



## FACULDADE DE TECNOLOGIA, CIÊNCIAS E EDUCAÇÃO

### Graduação

## GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

### Estudo comparativo entre engines de jogos

Carlos Gabriel Agostinho Gonçalo Klinke  
Adinovam Henriques de Macedo Pimenta (Orientador)

### RESUMO

A evolução das engines está agora se movendo para jogos mais realistas e tecnicamente ricos em vários campos, como a física, sons e renderização. Das primeiras engines até as engines mais atuais voltadas para jogos 3D, o objetivo do desenvolvimento permanece o mesmo: fornecer ao desenvolvedor uma plataforma para criar jogos exclusivos de tal maneira que os desenvolvedores não precisem escrever ou desenvolver o jogo a partir do zero, mas apenas implementar a ideia com a ajuda de uma engine. Desta maneira, muitos desenvolvedores (experientes ou iniciantes) têm encontrado nestas ferramentas a maneira mais simples, rápida e profissional de desenvolver novos jogos. Porém, com tantas engines disponíveis, o desenvolvedor precisa fazer uma escolha considerando os dados mais relevantes ao seu projeto. Este trabalho analisou dados de vários atributos relevantes ao processo de escolha considerando as 10 engines mais populares atualmente, servindo de conhecimento e auxílio à este processo decisório.

**Palavras-chave:** Motores de jogos. Desenvolvimento de jogos. Jogos digitais.

### ABSTRACT

The evolution of the game engines is now moving to more realistic and technically rich games in various fields such as physics, sounds and rendering. From the first engines to the most current engines for 3D gaming, the goal of development remains the same: to provide the developer with a platform to create exclusive games in such a way that developers do not have to write or develop the game from scratch, but only implement the idea with the help of an engine. In this way,

many developers (experienced or beginners) have found in these tools the simplest, fastest and most professional way of developing new games. However, with so many engines available, the developer needs to make a choice considering the data most relevant to his project. This work analyzed data of several attributes relevant to the process of choosing considering the 10 most popular engines currently, serving as knowledge and aid to this decision making process.

**Keywords:** Engines. Games. Mobile. Console. Comparison. Creation.

## **Introdução**

Nos últimos anos, o mercado de jogos vem crescendo à uma velocidade expressiva. As vendas de jogos em 2015 atingiram a margem de US\$ 91,5 bilhões, com um aumento de 11,84% em relação a 2014. Isso fez com que as receitas globais chegassem a US\$ 107 bilhões em 2017. No entanto, o crescimento não é apenas em vendas, mas também na miscelânea de jogos oferecida, oscilando de jogos escolares à tiroteios em primeira pessoa (CHAUDY; CONNOLLY, 2018).

Um dos fatores que impulsionam o crescimento desta miscelânea é o aumento do número de desenvolvedores de jogos. Crescimento este, que veio principalmente ao uso de Game Engines para o desenvolvimento dos mesmos (HARDY et al., 2015). Uma Game Engine (Motores de jogos, em tradução livre), ou simplesmente engine, é uma plataforma utilizada para a construção de jogos e tem por característica facilitar o desenvolvimento de tarefas relacionadas a jogos, tais como: interpretação e execução do jogo, cálculo relacionado à física empregada aos objetos dos jogos, e ainda permite que os desenvolvedores se concentrem nas sutilezas que tornam o jogo inimitável, como as texturas, relacionamento entre os objetos do jogo, efeitos sonoros, jogabilidade, etc. (BISHOP et al., 1998).

As engines são um conjunto de módulos reutilizáveis que podem ser manipulados para levar um jogo em direção ao realismo (LEWIS; JACOBSON, 2002). Os principais componentes técnicos de um jogo pertencentes às engines são: gráficos, animação, áudio, física, interface do usuário, inteligência artificial, alguns códigos já prontos para determinadas ações como pular, correr ou atacar, e conteúdos gráficos como cenários e personagens. A instanciação destes

componentes, associada à história do jogo, à definição de seus personagens, ao plano de fundo, ao avatar do mundo real e seus comportamentos, etc., são os componentes que criam o jogo real. Desta forma, podemos enxergar as engines como um middleware, oferecendo todos os recursos necessários não apenas para a criação dos jogos, mas também para a execução dos mesmos.

O termo game engines teve sua origem em meados da década de 1990 e foi utilizado para classificar jogos de empresas que vendiam os núcleos (motores) de seus jogos para outras empresas (BATTAIOLA, A. L. et al., 2002). Esse núcleo começou a ser cada vez mais utilizado com o crescimento de jogos 3D, como DOOM e o Quake, que fazem parte dos jogos do gênero FPS (First Person Shooter - Atirador em Primeira Pessoa, em tradução livre), pois, com a pouca tecnologia que havia na época era mais conveniente utilizar um núcleo pronto (TRENHOLME; SMITH, 2008). Engines são o principal formato para se construir jogos digitais hoje em dia, possuindo uma vasta gama de biblioteca de conteúdos para auxiliar no desenvolvimento de projetos de jogos (COWAN; KAPRALOS, 2014).

Com o crescimento do mercado de jogos aumentando a cada dia, novas ferramentas de desenvolvimento de jogos vêm surgindo, confundindo ou até mesmo atrapalhando o aprendizado e a criação de um jogo. Com o intuito de melhorar a tecnologia, constantemente os softwares de criação estão sendo atualizados e acabam, de certa forma, dificultando a escolha de uma ferramenta adequada para um determinado projeto de desenvolvimento de jogo (CHAUDY; CONNOLLY, 2018). Desta forma, este trabalho propõe um estudo comparativo entre as principais engines com o intuito de orientar a escolha da engine mais adequada para o desenvolvimento de jogos por parte de desenvolvedores iniciantes.

### **Objetivos Gerais**

O objetivo geral deste trabalho é apresentar um estudo comparativo entre as principais engines disponíveis hoje no mercado, especificando suas características a fim de apontar qual é a mais adequada para um projeto de jogo.

## Objetivos Específicos

- Identificar as principais engines disponíveis;
- Identificar as principais empresas suportadas pelas engines, bem como as plataformas e as linguagens de programação suportadas pelas engines.

## Metodologia

Para o levantamento dos dados necessários ao desenvolvimento deste trabalho, será utilizada como base a pesquisa qualitativa que, segundo Gerhardt e Silveira (2009):

Não se preocupa com representatividade numérica, mas, sim, com o aprofundamento da compreensão de um grupo social, de uma organização, e objetiva gerar conhecimentos novos, úteis para o avanço da ciência, sem aplicação prática prevista. Envolve verdades e interesses universais.

