



**FUNDAÇÃO EDUCACIONAL DE FERNANDÓPOLIS
FACULDADES INTEGRADAS DE FERNANDÓPOLIS
CURSO DE SISTEMAS DE INFORMAÇÃO**

USO DE APRENDIZADO DE MÁQUINA EM JOGOS ELETRÔNICOS

**FERNANDÓPOLIS – SP
2022**

**FÁBIO ORTIZ GONÇALVES
LUCAS PAVÃO PAIVA**

USO DE APRENDIZADO DE MÁQUINA EM JOGOS ELETRÔNICOS

Trabalho de conclusão apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de informação das Faculdades Integradas de Fernandópolis - FIFE.

Orientador (a): Welington Luis Codinhoto Garcia

USO DE APRENDIZADO DE MÁQUINA EM JOGOS ELETRÔNICOS

Fábio Ortiz Gonçalves¹

Lucas Pavão Paiva²

Wellington Luis Codinhoto Garcia³

GONCALVES, F. O.; PAIVA, L. P.; GARCIA, W. L. C; Uso de aprendizado de máquina em jogos eletrônicos. Fernandópolis, 2022.

RESUMO: O mercado de jogos eletrônicos é um ambiente competitivo que está em constante crescimento e evolução, nesse âmbito, empresas buscam formas para se destacar em relação aos seus concorrentes, e a criação de ambientes cativantes e que se destoe dos demais é um dos pontos que compõem os fatores que trazem interesse para um jogo, dentro de tais ambientes, existem seres virtuais conhecidos como NPC'S (personagens não jogáveis). Este artigo mostra a utilização de tecnologias de inteligência artificial e aprendizado de máquina para a criação de comportamentos, que foram atribuídos a esses seres através de treinamentos que utilizam um *framework* de aprendizagem de máquina conhecido como *ML-Agents* para o motor de desenvolvimento *Unity Engine*. O processo foi dado através da criação e treinamento de um agente, tal qual do ambiente de treinamento necessário para que os processos de aprendizado fossem realizados e testados. Após o término da fase de treinamento, o modelo cerebral gerado foi aplicado em uma demonstração interativa, com o intuito de confirmar a capacidade de utilização de tais tecnologias durante o desenvolvimento de jogos eletrônicos.

PALAVRAS CHAVES: Inteligência Artificial. Aprendizado de máquina. Unity Engine. Jogos eletrônicos. Árvore de comportamento. Personagens não jogáveis.

USE OF MACHINE LEARNING IN ELECTRONIC GAMES

ABSTRACT: The electronic games market is a competitive environment that is constantly growing and evolving, where companies seek ways to stand out from their competitors, where the creation of captivating environments that stand out from the others is one of the points that make up the factors that bring interest to a game. To develop these environments, there are virtual beings known as NPC'S (non-playable characters). This article shows the use of artificial intelligence technologies to create behaviors, which were assigned to these beings through training using a machine learning framework known as *ML-Agents* for the *Unity Engine* development engine. The process was given through the creation and training of an agent, as was the training environment necessary for the learning processes to be performed and tested. After the training phase was completed, the generated brain model was applied in an interactive demonstration in order to confirm the ability to use such technologies during the development of electronic games.

KEYWORDS: Artificial Intelligence. Machine learning. Unity Engine. Electronic games. Behavior tree. Non-playable characters.

¹ Estudante de graduação em Sistemas de informação das Faculdades Integradas de Fernandópolis – FIFE

² Estudante de graduação em Sistemas de informação das Faculdades Integradas de Fernandópolis – FIFE

³ Mestre em Engenharia Elétrica. Professor das Faculdades Integradas de Fernandópolis – FIFE *Endereço para correspondência: Rua Teotônio Vilela, 1.685 – Campus Universitário, Fernandópolis – SP, 15.608-380 Tel.: (17)3465 0000 E-mail: welington.unesp@gmail.com

INTRODUÇÃO

O mercado de jogos eletrônicos ganhou mais popularidade e torna-se cada vez mais tecnológico e inovador ao longo dos anos. A indústria dos *games* já é avaliada em US\$ 163,1 bilhões, de modo a se transformar no maior setor da indústria de entretenimento, ultrapassando demais áreas como música e cinema (TECHNET IMMERSIVE, 2021).

