



---

## SUMÁRIO

### **12991 - ANÁLISE DE FUNCIONALIDADES E DESEMPENHO DE MOTORES DE JOGOS GRATUITOS**

Cassiano Pereira Sonaglio, Mariana Lopes Accordi, Luciano Antunes<sup>1</sup> .....



## Trabalho Completo de Pesquisa

### 12991 - ANÁLISE DE FUNCIONALIDADES E DESEMPENHO DE MOTORES DE JOGOS GRATUITOS

**Cassiano Pereira Sonaglio, Mariana Lopes Accordi, Luciano Antunes<sup>1</sup>**

<sup>1</sup>Universidade do Extremo Sul Catarinense – UNESC

**Resumo:** A realização desse projeto tem como objetivo identificar e analisar os principais motores de jogos gratuitos. Para a realização deste projeto torna-se necessário elencar os principais motores de jogos gratuitos, para que possa ser realizada uma análise de suas funcionalidades e desempenho. Com os resultados obtidos pretende-se apresentar ao futuro desenvolvedor de jogos a escolha do motor mais apropriado para o desenvolvimento de sua aplicação. Essa pesquisa está inserida na linha de pesquisa Jogos Digitais, Dispositivos Móveis, TV Digital e Robótica do grupo de pesquisa Desenvolvimento em Realidade Aumentada, Jogos Digitais e Dispositivos Móveis.

**Palavras-chave:** Desenvolvimento de jogos. Jogos digitais. Motor de jogo.

## 1 INTRODUÇÃO

Com o crescimento do mercado de jogos, a apresentação de novos lançamentos de hardware, mecânicas e métodos inovadores de trabalhar a interação do jogador com um sistema de computador, acompanha a crescente demanda de novos títulos de jogos e, a partir disso, de criadores de conteúdo, mais especificamente, desenvolvedores. Por motivos assim, além da vigorosa paixão demonstrada pela própria indústria, cada vez mais pessoas são atraídas por carreiras relacionadas à criação e ao desenvolvimento de jogos digitais.

Este mercado, entretanto, utiliza-se de uma vasta rede de conhecimentos, envolvendo, segundo Bessa et al. (2007), domínio sobre ciências como computação, matemática e física. Tal necessidade torna mais difícil, para um desenvolvedor que trabalha sozinho, produzir um jogo do começo ao fim. Essa amplitude de estudos, que podem ser úteis ao produzir um jogo e tangenciam diversas áreas, não é algo que se realiza simplesmente tendo conhecimento sobre a área da tecnologia da informação. Um jogo oferece muito espaço e oportunidade a serem aproveitados e, por conta disso, vários profissionais se dedicam à produção de um único projeto.

Entre as possíveis aplicações de conteúdo dentro de um jogo, as variações são diversas e podem englobar, principalmente, as áreas: criativa, abarcando roteiro, história



e ambientação; artística, tanto para áudio quanto para vídeo, reforçando a ambientação e cenário; de exatas e lógica, presentes, principalmente, na programação de um jogo e desenvolvimento de mecânicas e sistemas; além de áreas de estudos humanos, que tendem a utilizar conceitos de psicologia e sociologia para compreender maneiras de usufruir todo o potencial interativo que a mídia de jogos proporciona, visando entender o jogador e de que forma ele pode sentir-se mais imerso na experiência.

Por conta disso, independente da área de competência de um desenvolvedor, o uso de uma *game engine* (motor de jogo) é um dos maiores facilitadores para o progresso de um projeto. Ela propicia maior agilidade e, em muitos casos, auxilia em tarefas que envolvam áreas que um desenvolvedor solitário pode não possuir proficiência, como no caso da programação. Dias (2016) explica que um motor é a plataforma onde ocorre o desenvolvimento do jogo em si, é nele que são unidos os arquivos de áudio, vídeo e scripts de programação, por exemplo. Portanto, uma *engine* reduz boa parte do peso bruto do trabalho de produção.

