

ANÁLISE DE DESEMPENHO DE DIFERENTES ENGINES NO DESENVOLVIMENTO DE JOGOS COM IMPLEMENTAÇÃO DE WIZARD PARA RECOMENDAÇÃO DAS ENGINES

Lucas Rodrigues da Silva, Luciene Cavalcanti Rodrigues

e-mail: lucas.silva693@fatec.sp.gov.br; luciene.rodrigues@fatec.sp.gov.br

Faculdade de Tecnologia de São José do Rio Preto

Resumo: A análise de desempenho das principais engines de jogos – Unity, Unreal Engine, Godot e CryEngine – destaca suas características, funcionalidades e adequação a diferentes tipos de projetos. A Unity é valorizada por sua flexibilidade e suporte multiplataforma, facilitando o desenvolvimento através de uma interface intuitiva e uma vasta biblioteca de ativos. A Unreal Engine se sobressai com capacidades gráficas avançadas e ferramentas robustas, ideal para jogos de alta fidelidade visual, apesar de sua curva de aprendizado mais íngreme. A Godot, uma engine open-source, oferece simplicidade e eficiência, sendo uma excelente escolha para desenvolvedores independentes e projetos menores. A CryEngine proporciona gráficos realistas e detalhados, exigindo, contudo, um conhecimento técnico avançado e recursos de hardware robustos. A escolha da engine adequada depende das necessidades específicas do projeto, do orçamento e das habilidades da equipe de desenvolvimento, com cada engine apresentando vantagens e desafios distintos, para tanto, foi desenvolvido um APP que de acordo com as respostas do usuário, recomendará uma Engine.

Palavras-chave: engines de jogos, Unity, Unreal Engine, desenvolvimento, desempenho técnico, flexibilidade, open-source.

***Abstract:** The performance analysis of major game engines – Unity, Unreal Engine, Godot, and CryEngine – highlights their characteristics, functionalities, and suitability for different types of projects. Unity is valued for its flexibility and cross-platform support, facilitating development through an intuitive interface and an extensive asset library. Unreal Engine stands out with advanced graphical capabilities and robust tools, ideal for high-fidelity visual games, despite its steeper learning curve. Godot, an open-source engine, offers simplicity and efficiency, making it an excellent choice for independent developers and smaller projects. CryEngine provides realistic and detailed graphics but requires advanced technical knowledge and robust hardware resources. The choice of the appropriate engine depends on the project's specific needs, budget, and the development team's skills, with each engine presenting distinct advantages and challenges.*

Keywords: game engines, Unity, Unreal Engine, development, technical performance, flexibility, open-source.

1. INTRODUÇÃO

O desenvolvimento de jogos eletrônicos tem evoluído significativamente ao longo das últimas décadas, tornando-se uma das indústrias de entretenimento mais lucrativas e dinâmicas do mundo. A escolha da engine de jogo é uma decisão crucial que impacta diretamente a eficiência do desenvolvimento, a qualidade do produto final, e a experiência do usuário. Uma engine de jogo, ou motor de jogo, é um software que fornece uma estrutura necessária para criar e desenvolver videogames.

O desenvolvimento de jogos eletrônicos é uma área dinâmica e altamente competitiva, que demanda ferramentas poderosas e eficientes para a criação de experiências de jogos imersivas e de alta qualidade. As engines de jogos, que fornecem a base tecnológica para o desenvolvimento de jogos, desempenham um papel crucial nesse processo. Entre as diversas opções disponíveis no mercado, Unity e Unreal Engine se destacam como duas das mais populares e amplamente utilizadas, cada uma oferecendo um conjunto distinto de características e funcionalidades que atendem a diferentes necessidades dos desenvolvedores.

A Unity, lançada em 2005, é amplamente reconhecida por sua flexibilidade e suporte multiplataforma, permitindo que desenvolvedores criem jogos para PCs, consoles, dispositivos móveis e sistemas de realidade virtual com uma única base de código. Sua interface intuitiva e a vasta biblioteca de ativos na Unity Asset Store fazem dela uma escolha popular para desenvolvedores independentes e grandes estúdios. A Unity é especialmente valorizada pela sua capacidade de facilitar o desenvolvimento rápido e eficiente, embora a otimização para gráficos complexos possa apresentar desafios adicionais.

Por outro lado, a Unreal Engine, desenvolvida pela Epic Games e lançada em 1998, é famosa por suas capacidades gráficas avançadas e ferramentas robustas. Esta engine tem sido a escolha preferida para muitos jogos de alta fidelidade visual e projetos AAA, graças ao seu sistema de renderização avançado que permite a criação de gráficos fotorrealistas e efeitos visuais impressionantes. A *Unreal Engine* também se destaca pelo seu sistema de script visual Blueprints, que facilita a criação de lógica de jogo sem necessidade de programação extensiva, permitindo uma rápida prototipagem e implementação de ideias.

A análise de desempenho dessas duas engines é essencial para compreender suas capacidades técnicas e práticas, assim como para identificar as melhores práticas e estratégias de otimização. Aspectos como tempo de renderização, uso de CPU e GPU, latência, e experiência do usuário são fundamentais para determinar qual engine é mais adequada para determinados tipos de projetos e quais *trade-offs* os desenvolvedores podem esperar ao optar por uma dessas ferramentas.

2. JUSTIFICATIVA

Este estudo tem como objetivo comparar o desempenho da *Unity* e da *Unreal Engine*, destacando suas principais vantagens e limitações e fornecendo insights valiosos para desenvolvedores que buscam maximizar a eficiência de seu processo de desenvolvimento e a qualidade de seus jogos. No desenvolvimento de jogos, a escolha da engine adequada é uma decisão crucial que pode impactar significativamente a eficiência do desenvolvimento e a qualidade do produto final. Para facilitar essa escolha, foi criado um Wizard interativo que guia o usuário através de uma série de perguntas, recomendando a engine mais adequada com base nas respostas fornecidas. Este Wizard foi desenvolvido em HTML e pode ser acessado facilmente via dispositivos móveis, proporcionando uma ferramenta prática e acessível para desenvolvedores.

3. METODOLOGIA

A metodologia significa estudo do método e refere-se ao conjunto de processos necessários para alcançar os fins de uma investigação, de uma implantação ou da criação de um novo produto. É o procedimento geral que mostra como se respondeu, aos objetivos estabelecidos e como se ajustou aos objetivos específicos, envolvendo a definição de como o trabalho se realizou. A seguir, detalha-se a metodologia utilizada no projeto de análise de desempenho de diferentes engines no desenvolvimento de jogos.

A revisão da literatura para a análise de desempenho das engines Unity e Unreal Engine seguiu uma metodologia estruturada e sistemática, essencial para coletar, analisar e sintetizar informações relevantes de diversas fontes acadêmicas e práticas. Inicialmente, definiu-se

claramente os objetivos da revisão.

O objetivo principal foi comparar o desempenho das engines Unity e Unreal Engine, destacando suas características, funcionalidades e a adequação a diferentes tipos de projetos de desenvolvimento de jogos. Subobjetivos incluíram a análise do tempo de renderização, uso de CPU/GPU e latência de cada engine, além de examinar a experiência do usuário e o feedback dos desenvolvedores, identificando as melhores práticas e estratégias de otimização.

