

16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

ROQUEMIND ARCHER DUEL: EXPLORANDO A APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL EM PERSONAGENS NÃO JOGÁVEIS DE JOGOS DIGITAIS

FELIPE ELIAS SOARES¹, JOHNNY EDSON M. DE SOUZA², OSVANDRE A. MARTINS³

¹ Aluno do curso de Bac. em Sistemas de Informação, IFSP, Campus Votuporanga, soares.elias@aluno.ifsp.edu.br.

² Aluno do curso de Bac. em Sistemas de Informação, IFSP, Campus Votuporanga, johnny.miranda@aluno.ifsp.edu.br.

³ Professor do Ensino Básico, Técnico e Tecnológico, IFSP, Campus Votuporanga, osvandre@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 1.03.03.00-6 - Metodologias e Técnicas da Computação.

RESUMO: Os jogos digitais representam uma das principais formas de entretenimento na atualidade e, assim, uma indústria com potencial de crescimento e impacto positivo na economia global. Dentre os diversos estilos, destaca-se o *Roguelike* que envolve a interação entre personagens comandados por humanos e personagens não jogáveis (NPC - *Non-Player Character*) cujo desenvolvimento envolve o desafio de torná-los oponentes mais eficientes e eficazes. Neste contexto, a Inteligência Artificial (IA), especialmente o Aprendizado por Reforço, surge como um campo de solução promissor, capaz de tornar os jogos mais realistas, desafiadores e interessantes. Este documento apresenta os resultados parciais de uma iniciativa em andamento, voltada a demonstrar como a IA pode ser aplicada nesses casos. Para tanto, realizam-se pesquisas bibliográficas, estudos de ferramentas tecnológicas e um estudo de caso de desenvolvimento de um jogo *Roguelike* minimalista que recebeu o nome **Roguemind Archer Duel**. Busca-se, assim, contribuir para o domínio de tal aplicação, favorecendo o desenvolvimento de NPCs mais interativos, responsivos, adaptativos e, portanto, capazes de proporcionar experiências de jogo mais envolventes.

PALAVRAS-CHAVE: jogos digitais; jogos roguelike; inteligência artificial aplicada a jogos; aprendizagem por reforço.

ROQUEMIND ARCHER DUEL: EXPLORING THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN DIGITAL GAMES NON-PLAYER CHARACTERS

ABSTRACT: Digital games represent one of the main forms of entertainment today and, as such, an industry with potential for growth and a positive impact on the global economy. Among the various genres, Roguelike stands out, involving interactions between human-controlled characters and non-playable characters (NPCs). Their development involves the challenge of making them more efficient and effective opponents. In this context, Artificial Intelligence (AI), especially Reinforcement Learning, emerges as a promising field of solutions capable of making games more realistic, challenging, and interesting. This document presents the partial results of an ongoing initiative aimed at demonstrating how AI can be applied in these cases. To this end, we conduct bibliographical research, study technological tools, and develop a case study of a minimalist Roguelike game called **Roguemind Archer Duel**. The aim is to contribute to the mastery of this application, favoring the

development of more interactive, responsive, adaptive NPCs, and therefore, capable of providing more engaging gaming experiences.

KEYWORDS: digital games; roguelike games; artificial intelligence applied to games; reinforcement learning.

INTRODUÇÃO

Os jogos digitais têm se consolidado como um fenômeno global, reunindo aproximadamente 3,32 bilhões de usuários em todo o mundo, representando cerca de 42% da população mundial (Duarte, 2025). Segundo o Sebrae RS (2025), trata-se de um mercado que tem experimentado um crescimento expressivo, consolidando-se como uma das indústrias mais promissoras em nível global.

