

DO CONTROLE DE BAIXO NÍVEL À VERSATILIDADE MULTIPLATAFORMA: O PODER DO C++ E C# NO DESENVOLVIMENTO DE JOGOS

**Maria Flávia da Silva Ribeiro¹, Anna Patricia Zakem China¹,
Débora Pelicano Diniz¹**

¹Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)
Ribeirão Preto, SP – Brasil

maria.ribeiro20@fatec.sp.gov.br,
anna.china@fatec.sp.gov.br,
debora.diniz2@fatec.sp.gov.br,

Resumo. *O uso das linguagens de programação C++ e C# é fundamental no desenvolvimento de jogos, apresentando vantagens específicas para diferentes plataformas. C++ se destaca por seu alto desempenho e é preferido em jogos para PC e consoles, como Xbox One. Por outro lado, o C# é mais orientado a objetos, favorece o desenvolvimento rápido e é comumente utilizado em jogos móveis, integrando-se bem com plataformas como Unity e Xamarin. Ambas as linguagens possuem suas vantagens distintas: C++ enfoca o desempenho e o controle de hardware, enquanto C# prioriza a facilidade de uso e a adaptabilidade a jogos móveis. O objetivo deste estudo é analisar as capacidades das linguagens C++ e C# no desenvolvimento de jogos, destacando seus pontos fortes e fracos, suas aplicações práticas e tendências futuras. Além disso, o trabalho discute as principais bibliotecas, frameworks e ferramentas que auxiliam no uso dessas linguagens na criação de jogos. Para alcançar esse objetivo, este estudo baseou-se em uma revisão criteriosa de literatura especializada, incluindo fontes como artigos científicos, livros, sites especializados e documentações oficiais. A metodologia abrangeu a coleta de informações relevantes sobre as características, vantagens e desvantagens de C++ e C# no contexto específico do desenvolvimento de jogos, bem como a análise detalhada de exemplos de jogos desenvolvidos com essas linguagens. Esta abordagem possibilitou um panorama abrangente e embasado para a compreensão das capacidades e usos mais adequados de C++ e C# na indústria de jogos.*

Abstract. *The use of programming languages such as C++ and C# is pivotal in game development, offering specific advantages for different platforms. C++ stands out for its high performance and is preferred in games for PC and consoles like Xbox One. On the other hand, C# is more object-oriented, facilitating rapid development and is commonly used in mobile games, integrating well with platforms such as Unity and Xamarin. Both languages have their distinct advantages: C++ emphasizes performance and hardware control, while C# prioritizes ease of use and adaptability to mobile gaming. The aim of this study is to analyze the capabilities of C++ and C# languages in game development, highlighting their strengths, weaknesses, practical*

applications, and future trends. Moreover, this work discusses the primary libraries, frameworks, and tools that assist in utilizing these languages in game creation. To achieve this goal, this study relied on a comprehensive review of specialized literature, including sources such as scientific articles, books, specialized websites, and official documentations. The methodology encompassed gathering relevant information on the characteristics, advantages, and disadvantages of C++ and C# in the specific context of game development, along with a detailed analysis of game examples developed using these languages. This approach provided a comprehensive and well-founded overview for understanding the most suitable capabilities and uses of C++ and C# in the gaming industry.

1. Introdução

O desenvolvimento de jogos é uma área que atrai muitos programadores, pois envolve criatividade, desafio e diversão. Além disso, o mercado de jogos é um dos que mais cresce e lucra no mundo, oferecendo oportunidades para profissionais qualificados. Por isso, é importante conhecer as ferramentas e tecnologias que permitem criar jogos de qualidade e que atendam às expectativas dos jogadores. As linguagens C++ e C# são duas das mais utilizadas pelos desenvolvedores de jogos, pois oferecem recursos avançados, flexibilidade e compatibilidade com diversas plataformas. No entanto, elas também possuem diferenças significativas que podem influenciar na escolha da melhor linguagem para cada projeto

O objetivo do texto é analisar a capacidade das linguagens C++ e do C# no desenvolvimento de jogos, destacando seus pontos fortes e fracos, suas aplicações práticas e suas tendências futuras. O trabalho também irá discutir as principais bibliotecas, *frameworks* e ferramentas que auxiliam o uso dessas linguagens na criação de jogos.