O trabalho caracterizou-se como uma pesquisa descritiva e exploratória. Foi descritiva porque procurou caracterizar as propriedades e o desempenho das diferentes engines de jogos. Foi exploratória porque buscou compreender melhor os aspectos técnicos e práticos envolvidos no uso dessas engines. A abordagem foi tanto quantitativa, através de testes de desempenho e métricas técnicas, quanto qualitativa, através de entrevistas e feedback de desenvolvedores.

Para a coleta de dados, selecionaram-se fontes de informações baseadas em relevância, credibilidade e atualidade. As principais fontes incluíram artigos científicos e conferências relacionadas ao desenvolvimento de jogos e tecnologias de engines, teses e dissertações acadêmicas, livros especializados em arquitetura e desenvolvimento de engines de jogos, publicações e relatórios técnicos de empresas desenvolvedoras de engines, como *Unity Technologies* e *Epic Games*, além de fóruns de desenvolvedores e comunidades online, como *Unity Forum* e *Unreal Engine Community*. Entrevistas e depoimentos de desenvolvedores experientes também foram considerados.

A fim de garantir a relevância e a qualidade das fontes selecionadas, estabeleceram-se critérios específicos de inclusão e exclusão. Estudos que analisaram o desempenho técnico, como tempo de renderização, uso de CPU/GPU e latência, experiência do usuário e feedback de desenvolvedores para *Unity* e *Unreal Engine* foram incluídos. Além disso, priorizaram-se publicações dos últimos dez anos para assegurar a atualidade das informações. Por outro lado, estudos que não abordavam diretamente os aspectos de desempenho das engines ou que eram anteriores a esse período foram excluídos.

Após a coleta de dados, os materiais foram revisados e analisados de forma crítica. A análise focou em identificar padrões, pontos fortes e fracos de cada engine, bem como as percepções dos desenvolvedores sobre o uso dessas ferramentas. A síntese das informações coletadas permitiu uma compreensão abrangente das capacidades e limitações das engines Unity e Unreal Engine, proporcionando uma base sólida para a comparação de seu desempenho e auxiliando na identificação das melhores práticas e estratégias de otimização.

Essa abordagem metodológica garantiu uma revisão detalhada e sistemática da literatura, contribuindo para uma análise fundamentada e informada sobre o desempenho das principais engines de jogos, Unity e Unreal Engine.

Os recursos materiais necessários incluíram computadores de alto desempenho para executar os testes de desempenho, notebooks para coleta e análise de dados em campo, e licenças de software para acesso às engines de jogos. As etapas e atividades desenvolvidas incluíram a revisão da literatura sobre engines de jogos, a seleção das engines a serem analisadas, a coleta de dados técnicos e qualitativos, a realização de testes de desempenho, a análise e interpretação dos dados coletados, e a elaboração do relatório final com os resultados.

3.1 Desenvolvimento do Wizard

O Wizard foi desenvolvido utilizando HTML e JavaScript, garantindo compatibilidade com navegadores modernos e facilidade de acesso. A lógica de decisão foi baseada nas características principais de quatro engines populares: Unity, Unreal Engine, Godot e CryEngine. As perguntas foram formuladas para capturar as necessidades específicas dos desenvolvedores, como o tipo de jogo, nível de experiência, plataformas de destino, orçamento e preferências de linguagem de programação.

3.2 Lógica de Decisão

A lógica do Wizard foi projetada para filtrar as opções com base nas respostas às perguntas. Por exemplo, se um usuário deseja desenvolver um jogo 2D com orçamento gratuito e facilidade de uso, o Wizard recomendará a Godot. Cada resposta elimina engines que não atendem aos critérios especificados, proporcionando uma recomendação precisa.

3.3 Ferramentas Utilizadas

Para o desenvolvimento do Wizard, foi utilizado um editor de texto (como VSCode) para escrever o código HTML e JavaScript. O design foi mantido simples e funcional para garantir a clareza e a usabilidade da ferramenta.

4. REVISÃO DA LITERATURA

4.1. Desenvolvimento de jogos

O desenvolvimento de jogos eletrônicos é uma área em constante evolução, refletindo mudanças tecnológicas e culturais que moldam a forma como interagimos com o entretenimento digital. Desde os primeiros experimentos com gráficos rudimentares e jogabilidade simples, a indústria de jogos transformou-se em um campo vasto e complexo, abrangendo desde pequenos desenvolvedores independentes até grandes estúdios que produzem títulos de sucesso mundial. Essa evolução não só mudou a maneira como os jogos são desenvolvidos, mas também ampliou significativamente o impacto social, econômico e cultural dos jogos eletrônicos.

Historicamente, o desenvolvimento de jogos começou como uma atividade experimental e acadêmica nos anos 1950 e 1960, com jogos simples como "Tennis for Two" e "Spacewar!". No entanto, foi na década de 1970 que a indústria de jogos começou a ganhar forma comercial, com o lançamento de jogos de arcade como "Pong" e "Space Invaders". A partir dos anos 1980, com o advento dos consoles domésticos e dos computadores pessoais, a criação de jogos se expandiu rapidamente, levando à formação de um mercado global de entretenimento digital.

A complexidade dos jogos modernos exige o uso de ferramentas sofisticadas conhecidas como engines de jogos. Essas engines fornecem a infraestrutura necessária para o desenvolvimento de jogos, incluindo renderização gráfica, física, inteligência artificial e gerenciamento de som e entrada de dados do usuário. Segundo Gregory (2017), uma engine de jogo é essencialmente um software que permite aos desenvolvedores criar e gerenciar os diversos componentes de um jogo de forma integrada e eficiente.

Atualmente, existem várias engines de jogos amplamente utilizadas, cada uma com suas próprias características e áreas de especialização. Unity, Unreal Engine, Godot e CryEngine são algumas das mais conhecidas. A Unity é conhecida por sua flexibilidade e acessibilidade, tornando-se popular tanto entre desenvolvedores independentes quanto em grandes estúdios.

A Unreal Engine, por sua vez, destaca-se por seus gráficos de alta qualidade e ferramentas poderosas, sendo preferida para jogos de alto orçamento e experiências visuais imersivas. Godot é uma engine open-source que tem ganhado popularidade devido à sua simplicidade e eficiência, ideal para projetos menores. Já a CryEngine é famosa pelo seu desempenho gráfico impressionante e realismo visual, frequentemente utilizada em jogos que demandam alta fidelidade gráfica.

O'Hagan, Coleman e O'Connor (2014) destacam que o desenvolvimento de jogos é um processo iterativo que envolve várias etapas, desde a concepção e design até a programação, testes e lançamento. Cada etapa do processo de desenvolvimento requer ferramentas específicas e habilidades diversas, tornando a escolha da engine de jogo uma decisão crítica que pode impactar diretamente o sucesso do projeto.

Além dos aspectos técnicos, o desenvolvimento de jogos também envolve considerações de mercado e público-alvo. Medeiros (2019) argumenta que a publicidade em jogos digitais, conhecida como *advergaming*, é uma estratégia crescente para atrair consumidores e aumentar o engajamento do público. Jogos bem-sucedidos não são apenas tecnicamente robustos, mas também precisam ser atraentes e envolventes para os jogadores.