Outra informação levantada pela *TECHNET IMMERSIVE* (2021) é que há 2,5 bilhões de jogadores em todo mundo, esta apuração demonstra a importância e a viabilidade de projetos voltados para a área. Diante de tais números mostra-se a existência de uma necessidade de criar produtos de alta qualidade para que se destaquem em um cenário que é altamente competitivo.

São vários os fatores que determinam o sucesso de um jogo eletrônico, dentre eles a jogabilidade, acessibilidade, desempenho e demais pontos que envolvem a busca pela entrega de entretenimento ao consumidor final.

Um dos aspectos que trazem interesse ao consumidor é a fidelidade e o desafio que os personagens não jogáveis podem proporcionar através da inteligência artificial dos jogos (*Game AI*), de modo que esses seres que compõem o mundo virtual sejam cada vez mais fiéis e realistas, trazendo maior imersão, dinâmica e realismo na construção de mundo de um jogo.

Atualmente, a inteligência artificial dos jogos é desenvolvida por meio de um modelo conhecido como árvores de comportamento. Esse que também possui alta usabilidade dentro da área da robótica.

Agis, Gottifredi e García (2020) descrevem as árvores de comportamento como um método de criação de comportamento que é popular na indústria dos jogos, que facilitam a concepção e a implementação de seres que precisam de ser coordenados entre si.

O uso de algoritmos de Aprendizado de máquina não é muito difundido dentro do desenvolvimento das *Game AIs*, porém, hoje existem ferramentas que facilitam o uso desses algoritmos para realizar o treinamento dessas inteligências.

Mitchell (1997) define que o campo da aprendizagem de máquinas está preocupado com a questão de como construir programas de computador que melhoram automaticamente com a experiência.

A *Unity Engine* é um dos maiores motores de desenvolvimento de jogos e possui planos gratuitos de desenvolvimento. Dentro da área abordada, o *Unity* possui um *framework*⁴ conhecido como *Unity ML-Agents*, que torna mais simples e fácil o processo de treinamento das *Game AIs*.

De acordo com a *Unity* (2020) o *ML-Agents* pode ser definido como um projeto de código aberto, que permite jogos e simulações para servir como ambientes de formação de agentes inteligentes usando aprendizagem de reforço profundo e imitação de aprendizagem.

Este artigo tem o objetivo de expor a possibilidade de utilização algoritmos de aprendizado de máquina dentro de jogos eletrônicos, para a criação de personagens não jogáveis cada vez mais inteligentes e convincentes.

MATERIAL E MÉTODO

Para o desenvolvimento do artigo de pesquisa, o método de trabalho utilizado foi dado por meio do treinamento de uma *Game AI* e da criação de uma demonstração interativa em formato de jogo eletrônico, utilizando o motor de desenvolvimento *Unity Engine*.

O desenvolvimento dos *scripts* dentro da *Unity Engine* foi feito utilizando a linguagem de programação conhecida como C#⁵ (*C Sharp*).

A demonstração possui a finalidade de produzir um ambiente controlado e propício para a aplicação de algoritmos de aprendizado de máquina, por meio do *framework ML-Agents* e realizar o treinamento dos personagens não jogáveis concebidos.

Através dos resultados previstos, se tornará possível evidenciar a viabilidade do uso de algoritmos de aprendizado de máquina na construção de inteligências artificiais mais autênticas, críveis e fidedignas.

⁴ *Framework* são bibliotecas e estruturas de códigos que possibilitam o a criação de sistemas e aplicações de maneira mais simples e facilitada.

⁵ O C# é uma linguagem de programação, desenvolvida atualmente pela *Microsoft*.

Inteligência Artificial

Inteligência artificial é uma área de pesquisa e desenvolvimento que ficou em evidência nos últimos anos e é utilizada para definir sistemas e máquinas que possuem comportamentos semelhantes ou similares com ações humanas, com o intuito de auxiliar na resolução de problemas e tarefas.

Kurzweil (1990) retrata a inteligência artificial como “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas”.

Através da ampliação do desenvolvimento e maior atenção para essa área de pesquisa, o ramo da inteligência artificial está sendo difundido em diversos outros meios de atuação.

Bellman (1978) considera a inteligência artificial como “A automação de atividades que nós associamos com o pensamento humano, atividades tais como tomada de decisões, resolução de problemas e aprendizado”.