A partir do levantamento de dados, distinguir-se-á os atributos das engines, sendo eles: a compatibilidade de plataforma; empresas que as usam ou usaram; versões disponíveis (pagas ou gratuitas); linguagens de desenvolvimento; e foco de desenvolvimento (2D, 3D, Realidade Virtual e Realidade Aumentada).

## Evolução da engines

No início de 1989, uma engine de ficção científica chamado Ultima Underworld foi desenvolvido (BISHOP et al., 1998). Essa engine tinha o mesmo nome do jogo em si. Mais tarde, após o lançamento do jogo Space Rogue, a Origin System havia consolidado a engine Ultima Underworld no mercado e desenvolvido o algoritmo para mapeamento de textura, que pôde ser aplicado em pisos, tetos, paredes, etc. (DARKEN; MCDOWELL; JOHNSON, 2002).

Em 1993, a empresa ID Software desenvolveu a engine Doom, que não é uma engine 3D, mas tinha capacidade para representar objetos, personagens e mapa de nível inteiro por representação de sprites 2D (HERWIG; PAAR, 2002). A renderização era muito rápida para os padrões da época, e precisava de um

PC baseado na arquitetura 386 com suporte VGA padrão para rodar. Embora fosse uma engine 2D, a ilusão criada pelo desenvolvedor fez com que fosse um título em 3D.

A NovaLogic, em 1992, era proprietária da engine Voxel, que foi a engine básica de todos os jogos da série Comanche (FRITSCH et al., 2004).

A Voxel tinha o seu próprio caminho para representar objetos volumétricos, como bitmaps tridimensionais. Antes disso, todos os mecanismos aplicavam gráficos vetoriais, o que era um pouco mais lento em velocidade e apresentava menos detalhamento comparado à apresentação de bitmap 3D. Blade Runner, Command & Conquer são os jogos de destaque desenvolvidos pela engine Voxel (KOT, B. et al., 2005).

No final de 1993, outro jogo chamado Duke Nukem 3D foi lançado no mercado, tendo sido desenvolvido com a ajuda da engine Build. Assim como a engine Doom, ele também criou efeitos 3D na interface 2D simplesmente variando os setores com diferentes alturas para alcançar a ilusão de que o jogo era em 3D. Ao aplicar tags especiais a vários locais dentro de um setor específico, os desenvolvedores podem fazer com que, sempre que um jogador passar para esses pontos específicos, ele mude para um setor diferente, dando a ilusão de alterar os níveis ou o ambiente em tempo de execução (CLUA; BITTENCOURT, 2005).

A XnGine, de 1995, foi a primeira engine 3D desenvolvida no sistema operacional DOS. Mais tarde, esta engine começou a utilizar gráficos de alta resolução e que viria a ser compatível com placas de vídeo 3dfx (MARKS; WINDSOR; WÜNSCHE, 2007).

A engine Quake, desenvolvida pela Software ID em 1996, foi a primeira engine verdadeiramente 3D (MÓL; JORGE; COUTO, 2008). Ela tinha uma capacidade de processamento exclusiva para renderizar mapas eliminando determinadas áreas do processamento que o jogador não conseguiria ver. Isso advém do fato desta engine usar o Z-buffering, que é um método para determinar quais partes dos mapas são visíveis para o jogador e renderizar apenas essas seções.

A engine Renderware, de 1996, foi a engine mais popular para jogos multiplataforma. Ela suporta as plataformas PlayStation 2, Wii, GameCube, Xbox, Xbox 360, PlayStation 3 e PSP (FRIESE; HERRLICH; WOLTER, 2008).

A engine ID Tech 2 (conhecida como Quake II), de 1997, tem suporte à OpenGL nativo, efeitos de iluminação coloridos e suporte à linguagem C.

A engine GoldSRC, de 1998, levou os jogos de PC para uma nova era suportando OpenGL e Direct3D. Alguns exemplos de jogos bem-sucedidos baseados no OpenGL são: Half-Life, Day of Defeat e Counter Strike.

Uma das engines mais populares é o Unreal engine, de 1998, que deu origem ao famoso jogo Unreal Tournament. Ele integra sua própria linguagem de script chamada UnrealScript e editor de mapas chamado UnrealEd (PASSOS et al., 2009). Uma versão modificada da engine Quake II foi o Quake III, projetado em 1999, que suporta cores de 32 bits, shaders e redes avançadas.

John Slagel, o principal programador da Red Faction, desenvolveu a engine Geodmod (do inglês Geometry Modification) em 2001, com foco na reconstrução gráfica.

Em 2001, a engine de torque foi desenvolvida para modificar o jogo FPS Tribes.

2. Ela tinha uma opção de renderização on-the-fly com menos contagens de polígonos e também tinha um editor de mapas do mundo integrado à ele.

A engine Serious, de 2001, foi projetada para permitir grandes espaços e um grande número de personagens na tela a qualquer momento e dar origem à popular série Serious Sam (HUDLICKA, 2009). Mais tarde, com o Doom 3, de 2004, a maioria das superfícies de luz também foram feitas em tempo real, permitindo sombras mais realistas, mas à custa de serem capazes de renderizar sombras suaves. Para contornar isso, as luzes projetadas poderiam ser usadas para criar a ilusão de sombras suaves. Motor de Source Creator do Half Life 2, de 2005, incluindo tecnologias avançadas de sombreamento, iluminação e sombras dinâmicas, física, e outros vários efeitos, como superfícies de água refletivas de aparência realista, sombra de movimento em tempo real, etc. (INDRAPRASTHA; SHINOZAKI, 2009).

A engine CryEngine, de 2004, usou pixel shaders para obter o efeito de água realista em Far Cry. Desenvolvido pela empresa Crytek, ela produziu o jogo chamado Crysis, um jogo DirectX 10 com sombreador de objetos.

A Rockstar Advanced Game Engine (RAGE), de 2006, combina uma estrutura de renderização, mecanismo de física, mecanismo de áudio, bibliotecas de rede, mecanismo de animação e linguagem de script em um

pacote. O jogo Grand Theft Auto IV foi o mais bem sucedido produto dessa engine.

Em 2007, a engine Unreal e Frostbite, de 2008, levaram o avanço das engines à um patamar mais elevado com o jogo Battlefield: Bad Company, com efeitos inovadores desenvolvidos na engine Frostbite (PETRIDIS et al., 2012).