A produção e mercado de entretenimento de jogos digitais mostra um grande crescimento no mundo atual. Independentemente da plataforma, em sua natureza, um jogo digital é um *software* interativo. Para produzir qualquer tipo de software, normalmente, utiliza-se alguma ferramenta que possa facilitar o trabalho. A ferramenta usada no auxílio ao desenvolvimento de jogos é o motor de jogo (TORI, 2006).

Os motores de jogo são uma peça-chave no desenvolvimento, pois permitem adicionar recursos sofisticados aos jogos, tornando-os mais interessantes, com um maior realismo e melhor jogabilidade.

Uma tarefa importante no desenvolvimento é escolha nas ferramentas de suporte, conectada à definição do motor de jogo que é uma peça essencial para a criação do jogo em si. O motor de jogo permitirá, ao desenvolvedor, um grande reaproveitamento de códigos, ou seja, economia de tempo (WATT, 2003).

## 2 Trabalhos Relacionados

Em meio a diversos lançamentos e novidades, no que diz respeito a conteúdo específico para jogadores e suas plataformas, existem também pesquisas e aprofundamentos visando a confecção de jogos e aplicativos. Diversos trabalhos, como o realizado por Bessa et al. (2007), já citado anteriormente, focalizam-se na criação de



ambientes destinados ao desenvolvimento de jogos, passando pelo processo de criar um motor de jogos.

Entretanto, também são publicados artigos com o intuito de explicar conceitos e funcionalidades desses programas, utilizados para o desenvolvimento de jogos. Um exemplo é Lima (2008), que divide uma *game engine* em seus sistemas individuais, como o motor gráfico, gerador de física, entre outros, em busca de entender o processo de criação de um jogo.

### 3 Referencial Teórico

#### 3.1 Motor de Jogo

Uma *game engine*, ou motor de jogo, é uma ferramenta utilizada para o desenvolvimento de jogos, aplicativos ou quaisquer experiências virtuais. O motor funciona como um programa de computador que agiliza e simplifica parte do trabalho pesado da produção de jogos. Em geral, uma *engine* conta, principalmente, com: *render* e suporte de vídeo, seja 3D ou 2D; sistema de *script*; gerador de física; detector de colisões; sistema sonoro; e, por fim, animador. Podem, também, possuir bibliotecas de arquivos reaproveitáveis e suporte para localização de arquivos e exportação.

Sistemas básicos, como o de bibliotecas reutilizáveis e motor de física e colisão, aumentam a capacidade dos desenvolvedores de focar em partes mais valiosas da personalização de seu projeto, ao invés de trabalhar para “recriar” sistemas que são frequentemente utilizados em diversos jogos, como a gravidade e colisão entre objetos.

Além dessa utilidade de reaproveitar códigos e *scripts* de mecânicas já usadas, um motor de jogo também tem grande participação nos testes e nas atualizações de versões, com seu sistema de compilação e execução em tempo real.

Com essas funcionalidades e sistemas disponíveis, o uso de um motor de jogos apresenta grandes facilidades durante o processo de criação e desenvolvimento, não apenas para desenvolvedores solitários, mas também para equipes que tenham o desejo de acelerar os resultados.

Com o avanço do mercado de jogos, a procura e oferta de motores de jogos também cresceu. Empresas com renome disponibilizam versões de suas próprias *engines*



criando um sistema de compra e venda do programa e a liberdade de usá-lo para desenvolver jogos que não envolvam tal empresa.

### 3.2 Assets

Os *assets* são definidos como objetos úteis e valiosos. Podem ser encontrados em três tipos: de propriedade do motor, de criações de usuários ou de terceiros. Algumas *engines* utilizam este sistema para disponibilizar tutoriais.

### 3.3 Linguagem de Programação

O computador é capaz de compreender uma linguagem de programação composta apenas de números. Sendo assim, construir algoritmos na linguagem de máquina é um processo extremamente complicado. Para facilitar este processo, foi necessário a criação de códigos relacionados à linguagem de máquina com o tipo de linguagem em que se aproxima da “lógica humana”, para melhor compreensão dos códigos (MEDINA; FERTIG, 2005).