Dentre os diversos estilos de jogos, os conhecidos como *Roguelike* vêm se destacando por suas mecânicas acessíveis e alto potencial de rejogabilidade (motivação ao jogador repetir partidas) oferecendo experiências desafiadoras e imprevisíveis. Segundo Cerny e Dechterenko (2015) os *Roguelikes* são jogos desafiadores, baseados em turnos, com ambientes aleatórios e exigência de planejamento estratégico constante. Eles apresentam fases geradas aleatoriamente, garantindo que cada partida seja única, além de adotarem a chamada “morte permanente”, em que o jogador, ao ser derrotado, deve reiniciar a aventura a partir do começo. Essa combinação exige estratégias eficientes, adaptação rápida e replanejamento, contribuindo para o desenvolvimento de habilidades como atenção, memória e resolução de problemas. Um exemplo, com gráficos em 3D, é o jogo *The Spell Brigade*, ilustrado na Figura 1. Nele, o jogador precisa explorar um universo enquanto enfrenta uma série de criaturas dispostas a aniquilá-lo. Por outro lado, pode haver seres que o auxiliem em suas missões. Tratam-se de NPCs (*Non Player Characters*) atuando como aliados ou oponentes, interagindo diretamente com o jogador.

No desenvolvimento de um jogo como este, a repetição de padrões de comportamento nos NPCs representa um problema central do gênero, limitando a variedade de interações e, conseqüentemente, comprometendo o engajamento a longo prazo. Nestes casos, a aplicação de Inteligência Artificial (IA) oferece meios de superar essa limitação, tornando o comportamento dos NPCs mais dinâmico e imprevisível.

Figura 1: Captura de tela em instante de partida do jogo *Roguelike* 3D *The Spell Brigade*.



Assim, um NPC pode ser definido como uma entidade do jogo controlada por *software*, cujo papel é assumir funções que um jogador humano não poderia ou não desejaria desempenhar. Para muitos desenvolvedores, o termo "IA de jogos" é sinônimo do código que governa o comportamento dos NPCs, independentemente de sua complexidade, podendo seguir um roteiro simples ou complexo (Yannakakis; Togelius, 2018, p. 97).

Segundo Barros (2007), embora o desenvolvimento de jogos digitais seja visto como uma atividade lucrativa, ela apresenta desafios significativos, se distinguindo do desenvolvimento de *software* tradicional por suas características únicas, como a necessidade de equipes multidisciplinares, incluindo artistas, *designers* e programadores capazes de lidar com tecnologias consideradas complexas como as da IA, definida por Russell e Norvig (2023, p. 1) como “o estudo de agentes que recebem percepções do ambiente e realizam ações”. A ênfase recai sobre a racionalidade: um agente deve escolher, com base nas informações disponíveis, a ação que maximize seu desempenho em direção a metas explícitas. Nessa perspectiva, interessa menos imitar o comportamento humano e mais garantir respostas autônomas, adaptativas e eficientes em contextos variados.

Quando transferimos esses princípios para os jogos eletrônicos, a ênfase na racionalidade e na adaptação em tempo real ganha contornos práticos muito específicos. Segundo Millington (2006), a principal habilidade do programador de IA em jogos está em utilizar técnicas suficientemente simples para simular complexidade perceptível ao jogador, sem comprometer o desempenho ou a taxa de quadros. O autor também destaca que, mesmo em *hardwares* modernos — nos quais a IA pode consumir uma parcela significativa dos recursos de processamento, exigindo a decomposição dos algoritmos mais exigentes em subtarefas distribuídas ao longo de diversos *frames*, para garantir fluidez. Quando essa abordagem é bem aplicada, soluções aparentemente simples tornam-se mecanismos robustos de *gameplay*. Um exemplo pode ser observado no comportamento dos fantasmas do clássico jogo *Pac-Man*, cuja lógica semi-aleatória é suficiente para provocar tensão, surpresa e a sensação de uma estratégia deliberada, sustentando o engajamento do jogador.

Para Millington (2006), a maioria dos personagens digitais ainda utiliza FSMs (*Finite State Machines*), nas quais cada estado representa um comportamento coerente (como patrulhar, perseguir ou atacar). As transições entre esses estados, acionadas por condições do ambiente ou pela memória interna do agente, promovem uma fluidez natural sem comprometer a previsibilidade. A integração dessas FSMs com estruturas como árvores de decisão (*decision trees*) ou árvores de comportamento (*behavior trees*) viabiliza reações eficazes a eventos inesperados, mantendo, ao mesmo tempo, a eficiência computacional. Além disso, o autor evidencia que não basta a IA saber o que fazer; ela também deve determinar como e onde agir. Algoritmos clássicos de navegação, como *A** e *steering behaviours*, são combinados a mapas de influência que atribuem valores estratégicos ao ambiente, como risco ou vantagem. Esse vínculo entre análise tática e movimentação física permite que os inimigos contornem posições, recuem quando perdem vantagem e aproveitem falhas na defesa — tudo isso dentro de um orçamento computacional compatível com a taxa de quadros exigida pelo jogo.