Este trabalho é dividido em 6 seções. A seção a seguir apresenta o referencial teórico no qual foi feito o levantamento de informações para apoiar o desenvolvimento dessa pesquisa. Em seguida, é apresentada a metodologia, onde constam as informações de como foi feita a escolha do material e como ele foi analisado. A seguir, a seção de resultados traz as informações recolhidas através da metodologia e mostra se os objetivos foram alcançados, na sequência, são feitas as considerações finais onde o autor coloca algumas opiniões, dificuldades e encaminhamentos futuros e por fim as referências bibliográficas na qual o autor se baseou para a realização deste trabalho.

2. Referencial Teórico

Segundo a Associação Brasileira de Desenvolvedores de Jogos Digitais (Abragames), o mercado de games é muito expressivo porque o setor faturou aproximadamente R\$ 5,9 bilhões em 2019, o que representa um aumento de 13% em comparação com o ano anterior. Esse crescimento é impulsionado por vários fatores, como o aumento da penetração de dispositivos móveis, a expansão da indústria de e-sports e a crescente aceitação de games como forma de entretenimento (RAMIRES, 2023).

O C++ é uma linguagem de programação de alto nível que surgiu como uma extensão do C, adicionando recursos de orientação a objetos e melhorias de sintaxe. Foi desenvolvido por Bjarne Stroustrup na década de 1980, visando proporcionar controle de baixo nível e alto desempenho, especialmente para o desenvolvimento de software de

sistemas e jogos. Desde então, o C++ se tornou uma das linguagens mais populares e amplamente usadas na indústria de desenvolvimento de jogos (MICROSOFT, 2023).

O C# (C Sharp) é uma linguagem de programação moderna desenvolvida pela Microsoft no início dos anos 2000. Foi projetada para ser simples, segura, orientada a objetos e multiplataforma, com ênfase na facilidade de uso e produtividade. O C# é amplamente utilizado no desenvolvimento de jogos, especialmente em conjunto com o engine Unity (NOLETO.C., 2021)

Os primeiros games inventados tinham aparência bastante simples: programados para rodar em fliperamas, consoles com apenas um jogo e computadores bem menos modernos que os atuais, costumavam ser programados em linguagem Assembly, uma linguagem de baixo nível que possibilitou a criação de jogos como o famoso Pong, do Atari, e que abriu espaço para o desenvolvimento de jogos utilizando a linguagem C, seguida da C++ e posteriormente do C#. Outro fator que cooperou com a popularização do C++ e da linguagem C#, que abordaremos adiante, foi a utilização dos motores de jogo (ZAMBALDI, 2021).

Os motores de jogo, ou game engines, são bibliotecas ou softwares responsáveis por colaborar com o processo de desenvolvimento através do fornecimento de diversos facilitadores, como um motor gráfico que possibilita a renderização em tempo real do jogo, simulador de física, que confere mais realismo físico aos movimentos e ações dos personagens e elementos do cenário, entre outras funções. Com a chegada dos motores de jogo, os games ficaram cada vez mais realistas, com gráficos em altíssima resolução e qualidade, movimentos realistas e renderização 2 e 3D (ZAMBALDI, 2021).

2.1 As Linguagem C++ e C#

O C++ é uma evolução do C que se apresentou como uma versão moderna da linguagem. Uma das diferenças é que é um padrão orientado a objetos, mais típico da atualidade, e concorreu especificamente com o fenômeno Java. O C++ traz diferenças no nível de programação, que já é considerado mais alto para facilitar a vida das pessoas que programam (SACRAMENTO, 2021).

O C# é o que chamamos de linguagem fortemente tipada. Isso significa que seus tipos de dados são fixos e que não são alteráveis com facilidade. Em uma melhor explicação: se a gente define uma variável como numérica, não será possível atribuir caracteres a ela em outro ponto do código. O C# se destaca por ser uma opção tecnológica viável para diferentes projetos de variados tipos (SACRAMENTO, 2021).

