

Introdução à jornada game dev.

Em uma ferramenta
especializada
(*Godot Engine*)



Olá mundo!

De amador, para amadores!

Objetivo

Minicurso dividido em quatro partes:

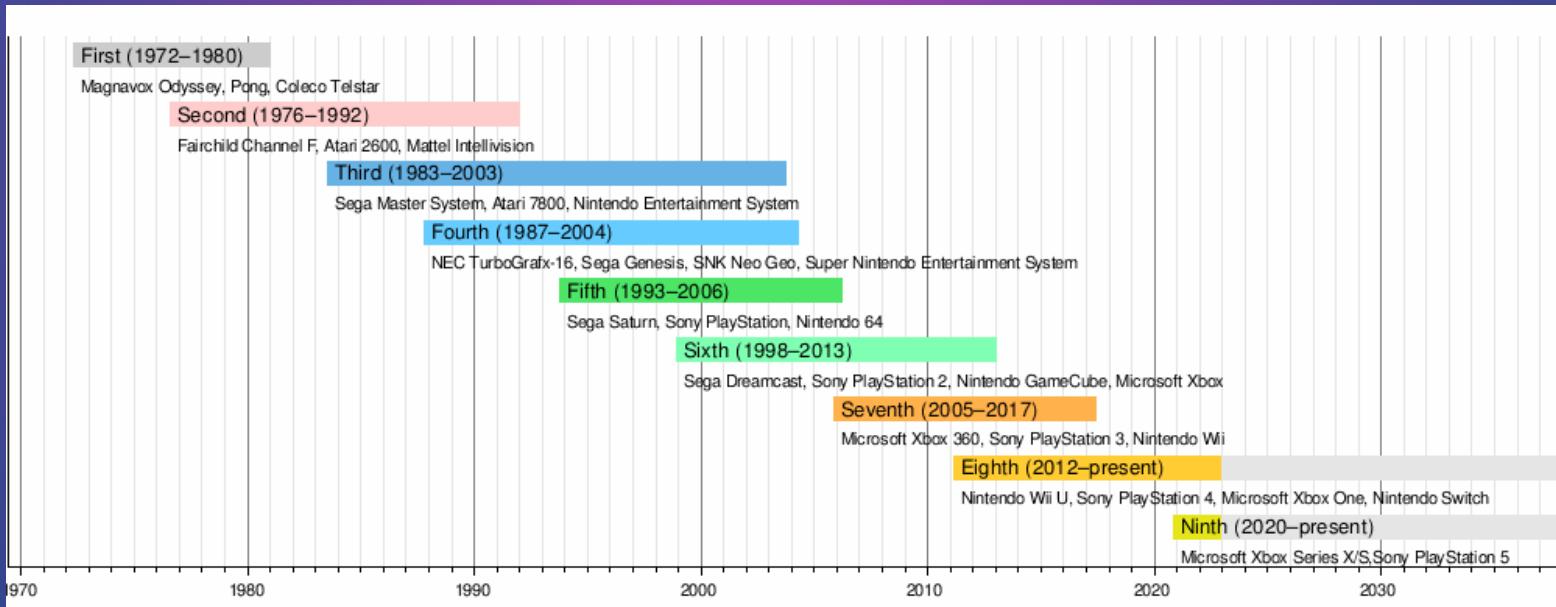
- I. Introdução expositiva
- II. Apresentação teórica de conceitos de programação de jogos
- III. Oficina teórica da linguagem de programação de jogos Godscript
- IV. Oficina prática usando o motor de jogos Godot desenvolvendo o jogo Flappy Bird (no laboratório)

O que nós iremos explorar

Como desenvolver jogos:

- Ferramentas
- Tópicos relevantes
- Conceitos fundamentais
- Como programação de jogos se relaciona com os conceitos da Ciência da Computação
- Como usar esses conceitos para participar de competições
- E as diversas possibilidades desse campo de trabalho artístico...

Uma breve viagem na história dos jogos...



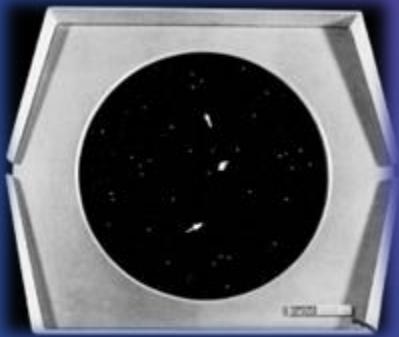
“Stone Age” dos games

- Jogos esportivos
- Ludo-narrativas (“faz de conta”)
- Jogos abstratos (pen & paper e table-top)
- RPGs – Jogos de interpretação de papéis
 - Parques temáticos

1º Geração



Tennis for two (1958)



Spacewar (1962)



PDP-1 (1962)

1º Videogame?



Steve Russell



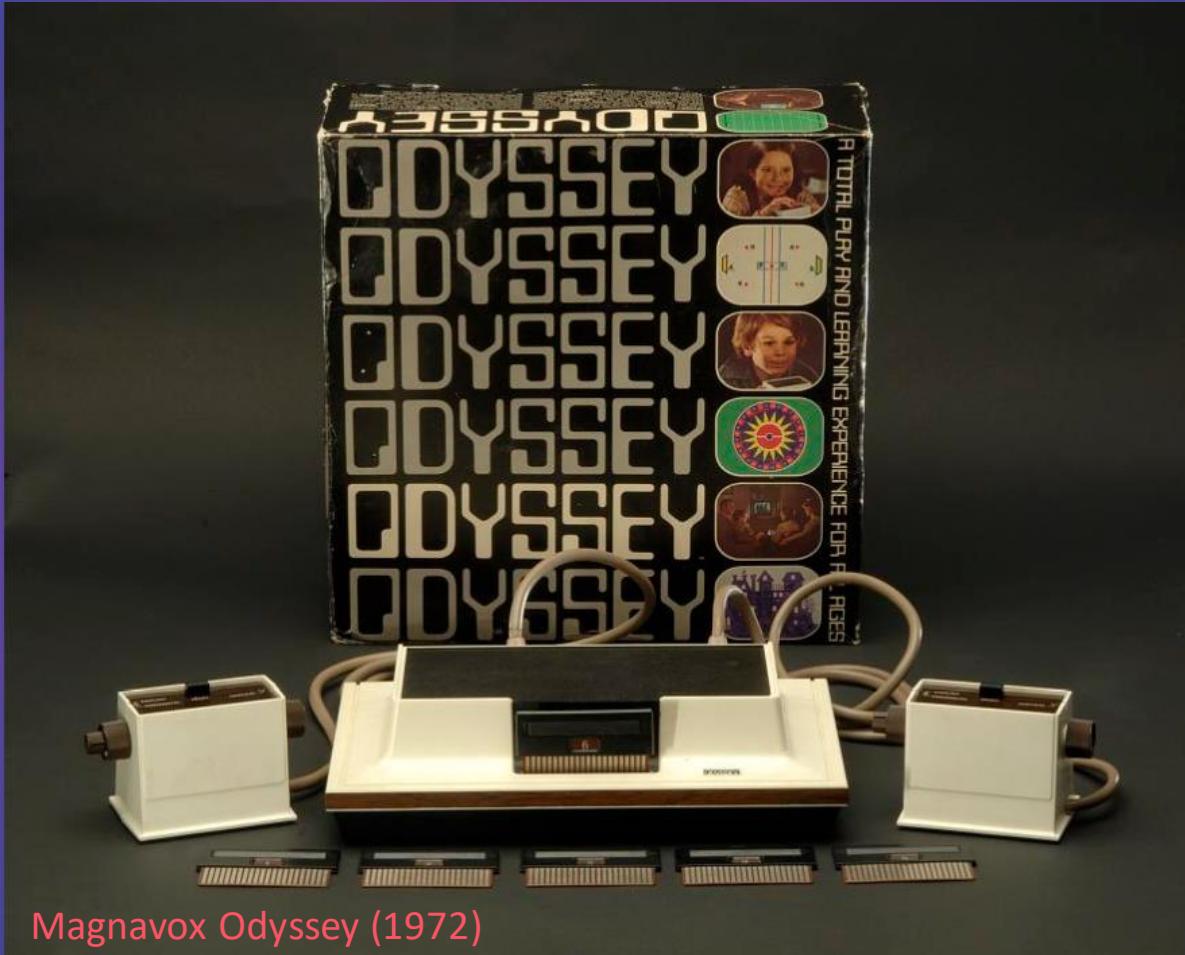
1º Geração



Maze War (1973)
1º multiplayer FPS



1º "home console"



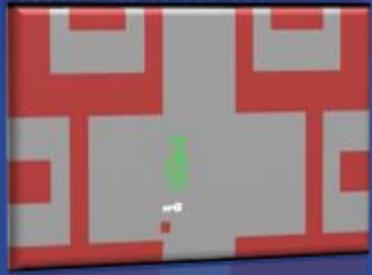
Magnavox Odyssey (1972)



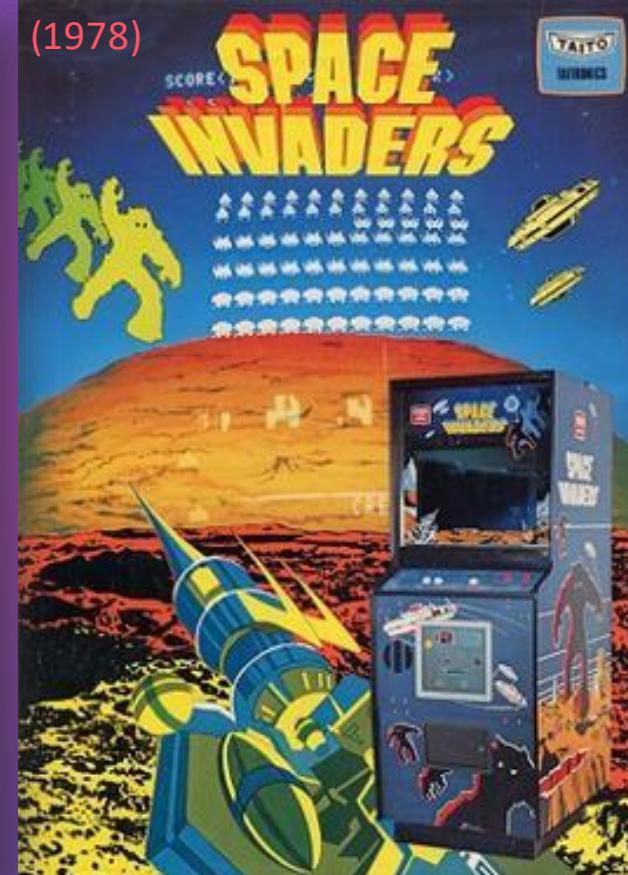
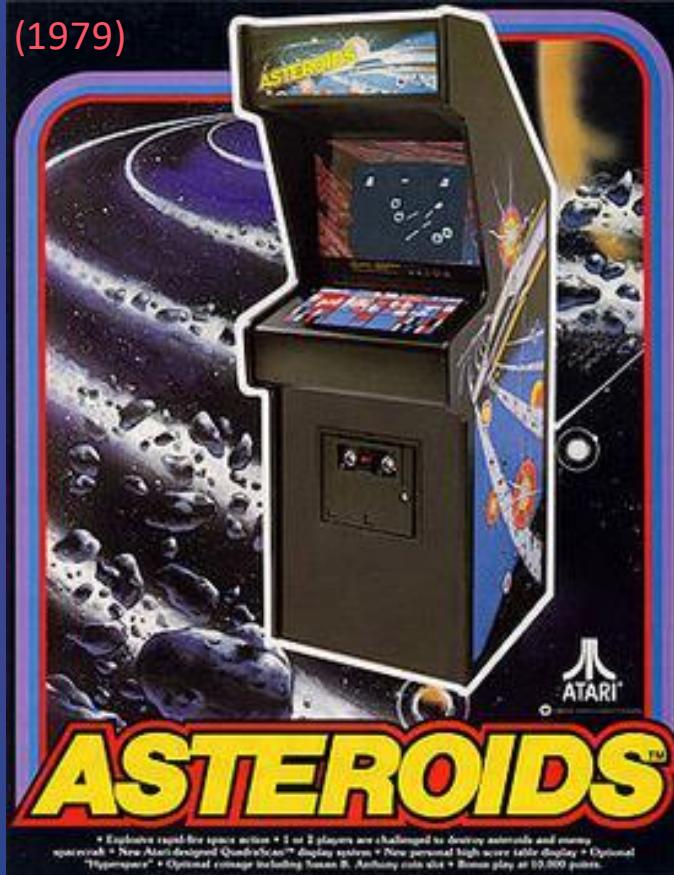
2º Geração



Pacman



Adventure

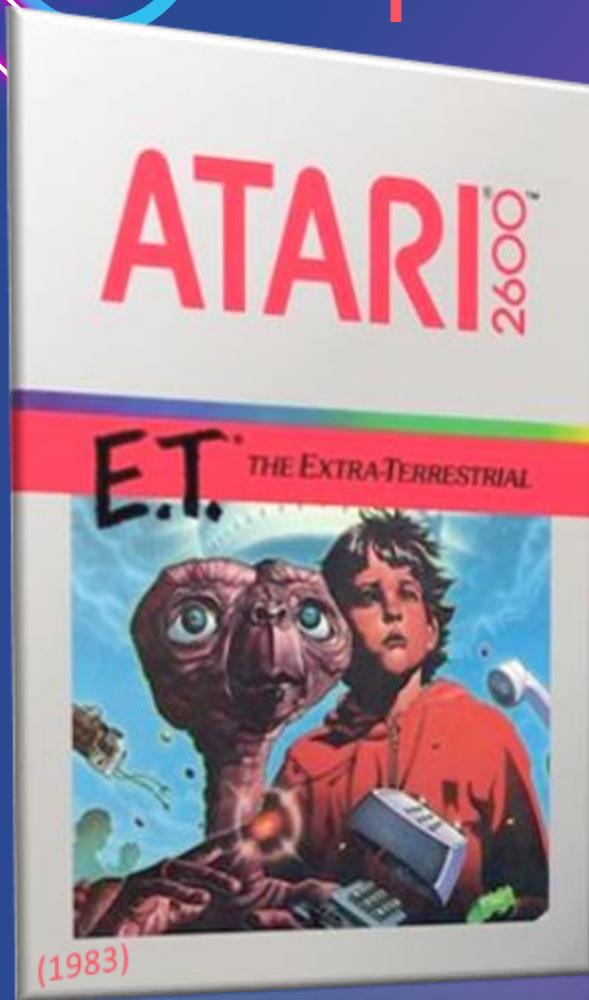


Atari 2600



Magnavox Odyssey 2

2º Geração



Crash dos games de 83





+

SEGA®
Master System™



Light Phaser

SEGA Scope
3D-glasses



Phantasy Star



(1987)



Alex Kid (1986)

3º Geração

Geração 8-Bit

+

Nintendo®
ENTERTAINMENT
SYSTEM



NES



Famicom



FAMILY COMPUTER
ファミリーコンピュータ



3º Geração



Super Mario Bros (1983)



The Legend of Zelda (1986)



Takashi Tezuka
(Game Designer)

Shigeru Miyamoto
(Game Designer)

Koji Kondō
(Composer)

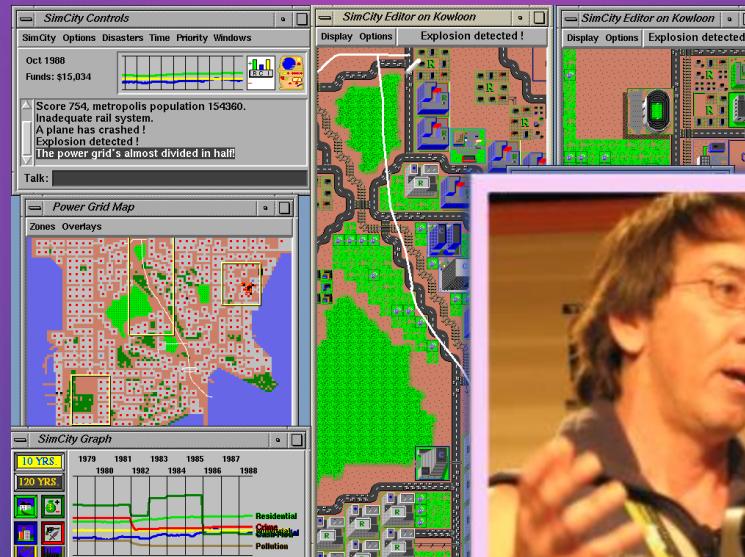
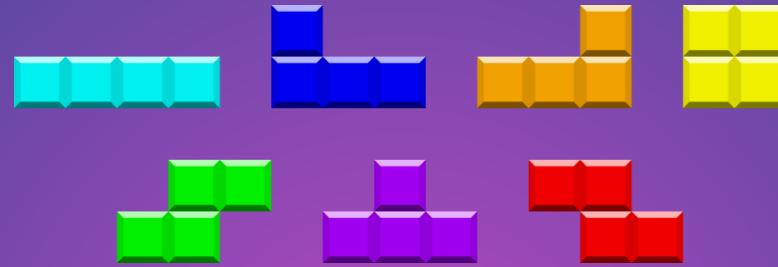
3º/4º Geração



Pokémon Red/Blue/
Yellow (1996)