A interatividade e a imersão são elementos chave no design de jogos modernos. Amaro (2014) explora como os jogos digitais se tornaram uma forma crucial de comunicação nas novas mídias, permitindo aos jogadores interagir de maneiras inovadoras e significativas. Jogos como "The Last of Us", desenvolvido na Unreal Engine, exemplificam como narrativas complexas e emocionais podem ser entregues através de ambientes altamente interativos e visualmente impressionantes.

Com o avanço da tecnologia, novas possibilidades estão surgindo no desenvolvimento de jogos. Fernandes et al. (2015) discutem a aplicação da inteligência artificial e da realidade aumentada em jogos, ampliando ainda mais o potencial de inovação na indústria. Essas tecnologias permitem criar experiências mais imersivas e personalizadas, onde os jogadores podem interagir com ambientes virtuais de maneira cada vez mais natural e intuitiva.

4.2. Importância da escolha da engine de jogo

A escolha da engine de jogo é uma das decisões mais cruciais no processo de desenvolvimento de jogos eletrônicos. Essa escolha impacta diretamente diversos aspectos do desenvolvimento, desde a qualidade gráfica até a eficiência do processo de produção e a experiência final do jogador. De acordo com Gregory (2017), uma engine de jogo fornece a infraestrutura essencial que permite aos desenvolvedores criar e gerenciar os componentes de um jogo de maneira integrada e eficiente, facilitando a implementação de funcionalidades complexas e garantindo um desempenho otimizado.

Um dos principais fatores que tornam a escolha da engine tão importante é a flexibilidade e a escalabilidade que ela oferece. Por exemplo, a Unity é amplamente reconhecida por sua versatilidade, suportando uma vasta gama de plataformas, desde dispositivos móveis até consoles

de última geração e aplicações de realidade virtual. Essa flexibilidade permite que desenvolvedores independentes e grandes estúdios adaptem seus projetos para diferentes mercados e públicos. Como destaca Šmíd (2017), a Unity é particularmente popular entre desenvolvedores independentes devido à sua facilidade de uso e vasta comunidade de suporte.

Por outro lado, a Unreal Engine, desenvolvida pela Epic Games, é conhecida por suas capacidades gráficas avançadas e ferramentas de desenvolvimento robustas. A Unreal Engine é frequentemente escolhida para projetos de alto orçamento que exigem gráficos de alta fidelidade e experiências visuais imersivas. Segundo Cavalcante e Pereira (2018), a Unreal Engine oferece um conjunto de ferramentas que permite aos desenvolvedores criar jogos com gráficos realistas e efeitos visuais impressionantes, tornando-a a preferida para jogos AAA e aplicações que demandam um alto nível de detalhe visual.

A escolha da engine também influencia o tempo e os custos de desenvolvimento. Engines que oferecem ferramentas integradas e fluxos de trabalho eficientes podem reduzir significativamente o tempo necessário para desenvolver e lançar um jogo. Por exemplo, Godot, uma engine open-source, é valorizada por sua simplicidade e eficiência, tornando-se uma opção atraente para projetos menores ou desenvolvedores que buscam um maior controle sobre o código fonte. Como Alves (2021) observa, a simplicidade da Godot permite um desenvolvimento mais rápido e menos propenso a erros, especialmente em projetos de menor escala.

Além dos aspectos técnicos e econômicos, a escolha da engine de jogo também tem implicações na capacidade de inovação e na experiência do usuário. Engines como a CryEngine, conhecida por seu desempenho gráfico superior, permitem a criação de ambientes realistas e envolventes, que podem proporcionar uma experiência de jogo mais imersiva e satisfatória. A capacidade de renderizar gráficos de alta qualidade e simular físicas complexas é crucial para jogos que buscam oferecer uma experiência visual e interativa de ponta. Conforme destaca Scherer, Batista e Mendes (2020), a CryEngine se destaca por seu desempenho gráfico e realismo visual, frequentemente utilizada em jogos que demandam alta fidelidade gráfica.

A comunidade de suporte e os recursos disponíveis também são fatores importantes na escolha da engine. Engines com uma grande base de usuários e uma comunidade ativa oferecem uma riqueza de recursos, tutoriais e suporte técnico, o que pode ser crucial para resolver problemas e implementar funcionalidades avançadas. Por exemplo, a comunidade da Unity é conhecida por ser particularmente ativa e abrangente, oferecendo uma ampla gama de recursos de aprendizado e suporte técnico, o que facilita a vida dos desenvolvedores, especialmente os iniciantes (Šmíd, 2017).

Em resumo, a escolha da engine de jogo adequada é fundamental para o sucesso de um projeto de desenvolvimento de jogos. Ela afeta não apenas a qualidade técnica do produto final, mas também a eficiência do processo de desenvolvimento, os custos envolvidos e a capacidade de inovar e proporcionar uma experiência de jogo envolvente. Como ressalta Gregory (2017), a escolha da engine deve ser baseada nas necessidades específicas do projeto, nas habilidades da equipe de desenvolvimento e nos objetivos finais do jogo.

4.3. Histórico e evolução das engines de jogos

O desenvolvimento de jogos eletrônicos tem uma rica história que remonta aos primeiros experimentos com computadores e gráficos digitais. As engines de jogos, que são a espinha dorsal

do desenvolvimento de jogos modernos, passaram por uma evolução significativa, moldada por avanços tecnológicos e mudanças nas expectativas dos jogadores.

Nos primórdios dos jogos eletrônicos, na década de 1950 e 1960, os jogos eram simples e geralmente criados por programadores individuais ou pequenos grupos em universidades e laboratórios de pesquisa. Jogos como "Tennis for Two" e "Spacewar!" foram alguns dos primeiros exemplos de jogos eletrônicos, mas a noção de uma engine de jogo ainda não existia (Gregory, 2017).

A década de 1970 viu o surgimento da indústria de jogos de arcade, com títulos icônicos como "Pong" (1972) da Atari e "Space Invaders" (1978) da Taito. Esses jogos eram desenvolvidos usando hardware específico e software altamente otimizado, mas ainda sem uma separação clara entre o código do jogo e o mecanismo subjacente (Scherer, Batista e Mendes, 2020).

Foi nos anos 1980, com o advento dos computadores pessoais e dos consoles domésticos, que a ideia de engines de jogos começou a tomar forma. Desenvolvedores começaram a criar bibliotecas reutilizáveis de código para facilitar o desenvolvimento de novos jogos. Um exemplo notável é a "id Tech 1", desenvolvida pela id Software para o jogo "Doom" (1993). Esta engine permitiu uma flexibilidade sem precedentes na criação de níveis e na modificação do jogo, influenciando profundamente o design de jogos futuros (Gregory, 2017).

A década de 1990 trouxe avanços significativos com o surgimento de engines de jogos mais sofisticadas e versáteis. A Unreal Engine, lançada pela Epic Games em 1998, introduziu ferramentas poderosas para criação de gráficos 3D e efeitos visuais, estabelecendo um novo padrão para a indústria. A CryEngine, lançada em 2002 pela Crytek, destacou-se por seu realismo visual e capacidade de renderização de grandes ambientes abertos, como demonstrado no jogo "Far Cry" (2004) (Scherer, Batista e Mendes, 2020).