Na figura abaixo será demonstrado as top 10 das linguagens de programação que mais foram utilizadas ao longo do tempo até o ano de 2022. Pode se notar que a linguagem de C# está em 8º lugar e seguiu aumentando até chegar na 5ª posição, já a linguagem de C++ estava em 6º lugar e por um tempo se manteve na 5ª posição, entretanto teve uma pequena oscilação e por fim continuou no seu 6º lugar como no início.

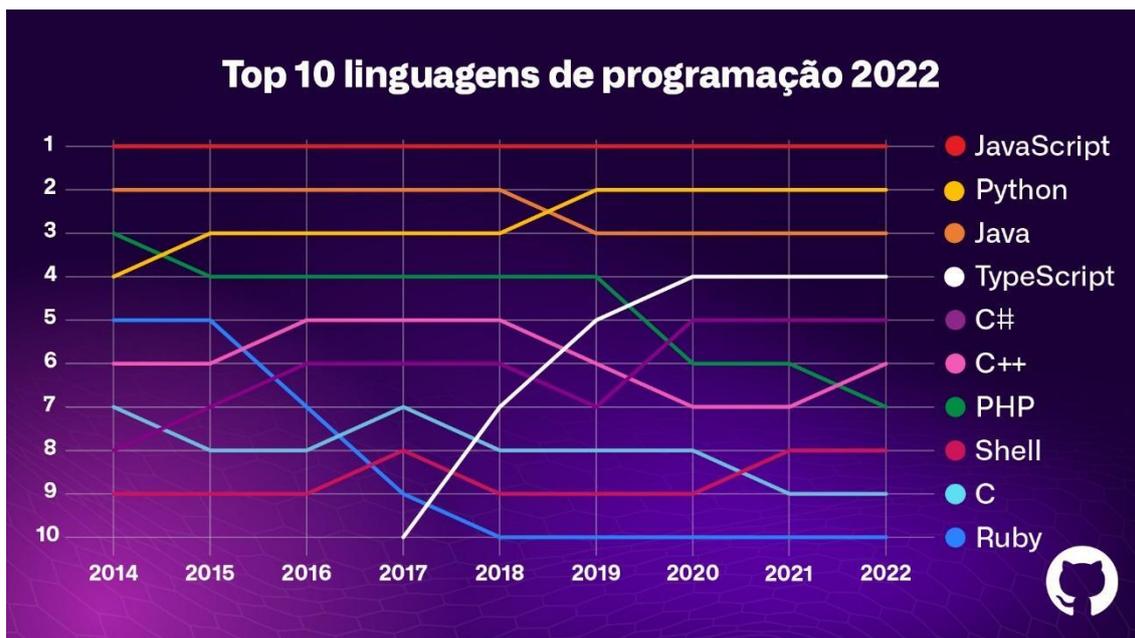


Figura 1. Confira quais foram as linguagens de programação mais usadas em 2022
Fonte: (Imasters, 2023)

3. Materiais e Métodos

O texto será baseado em uma revisão bibliográfica de fontes confiáveis, como artigos científicos, livros, sites especializados e documentações oficiais. O desenvolvimento irá abordar os seguintes tópicos:

- Linguagem e conceitos básicos do C++ e do C#;
- Vantagens e desvantagens do C++ e do C# no desenvolvimento de jogos;
- Exemplos de jogos desenvolvidos com C++ e com C#;
- Bibliotecas, *frameworks* e ferramentas para o desenvolvimento de jogos com C++ e com C#;

3.1. Linguagem C++

O C++ introduziu diversos conceitos importantes que o diferenciam do C e o tornam uma linguagem poderosa para o desenvolvimento de jogos. Alguns desses conceitos incluem:

- Orientação a objetos:** O C++ oferece suporte completo à programação orientada a objetos, permitindo a criação de classes, objetos, herança, polimorfismo e encapsulamento. Esses recursos são essenciais para a organização e reutilização de código em projetos de jogos complexos.
- Templates:** Os templates do C++ fornecem um mecanismo para criar algoritmos e estruturas de dados genéricas, permitindo que os desenvolvedores escrevam código flexível e eficiente que pode ser reutilizado em diferentes contextos de jogo.
- Gerenciamento de memória:** O C++ permite um controle preciso da alocação e desalocação de memória, o que é particularmente importante em jogos, onde o desempenho e a eficiência são essenciais. Os desenvolvedores podem usar recursos como alocação no heap e stack, bem como gerenciamento manual de ponteiros.