Neverwinter Nights
(1991-1997) - 1º MMO



Sim City (1989)



Will Wright



Mystery House (1980)

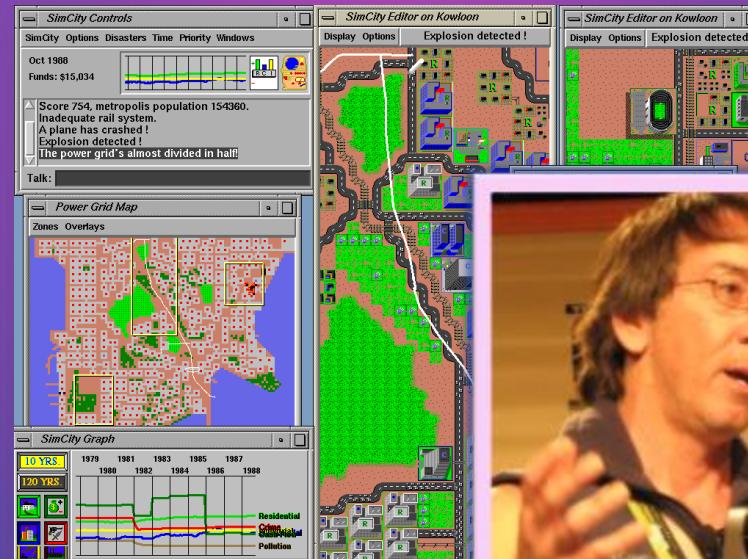
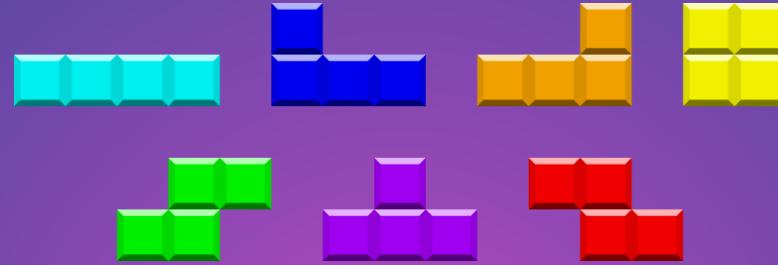
3º/4º Geração



Pokémon Red/Blue/
Yellow (1996)



Neverwinter Nights
(1991-1997) - 1º MMO



+

Sim City (1989)



Will Wright



Mystery House (1980)

4º Geração



4º Geração



MORTAL KOMBAT 2 (1992)



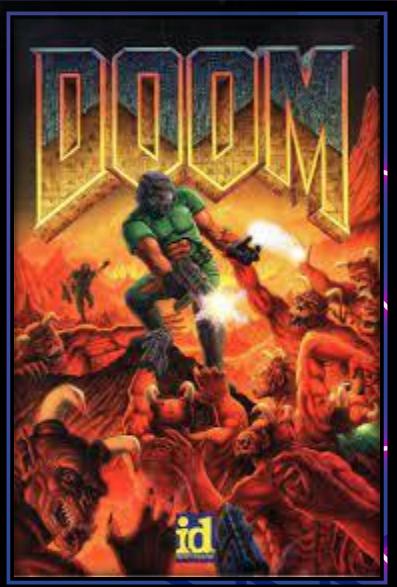
MORTAL KOMBAT 3 (1995)



STREET FIGHTER 2 (1992)



4°/5° Geração



5º Geração



+



+



+

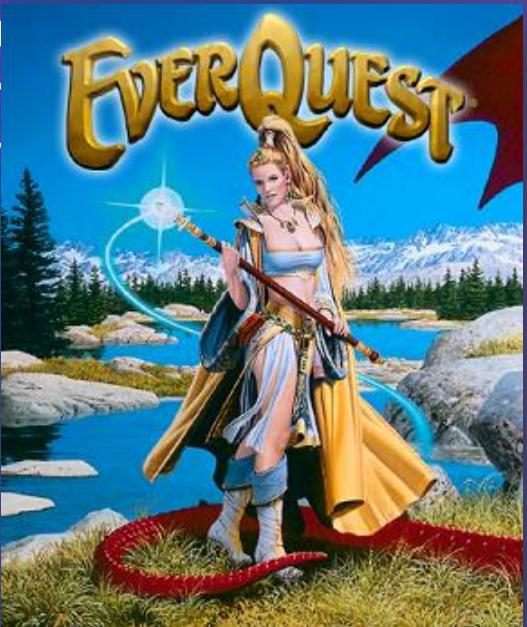


+



5°/6° Geração

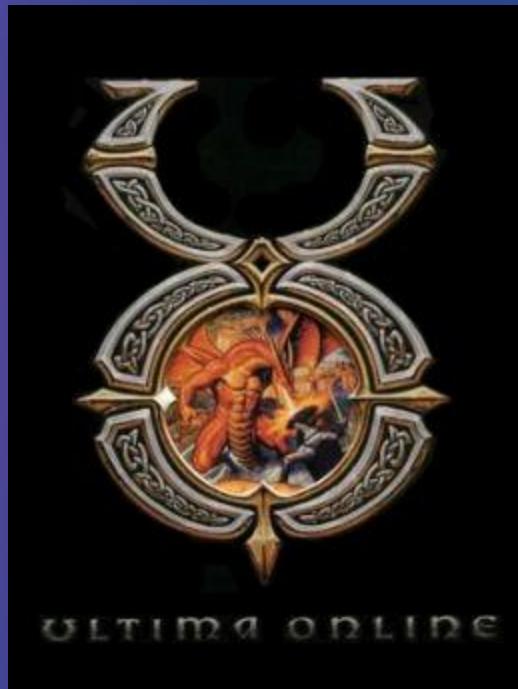
Rise of MMOs



(1999-)



(1997-)



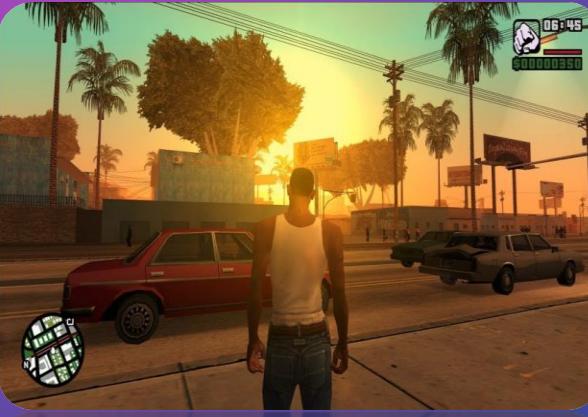
6º Geração



CALL OF DUTY 3 (2006)



DBZ BUDOKAI TENKAICHI 3 (2007)



GTA SAN ANDREAS (2004)



RESIDENT EVIL 4 (2005)



NEED FOR SPEED MOST
WANTED (2012)

6º Geração



+

SONIC ADVENTURE 2 (2001)



+

LUIGI'S MANSION (2001)



+

HALO: COMBAT EVOLVED (2001)



+



7º Geração



7º Geração





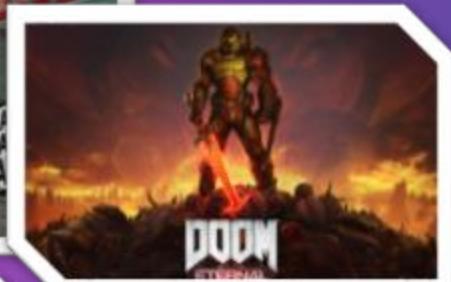
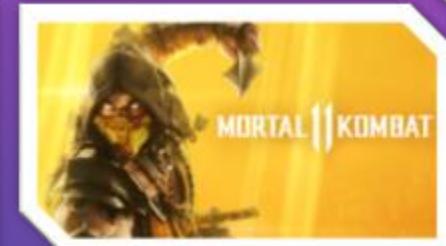
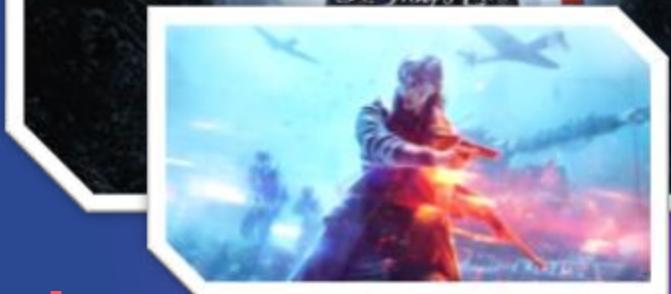
THE GAME AWARDS



8º Geração



8º Geração



9º Geração

Geração atual

RED DEAD REDEMPTION 2



CYBERPUNK 2077



ZELDA BREATH OF THE WILD



GOD OF WAR RAGNAROK



THE LAST OF US PART 2



Geração atual

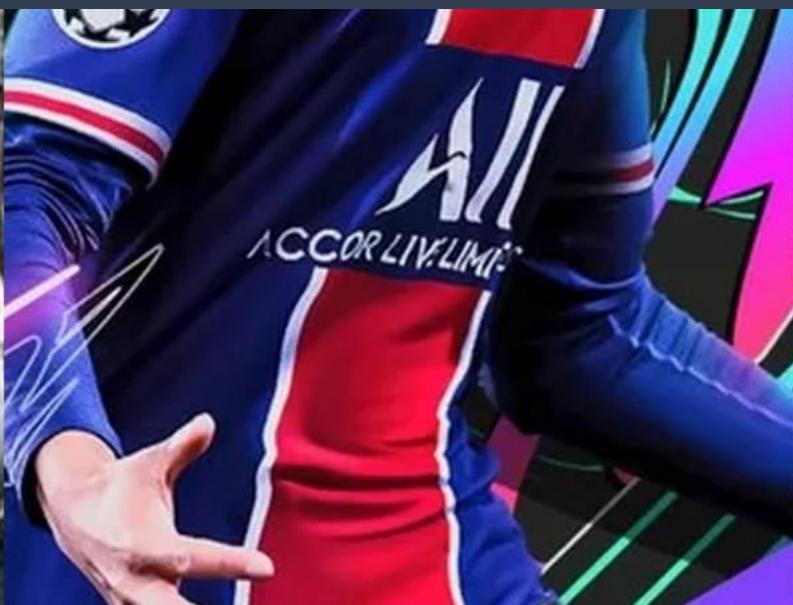
- A indústria em 2023
- Foco na imersão e realismo
- Jogos independentes tendo mais relevância
 - Sede por jogos artísticos vs. o brilho ofuscante da tecnologia imersiva

Futuro dos Games

- Tecnologia: alta fidelidade e refinamento (simulações em tempo real)
 - Foco na imersão: VR & AR
 - Jogos no cotidiano: Gamification
- Geração Procedural e Inteligências Artificiais em jogos
 - A volta do lúdico e artístico?
Ou a singularidade do capitalismo?

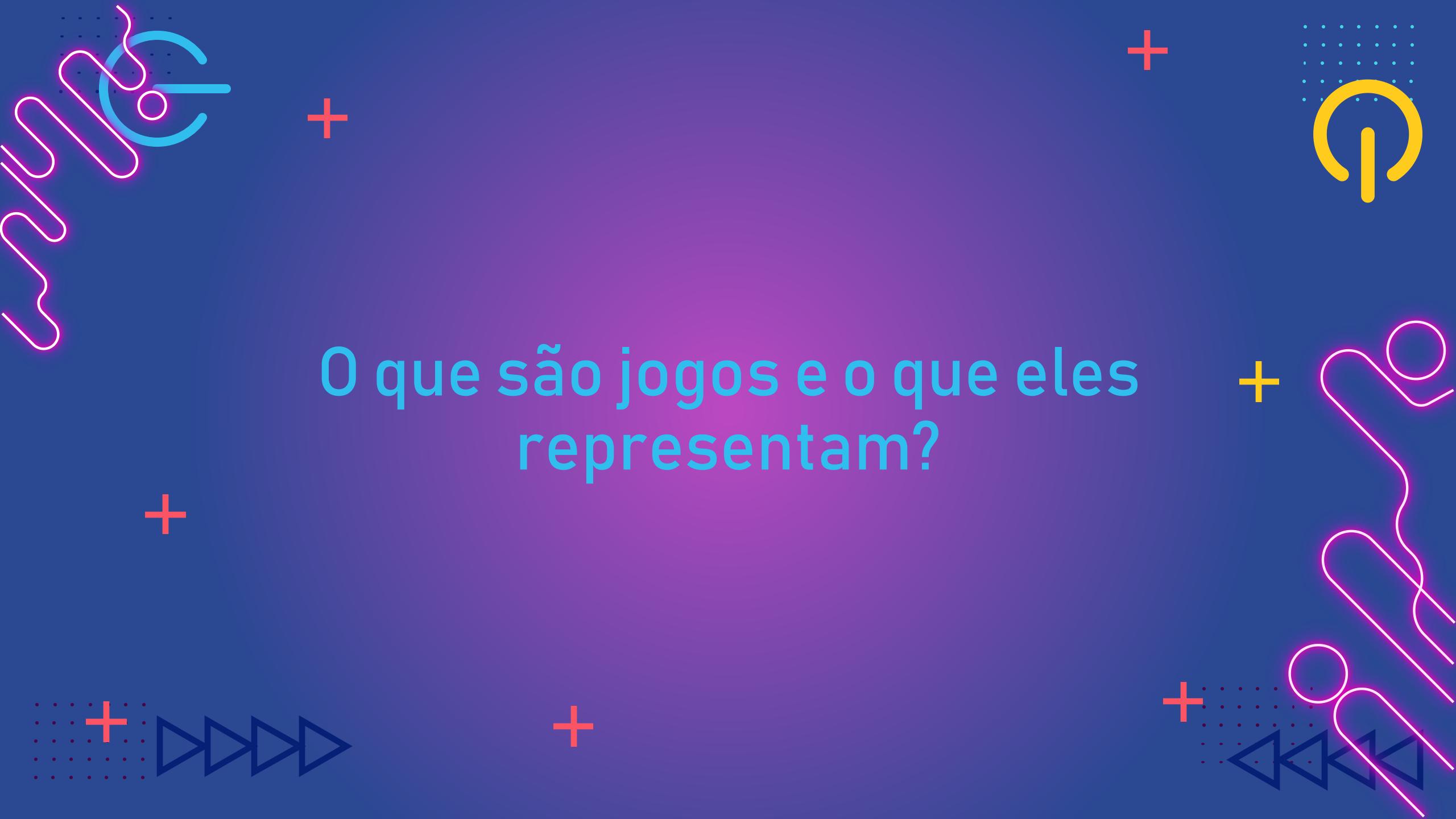


•A singularidade dos jogos AAA?

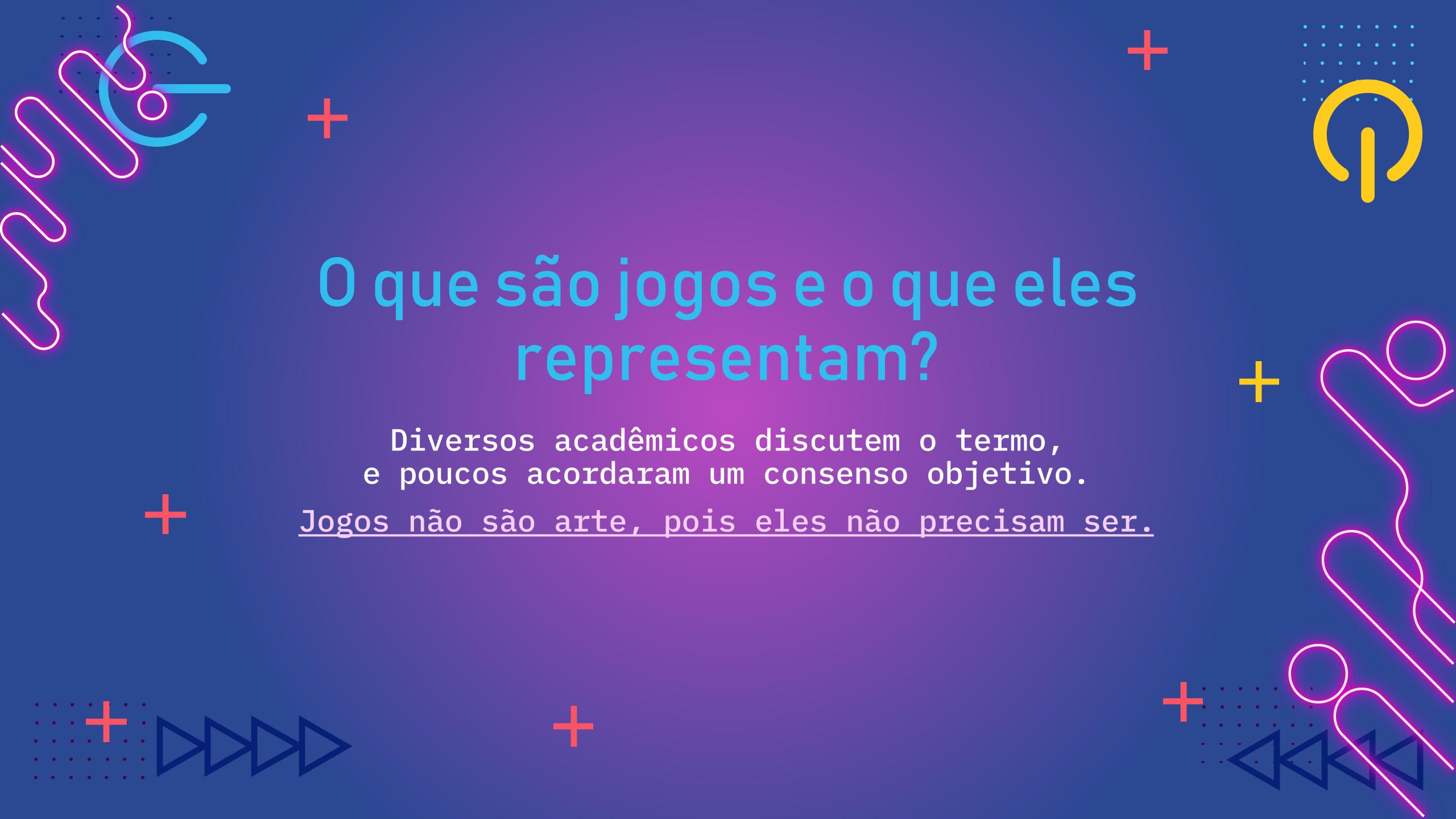




• A volta do lúdico e artístico?