Jogos Eletrônicos

Os jogos eletrônicos são experiências interativas onde o usuário necessita de periféricos para realizar as ações com o intuito de concluir os seus objetivos. Usando *mouses*, teclados, controles e outros periféricos, os usuários conseguem fazer as ações e comandos, que serão visualizadas por uma televisão ou *monitor*. Todos esses periféricos deverão estar conectados a um computador ou *console*⁶, assim como o jogo, que também será processado através desses dispositivos.

Os *videogames* estão em constante evolução, e a cada ano que se passa, é possível notar o aparecimento de novas tecnologias para inovar a área, tornando-os cada vez mais realistas em relação aos gráficos e inteligência no mundo apresentando.

⁶ Popularmente conhecidos como *videogames*, *consoles* são aparelhos eletrônicos com componentes muito parecidos com computadores, porém são dedicados para a execução de jogos virtuais.

Inteligência artificial em jogos eletrônicos

Ao introduzir o termo de inteligência artificial dentro do âmbito de jogos eletrônicos, cria-se um conflito entre a área de estudo acadêmica já citada anteriormente, e o termo utilizado para definir o conjunto de comportamentos realizados pelas personagens não jogáveis, mesmo que essas não façam a utilização de algoritmos de aprendizado.

Para os desenvolvedores de jogos eletrônicos, as aplicações computacionais de IA e o significado do termo IA são diferentes dos encontrados no meio acadêmico. “Para distinguir a IA utilizada em jogos e no meio acadêmico, os desenvolvedores adotaram o termo *Game AI*” (FUNGE, 2004, apud KISHIMOTO, 2004).

A diferenciação dos termos acadêmico e o termo utilizado dentro do universo dos jogos eletrônicos é de extrema importância, visto que o trabalho desenvolvido demonstra que é possível utilizar ferramentas de aprendizado de máquina para realizar o treinamento das *Game AIs*.

Personagens não jogáveis

Os NPCS (personagens não jogáveis) são peças que compõem o mundo de jogos eletrônicos, e são controlados através da *Game AI*, diferentemente dos personagens controlados pelo jogador.

Khan (2014) ressalta que, o ambiente de um jogo é populado por muitos NPCS, esses que normalmente seguem eventos pré-definidos e não exibem ou realizam tarefas alinhadas de acordo com o que está acontecendo dentro do ambiente virtual.

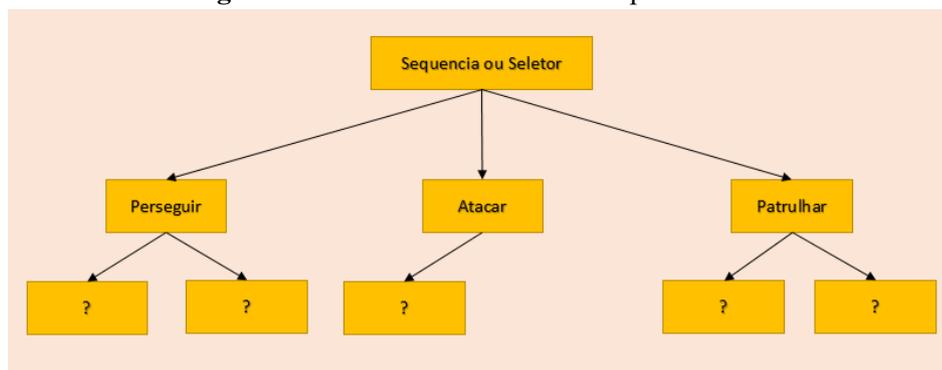
Criar NPCS mais inteligentes e mais realistas é uma tarefa essencial para desenvolver um ambiente virtual mais atrativo para os consumidores. Ao utilizar métodos de aprendizado de máquina para treinamento e desenvolvimento dos NPCS, é possível torná-los mais humanizados e reativos ao ambiente em que se encontram.

No momento atual, a indústria de jogos utiliza o método de árvores de comportamento para a criação das *Game AIs*, esse que traz apenas maneiras já pré-definidas e direcionadas pelo desenvolvedor.

