteligência Artificial (IA) na experiência e na imersão do jogador em jogos digitais. Partindo da fundamentação teórica e das hipóteses levantadas no MSI I, que sugeriam que arquiteturas de IA mais avançadas tendem a gerar comportamentos mais naturais, variados e responsivos, o presente estudo desenvolveu um protótipo jogável composto por duas fases distintas. Embora diferentes em ambientação e progressão, ambas seguem uma estrutura linear inspirada em jogos comerciais, permitindo uma experiência mais próxima do contexto real de gameplay. Sua principal diferença reside na lógica de comportamento dos inimigos: a Fase 1 utiliza uma Máquina de Estados Finitos (FSM), enquanto a Fase 2 implementa uma Behavior Tree.

Através de um experimento com jogadores reais, foram coletados dados quantitativos e qualitativos sobre percepções de engajamento, imersão e comportamento dos NPCs. Os resultados foram analisados por meio de estatísticas descritivas e do teste Qui-Quadrado. Os achados confirmam que a implementação com Behavior Tree proporciona uma experiência substancialmente mais rica, sendo associada a maior diversidade comportamental, maior responsividade dos inimigos, decisões mais coerentes e níveis mais elevados de imersão. O estudo demonstra, de forma prática e empírica, que técnicas de IA mais robustas elevam significativamente a qualidade da experiência interativa, reforçando a importância da IA como elemento central de design em jogos digitais.

Palavras-chave: inteligência artificial, imersão, jogos digitais, behavior tree, FSM, Maquina de Estados Finita, Árvore de comportamento.

Sumário

| | | |
|---------------|--|-----------|
| Resumo | 1 | |
| 1 | Introdução | 3 |
| 1.1 | Objetivo Geral | 3 |
| 1.2 | Objetivos Específicos | 3 |
| 2 | Referencial Teórico | 4 |
| 2.1 | Inteligência Artificial em Jogos Digitais | 4 |
| 2.2 | Soluções Existentes e Contribuição do Trabalho | 4 |
| 2.3 | Principais Técnicas de IA Utilizadas em Jogos Digitais | 5 |
| 2.3.1 | Máquina de Estados Finitos (FSM) | 5 |
| 2.3.2 | Árvore de Comportamento (Behavior Tree) | 6 |
| 2.4 | Características | 7 |
| 2.5 | Imersão | 8 |
| 2.5.1 | Modelo teórico | 9 |
| 3 | Metodologia | 10 |
| 3.1 | Tipo de Pesquisa | 10 |
| 3.2 | Descrição do Jogo e das Implementações | 11 |
| 3.2.1 | Versão com FSM | 11 |
| 3.2.2 | Versão com Behavior Tree | 14 |
| 3.3 | Coleta de Dados e Participantes | 17 |
| 3.3.1 | Participantes | 17 |
| 3.3.2 | Instrumento de Coleta (Questionário) | 17 |
| 3.3.3 | Procedimentos de Aplicação | 18 |
| 3.4 | Tratamento e Análise dos Dados | 19 |
| 3.4.1 | Teste qui-quadrado | 19 |
| 3.4.2 | Procedimentos de Implementação | 20 |
| 4 | Resultados | 21 |
| 4.1 | Análise Descritiva dos Dados | 21 |
| 4.2 | Análise Inferencial: Teste Qui-Quadrado | 22 |
| 5 | Conclusão | 25 |
| 5.1 | Trabalhos Futuros | 26 |
| 6 | Referências Bibliográficas | 27 |

1 Introdução

Nas últimas décadas, a indústria dos jogos digitais passou por uma transformação significativa, impulsionada pelo avanço tecnológico e pela crescente demanda por experiências cada vez mais realistas e envolventes. O jogador contemporâneo não busca apenas desafios mecânicos, mas também mundos virtuais capazes de reagir às suas ações de forma convincente, criando uma sensação de presença e pertencimento ao ambiente digital. Nesse cenário, a imersão torna-se um dos elementos centrais para o sucesso de um jogo, uma vez que está diretamente relacionada ao engajamento e à satisfação do usuário.

Entre os diversos recursos que contribuem para aumentar esse nível de imersão, a inteligência artificial (IA) se destaca como um dos pilares fundamentais. Mais do que um componente técnico, a IA é um instrumento de game design, responsável por dar vida a personagens não jogáveis (NPCs), modelar comportamentos dinâmicos e gerar respostas adaptativas que enriquecem a experiência interativa. Quando bem aplicada, a IA permite que NPCs deixem de ser previsíveis e mecânicos, passando a desempenhar papéis mais críveis dentro da narrativa e da jogabilidade, o que eleva significativamente a qualidade da experiência do jogador.

No trabalho anterior (MSI I), foi realizada uma investigação teórica sobre as principais técnicas de inteligência artificial aplicadas a jogos digitais, entre elas, as Máquinas de Estados Finitos (Finite State Machines – FSM), as Árvores de Comportamento (Behavior Trees), abordagens baseadas em aprendizado de máquina e a recente IA generativa. A partir dessa análise, verificou-se que quanto maior a capacidade de adaptação e autonomia do sistema de IA, mais significativa tende a ser sua influência sobre a imersão e a experiência do jogador.

Dando continuidade a essa pesquisa, o presente trabalho propõe uma abordagem prática voltada à implementação e comparação de duas técnicas específicas de IA, a FSM e a Behavior Tree, em um jogo digital previamente desenvolvido. O objetivo é avaliar como diferentes níveis de complexidade no comportamento dos NPCs impactam a percepção dos jogadores em relação ao realismo, engajamento e imersão durante a jogabilidade.

Assim, este trabalho busca responder à seguinte questão: como diferentes níveis de complexidade da IA influenciam a experiência e a imersão do jogador? A expectativa é comprovar, por meio da prática, a hipótese levantada no MSI I de que técnicas mais avançadas de inteligência artificial tendem a proporcionar maior realismo, engajamento e imersão no contexto dos jogos digitais.

1.1 Objetivo Geral

Implementar e comparar duas técnicas de inteligência artificial (FSM e Behavior Tree) em um jogo digital simples, avaliando a influência da complexidade da IA na imersão e na percepção do jogador por meio de um questionário aplicado após a jogabilidade.

1.2 Objetivos Específicos

- Aprimorar e expandir um jogo digital previamente desenvolvido como ambiente de experimentação.
- Implementar uma IA baseada em FSM (Finite State Machine), com comportamentos básicos e previsíveis.

- Implementar uma IA baseada em Árvore de Comportamento (Behavior Tree), permitindo maior flexibilidade e adaptabilidade nas ações do NPC.
- Aplicar um questionário a jogadores que testarem o protótipo, coletando percepções sobre realismo, imersão e engajamento.
- Comparar qualitativamente os resultados obtidos, destacando as diferenças percebidas entre as duas abordagens de IA.
- Avaliar se a maior complexidade da IA contribui para uma experiência de jogo mais envolvente e realista.

2 Referencial Teórico

2.1 Inteligência Artificial em Jogos Digitais

A inteligência artificial (IA) aplicada a jogos digitais refere-se ao uso de algoritmos e técnicas computacionais para simular comportamentos inteligentes em personagens e sistemas dentro de um jogo. Em geral, a IA é responsável por controlar personagens não jogáveis (NPCs), criar estratégias para inimigos, organizar eventos dinâmicos e até gerar narrativas que respondem ao comportamento do jogador.

No contexto do game design, a IA vai além da criação de desafios mecânicos. Ela tem o papel de contribuir diretamente para a construção de um mundo mais crível e interativo, oferecendo experiências personalizadas, instigantes e imersivas. Quanto mais a IA for capaz de simular decisões e respostas realistas, maior tende a ser a conexão emocional do jogador com o jogo.

2.2 Soluções Existentes e Contribuição do Trabalho

Para embasar teoricamente este trabalho, foram analisadas obras e estudos de autores renomados na área de inteligência artificial aplicada a jogos digitais. Yannakakis e Togelius (2018), em *Artificial Intelligence and Games*, defendem que a IA vai além do controle de personagens não jogáveis (NPCs), sendo uma ferramenta essencial no design de mecânicas, narrativa e experiência do jogador. Eles enfatizam o uso da IA para adaptar o jogo ao comportamento do jogador, promovendo maior imersão e equilíbrio nos desafios.

Millington (2016), por sua vez, abordam em *Artificial Intelligence for Games* técnicas clássicas como Máquinas de Estados Finitos (FSM), árvores de decisão e sistemas baseados em regras, ressaltando a importância dessas abordagens pela sua simplicidade, previsibilidade e baixo custo computacional, características úteis para jogos com comportamentos mais fixos.

Além dessas contribuições clássicas, estudos mais recentes destacam a consolidação das Máquinas de Estados Finitos e das Árvores de Comportamento como arquiteturas amplamente adotadas na indústria de jogos devido à sua robustez e versatilidade. Essas técnicas continuam sendo utilizadas tanto em títulos comerciais quanto em projetos acadêmicos, pois permitem estruturar comportamentos de NPCs de forma modular, previsível e escalável.

Nesse sentido, os modelos apresentados pela literatura serviram como referência direta para este trabalho, orientando a implementação prática das duas abordagens no jogo de-

senvolvido neste semestre e evidenciando como soluções consolidadas podem ser aplicadas de maneira eficiente no design de comportamentos inimigos.