O que são jogos e o que eles representam?



O que são jogos e o que eles representam?

Diversos acadêmicos discutem o termo,
e poucos acordaram um consenso objetivo.

Jogos não são arte, pois eles não precisam ser.

O que são jogos e o que eles representam?

- Pensando sobre jogos em um nível mais fundamentalista
 - Arte & Design
 - Tecnologia & Inovação
- “Expressividade cultural” vs. “obstáculo intelectual”
- A maratona criativa e intelectual no mercado de jogos.
Expectativas: the Hollywood of gaming



Exemplos & características

- Categorias de jogos;
- Tecnologias.





Categorias de jogos digitais

- Jogos educacionais e sérios;
- Jogos de larga escala (AAA), de estúdios medianos e independentes;
- Plataformas: Mobile, AR/VR, console, PC e híbrido;





Jogos educacionais & “serious games”

- Jogos que proporcionam valor de aprendizado ou treinamento a um jogador.^[1]
- “*Edutainment*”^[2] descreve uma mescla intencional de videojogos e softwares educacionais em um único produto





A fantástica viagem dos zoombinis







Jogos independentes (indie)

- Jogos criados por indivíduos ou pequenos times de desenvolvimento
- Geralmente sem a ajuda financeira e suporte técnico de uma *publisher*^[8] (publicadora).^[11]
- O termo indie pode ser aplicado onde o desenvolvimento do jogo tem algumas medidas de **independência** da publicadora, mesmo se uma publicadora ajuda no financiamento e distribuição do jogo.
- Vantagens:
 - liberdade criativa;
 - foco na inovação;
 - jogabilidade experimental;
 - tomada de riscos;
 - jogos artísticos.
- Indie games tendem a ser vendidos por canais digitais de distribuição ao invés de varejo.
- O termo é equivalente à música independente^[9] ou filme independente^[10] nesses respectivos meios.





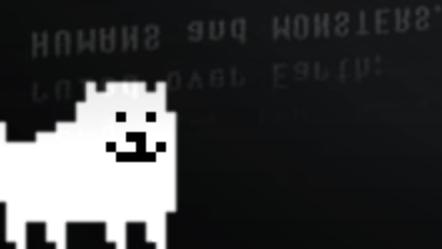
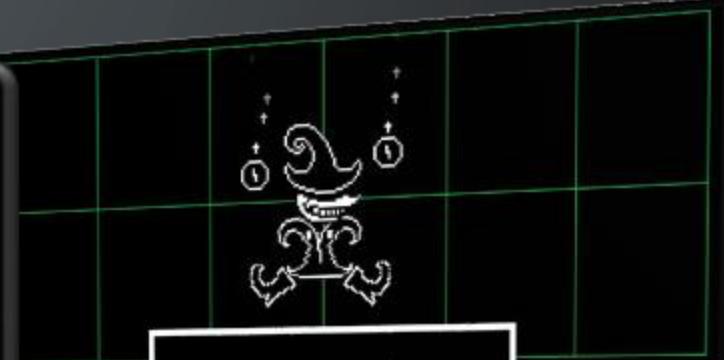
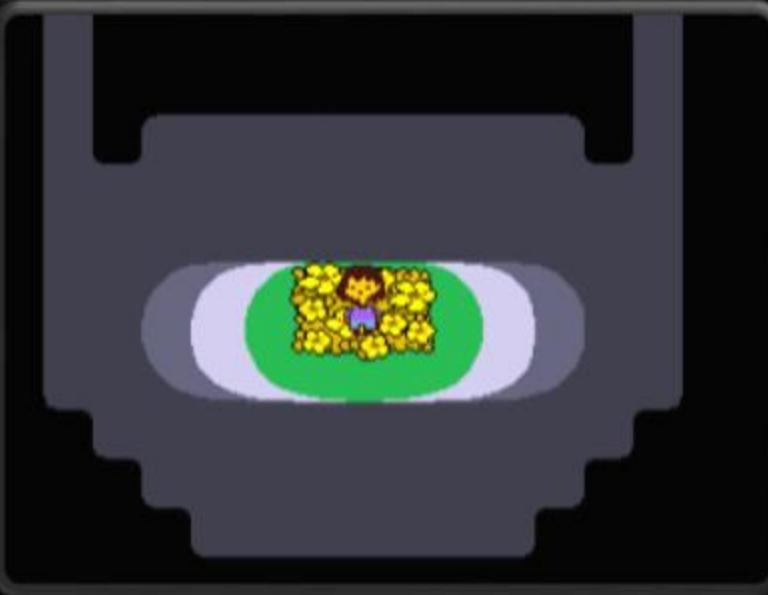
STARDREW VALLEY



UNDERTALE™



Long ago, two races
ruled over Earth:
HUMANS and **MONSTERS**.



CELESTE



Celeste





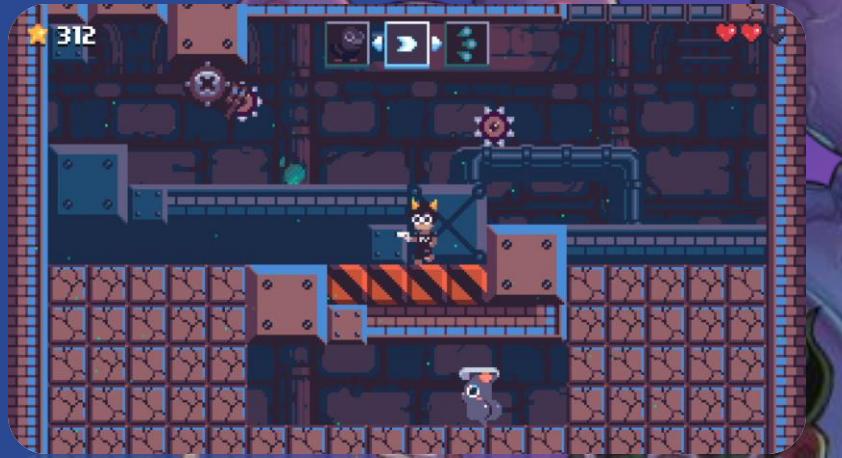


CHROMA SQUAD

TACTICAL RPG

Inspired by Saban's Power Rangers™

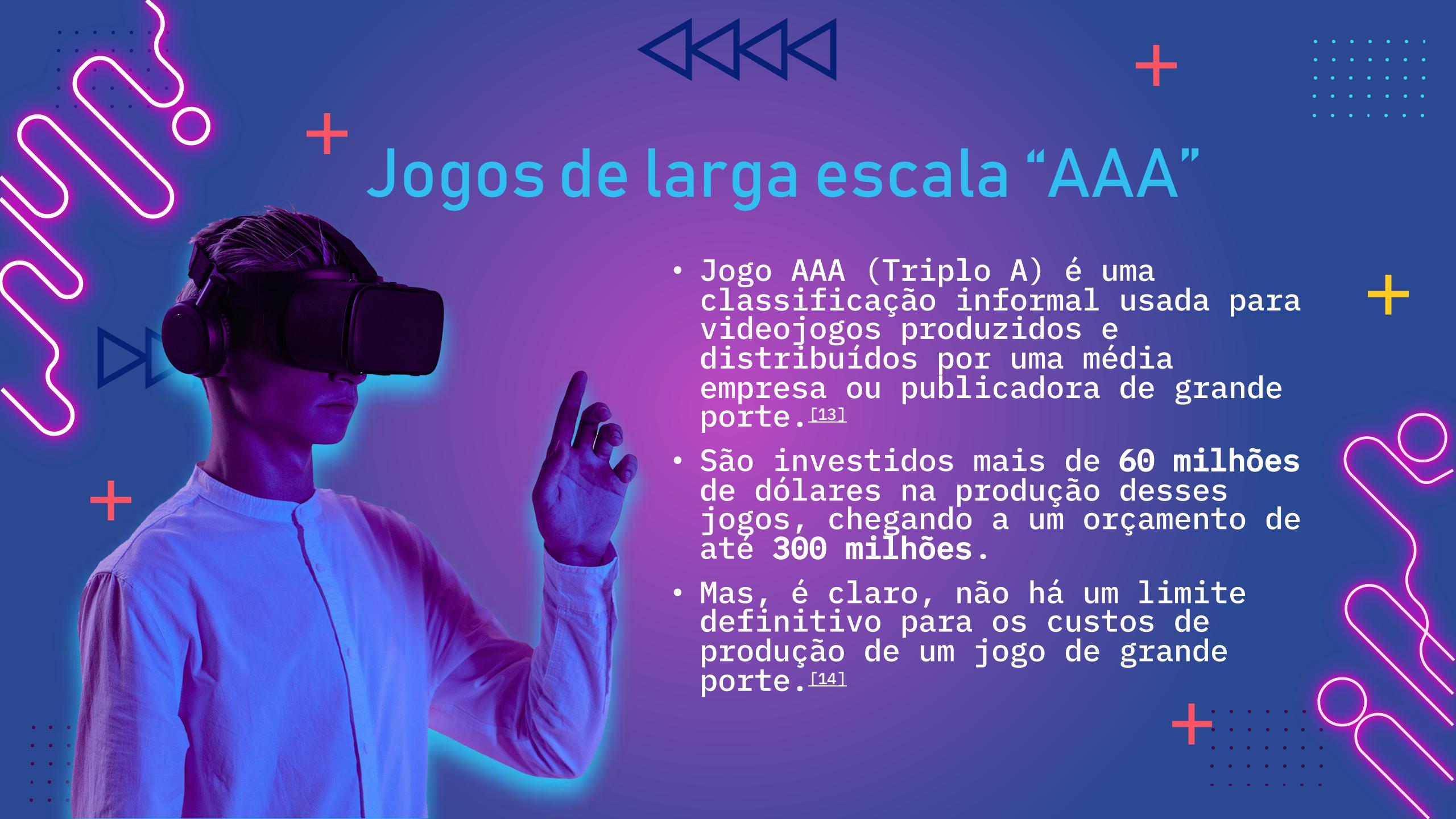




moonleap







Jogos de larga escala “AAA”

- Jogo AAA (Triplo A) é uma classificação informal usada para videojogos produzidos e distribuídos por uma média empresa ou publicadora de grande porte.^[13]
- São investidos mais de **60 milhões** de dólares na produção desses jogos, chegando a um orçamento de até **300 milhões**.
- Mas, é claro, não há um limite definitivo para os custos de produção de um jogo de grande porte.^[14]

*Ainda em desenvolvimento?



ARAÑÍ







DOLMEN®

GAMEPLAY TRAILER



A young woman with short, spiky red hair is sitting on a sandy beach. She is wearing a brown long-sleeved shirt with a circular emblem on the left chest and green cargo pants. Her right leg is bent, and her foot rests on a small pile of dark stones. She is looking off to the side with a thoughtful expression. In the background, there's a tropical landscape with palm trees, blue skies with white clouds, and waves crashing onto the shore.

HAZEL
S · K · Y



RIO

RAISED IN OBLIVION



RIO

RAISED IN OBLIVION



+ Estúdios de pequeno & médio porte

- Nem sempre uma empresa tem as capacidades de um AAA, e as liberdades de um indie.
- Jogos de baixo orçamento e regulares são abundantes no mercado:
 - a grande maioria é composta de jogos mobile.
- Muitos desses estúdios desenvolve outros tipos de software além de jogos
 - ou ministram cursos técnicos e materiais sobre a área.
- Também é comum que essas empresas exportem *assets* para outras maiores.
- Geralmente, se destacam ao focar em um nicho específico.
- Podem crescer ao ponto de se tornar publishers.