Com o início do século XXI, as engines de jogos continuaram a evoluir, impulsionadas pela crescente demanda por gráficos mais realistas e jogabilidade mais complexa. A Unity, lançada em 2005, revolucionou o desenvolvimento de jogos ao oferecer uma plataforma acessível e flexível, suportando uma ampla gama de dispositivos, desde PCs e consoles até dispositivos móveis e realidade virtual. A Unity tornou-se rapidamente popular entre desenvolvedores independentes e grandes estúdios, devido à sua facilidade de uso e ampla comunidade de suporte (Šmíd, 2017).

A Unreal Engine 3, lançada em 2006, e sua sucessora, Unreal Engine 4, lançada em 2014, continuaram a empurrar os limites da tecnologia de jogos, oferecendo ferramentas avançadas para criação de gráficos fotorrealistas e simulações físicas complexas. Essas engines são amplamente usadas em jogos AAA, que exigem gráficos de alta fidelidade e experiências imersivas (Gregory, 2017).

Nos últimos anos, a tendência de engines de jogos open-source ganhou força, com a Godot sendo um exemplo notável. Lançada oficialmente em 2014, a Godot oferece uma alternativa gratuita e aberta às engines comerciais, permitindo aos desenvolvedores acesso total ao código fonte e a liberdade de modificar e adaptar a engine às suas necessidades específicas. Esta flexibilidade tem atraído muitos desenvolvedores independentes e pequenos estúdios (Alves, 2021).

A evolução das engines de jogos reflete as mudanças tecnológicas e as crescentes expectativas dos jogadores por experiências de jogo mais envolventes e realistas. Hoje, as engines de jogos não são apenas ferramentas para desenvolver jogos, mas plataformas completas que

suportam uma ampla gama de aplicações, desde simuladores de treinamento até experiências de realidade aumentada e virtual. Como observa Gregory (2017), as engines de jogos modernas oferecem um conjunto abrangente de ferramentas que permitem aos desenvolvedores criar experiências de jogos ricas e variadas, atendendo a uma ampla gama de necessidades e públicos.

4.4. Principais engines de jogos (Unity, Unreal Engine, Godot, CryEngine)

As engines de jogos são ferramentas essenciais que fornecem a infraestrutura necessária para o desenvolvimento de jogos eletrônicos. Entre as mais utilizadas e reconhecidas no mercado estão a Unity, a Unreal Engine, a Godot e a CryEngine. Cada uma dessas engines possui características únicas, que as tornam adequadas para diferentes tipos de projetos e necessidades de desenvolvimento.

4.4.1. Unity

A Unity é uma das engines de jogos mais populares e versáteis do mercado, amplamente utilizada tanto por desenvolvedores independentes quanto por grandes estúdios. Lançada em 2005, a Unity se destaca por sua capacidade de suportar uma ampla gama de plataformas, incluindo PCs, consoles, dispositivos móveis e realidade virtual. Sua flexibilidade e facilidade de uso a tornaram a escolha preferida para muitos desenvolvedores.

Segundo Šmíd (2017), a Unity é especialmente popular devido à sua interface amigável e à vasta biblioteca de ativos disponíveis na Unity Asset Store, que permite aos desenvolvedores acelerar o processo de criação de jogos. Além disso, a Unity oferece um robusto sistema de scripts baseado em C#, que proporciona grande flexibilidade e poder aos desenvolvedores. A engine também possui uma comunidade ativa e um grande número de recursos de aprendizado, o que facilita a resolução de problemas e a implementação de novas funcionalidades.

4.4.2. Unreal Engine

Desenvolvida pela Epic Games, a Unreal Engine é conhecida por suas capacidades gráficas avançadas e ferramentas poderosas, que a tornam ideal para o desenvolvimento de jogos de alta qualidade e visualmente impressionantes. Desde seu lançamento em 1998, a Unreal Engine tem sido a escolha preferida para muitos jogos AAA e experiências de realidade virtual.

De acordo com Cavalcante e Pereira (2018), a Unreal Engine oferece uma série de ferramentas avançadas para criação de gráficos 3D, incluindo um sistema de iluminação realista, simulação de física e efeitos visuais de alta qualidade. A Unreal Engine 4, lançada em 2014, introduziu um novo sistema de renderização que permite gráficos fotorrealistas, além de um editor visual de scripts, o Blueprints, que facilita a criação de lógica de jogo sem a necessidade de codificação extensiva. A engine também suporta uma ampla gama de plataformas, o que a torna uma opção versátil para desenvolvedores.

4.4.3. Godot

A Godot é uma engine de jogos open-source que tem ganhado popularidade entre

desenvolvedores independentes devido à sua simplicidade e eficiência. Lançada oficialmente em 2014, a Godot oferece uma alternativa gratuita às engines comerciais, permitindo aos desenvolvedores acesso total ao código fonte e a liberdade de modificar e adaptar a engine às suas necessidades específicas.

Alves (2021) destaca que a Godot é especialmente adequada para projetos menores ou para desenvolvedores que desejam um maior controle sobre o processo de desenvolvimento. A engine possui uma interface intuitiva e um sistema de scripts baseado em GDScript, uma linguagem de programação semelhante ao Python, que é fácil de aprender e usar. A Godot também suporta tanto gráficos 2D quanto 3D, oferecendo flexibilidade para diferentes tipos de projetos.

4.4.4. CryEngine

Desenvolvida pela Crytek, a CryEngine é famosa por seu desempenho gráfico superior e realismo visual. Desde seu lançamento em 2002, a CryEngine tem sido utilizada para criar jogos que impressionam pelo nível de detalhe e pela qualidade dos gráficos. Jogos como "Far Cry" e "Crysis" demonstraram as capacidades da engine em criar ambientes imersivos e visualmente impressionantes.

Scherer, Batista e Mendes (2020) observam que a CryEngine oferece uma série de ferramentas avançadas para criação de gráficos 3D, incluindo simulação de física, sistemas de partículas e renderização em tempo real. A engine é particularmente adequada para jogos que exigem alta fidelidade gráfica e ambientes detalhados. No entanto, a complexidade e os requisitos de hardware da CryEngine podem ser desafiadores para desenvolvedores menores ou com recursos limitados.

Cada uma dessas engines de jogos oferece um conjunto distinto de ferramentas e funcionalidades, adequadas para diferentes tipos de projetos e necessidades de desenvolvimento. A escolha da engine certa depende de vários fatores, incluindo o tipo de jogo que se deseja criar, o orçamento disponível, as habilidades da equipe de desenvolvimento e as plataformas de destino. Como destaca Gregory (2017), entender as características e capacidades de cada engine é crucial para fazer uma escolha informada que maximizará a eficiência do desenvolvimento e a qualidade do produto final.