Cryengine 3, da Crytek, foi lançado em 14 de outubro de 2009 e tem suporte ao desenvolvimento no DirectX 9, 10 e 11. Ainda em 2009, foi anunciado que o jogo Crysis 2 foi desenvolvido pela Crytek utilizando o CryEngine 3. Em 2011, a Crytek liberou a CryEngine 3 para download de forma gratuita, liberado para uso em desenvolvimento sem fins lucrativos.

### **Comparação entre as engines**

A comparação entre várias engines de jogo não é uma tarefa simples, dado ao fato de existirem vários gêneros, tipos, suporte a multimídia, suporte a middleware, dependências de linguagem e plataforma, técnicas de renderização e muitos outros recursos secundários (COWAN; KAPRALOS, 2014; SALEN; ZIMMERMAN, 2012).

Um dos trabalhos mais recentes envolvendo a comparação entre engines considerou seis engines: Cry Engine, Hero Engine, Source Engine, Unity, Unreal Engine 3 e Vision Engine, que eram as mais populares em sua época e utilizou como critério de comparação a plataforma de criação, a linguagem de programação suportada, e qual a biblioteca física utilizada (LEWIS; JACOBSON, 2002).

Neste trabalho, além das características usadas por Lewis e Jacobson (2002) serão considerados: empresas suportadas, estilos gráficos, suporte à realidade aumentada e realidade virtual, e tipos de distribuição, tendo como critério de escolha as engines mais usadas por empresas e desenvolvedores independentes, conforme o Quadro 1.

Quadro 1. Engines escolhidas para o estudo.

Engine
Unity
Construct 2
MonoGame
GameMaker Studio 2
Cocos2d-x
Godot Engine
Unreal 4
Amazon Lumberyard
Libgdx
CryEngine V

### **Suporte das empresas às engines**

Além de utilizar as engines para o desenvolvimento de jogos, muitas empresas também utilizam essa ferramenta para a divulgação de seus produtos no meio comercial criando uma propaganda interativa e divertida para um determinado produto. Um exemplo disso são os jogos da saga Lara Croft - Tomb Raider, cujos primeiros filmes alavancaram o comércio de jogos da saga. Seguindo essa ideia, o prestígio que as empresas fornecem às engines acabam gerando um incentivo para os desenvolvedores dessas ferramentas melhorarem seus softwares .

Os Quadros 2 e 3 elencam as empresas suportadas por cada engine. A partir destes quadros, foi montado um ranking com as engines de maior suporte à empresas. Este ranking pode ser observado na Tabela 1. A ordem em que as engines empatadas aparecem segue a ordem de maior popularidade (da maior popularidade para a menor popularidade).

Podemos observar que a Unity é a engine que possui o maior suporte à empresas dentre as 10 engines mais populares, seguida pela GameMaker



Studio 2 e Cocos2d-x.

Quanto maior o público a que se quer atingir considerando a portabilidade, maior deve ser o número de empresas suportada pela engine. Desta forma, este ranking justifica parte da popularidade da Unity entre as equipes de desenvolvimento de jogos.

A Figura 1 mostra o gráfico com as empresas que têm maior suporte pelas engines. A Microsoft se destaca entre as empresas com o maior número de suporte dentre as engines mais populares, sendo suportada pelas 10 engines mais populares.

Tabela 1. Ranking das engines segunda o seu suporte à empresas.

Posição	Engine	Número de empresas suportadas
1 <sup>a</sup>	Unity	13
2 <sup>a</sup>	GameMaker Studio 2	11
3 <sup>a</sup>	Cocos2d-x	11
4 <sup>a</sup>	Godot Engine	8
5 <sup>a</sup>	Unreal 4	8
6 <sup>a</sup>	Amazon Lumberyard	8
7 <sup>a</sup>	Construct 2	7
8 <sup>a</sup>	MonoGame	6
9 <sup>a</sup>	CryEngine V	6
10 <sup>a</sup>	Libgdx	5

Quadro 2. Suporte à empresas dado pelas engines Unity, Construct, MonoGame, GameMaker Studio 2 e Cocos2d-x.

Unity	Construct 2	MonoGame	GameMaker Studio 2	Cocos2d-x
Google	Microsoft	Microsoft	Adobe	Google
Apple	Google	Ubuntu	Apple	Apple
Facebook	Amazon	Google	Ubuntu	Microsoft
Sony	Kongregate	Sony	Google	Ubuntu
Microsoft	Nintendo	Apple	Microsoft	Samsung
Nintendo	Ubuntu	Nintendo	Amazon	Panasonic
Adobe	Valve		Sony	Intel
Ubuntu			Nintendo	Zynga
Oculus			Raspberry	Glu
Electronic Arts			Valve	Big Fish Games
LEGO			Yoyo games	Disney
Ubisoft				
Square Enix				

Fonte: Elaborado pelo autor.

Quadro 3. Suporte à empresas dado pelas engines Godot Engine, Unreal 4, Amazon Lumberyard, Libgdx e CryEngine V.

Godot Engine	Unreal 4	Amazon Lumberyard	Libgdx	CryEngine V
Microsoft	Microsoft	Microsoft	BlackBerry	Sony
Google	Apple	Sony	Google	Microsoft
Apple	Ubuntu	Twitch	Apple	Amazon
Valve	Nintendo	Apple	Microsoft	Poppermost Productions
Oculus	Valve	Google	Ubuntu	CI Games
Rock Milk	Sony	Amazon		Obsidian Entertainment
Guaranapps	Google	Cloud Imperium Games		
Searchlight Games	Activision	Behaviour Interactive		

Fonte: Elaborado pelo autor.