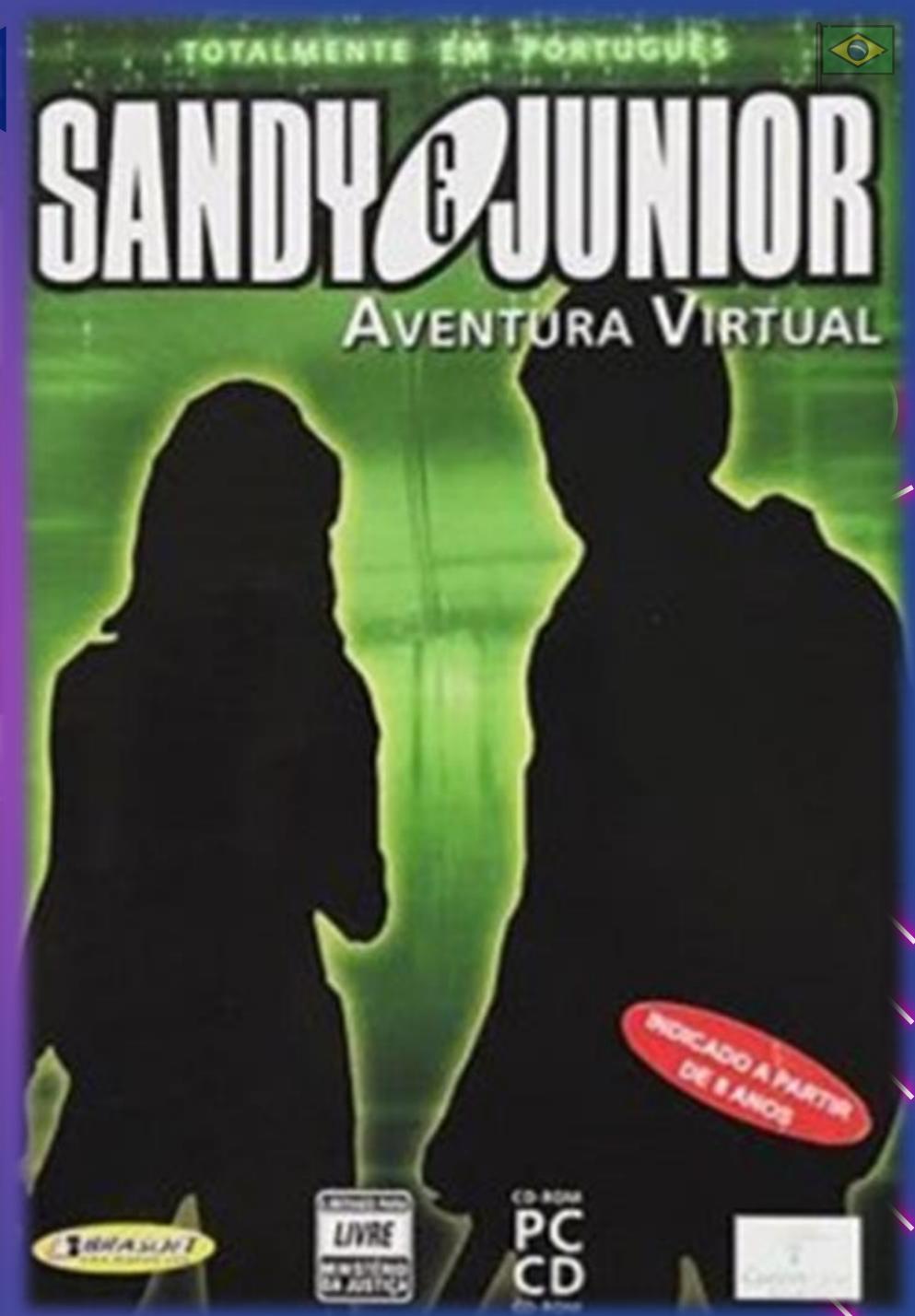


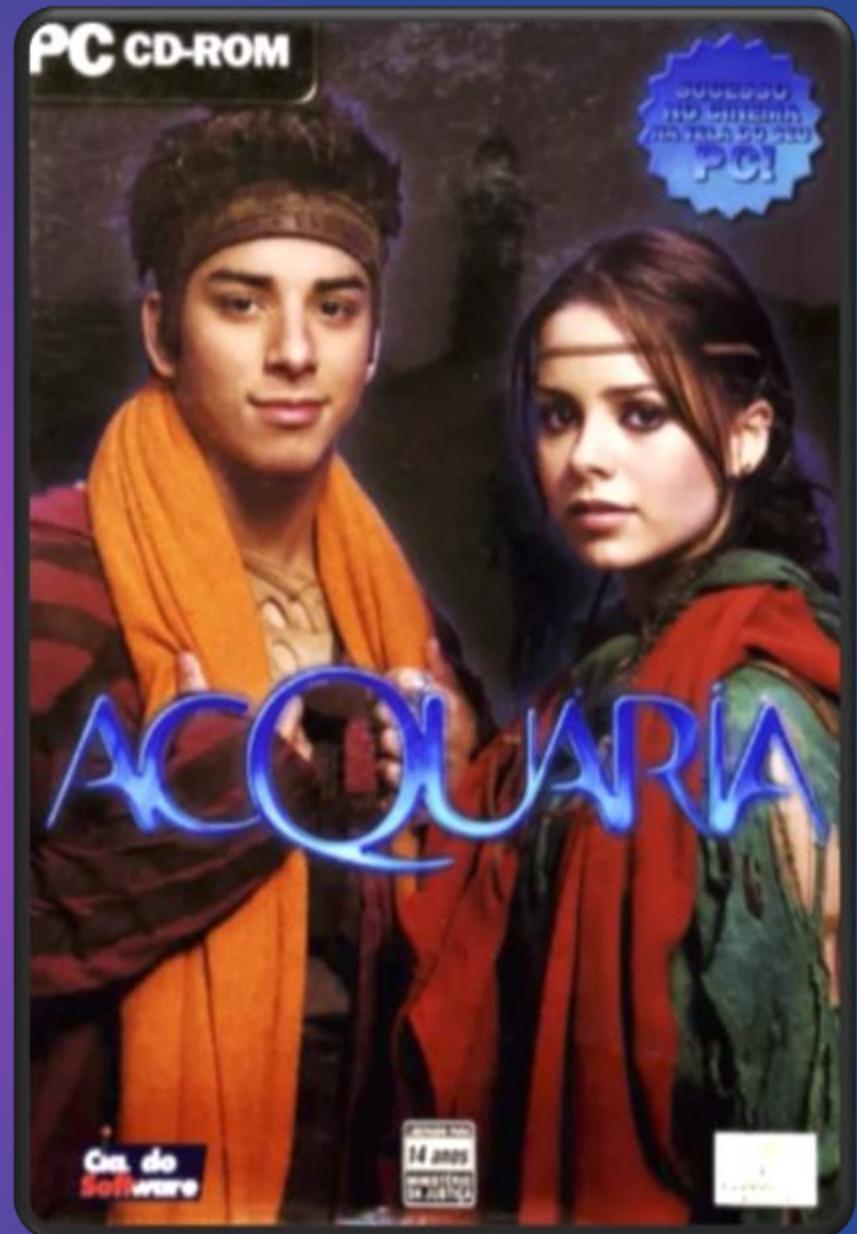
+ Aeroporto 83



- É o primeiro jogo brasileiro da história
 - Desenvolvido em 1983, por Renato Degiovani, para o PC Sinclair ZX81.
 - Foi distribuído através de uma das maiores revistas de informática brasileiras da época, a Micro sistemas.^[12]









Ordem paranormal: O enigma do medo





A LENDA DO
herói
- EDIÇÃO DEFINITIVA -





Plataformas

- Mobile
- Consoles
- AR & VR
- Híbridos



Jogos Mobile

- Videojogo que é jogado em um aparelho telefone móvel - desde smartphones à tablets, smartwatches, PDAs e outros.
- Jogo mobile mais antigo conhecido:
 - variante de Tetris do dispositivo Hagenuk MT-2000 de 1994.
- Se destacam em relação a outras plataformas devido suas propriedades únicas:
 - necessidade de serem adaptados para controles de toque;
 - quantidade limitada de recursos;
 - devem ser portados para plataformas de diferentes tamanhos e capacidades.
- Indústria mobile atingiu aproximadamente \$68.5 bilhões de dólares em 2019, com uma taxa de crescimento de 10% ao ano
- 45% do total da indústria de jogos^[5] (Newzoo)
- Fatia do mercado de games mais comum no Brasil atualmente.^{[6][7]}



JOGABILIDADE ARCADE CLÁSSICA

SAI DA FRENTE!



Shop

day 12

\$ 12,838 +



Record
You
Lap 1
Lap 2
Lap 3

01:00'12
00:10'94

#5 Laps
1/3





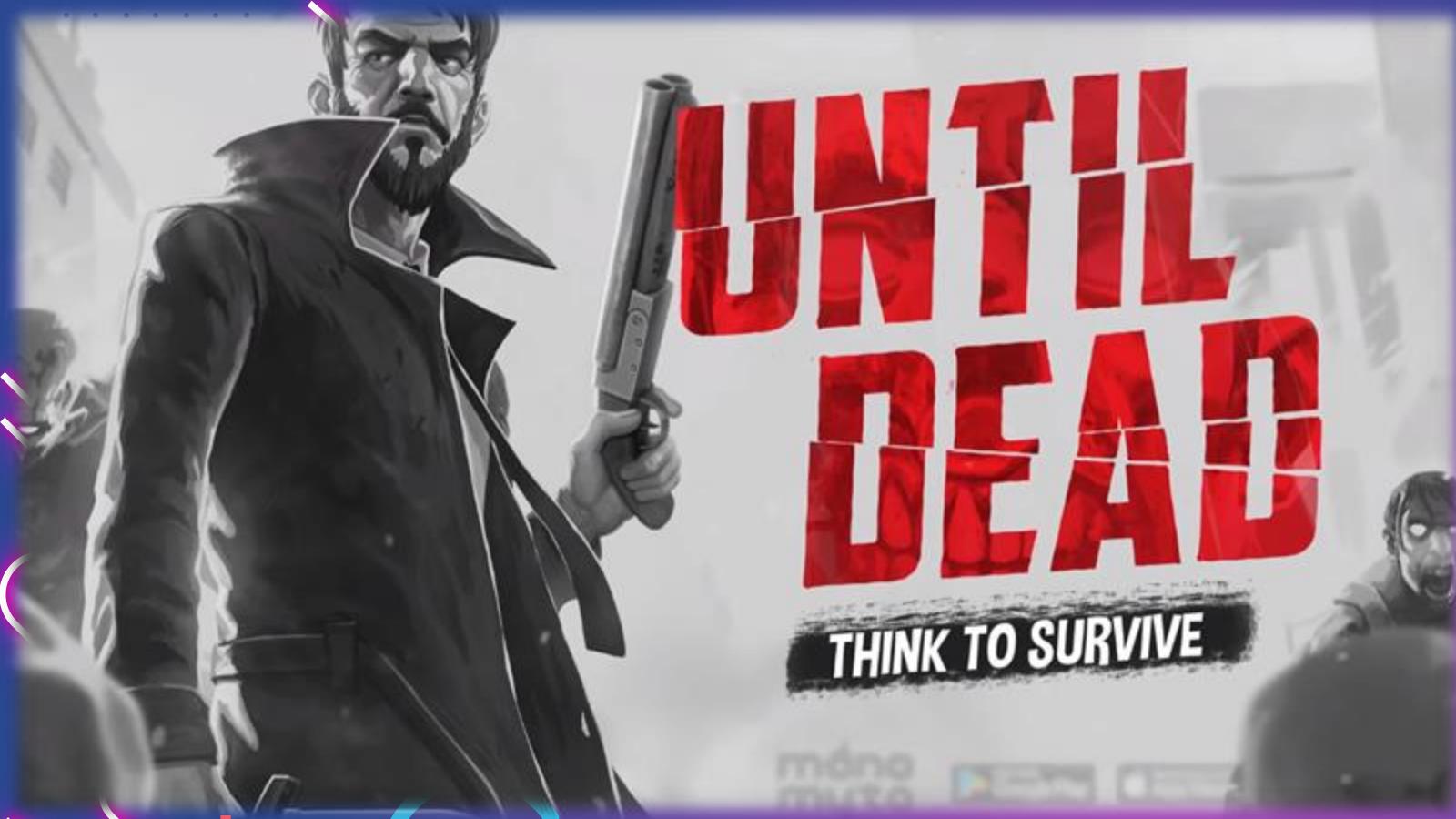
UNTIL DEAD

THINK TO SURVIVE

SOBREVIVA AO APOCALIPSE ZUMBI

CONCLUA DEZENAS DE FASES DE QUEBRA-CABEÇAS

PENSE NA MELHOR JOGADA



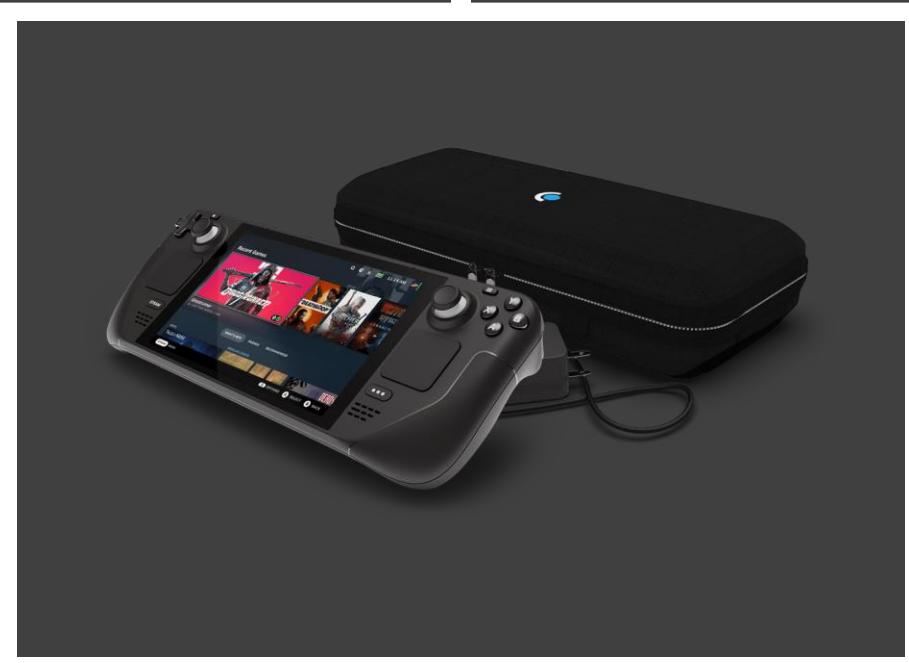




Consoles

- Consoles tradicionais
- Handhelds
- Arcades









AR & VR

- Realidade Aumentada
- Realidade Virtual
- ARGs





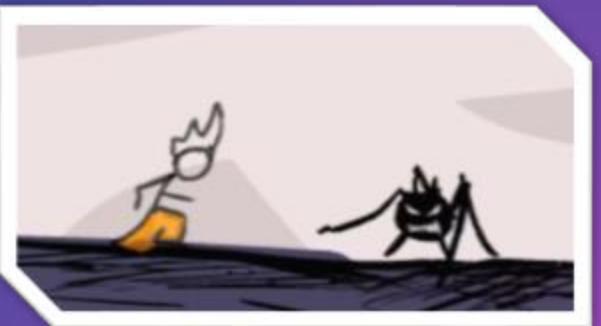


Híbridos

- PC
- Web browsers & Embutidos
- Multiplataforma



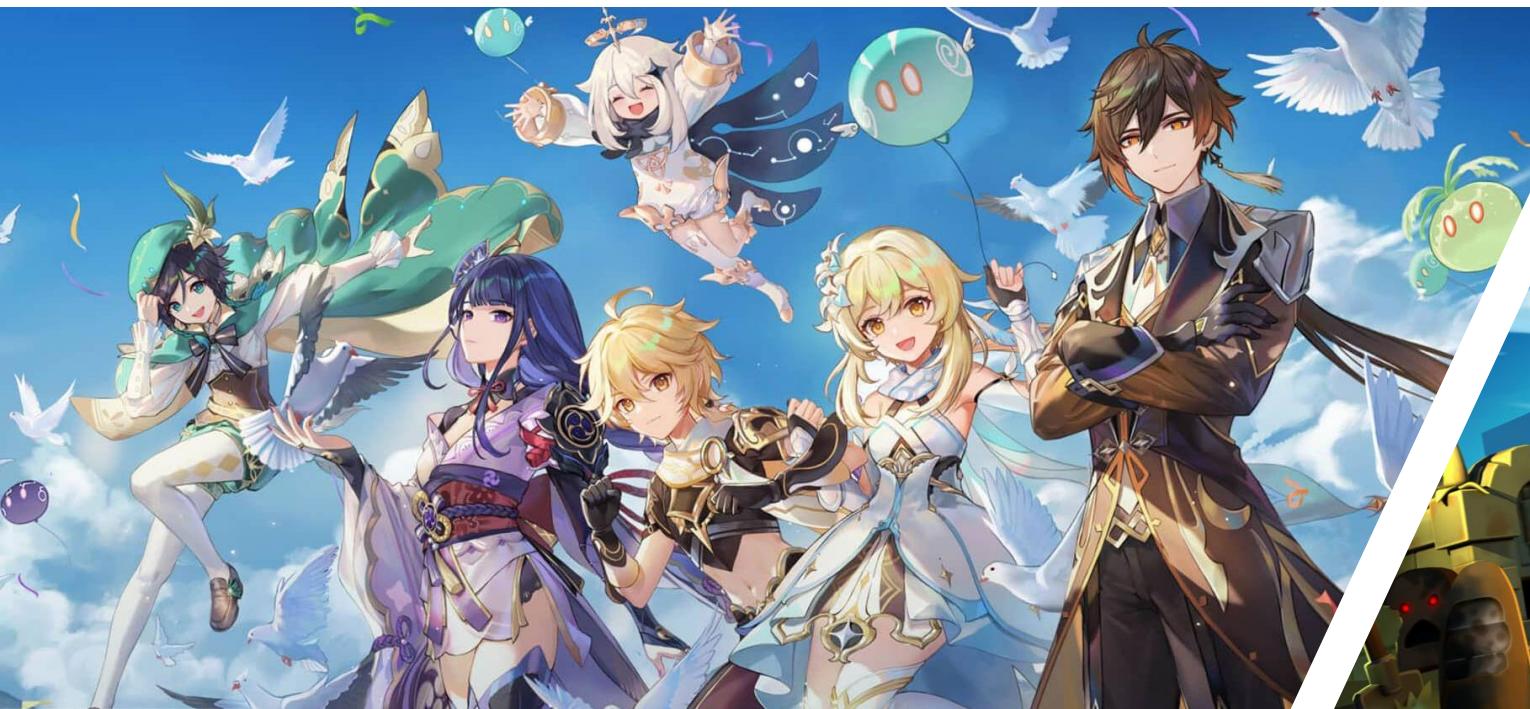






A screenshot of the GeForce Now interface. At the top, it says "Mais populares na nuvem" and shows thumbnails for "FORTNITE", "FORZA HORIZON 5", "RAINBOW SIX SIEGE", "WOLONG: GUARDIAN OF THE QI", "HALO INFINITE", "ATOMIC HEART", and "FORZA HORIZON 4". Below that, it says "GAME PASS" and shows thumbnails for "DEAD SPACE", "MERGE & BLADE", "SOUL HACKERS 2", and "ATOMIC HEART". At the bottom, it says "Adicionados recentemente" and shows thumbnails for "DEAD SPACE", "DEAD SPACE", "UNITY: EARTH", "WOLONG: GUARDIAN OF THE QI", "MERGE & BLADE", "SOUL HACKERS 2", and "ATOMIC HEART".







Tecnologias

- Game Engines;
- FL0SS;
- Frameworks, Bibliotecas & SDKs;
- APIs & Middlewares;
- Principais linguagens.





+ Game Engines



- Motores de jogos, também conhecidos como Game Frameworks
- Ambientes de desenvolvimento de software projetados para jogos
- Facilitam o processo de criação, por meio da abstração de funcionalidades fundamentais:
 - Rendering - motor gráfico;
 - Detecção de colisões - motor de física;
 - Ambiente próprio de SDKs;
 - Middlewares e APIs de alto nível;
 - Interfaces gráficas: editores visuais;
 - Outros sistemas gerais...