5. DESENVOLVIMENTO

5.1. Características e funcionalidades das engines

As engines de jogos são fundamentais no desenvolvimento de jogos eletrônicos, fornecendo as ferramentas e funcionalidades necessárias para criar experiências de jogo imersivas e envolventes. Cada engine possui características e funcionalidades únicas que a tornam adequada para diferentes tipos de projetos e necessidades. A seguir, destacamos as principais características e funcionalidades das engines Unity, Unreal Engine, Godot e CryEngine.

5.1.1. Unity

A Unity é conhecida por sua flexibilidade e ampla gama de funcionalidades, o que a torna

uma escolha popular para desenvolvedores de todos os níveis. Lançada em 2005, a Unity suporta diversas plataformas, incluindo PC, consoles, dispositivos móveis, e realidade aumentada/virtual (Šmíd, 2017). Essa versatilidade permite que os desenvolvedores alcancem um público amplo com um único código-base.

A Unity utiliza C# como sua principal linguagem de script, oferecendo um ambiente de desenvolvimento robusto e intuitivo. Além disso, a Unity Asset Store proporciona uma vasta biblioteca de ativos prontos para uso, como modelos 3D, texturas, animações e scripts, que podem acelerar significativamente o processo de desenvolvimento. Outra funcionalidade importante é o sistema de componentes da Unity, que permite aos desenvolvedores construir funcionalidades complexas através da combinação de scripts modulares e reutilizáveis.

5.1.2. Unreal Engine

Desenvolvida pela Epic Games, a Unreal Engine é reconhecida por suas capacidades gráficas avançadas e ferramentas poderosas. Lançada em 1998, a Unreal Engine suporta uma ampla gama de plataformas e é ideal para o desenvolvimento de jogos de alta qualidade e visualmente impressionantes (Cavalcante e Pereira, 2018).

A Unreal Engine inclui um sistema de renderização em tempo real que permite a criação de gráficos fotorrealistas e efeitos visuais de alta qualidade. A engine também oferece um editor visual de scripts chamado Blueprints, que facilita a criação de lógica de jogo sem necessidade de programação extensiva. Além disso, a Unreal Engine suporta ferramentas avançadas para simulação de física, sistemas de partículas, e animação, e oferece integração com diversas ferramentas externas e plugins, expandindo suas capacidades.

5.1.3. Godot

A Godot é uma engine de jogos open-source que tem ganhado popularidade devido à sua simplicidade e eficiência. Lançada oficialmente em 2014, a Godot suporta gráficos 2D e 3D, e é especialmente adequada para projetos menores ou para desenvolvedores independentes (Alves, 2021).

Uma das características distintivas da Godot é seu sistema de cenas, que organiza os projetos em uma hierarquia de nós, facilitando a reutilização de componentes e a organização do código. A Godot utiliza GDScript, uma linguagem de script semelhante ao Python, que é fácil de aprender e usar. Além disso, a engine oferece ferramentas integradas para animação, física, e scripts, e suporta exportação multiplataforma, permitindo que os desenvolvedores lancem seus jogos em diversas plataformas com facilidade.

5.1.4. CryEngine

Desenvolvida pela Crytek, a CryEngine é conhecida por seu desempenho gráfico superior e realismo visual. Desde seu lançamento em 2002, a CryEngine tem sido utilizada para criar jogos que impressionam pelo nível de detalhe e pela qualidade dos gráficos (Scherer, Batista e Mendes, 2020).

A CryEngine inclui um avançado sistema de renderização que permite a criação de

gráficos detalhados e realistas. A engine oferece ferramentas poderosas para simulação de física, animação, e inteligência artificial, permitindo a criação de ambientes imersivos e dinâmicos. A CryEngine também possui um editor visual de materiais, que facilita a criação e ajuste de shaders e texturas, e suporta uma ampla gama de plataformas, embora sua complexidade e requisitos de hardware possam representar um desafio para desenvolvedores com recursos limitados.

5.2 Desenvolvimento do Wizard para Escolha de Engine de Desenvolvimento de Jogos

Para facilitar a escolha de uma engine de desenvolvimento de jogos, foi criado um Wizard interativo em HTML. Este Wizard faz perguntas específicas ao usuário e, com base nas respostas, recomenda a engine mais adequada. As perguntas e as possíveis respostas foram cuidadosamente elaboradas para refletir as necessidades comuns dos desenvolvedores de jogos.

5.3 Estrutura do Wizard

O Wizard consiste nas seguintes perguntas:

1. **Qual tipo de jogo você deseja desenvolver?**
 - Jogos 2D
 - Jogos 3D
 - VR (Realidade Virtual)
2. **Qual é o seu nível de experiência com desenvolvimento de jogos?**
 - Iniciante
 - Intermediário
 - Avançado
3. **Para quais plataformas você deseja desenvolver?**
 - PC
 - Console
 - Mobile
 - Multiplataforma (PC, Console, Mobile)
4. **Qual é o seu orçamento?**
 - Gratuito
 - Moderado
 - Alto
5. **Você tem alguma preferência por linguagem de programação?**
 - Não tenho preferência
 - Prefiro C#
 - Prefiro C++
 - Prefiro outra linguagem
6. **Quais são as características mais importantes para você?**
 - Desempenho gráfico
 - Facilidade de uso
 - Comunidade ativa

5.4 Lógica e Recomendações do Wizard

Com base nas respostas fornecidas pelo usuário, o Wizard recomenda uma das seguintes engines: Unity, Unreal Engine, Godot ou CryEngine. As recomendações são feitas considerando as características e funcionalidades de cada engine. Por exemplo, a Unity é recomendada para jogos 3D com uma curva de aprendizado razoável e suporte multiplataforma, enquanto a Unreal Engine é recomendada para desenvolvedores avançados que necessitam de gráficos realistas.

5.5 Prints da tela e interface do Wizard

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Qual tipo de jogo você deseja desenvolver?

Jogos 2D

Jogos 3D

VR (Realidade Virtual)

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Qual é o seu nível de experiência com desenvolvimento de jogos?

Iniciante

Intermediário

Avançado

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Para quais plataformas você deseja desenvolver?

PC

Console

Mobile

Multiplataforma (PC, Console, Mobile)

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Qual é o seu orçamento?

Gratuito

Moderado

Alto

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Você tem alguma preferência por linguagem de programação?

- Prefiro C#
- Prefiro C++
- Prefiro outra linguagem

Continuar

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Quais são as características mais importantes para você?

- Desempenho gráfico
- Facilidade de uso
- Comunidade ativa

Ver Recomendação

6. RESULTADOS E DISCUSSÕES

A análise de desempenho de Unity e Unreal Engine revela as capacidades e limitações dessas duas plataformas amplamente utilizadas no desenvolvimento de jogos. Ambas as engines possuem características distintas que as tornam adequadas para diferentes tipos de projetos, influenciando a escolha dos desenvolvedores com base nas necessidades específicas de seus jogos.

A Unity é conhecida por sua flexibilidade e suporte multiplataforma. Esta engine permite que desenvolvedores criem jogos para uma ampla gama de dispositivos, incluindo PCs, consoles, dispositivos móveis e sistemas de realidade virtual. Uma das maiores vantagens da Unity é a sua interface intuitiva, que facilita o aprendizado e o uso, especialmente para desenvolvedores iniciantes. A Unity Asset Store é outro ponto forte, oferecendo uma vasta biblioteca de ativos e plugins que podem acelerar significativamente o desenvolvimento de jogos.