As árvores de comportamento, também chamadas de *Behaviour Trees* (BTs), são arquiteturas que permitem que decisões sejam tomadas de acordo com os eventos e obstáculos apresentados aos *Non Player Characters* (NPC). “As BTs possuem um sistema muito em comum com uma Máquina de Estado Finita Hierárquica, mas ao invés de utilizar estados, o principal componente é chamado de Tarefa” (Moreira, 2022).

O nome do sistema é sugestivo, onde seguindo um modelo de árvores, as folhas são os comandos, que geram vários ramos como se fossem caminhos, onde dependendo do comando dado, chegará a melhor opção para aquele comando.

Figura 1. Modelo de Árvore de Comportamento.



Fonte: Moreira (2022)

Na Figura 1, é possível verificar um modelo árvore de comportamentos que indica algumas observações produzidas pelo NPC e as ações que o personagem pode tomar a partir delas.

Framework ML-Agents

Para a realização todo o processo de treinamento proposto no projeto, foi utilizado o *framework* de aprendizado de máquina oficial da *Unity Engine*, chamado de *ML-Agents*, que tem as suas funções e implementações baseadas em *PyTorch*.

Oracle (2022) define o *Pytorch* como “uma biblioteca de código-fonte aberto projetada com o Python em mente e construída para projetos de aprendizado de máquina”.

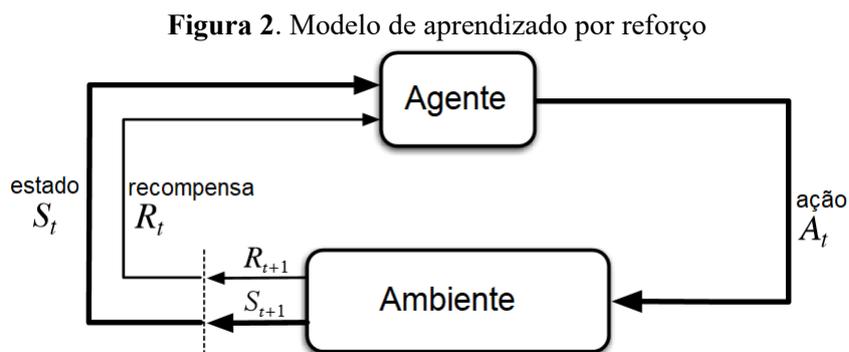
O *ML-Agents* possui uma vasta coletânea de bibliotecas e ferramentas que, auxiliam e tornam possível todo o processo de criação de ambientes, treinamento, aplicação e apuração dos resultados das inteligências artificiais.

Aprendizado por reforço

Dentro dos métodos de aprendizado de máquina disponíveis, o *Unity ML-Agents* possui suporte para alguns dos mais utilizados, entre eles está algoritmo de aprendizado por reforço, esse que foi utilizado no treinamento dos personagens não jogáveis.

No aprendizado por reforço (AR), a ideia principal se baseia na otimização da tomada de decisão via retornos de sucessos e fracassos no ambiente (OTTONI et al., 2015).

A figura abaixo mostra o ciclo treinamento através de um método de aprendizado por reforço.



Fonte: Neves (2021)⁷

Todo o processo de ajuste de recompensas e equações que envolvem o resultado são ajustadas automaticamente pelo *ML-Agents* e não necessitam de serem reajustadas manualmente em cada execução.

Para a compreensão do método utilizado, é de suma importância a assimilação de alguns elementos que serão utilizados no processo de treinamento da *Game AI*.

⁷NEVES, E. C. **Aprendizado por Reforço (2021)**. Disponível em: <<https://medium.com/turing-talks/aprendizado-por-reforco-1-introducao-7382ebb641ab>>. Acesso em nov. de 2022.

O agente (*agent*) é um componente da *Unity* que produz observações e realiza ações no ambiente, tais ações são determinadas através das observações do agente, produzidas por uma política definida dentro do motor de desenvolvimento.

A observação é uma informação parcial que descreve o estado do ambiente para o agente, dado as circunstâncias que ele se encontra. A partir da observação, o agente pode tomar uma decisão produzida pela política citada anteriormente e por fim, gerando a ação.

No término da ação realizada no ambiente, é definido a possibilidade de recebimento de recompensas, essas que são sinais providos ao fim cada passo que o agente realiza, para indicar se ele está correspondendo de acordo com o objetivo a ser atingido pelo treinamento.