+ Game Engines



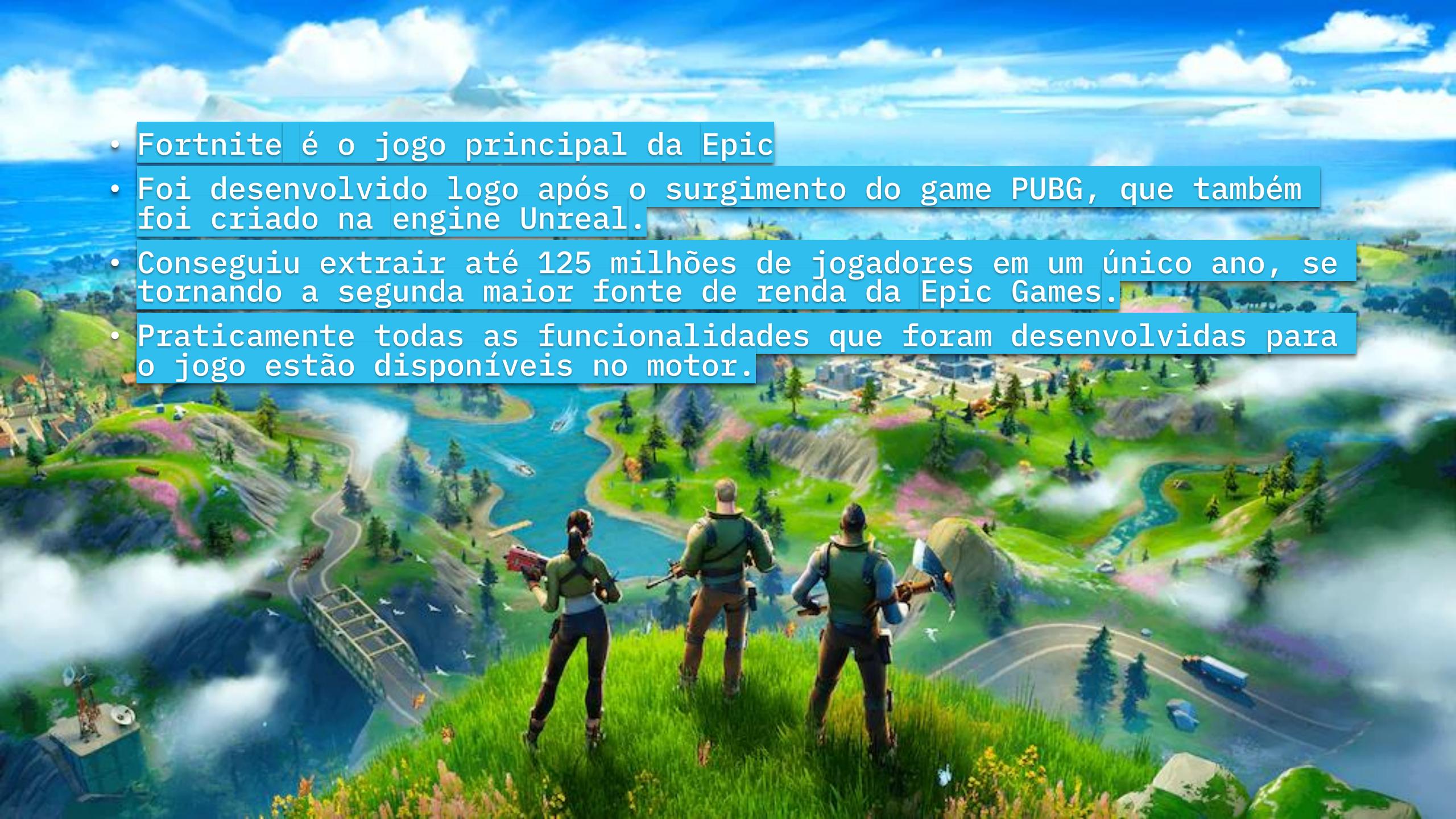
- Permite desenvolvedores dedicarem-se à jogabilidade.^[23]
- Acessíveis por meio de uma interface unificada, ou via scripting e modularização.
- O motor de jogo é a **ferramenta-mestre** do programador de jogos, e determina muito do seu workflow:
 - estilos arquiteturais
 - linguagens usadas
 - opções de IDEs e outras ferramentas
 - cultura de comunidade
 - conjunto de habilidades necessárias e limitações

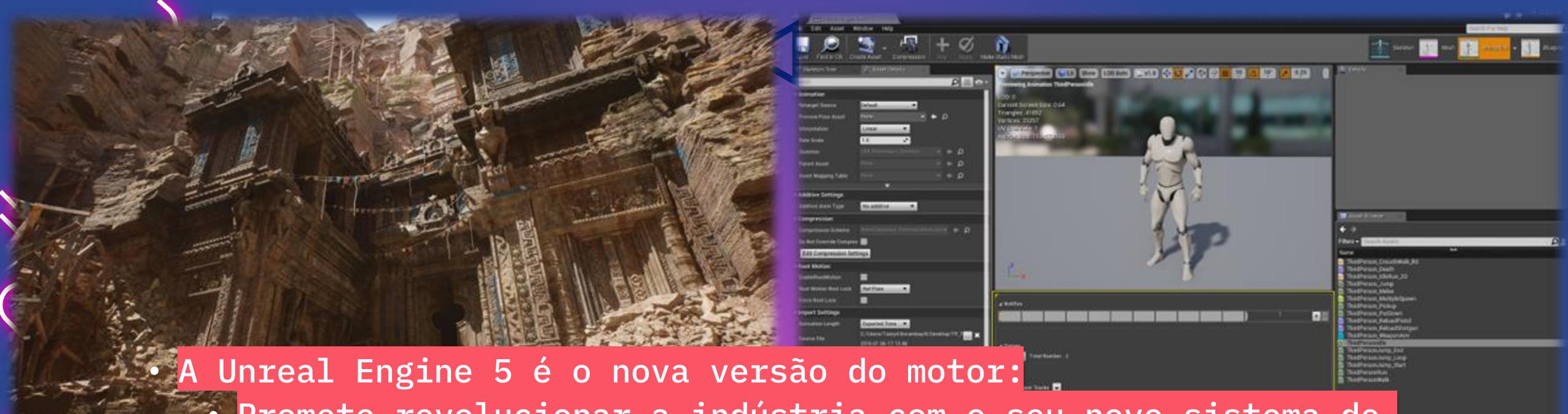


A engine mais visualmente poderosa em realismo

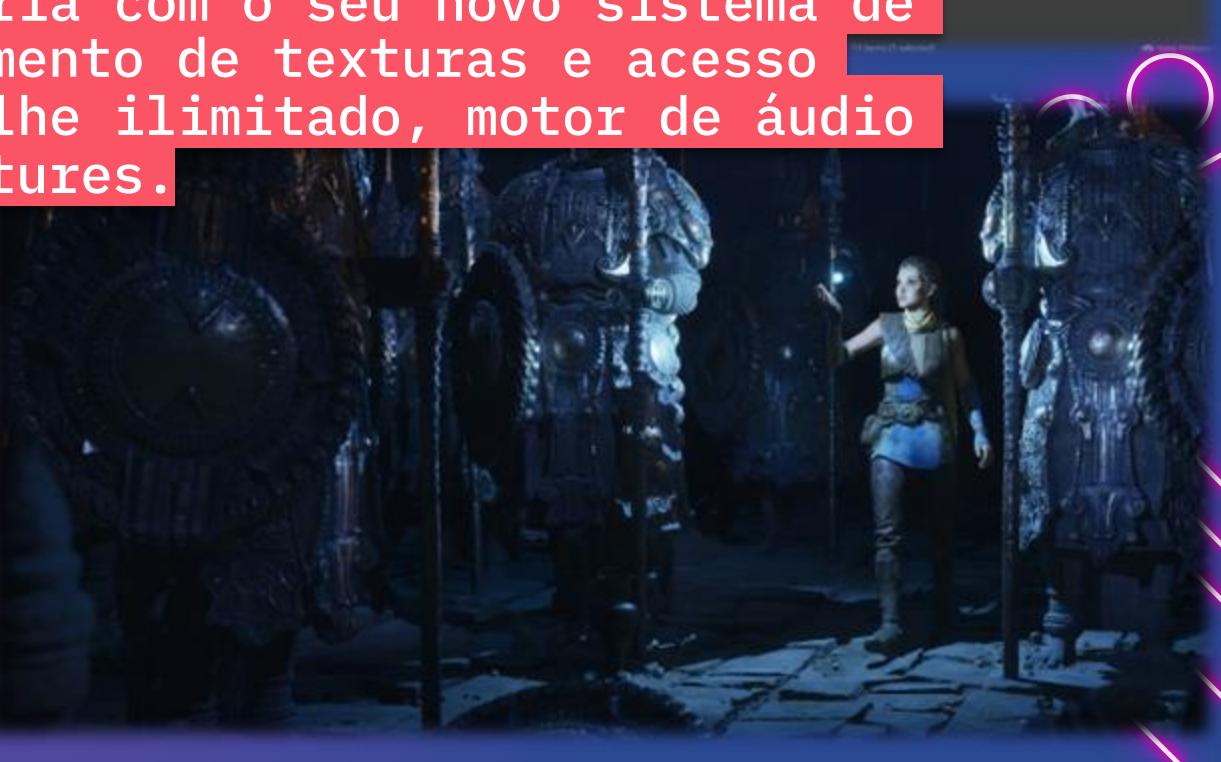
- Criado pela Epic Games
- Usado pela primeira vez em 1998 na produção do jogo Unreal Tournament
- Embora inicialmente projetado para jogos de tiro em primeira pessoa (FPS), tem sido usada com sucesso em uma variedade de gêneros de jogos.^[24]
- Também largamente usada em outros setores que não jogos:
 - Estúdios de animação;
 - Engenharia;
 - Arquitetura;
 - Simulação;
 - Entre outras...
- É um dos softwares de renderização em tempo real mais fotorealistas do mercado.
- Seu núcleo é escrito em C++
 - Permite a portabilidade para múltiplas plataformas incluindo Microsoft Windows, Linux, Mac, e praticamente todos os consoles de mesa modernos mais populares.
- Blueprints como linguagem opcional
- Motor de código aberto (com licença proprietária)

- Fortnite é o jogo principal da Epic
- Foi desenvolvido logo após o surgimento do game PUBG, que também foi criado na engine Unreal.
- Conseguiu extrair até 125 milhões de jogadores em um único ano, se tornando a segunda maior fonte de renda da Epic Games.
- Praticamente todas as funcionalidades que foram desenvolvidas para o jogo estão disponíveis no motor.





- A Unreal Engine 5 é o nova versão do motor:
 - Promete revolucionar a indústria com o seu novo sistema de tratamento de *assets* (carregamento de texturas e acesso veloz aos dados em SSD), detalhe ilimitado, motor de áudio espacial, e muitas outras features.

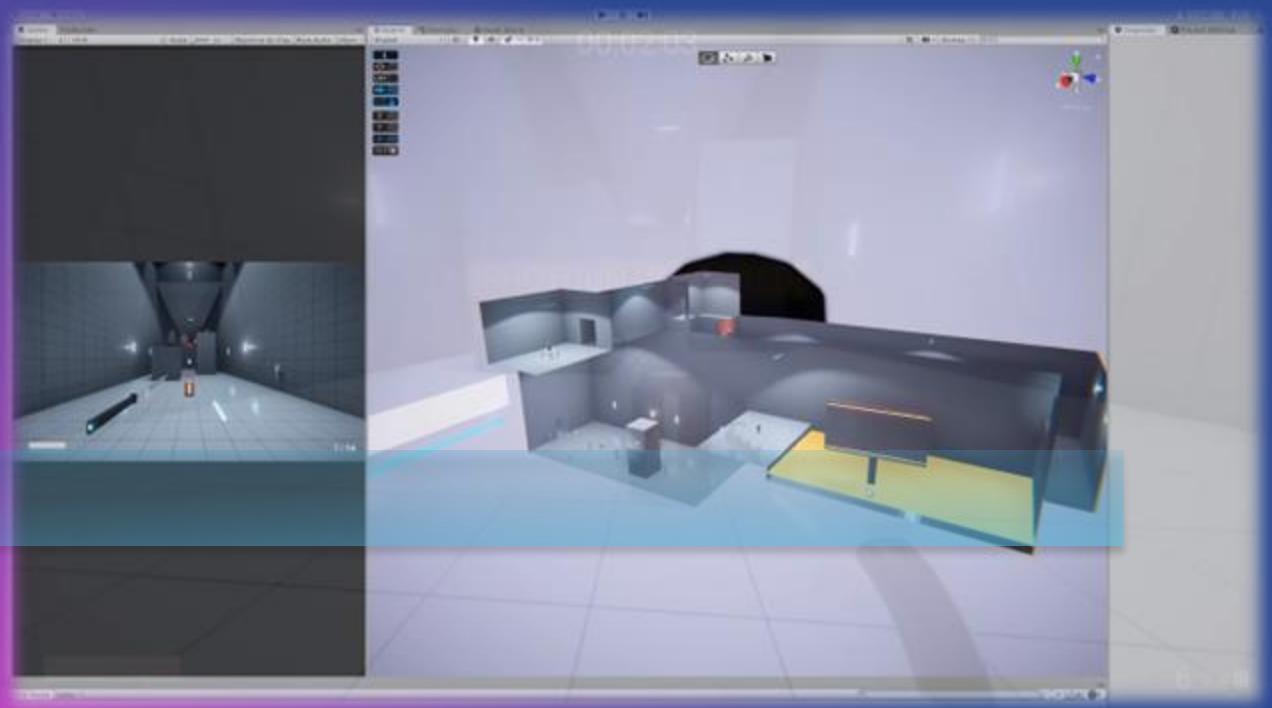
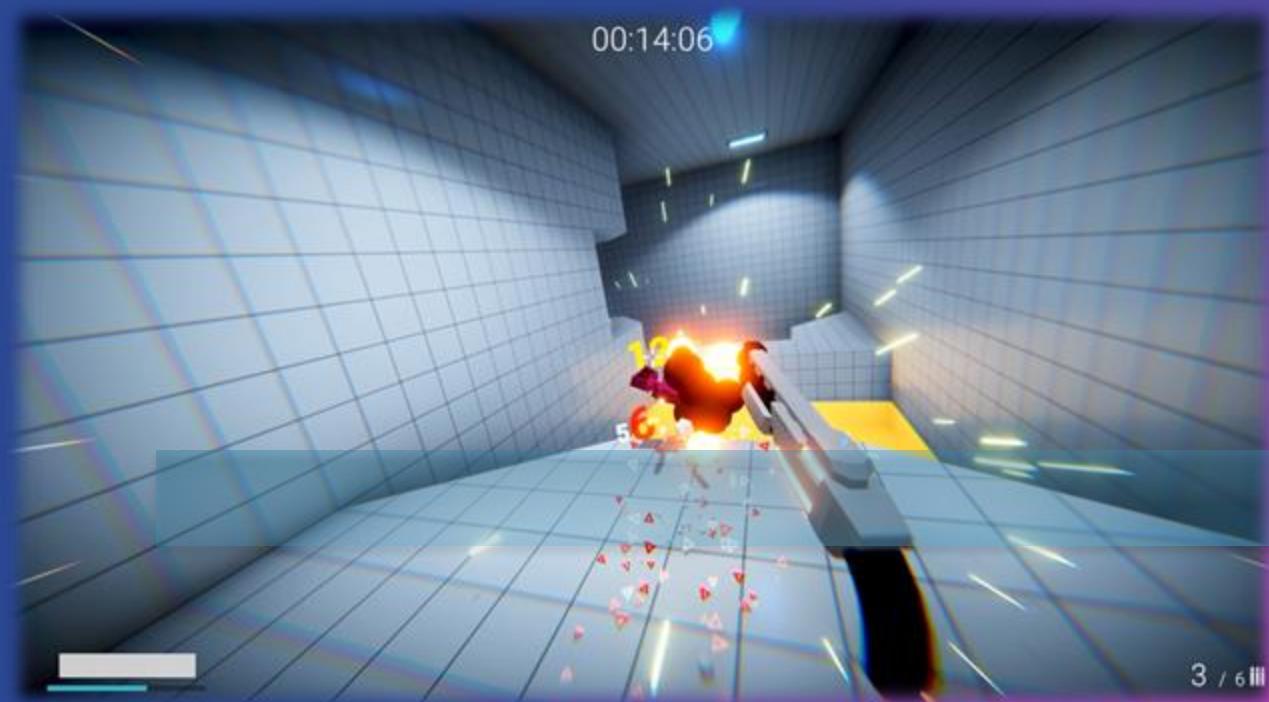