No entanto, a Unity pode apresentar desafios de desempenho, especialmente em projetos que requerem gráficos complexos e intensivos. A necessidade de otimizações adicionais para alcançar tempos de renderização competitivos pode aumentar a carga de trabalho dos desenvolvedores. Apesar disso, a flexibilidade da Unity e sua capacidade de integrar uma variedade de ferramentas e recursos a tornam uma escolha popular para uma ampla gama de projetos, desde jogos independentes a títulos comerciais de médio porte.

Por outro lado, a Unreal Engine é amplamente reconhecida por suas capacidades gráficas avançadas e ferramentas robustas. Desenvolvida pela Epic Games, esta engine é frequentemente escolhida para jogos de alta fidelidade visual e projetos AAA. A Unreal Engine utiliza um sistema de renderização avançado que permite a criação de gráficos fotorrealistas, oferecendo ferramentas poderosas para simulação de física, efeitos visuais e iluminação.

Um dos principais atrativos da Unreal Engine é o Blueprints, um sistema de script visual que permite a criação de lógica de jogo sem necessidade de codificação extensa. Isso facilita o desenvolvimento para designers e artistas, permitindo a rápida prototipagem e implementação de ideias. Contudo, a Unreal Engine possui uma curva de aprendizado mais íngreme em comparação com a Unity, exigindo mais tempo e esforço para dominar suas ferramentas avançadas. Além disso, a Unreal Engine pode exigir hardware mais potente para alcançar seu pleno potencial, o que pode ser um obstáculo para desenvolvedores com recursos limitados.

A escolha entre Unity e Unreal Engine depende de vários fatores, incluindo o tipo de projeto, os objetivos do desenvolvedor e os recursos disponíveis. Para projetos que requerem alta flexibilidade e suporte a múltiplas plataformas, a Unity pode ser a melhor escolha. Sua interface amigável e a vasta comunidade de suporte oferecem uma base sólida para desenvolvedores de todos os níveis de experiência.

Por outro lado, para jogos que exigem gráficos de alta qualidade e complexidade visual, a Unreal Engine oferece ferramentas superiores e um desempenho gráfico robusto. A capacidade de criar efeitos visuais impressionantes e ambientes detalhados faz da Unreal Engine a escolha preferida para jogos de alta produção e títulos AAA.

Em termos de desempenho técnico, a Unreal Engine tende a superar a Unity em projetos que demandam gráficos intensivos e simulações complexas. No entanto, a Unity continua a ser altamente competitiva em projetos que valorizam a flexibilidade e a facilidade de desenvolvimento.

Diversos estudos têm sido realizados para avaliar o desempenho de diferentes engines de jogos, considerando fatores como eficiência de processamento, qualidade gráfica, facilidade de uso e versatilidade. Esses estudos são essenciais para ajudar desenvolvedores a escolher a engine mais adequada para seus projetos, fornecendo uma compreensão detalhada das capacidades e limitações de cada plataforma.

Šmíd (2017), em seu estudo de comparação entre Unity e Unreal Engine, analisou aspectos como tempo de renderização, uso de CPU e GPU, e qualidade gráfica. Os resultados indicaram que a Unreal Engine oferece um desempenho superior em termos de gráficos fotorrealistas e efeitos visuais avançados, enquanto a Unity se destacou pela flexibilidade e eficiência em projetos multiplataforma. A pesquisa também apontou que a Unreal Engine, apesar de seu desempenho gráfico superior, possui uma curva de aprendizado mais íngreme devido à sua complexidade.

Outro estudo significativo é o de Cavalcante e Pereira (2018), que compararam várias engines de jogos, incluindo Unity e Unreal Engine, com foco no desenvolvimento de um jogo de educação financeira. O estudo revelou que a Unity e a Unreal Engine fornecem ferramentas poderosas para a criação de jogos 3D com gráficos avançados, mas a Godot se destacou pela simplicidade e eficiência no desenvolvimento de jogos 2D. A pesquisa destacou que a Godot é uma opção atraente para desenvolvedores independentes e projetos menores devido à sua natureza open-source e interface intuitiva.

Scherer, Batista e Mendes (2020) conduziram uma análise detalhada da evolução das engines de jogos, examinando como diferentes engines têm se desenvolvido ao longo dos anos para atender às crescentes demandas dos jogos modernos. O estudo apontou que é particularmente eficaz na criação de ambientes altamente detalhados e realistas, graças ao seu avançado sistema de renderização e simulação de física.

Alves (2021) realizou um estudo comparativo entre engines de desenvolvimento de jogos 2D, incluindo Unity. O estudo focou na eficiência de desenvolvimento, facilidade de uso e desempenho em plataformas móveis. Alves concluiu que a Unity, embora mais complexa, oferece uma maior versatilidade e suporte a uma ampla gama de plataformas.

Fernandes et al. (2015) exploraram a aplicação de inteligência artificial em jogos, utilizando diferentes engines para avaliar como cada uma lida com algoritmos complexos e simulações de comportamento. O estudo mostrou que a Unreal Engine e a Unity oferecem robustos suportes para a integração de IA, com a Unreal Engine se destacando em simulações

mais complexas devido às suas poderosas ferramentas de desenvolvimento e capacidade de processamento.

Além disso, O'Hagan, Coleman e O'Connor (2014) realizaram uma revisão sistemática dos processos de desenvolvimento de software para jogos, incluindo a análise de diferentes engines. Eles descobriram que a escolha da engine impacta significativamente a produtividade da equipe de desenvolvimento, a qualidade do jogo final e o tempo de desenvolvimento. Engines com boas ferramentas de depuração, documentação detalhada e comunidades ativas de desenvolvedores, como a Unity e a Unreal Engine, tendem a facilitar um desenvolvimento mais eficiente e menos propenso a erros.

Os estudos anteriores fornecem uma visão abrangente das capacidades e limitações das principais engines de jogos. A Unity é amplamente reconhecida por sua versatilidade e suporte a múltiplas plataformas, enquanto a Unreal Engine se destaca por suas capacidades gráficas avançadas e ferramentas robustas.

A avaliação do desempenho técnico das engines de jogos é crucial para determinar sua adequação a diferentes tipos de projetos e suas capacidades de atender às demandas dos desenvolvedores e dos jogadores. Entre os principais critérios de desempenho técnico estão o tempo de renderização, o uso de CPU e GPU, e a latência. Estes fatores afetam diretamente a qualidade visual do jogo, a eficiência do desenvolvimento e a experiência do usuário.

O tempo de renderização refere-se ao tempo necessário para gerar cada frame de um jogo. Este aspecto é fundamental para garantir uma experiência de jogo fluida e responsiva. Uma menor duração de renderização implica uma taxa de frames por segundo (FPS) mais alta, resultando em uma jogabilidade mais suave.

Estudos mostram que a Unreal Engine, com seu sistema de renderização avançado, frequentemente oferece tempos de renderização menores para gráficos complexos em comparação com outras engines, especialmente quando se trata de gráficos fotorrealistas e efeitos visuais intensivos (Cavalcante e Pereira, 2018). Por outro lado, a Unity, apesar de sua flexibilidade, pode requerer otimizações adicionais para alcançar tempos de renderização equivalentes, especialmente em projetos gráficos intensivos (Šmíd, 2017).