2.3 Principais Técnicas de IA Utilizadas em Jogos Digitais

No MSI I, foi realizada uma análise abrangente de quatro técnicas de Inteligência Artificial aplicadas ao desenvolvimento de jogos digitais: Máquinas de Estados Finitos (FSM), Árvores de Comportamento (Behavior Trees), Algoritmos de Pathfinding e IA baseada em Aprendizado. Neste trabalho, retomamos apenas duas dessas abordagens, FSM e Behavior Trees, que constituem o foco principal da pesquisa atual. Serão apresentadas suas definições, funcionamento, vantagens e desvantagens, que evidenciam como essas técnicas contribuem para a construção de comportamentos inteligentes e experiências mais imersivas nos jogos digitais.

2.3.1 Máquina de Estados Finitos (FSM)

1. Definição: As Máquinas de Estados Finitos (FSM) são modelos computacionais que descrevem o comportamento de sistemas através de uma série de estados bem definidos e das transições que permitem a passagem de um estado para outro. No contexto dos jogos, esse modelo é largamente empregado para controlar as ações de NPCs, permitindo que eles executem comportamentos distintos de acordo com determinadas condições.

A FSM estrutura a lógica do NPC de forma sequencial e previsível, garantindo que em qualquer momento ele esteja realizando apenas uma ação específica. Segundo Yannakakis e Togelius (2018), esse método de IA dominou os processos de controle e tomada de decisão dos NPCs até meados dos anos 2000, sendo uma das abordagens mais utilizadas nesse período e, ainda hoje, é uma solução bastante aplicada em projetos que exigem simplicidade, clareza e eficiência.

2. Funcionamento: O funcionamento de uma FSM se baseia em três componentes principais:

- Estados: cada estado representa um comportamento ou atividade que o NPC pode desempenhar, como "patrulhar", "perseguir".
- Transições: são condições pré-estabelecidas que determinam quando um NPC deve mudar de um estado para outro.
- Ações: correspondem às tarefas específicas realizadas enquanto o NPC permanece em determinado estado. Por exemplo, enquanto estiver no estado "patrulhar", o NPC pode seguir um caminho predefinido, observando o ambiente em busca de ameaças.

Esse sistema garante que o NPC siga uma sequência lógica de ações, reagindo de maneira reativa aos eventos do ambiente, o que é ideal para jogos que necessitam de comportamentos claros e bem estruturados.

3. Vantagens: As FSMs oferecem várias vantagens que justificam seu amplo uso, especialmente em jogos com sistemas de IA mais simples:

- Facilidade de implementação: o modelo é intuitivo e visualmente representável através de diagramas.
- Facilidade na depuração: identificar e corrigir erros se torna mais simples.
- Controle total sobre o comportamento: o desenvolvedor consegue prever com precisão como o NPC se comportará em cada situação.
- Baixo custo computacional: a execução de FSMs exige pouco processamento.

4. Desvantagens:

Apesar de suas qualidades, as FSMs apresentam algumas limitações importantes, especialmente quando aplicadas a sistemas mais complexos:

- Baixa flexibilidade: uma vez que os comportamentos são definidos previamente, o NPC não consegue se adaptar a novas situações sem que o desenvolvedor altere manualmente as regras e transições.
- Problemas de escalabilidade: à medida que o número de estados e transições cresce, o sistema se torna difícil de manter, sofrendo do chamado problema da "explosão de estados", onde pequenas mudanças podem afetar muitas partes do modelo.
- Comportamentos previsíveis e repetitivos: como a lógica é fixa, os NPCs acabam apresentando padrões facilmente detectáveis, o que pode reduzir o desafio e a imersão do jogador.

2.3.2 Árvore de Comportamento (Behavior Tree)

1. **Definição:** Uma Árvore de Comportamento é um modelo hierárquico que organiza e controla a lógica de decisão de IA. O princípio central é a modularidade, onde cada nó na árvore representa uma tarefa ou decisão específica, e a estrutura geral da árvore dita o fluxo de execução (Moreira, 2022). Diferentemente das Máquinas de Estados Finitos, as árvores de comportamento oferecem maior flexibilidade e escalabilidade, tornando-as ideais para comportamentos de IA mais complexos e variados.

2. **Funcionamento:** O funcionamento de uma Árvore de Comportamento baseia-se na execução hierárquica e sequencial dos nós, iniciando sempre a partir da raiz da árvore. Os nós são divididos em dois grupos principais: nós de controle e nós folha. Os nós de controle definem como os nós filhos devem ser executados e incluem tipos como sequência (Sequence), seleção (Selector), paralelo (Parallel) e decoradores (Decorators). Já os nós folha representam ações específicas do agente como mover, atacar ou fugir, ou condições a serem verificadas, como detectar um inimigo ou checar se a vida está baixa.

Durante a execução, a árvore é percorrida em ordem predefinida e os nós retornam um dos três possíveis estados: Sucesso, Falha ou em Execução. Por exemplo, em um nó do tipo Sequence, os filhos são executados da esquerda para a direita, e a execução continua até que um nó retorne Falha ou em Execução, se todos retornarem Sucesso, o nó Sequence retorna Sucesso. Já um nó Selector tenta os filhos em ordem até encontrar um que retorne Sucesso ou em Execução, retornando Falha apenas se todos falharem.

Os nós decoradores modificam o comportamento de um único nó filho, podendo inverter o resultado de sua execução, limitar o número de repetições ou introduzir condições adicionais. Já os nós paralelos tentam executar múltiplos nós filhos simultaneamente, sendo úteis para agentes que precisam realizar várias tarefas ao mesmo tempo, como patrulhar e observar.

Esse modelo permite que comportamentos complexos sejam compostos por partes menores e reutilizáveis, promovendo clareza, manutenção facilitada e expansão modular da lógica de IA. Além disso, as árvores de comportamento podem ser interrompidas ou reavaliadas a cada ciclo do jogo, o que permite comportamentos dinâmicos e responsivos ao ambiente do jogador.

3. Vantagens: As árvores de comportamento oferecem diversas vantagens que as consolidaram como uma das arquiteturas de IA mais comuns em jogos:

- Reatividade: As árvores de comportamento respondem rapidamente às mudanças no ambiente do jogo, reavaliando continuamente as condições e ajustando as ações em tempo real.
- Modularidade: Permitem que comportamentos sejam divididos em partes independentes e reutilizáveis, facilitando ajustes e manutenção.
- Escalabilidade: Suportam a criação de comportamentos simples ou complexos, sendo ideais para projetos de qualquer porte.
- Adaptabilidade: Respondem dinamicamente às mudanças do ambiente, permitindo comportamentos mais inteligentes e reativos.

4. Desvantagens: Apesar das suas vantagens, as árvores de comportamento também apresentam algumas desvantagens:

- Complexidade crescente: Em jogos com muitos comportamentos interdependentes, a árvore pode crescer demais e se tornar difícil de gerenciar e visualizar.
- Requer planejamento detalhado: A construção eficaz exige uma estrutura bem pensada desde o início, sem isso, o comportamento pode se tornar inconsistente ou confuso.
- Dificuldade de manutenção em árvores muito grandes: À medida que a árvore cresce, localizar e ajustar comportamentos específicos pode demandar tempo e atenção extra.

2.4 Características

Durante o desenvolvimento do MSI I, foi realizada uma análise teórica aprofundada sobre a aplicação da Inteligência Artificial em jogos digitais. A partir desse estudo, foi possível identificar e sistematizar um conjunto de características essenciais da IA que influenciam diretamente a imersão e a experiência do jogador. Esses aspectos, observados como resultados da pesquisa anterior, representam os principais elementos que tornam a IA capaz de simular comportamentos inteligentes, reagir de forma dinâmica ao ambiente e contribuir para experiências mais realistas e envolventes. Entre eles, destacam-se a responsividade, a tomada de decisão, a adaptação, a navegação e a simulação de comportamento.

A responsividade refere-se à capacidade da IA de reagir rapidamente às ações do jogador, oferecendo respostas imediatas e coerentes que tornam a experiência mais fluida e envolvente. Quando o jogo responde de maneira precisa e natural às interações do usuário, o jogador sente que suas ações têm impacto real no ambiente virtual, o que reforça a sensação de controle e presença. Essa troca constante entre ação e reação é essencial para sustentar o ritmo do jogo e manter o jogador imerso, já que uma IA responsiva cria um fluxo contínuo de interações significativas.

A tomada de decisão é responsável por definir como os NPCs escolhem suas ações com base nas informações do ambiente e em seus próprios estados internos. Essa característica garante que os NPCs ajam de forma coerente e contextual, reagindo adequadamente às situações do jogo. A aplicação de algoritmos de decisão permite criar comportamentos mais ricos e imprevisíveis, variando desde estratégias simples até decisões complexas em tempo real. Esse processo torna o jogo mais dinâmico e contribui para a construção de narrativas interativas, em que os personagens aparentam possuir raciocínio próprio.

A adaptação está relacionada à capacidade da IA de ajustar seu comportamento, estratégias ou nível de dificuldade conforme o desempenho e as preferências do jogador. Diferente de sistemas estáticos baseados em regras fixas, uma IA adaptativa é capaz de aprender e evoluir ao longo da partida, moldando-se ao estilo e à habilidade do jogador. Essa característica é essencial para manter o equilíbrio entre desafio e acessibilidade, evitando tanto a frustração quanto a monotonia. Assim, a adaptação funciona como um elemento de personalização que garante que cada jogador tenha uma experiência única e proporcional à sua evolução dentro do jogo.