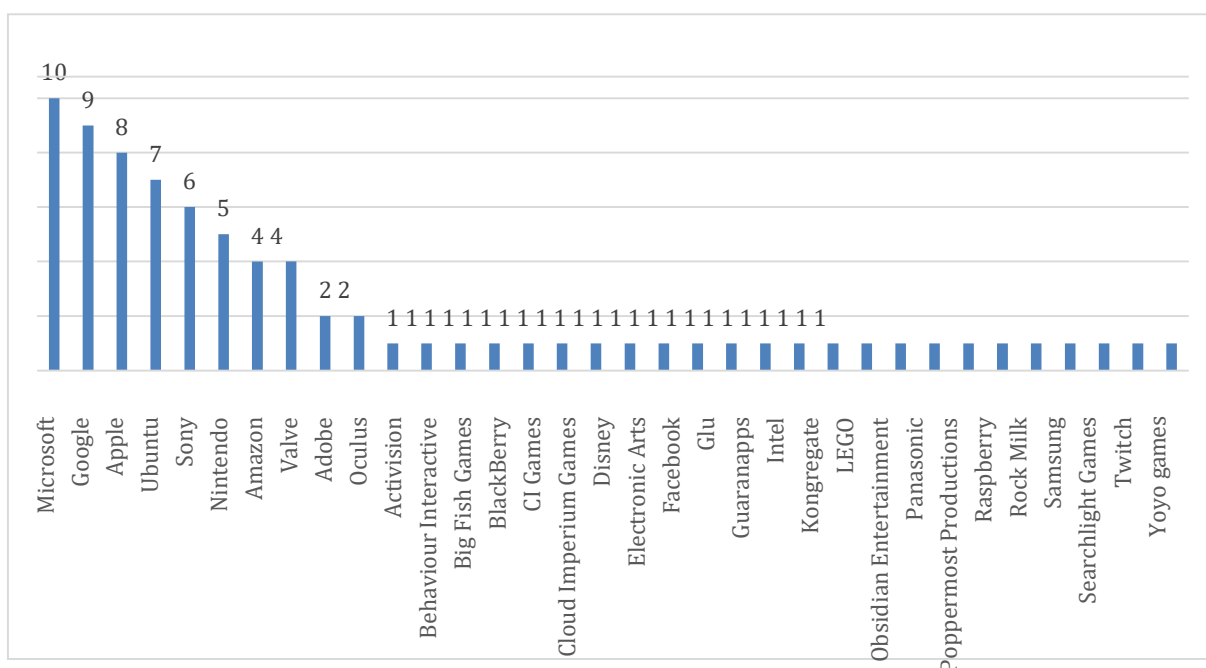


Figura 1. Empresas com maior suporte pelas engines

Fonte: Elaborado pelo autor.

### Plataformas de criação

As plataformas de criação são os ambientes onde a aplicação será desenvolvida. Alguns exemplos de plataforma incluem: smartphones, consoles, computadores, etc.

Com as empresas disponibilizando as suas plataformas para o

desenvolvimento de aplicativos, softwares e jogos, diversas novas empresas surgiram e novas engines foram criadas. Com isso, muitas oportunidades foram abertas para pequenos desenvolvedores, iniciantes e entusiastas, e tiveram mais chances de demonstrar o seu potencial, aumentando drasticamente a procura de desenvolvedores para as mais diversas plataformas.

Os quadros 4 e 5 mostram quais são as plataformas de criação suportadas pelas respectivas engines.

A Tabela 2 resume e ranqueia as engines com o maior número de plataformas suportadas. Mais uma vez a Unity, desta vez empatada com a GameMaker Studio 2, se destaca como a engine com o maior número de plataformas suportadas.

A Figura 2 mostra o gráfico com o ranking das plataformas que possuem suporte do maior número de engines. O Android e o iOS se destacam entre as plataformas mais suportadas.

Quadro 4. Plataformas de criação suportado pelas engines Unity, Construct, MonoGame, GameMaker Studio 2 e Cocos2d-x.

Unity	Construct 2	MonoGame	GameMaker Studio 2	Cocos2d-x
Android	HTML5	TvOS	Adobe Flash	iOS
iOS	Windows SO	Android	Xbox One	Windows SO
Windows SO	Mac OS	Microsoft UWP	Windows SO	Android
PlayStation 4	Linux	Mac OS	Ubuntu	Mac OS
PlayStation 3	Nintendo Wii U	Linux	Android	Linux
BlackBerry 10	Xbox One	PlayStation 4	Mac OS	Tizen
Mac OS		Xbox One	HTML5	
Linux		PlayStation Vita	iOS	
Nintendo Switch		Nintendo Switch	Raspberry Pi	
Nintendo 3DS			PlayStation Portable	
Nintendo Wii U			Nintendo Switch	
Nintendo Wii			PlayStation 4	
Xbox 360			Microsoft UWP	

**Fonte:** Elaborado pelo autor

Quadro 5. Plataformas de criação suportado pelas engines Godot Engine, Unreal 4, Amazon Lumberyard, Libgdx e CryEngine V.

Godot Engine	Unreal 4	Amazon Lumberyard	Libgdx	CryEngine V
Windows SO	Android	Android	Windows	Windows
Android	Mac OS	Windows SO	Mac OS	Xbox One
Mac OS	Mac OS X	Xbox One	Linux	PlayStation 4
WebAssembly	Windows SO	PlayStation 4	Android	
PlayStation 4	Xbox One	iOS	iOS	
Switch	WiiU		BlackBerry	
Xbox One	PlayStation 4		HTML5	
	HTML5			
	Nintendo Switch			

Fonte: Elaborado pelo autor.

Tabela 2. Ranking das engines segundo o seu suporte às plataformas.

Posição	Engine	Número de plataformas suportadas
1 <sup>a</sup>	Unity	13
2 <sup>a</sup>	GameMaker Studio 2	13
3 <sup>a</sup>	MonoGame	9
4 <sup>a</sup>	Unreal 4	9
5 <sup>a</sup>	Godot Engine	7
6 <sup>a</sup>	Libgdx	7
7 <sup>a</sup>	Construct 2	6
8 <sup>a</sup>	Cocos2d-x	6
9 <sup>a</sup>	Amazon Lumberyard	5
10 <sup>a</sup>	CryEngine V	3

Fonte: Elaborado pelo autor.