O uso eficiente dos recursos do sistema, particularmente da CPU (unidade central de processamento) e da GPU (unidade de processamento gráfico), é outro critério essencial na avaliação do desempenho técnico das engines. A eficiência no uso da CPU e GPU determina a capacidade da engine de manter altos níveis de desempenho gráfico e processamento sem sobrecarregar o sistema.

A Unreal Engine é conhecida por seu uso otimizado da GPU, o que permite a criação de gráficos detalhados e efeitos visuais complexos sem comprometer o desempenho. A engine utiliza técnicas avançadas de sombreamento e renderização, aproveitando ao máximo o hardware disponível (Scherer, Batista e Mendes, 2020). A Unity, embora também eficiente, pode apresentar variabilidade no uso de recursos dependendo das especificidades do projeto e da experiência do desenvolvedor na otimização de código e ativos (Šmíd, 2017).

Latência refere-se ao atraso entre a entrada do usuário e a resposta do jogo. Este fator é crítico para a experiência do jogador, especialmente em jogos que exigem respostas rápidas, como jogos de ação e esportes eletrônicos. Uma baixa latência é essencial para garantir que as ações do jogador sejam refletidas instantaneamente no jogo.

Estudos indicam que a Unreal Engine e a Unity possuem mecanismos eficientes para minimizar a latência, através de otimizações de código e técnicas avançadas de renderização. A Unreal Engine, em particular, oferece ferramentas integradas para monitorar e reduzir a latência, garantindo uma resposta rápida às ações do jogador (Fernandes et al., 2015). A Unity também permite a implementação de técnicas de otimização para reduzir a latência, embora a eficácia dessas otimizações possa variar conforme a complexidade do jogo e a habilidade do desenvolvedor em utilizar as ferramentas da engine (Šmíd, 2017).

Em conclusão, tanto a Unity quanto a Unreal Engine possuem forças e fraquezas distintas. Desenvolvedores devem considerar cuidadosamente suas necessidades específicas, o tipo de jogo que desejam criar, seu orçamento e suas habilidades técnicas ao escolher a engine mais adequada para seu projeto. Ambas as engines têm demonstrado ser ferramentas poderosas no desenvolvimento de jogos, contribuindo significativamente para a indústria de jogos eletrônicos.

6.1 Complemento à Análise de Desempenho das Engines

O Wizard complementa a análise de desempenho das engines ao oferecer uma ferramenta prática para os desenvolvedores tomarem decisões informadas. Ele simplifica o processo de escolha, economizando tempo e esforço. A lógica de decisão implementada no Wizard reflete a análise detalhada das características e funcionalidades de cada engine, proporcionando recomendações baseadas em critérios objetivos.

Benefícios

- **Acessibilidade:** Pode ser acessado facilmente via dispositivos móveis.
- **Facilidade de Uso:** Interface intuitiva que guia o usuário através do processo.
- **Personalização:** Recomendações personalizadas com base nas necessidades específicas dos usuários.

Limitações

- **Generalização:** As recomendações são baseadas em critérios gerais e podem não considerar necessidades específicas não abordadas pelas perguntas.
- **Atualização:** As engines de desenvolvimento de jogos estão em constante evolução, exigindo atualizações periódicas do Wizard.

6.2 Recomendações do Wizard

Para avaliar a eficácia do Wizard desenvolvido para a escolha de engines de desenvolvimento de jogos, foram realizados testes com diferentes perfis de usuários e cenários de desenvolvimento. A seguir, são apresentados os resultados e as recomendações fornecidas pelo Wizard.

6.3 Print da Tela: Recomendação Final

Wizard de Escolha de Engine de Desenvolvimento de Jogos

Recomendação de Engine

Recomendamos usar a Unreal Engine, pois ela oferece gráficos realistas e recursos avançados, possui uma comunidade ativa e muitos recursos disponíveis. No entanto, sua curva de aprendizado é mais íngreme. A Unreal Engine oferece suporte multiplataforma (PC, console, VR) e sua linguagem de programação principal é o C++.

Os resultados demonstram que o Wizard é capaz de fornecer recomendações precisas e personalizadas com base nas respostas dos usuários. As engines recomendadas incluem Unity, Unreal Engine, Godot e CryEngine, cada uma selecionada de acordo com as necessidades específicas de desenvolvimento.

6.4 Importância do Wizard

O Wizard de Escolha de Engine de Desenvolvimento de Jogos representa uma contribuição significativa para o campo do desenvolvimento de jogos, facilitando a tomada de decisões informadas e otimizando o processo de escolha de ferramentas. Através deste Wizard, os desenvolvedores podem encontrar rapidamente a engine que melhor atende às suas necessidades específicas, baseando-se em uma lógica clara e objetiva.

6.5 Insights Obtidos

A implementação do Wizard demonstrou a importância de ferramentas auxiliares no desenvolvimento de jogos, destacando como uma abordagem sistemática pode simplificar processos complexos. Além disso, reforçou a necessidade de considerar múltiplos fatores na escolha da engine, proporcionando uma visão abrangente e detalhada das opções disponíveis.

7. CONCLUSÃO

A comparação entre Unity e Unreal Engine revela as distintas capacidades e potenciais dessas duas poderosas ferramentas de desenvolvimento de jogos. Ambas as engines têm demonstrado ser instrumentos cruciais no setor de jogos eletrônicos, cada uma com suas próprias forças que atendem a diferentes tipos de projetos e necessidades dos desenvolvedores.

A Unity destaca-se pela sua flexibilidade, interface amigável e suporte multiplataforma, facilitando o desenvolvimento tanto para iniciantes quanto para desenvolvedores experientes. Sua vasta biblioteca de ativos e plugins na Unity Asset Store acelera o processo de criação, tornando-a uma escolha popular para uma ampla gama de projetos, desde jogos independentes a títulos

comerciais de médio porte. No entanto, a Unity pode requerer otimizações adicionais para atingir um desempenho gráfico competitivo em jogos mais complexos.

Por outro lado, a Unreal Engine é conhecida por suas capacidades gráficas avançadas e ferramentas robustas, sendo ideal para jogos de alta fidelidade visual e projetos AAA. O sistema de script visual Blueprints facilita o desenvolvimento para designers e artistas, permitindo a rápida prototipagem e implementação de ideias sem a necessidade de programação extensiva. Embora a Unreal Engine tenha uma curva de aprendizado mais íngreme e possa exigir hardware mais potente, ela oferece um desempenho gráfico superior e ferramentas avançadas para a criação de efeitos visuais impressionantes e ambientes detalhados.

A escolha entre Unity e Unreal Engine depende de vários fatores, incluindo o tipo de projeto, os objetivos do desenvolvedor e os recursos disponíveis. Para projetos que requerem alta flexibilidade e suporte a múltiplas plataformas, a Unity pode ser a melhor escolha. Para jogos que exigem gráficos de alta qualidade e complexidade visual, a Unreal Engine oferece ferramentas superiores e um desempenho gráfico robusto.