Definição do ambiente de treinamento e recompensa

O ambiente desenvolvido para os testes dentro da *Unity Engine* foi restrito a um tamanho reduzido, dentro de uma cena de pouca complexidade em relação aos objetos para realizar o treinamento. A proposta principal ambiente é treinar o agente para que ele consiga perseguir e atingir um alvo que está em diversas posições diferentes dentro do espaço da demonstração.

O planejamento apropriado do ambiente é indispensável para que o processo seja eficaz e possa ser realmente utilizado em um NPC.

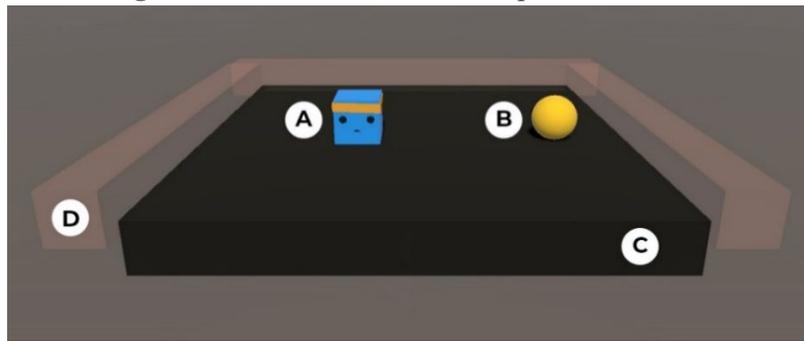
No aprendizado por reforço, o propósito ou objetivo do agente é formalizado em termos especiais, chamado de recompensa, essa que é passada pelo ambiente para o agente (BARTO; SUTTON, 2020).

Além do ecossistema feito dentro do motor de desenvolvimento através de uma cena, também é criado um ambiente virtual utilizando o *venv*⁸, que irá suportar e manter todo o processo de treinamento da *Game AI*.

Dentro do ambiente de treinamento projetado (letra C na figura 3), a recompensa será ativada quando o elemento utilizado para identificar o agente (letra A) tocar o alvo (letra B) que também está visível no espaço de treinamento.

⁸ Ferramenta que permite a criação e isolamento de diferentes de ambientes virtuais baseados em Python para criação de inteligências artificiais

Figura 3. Ambiente desenvolvido para treinamento



Fonte: Os autores

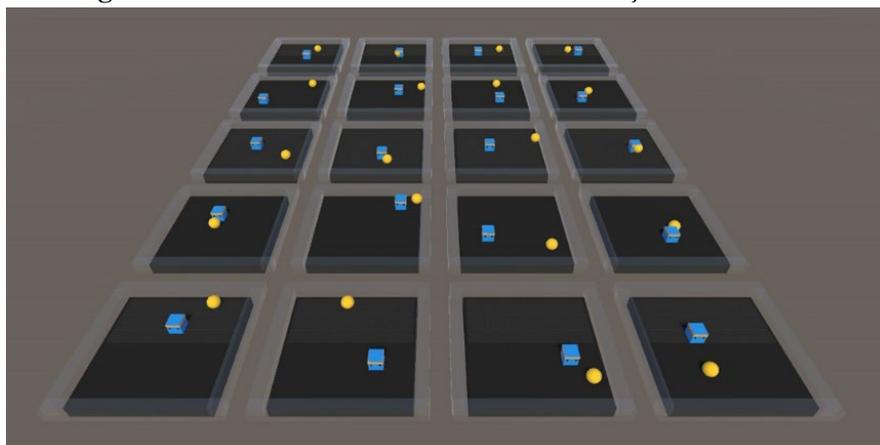
A recompensa de forma positiva possui peso de 1 e a forma negativa possui peso de -1 caso o agente atinja as bordas do ambiente proposto (letra D). O resultado esperado do treinamento é a média de recompensa do valor seja maior que 0.90.

O *ML-Agents* realiza o procedimento através de episódios. Esses iniciam-se quando o projeto passa para o estado de execução e terminam ao atingir algum dos critérios de recompensa, atribuindo os pesos respectivos para cada um dos resultados.

Caso o agente atinja mil passos dentro do mesmo episódio, ele será reiniciado sem que um peso seja atribuído no resultado, essa opção foi desenvolvida com o intuito de modelar o ambiente para e que o agente não aprenda apenas a evitar as paredes, mas sim, ir atrás do alvo.