3.2. A linguagem C#

O C# compartilha muitos conceitos com outras linguagens orientadas a objetos, mas também possui recursos exclusivos que o tornam uma escolha popular para o desenvolvimento de jogos. Alguns desses conceitos incluem:

- a) **Orientação a objetos:** Segue os princípios da programação orientada a objetos, permitindo a criação de classes, objetos, herança, polimorfismo e encapsulamento. Esses conceitos são fundamentais para a estruturação e organização do código em projetos de jogos.
- b) **Garbage Collection:** O C# usa um sistema de coleta de lixo (*garbage collector*) para gerenciar automaticamente a memória alocada para objetos. Isso simplifica o gerenciamento de memória para os desenvolvedores, permitindo que eles se concentrem no desenvolvimento do jogo em vez de se preocuparem com a alocação e desalocação de memória.
- c) **Bibliotecas e Frameworks:** O C# possui um ecossistema robusto de bibliotecas e *frameworks*, como o Unity Engine, que facilitam o desenvolvimento de jogos multiplataforma. Essas ferramentas fornecem uma ampla gama de recursos e funcionalidades para acelerar o processo de criação de jogos, desde a criação de gráficos e animações até a implementação de física realista e lógica de jogo.

4 Resultados e Discussões

C++ e C# são duas linguagens amplamente utilizadas no desenvolvimento de jogos, cada uma com suas próprias vantagens e desvantagens. É importante considerar esses aspectos ao escolher a melhor opção para um projeto de desenvolvimento de jogos. Na Tabela 1 estão apresentadas algumas das vantagens e desvantagens de cada linguagem.

É importante considerar essas vantagens e desvantagens ao escolher entre C++ e C# para o desenvolvimento de jogos. Cada linguagem tem suas próprias características e é mais adequada para diferentes cenários. Enquanto C++ oferece maior controle e desempenho máximo, C# se destaca pela facilidade de uso, produtividade e integração com o Unity Engine. A escolha entre as duas depende das necessidades específicas do projeto, do desempenho desejado, da disponibilidade de recursos e da experiência da equipe de desenvolvimento.

Tabela 1. Vantagens e Desvantagens do C# e C++ no desenvolvimento de jogos.

LINGUAGEM C++		LINGUAGEM C#	
VANTAGENS	DESADVANTAGENS	VANTAGENS	DESADVANTAGENS
C++ permite um controle preciso sobre o hardware e recursos do sistema, o que é essencial para otimização de desempenho e criação de jogos que exigem máximo controle.	Em C++, os desenvolvedores têm a responsabilidade de gerenciar manualmente a alocação e deslocação de memória, o que pode levar a erros como vazamentos de memória e acesso inválido.	C# possui uma sintaxe mais simples e fácil de entender, o que torna a linguagem mais acessível para desenvolvedores iniciantes.	Em comparação com C++, C# pode ter um desempenho ligeiramente inferior devido ao seu nível de abstração e gerenciamento automático de memória.

Devido ao seu controle de baixo nível, C++ oferece uma performance excepcionalmente alta, permitindo a criação de jogos com gráficos avançados e física complexa.	Devido ao seu controle de baixo nível e complexidade, o desenvolvimento de jogos em C++ pode levar mais tempo em comparação com outras linguagens de mais alto nível.	C# tem uma forte integração com o Unity Engine, um dos motores de jogos mais populares, fornecendo recursos poderosos e uma ampla comunidade de suporte.	C# oferece menos controle direto sobre o hardware e recursos do sistema em comparação com menor controle de baixo nível. Isso significa que certas otimizações de desempenho e personalizações específicas podem ser mais difíceis de implementar em C#.
C++ possui uma vasta gama de bibliotecas e <i>frameworks</i> disponíveis, oferecendo aos desenvolvedores um grande leque de opções para aprimorar seus jogos.		C# utiliza um sistema de coleta de lixo (garbage collector) que gerencia automaticamente a alocação e desalocação de memória, aliviando os desenvolvedores dessa tarefa.	Embora C# seja altamente integrado com o Unity Engine, isso também significa que seu uso pode ser limitado a esse ambiente específico. Isso pode restringir a portabilidade do jogo para outras plataformas ou motores de jogo.
C++ permite uma alta flexibilidade e personalização no desenvolvimento de jogos, permitindo a implementação de recursos específicos e otimização personalizada.		Devido à sua facilidade de uso e ao ambiente integrado de desenvolvimento do Unity, C# permite um desenvolvimento mais rápido e iterativo de jogos.	Comparado ao C++, o C# oferece menos controle direto sobre o hardware e os recursos de baixo nível. Isso pode ser uma desvantagem em projetos que exigem otimizações específicas de desempenho ou acesso direto a componentes de hardware.