Ampliando esse panorama, Maleki e Zhao (2024) explicam que a IA, hoje, vai além do controle de NPCs, atuando como coautora de conteúdo. A PCG (*Procedural Content Generation* – Geração Procedural de Conteúdo) evoluiu desde abordagens baseadas em ruído fractal e gramáticas formais até técnicas mais avançadas, como o *deep learning* (aprendizado profundo), culminando na aplicação de *large language models* (modelos de linguagem de grande escala) que podem criar níveis, diálogos e missões adaptativas. Essa capacidade não apenas aumenta a rejogabilidade dos jogos, como também contribui para a redução de custos de produção. Desse modo, os jogos eletrônicos consolidam-se como campo privilegiado para a experimentação com IA, ao mesmo tempo, em que essa tecnologia redefine continuamente os limites do *design* interativo.

Diante da expressividade do mercado de jogos digitais, dos desafios associados ao seu desenvolvimento e das potencialidades da IA, acredita-se na relevância promover a obtenção de noções a programadores, sobre como formas de se aplicar a IA neste contexto. Assim, encontra-se em andamento uma iniciativa voltada a descrever uma forma de se aplicar IA na interação jogador-NPC em um jogo *RogueLike*, viabilizando experiências desafiadoras por meio de comportamentos inteligentes e adaptativos. Para tanto, optou-se por realizar um estudo de caso de desenvolvimento de um protótipo de jogo *Roguelike* que integre essas mecânicas e a análise do impacto da IA na experiência do jogador, documentando o processo para futuras contribuições ao campo.

MATERIAIS E MÉTODOS

Os trabalhos vêm se desenvolvendo por meio da aplicação de uma metodologia que combina pesquisa bibliográfica e o emprego de métodos, técnicas e tecnologias para o desenvolvimento de jogos digitais, incluindo recursos de IA, principalmente do campo de Aprendizado por Reforço.

A pesquisa bibliográfica se pauta na reunião de conhecimentos fundamentais sobre metodologias e processos de desenvolvimento de jogos eletrônicos, bem como de técnicas e conceitos essenciais, como aprendizado de máquina, aprendizado por reforço, redes neurais e comportamentos inteligentes adaptativos. Destaca-se a identificação de desafios da aplicação desses conceitos na criação de NPCs eficiente, eficazes e com capacidade de evoluir seu comportamento.

Mediante a obtenção de conhecimentos fundamentais, desenvolve-se um estudo de caso de desenvolvimento de jogo digital, envolvendo a especificação, o desenho, a implementação e testes de um protótipo de jogo *Roguelike*. Para tanto, emprega-se o motor de jogos *Godot* (Godot, 2014).

A especificação do jogo se materializa por meio de um Documento de Desenho de Jogo (*Game Design Document* — GDD). Trata-se de uma descrição estruturada envolvendo informações essenciais do projeto de jogo, como o enredo, as características e os objetivos de cada personagem, identificando aqueles que são NPC. Ademais descrevem-se os requisitos técnicos, restrições, mecânicas e regras. Esta descrição é produzida na fase de pré-produção, após a concepção da ideia, e acompanha todo o ciclo de vida do jogo, passando pelas etapas de produção, pós-produção e pós-lançamento. Mostra-se, portanto, como um documento vivo, atualizado constantemente para garantir alinhamento e coerência entre todos os envolvidos no desenvolvimento (Barros, 2007).

A escolha do motor de jogos *Godot Engine* (Godot, 2014) se deu por este ser *open source* e por sua arquitetura baseada em nós e cenas facilitar a modularidade no desenvolvimento (Bradfield, 2023). A lógica do jogo vem sendo implementada com emprego da *GDScript*, uma linguagem de programação nativa da *Godot*.