A duração de todo o processo depende da capacidade e facilidade de processamento do computador que está executando o treinamento. Uma das soluções existentes e capazes de tornar o processo mais rápido é replicar os ambientes de treinamento para que sejam executados vários episódios simultaneamente, como pode ser observado na figura 4.

Figura 4. Ambiente de treinamento com execuções simultâneas



Fonte: Os autores

O processo de treinamento proposto foi dividido em três partes, com o intuito de treinar os módulos de forma mais assertiva para a obtenção de resultados consistentes durante todo o processo. Cada uma das etapas possui um limite máximo de execução de 500 mil passos, onde cada episódio está limitado na execução de mil passos.

O primeiro ambiente de treinamento ocorreu com ambos os modelos estáticos, tanto o agente quanto o alvo estático, sempre aparecem na mesma posição dentro do ambiente.

A segunda fase do treinamento ocorreu com a randomização de locais de aparição, tanto do agente quanto do alvo, com o intuito do modelo aprender a alcançar o alvo em locais diferentes, porém com um critério de randomização que não atinge o ambiente por completo, mas, sim, uma área central.

Por fim, a terceira e última fase do treinamento, assim como a anterior, possui randomização na posição de aparição dos objetos, no entanto, o raio de aparição dos modelos cobre a área completa do ambiente de treinamento.

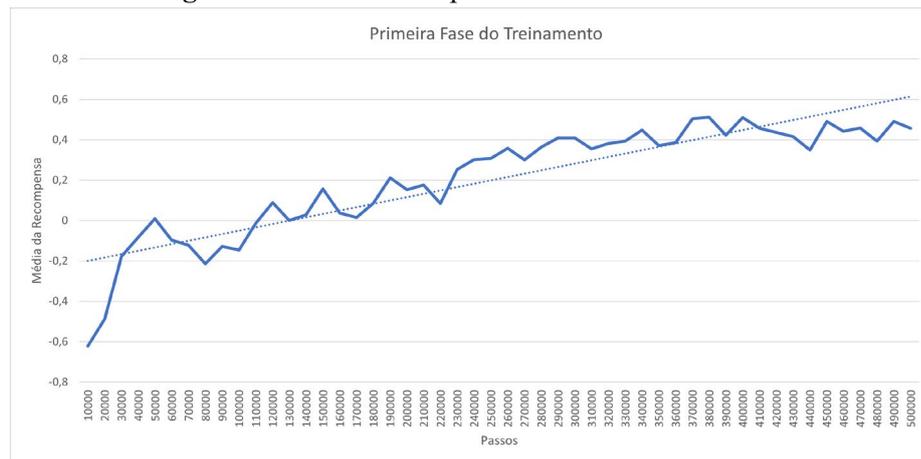
RESULTADOS

O processo de treinamento contou com 1,5 milhões de passos no total e teve a duração de aproximadamente 5 horas para a obtenção de resultados satisfatórios.

Ao observar os dados gerados através do *TensorBoard*⁹, é possível verificar a média de recompensa obtida, em função dos passos executados em cada treinamento.

No gráfico exposto na Figura 5 está representado os resultados da primeira fase do treinamento, onde é possível notar o crescimento contínuo da taxa de aprendizado.

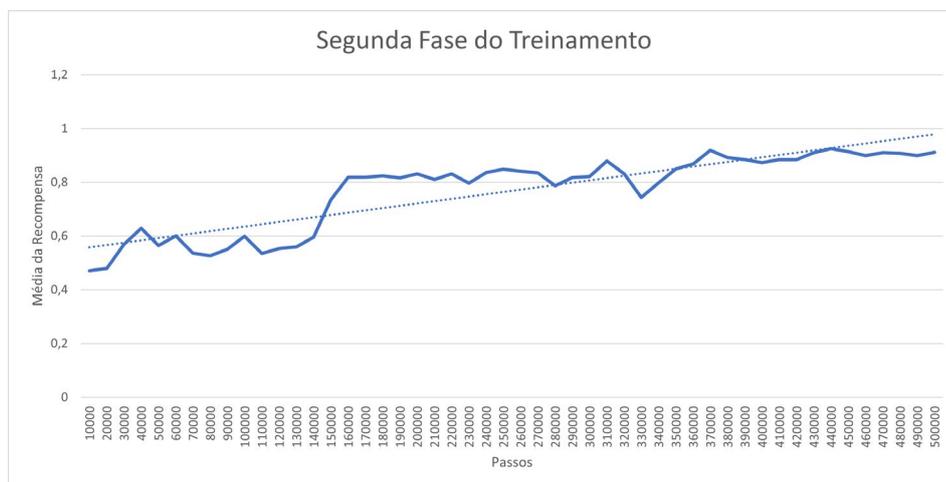
⁹ O *TensorBoard* é utilizado para gerar gráficos e tornar possível a visualização e o uso das ferramentas necessárias para a experimentação de *machine learning*.