Fonte: (Autoria própria, 2023)

4.1. Exemplos de jogos desenvolvidos com C++ e com C#

Use C++ para criar jogos de alta qualidade, como os jogos famosos: Fortnite, Mortal Kombat, e Counter Strike, devido ao seu alto desempenho e controle especializado:

- a) **Fortnite:** Desenvolvido pela Epic Games, "Fortnite" utiliza a Unreal Engine, que é uma engine de jogos conhecida por usar C++ como uma das principais linguagens de programação para desenvolver jogabilidade, gráficos e outros sistemas do jogo (MARCHIORI.L,2022).
- b) **Mortal Kombat 11:** A NetherRealm Studios desenvolveu "Mortal Kombat 11" utilizando C++. A série "Mortal Kombat" é famosa por seus combates intensos e detalhes gráficos, o que requer um alto desempenho, tornando o C++ uma escolha natural para o desenvolvimento (MARCHIORI.L,2022).
- c) **Counter Strike:** Essa linguagem foi desenvolvida por Bjarne Stroustrup a partir de 1979, como um aprimoramento da linguagem C. Foi originalmente chamado C com Classes, mas depois foi renomeado C++ em 1983 (MARCHIORI.L,2022).

Experimente usar C# para criar jogos multiplataforma e independentes, Escape from Tarkov, Cities: Skylines e Hearthstone, aproveitando sua presença marcante e flexibilidade no desenvolvimento de jogos

- d) **Escape from Tarkov:** Desenvolvido pela Battlestate Games, "Escape from Tarkov" utiliza a engine Unity e é amplamente conhecido por empregar a linguagem C# para a criação de jogabilidade, sistemas e interações do jogo (CLARK.M.,2022).
- e) **Cities: Skylines:** Desenvolvido pela Colossal Order, "Cities: Skylines" também utiliza a engine Unity e é construído principalmente com C#. O jogo é um simulador de construção e gerenciamento de cidades que se tornou muito popular devido à sua jogabilidade envolvente (CLARK.M.,2022).
- f) **Hearthstone:** Este popular jogo de cartas online desenvolvido pela Blizzard Entertainment também usa C#. O jogo apresenta uma mecânica estratégica viciante e foi lançado em várias plataformas, incluindo PC, dispositivos móveis e consoles (MARCHIORI.L,2022).

Esses são apenas alguns exemplos notáveis de jogos desenvolvidos tanto com C++ quanto com C#. Essas linguagens são usadas em uma ampla variedade de jogos, desde grandes produções até jogos indie, oferecendo aos desenvolvedores as ferramentas necessárias para criar experiências de jogo envolventes e diversificadas.

4.2. Bibliotecas, *frameworks* e ferramentas para o desenvolvimento de jogos com C++ e com C#

A seção a seguir descreve um pouco sobre o desenvolvimento de jogos que é uma área empolgante e desafiadora da programação de computadores, e uma parte fundamental desse processo é a escolha das ferramentas certas para a criação de jogos eficientes e envolventes.

Tanto C++ quanto C# possuem uma ampla gama de bibliotecas, *frameworks* e ferramentas disponíveis para facilitar o desenvolvimento de jogos. Essas ferramentas fornecem funcionalidades adicionais, aceleram o processo de desenvolvimento e ajudam os desenvolvedores a criarem jogos de alta qualidade. A seguir, estão algumas das principais opções para cada linguagem.