A navegação diz respeito à habilidade dos personagens controlados pela IA de se movimentarem de forma inteligente e eficiente pelo cenário virtual. Isso inclui encontrar rotas viáveis, desviar de obstáculos e ajustar o trajeto em ambientes dinâmicos. Uma boa navegação é indispensável para que os NPCs ajam de maneira crível e contribuam para a coerência do mundo do jogo. Sem ela, até mesmo algoritmos de decisão bem estruturados podem falhar, já que os personagens não seriam capazes de executar suas ações de forma convincente. Dessa forma, a navegação sustenta grande parte da ilusão de inteligência percebida pelo jogador.

Por fim, a simulação de comportamento busca reproduzir emoções, rotinas e interações que se assemelham às humanas, tornando os personagens virtuais mais críveis e expressivos. Essa característica dá vida ao ambiente do jogo, fazendo com que os NPCs pareçam realmente pensar e sentir. Quando os personagens reagem de forma coerente ao contexto, demonstrando medo, curiosidade, empatia ou agressividade, o jogador passa a perceber o universo do jogo como um espaço habitado por seres autênticos. Essa simulação reforça a imersão e transforma o jogo em uma experiência mais rica, profunda e emocionalmente envolvente.

2.5 Imersão

No MSI I, um dos principais resultados obtidos foi a análise da relação entre o uso da Inteligência Artificial e o aumento da imersão em jogos digitais. Constatou-se que a IA desempenha papel essencial na personalização da experiência do jogador, permitindo ajustar mecânicas, ambientes e narrativas conforme o comportamento individual. Essa adaptação inteligente cria jornadas únicas e aumenta o engajamento, tornando a experiência mais envolvente e significativa (Yannakakis e Togelius, 2018).

Outro ponto identificado foi o impacto da adaptação de dificuldade, da simulação

emocional e da responsividade na sensação de presença do jogador. Quando o jogo reage coerentemente às ações do usuário, equilibra desafio e recompensa e apresenta personagens com reações emocionais realistas, o mundo virtual se torna mais autêntico e dinâmico (Jiang, 2024; Jorg, Normoyle e Safonova, 2012). Além disso, técnicas de aprendizado de máquina mostraram potencial para ampliar essa imersão, pois permitem que a IA aprenda com as interações e evolua junto com o jogador, gerando experiências personalizadas e imprevisíveis.

2.5.1 Modelo teórico

Como parte dos resultados do MSI I, foi desenvolvido um modelo teórico comparativo entre as principais técnicas de IA utilizadas em jogos, FSM, Behavior Tree, Pathfinding, IA baseada em Aprendizado e IA Generativa. Esse modelo analisou dimensões como responsividade, adaptação, tomada de decisão, navegação, simulação de comportamento e impacto na imersão.

Os resultados mostraram que a FSM oferece maior controle e previsibilidade, sendo adequada para comportamentos simples e bem definidos. No entanto, sua estrutura rígida dificulta a adaptação a situações complexas, o que pode limitar a naturalidade das respostas dos agentes virtuais. Já a Behavior Tree apresenta maior flexibilidade e modularidade, permitindo a criação de comportamentos mais dinâmicos e reativos ao ambiente. Essa abordagem favorece a expansão e a manutenção do código, além de proporcionar maior variedade nas ações dos personagens.

Por outro lado, técnicas mais avançadas, como a IA baseada em aprendizado e a IA generativa, demonstraram maior capacidade de adaptação e tomada de decisão complexa, o que contribui para a criação de mundos mais dinâmicos e interações realistas. Esses achados reforçam a importância de alinhar a escolha da técnica de IA aos objetivos de design e à profundidade da experiência desejada pelo jogador.

| Comparação entre Técnicas de Inteligência Artificial em Jogos Digitais | | | |
|--|---|---|---|
| Técnica | Responsividade | Tomada de Decisão | Adaptação |
| FSM (Finite State Machine) | Rápida e previsível para transições conhecidas, porém pouca flexibilidade para novas entradas. | Clara e eficiente para regras fixas, mas limitada para decisões complexas. | Baixa: não aprende, depende de reprogramação manual, o que limita reações naturais a situações inesperadas. |
| Behavior Tree | Alta reatividade, avaliação contínua permite respostas rápidas e coordenadas. | Flexível e hierárquica, permite decisões mais complexas e estruturadas do que a FSM. | Média: modular e flexível, mas sem aprendizado autônomo. |
| Pathfinding | Muito boa para respostas rápidas na movimentação, mas limitada a navegação. | Não realiza tomada de decisão comportamental, apenas cálculo de caminhos. | Baixa: Não se adapta em comportamento, apenas recalcula rotas. |
| IA baseada em Aprendizado | Pode ter latência durante o treinamento, mas se torna extremamente rápida para reagir a situações complexas após aprendê-las. | Excelente: decisões complexas e adaptadas ao ambiente e experiência do jogador | Alta: aprendizado contínuo permite evolução de comportamentos e ajuste dinâmico ao jogador. |
| IA Generativa | Capaz de gerar respostas complexas e criativas, porém com latência dependente do processamento. | Muito boa: gera decisões autônomas e criativas, simulando diálogos, ações e respostas variadas. | Alta: pode gerar conteúdo e comportamentos adaptados. |

| Técnica | Navegação | Simulação de Comportamento | Impacto na Imersão |
|-----------------------------------|---|--|---|
| FSM (Finite State Machine) | Navegação simples e codificada, pouca fluidez e adaptabilidade. | Comportamentos previsíveis, repetitivos, pouco naturais e sem variação contextual. | Imersão limitada, pois NPCs tendem a parecer mecânicos, previsíveis e pouco realistas, afetando a sensação de mundo vivo. |
| Behavior Tree | Gerencia a navegação de forma reativa, combinando o ato de seguir caminhos com outras ações e decisões. | Simula rotinas e comportamentos variados, mas ainda depende de programação manual. | Boa imersão devido a comportamentos variados e reativos, contribuindo para NPCs com ações mais críveis e menos previsíveis. |
| Pathfinding | Excelente, foco principal, desvia obstáculos e planeja rotas eficientes. | Muito fraca para simulação comportamental, apenas movimentação. | Melhora a imersão espacial com movimentação fluida, mas sozinha não cria profundidade comportamental nos NPCs. |
| IA baseada em Aprendizado | Pode aprender rotas e se adaptar a mapas dinâmicos, otimizando navegação. | Excelente: Reações naturais e imprevisíveis, resultando em NPCs mais críveis e expressivos | Alta imersão devido à evolução comportamental, personalização de desafios e interação realista com o jogador. |
| IA Generativa | Pode gerar objetivos e destinos dinâmicos, influenciando a navegação de forma imprevisível e criando jornadas não roteirizadas. | Excelente: simula emoções, rotinas e interações ricas, proporcionando personagens com personalidade e profundidade social. | Muito alta imersão por criar mundos e NPCs dinâmicos, variados e com comportamento social complexo, aproximando-se do realismo. |

Figura 1: Comparação das principais características da inteligência artificial em jogos digitais segundo diferentes técnicas utilizadas, evidenciando como cada característica se manifesta em cada técnica e seu impacto na imersão do jogador.

3 Metodologia

3.1 Tipo de Pesquisa

O presente estudo caracteriza-se como uma pesquisa aplicada, uma vez que tem como objetivo utilizar conhecimentos teóricos para o desenvolvimento e análise prática de um jogo digital com diferentes abordagens de Inteligência Artificial. O foco está na investigação de como técnicas específicas, Máquina de Estados Finitos (FSM) e Árvore de Comportamento (Behavior Tree), influenciam a imersão e a experiência do jogador, permitindo gerar conhecimento voltado à solução de um problema real relacionado ao design de jogos digitais.

Quanto aos objetivos, trata-se de uma pesquisa exploratória e comparativa. É exploratória por buscar compreender de forma aprofundada a relação entre complexidade da IA e a percepção do jogador, considerando um campo em que ainda há lacunas quanto à mensuração dessa experiência subjetiva. É também comparativa por confrontar duas técnicas de IA, analisando diferenças perceptíveis entre elas no comportamento de NPCs e nos níveis de interação, desafio e engajamento proporcionados ao jogador.

Por fim, quanto à abordagem, adotou-se uma metodologia quantitativa, já que os dados coletados foram analisados estatisticamente por meio de testes de associação entre variáveis categóricas, especialmente utilizando o teste Qui-Quadrado, com o objetivo de identificar se características percebidas pelos jogadores apresentaram influência significativa na escolha da fase preferida.

3.2 Descrição do Jogo e das Implementações

3.2.1 Versão com FSM

O código-fonte completo do jogo Éter está disponível no repositório oficial: GitHub - Jogo Éter. O jogo foi desenvolvido inteiramente em C++ utilizando a biblioteca SDL2 para a criação dos sistemas gráficos, entrada e áudio. E o projeto foi implementado e organizado no ambiente de desenvolvimento CLion.

Na primeira fase do desenvolvimento, foi implementada uma FSM, responsável por controlar os comportamentos básicos do inimigo de forma reativa e determinística.



Figura 2: Fase 1 com implementação da FSM

A FSM implementada para esse inimigo é composta por três estados principais:

- Patrol (Patrulha)
- Chase (Perseguição)
- Attack (Ataque)

Figura 3: Trecho de código demonstrando a estrutura dos estados na implementação da FSM.

Cada estado segue a estrutura convencional de uma FSM, contendo os métodos `Enter()`, `Update()` e `Exit()`, responsáveis por definir o comportamento ao entrar no estado, sua lógica executada a cada quadro e as ações realizadas ao sair do estado.

- Estado Patrol (Patrulha)

O inimigo inicia normalmente no estado Patrol, movimentando-se entre dois limites definidos pela variável `mPatrolDistance`. Quando atinge um dos extremos, o inimigo inverte a direção, criando um padrão clássico de patrulhamento lateral.