Figura 5. Resultados da primeira fase do treinamento

Fonte: Os autores

Percebe-se que a execução só começou a obter resultados favoráveis com o objetivo desejado a partir dos cem mil passos.

A Figura 6 é apresenta os resultados da segunda fase do treinamento, onde no início desta etapa é possível verificar uma pequena dificuldade de manter a média de 0,6, o que está relacionado com as novas configurações do ambiente.

Figura 6. Resultados da segunda fase do treinamento

Fonte: Os autores

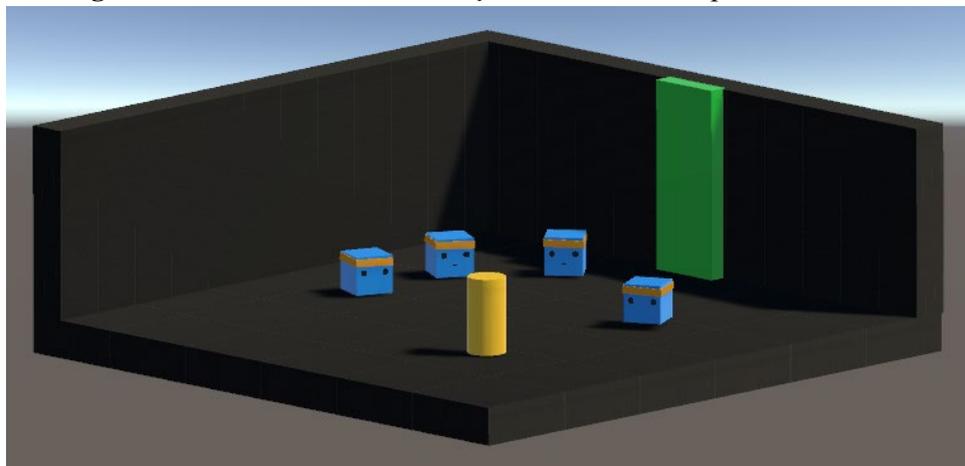
Na terceira e última fase do treinamento, é alcançado o objetivo definido para o treinamento, que é uma média de recompensa no valor de 0,92.

Figura 7. Resultados da terceira fase do treinamento

Fonte: Os autores

No processo de finalização de cada etapa é gerado um modelo de cérebro treinado em um formato de arquivo de extensão ONNX¹⁰, este que foi aplicado nos agentes dentro de um ambiente de execução para realizar o teste de funcionamento por inferência.¹¹

Na figura 8 é possível verificar o ambiente de demonstração desenvolvido para utilização dos agentes, contando com mais de um agente simultâneo.

Figura 8. Cena final de demonstração com o modelo aplicado

Fonte: Os autores

¹⁰ ONNX é um formato de arquivo de código aberto, utilizado para representar modelos de aprendizado de máquina.

¹¹ Método de comportamento definido no agente para que ele utilize o modelo treinado no momento de tomar decisões para realizar suas ações.

DISCUSSÃO

Através da apuração dos números atingidos e dos testes realizados, pode-se notar que ao fim da terceira etapa do treinamento, foi possível alcançar a média de 0.92 de recompensa, tornando o resultado obtido satisfatório para o modelo proposto.

Após a aplicação do modelo de cérebro gerado ao fim da terceira fase, foi possível realizar o teste por inferência e verificar a completa funcionalidade do agente fora do ambiente de treinamento, dessa vez, dentro do ecossistema de um jogo eletrônico.

O processo de divisão do treinamento em diferentes etapas se mostrou extremamente importante e funcional, visto que há uma evolução sólida ao fim de cada fase.