C++:

- a) **Unreal Engine:** é tido como o grande concorrente do Unity 3D e, assim como ele, permite o desenvolvimento em 3D e 2D. Devido à sua estrutura, acaba sendo uma ferramenta mais voltada para times e projetos de games mais complexos. Também por isso, os jogos possuem visual e desempenho superior ao visto no Unity 3D, especialmente quando temos dispositivos com maior poder de processamento (HENRIQUE, 2017).
- b) **DirectX:** é uma coleção de APIs (Interfaces de Programação de Aplicativos) da Microsoft que permite o desenvolvimento de jogos para Windows. Ele oferece suporte para gráficos avançados, áudio, entrada de dispositivos e muito mais, tudo isso usando C++ (MICROSOFT, 2023).
- c) **OpenGL:** é uma API gráfica multiplataforma que permite a criação de gráficos 2D e 3D de alta qualidade em C++. Ele é amplamente utilizado para desenvolver jogos que visam diferentes sistemas operacionais e plataformas (TECMUNDO, 2008).

C#:

- a) **Unity 3D:** é o mais famoso desses motores de jogos (game engines) e conquistou os desenvolvedores por sua simplicidade e pelo visual diferenciado dos games criados com ele. Apesar do nome, temos uma ferramenta voltada para o desenvolvimento de games com a mesma facilidade tanto em 2D quanto em 3D (HENRIQUE, 2017).
- b) **MonoGame:** é uma estrutura de código aberto que permite o desenvolvimento de jogos multiplataforma usando C#. Ele é baseado na implementação da Microsoft do .NET *Framework* e oferece suporte a várias plataformas, incluindo Windows, macOS, Linux, Android e iOS (IMASTERS, 2023).
- c) **Xamarin:** é um conjunto de ferramentas para desenvolvimento de aplicativos multiplataforma usando C#. Embora seja mais conhecido pelo desenvolvimento de aplicativos móveis, também pode ser usado para criar jogos para dispositivos móveis usando C# (MICROSOFT, 2023).

Essas são apenas algumas das bibliotecas, *frameworks* e ferramentas disponíveis para o desenvolvimento de jogos com C++ e C#. Existem muitas outras opções e cada uma delas oferece recursos e vantagens diferentes. A escolha depende das necessidades específicas do projeto, das habilidades da equipe de desenvolvimento e do objetivo final do jogo.

4.3. Tendências e perspectivas futuras do C++ e do C# no desenvolvimento de jogos

O desenvolvimento de jogos está em constante evolução, impulsionado pelo avanço da tecnologia e pelas demandas dos jogadores. Tanto C++ quanto C# desempenham papéis significativos nesse cenário e continuarão a ter importância no futuro do desenvolvimento de jogos. Algumas tendências e perspectivas futuras para essas linguagens são:

- a) C++ se manterá como uma escolha sólida para jogos de alto desempenho e controle de baixo nível. Com seu poder de otimização e controle direto sobre o hardware, C++ continuará a ser amplamente utilizado em jogos AAA e em projetos que exigem o máximo de desempenho (ALMEIDA.F.(2009).
- b) C# continuará a ser uma linguagem popular no desenvolvimento de jogos multiplataforma e indie. Com a crescente popularidade do Unity Engine e a facilidade de uso do C#, é esperado que mais desenvolvedores optem por essa combinação para criar jogos acessíveis em várias plataformas (HUGO.V,2023).
- c) O surgimento de novas bibliotecas, *frameworks* e ferramentas para o desenvolvimento de jogos em C++ e C# irá facilitar ainda mais o processo de criação. A comunidade de desenvolvimento está constantemente inovando e criando soluções que aumentam a produtividade e melhoram a experiência do desenvolvedor (ASRON.S,2023).
- d) A realidade virtual (VR) e a realidade aumentada (AR) são áreas em crescimento no desenvolvimento de jogos. Ambas as linguagens, C++ e C#, estão sendo usadas para criar experiências imersivas nesses campos emergentes, e é provável continuarem a desempenhar um papel importante no futuro do desenvolvimento de jogos VR e AR (KIRNER.C,2007).