Quanto à aplicação de IA, tem-se lançado mão da biblioteca em linguagem de programação *Python Stable-Baselines3* (Raffin *et al.*, 2021). Esta oferece suporte à aplicação de Aprendizado por Reforço e possibilita abstrair detalhes de codificação dos agentes que convivem no ambiente do jogo. No caso, as interações dos agentes com o ambiente do jogo exigem comunicação realizada por meio de funcionalidades do plugin *Godot RL Agents*.

Em suma, a estratégia de implementação consiste em treinar um agente de IA para controlar o comportamento de um NPC adversário dentro do ambiente. Espera-se assim que o NPC, por meio de Aprendizado por Reforço, desenvolva táticas emergentes e adaptativas em resposta às ações do jogador humano, superando a previsibilidade de comportamentos puramente roteirizados.

Por fim, realiza-se a validação da abordagem e dos resultados de desenvolvimento por meio de testes por usuários convidados, coletando *feedbacks* qualitativos sobre a experiência de jogo, incluindo percepção de desafio, imprevisibilidade e “inteligência” do oponente.

RESULTADOS E DISCUSSÃO

A pesquisa bibliográfica realizada viabilizou a obtenção de conhecimentos acerca: **a)** do universo dos jogos digitais; **b)** das características de NPCs e jogos *Roguelike*; **c)** de metodologias aplicáveis ao desenvolvimento de jogos como a *Game Waterfall Process* e *Extreme Game Development* (XGD) citadas por Barros (2007); **d)** da aplicação de IA em jogos digitais; e **e)** de tecnologias comumente empregadas no desenvolvimento desses jogos, como a plataforma *Godot*, a biblioteca *Stable-Baselines3* e o plugin *RL Agents* que oferece suporte para a criação de NPCs e outros agentes virtuais capazes de se mostrar adaptativos e realistas.

Os conhecimentos obtidos se mostraram úteis à estruturação do desenvolvimento do estudo de caso pretendido, estabelecendo um ciclo de vida que envolve as fases de: **Concepção** (definição da ideia do jogo e análise de viabilidade); **Pré-Produção** (planejamento e produção do GDD); **Produção** (desenhos e codificações de implementação); **Pós-Produção** (testes e preparação para lançamento); e **Pós-Lançamento** (suporte a usuários e possíveis atualizações).

Como resultado de **Concepção**, cita-se a ideia de um jogo em cenário medieval onde dois arqueiros, um humano e um NPC, se encontram no campo de batalha e lutam pela sobrevivência usando arcos e flechas. A cada morte, os personagens ressurgem e espera-se, no caso do NPC, que este demonstre evolução em suas habilidades e estratégias de luta. De posse dessa definição, deu-se início à fase de **Pré-Produção**, intitulando o jogo e produzindo o GDD que envolve: uma visão geral; uma descrição do *Gameplay* (objetivo, mecânicas incluindo IA adaptativa no NPC, regras básicas e condições de vitória e derrota; a história e narrativa do jogo; uma descrição dos personagens com suas habilidades, formas de controle e armas; detalhes da arte e estilo visual; definições de áudios (trilha sonora e efeitos); definições para interface de usuário; definições de níveis e fases; e tecnologias e ferramentas para implementação.

Com uma versão inicial do GDD, deu-se início aos trabalhos de **Produção**, implementando cenários, personagens e alguns comportamentos por meio da plataforma *Godot*. Códigos em *GDScript* foram produzidos para implementar a estrutura hierárquica de cenas e nós, bem como algumas mecânicas elementares de jogabilidade. A Figura 2 apresenta um mosaico de imagens que ilustram versões iniciais de cenas implementadas em 2D, para o jogo **RogueMind: Archer Duel**. A imagem **A** ilustra o menu inicial e a imagem **B** ilustra a cena de execução de partida onde o jogador humano e o NPC duelam.

Quanto à aplicação de IA em NPC, nesta versão inicial, a integração da *Stable-Baselines3* com a *Godot* foi implementada por meio de comunicação local, estabelecida via porta configurada para troca de dados. Nessa arquitetura, os parâmetros de treinamento são transmitidos diretamente para o ambiente do jogo, a partir de um *script* em Python. Na *Godot Engine* o controle do agente é delegado a um nó do tipo *AIController*, responsável por interpretar suas ações. A sincronização entre episódios e recompensas é conduzida por um nó do tipo *sync*, garantindo consistência entre o ambiente simulado e o processo de aprendizagem por reforço.