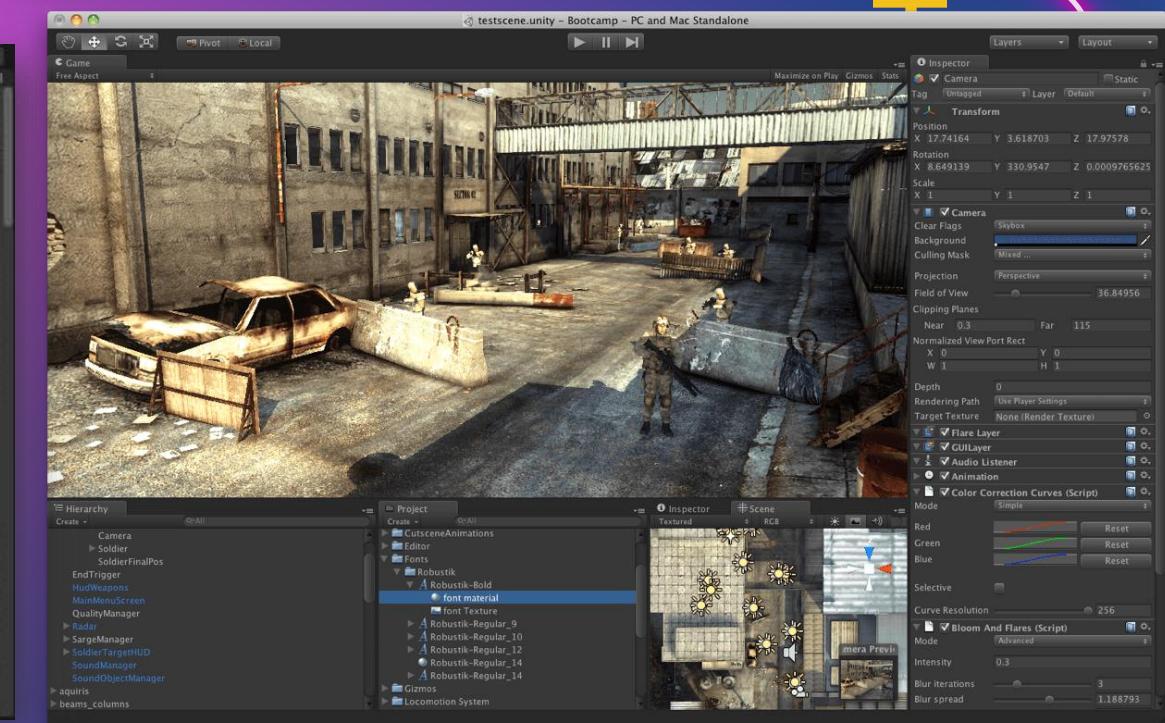
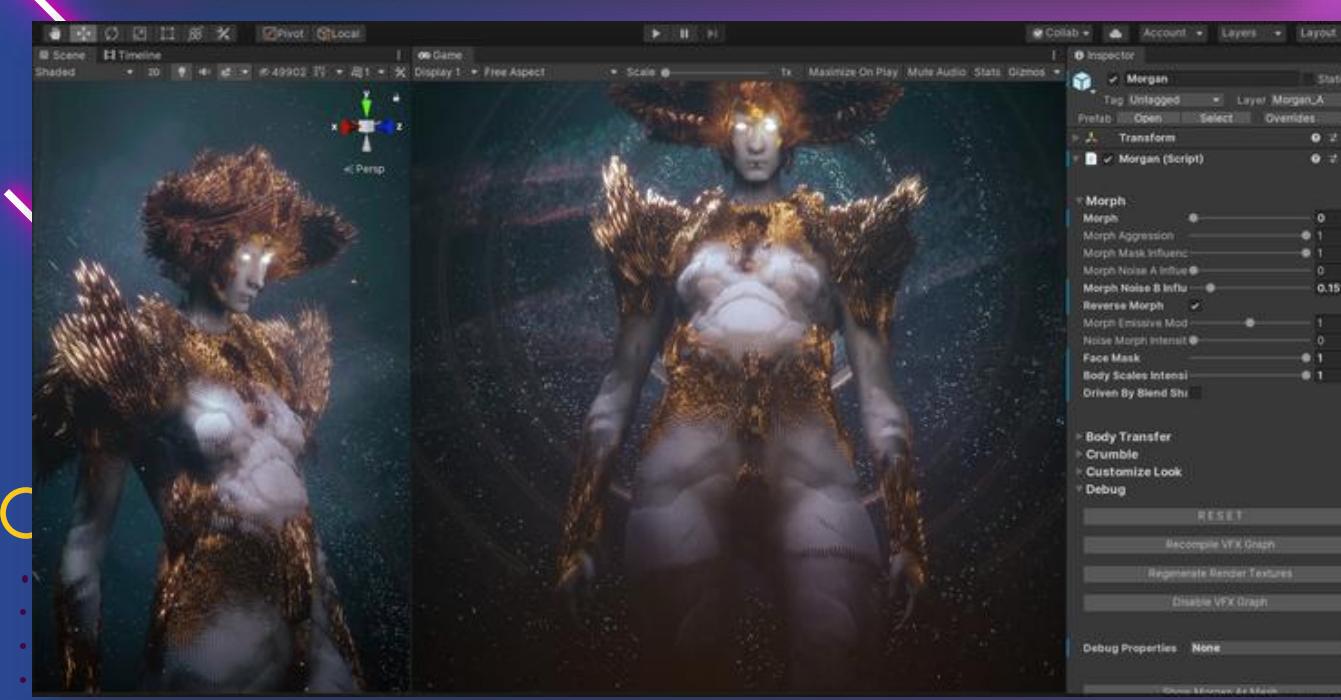
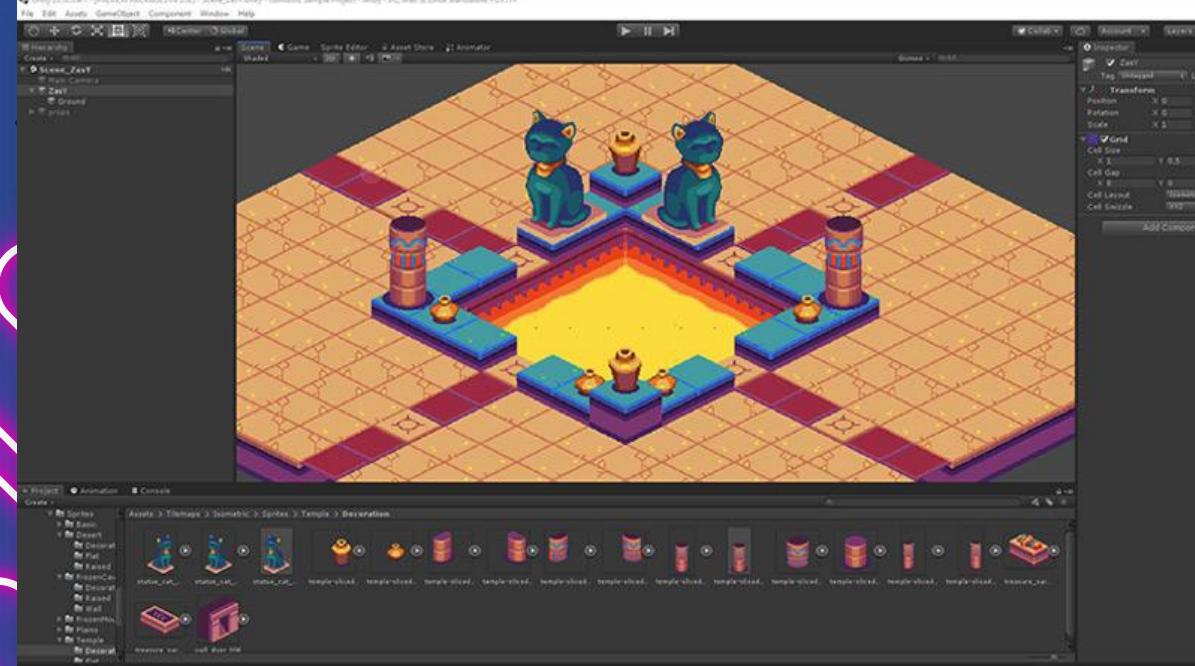


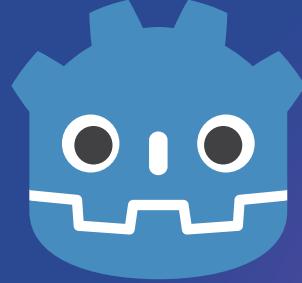
unity

O motor de jogo mais popular do mercado

- Permite a criação de jogos 3D & pseudo-2D - sendo mais especializada em jogos tridimensionais
- É muito usada entre desenvolvedores independentes:
 - Foi uma das primeiras engines de grande porte gratuitas: JavaScript era uma das linguagens adotadas na época;
 - Facilidade de uso das funcionalidades 3D;
 - C# da Microsoft como linguagem principal.^[25]
- Lançada em 2005 em uma conferência da Apple, com o objetivo de “democratizar” o desenvolvimento de jogos. Desde então, vem expandindo suporte para mais de 25 plataformas. Assim como a Unreal é usada em diversas áreas que não jogos, e assim como as outras 2 engines maiores também suporta VR & AR.^[25]
- Vantagens:
 - Grande comunidade;
 - Quantidade exorbitante de conteúdo produzido para ela;
 - À princípio gratuita.
- Desvantagens:
 - taxas aplicadas conforme a licença escolhida e rentabilidade produzido pelo seu jogo;
 - há também um limite de recursos disponíveis ou restrições conforme suas licenças.^[26]





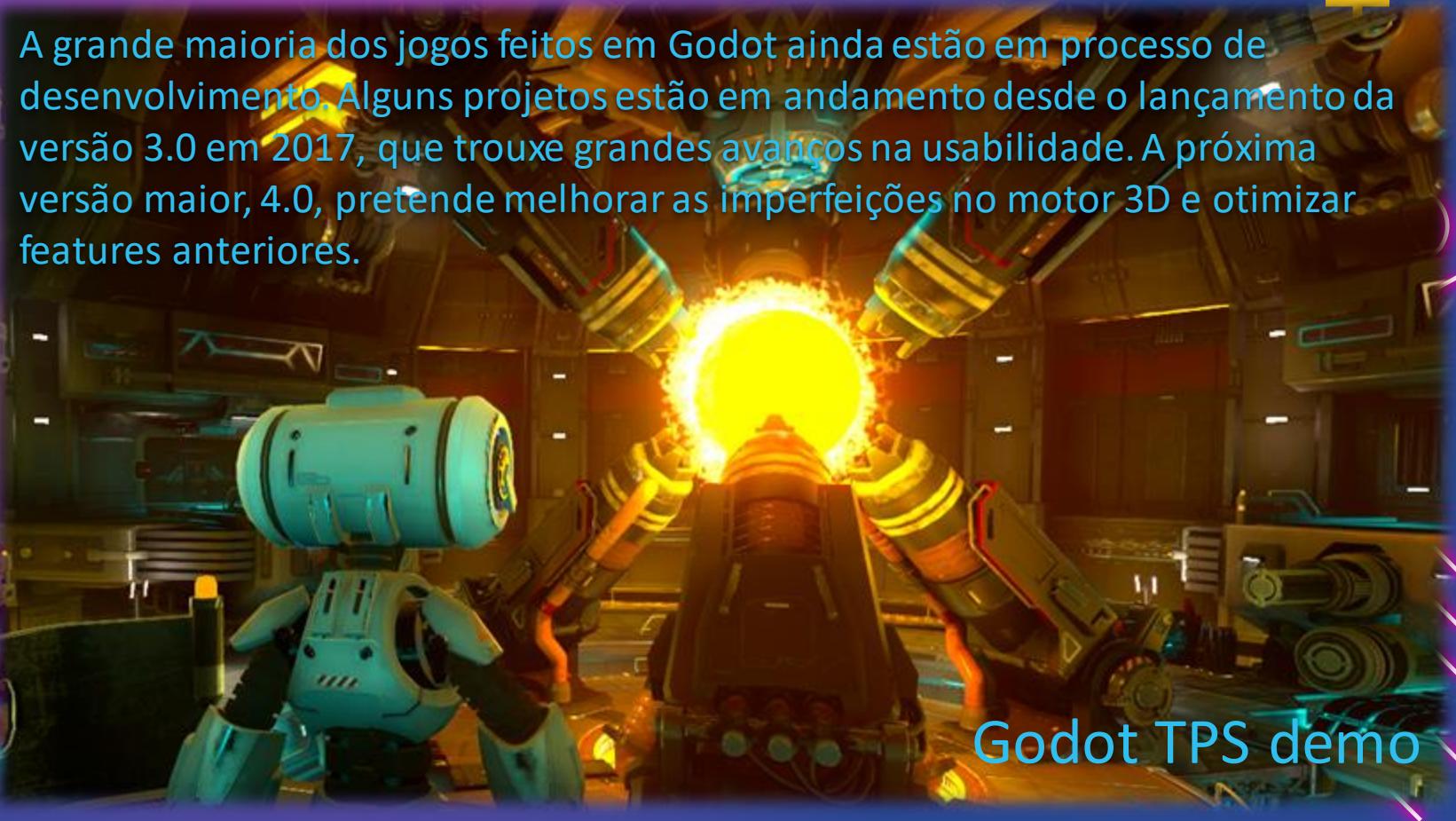


GODOT

Game engine

The game engine you waited for

- Motor 2D & 3D: [\[27\]](#)
 - Atualmente, possui um dos motores 2D mais poderosos;
 - Editor repleto de funcionalidades;
 - Porém não é tão forte em 3D quanto as outras opções.
- Lançada ao público em 2014^[28] sobre a licença MIT (uma das mais permissivas)
- Inicialmente desenvolvida pelos argentinos Ariel Manzur e Juan Linietsky para diversas empresas na América Latina
- Vantagens:
 - Multiplataforma (exportação e desenvolvimento);
 - Livre e de código aberto - desenvolvido em meio a comunidade FLOSS;
 - Design moderno: Extremamente extensível e modular;
 - Comunidade em ascensão.
- Desvantagens:
 - Escassez de material em português – comunidade muito mais jovem em relação à outras tecnologias;
 - Performance relativamente menor;
 - Special-case-bugs ocasionais de um projeto FLOSS em constante movimento.
- Suporta oficialmente as linguagens GDScript, C#, C++ & Visual Script;
 - Também é possível estender o motor através de bindings em C/C++. Sendo os comuns:
 - Javascript^[29], Lua^[30], Nim^[31], Python^[32] & Rust^[33].



Sealed Byte

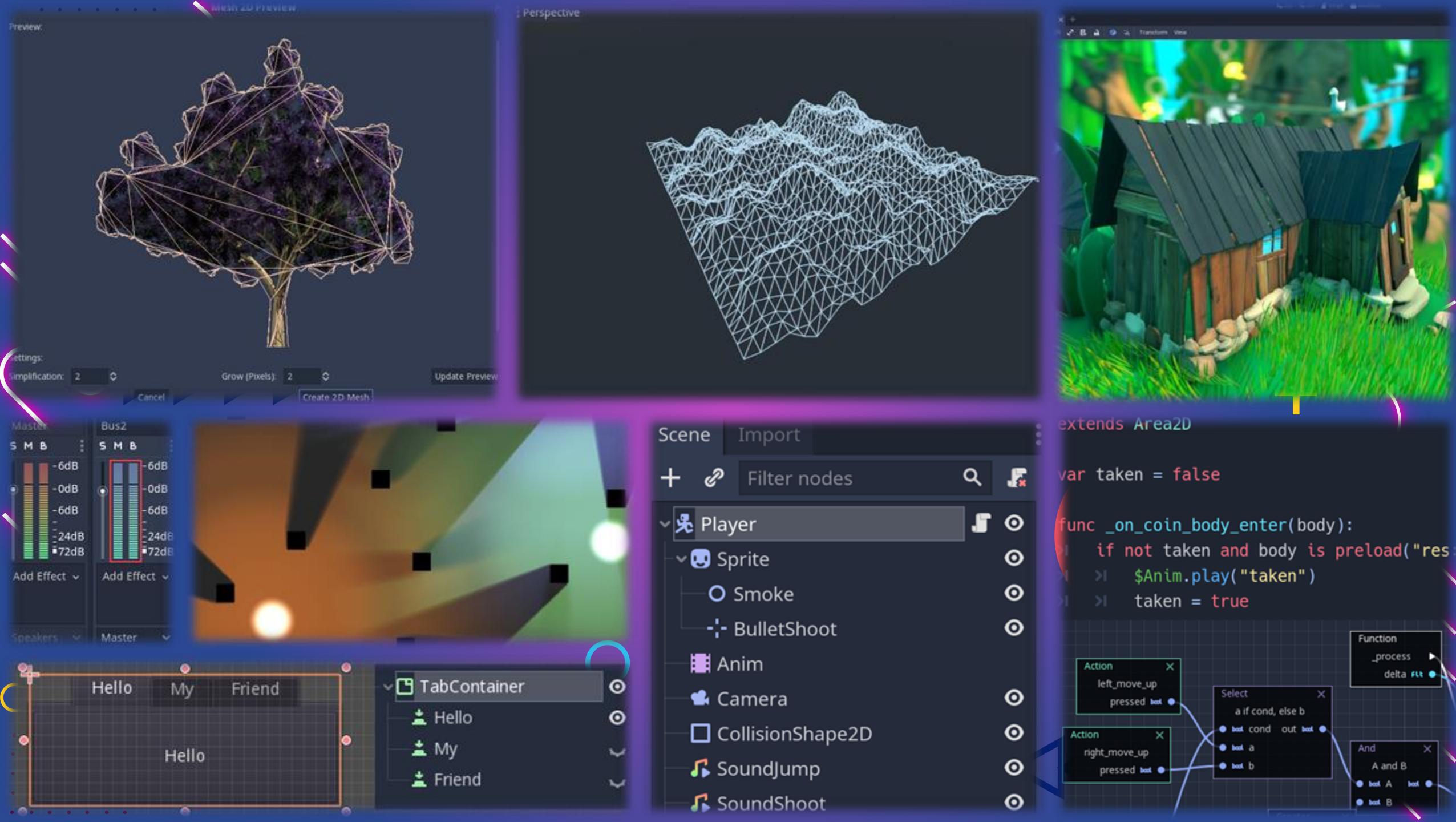
Ex Zodiac

Infinistate

Oddventure

A grande maioria dos jogos feitos em Godot ainda estão em processo de desenvolvimento. Alguns projetos estão em andamento desde o lançamento da versão 3.0 em 2017, que trouxe grandes avanços na usabilidade. A próxima versão maior, 4.0, pretende melhorar as imperfeições no motor 3D e otimizar features anteriores.