5 Consideração Finais

No contexto do desenvolvimento de jogos, a escolha entre as linguagens C++ e C# é crucial e depende das necessidades específicas do projeto, bem como das preferências e

habilidades da equipe de desenvolvimento. Ambas as linguagens têm suas vantagens e desvantagens, e ambas desempenham papéis importantes na indústria de jogos.

C++ é uma linguagem poderosa que oferece controle de baixo nível e alto desempenho. É amplamente utilizado em jogos de alto desempenho, como jogos AAA, devido à sua capacidade de otimização e ao controle direto sobre o hardware. No entanto, sua curva de aprendizado íngreme e o gerenciamento manual de memória podem representar desafios para desenvolvedores iniciantes. O C++ continua a ser uma escolha sólida para jogos que exigem o máximo de desempenho e flexibilidade.

Por outro lado, C# se destaca pela facilidade de uso e integração com o Unity Engine, tornando-o uma escolha popular para desenvolvedores que buscam desenvolver jogos multiplataforma e indie. A linguagem oferece um desenvolvimento rápido e eficiente, mas pode ter um desempenho ligeiramente inferior em comparação com o C++. A dependência da Microsoft e a limitação à plataforma Unity são fatores a serem considerados ao escolher C#.

Ambas as linguagens têm um ecossistema robusto de bibliotecas, *frameworks* e ferramentas que facilitam o desenvolvimento de jogos. A escolha de uma linguagem deve levar em consideração as necessidades do projeto e a experiência da equipe.

Quanto aos desafios encontrados, o desenvolvimento de jogos é uma área em constante evolução, com novas tecnologias e tendências emergentes. Portanto, os desenvolvedores enfrentam o desafio constante de se manterem atualizados e acompanharem as mudanças na indústria de jogos. Além disso, a escolha entre C++ e C# deve ser feita com base nas necessidades do projeto, o que requer uma análise cuidadosa das características e requisitos específicos.

É importante destacar que ambas as linguagens continuarão a desempenhar papéis significativos no desenvolvimento de jogos. A tendência é que C++ seja amplamente utilizado em jogos de alto desempenho, enquanto C# permanecerá popular para jogos multiplataforma e indie, especialmente com a crescente popularidade da realidade virtual (VR) e da realidade aumentada (AR). A inovação na criação de bibliotecas e ferramentas para o desenvolvimento de jogos em ambas as linguagens oferecerá ainda mais recursos aos desenvolvedores.

Portanto, é fundamental que os desenvolvedores estejam preparados para acompanhar as tendências e evoluções na indústria de jogos, aprimorando suas habilidades e adaptando-se às mudanças. O desenvolvimento de jogos é uma área emocionante e desafiadora, e o uso eficiente das linguagens de programação certas desempenha um papel crucial na criação de experiências de jogo envolventes e emocionantes para os jogadores.

Referências

ALMEIDA.F.(2009). O próximo padrão de c++. RevistaProgramar.
Disponível em: <<https://www.revista-programar.info/artigos/o-proximo-padrao-de-c/>>.
Acesso em: 3 nov. 2009.

ASRON.S.(2023). C# ou C++: qual você deve aprender?.BitDegree.
Disponível em: <<https://br.bitdegree.org/tutoriais/c-sharp-ou-c-plus-plus/>>. Acesso em: 18 mar. 2023.

BRUNA.(2023). As 10 Linguagens de Programação Mais Usadas em 2023.Hostinger.

Disponível em: < <https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas> >. Acesso em: 04 ago. 2023.

CLARK.M.(2022). As 10 principais startups usando C#.BackApp.

Disponível em: < <https://blog.back4app.com/pt/startups-usando-c-sharp/#:~:text=Embora%20originalmente%20destinado%20como%20um,Scrolls%3A%20Legends%2C%20etc.%20> >.

DOURADO.R.(2015).Programação Orientada a Objetos.Renatomaia.

Disponível em: < https://renatomaia.net/arquivos/Unimontes/POO/17_OutrosMecanismos.pdf >. Acesso em: 24 ago. 2015.

HENRIQUE.(2017). Como criar jogos: conheça as principais ferramentas. DevMedia.

Disponível em: < <https://www.devmedia.com.br/como-criar-jogos-conheca-as-principais-ferramentas/37848> >.

HUGO.V.(2023). A Poderosa Combinação de C# e Unity no Desenvolvimento.