## 4 Desenvolvimento

Para a obtenção dos resultados desejados, foi necessária uma pesquisa para destacar os principais motores de jogos gratuitos para uso atualmente no mercado. A partir dessa, foi decidido utilizar como objetivo classificar e aprofundar a pesquisa nos motores Unity 3D e Unreal Engine 4, que, segundo pesquisado, ficaram marcadas como as principais *engines* gratuitas, de acordo com seu potencial e benefícios oferecidos.

Após decididas as *engines*, foram realizados diversos testes e buscas para apontar comparações, diferenças e especificidades para cada uma delas. Entre as informações que foram procuradas, as principais eram: capacidade e potencial gráfico do motor; custo e peso de processamento; facilidade de aprendizado e uso; comunidade e suporte; e estudo sobre sua interface.

A busca por tais informações aconteceu, especialmente, em forma de pesquisas nos sites e documentações oficiais das próprias *engines*, além de testes com o próprio *software* e a utilização de projetos-teste da equipe de pesquisa para a verificação



da acessibilidade de conteúdo e aprendizado, e teste com projetos pré realizados e disponibilizados pela comunidade ou empresa para análise de ambientes mais pesados no que faz relação ao processamento e qualidade gráfica.

Por meio da utilização dos motores para realização das análises, foi realizado a pesquisa te requisitos do sistema. No próprio site oficial de ambos possuem informações dos requisitos de máquina, podendo ser observado na tabela a seguir.

Na utilização destes motores, foram realizados testes em máquinas diferentes. Identificando a primeira como M1 (máquina um), que possui processador Intel I3-4130, 3.4 GHz e 4 GB em memória RAM com DirectX 11. M2 (máquina dois), que possui processador Intel XEON E3 1240, 3.33GHz e 8 GB em memória RAM com DirectX 11. Por fim, M3 (máquina 3), que possui processador Intel I7-5500U, 2.4 GHz e 8 GB em memória RAM com DirectX 11. O único diferencial da última máquina para as outras restantes, é por possui uma placa de vídeo dedicada Nvidia GeForce 920M de 2 GB.

Para avaliar a execução dos motores, foram realizados os mesmos protótipos iniciais de cada motor. A versão utilizada para as análises dos motores são: Unity 3D 5.3.4f1 e UE4 4.12.3.

Tabela 1 - Comparação de requisitos

Requisitos	Unity 3D	Unreal Engine 4
S. O.	Windows 7/8/10; Mac OS X 10.8+	Windows 7/8 64-bit
Processador	Intel Pentium Dual Core, 1.8HGz	Quad-core Intel ou AMD, 2.5 GHz
Memória	4 GB RAM	8 GB RAM
Placa de Vídeo/Versão DirectX	Placa Gráfica com Recursos do DirectX	DirectX 11 compatível com a placa de vídeo

Fonte: Informações retiradas diretamente dos sites das aplicações Unity e Unreal referenciados ao fim do documento.

## 5 Resultados Obtidos



## 5.1 Desempenho nos computadores

Nos testes com M1 não foi possível executar corretamente os motores Unreal e Unity 3D. Para renderização, o protótipo realizado não chegou aos 10 fps. Com avaliação, não é recomendado a utilização deste tipo de máquina, para o desenvolvimento de jogos. Os motores não realizam as tarefas corretamente, e o tempo de resposta prejudica gradativamente no tempo do desenvolvimento.

Os testes realizados com M2 foi considerável, porém, apenas com Unity 3D. O tempo de resposta da Unreal se assemelha ao mesmo tempo da Unity 3D utilizada na M1. Na execução do protótipo a Unreal não passou de 16 fps, já Unity 3D teve melhor resposta com seus 60 fps.

Para M3 todos os motores obtiveram um tempo de resposta instantâneo, não ocorreu problemas de carregamento como as outras máquinas e a execução do protótipo fluída.

Como apenas M3 possui melhor capacidade para o desenvolvimento dos protótipos, foi realizado os testes dos protótipos apenas na M3.

## 5.2 Potencial Gráfico

### 5.2.1 Unreal Engine 4

Definitivamente, um dos motores mais poderosos do Mercado, apresenta alta qualidade de rendering, reflexões HDR (High dynamic range), profundidade de visão, motion blur e muitas outras funções para aplicar atmosféricamente ou especificamente.