O modelo de cérebro exportado a partir das primeiras etapas trouxeram resultados aceitáveis, porém não satisfatórios. Devido a limitação no ambiente, ainda que a média de recompensa tenha atingido 0.9 em certos momentos da segunda fase, existiam áreas dentro do ambiente que não eram alcançadas pelo agente.

CONCLUSÃO

Neste trabalho, foi produzido um ambiente de testes com o propósito de evidenciar a possibilidade de utilização de *frameworks* de aprendizado de máquina para a busca e desenvolvimento de personagens não jogáveis mais inteligentes, reativos e críveis dentro do universo de um jogo eletrônico.

A partir da finalização dos procedimentos de treinamento e dos testes com os resultados obtidos, foi possível verificar o êxito na utilização do *ML-Agents* em conjunto com o modelo de aprendizado por reforço, para a criação de ambientes de aprendizagem, tal qual todo o processo de treinamento dos NPC's.

O uso dos modelos cerebrais gerados mostrou o quão eficaz o framework pode ser na criação de ações mais naturais e suaves para os NPC's, sem que haja uma definição feita via script de qual comportamento o agente deve seguir, mas sim através de um processo de aprendizado baseado no ambiente em que o agente se encontra e suas observações.

É importante ressaltar que a utilização de tais *frameworks* não implica na substituição completa dos métodos já utilizados pela indústria, tal qual as árvores de comportamento. Contudo, a junção destes sistemas pode trazer muitos benefícios durante o processo de desenvolvimento de jogos eletrônicos, visto que é um ambiente que ainda não é tão explorado.

REFERÊNCIAS

- AGIS, Ramiro A. GOTTIFREDI, Sebastian. GARCÍA, Alejandro J (2020). **An event-driven behavior trees extension to facilitate non-player multi-agent coordination in video games**. Expert Systems with Applications, v.155. Disponível em: <<https://doi.org/10.1016%2Fj.eswa.2020.113457>>. Acesso em ago. de 2022.
- BARTO, A. G. SUTTON. **Reinforcement Learning**. Cambridge, MA : The MIT Press, (2018). p 53.
- BELLMAN, R.E. **An Introduction to Artificial Intelligence: Can Computers Think?** Boyd & Fraser, 1978.
- FUNGE, John David. **Artificial Intelligence for Computer Games: An Introduction**. Natick: AK Peters. 2004.
- Invision Community. 2021. **Video game industry now worth \$163.1b**. Disponível em <<https://invisioncommunity.co.uk/video-game-industry-now-worth-163-1b/>>. Acesso em set. de 2022.
- KHAN, Umair A. OKADA, Yoshihiro. (2014). **Emotional Decision Making Response of Non-Playable Characters In a Role-Playing Game**. IADIS International Journal on Computer Science and Information Systems, 9(2), 53-66.
- KISHIMOTO, A. **Inteligência Artificial em Jogos Eletrônicos**. Disponível em: <https://www.academia.edu/2231411/Intelig%C3%Aancia_Artificial_em_Jogos_Eletr%C3%B4nicos>. Acesso em ago. de 2022.
- KURZWEIL, Ray. **The Age of Spiritual Machines**. Massachusetts: The MIT Press, 1990, p. 22.

MITCHELL, T. M. **Machine Learning**. United States: McGraw-Hill, (1997), p.26.

MOREIRA, F. **Árvore de Comportamento – IA**. Disponível em:<<https://evolvers.com.br/arvore-de-comportamento-ia/>>. Acesso em out. de 2022.

ORACLE. **O que é o Pytorch? (2022)**. Disponível em: <<https://developer.oracle.com/pt-BR/learn/technical-articles/1481879246306-134-what-is-pytorch>>. Acesso em nov. de 2022.

OTTONI, A. L. C., OLIVEIRA, M. S., Nepomuceno, E. G. e LAMPERTI, R. D. (2015). **Análise do aprendizado por reforço via modelos de regressão logística: Um estudo de caso no futebol de robôs**. Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia 10: 44–49.

UNITY. **ML-Agents Toolkit Overview (2020)**. Disponível em <<https://github.com/Unity-Technologies/ml-agents/blob/main/docs/ML-Agents-Overview.md>>. Acesso em ago. de 2022.