- Lógica de ação:

- * Define o ponto inicial da patrulha;
 - * Move-se continuamente para a direita ou esquerda;
 - * Altera a rotação do sprite ao inverter direção;
 - * Aplica força horizontal por meio do `RigidBodyComponent`.

- Transição:

O inimigo começa a perseguir o jogador caso ele esteja dentro de um raio de detecção:

```
if (distancia < 200.0f) {
    enemy->ChangeState(std::make_unique<Chase>());
}
```

Esse comportamento representa a detecção inicial do jogador.

- Estado Chase (Persegução)

No estado **Chase**, o inimigo aproxima-se continuamente do jogador utilizando a função `MoveTowardsPlayer()`, recalculando a direção a cada atualização.

- Lógica de ação:

- * Aproxima-se do jogador utilizando vetores de direção;
- * Mantém a animação de corrida ativa;
- * Verifica a distância até o jogador a cada quadro.

- Transições definidas:

Retorna à patrulha caso o jogador se afaste além do limite de visão:

```
if (distance > loseSightRange) {  
    enemy->ChangeState(std::make_unique<Patrol>());  
}
```

Entra no estado de ataque caso o jogador esteja perto o suficiente:

```
if (distance < attackRange) {  
    enemy->ChangeState(std::make_unique<Attack>());  
}
```

Assim, o estado **Chase** funciona como um estágio intermediário: o inimigo está engajado, mas ainda não está na distância ideal para atacar.

- Estado Attack (Ataque)

No estado **Attack**, o inimigo mantém uma postura agressiva, aproximando-se do jogador enquanto dispara projéteis.

- Lógica de ação:

- * Continua aproximando-se do jogador mesmo durante o ataque;
- * Executa disparos por meio de `ShootAtPlayer()`;
- * Utiliza um temporizador (`mTimer`) para controlar a duração do ataque.

Quando o tempo de ataque se esgota, o inimigo retorna ao estado **Chase** para reavaliar a situação:

```
if (mTimer <= 0.0f) {  
    enemy->ChangeState(std::make_unique<Chase>());  
}
```

Essa lógica evita que o inimigo fique preso no estado de ataque e permite que ele volte a perseguir ou atacar novamente, caso o jogador permaneça próximo.

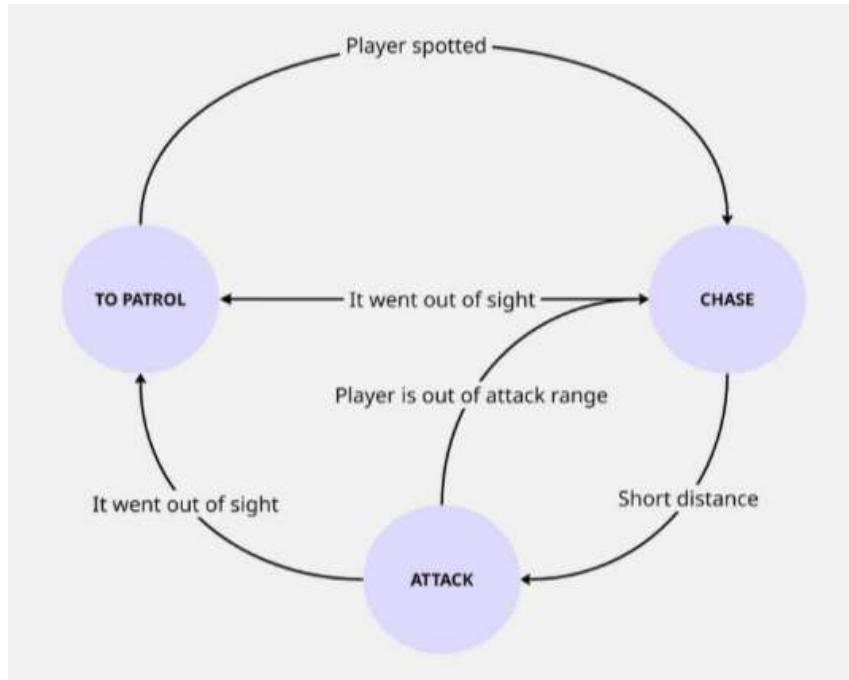


Figura 4: Diagrama de transição de estados da IA utilizada na Fase 1.

3.2.2 Versão com Behavior Tree

Para o inimigo da segunda fase do jogo, foi implementada uma árvore de comportamento (Behavior Tree) com o objetivo de organizar ações complexas de IA de forma modular, reativa e fácil de estender.



Figura 5: Fase 2 com implementação da Behavior Tree

O código a seguir mostra a construção dessa árvore utilizando nós do tipo Selector, Sequence e nós condicionais e de ação específicos do inimigo:

```

std::shared_ptr<BehaviorTree> EnemyABehaviorTree::CreateSimpleFleeBehaviorTree() {
    auto behaviorTree = std::make_shared<BehaviorTree>();

    // Raiz da árvore
    auto root = std::make_shared<SelectorNode>();

    // --- NÓ 1: FUGA QUANDO VIDA ESTÁ MUITO BAIXA ---
    auto lowHealthSequence = std::make_shared<SequenceNode>();
    lowHealthSequence->AddChild(std::make_shared<IsLowHealthNode>(3.0f)); // Vida < 3
    lowHealthSequence->AddChild(std::make_shared<IsPlayerNearNode>(
        detectionRange: 200.0f)); // Player está por perto

    // Sub-sequência de fuga e cura
    auto fleeAndHeal = std::make_shared<SequenceNode>();
    fleeAndHeal->AddChild(std::make_shared<FleeActionNode>(
        fleeSpeed: 300.0f)); // Fuga
    fleeAndHeal->AddChild(std::make_shared<HealOverTimeNode>(
        healRatePerSecond: 0.1f)); // Cura

    lowHealthSequence->AddChild(fleeAndHeal);
    root->AddChild(lowHealthSequence);

    // --- NÓ 2: RECUAR PARA O LÍDER ---
    auto retreatToLeader = std::make_shared<SequenceNode>();
    retreatToLeader->AddChild(std::make_shared<HasLeaderAndIsWeakNode>(
        healthThreshold: 5.0f));
    retreatToLeader->AddChild(std::make_shared<RepositionBehindLeaderNode>(
        followDistance: 50.0f)); // Ação: Mova-se para atrás do líder
    retreatToLeader->AddChild(std::make_shared<AttackActionNode>()); // Continuar atacando

    root->AddChild(retreatToLeader);

    // --- NÓ 3: COMPORTAMENTO NORMAL ---
    auto normalBehavior = std::make_shared<SelectorNode>();

    auto attackBehavior = std::make_shared<SequenceNode>();
    attackBehavior->AddChild(std::make_shared<IsPlayerNearNode>(
        detectionRange: 100.0f));
    attackBehavior->AddChild(std::make_shared<AttackActionNode>());

    normalBehavior->AddChild(attackBehavior);
    normalBehavior->AddChild(std::make_shared<ChaseActionNode>()); // persegue o player
    normalBehavior->AddChild(std::make_shared<PatrolActionNode>(
        patrolDistance: 150.0f)); // ou patrulha

    root->AddChild(normalBehavior);

    behaviorTree->SetRoot(root);
    return behaviorTree;
}

```

Figura 6: Trecho de código demonstrando a estrutura da Behavior Tree implementada na fase 2.

- Estrutura Geral da Árvore

A árvore é organizada com um nó raiz do tipo SelectorNode, que avalia seus filhos em ordem de prioridade. Esse padrão permite que comportamentos emergenciais (como fuga) tenham precedência sobre comportamentos normais como ataque ou patrulha.

A árvore possui três grandes blocos:

- Comportamento de fuga (quando a vida está muito baixa)
- Recuo estratégico para o líder
- Comportamento padrão (ataque, perseguição ou patrulha)

- Comportamento de Fuga por Baixa Vida

O primeiro filho da raiz é uma Sequence, que só é executada quando todas suas condições são verdadeiras:

- IsLowHealthNode: verifica se o inimigo está com menos de 3 pontos de vida;
- IsPlayerNearNode: confirma se o jogador está próximo;
- Caso ambas as condições sejam verdadeiras, executa a sequência fleeAndHeal, composta por:
 - * FleeActionNode: inicia uma fuga em direção contrária ao jogador;
 - * HealOverTimeNode: aplica regeneração gradual enquanto foge.

Esse bloco garante que o inimigo priorize a sobrevivência, fugindo quando está em situação crítica.

- Recuo Estratégico para o Líder

O líder do grupo é tratado como um inimigo de categoria superior, funcionando como uma unidade elite. Esse líder possui mais pontos de vida que os demais inimigos. O segundo bloco é outra Sequence, verificando se o inimigo tem um líder próximo e se está enfraquecido:

- HasLeaderAndIsWeakNode: verifica se existe um líder no grupo e se a vida está abaixo de um threshold moderado;
- RepositionBehindLeaderNode: move o inimigo para uma posição segura atrás do líder;
- AttackActionNode: ao se reposicionar, o inimigo volta a atacar.

Esse comportamento cria uma dinâmica de grupo, fazendo com que o inimigo demonstre coordenação e proteção usando o líder como escudo.