Figura 1 – “A boy and his kite”

Fonte: Captura de tela do vídeo em <https://www.youtube.com/watch?v=JNgsbNvkNjE>

“A boy and his kite” é uma animação totalmente produzida e executada utilizando o software da Unreal, que foi utilizada como propaganda para o motor quando este tornou-se gratuito. Durante o vídeo, são demonstrados diversos destes métodos gráficos que a Unreal disponibiliza. Possuindo todo este potencial, a Unreal é, muitas vezes utilizado para grandes projetos, extrapolando a qualidade gráfica sem perder em outros aspectos do jogo.

### 5.2.2 Unity 3D

O motor Unity é um pouco mais limitado em termos gráficos, apesar de não ficar muito para trás, ele possui boa parte das ferramentas que a Unreal apresenta, mas, de forma geral, a qualidade máxima atingida acaba sendo ultrapassada pelo outro software.



Figura 2 – “The blacksmith”

Fonte: Captura de tela do vídeo “The Blacksmith” encontrado em <https://www.youtube.com/watch?v=pXWAsayTFT0>

“The blacksmith” video promocional da Unity, criado com o mesmo propósito de testar e expor a qualidade gráfica atingível com o uso do próprio motor. Por conta de sua qualidade gráfica, o Unity pode ser utilizado para projetos que necessitem de gráficos avançados, mas nem sempre é o favorito, já que a Unreal oferece mais poder bruto.

## 5.3 Processamento

### 5.3.1 Unreal Engine 4

Em termos de processamento, a Unreal Engine está entre os mais pesados programas de desenvolvimento, especialmente por conta de sua qualidade gráfica, enquanto a Unity é mais abrangente, apesar de também oferecer alta capacidade e, dependendo do projeto, tornar-se extremamente consumidora de processamento.

As informações da tabela foram retiradas diretamente do site da Unreal e apresentam os requisitos recomendados para o uso e desenvolvimento no motor. Apesar de não serem mostrados os requisitos mínimos, o motor apresenta um peso elevado, recomendando 8GB de memória RAM e forçando muito a utilização de máquinas com menos memória disponível. Em testes com um processador Intel® Core™ I5 com 6GB de RAM disponível, o programa apresentou baixo framerate, tendo que reduzir sua qualidade gráfica, o que, apesar de demonstrar o custo de processamento, mostra também o



interesse da engine em facilitar o uso em computadores menos potentes, já que o próprio motor realiza o downgrade dos gráficos.

Quanto a potencial, a da Unreal Engine 4 é altíssima, tendo projetos como o “A boy and his kite”, que foi utilizado para divulgação da engine, quando ela se tornou gratuita, que tem como requisitos recomendados 24GB de RAM. Portanto, mesmo com estes requisitos, a Unreal Engine apresenta uma enorme capacidade de desenvolvimento

### 5.3.2 Unity 3D

A Unity, por outro lado, apresenta requisitos extremamente abrangentes, demonstrando que pode ser utilizada em praticamente qualquer máquina, com um enorme alcance ao público em geral.

O software da Unity é, claramente, muito mais leve e acessível, apresenta muito poucas limitações em relação às necessidades para uso e desenvolvimento com a ferramenta. Com isso, a Unreal acaba sendo ofuscada quando o quesito é jogos pequenos, o custo de processamento em um jogo que busca simplicidade é o suficiente para pender os usuários a desenvolverem seus títulos no motor mais leve que continua atendendo às suas necessidades.

### 5.4 Comunidade e Acesso

Ambos os motores oferecem sistemas de tutoriais e manual de usuário online, entretanto, neste quesito a Unity tem a vantagem por possuir uma comunidade muito maior e mais ativa por conta das limitações e utilizações de cada engine. Já que a Unreal tem tanto poderio gráfico e demanda tanto consumo de processamento, acaba sendo melhor utilizada por estúdios com estrutura suficiente e objetivo delimitando a utilização dos recursos do software. A Unity, por outro lado, oferece um sistema mais leve e melhor utilizado para pequenos projetos, o que costuma ser o alvo de usuários independentes ou pequenos estúdios.