Godot TPS demo

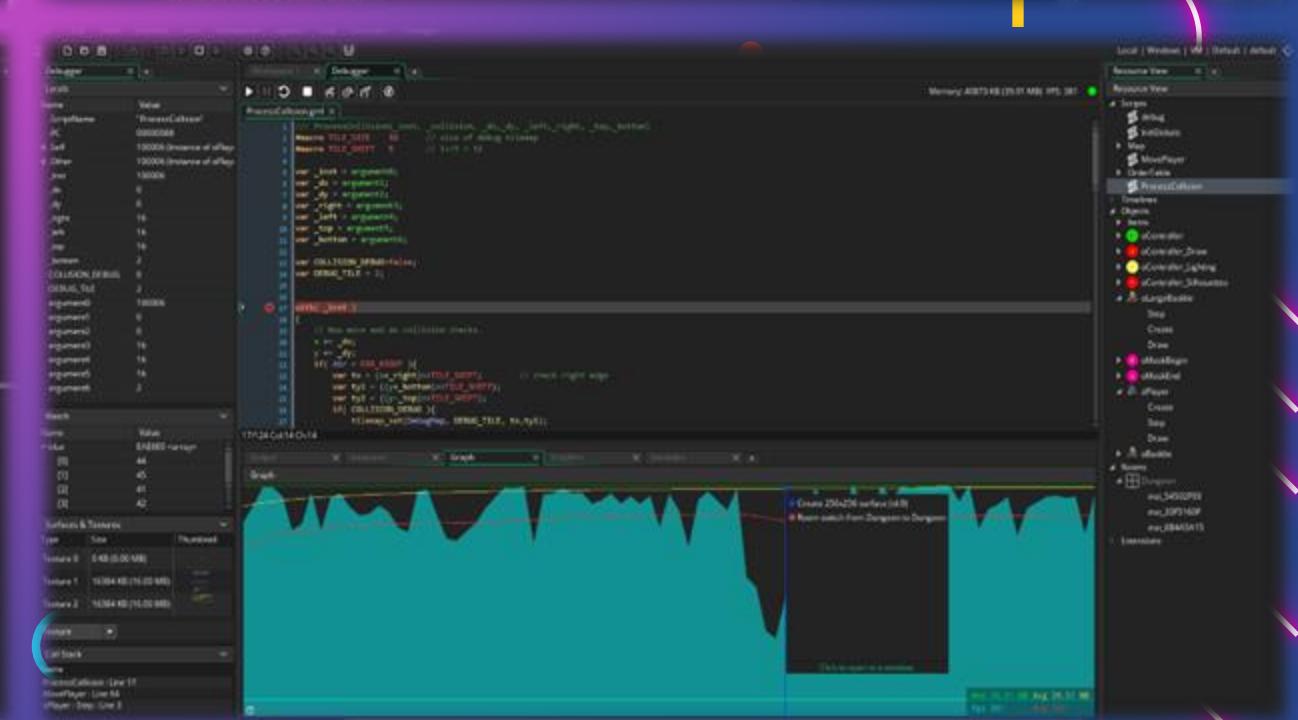
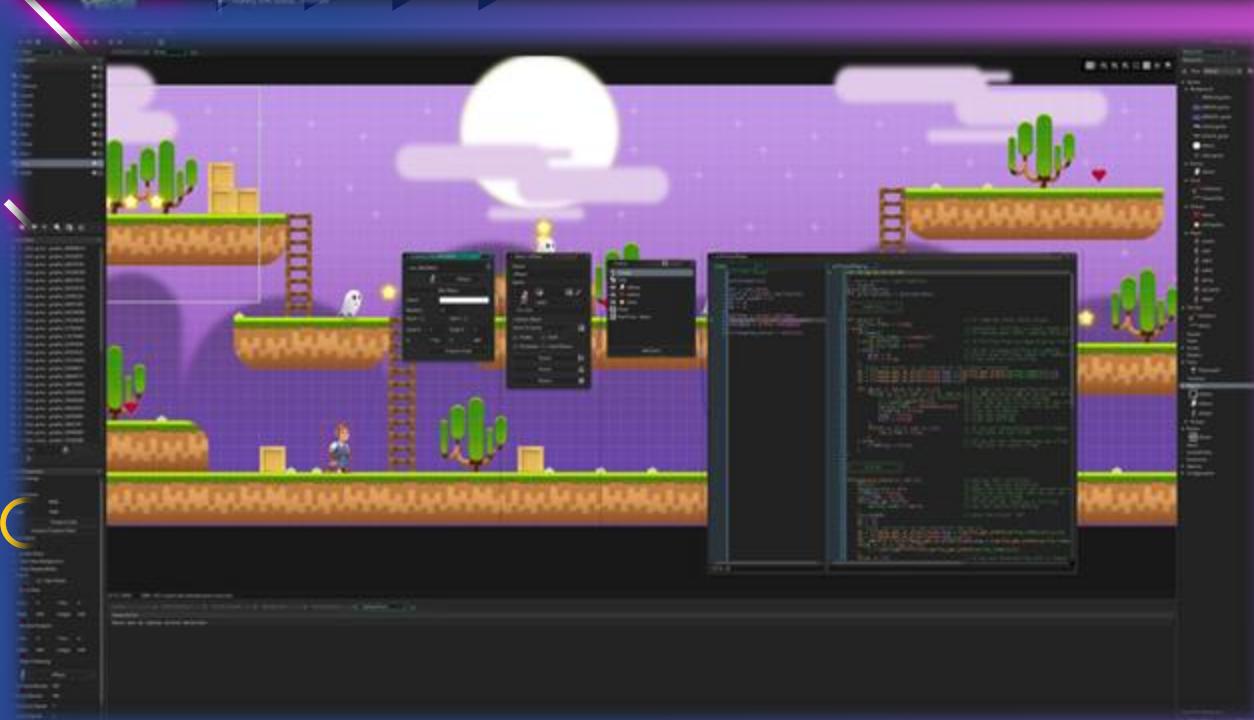




O motor 2D mais popular, amado por devs indies

- Motor de jogo proprietário, desenvolvido pela YoYo Games
- À princípio, surgiu como um software de animação 2D chamado de Animo até 1999.^[34]
- É considerado um dos melhores motores 2D do mercado (rivalizando com a Godot)
- Vantagens:
 - Usabilidade intuitiva;
 - Grande quantidade de ferramentas embutidas, como:
 - Editor de sprites,
 - Drag & drop,
 - Editor de “salas” e mapas,
 - Janelas de eventos;
 - painéis de scripts.
- Desvantagens:
 - *Não é gratuito,*
 - Maiores limitações em relação as suas funcionalidades-base;
 - Não suporta 3D;
- Ele usa sua própria linguagem de programação compilada e procedural chamada: Game Maker Language (GML)





Há inúmeras outras opções de engines...

Cabe ao desenvolvedor decidir qual é mais adequada para suas habilidades, necessidades e preferências.





+ FLOSS - Free, Libre and Open Source Software

“Softwares livres de código fonte aberto e de graça”

- São softwares livremente licenciados para conceder aos usuários os direitos de uso, cópia, estudo, mudança e melhoria em seu design.^[35]
- Quando se refere à games não há tantas opções FLOSS quanto outras tecnologias.
- Recentemente novas tecnologias para jogos vem ganhando espaço no mundo do software livre:
 - A Godot engine é o exemplo mais forte - é um dos primeiros e o mais popular motor de alto nível a adotar uma licença MIT.
 - O motor UPBGE - Uchronia Project Blender Game Engine - foi outro motor de jogos livre. Desenvolvido sobre um fork do software de gráficos 3D Blender, sobre uma licença GNU, que é um pouco mais restritiva que a MIT.





+ Frameworks



- Frameworks são abstrações que reúnem códigos comuns entre vários projetos de software provendo um conjunto de funcionalidades genéricas.
- Muitas vezes é citado como “conjunto de conceitos usados na resolução um dado problema”, porém não deve ser confundido com um padrão de projeto de software já que os frameworks são implementações (código definido).^[36]
- Frameworks muitas vezes são usadas como sinônimo de engines, mas em geral, se diferenciam pela presença de editores visuais nos motores. Note que Frameworks podem ter editores, mas, geralmente, contém apenas uma base de código extensível.





Frameworks, Bibliotecas & SDKS (Software development kits)

- Cocos2D^[37]: é usado largamente na vertente dos jogos mobile, desenvolvido em C++, Python & Objective-C. Semelhante à Godot suporta múltiplas linguagens de forma extensível, porém por meio de diferentes versões do framework. A principal sendo a Cocos2D-x, com C++, Lua ou JavaScript. Suas subversões mais conhecidas, são: Cocos2D-Swift, Cocos2D-html5 & Cocos2D-XNA.^[38]
- Lua: Love2D^[39], Gideros^[40], Amulet^[41]...
- Python: Panda3D^[42], PyGame^[43] (lib.), Pykyra^[44]...
- Js: Phaser^[45], Pixe.JS^[46], Kiwi.JS^[47], Quintus^[48]...
- Java: LWJGL^[49] (lib.), LibGDX^[50]...
- C++: Oxygine^[51], SDL^[52] (lib.), Allegro^[53] (lib.), OpenCL^[54]...
- Rust: Bevy^[54], Amethyst^[55], GGEZ^[56] (lib.)...
- SDKs: Solar2D (antiga Corona)^[57], Moai^[58], Marmalade SDK^[59].

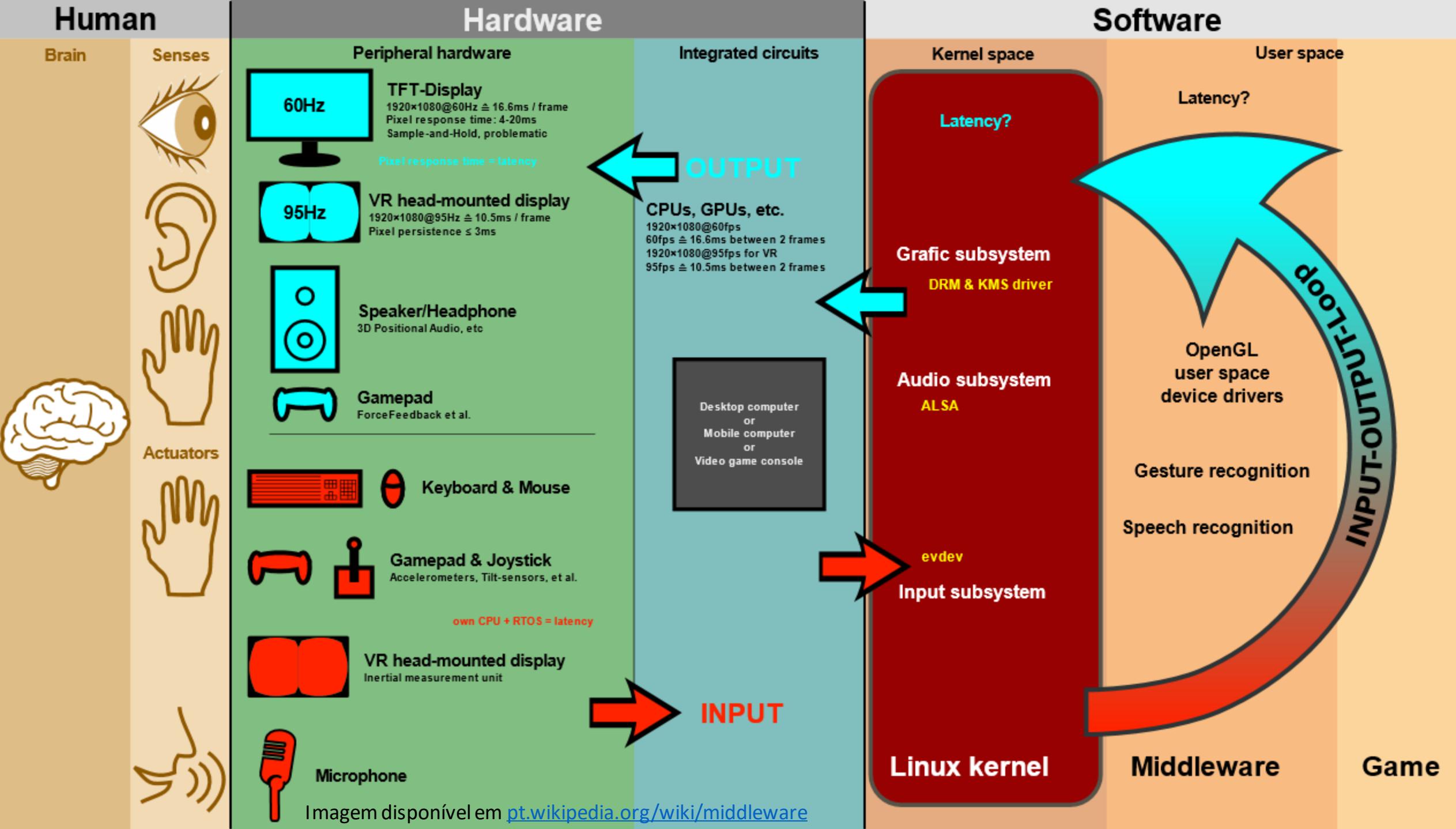


+ Middlewares



- Middleware é um software que fornece serviços para aplicações, além daqueles disponíveis pelo sistema operacional. Pode ser descrito como “cola de software”;
- Eles facilitam os desenvolvedores implementarem comunicação e entrada/saída, de forma que eles possam focar no propósito específico da sua aplicação.^[60].
- Exemplos: [FMOD^{\[61\]}](#) (sons), [Havok^{\[62\]}](#) / [Bullet](#) (física), [SpeedTree^{\[63\]}](#) (geração procedural de vegetação), [Euphoria^{\[64\]}](#) (simulação de animação de humanos) [AiLive^{\[65\]}](#) (IA)...





Routing

VCAs

Snapshots

Mixing Desk Selected Buses

Q-

- char/madeline/stand
- char/badeline/stand
- char/madeline/summit_areastart
- char/madeline/summit_flytonext
- char/madeline/summit_sit
- char/badeline/temple_move_chats
- char/badeline/temple_move_first
- char/madeline/theo_collapse
- char/madeline/wallslide
- char/badeline/wallslide
- char/madeline/water_dash_gen
- char/madeline/water_dash_in
- char/madeline/water_dash_out
- char/madeline/water_in
- char/madeline/...r_move_general
- char/madeline/...r_move_shallow
- char/madeline/water_out
- char/theo/yolo_fist

► Grp classic

► Grp game

Rtn prologue_verb

Grp music

► Grp stings

► Grp tunes

▼ Grp ui_sfx

► Grp rename_piano

► Grp worldmap_whoosh

Rtn rename_verb

New Group

New Return

Flatten

In

Fader

Multiband EQ

Send

Out

Group Macros

FMOD

Group
stingsGroup
tunesGroup
rename_pianoGroup
worldmap...whooshReturn
rename_verbGroup
design_toolsGroup
gameplay_sfxGroup
musicGroup
ui_sfMaster
Master BusSends
ren...erbSends
ren...erb

MONITOR FLIP

dB

10

SOLO

MUTE

SOLO

MONITOR

FLIP

dB

10

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

0

-10

-20

-30

-40

-50

-60

-70

-80

-90

-100

Pre

Volume

In

Post

Fader

Out

A

B

C

D

E

Send

LP 12dB

Off

Return

Off

Off

Group

Live Update Off

Platform

Desktop

Bullet Physics



R&D using Bullet Physics

SpeedTree





+ APIs - Application Programming interface



- Uma Interface de Programação de Aplicações, é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da programação.^[66]
- Como os jogos, muitas vezes, precisam acessar mecanismos de baixo nível, costuma-se criar camadas de abstrações em cima das APIs das plataformas, e “escondê-las” por meio de métodos contidos na biblioteca/ framework/ middleware/ engine. Esses métodos constituem uma “API de alto nível”, que lida com o gerenciamento dos processos da aplicação automaticamente “por baixo dos panos”.
- APIs Gráficas: OpenGL, DirectX, Metal, Vulkan;
- API de áudio espacial: Open AL.





+ Principais linguagens



- C/C++ (CryEngine, Godot, Unreal, entre muitos outros motores). Game engines são tipicamente codificadas em C/C++, e grande parte das APIs e SDKs também estão disponibilizadas nessa linguagem;
- C# (CryEngine, Godot, Monogame & Unity) é uma linguagem multi-paradigma, de tipagem forte, desenvolvida pela Microsoft, e popularmente usada na Unity, como opção principal;
- Java (Litiengine) é a opção óbvia quando se trata de mobile, porém são poucos os jogos escritos estritamente nela. Embora seja raro motores escritos em Java^[67] (é usada comumente com SDKs), ela sempre estará presente na plataforma Android;





+ Principais linguagens



- JavaScript (Web) é a linguagem mais usada atualmente, e embora não seja tão performática para jogos quanto as alternativas estáticas, ganhou muito espaço devido sua praticidade e popularidade. E, é claro, há inúmeros game frameworks web JS. Alguns motores recentes experimentam sobre estendê-la com a linguagem TypeScript.
- Lua (CryEngine, Solar2D SDK, LOVE2D & Pico8) é uma linguagem de script desenvolvida por brasileiros, projetada primariamente para uso incorporado^[68]. É também muito usada para scripting e MODs^[69] (modificações) em diversos jogos.
- Rust (Piston^[70]) é uma linguagem muito recente que vem ganhando espaço e popularidade entre programadores C++. É particularmente usada em conjunto com a tecnologia WebAssembly^[71] que permite compilar código para ser executado em navegadores web.