- Comportamento Normal

Caso nenhum comportamento de prioridade superior seja ativado, o inimigo adota o comportamento padrão, organizado em um Selector:

- Atacar se o jogador estiver próximo:
 - * IsPlayerNearNode
 - * AttackActionNode
- Se o jogador está longe, mas dentro do campo de visão do inimigo, perseguir:
 - * ChaseActionNode
- Se nada mais se aplica, patrulhar a área:
 - * PatrolActionNode

Esse bloco garante que o inimigo permaneça ativo mesmo quando não há ameaças imediatas, alternando entre ataque, perseguição e patrulha.

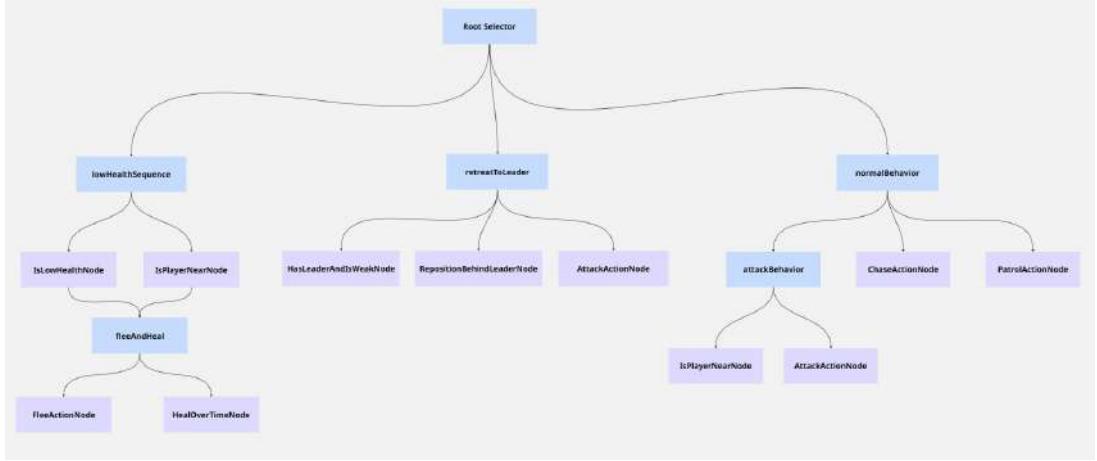


Figura 7: Representação visual da hierarquia de nós da Árvore de Comportamento

3.3 Coleta de Dados e Participantes

3.3.1 Participantes

Participaram do estudo 13 jogadores, com idades entre 16 e 25 anos, todos com experiência prévia em jogos digitais. O questionário foi distribuído a dois grupos principais de respondentes: o primeiro formado por colegas que cursaram a disciplina de Desenvolvimento de Jogos Digitais, que possuem um conhecimento mais técnico sobre aspectos de design e implementação de jogos e o segundo composto por jogadores que, embora não atuem diretamente no desenvolvimento, demonstram interesse e familiaridade com jogos eletrônicos.

Essa combinação permitiu obter perspectivas complementares, conciliando visões técnicas e experiências práticas de jogo, o que enriquece a análise dos resultados.

3.3.2 Instrumento de Coleta (Questionário)

O instrumento de coleta utilizado foi um questionário estruturado, elaborado com o objetivo de identificar quais aspectos influenciaram a preferência dos participantes por uma das fases do jogo. O questionário foi composto por onze perguntas fechadas, formuladas de maneira objetiva e direta, de modo a facilitar o preenchimento e aumentar a taxa de resposta, considerando que questionários muito extensos tendem a gerar desinteresse por parte dos respondentes.

As questões foram organizadas em torno de dimensões relacionadas à experiência do jogador, com foco principal nos comportamentos dos inimigos controlados por IA e em aspectos associados à imersão e à percepção de qualidade da jogabilidade. Entre os eixos avaliados, destacam-se:

- Comportamento dos inimigos, incluindo rapidez de resposta, variedade e imprevisibilidade das ações, e o nível de tomada de decisão percebido;
- Fatores de imersão, como a sensação de envolvimento e engajamento com a fase;
- Aspectos gerais da experiência, como a influência da dificuldade e da duração das fases na preferência do jogador.

| | | | | | |
|---|--|--|--|---|--|
| Análise da Imersão e Dificuldade em Jogo com IAs de Comportamento Variado - 2 Esta é um experimento destinado a implementar e testar diferentes alternativas da probabilidade de influência e experiência de jogador. Você jogará duas fases, uma com cada uma das alternativas de comportamento propostas. Se optar por jogar a segunda fase, pode sair da experiência. Deseja jogar comendo que observará diferenças de desempenho, nível de imersão e o efeito do comportamento genial durante a partida. O seu feedback é essencial, com essa ajuda os diferentes algoritmos do jogo podem melhorar suas qualidades e a sua experiência de jogabilidade. Ao finalizar as duas fases, por favor, preencha o questionário, composta as suas opiniões e percepções que teve em cada uma delas. | | Qual fase você mais gostou? <input type="radio"/> Fase 1 <input type="radio"/> Fase 2 | | A imprevisibilidade das ações dos inimigos (mudanças inesperadas de comportamento) influenciou na sua escolha? <input type="radio"/> Sim <input type="radio"/> Não | |
|  Primeira Fase | | O tamanho/altura das fases influenciou na sua escolha? <input type="radio"/> Sim <input type="radio"/> Não | | O nível de resposta dos inimigos aos seus movimentos ou ataques influenciou na sua preferência? <input type="radio"/> Sim <input type="radio"/> Não | |
|  Segunda Fase | | A forma como os inimigos se comportaram influenciou na sua preferência pela fase? <input type="radio"/> Sim <input type="radio"/> Não | | A forma como os inimigos respondiam às suas ações influenciou na sua preferência por alguma fase? <input type="radio"/> Sim <input type="radio"/> Não | |
| | | A variedade de comportamentos dos inimigos (por exemplo: atacar, fugir, mudar de posição) influenciou na sua escolha? <input type="radio"/> Sim <input type="radio"/> Não | | Você sentiu que a fase escolhida era mais imersiva? <input type="radio"/> Sim <input type="radio"/> Não | |
| | | | | Você sentiu que sua experiência no jogo foi melhor na fase escolhida? <input type="radio"/> Sim <input type="radio"/> Não | |

Figura 8: Questionário aplicado aos participantes contendo as questões utilizadas na coleta de dados.

Para viabilizar a análise estatística, todas as perguntas apresentavam apenas duas opções de resposta: "Sim" ou "Não".

Essa estrutura foi uma escolha metodológica deliberada para gerar dados binários. Esse dados permitiram, primeiramente, uma análise quantitativa direta, como a frequência de respostas "Sim" para cada IA, e, principalmente, possibilitou a aplicação do teste qui-quadrado, utilizado para verificar se havia associação estatisticamente significativa entre as características avaliadas (variáveis independentes) e a preferência do jogador por uma das fases do jogo (variável dependente).

3.3.3 Procedimentos de Aplicação

O processo de coleta de dados foi conduzido inteiramente de forma remota, visando facilitar a participação e garantir um ambiente de teste natural para o jogador (seu próprio computador).

Os participantes selecionados receberam por e-mail um pacote de teste contendo três itens essenciais:

- O link para o repositório do jogo no GitHub.
- Um documento com instruções detalhadas para a configuração do ambiente e execução do jogo.
- O link para o questionário estruturado.

O fluxo experimental foi desenhado de maneira uniforme e sequencial para todos os participantes. O protótipo do jogo foi dividido em duas fases obrigatórias, jogadas em ordem fixa:

- Fase 1: O jogador interagia com inimigos controlados pela implementação de Máquina de Estados Finitos.

- Fase 2: Imediatamente após completar a primeira, o jogador avançava para a segunda fase, onde os mesmos inimigos eram controlados pela implementação de Árvore de Comportamento (Behavior Tree).

Após a conclusão de ambas as fases, os participantes foram instruídos a preencher e enviar o questionário. O questionário continha perguntas comparativas que exigiam que o jogador refletisse sobre as duas experiências para responder.

A duração média de cada sessão de teste, incluindo o tempo de jogabilidade das duas fases e o preenchimento do questionário, foi de aproximadamente 10 minutos, com pequenas variações dependendo da habilidade e ritmo de cada jogador.

3.4 Tratamento e Análise dos Dados

3.4.1 Teste qui-quadrado

Para analisar a relação entre as características percebidas pelos jogadores durante as fases do jogo e a escolha final da fase preferida, foi utilizado o teste Qui-Quadrado (χ^2). Esse teste estatístico é amplamente empregado quando se deseja verificar se duas variáveis categóricas apresentam associação significativa entre si, comparando suas frequências observadas com as frequências que seriam esperadas caso não houvesse relação entre elas.

Dessa forma, o teste Qui-Quadrado sempre avalia a relação entre duas variáveis por vez. Ele não permite analisar simultaneamente vários fatores, como ocorre em modelos multivariados. Por isso, no contexto deste trabalho, cada característica do jogo foi testada individualmente em relação à variável fase escolhida. Assim, foram criadas diversas análises em pares, por exemplo:

- comportamento dos inimigos \times fase escolhida;
- rapidez dos inimigos \times fase escolhida;
- variedade dos inimigos \times fase escolhida;
- imprevisibilidade \times fase escolhida;
- imersão \times fase escolhida.

Dessa forma, é possível verificar separadamente se cada característica apresenta associação estatisticamente significativa com a escolha da fase preferida pelo jogador.

Ao final de cada teste, foi obtido um valor-p (p-value). Esse valor indica a probabilidade de observarmos aquela diferença entre as frequências caso não exista associação entre as variáveis. Neste trabalho, adotou-se o critério amplamente utilizado na literatura científica de $\alpha = 0,05$ como nível de significância. Assim:

- se $p < 0,05$, considera-se que existe associação significativa entre as variáveis;
- se $p \geq 0,05$, conclui-se que não há evidências de associação.