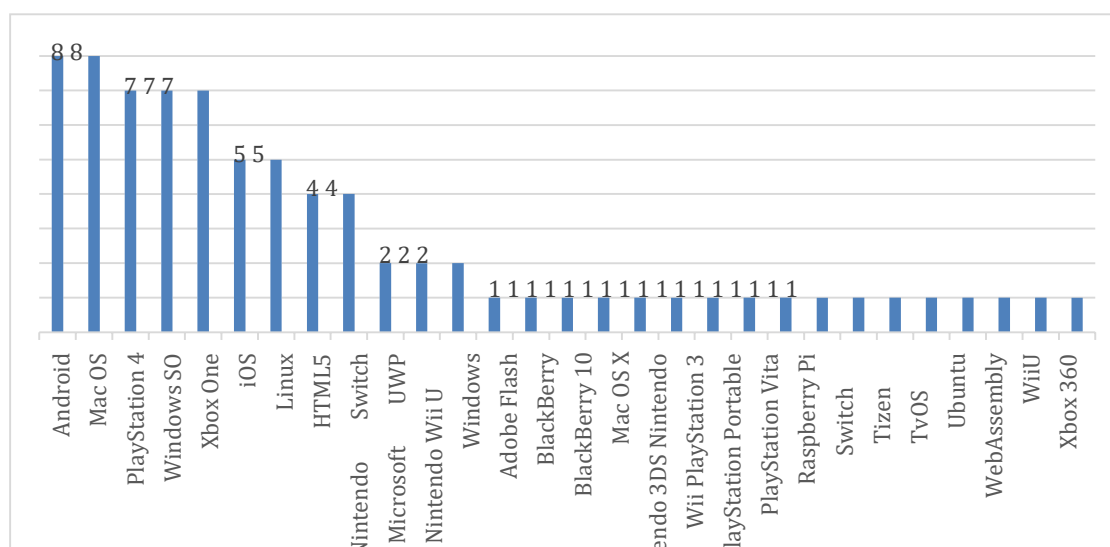


Figura 2. Plataformas com maior suporte pelas engines.

Fonte: Elaborado pelo autor.

## Linguagens de Programação

Para o desenvolvimento completo do jogo, é necessário programar a ação de cada objeto inserido no jogo. Para isso, as engines disponibilizam por padrão suporte à linguagens de programação. À estas linguagens de programação padrão das engines damos o nome de linguagem de programação primária.

Através da criação de jogos e aplicativos e, devido a linguagem mais usada por estudantes de desenvolvimento, criou-se uma abrangência para os mais diversos requisitos atribuídos num programador. Desta forma, algumas engines permitem a codificação do jogo em linguagens de programação que não são sua linguagem primária, como por exemplo a Unity, que permite suporte à linguagem Java, mas cuja compilação é feita em C#, que é a sua linguagem primária.

Os Quadro 6 e 7 reportam as linguagens de programação primárias de cada engine e a Tabela 3 ranqueia as engines com o maior número de linguagens de programação primária suportadas.

Neste novo ranking podemos observar que as engines MonoGame e Cocos2d-x possuem a maior quantidade de linguagem de programação primária suportada.

A Figura 3 mostra o gráfico com a linguagem de programação mais utilizada pelas engines como linguagem primária. A linguagem C++ é a linguagem que mais recebe suporte pelas engines mais populares do mercado.

Quadro 6. Linguagens de programação suportadas pelas engines Unity, Construct, MonoGame, GameMaker Studio 2 e Cocos2d-x.

Unity	Construct 2	MonoGame	GameMaker Studio 2	Cocos2d-x
C#	JavaScript	C#	GML	C++
C++	C++	.Net	Java	HTML
		Java	C++	Lua
		C++		JavaScript

**Fonte:** Elaborado pelo autor.

Quadro 7. Linguagens de programação suportadas pelas engines Godot Engine, Unreal 4, Amazon Lumberyard, Libgdx e CryEngine V.

Godot Engine	Unreal 4	Amazon Lumberyard	Libgdx	CryEngine V
C# 7	C++	C++	Java	C++
C++	BluePrint		C++	C#
GDScript	UnrealScript		C	Lua

**Fonte:** Elaborado pelo autor

Tabela 3. Ranking das engines segundo a quantidade de linguagens.

Posição	Engine	Número de linguagens suportadas
1 <sup>a</sup>	MonoGame	4
2 <sup>a</sup>	Cocos2d-x	4
3 <sup>a</sup>	GameMaker Studio 2	3
4 <sup>a</sup>	Godot Engine	3
5 <sup>a</sup>	Unreal 4	3
6 <sup>a</sup>	Libgdx	3
7 <sup>a</sup>	CryEngine V	3
8 <sup>a</sup>	Unity	2
9 <sup>a</sup>	Construct 2	2
10 <sup>a</sup>	Amazon Lumberyard	1

**Fonte:** Elaborado pelo autor

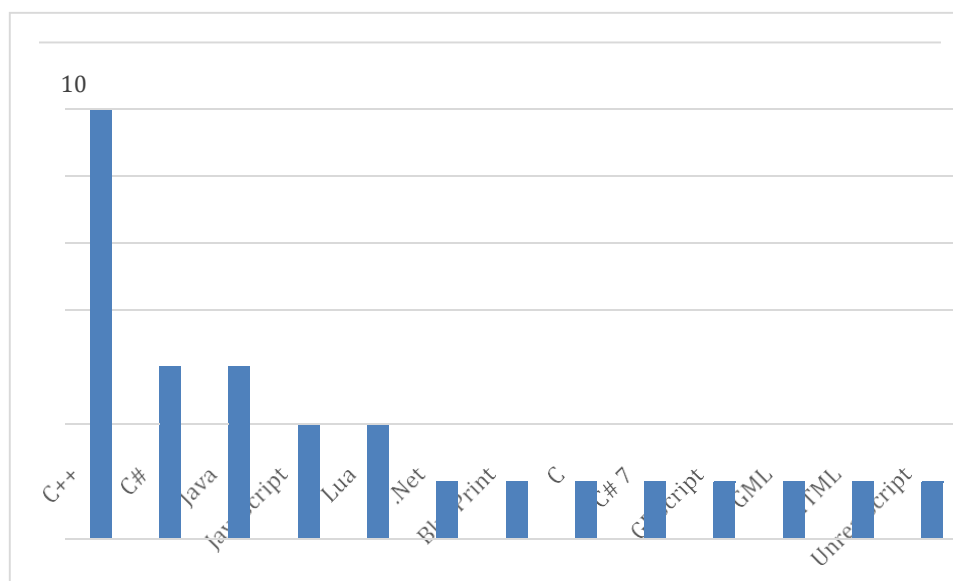


Figura 3. Linguagens com maior suporte pelas engines

**Fonte:** Elaborado pelo autor

## Estilos gráficos e suas demandas

A demanda por estilos gráficos de jogos varia de projeto para projeto. Desta forma, diversas engines utilizam a ideia de poder ser implementada em qualquer tipo de aplicação, não limitando somente a um único estilo gráfico. Todavia, existe também os ambientes exclusivos para propósito ao qual foram desenvolvidos. Sendo assim, o Quadro 8 mostra quais engines suportam estilos gráficos 2D, 3D ou ambos.