Disponível em: < https://www.linkedin.com/pulse/poderosa-combina%C3%A7%C3%A3o-de-c-e-unity-desenvolvimento-manata-pontes/?trk=public_post&originalSubdomain=pt >.Acesso em: 28 Set. 2023.

IMASTERS.(2023). Confira quais foram as linguagens de programação mais usadas em 2022.Imasters.

Disponível em: < <https://imasters.com.br/noticia/confira-quais-foram-as-linguagens-de-programacao-mais-usadas-em-2022> >. Acesso em: 6 fev. 2023.

KIRNER.C,SISCOOTTO.R.(2007). Realidade virtual e aumentada.UFPB.

Disponível em: < http://de.ufpb.br/~labteve/publi/2007_svrps.pdf >. Acesso em: 28 mai. 2007.

MACORATTI.J.(2014). C# - Garbage Collection, Dipose e Finalize. Macaratti.

Disponível em: < https://macoratti.net/14/09/c_garb1.htm >. Acesso em: 25 set. 2014.

MARCHIORI.L.(2022). As 9 linguagens mais usadas na programação de jogos?

BeTrype.

Disponível em: < <https://blog.betrybe.com/linguagem-de-programacao/linguagens-para-programacao-jogos/> >. Acesso em: 31 out. 2022.

MICROSOFT.(2023). Um tour pela linguagem C#. Microsoft.

Disponível em: < <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/> >. Acesso em: 15 fev. 2023.

MICROSOFT.(2023). Elementos gráficos e jogos do DirectX. Microsoft.

Disponível em: < <https://learn.microsoft.com/pt-br/windows/win32/directx> >. Acesso em: 13 jun. 2023.

MICROSOFT.(2023). O que é o Xamarin?. Microsoft.

Disponível em: < <https://learn.microsoft.com/pt-br/xamarin/get-started/what-is-xamarin> >. Acesso em: 13 jul. 2023.

NOLETO.C.(2021). C++: o que é, porque usar e quais as diferenças com C.BeTrybe.
Disponível em: < <https://blog.betrybe.com/linguagem-de-programacao/cpp/> >. Acesso em: 13 out. 2021.

OLIVEIRA.F. c++ orientado a objetos. Facom.
Disponível em: < https://www.facom.ufu.br/~flavio/ed1/files/C++%20ORIENTADO%20A%20OBJETO_S.pdf >.

RICARTE.I.(2001). Origens de C++.Unicamp
Disponível em: < https://www.dca.fee.unicamp.br/cursos/POO_CPP/node18.html >.
Acesso em: 01 jun. 2001.

SACRAMENTO.G.(2021). Conheça a linguagem c#, suas características e para que serve.TeraBlog.
Disponível em: < <https://blog.somostera.com/desenvolvimento-web/linguagem-c> >.

SILVA.E.(2020). 5 melhores linguagens de programação para Android. GeekHunter.
Disponível em: < <https://blog.geekhunter.com.br/linguagens-de-programacao-para-android/> >.

TECMUNDO.(2008). O que é OpenGL?
Disponível em:< <https://www.tecmundo.com.br/video-game/872-o-que-e-opengl-.htm>>.Acesso em: 4 Nov. 2008.

WIKILIVROS.(2017). Programar em C++/Alocação dinâmica de memória.WikiLivros.
Disponível em: < https://pt.wikibooks.org/wiki/Programar_em_C%2B%2B/Aloca%C3%A7%C3%A3o_din%C3%A2mica_de_mem%C3%B3ria >. Acesso em: 7 jul. 2017.

YUGE.C.(2022). As 10 linguagens de programação mais utilizadas no desenvolvimento de games.CanalTech.
Disponível em: < <https://canaltech.com.br/mercado/as-10-linguagens-de-programacao-mais-utilizadas-no-desenvolvimento-de-games-213047/> >. Acesso em: 01 abril 2022.

ZAMBALDI.L.(2021). 4 linguagens de programação mais usadas para criar jogos.IDoCode.
Disponível em: < <https://idocode.com.br/blog/programacao/4-linguagens-de-programacao-jogos/>>. Acesso em: 20 out. 2021.