+ Principais linguagens



- Linguagens de domínio específico: GMSL (Game Maker), GDScript (Godot);
- Python (Ursina, biblioteca PyGame, Ren'py), não é comumente usada na base de código de jogos, devido suas limitações técnicas e performance, mas é extensamente usada como uma ferramenta auxiliar de automação, e criação de utilitários simples, assim como em Lua. Está presente em diversos softwares gráficos como Blender, Krita, GIMP, Maya, Photoshop, entre outros;
- Basicamente qualquer outra linguagem pode ser usada para criar jogos, cada uma tem suas vantagens e limitações (como suporte à plataformas), há diversos fatores que influenciam numa escolha





Visual Languages (Linguagens visuais)

- Linguagens visuais são linguagens de programação que permitem os usuários criarem programas manipulando elementos do software graficamente, ao invés de especificá-los via texto. Comumente são formadas por blocos, textos, símbolos e fios permitindo um arranjo visual das expressões. Também são chamadas de programação diagramática.^[72]
- Em geral, são projetadas para não programadores (em game dev são destinadas à artistas e designers), mas requerem conhecimentos básicos de programação.





Scene Project Debug Editor Help

2D 3D Script AssetLib

FileSystem res://

Favorites: res://

VisualScript (Godot)

File Edit Debug

Online Docs Classes Search Help

game left right ball ball_ceiling floor paddle

ball.png ball.vs ceiling_floor.vs icon.png left_pallete.png pong.tscn right_pallete.png

Function _on_area_entered area obj

TypeCast Is ball.vs? yes no

InstanceSet Object:direction obj instance pass obj

CallBasic Vector2.normalized() vector2

Variable ball_dir value

Construct new Vector2() x value

BuiltinFunc rand_range from -1 to 1

Age como uma extensão visual da linguagem integrada no motor: GDScript.
- Infelizmente foi descontinuado na Godot 4.

Scene Import

Filter nodes

Inspector Node

VisualScriptProperties

VisualScriptPropertySet

Set Mode Instance

Base Type Object

Base Script res://ball

Property direction

Index

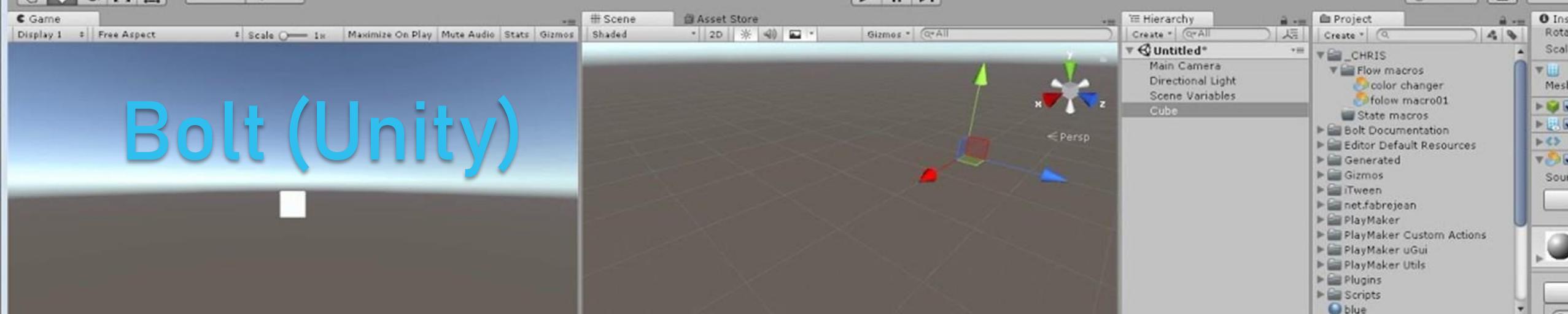
Assign Op Assign

Resource

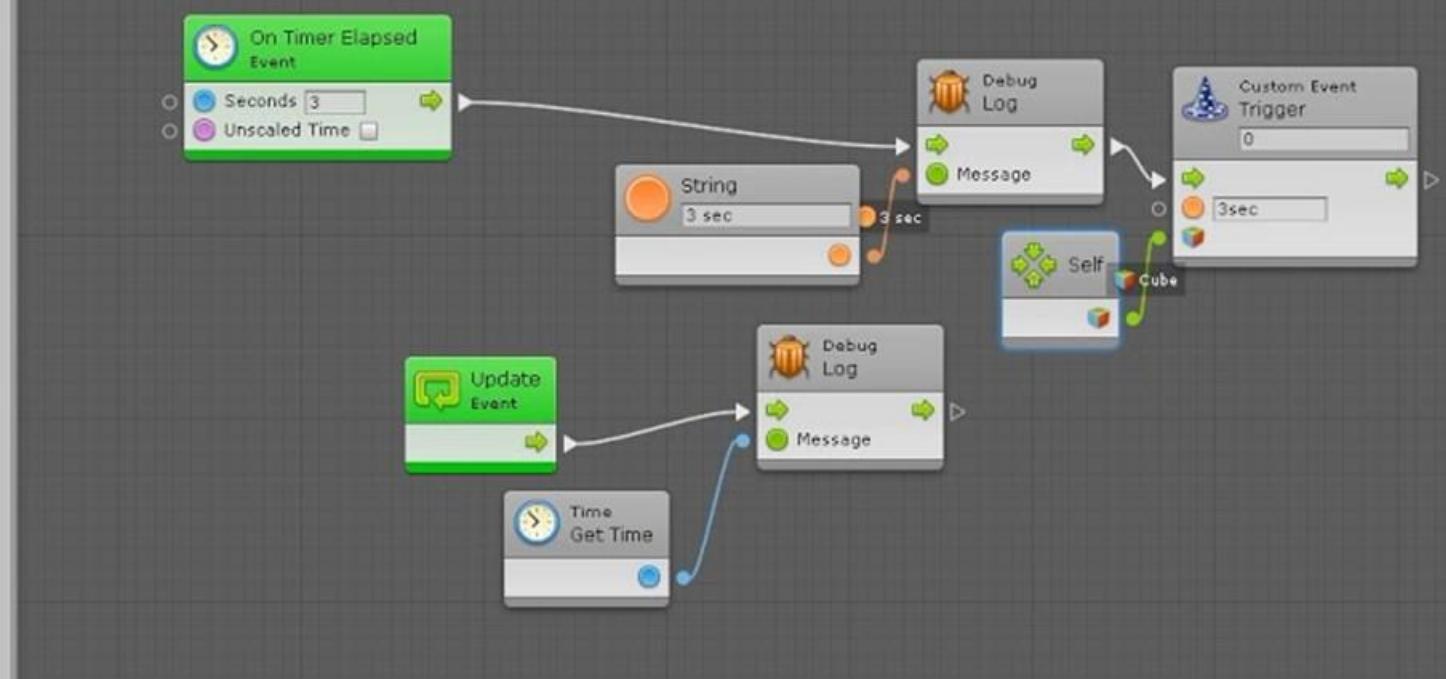
Reference

Script <null>

Bolt (Unity)



Inicialmente desenvolvida como um Plug-in na Unity Asset Store, foi recentemente comprada pela empresa para torná-la uma ferramenta padrão.





O Desenvolvedor de jogos

- Características dos profissionais da área;
- Pré-requisitos;
- Cursos.



Usar múltiplos chapéus



VS.



Usar o melhor chapéu

Desenvolver jogos requer uma gama de conhecimentos diversos. Ao trabalhar de forma independente, espera-se que o time ou desenvolvedor solo cumpra todos os papéis necessários. Logo, é comum que alguns campos do jogo se tornem menos atrativos que outros em jogos independentes de grande escopo.

Já os desenvolvedores AAA, tendem a ser mais especializados em áreas específicas, com exceção dos Game Designers e Project Managers que geralmente lideram os diversos setores que compõem o estúdio, cada área possui diversas vertentes de especializações.



+ Desenvolvimento independente

- Um programador de jogos independentes costuma ter mais liberdade criativa que artistas, devido sua familiaridade com os diversos conceitos necessários para criação do jogo. Mas, são raros aqueles que obtêm sucesso trabalhando sozinhos.
- Desenvolvedores indie costumam ser mais individualistas devido terem uma visão romantizada do *seu projeto dos sonhos*. Por essa razão, é muito difícil formar times duradouros sem recorrer a uma solução judicial, por meio de contratações.
- Estúdios independentes são negócios de risco, muitas vezes iniciados por meio de startups. É comum que esses estúdios recorram a outras formas de financiamento para se manterem, como as campanhas de crowdfunding, ou venda de produtos e serviços alternativos.



A grande indústria (AAA & Pequenos estúdios)

- É notavelmente difícil conseguir um bom emprego num estúdio de jogos equiparado a outras empresas de tecnologia.
- Pequenos estúdios costumam oferecer propostas muito arriscadas, e salários muito baixos.
- Estúdios de médio porte são muito procurados, e as vagas acabam tendo um alto grau de competição.
- Empresas AAA são ainda mais seletivas e requerem um alto grau de conhecimento técnico por parte do candidato, que ainda deverá estar propenso as necessidades únicas e específicas daquela empresa, e ainda é sujeito a um trabalho intensivo.
- Note que não há reais estúdios AAA no Brasil, atualmente.





+Tipos de desenvolvedores

- Tech Artists:
 - Processes automation, DevOps;
 - Optimization;
 - Processing, VFXs...
- Core developers (engineers):
 - Network engineers;
 - Graphics Engine engineers;
 - Physics Engine engineers...
- Designers:
 - Front-end;
 - Level designers;
 - Character designers;
 - Gameplay programmers (scripters)...
- Quality Assurance:
 - Playtest;
 - Automated tests;
 - Tweaking & Refinement...

Engenheiros de software estão envolvidos em todas as áreas.

Skills necessárias

HARD SKILLS

- Matemática
 - Noções de física (cinemática),
 - Probabilidade e estatística,
 - Cálculo (numérico, vetorial),
 - Álgebra linear, etc...
- Pensamento lógico (algoritmos)
- Conhecimentos de computação
 - Estruturas de dados;
 - Paradigmas de programação;
 - Linguagens de programação;
 - Arquiteturas e Sistemas Operacionais;
 - Optimização;
 - Computação Gráfica...
- Design
 - Engenharia de software;
 - Design gráfico;
 - Design de som;
 - Design de produtos...
- Outros
 - Redes de computadores;
 - Inteligência Artificial...

You are filled
with

SOFT SKILLS

- Inglês técnico
- Criatividade
- Trabalho em equipe
- Noções artísticas
- Paixão por jogos
- Determinação!



DETERMINATION

Tópicos Gerais

CIÊNCIA DA COMPUTAÇÃO

- Algoritmos e estruturas de dados;
- Fundamentos de Linguagens de programação;
- OOP (Programação Orientada à Objetos);
- Engenharia de Software;
- Networking (Redes);
- Interface Homem-Máquina...

MATEMÁTICA

- Trigonometria;
- Álgebra Vetorial & Geometria Analítica;
- Álgebra Linear;
- Física (Mecânica & Cinemática);
- Probabilidade & Estatística, e Teoria dos Jogos (Game Design)...

OUTROS

- Scripting;
- Rendering & Processing (Computação Gráfica);
- IA (Inteligência Artificial) para jogos;
- Programação para dispositivos móveis;
- VR & AR (Realidade Virtual & Aumentada...)

Competições: “Game Jams”

- Espécie de Hackathon^[1] de jogos
- Participações individuais ou em times
- Produzir um jogo do zero
(com as ferramentas adequadas)
- Período limitado de tempo
 - geralmente alguns poucos dias,
 - varia de 6h até 2 semanas,
 - mas podem levar até 1 ou 2 meses^[2]
- Pode ocorrer presencialmente ou online
- Costuma abranger algumas condições e regras:
 - um tema,
 - um conjunto de limitações,
 - regras específicas
- Grandes Jams costumam ser premiadas:
 - Prêmios variam de Jogos, materiais didáticos à algumas centenas de dólares.
 - “*O maior prêmio é a experiência e os amigos que fazemos ao longo da jornada*” :P
- Parte da subcultura da internet:
a *demoscene*.^[3]



+ Global Game Jam

- Criada em 2009.
- Ocorre anualmente.
- Maior Game Jam presencial do mundo.
- Sediada em até 110 países^[4]



Ludum Dare

- Termo em latim para “dar um jogo”.
- LDJAM é a segunda maior *Game Jam* online do mundo;
 - rivalizando com a *GMTK Jam*.
- Iniciou como um fórum da internet;
 - primeira *Game Jam* em 2002, com apenas 18 participantes.
- Tempo limite original de 24h foi aumentado para 48h.
- Ocorre duas vezes ao ano.
- Em Abril de 2020, alcançou 10330 inscrições, e 4959 jogos.
- Edição mais recente: 51º (Out, 2022)
 - Tema: “*Every 10 seconds*”;
 - 4887 entradas e 2414 submissões.^[5]





G M T K + GAME JAM2022

- Abreviação de “Game Makers Toolkit” que é o maior canal sobre Game Design no YouTube atualmente.
- Sediada no site Itch.io.
- Ocorre uma vez por ano.
- Alcançou um total de 5371 entradas em Julho de 2020, superando, pela primeira vez, a LDJAM, tomando seu pódio.^[6]
- Edição mais recente: 6° (Jul, 2022)
 - Tema: “roll the dice”;
 - 22,077 entradas e 6,168 submissões.^[7]



GAME JAAJ 7

- Maior game jam online nacional
- Produzida pelo criador de vídeos e estudante de biologia conhecido como “Amdré Young”.
- Também sediada no itch.io.
- Alcançou um total de 304 submissões na competição mais recente (Jan, 2022) com um prêmio acumulado de R\$4,000.^[8]
- Presença do bom e velho “jeitinho brasileiro”



O Mercado dos sonhos

O mercado de jogos independentes é repleto de jovens apaixonados pelo meio e amadores de diversas idades. O mercado brasileiro é ainda muito informal, e carece das iniciativas necessárias para atender essa demanda.

Em geral, os amadores se tornam *hobbiestas*, trabalhando em seus projetos nos seus tempos livres, sacrificando parte do seu tempo de lazer.

Trabalhar como programador ou artista freelancer requer diversos sacrifícios.

É comum que aqueles que desejam se profissionalizar tenham a necessidade de imigrar para as capitais, outros estados, ou até mesmo, para fora do país.



+ É possível ser desenvolvedor de jogos no Brasil!

O mercado de desenvolvimento de jogos no Brasil ainda é muito novo em comparação à indústria no exterior, e está dando os seus primeiros passos.

Assim como a indústria tecnológica em geral, há muitos avanços necessários para a indústria brasileira de jogos acompanhar as diversas tendências e conquistas alcançadas em países mais desenvolvidos.

Mas há sim algumas oportunidades em território nacional. Além de que, graças a internet, é possível estudar, desenvolver e comercializar jogos sem sequer sair de casa.



Cursos

Há pouquíssimos cursos de graduação em desenvolvimento de jogos no Brasil, e a grande maioria deles tendem a ser mais generalistas, abrangendo conhecimentos de engenharia de software, matemática computacional, design, até marketing e administração.

Cursos técnicos costumam especializar-se em ferramentas mais específicas como motores de jogos, softwares gráficos, ou vertentes de game design. Muitos deles podem ser cursados via EAD.

No Brasil, há tantos cursos técnicos sobre desenvolvimento de jogos quanto estúdios de jogos em si.



Pré-requisitos

O que é necessário para se tornar um programador de jogos?

- Dominar algum motor de jogo dentre as várias opções;
- Ter prática e familiaridade em uma ou mais linguagens;
- Conhecimentos gerais sobre programação de jogos e desenvolvimento de software;
- Tempo;
- Determinação & Persistência;
- ...e muita Criatividade!





Choque de realidade!

CULTURE

The tragic end of Telltale Games

How an award-winning studio abruptly shuttered, as told by the people who were there

By MEGAN FAROKHMANESH / @megan_nicolett
Illustrations by ALEX CASTRO

Oct 4, 2018, 10:27 AM GMT-3 | □ 0 Comments / 0 New



When employees showed up for work on Friday, September 21st, at Telltale Games, there was nothing to suggest the day would be different

1 Here's what it's like to wear the Dyson Zone





+ Projetos Brasileiros



- Desenvolver jogos no Brasil é um grande desafio, principalmente de forma independente. Devido a falta de incentivos governamentais, a escassez de cursos profissionalizantes de qualidade atualizados com as tendências mundiais, além da pouquíssima quantidade de materiais em português, os custos de desenvolvimento, entre outros fatores criam uma grande barreira para esses desenvolvedores.
- A maioria dos indies brasileiros são amadores lutando por um pouco de espaço na sufocante competição do mercado internacional, através das grandes Games Stores (GOG, Steam, Itch.io, Apple App Store, Google Play Store & Epic Game Store);
- Muitos daqueles que se profissionalizam em Game Dev no Brasil acabam imigrando para outros países.
- O mercado internacional é repleto de brasileiros. Por trás de grandes jogos 3A há sempre alguns brasileiros talentosos.
- **Valorize o artista nacional!**



Game dev. is hard!

Perguntas?

Questions?

What is man? A
miserable
little pile of
secrets?

What is the
meaning of
life?

Doubts?

Is it Worth
it?

Never gonna
give you up,
never gonna
make you
cry...





THE INDIE GAME DREAM





Referências

Os links de materiais usados nesta apresentação estão disponíveis em
github.com/GersonFeDutra/gd_start

Agradecimentos:

Daniel Nascimento

<https://github.com/danielnasc>

Miguel Reis de Araújo

<https://github.com/reisaraujo-miguel>

Nas próximas seções...

- Conceitos fundamentais de programação de jogos
- + • Introdução ao motor de jogos Godot e sua linguagem de scripting
- “Hello World Game”: Flappy Bird!