Quadro 8. Engines para ambientes 2D e 3D

Engines 2D	Engines 3D	Engines 2D/3D
Construct 2	Amazon Lumberyard	Unity
Cocos2d-x	CryEngine V	Unreal 4
		GameMaker Studio 2
		Godot Engine
		MonoGame
		Libgdx

**Fonte:** Elaborado pelo autor

## Realidade Virtual e Realidade Aumentada

A Realidade Virtual (RV) é vista como uma interface avançada utilizada para acessar aplicações executadas no computador propiciando a visualização, movimentação e interação do usuário, em tempo real, em ambientes tridimensionais gerados por computador (KIRNER; SISCOOTTO, 2007).

A RV é uma das tecnologias que está em expansão nos dias atuais. Ela proporciona a imersão do jogador a um novo mundo utilizando ferramentas específicas como um óculos RV e controles (Figura 4).

E como foi definida por Pimentel e Teixeira (1993): “é o uso da alta tecnologia para convencer o usuário de que ele está em ou?”



Figura 4. Óculos e controle para RV

**Fonte:** Elaborado pelo autor

Os óculos RV funcionam com imagens espelhadas em dois visores posicionados bem próximos aos olhos (Figura 5) fazendo com que o cérebro interprete como uma única imagem e também conta com um giroscópio em que o eixo de rotação se mantém sempre na mesma posição tornando possível a movimentação da cabeça do jogador para se direcionar no mundo virtual. Os controles servem como auxiliares dos óculos, podendo também ter giroscópios.



Figura 5. Espelhos dos óculos RV

**Fonte:** Elaborado pelo autor

A Realidade Aumentada (RA) é definida como a sobreposição de objetos virtuais no mundo real, por meio de um dispositivo tecnológico, melhorando ou aumentando a visão do usuário (AZUMA, 1997; BAJURA, 1995; KIRNER, 2004).

Além de permitir essa sobreposição de objetos virtuais no mundo real, a RA também permite o manuseio destes objetos com as próprias mãos, possibilitando que o usuário tenha uma interação atrativa e motivadora com o ambiente (BILLINGHURST, 2001; KIRNER, 2004; SANTIN, 2008).

A RA funciona ao contrário da realidade virtual, pois o jogo é realizado no



mundo real por meio de uma tela, como a de um smartphone, por exemplo. Um exemplo deste modelo de jogo é o Pokémon GO, que utiliza a câmera do smartphone para sincronizar e projetar suas criaturas no nosso mundo em determinados pontos da região (Figura 6).

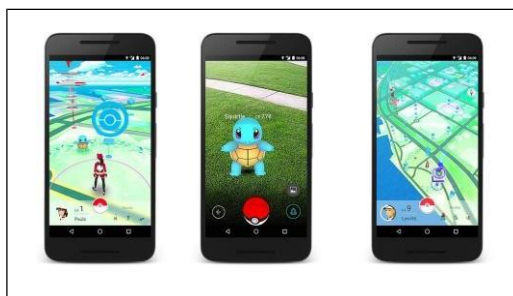


Figura 6. Ilustração do Pokémon GO

**Fonte:** Elaborado pelo autor

Tanto a RV, quanto a RA vêm ganhando popularidade no mercado de jogos. Acompanhando esta popularidade, as engines atuais já dão suporte ao uso destas tecnologias em seu ambiente de desenvolvimento de jogo.

O Quadro 9 mostra como é o suporte das engines com relação ao suporte à RV e RA.

Quadro 9. Engines com suporte à RV e RA.

Engines VR	Engines AR
Unity	Unity
Unreal 4	Unreal 4
Amazon Lumberyard	Amazon Lumberyard
CryEngine V	

**Fonte:** Elaborado pelo autor

### Versões para desenvolvimento

As versões das engines influenciam no desenvolvimento dos projetos que irão ser elaborados e várias das empresas responsáveis criam mais de uma versão da engines sendo que algumas poderão ser bem distintas umas das outras. Como exemplo de versões, temos algumas que podem ser pagas e que incluem funções extras e ferramentas que facilitam no desenvolvimento. Por conta disso, cabe ao desenvolvedor decidir com qual ferramenta deverá trabalhar para facilitar e adequar as suas necessidades.

Os Quadros 10 e 11 mostram as versões gratuitas de cada engines, ao passo que o Quadro 12 mostra as versões pagas de algumas engines que cobram por funcionalidades extras.

Quadro 10. Versões gratuitas das engines Unity, Construct, MonoGame, GameMaker Studio 2 e Cocos2d-x.

	Unity	Construct 2	MonoGame	GameMaker Studio 2	Cocos2d-x
Pessoal	✓	✓			
Educacional		✓		✓	
Mínimo					✓
Código aberto			✓		✓
Grátis			✓		✓
Amostra Grátis				✓	
Completo					✓

**Fonte:** Elaborado pelo autor

Quadro 11. Versões gratuitas das engines Godot Engine, Unreal 4, Amazon Lumberyard, Libgdx e CryEngine V

	Godot Engine	Unreal 4	Amazon Lumberyard	Libgdx	CryEngine V
Pessoal					
Educacional					
Mínimo					
Código aberto	✓			✓	
Grátis	✓	✓	✓	✓	✓
Amostra Grátis					
Completo					

**Fonte:** Elaborado pelo autor

Quadro 12. Engines com versões pagas

	Unity	Construct 2	GameMaker Studio 2
Criador - \$			✓
Desenvolvedor - \$			✓
Console - \$			✓
Profissional - \$	✓		
Plus - \$	✓	✓	
Empresarial - \$		✓	

**Fonte:** Elaborado pelo autor

## Discussões

Conforme visto na seção anterior, a análise das 10 engines mais populares envolve características relevantes ao processo de escolha por determinada engine. No intuito de fazer uma análise global, foram utilizados alguns dados desta pesquisa para montar o gráfico Radar destas engines conforme a Figura 7.

Neste gráfico foram considerados os suportes às empresas, às plataformas e à linguagens, que são as características utilizadas para montar os 3 rankings da Seção 4, mostrados nas Tabelas 1, 2 e 3 respectivamente.

O gráfico da Figura 7 nos mostra uma inclinação de destaque para a engine Unity. Embora o GameMaker Studio 2 dê suporte ao mesmo número de plataformas de criação, a Unity vence em quantidade de empresas suportadas. Isso a coloca em vantagens em termos de alcance que os projetos realizados nesta engine poderão alcançar. Como ponto negativo, a quantidade de linguagens de programação primária suportada pela Unity está entre as menores, diminuindo a liberdade de escolha entre os programadores.