Por possuir uma comunidade maior, a Unity possui fóruns e diversos tutoriais e aulas online fora dos oferecidos pelo motor em si. No próprio site do programa, são oferecidos tutoriais oficiais e fóruns com conteúdo diretamente de usuários.

Coisas desse tipo são muito mais escassas no caso da Unreal, mas o motor oferece um manual bem estabelecido com dicas e conteúdo para auxiliar o usuário em diversos momentos, além de uma série de vídeos visando satisfazer a falta de conteúdo de fontes externas.

## 5.5 Interface e facilidade de uso

### 5.5.1 Unreal Engine 4

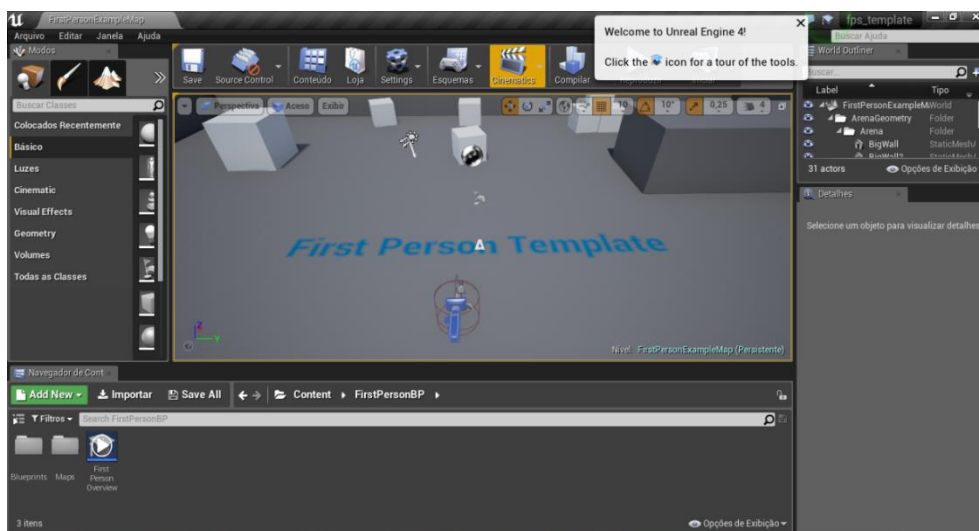


Figura 3 –Interface de usuário da Unreal

Fonte: Captura de tela da aplicação no sistema operacional Windows 10

A figura 6 é uma imagem de tela da interface de usuário da Unreal. Sua interface contém muita informação sobre o que o usuário está utilizando, especificações de objeto, atalhos para adicioná-lo e para executar o projeto, entretanto, seus ícones são grande e acabam tendo uma poluição visual na tela, deixando tudo apertado e sobrepondo-se.

Em geral, para a utilização de softwares como este, são utilizadas duas ou mais telas, para dividir as funções e não as deixar esmagadas, mas utilizar a Unreal Engine em uma única tela torna-se muito difícil e extremamente desagradável por conta disso.

Apesar disso, o software é bastante acessível e tem um tutorial “in-action” que é aberto assim que se inicia o programa, apresentando função a função, seu objetivo e como utilizá-lo.