Essa interpretação permitiu identificar quais características dos inimigos e das fases tiveram impacto estatisticamente relevante na escolha da fase preferida pelos jogadores.

3.4.2 Procedimentos de Implementação

Para realizar as análises estatísticas deste estudo, foi utilizado o software Jamovi, uma ferramenta de análise de dados que permite executar testes estatísticos de forma intuitiva e visual. Inicialmente, todos os dados coletados por meio do questionário foram inseridos manualmente na interface de planilha do programa, onde cada linha representa um participante e cada coluna corresponde a uma das variáveis avaliadas no estudo, como comportamento dos inimigos, rapidez, imersão, entre outras, além da variável dependente fase escolhida.

Após a organização da base de dados, utilizamos o módulo "Análises → Tabelas de Contingência", que permite investigar a relação entre duas variáveis categóricas. Para cada variável independente do questionário foi realizada uma análise separada, sempre cruzando essa variável com a variável dependente Fase (Fase 1 ou Fase 2).

O Jamovi gera automaticamente uma tabela de contingência, que apresenta:

- Frequências observadas: quantas respostas "Sim" e "Não" ocorreram em cada fase.
- Porcentagens em linha: mostram a proporção de respostas dentro de cada linha da tabela, facilitando a comparação entre as categorias.
- Totais gerais: quantos respondentes indicaram cada opção no total.

Além da tabela, o software também retorna os valores do teste de associação Qui-Quadrado (χ^2), composto por:

- Valor de χ^2 : mede o quanto as respostas observadas diferem daquilo que seria esperado caso não existisse relação nenhuma entre as variáveis.
 - Se as frequências observadas forem muito diferentes das esperadas, o valor de χ^2 aumenta.
 - Quanto maior o χ^2 , maior a evidência de que existe alguma associação entre as variáveis.
- Graus de Liberdade (gl) : Os graus de liberdade dependem do número de categorias das variáveis comparadas e representam quantas células da tabela podem variar livremente sem quebrar sua estrutura. No caso de tabelas 2×2 , como a maior parte das analisadas neste estudo, o gl é igual a 1. Ele é utilizado apenas para determinar se o valor de χ^2 é grande o bastante para indicar uma associação real.
- p-valor: indica a probabilidade da diferença observada ocorrer apenas por acaso, considerando a hipótese de que não há relação entre as variáveis.
 - $p \geq 0.05 \rightarrow$ não há evidência suficiente para afirmar que existe associação, as diferenças podem ser aleatórias.
 - $p < 0.05 \rightarrow$ a associação é considerada estatisticamente significativa, sugerindo que a variável independente influenciou a preferência pela fase.

O Jamovi também exibe, nessa mesma análise, gráficos de barras empilhadas, que mostram visualmente a distribuição das respostas entre as fases. Esses gráficos apresentam

a porcentagem do total para cada categoria da variável independente, o que facilita a interpretação dos padrões de escolha dos participantes.

Esse procedimento foi repetido para cada variável independente, de forma pareada: comportamento dos inimigos \times fase, rapidez dos inimigos \times fase, imersão \times fase. Para cada par, obtivemos uma tabela de contingência, o resultado do teste Qui-Quadrado e o respectivo gráfico. Dessa forma, foi possível verificar se cada característica influenciou a escolha da fase pelos jogadores e, adicionalmente, realizar uma análise descritiva comparando visualmente os valores observados entre as categorias.

4 Resultados

4.1 Análise Descritiva dos Dados

A análise descritiva dos dados permite observar padrões importantes nas percepções dos participantes em relação às duas fases do jogo. Os resultados mostram uma tendência clara: os jogadores atribuíram maior relevância às características comportamentais dos inimigos e associaram essas qualidades predominantemente à Fase 2, que utilizou a IA baseada em Behavior Tree. Elementos como comportamento, diversidade, imprevisibilidade, responsividade e tomada de decisão foram marcados como influentes por quase todos os participantes ao avaliar essa fase. Em contraste, essas mesmas características foram pouco ou quase nada mencionadas na Fase 1 (FSM), indicando uma percepção significativamente inferior.

A análise visual do gráfico reforça a superioridade da implementação com Behavior Tree (Fase 2), evidenciando uma disparidade clara na percepção dos jogadores. Enquanto atributos técnicos cruciais como diversidade, tomada de decisão e responsividade apresentam aprovação quase unânime na Fase 2 ($n = 10$ a 11), na Fase 1 (FSM) esses elementos foram majoritariamente classificados como ausentes (barras vermelhas) ou de baixa incidência. Essa predominância técnica reflete-se diretamente nos aspectos subjetivos: a imersão e a experiência geral seguiram o mesmo padrão de alta aprovação para a Fase 2, confirmando que a maior complexidade comportamental dos NPCs foi determinante para elevar o engajamento e a qualidade percebida do jogo.

Outro ponto relevante é que, além de perceberem a Fase 2 como mais rica em atributos, a maioria dos participantes também a escolheu como a fase preferida. Isso indica uma conexão direta entre a performance da IA e a satisfação do jogador. A diversidade e imprevisibilidade dos inimigos, proporcionadas pela arquitetura de Behavior Trees, parecem ter contribuído significativamente para tornar a jogabilidade mais dinâmica, desafiadora e interessante, aspectos que se traduziram em preferência e maior engajamento.

Assim, a análise dos resultados sustenta a hipótese inicial do estudo. A implementação de uma IA mais complexa, capaz de gerar comportamentos variados, responsivos e menos previsíveis, contribuiu de forma evidente para melhorar a imersão, a percepção de qualidade e a experiência geral dos participantes. Esses achados reforçam a importância de sistemas de IA sofisticados no design de jogos, demonstrando que, além de influenciar o comportamento dos inimigos, eles impactam a percepção subjetiva do jogador e sua satisfação com a fase jogada.

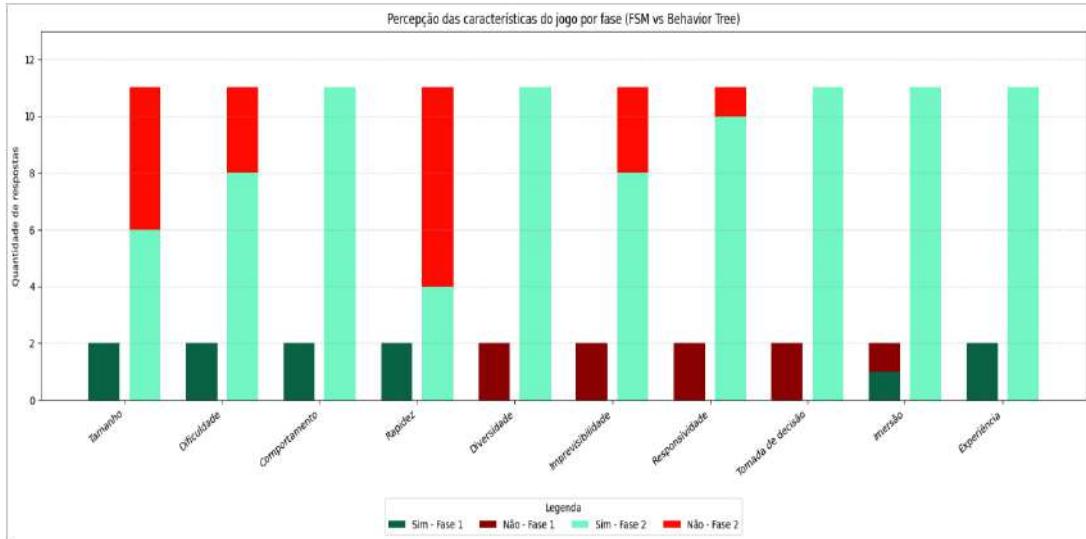


Figura 9: Distribuição das respostas dos participantes para cada característica avaliada nas duas fases do jogo (FSM e Behavior Tree)

Por fim, os resultados obtidos confirmam que a complexidade da IA desempenha um papel significativo e direto na construção de uma experiência de jogo mais rica. A Fase 2, com sua arquitetura comportamental mais avançada, foi percebida como mais envolvente em praticamente todos os aspectos avaliados, destacando-se não apenas na dimensão técnica, mas também afetiva e cognitiva. Dessa forma, fazendo esta análise podemos perceber que a adoção de sistemas de IA mais robustos tende a elevar a qualidade da experiência interativa, fortalecendo atributos como imersão, variedade de comportamentos inimigos e satisfação geral do jogador.

4.2 Análise Inferencial: Teste Qui-Quadrado

A análise estatística foi realizada utilizando o teste Qui-Quadrado (χ^2), adotando-se um nível de significância de $\alpha = 0,05$. O objetivo foi verificar se existe uma associação estatisticamente significativa entre a fase escolhida pelo jogador (Fase 1 - FSM ou Fase 2 - Behavior Tree) e os atributos percebidos durante a jogabilidade. Os resultados foram agrupados em três categorias: fatores não determinantes, fatores comportamentais da IA e a percepção de imersão.

- Fatores Não Determinantes (Controle Experimental)

Inicialmente, avaliou-se se aspectos estruturais do jogo, como o tamanho da fase e a dificuldade, influenciaram a preferência dos jogadores.

- Tamanho da fase: O teste apresentou $\chi^2 = 1,48$ e $p = 0,224$.
- Dificuldade: Obteve-se $\chi^2 = 0,709$ e $p = 0,400$.