Figura 2: Capturas de cenas da versão inicial do RogueMind: Archer Duel



Os resultados apresentados representam avanços até porções iniciais da fase de **Produção** do jogo referente ao estudo de caso. Os próximos passos se referem a concluir as implementações, seguidas de testes, análises e documentação.

CONCLUSÕES

Apresentaram-se os resultados parciais de um trabalho em andamento que visa trazer luz a pelo menos uma das formas de se aplicar IA no desenvolvimento de jogos digitais. Destaca-se, neste sentido, a obtenção de conhecimentos fundamentais sobre o desenvolvimento de jogos digitais, bem como o domínio inicial de ferramentas empregadas neste sentido e que envolvem recursos de aplicação de IA para criar NPCs adaptativos em jogos do gênero Roguelike. Acredita-se, portanto, estar trilhando caminho promissor rumo ao alcance dos objetivos estabelecidos, restando a conclusão das implementações do estudo de caso, a aplicação de testes com usuários, a coleta de dados desses testes, a realização de análises e a documentação dos resultados.

CONTRIBUIÇÕES DOS AUTORES

A.B.C se encarregou do aspectos teóricos enquanto D.E.F. se dedicou à exploração das tecnologias e implementações. G.H.I. contribuiu com a orientação acadêmica, técnica e científica do trabalho. Todos os autores contribuíram com a revisão do trabalho e aprovaram a versão submetida.

AGRADECIMENTOS

A Ricardo S. Casado, pela coorientação deste trabalho quanto a Aprendizado por Reforço em IA.

REFERÊNCIAS

- BARROS, R. Análise de metodologias de desenvolvimento de software aplicadas ao desenvolvimento de jogos eletrônicos. 2007. Disponível em: <<https://www.cin.ufpe.br/~tg/2007-1/rlbb.pdf>>.
- BRADFIELD, C. **Godot 4: Game Development Projects**. [S.l.: s.n.], 2023. (Second Edition).
- CERNY, V.; DECHTERENKO, F. Rogue-like games as a playground for artificial intelligence – evolutionary approach. In: CHORIANOPOULOS, K. *et al.* (Ed.). **Entertainment Computing - ICEC 2015**. Springer International Publishing, 2015. v. 9353, p. 261–271. ISBN 978-3-319-24588-1 978-3-319-24589-8. Disponível em: <http://link.springer.com/10.1007/978-3-319-24589-8_20>.
- DUARTE, F. Exploding topics. **Exploding Topics**, 2025. Disponível em: <<https://explodingtopics.com/blog/number-of-gamers>>. Acesso em: 12 abr 2025.
- GODOT, e. **Godot engine**. 2014. Disponível em: <<https://docs.godotengine.org/en/stable/>>.
- MALEKI, M. F.; ZHAO, R. Procedural content generation in games: A survey with insights on emerging LLM integration. v. 20, n. 1, p. 167–178, 2024. ISSN 2334-0924, 2326-909X. Disponível em: <<http://arxiv.org/abs/2410.15644>>.
- MILLINGTON, I. **Artificial intelligence for games**. [S.l.]: Elsevier, 2006. (The Morgan Kaufmann series in interactive 3D technology). OCLC: ocm71264730. ISBN 978-0-12-497782-2 978-0-12-373661-1.
- RAffIN, A. *et al.* Stable-baselines3: Reliable reinforcement learning implementations. 2021.
- RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. Fourth edition, global edition. [S.l.]: Pearson, 2023. (Prentice Hall series in artificial intelligence). ISBN 978-1-292-40113-3 978-1-292-40117-1.
- Sebrae RS. **Tendências do mercado de games para 2025**. 2025. Disponível em: <<https://digital.sebraers.com.br/blog/mercado/tendencias-do-mercado-de-games-para-2025/>>.
- YANNAKAKIS, G. N.; TOGELIUS, J. **Artificial Intelligence and Games**. [S.l.]: Springer, 2018. <<https://gameaibook.org>>.