## 5.5.2 Unity 3D

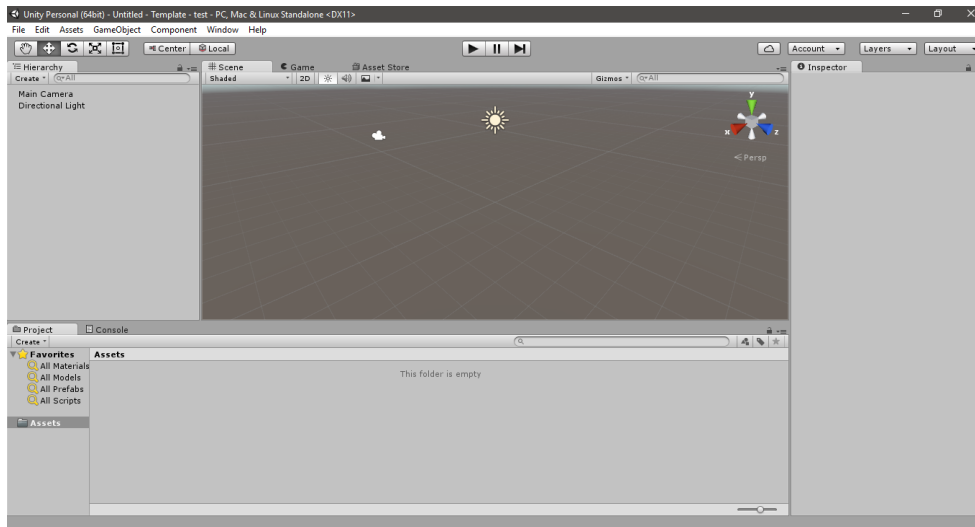


Figura 4 – Interface Unity 3D

Fonte: Captura de tela da aplicação no sistema operacional Windows 10

A UI (user interface) da Unity é muito mais clara, mas espessa e em uma escala bem mais reduzida e bem aproveitada. Ainda assim, a Unity oferece a mesma quantidade de informações passadas pela Unreal, o que a deixa muito mais agradável e amigável. Apesar disso, ainda é melhor utilizada com, ao menos, uma tela extra, abrindo ainda mais o canvas e facilitando o acesso às informações.

Ainda que a interface da Unreal seja ajustável, a Unity faz um trabalho muito mais avançado, simplificando cada passo e permitindo uma customização de interface muito mais rapidamente e melhor organizada.

## 5.6 Linguagens de programação

Outro ponto a ser destacado importante para a acessibilidade e escolha da engine ideal, é a linguagem de programação utilizada em cada uma delas. A Unity 3D tem opção de utilização de duas linguagens, C# e Java Script, ambas as linguagens são muito comuns entre os programadores e são bastante utilizadas, de maneira geral, para o desenvolvimento de softwares e aplicativos. Isso faz com que o motor Unity 3D ofereça certa facilidade de aproximação para o usuário que já possui conhecimento de programação ou que tem interesse de estudar e aprofundar-se como programador.



Enquanto isso, o Unreal Engine 4 apresenta um sistema de programação diferenciado, o Blueprint, um sistema desenvolvido especificamente para o uso no motor, uma linguagem de programação e códigos visuais, facilitando o aprendizado dos leigos e agilizando a produção ainda mais. O Blueprint tem tutoriais específicos lançados pela própria engine e é bastante acessível. Apesar disso, a Unreal continua oferecendo uma linguagem mais padrão, o C++, o que não afasta usuários que tenham conhecimento prévio, forçando-os a estudar ou utilizar o sistema Blueprint.

## 6 Conclusão

Para a realização deste artigo, buscou-se adquirir conhecimento suficiente para o esclarecimento e definição de que motor de jogos teria preferência no que diz respeito a diferentes projetos e aplicações no desenvolvimento e produção de jogos digitais, facilitando, assim, a entrada e formação de novos desenvolvedores para o mercado de jogos. Com este fim, foram realizados testes nos respectivos motores de desenvolvimento, além de pesquisas bibliográficas e buscas por documentos previamente executados.

Com base nos materiais verificados, foi possível determinar que as aplicações de teorias envolvidas em motores de jogos são muito variadas e suas dúvidas estendem-se para várias direções, como os exemplos de documentos relacionados exemplificam.

Partindo das pesquisas praticadas e resultados alcançados, aponta-se que, de maneira geral, ambos os motores de jogos apresentam características apreciáveis que justificariam sua utilização em ampla escala. A distinção entre eles, entretanto, mantém-se, principalmente, nas questões de especificidade e nas variações de necessidade de conhecimento prévio para a sua utilização, já que cada software apresenta a opção de aceitar diferentes linguagens de programação. Além disso, recomenda-se a utilização de uma placa gráfica dedicada para resultados melhores em sua utilização.

Unity 3D tem linguagens mais padronizadas, com o C# e o JavaScript, mas a Unreal Engine, além de aceitar programação em C++, para desenvolvedores que tenham conhecimento prévio de programação, tem a facilidade do sistema Blueprint e seus códigos visuais, sendo favorecida neste quesito.