A Unity pode servir como uma ferramenta adequada para quem está começando a trabalhar com desenvolvimento de jogos. Além de sua abrangência, esta engine oferece uma versão gratuita para fins pessoais, provendo recursos profissionais de maneira acessível.

Da mesma forma, a engine GameMaker Studio 2 apresentou boas colocações no ranking. Embora tenha empatado com relação ao número de plataformas de criação e perdido para a Unity no critério de suporte à empresas, tem uma linguagem de programação primária a mais que a Unity. Esse pode ser um fator decisivo para quem já tem conhecimento prévio em JAVA ou GML.

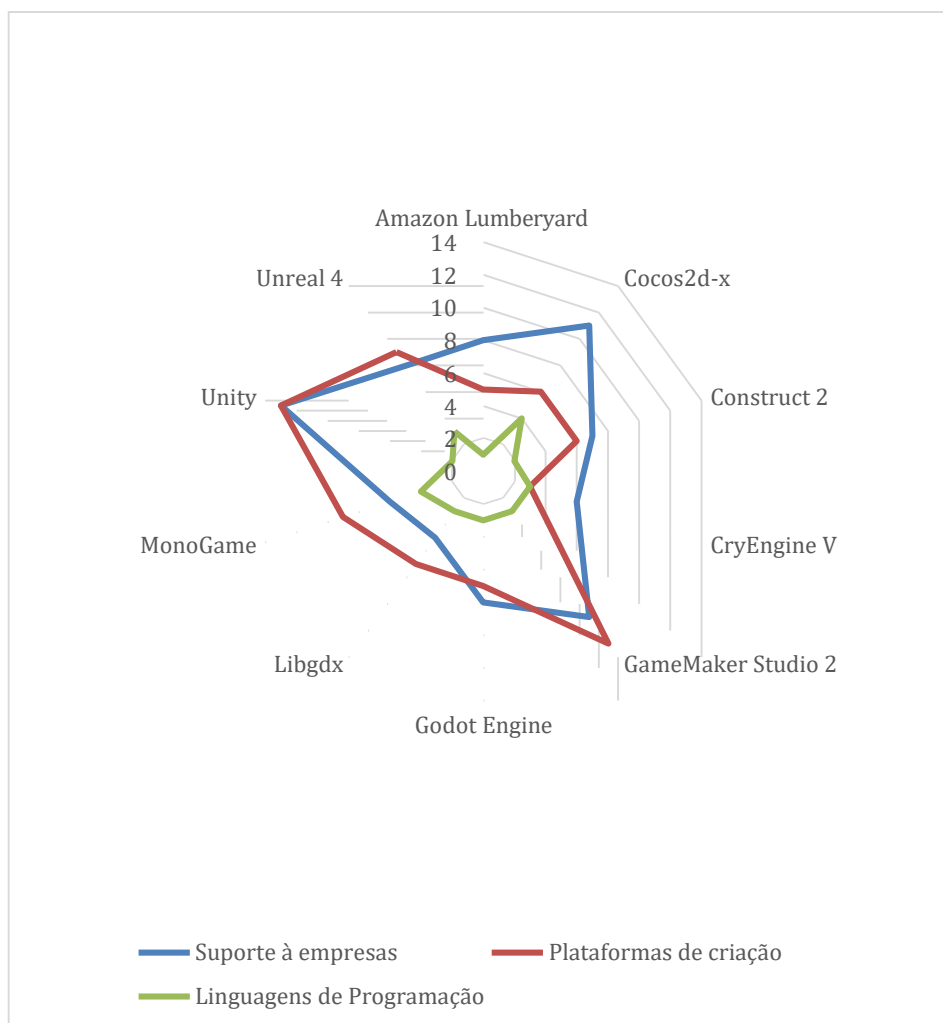


Figura 7. Análise global das engines

Fonte: Elaborado pelo autor

## Considerações Finais

Este trabalho propôs fazer uma análise comparativa entre as 10 engines mais populares do mercado, podendo servir como uma ferramenta auxiliar no processo de escolha por uma engine entre os desenvolvedores de jogos.

Após identificar quais são as 10 engines mais populares atualmente, a pesquisa concluiu que a Unity possui o maior número de características consideradas importantes no processo de escolha.

Outra engine, a o GameMaker Studio 2, também apresentou-se como uma boa opção de escolha para quem procura um ambiente de desenvolvimento completo de jogos.

Como trabalhos futuros, pretende-se analisar a performance computacional destas 10 engines em duas abordagens: 1) o custo computacional (gasto de memória e processamento) em tempo de desenvolvimento; e 2) o custo computacional dos jogos gerados por estas engines.

## Referências

- AMAZON LUMBERYARD, Amazon Lumberyard – **Game Engine**. [aws.amazon.com/pt/lumberyard/](https://docs.aws.amazon.com/lumberyard/), 26 de outubro de 2018. Disponível em: <<https://docs.aws.amazon.com/lumberyard/index.html>>. Acesso em: 26 out. 2018
- AZUMA, R. T. A survey of augmented reality. **Presence: Teleoperators and virtual environments**, v. 6, n. 4, p. 355-385, 1997.
- BAJURA, M.; NEUMANN, U. Dynamic Registration Correction in Video-Based Augmented Reality Systems. **IEEE Computer Graphics & Applications**, v. 15, n. 5, p. 52-60, 1995.
- BATTAIOLA, A. L. et al. Desenvolvimento da Interface de um Software Educacional com base em Interfaces de Jogos. IHC Simpósio sobre Fatores Humanos em Sistemas Computacionais, v. 5, 2002.
- BILLINGHURST, M.; DUENSER, A. Augmented reality in the classroom. **Computer**, v. 45, n. 7, p. 56-63, 2012.
- BISHOP, L. et al. Designing a PC game engine. **IEEE Computer Graphics and Applications**, v. 18, n. 1, p. 46-53, 1998.
- CHAUDY, Y.; CONNOLLY, T. Specification and evaluation of an assessment engine for educational games: Empowering educators with an assessment editor and a learning analytics dashboard. **Entertainment Computing**, v. 27, p. 209-224, 2018.
- CLUA, E. W. G.; BITTENCOURT, J. R. Desenvolvimento de jogos 3D: concepção, design e programação. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA (JAI); CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, XXIV., 2005. p. 22-29.
- COCOS2D-X, Cocos2d-x – Game Engine. [cocos2d-x.org/](https://docs.cocos2d-x.org/), 30 de setembro de 2018. Disponível em: <<http://docs.cocos2d-x.org/creator/manual/en/getting-started/cocos2d-x-guide.html>>. Acesso em: 30 set. 2018.
- CONSTRUCT-2 Construct-2 – Game Engine. [scirra.com](https://www.scirra.com/), 05 de outubro de 2018. Disponível em: <<https://www.scirra.com/manual/1/construct-2>>. Acesso em: 5 out. 2018.

