



# Jogos Digitais

Saulo Popov Zambiasi  
[saulopz@gmail.com](mailto:saulopz@gmail.com)



# Sumário

- O que é um Jogo
- Equipe de Desenvolvimento
- API's e Bibliotecas
- Motores de Jogos
- Empresas de Jogos Digitais em Florianópolis



Mortal Kombat

# O que é um jogo?

- Regras
- Resultados quantificáveis
- Resultados com diferentes valores (positivos e negativos)
- O esforço do jogador deve influenciar o resultado (desafios)
- Podem ou não ter consequências na vida real
  - Podem ser uma atividade produtiva - apostas, jogo profissional
  - Pode "machucar" de verdade

# Exemplos de Jogos

- Esportivos: Futebol, Tênis, Vôlei...
- Tabuleiro: Resta Um, Xadrez, Damas...
- Jogos de Papel-e-caneta: Palavras cruzadas, Caça palavras, Sudoku...
- Jogos de Dados: Jogo do General, Craps...
- Jogos de Cartas: Pôquer, Escova, Truco, Uno...
- Jogos Eletrônicos: Skyrim, Counter Strike, GTA, D&D, War, Doom, World of Warcraft, Nier Automata 2B...

# O que é um Jogo Eletrônico?

- Software composto de :
  - Imagens
  - Música e Sons (Efeitos sonoros + trilha)
  - Enredo ou História
  - Interface
  - Regras
  - Controles



# O que é um Jogo Eletrônico?

- Objetivos :
  - Divertir, Entreter e/ou :
  - Aprender algo
  - Trabalhar uma habilidade
  - Desenvolver habilidades novas
  - Testar cenários...



# Tipos de Jogos Eletrônicos

- De ação: Doom, Counter Strike
- De estratégia: Civilization
- De RPG: Final Fantasy
- De esportes: Fifa Soccer



# Tipos de Jogos Eletrônicos

- De simulação de veículos: Need for Speed
- De aventura: The Legend of Zelda
- MMOs: World of Warcraft
- Simulações: Sim City, The Sims
- Serious Game: Army Battlezone



# Equipe de Desenvolvimento



Genshin Impact

# Equipe de Desenvolvimento

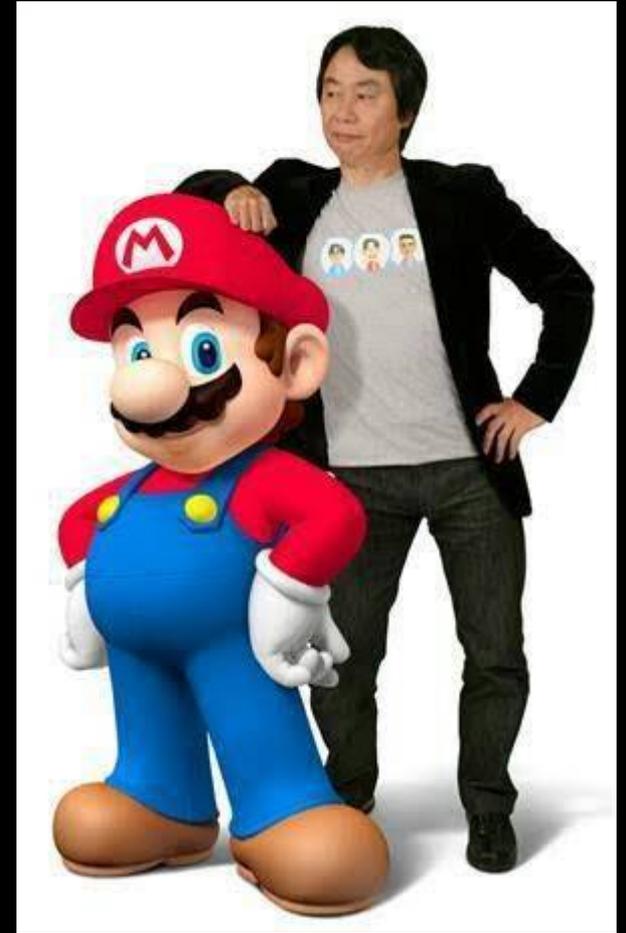
- Alguns dos profissionais envolvidos no desenvolvimento de jogos digitais:
  - Game Designer
  - Artista
  - Sonoplasta
  - Programador
  - Tester