Por outro lado, enquanto o motor Unity 3D apresenta maiores vantagens para o desenvolvimento de jogos rápidos e tem maior facilidade de acesso a tutoriais e conteúdo



gerado pela própria comunidade, a Unreal Engine 4 demonstra mais aspectos favoráveis no que diz respeito ao desempenho gráfico, sendo mais proveitosa sua utilização em projetos que visam aplicar toda a sua capacidade e desenvolver jogos maiores e mais pesados.

Para os outros aspectos estudados, especificamente facilidade de acesso a conteúdo de aprendizado e comunidade disponível, a Unity 3D sai, novamente, a frente. Esse fato ocorre pela quantidade de usuários que cada motor possui, já que, segundo a pesquisa da GediGames (2016) a Unity 3D é o motor mais utilizados pelas empresas brasileiras com 79% da pesquisa. O motor Unity 3D tem menos exigências de hardware e é mais propícia para a utilização no desenvolvimento de jogos com menor custo de produção e necessidades, tornando-se mais conveniente para o pequeno desenvolvedor, assim como os estúdios independentes, que é a porção da indústria que gera o maior volume de conteúdo, tanto comercializável quanto apenas por experiência.

Em vista dos fatos esclarecidos, salvo casos em que o desenvolvedor tenha maiores necessidades gráficas, ou apenas afinidade com as linguagens de programação da Unreal Engine 4, o motor Unity 3D é apresentado como o mais cabível e mais proveitoso em projetos de desenvolvimento de jogos. Sua agilidade, acessibilidade e portabilidade para diversas plataformas em conjunto com qualidade gráfica relativamente avançada e baixa necessidade de hardware o tornam a melhor opção para uma enorme gama de estúdios e desenvolvedores que necessitam de uma plataforma gratuita de confecção de jogos digitais.

**Fonte financiadora:** Propex

## REFERÊNCIAS

BESSA, A. et al. **O Desenvolvimento de um Motor Multiplataforma para Jogos 3D.** 2007 - inf.ufrgs.br. Disponível em: <<http://www.inf.usi.ch/phd/bezerra/publications/bessa2007dmm.pdf>>. Acesso: 20 jun 2016.

DIAS, R. **5 Dicas para escolher a melhor game engine para o seu jogo.** 2016. Disponível em: <<http://materiais.producaodejogos.com/game-engine-pd>>. Acesso em: 21 jun. 2016.



GEDIGAMES. **Relatório final – Mapeamento da indústria brasileira e global de jogos digitais.**

<[http://www.bndes.gov.br/SiteBNDES/bndes/bndes\\_pt/Galerias/Arquivos/conhecimento/seminario/seminario\\_mapeamento\\_industria\\_games042014\\_Relatorio\\_Final.pdf](http://www.bndes.gov.br/SiteBNDES/bndes/bndes_pt/Galerias/Arquivos/conhecimento/seminario/seminario_mapeamento_industria_games042014_Relatorio_Final.pdf)>. Acesso: 12 jul 2016.

LAVIERI, E. **Getting Started with Unity 5: Leverage the power of Unity 5 to create amazing 3D games.** Birmingham: Mumbai, 2015.

LIMA, E. **3D Game Builder: uma Game Engine para a Criação de Jogos 3D.** Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação), Universidade do Contestado, Porto União, 2008. Disponível em: <

[http://edirlei.3dgb.com.br/artigos/3d\\_game\\_builder\\_tcc.pdf](http://edirlei.3dgb.com.br/artigos/3d_game_builder_tcc.pdf)>. Acesso: 22 jun 2016.

MEDINA, M.; FERTIG, C. **Algoritmos e Programação: teoria e prática.** São Paulo: Novatec Editora, 2006.

TORI, R. et al. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada.** Belém: Viii Sysposium On Virtual Reality, 2006.

UNREAL ENGINE. **Unrealengine.com.** Disponível em: <<https://www.unrealengine.com/>>. Acesso: 10 jun 2016.

WATT, A.; POLICARPO, F. **3D Games: Animation and Advanced Real-time Rendering.** 2. ed. Boston: Addison-wesley Longman, 2003.