COWAN, B.; KAPRALOS, B. A survey of frameworks and game engines for serious game development. In: **ADVANCED LEARNING TECHNOLOGIES (ICALT), 2014 IEEE 14th INTERNATIONAL CONFERENCE ON**. IEEE, 2014. p. 662-664.

CRYENGINE V. **Game Engine**. cryengine.com, 16 de outubro de 2018.  
Disponível em:  
<<https://docs.cryengine.com/display/CEMANUAL/CRYENGINE+V+Manual>>.  
Acesso em: 16 out. 2018.

DARKEN, R.; MCDOWELL, P.; JOHNSON, E. Projects in VR: The Delta3D open source game engine. **IEEE Computer Graphics And Applications**, v. 25, n. 3, p. 10-12, 2005.

ESA. Essential facts about the computer and video game industry. **Retrieved August**, v. 16, p. 2006, 2002.

FRIESE, K-W.; HERRLICH, M.; WOLTER, F-E. Using game engines for visualization in scientific applications. In: **New Frontiers for Entertainment Computing**. Springer, Boston, MA, 2008. p. 11-22.

FRITSCH, D. et al. Visualisation using game engines. **Archiwum ISPRS**, v. 35, p. B5, 2004.

GAMEMAKER STUDIO 2. **Game Engine**. yoyogames.com, 29 de setembro de 2018. Disponível em: <[https://docs2.yoyogames.com/source/\\_build/index.html](https://docs2.yoyogames.com/source/_build/index.html)>.  
Acesso em: 29 set. 2018.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre: [s.n.], 2009.

GODOT ENGINE. **Game Engine**. godotengine.org, 02 de outubro de 2018.  
Disponível em:  
<[http://docs.godotengine.org/pt\\_BR/latest/getting\\_started/editor/](http://docs.godotengine.org/pt_BR/latest/getting_started/editor/)>. Acesso em: 02 out. 2018.

HARDY, S. et al. Framework for personalized and adaptive game-based training programs in health sport. **Multimedia Tools and Applications**, v. 74, n. 14, p. 5289- 5311, 2015.

HERWIG, A.; PAAR, P. Game engines: tools for landscape visualization and planning. **Trends in GIS and virtualization in environmental planning and design**, v. 161, p. 172, 2002.

HUDLICKA, E. Affective game engines: motivation and requirements. In: **Proceedings of the 4th international conference on foundations of digital games**. **ACM**, 2009. p. 299- 306.

INDRAPRASTHA, A.; SHINOZAKI, M. The investigation on using Unity3D game engine in urban design study. **Journal of ICT Research and**

**Applications**, v. 3, n. 1, p. 1-18, 2009.

KINER, C. **Projeto sistema complexo aprendente: um ambiente de realidade aumentada para educação (SICARA)**. 2007. Disponível em: <<http://www.ckirner.com/claudio/?PROJETOS:SICARA>>. Acesso em: 2 out. 2018.

KIRNER, C.; SISCOOTTO, R. A. **Realidade Virtual e Aumentada**. 2007. Disponível em: <<http://www.ckirner.com/download/livros/Livro-RVA2007-1-28.pdf>>. Acesso em: 2 out. 2018.

KOT, B. et al. Information visualisation utilising 3D computer game engines case study: a source code comprehension tool. In: ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural, 6., **Proceedings...**, 2005. p. 53-60.

LEWIS, M.; JACOBSON, J. Game engines. **Communications of the ACM**, v. 45, n. 1, p. 27, 2002.

LIBGDX, Libgdx – Game Engine. libgdx.badlogicgames.com, 29 de outubro de 2018. Disponível em: <<https://libgdx.badlogicgames.com/documentation/help/Documentation.html>>. Acesso em: 29 out. 2018.

MARKS, S.; WINDSOR, J.; WÜNSCHE, B. Evaluation of game engines for simulated surgical training. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES IN AUSTRALIA AND SOUTHEAST ASIA, 5., **Proceedings...**, 2007. p. 273-280.

MÓL, A. C. A.; JORGE, C. A. F.; COUTO, P. M. Using a game engine for VR simulations in evacuation planning. **IEEE Computer Graphics and Applications**, v. 28, n. 3, p. 6-12, 2008.

MONOGAME. **Game Engine**. monogame.net, 02 de outubro de 2018. Disponível em: <<http://www.monogame.net/documentation/?page=main>>. Acesso em: 2 out. 2018.

PASSOS, E. B. et al. Tutorial: desenvolvimento de jogos com unity 3d. In: BRAZILIAN SYMPOSIUM ON GAMES AND DIGITAL ENTERTAINMENT, VIII., 2009. p. 1-30.

PETRIDIS, P. et al. Game engines selection framework for high-fidelity serious applications. International Journal of Interactive Worlds, p. Article ID 418638, 2012.

PIMENTEL, K.; TEXEIRA, K. **Virtual Reality: through the new looking glass**. New York: McGraw Hill, 1993.

SALEN, K.; ZIMMERMAN, E. **Regras do jogo: fundamentos do design de jogos**. São Paulo: Blucher, 2012. V. 1.

SANTIN, R. **SACRA - Sistema de Autoria em Ambiente Colaborativo com Realidade Aumentada**. 2008. Disponível em: <<https://www.unimep.br/phpg/bibdig/pdfs/2006/DOIKPNGYIQHP.pdf>>. Acesso em: 2 out. 2018.

TRENHOLME, D.; SMITH, S. P. Computer game engines for developing first-person virtual environments. **Virtual reality**, v. 12, n. 3, p. 181-187, 2008.

UNITY3D. **Game Engine**. 26 de setembro de 2018. Disponível em: <<https://docs.unity3d.com/Manual/index.html>>. Acesso em: 26 set. 2018.

UNREAL 4. **Game Engine**. unrealengine.com. 11 de outubro de 2018. Disponível em: <<https://docs.unrealengine.com/en-us/>>. Acesso em: 11 out. 2018.