Como ambos os valores de p são superiores a 0,05, não se rejeita a hipótese nula. Isso indica que não houve associação significativa entre a escolha da fase e essas variáveis. Esse resultado é positivo para a validade do experimento, pois sugere que a preferência dos jogadores pela Fase 2 não foi motivada por ela ser "maior" ou

”mais difícil”, isolando assim a influência da Inteligência Artificial como o provável fator diferenciador.

A variável Rapidez dos inimigos também não apresentou significância estatística ($\chi^2 = 2,76; p = 0,097$), embora o valor de p sugira uma leve tendência visual não confirmada estatisticamente.

| Tabelas de Contingência | | | | Tabelas de Contingência | | | | Tabelas de Contingência | | | | Tabelas de Contingência | | | | | | | |
|-------------------------|--------------------------------|--------|--------|-------------------------|---------------------------------|----------|--------|-------------------------|-----------------------------|--------|----------|-------------------------|--------------------------|--------|--------|----------|------|---|--------|
| Fase | Tamanho da fase | | | Fase | Rapidez dos inimigos | | | Fase | Dificuldade | | | Fase | Diversidade dos inimigos | | | | | | |
| | Sim | Não | Total | | Não | Sim | Total | | Sim | Não | Total | | Sim | Não | Total | | | | |
| Fase 2: Observado | 6 | 5 | 11 | Fase 2: Observado | 4 | 7 | 11 | Fase 2: Observado | 8 | 3 | 11 | Fase 2: Observado | 11 | 0 | 11 | | | | |
| % em linha | 54,5% | 45,5% | 100,0% | % em linha | 36,4% | 63,6% | 100,0% | % em linha | 72,7% | 27,3% | 100,0% | % em linha | 100,0% | 0,0% | 100,0% | | | | |
| Fase 1: Observado | 2 | 0 | 2 | Fase 1: Observado | 2 | 0 | 2 | Fase 1: Observado | 2 | 0 | 2 | Fase 1: Observado | 0 | 2 | 2 | | | | |
| % em linha | 100,0% | 0,0% | 100,0% | % em linha | 100,0% | 0,0% | 100,0% | % em linha | 100,0% | 0,0% | 100,0% | % em linha | 0,0% | 100,0% | 100,0% | | | | |
| Total: Observado | 8 | 5 | 13 | Total: Observado | 6 | 7 | 13 | Total: Observado | 10 | 3 | 13 | Total: Observado | 11 | 2 | 13 | | | | |
| % em linha | 61,5% | 38,5% | 100,0% | % em linha | 46,2% | 53,8% | 100,0% | % em linha | 76,9% | 23,1% | 100,0% | % em linha | 84,6% | 15,4% | 100,0% | | | | |
| Testes χ^2 | | | | Testes χ^2 | | | | Testes χ^2 | | | | Testes χ^2 | | | | | | | |
| | Valor | gl | p | | Valor | gl | p | | Valor | gl | p | | Valor | gl | p | | | | |
| | χ^2 | 1,48 | 1 | 0,224 | | χ^2 | 2,76 | 1 | 0,097 | | χ^2 | 0,709 | 1 | 0,400 | | χ^2 | 13,0 | 1 | < .001 |
| | N | 13 | | | N | 13 | | | N | 13 | | | N | 13 | | | | | |
| Tabelas de Contingência | | | | Tabelas de Contingência | | | | Tabelas de Contingência | | | | Tabelas de Contingência | | | | | | | |
| Fase | Imprevisibilidade dos inimigos | | | Fase | Tomada de decisões dos inimigos | | | Fase | Responsividade dos inimigos | | | Fase | Imersão na fase | | | | | | |
| | Sim | Não | Total | | Sim | Não | Total | | Sim | Não | Total | | Sim | Não | Total | | | | |
| Fase 2: Observado | 8 | 0 | 8 | Fase 2: Observado | 11 | 0 | 11 | Fase 2: Observado | 10 | 1 | 11 | Fase 2: Observado | 11 | 0 | 11 | | | | |
| % em linha | 72,7% | 27,3% | 100,0% | % em linha | 100,0% | 0,0% | 100,0% | % em linha | 90,9% | 9,1% | 100,0% | % em linha | 100,0% | 0,0% | 100,0% | | | | |
| Fase 1: Observado | 0 | 2 | 2 | Fase 1: Observado | 0 | 2 | 2 | Fase 1: Observado | 0 | 2 | 2 | Fase 1: Observado | 1 | 1 | 2 | | | | |
| % em linha | 0,0% | 100,0% | 100,0% | % em linha | 0,0% | 100,0% | 100,0% | % em linha | 0,0% | 100,0% | 100,0% | % em linha | 50,0% | 50,0% | 100,0% | | | | |
| Total: Observado | 8 | 5 | 13 | Total: Observado | 11 | 2 | 13 | Total: Observado | 10 | 3 | 13 | Total: Observado | 12 | 1 | 13 | | | | |
| % em linha | 61,5% | 38,5% | 100,0% | % em linha | 84,6% | 15,4% | 100,0% | % em linha | 76,9% | 23,1% | 100,0% | % em linha | 92,3% | 7,7% | 100,0% | | | | |
| Testes χ^2 | | | | Testes χ^2 | | | | Testes χ^2 | | | | Testes χ^2 | | | | | | | |
| | Valor | gl | p | | Valor | gl | p | | Valor | gl | p | | Valor | gl | p | | | | |
| | χ^2 | 3,78 | 1 | 0,052 | | χ^2 | 13,0 | 1 | < .001 | | χ^2 | 7,88 | 1 | 0,005 | | χ^2 | 5,96 | 1 | 0,015 |
| | N | 13 | | | N | 13 | | | N | 13 | | | N | 13 | | | | | |

Figura 10: Tabelas de Contingência e Resultados do Teste Qui-Quadrado para as Variáveis Comportamentais da IA e variáveis estruturais

| Característica Avaliada | χ^2 | p | Interpretação |
|----------------------------|----------|--------|--|
| Tamanho da fase | 1,48 | 0,224 | Sem associação |
| Dificuldade da fase | 0,709 | 0,4 | Sem associação |
| Comportamento dos inimigos | - | - | - |
| Rapidez dos inimigos | 2,76 | 0,097 | Sem associação (tendência leve visual) |
| Diversidade dos inimigos | 13 | < .001 | Associação Significativa |
| Imprevisibilidade | 3,78 | 0,052 | Tendência de associação |
| Responsividade | 7,88 | 0,005 | Associação Significativa |
| Tomada de decisão | 13 | < .001 | Associação Significativa |
| Imersão | 5,96 | 0,015 | Associação Significativa |
| Experiência geral | - | - | - |

Figura 11: Resultados do teste Qui-Quadrado aplicado às características avaliadas

- Impacto da Complexidade da IA (Fatores Determinantes)

Ao analisar as variáveis diretamente ligadas à lógica dos NPCs, os testes revelaram associações fortes e significativas, confirmando a superioridade perceptível da implementação via Behavior Tree.

- Diversidade dos inimigos: O resultado foi $\chi^2 = 13,0$ com $p < 0,001$. Observou-se que 100% dos participantes que preferiram a Fase 2 apontaram a diversidade como um fator presente, contra 0% na Fase 1. A significância estatística extrema ($p < 0,001$) confirma que a variação de comportamentos proporcionada pela Behavior Tree foi decisiva.
- Tomada de decisões: De forma idêntica, obteve-se $\chi^2 = 13,0$ e $p < 0,001$. A capacidade dos inimigos de ”escolher” ações de forma lógica foi percebida

quase unanimemente na Fase 2, estabelecendo uma clara distinção em relação à rigidez da FSM na Fase 1.

- Responsividade: O teste indicou $\chi^2 = 7,88$ e $p = 0,0057$. Com $p < 0,05$, conclui-se que a capacidade de reação dos inimigos às ações do jogador foi significativamente associada à escolha da Fase
- Imprevisibilidade: esta variável apresentou $\chi^2 = 3,78$ e $p = 0,052$. Embora matematicamente superior a 0,05, o valor está extremamente próximo do nível de corte. Em pesquisas exploratórias, isso pode ser interpretado como uma forte tendência de associação, sugerindo que os jogadores perceberam os inimigos da Fase 2 como mais imprevisíveis, ainda que a amostra pequena ($N = 13$) tenha limitado a confirmação estatística plena.
- Síntese: O Impacto na Imersão

O objetivo final da implementação de uma IA mais complexa era elevar a qualidade da experiência do usuário. A análise da variável Imersão na fase confirmou essa hipótese.

- Imersão: O teste resultou em $\chi^2 = 5,96$ e $p = 0,015$.

Os resultados obtidos permitem concluir que a imersão percebida pelos jogadores está diretamente associada ao nível de complexidade da Inteligência Artificial empregada no jogo. A variável Imersão apresentou significância estatística, indicando que os participantes que vivenciaram a Fase 2, estruturada com Behavior Tree, relataram níveis mais elevados de envolvimento do que aqueles que jogaram a Fase 1, baseada em FSM.

Essa diferença não ocorreu de forma isolada, ela foi sustentada por um conjunto de atributos comportamentais igualmente significativos, como diversidade dos inimigos, tomada de decisões e responsividade. Esses elementos, conforme discutido no MSI I, são fundamentais para promover comportamentos mais ricos, naturais e alinhados às expectativas do jogador em relação à agência dos NPCs. Assim, a imersão emerge não como um componente subjetivo ou desconectado da jogabilidade, mas como a síntese perceptiva de comportamentos mais complexos, adaptativos e menos previsíveis, possibilitados pela técnica de IA empregada.