Em termos de desempenho técnico, a Unreal Engine tende a superar a Unity em projetos que demandam gráficos intensivos e simulações complexas. No entanto, a Unity continua a ser altamente competitiva em projetos que valorizam a flexibilidade e a facilidade de desenvolvimento.

Em resumo, tanto a Unity quanto a Unreal Engine possuem vantagens distintas que as tornam adequadas para diferentes tipos de desenvolvedores e projetos. Desenvolvedores devem considerar cuidadosamente suas necessidades específicas, o tipo de jogo que desejam criar, seu orçamento e suas habilidades técnicas ao escolher a engine mais adequada. Ambas as engines têm um papel vital na indústria de jogos eletrônicos, proporcionando as ferramentas necessárias para criar experiências de jogo envolventes e inovadoras.

Este trabalho contribuiu significativamente para o campo de desenvolvimento de jogos ao proporcionar uma análise detalhada e comparativa das engines Unity e Unreal Engine, destacando suas características, funcionalidades e desempenho técnico. Ao explorar aspectos como tempo de renderização, uso de CPU e GPU, latência, e a experiência do usuário, forneceu-se uma base sólida para que desenvolvedores façam escolhas informadas sobre a engine mais adequada para seus projetos específicos. Além disso, ao considerar o feedback de desenvolvedores e a usabilidade prática dessas ferramentas, o estudo ajudou a identificar melhores práticas e estratégias de otimização, contribuindo para o aprimoramento do processo de desenvolvimento de jogos.

No entanto, o estudo apresentou algumas limitações que devem ser reconhecidas. Uma das principais limitações foi a dependência de fontes secundárias, como artigos acadêmicos, relatórios técnicos e entrevistas, que podem conter vieses ou não abranger todas as variáveis relevantes. Além disso, o estudo focou-se principalmente nas capacidades técnicas das engines, podendo não capturar plenamente a complexidade das experiências individuais dos desenvolvedores. A diversidade dos projetos de jogos, variando em gênero, escala e requisitos técnicos, também representa um desafio, pois as conclusões podem não ser universalmente aplicáveis a todos os tipos de desenvolvimento de jogos.

Para pesquisas futuras, sugere-se a realização de estudos empíricos mais aprofundados que incluam testes práticos com desenvolvedores utilizando ambas as engines em projetos reais. Esses estudos poderiam fornecer dados mais concretos sobre o desempenho das engines em diferentes

cenários de desenvolvimento. Além disso, a exploração de novas engines emergentes e suas comparações com Unity e Unreal Engine seria valiosa para acompanhar as evoluções tecnológicas e as tendências no campo de desenvolvimento de jogos.

A inclusão de uma análise mais detalhada sobre o impacto econômico e a viabilidade financeira de utilizar diferentes engines também poderia enriquecer futuras pesquisas, oferecendo uma visão mais completa das implicações práticas de cada escolha para desenvolvedores e estúdios de jogos.

REFERÊNCIAS

ALVES, Gelderson Bezerra. Estudo comparativo entre engines de desenvolvimento de jogos 2D / Gelderson Bezerra Alves. – 2021. 43 f.

AMARO, Mariana. Interatividade e Novas Mídias: Os Jogos Digitais na Comunicação. In: Anais Eletrônicos do VIII Simpósio Nacional da Associação Brasileira de Pesquisadores em Cibercultura (ABCiber). São Paulo: ESPM, 2014.

CAVALCANTE, C. H. L.; PEREIRA, M. L. A. Comparativo entre Game Engines como Etapa Inicial para o Desenvolvimento de um Jogo de Educação Financeira. [S.l.: S.n], 2018.

COVALESKI, R. L.; SIQUEIRA, O. Publicidade em Jogos Sociais: engajamento digital a partir de socialização, competição e cooperação. Curitiba, v. 18. n.2. p. 39-52, jul./dez. 2014.

FERNANDES, A. M.; CAMARA, B. H. P.; PASCHOAL, A. R.; DAMASCENO, E. F. Inteligência artificial aplicada a jogos de tabuleiro com realidade aumentada. In: 15º Workshop de Realidade Virtual Aumentada, Marília-SP. [S.l.: s.n.], 2015. v. 1.

FERNANDES, D. R. Jogos digitais e história: Uma experiência interdisciplinar e multidisciplinar. Redin-Revista Educacional Interdisciplinar, [S.l.], v. 7, n. 1, 2018.

GREGORY, J. Game engine architecture. [S.l.]: AK Peters/CRC Press, 2017.

JACQUES, Edu. Glitchbusting: a Exposição de Glitches em Jogos Digitais como (Contra)Experiência Gamer. In: Anais Eletrônicos do VIII Simpósio Nacional da Associação Brasileira de Pesquisadores em Cibercultura (ABCiber). São Paulo: ESPM, 2014.

MARTINO, Luís Mauro Sá. Teoria da Mídias Digitais: Linguagens, Ambientes, Redes. Petrópolis, RJ: Vozes, 2014.

MEDEIROS, J.F. ADVERGAMES: A publicidade em jogos digitais como forma de atrair o consumidor. VIII Brazilian Symposium on Games and Digital Entertainment. Rio de Janeiro, RJ – Brazil, October 8th-10th 2019.

O’HAGAN, A. O.; COLEMAN, G.; O’CONNOR, R. V. Software development processes for games: A systematic literature review. In: SPRINGER. European Conference on Software Process Improvement. [S.l.], 2014. p. 182–193.

PERANI, Letícia. Jogando para Comunicar, Comunicando para Jogar: por Um Lugar dos Games nas Ciências da Comunicação e na Cibercultura. In: Anais Eletrônicos do VIII Simpósio Nacional da Associação Brasileira de Pesquisadores em Cibercultura (ABCiber). São Paulo: ESPM, 2014.

ROGERS, S. Level Up! The guide to great video game design. [S.l.]: John Wiley & Sons, 2014.

SCHERER, Daniel; BATISTA, Daniele Ventura; MENDES, Aline de Cantalice. Análise da Evolução de Engines de Jogos. In: CONGRESSO SOBRE TECNOLOGIAS NA EDUCAÇÃO (CTRL+E), 5. , 2020, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 425-434.

SERRANO, Rafael Meneguzzi. Streaming media: Twitch.tv e a percepção dos usuários em relação a plataforma. 2015. 89 f. Monografia (Bacharel em Publicidade e Propaganda). Curso de Publicidade e Propaganda. Universidade de Passo Fundo, Passo Fundo, RS, 2015.

SILVA, D. L. da. Análise de ferramentas e desenvolvimento de jogo para treinamento de paratletas. [S.l: s.n], 2018.

SILVA, R. E. da; MARTINS, S. W. Ensino de ciência da computação através do desenvolvimento de jogos. In: VII Congresso Iberoamericano de Informática Educativa. [S.l.: s.n.], 2015. p. 1286–1295.

ŠMÍD, A. Comparison of Unity and Unreal Engine. Bachelor Thesis. Praga, República Checa: DCGI/Faculty of Electrical Engineering, 2017.

SMITH, M.; TELANG, R. Streaming, sharing, stealing. Big data and the future of entertainment. Cambridge, Massachussets: The MIT Press, 2016.