De forma geral, os resultados descritivos e inferenciais demonstram que a Fase 2, desenvolvida com Behavior Tree, apresentou desempenho amplamente superior em praticamente todos os atributos avaliados. Enquanto fatores estruturais, como tamanho da fase e dificuldade, não influenciaram a preferência dos participantes, os elementos diretamente ligados ao comportamento dos NPCs exibiram associações estatísticas fortes com a escolha da fase, especialmente diversidade, tomada de decisões e responsividade. Esses achados confirmam que a complexidade da arquitetura de IA foi o principal diferencial entre as duas implementações, influenciando diretamente a forma como os jogadores perceberam a dinâmica, a lógica e a qualidade da jogabilidade.

5 Conclusão

O presente trabalho confirma que diferentes níveis de complexidade na Inteligência Artificial exercem influência direta, mensurável e significativa na experiência do jogador. No MSI I, estabeleceu-se a hipótese de que técnicas de IA mais avançadas, como Behavior Trees, seriam capazes de gerar comportamentos mais naturais, variados e responsivos, aumentando o engajamento e a imersão.

Nesta segunda etapa, essa hipótese foi colocada à prova por meio do desenvolvimento de um jogo completo, composto por duas fases linearmente estruturadas, inspiradas na lógica de progressão dos jogos comerciais. Embora distintas em design e ambientação, ambas foram planejadas para oferecer desafios parecidos, permitindo que o foco da análise recaísse sobre o impacto da IA no comportamento dos inimigos e na percepção dos jogadores.

Os resultados obtidos demonstraram que, apesar das diferenças naturais entre as fases, fatores estruturais como tamanho, dificuldade e ritmo da fase não influenciaram significativamente a escolha dos participantes. Isso reforça que a preferência dos jogadores está profundamente relacionada à forma como os NPCs se comportam, validando o objetivo metodológico deste trabalho de isolar a variável “complexidade da IA”.

Assim, confirmou-se que atributos diretamente ligados à IA, como diversidade comportamental, responsividade às ações do jogador, tomada de decisão consistente e imprevisibilidade, foram os elementos que realmente diferenciaram as duas fases na percepção dos participantes. Esses atributos mostraram-se fortemente associados à Fase 2, que utilizou Behavior Trees, enquanto foram pouco percebidos na Fase 1, estruturada com FSM.

Mais que confirmar superioridade técnica, os resultados revelam uma dimensão subjetiva fundamental: a imersão. Essa variável, central para o estudo, apresentou significância estatística, indicando que a Fase 2 não apenas funcionou melhor tecnicamente, mas foi sentida como mais envolvente, aprendida de forma mais fluida e vivenciada como mais satisfatória.

A imersão, portanto, não surge como um reflexo meramente emocional, mas como consequência direta de comportamentos mais críveis, adaptativos e coerentes com o contexto. Os jogadores reconheceram, de forma clara, que a IA da Fase 2 contribuía para uma sensação de presença mais forte dentro do ambiente de jogo, reforçando o papel da IA como elemento constitutivo do design de experiências e não apenas como mecanismo operacional dos inimigos.

Dessa forma, o conjunto das análises permite afirmar que a adoção de técnicas de IA mais sofisticadas não apenas melhora a jogabilidade em termos funcionais, mas transforma a forma como o jogador percebe, interpreta e interage com o ambiente virtual. O trabalho demonstra, de maneira prática e empiricamente sustentada, que IA e imersão são dimensões interdependentes: quanto mais rica, adaptativa e coerente é a lógica que orienta os NPCs, maior é a capacidade do jogo de produzir experiências envolventes, memoráveis e emocionalmente significativas.

Além disso, o protótipo desenvolvido evidencia que abordagens de IA mais complexas podem ser integradas de forma viável ao processo de design, mesmo em projetos acadêmicos e de pequena escala. A metodologia adotada, unir um protótipo jogável, análise estatística e avaliação empírica de usuários, também se mostra uma contribuição relevante para pesquisas futuras na área, oferecendo um modelo replicável e eficiente para estudar fenômenos subjetivos como engajamento, presença e imersão.

Em síntese, este trabalho confirma e amplia as conclusões do MSI I: a IA é um dos

pilares centrais no design de jogos digitais modernos, e sua complexidade tem impacto direto não apenas sobre o comportamento dos NPCs, mas sobre a experiência cognitiva, emocional e interativa do jogador. A Behavior Tree se mostrou uma solução superior para promover realismo e profundidade comportamental, e, consequentemente, maior imersão. Assim, este estudo reforça que investir em IA robusta é investir na própria qualidade da experiência de jogo, contribuindo para a criação de ambientes virtuais mais envolventes, interessantes e significativos.

5.1 Trabalhos Futuros

Como direção para trabalhos futuros, uma possibilidade promissora é expandir o estudo para um número maior de participantes e perfis variados de jogadores. Embora os resultados obtidos tenham sido estatisticamente relevantes, uma amostra mais ampla poderia fortalecer ainda mais as conclusões, permitindo análises mais robustas sobre diferentes estilos de jogo, tempo de experiência com games e preferências individuais.

Além disso, a realização de experimentos controlados em diferentes plataformas, como consoles, dispositivos móveis ou realidade virtual, ampliaria a compreensão sobre como a IA se comporta e é percebida em diferentes contextos de interação.

Outra linha de continuidade envolve a implementação de outras técnicas de inteligência artificial além das utilizadas neste trabalho. Métodos como aprendizado por reforço ou ia generativa poderiam ser explorados e comparados com a Behavior Tree e a FSM, permitindo uma investigação mais profunda sobre como cada abordagem influencia o comportamento dos NPCs e a percepção dos jogadores.

A inclusão de inimigos com aprendizado adaptativo, capazes de modificar seus padrões com base nas ações do jogador, também representaria um avanço significativo, possibilitando uma avaliação mais realista da evolução do comportamento da IA em jogos dinâmicos e complexos.

Por fim, outra vertente relevante consiste em expandir o protótipo desenvolvido, transformando-o em um jogo mais longo, com progressão narrativa, maior variedade de inimigos e cenários mais complexos. Isso permitiria analisar se os impactos observados neste estudo se mantêm em situações de gameplay prolongado e em ambientes com múltiplas camadas de interação.

Também seria possível expandir o projeto incorporando novos mecanismos de observação do comportamento do jogador, permitindo análises mais detalhadas sobre imersão, engajamento e tomada de decisão. Essas abordagens abririam espaço para estudos mais profundos da relação entre IA e experiência do jogador, especialmente em cenários onde o comportamento dos NPCs se adapta de forma mais dinâmica às ações do usuário.

6 Referências Bibliográficas

- [1] TSANAS, Jan Philip Schultz. *AI is coming for video game characters*. Associated Press, 2023. Disponível em: <https://apnews.com/article/ai-artificial-intelligence-video-games-npc-c1327bb9130136d0a5f658f44176c5e7>. Acesso em: 10 abr. 2025.
- [2] EAVIDEN. *The Impact of Artificial Intelligence on Game Development*. 2023. Disponível em: <https://www.eaviden.dk/wp-content/uploads/2023/05/The-impact-of-artificial-intelligence-on-game-development.pdf>. Acesso em: 10 abr. 2025.
- [3] MARTINS, Ernane Rosa. *A inteligência artificial como ferramenta no desenvolvimento de jogos digitais*. Revista Científica Multidisciplinar Núcleo do Conhecimento, 2023. Disponível em: <https://downloads.editoracientifica.com.br/articles/231215230.pdf>. Acesso em: 10 abr. 2025.
- [4] FATIMA, Wajeeha; AMIR, Muhammad; HUSSAIN, Saeed. *The Role of Artificial Intelligence in Video Game Development*. 2023. Disponível em: https://www.researchgate.net/publication/375488989_The_Role_of_Artificial_Intelligence_in_Video_Game_Development. Acesso em: 10 abr. 2025.
- [5] KARPOUZIS, Kostas; TSATIRIS, George. *AI in (and for) Games*. Disponível em: <https://arxiv.org/abs/2105.03123>. Acesso em: 10 abr. 2025.
- [6] YANNAKAKIS, Georgios N.; TOGELIUS, Julian. *Artificial Intelligence and Games*. Springer, 2018. Disponível em: <https://gameaibook.org/book.pdf>. Acesso em: 23 jun. 2025.
- [7] MILLINGTON, Ian; FUNGE, John. *Artificial Intelligence for Games*. 2. ed. CRC Press, 2016. Disponível em: <https://theswissbay.ch/pdf/Gentoomen%20Library/Game%20Development/Programming/Artificial%20Intelligence%20for%20Games.pdf>. Acesso em: 23 jun. 2025.
- [8] REIS, Mariana de Luca; ANDRADE, Kleber de Oliveira. *Áreas de Pesquisa e Técnicas de Inteligência Artificial em Jogos Digitais*. 2021. Disponível em: <https://www.fatec.edu.br/revista/index.php/RTecFatecAM/article/view/281>. Acesso em: 23 jun. 2025.
- [9] MOREIRA, F. *Árvore de Comportamento - IA*. Disponível em: <https://evolvers.com.br/arvore-de-comportamento-ia/>. Acesso em: 17 set. 2023.
- [10] SEKHAVAT, Y. A. *Behavior Trees for Computer Games*, 2017. Disponível em: https://www.researchgate.net/publication/312869797_Behavior_Trees_for_Computer_Games. Acesso em: 23 jun. 2025.