Anime New Game

# Game Designer

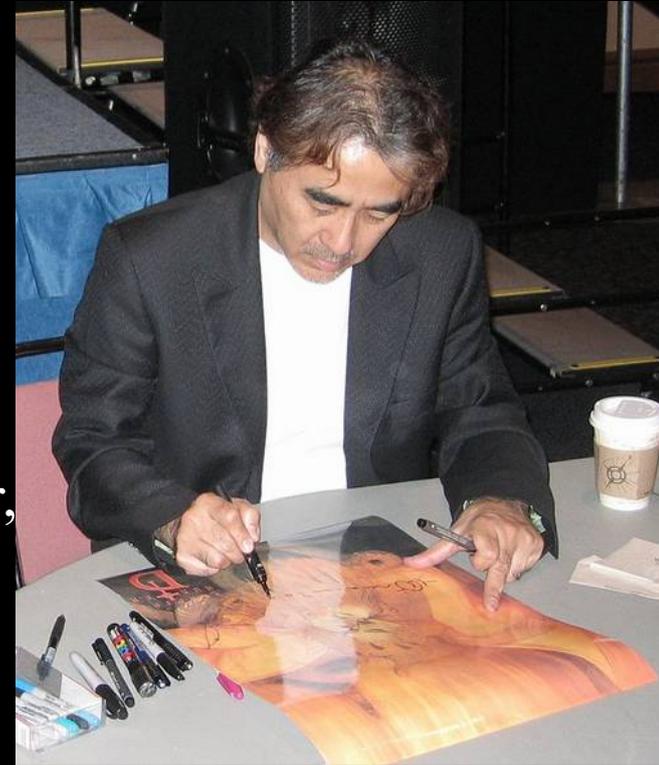
- Responsável pelas ideias do jogo
  - Cria as histórias, as regras, os personagens, o tema, etc.
  - Responsável por criar um jogo divertido e cativante.
- Game Designer pode ser um pouco de tudo
  - escritor, programador, artista...
- Deve conhecer:
  - Mercado, público alvo, consumidores, física, lógica, literatura, psicologia, articulação verbal e escrita (português e inglês)
- Cria o *Game Design Document* (GDD)
  - Detalha documentação, escreve resumo, etc.



Shigeru Miyamoto  
Zelda, Mario, etc.

# Artista

- Responsável pela parte visual do jogo
- Transforma as ideias em imagens
- Desenha o jogo usando softwares de desenho:
  - desenhos, ilustrações, modelagem dos personagens, animações, texturas, etc.
- Utiliza softwares como GIMP, Photoshop, Illustrator, Maya, Blender, etc.
- Os artistas são responsáveis por elementos específicos:
  - Cenários, personagens, objetos, etc.



Yoshitaka Amano  
Final Fantasy, etc.

# Sonoplasta

- Cria a sonorização dos jogos
  - Trilha sonora
  - Sonoplastia
  - Vozes
- Deve conhecer softwares próprios para sonorização:
  - Audacity, Audio Vorbis...



Mark Lampert  
Sonoplasta de Skyrim, Fallout

# Programador

- Faz a mecânica do jogo acontecer
- O programador transforma as ideias e regras do game design em lógica.
  - Linguagens: Assembler, C, C++, C#, Java, Python...
  - Resolução de problemas complexos, IA...
  - Noção de matemática e física: vetores, álgebra linear...
  - Apis Gráficas: OpenGL, DirectX, Vulkan, LibGDX, SFML, SDL2...
  - Motores: Unity3D, Unreal Engine, Godot, XNA...



Sid Meier – Civilization

# Tester

- Efetua testes no jogo
  - Joga sim, mas não se preocupa só com jogar!
  - Segue uma *checklist* de testes e faz anotações dos problemas encontrados
- Precisa ter conhecimento de mercado, de público, concorrência, jogabilidade e ergonomia.



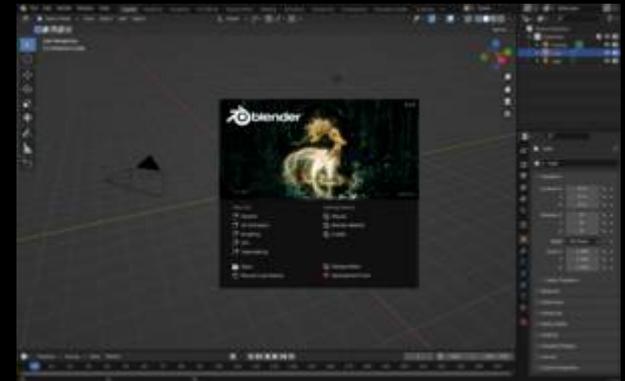
Anime New Game  
Nene Sakura – Tester  
Umiko Ahagon – Programmer

# API's e Bibliotecas

```
MotorApplication.cpp - motor - Visual Studio Code
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
EXPLORADOR
EDITOR ABERTOS
MotorApplication.cpp x MotorApplication.h Globals.h Sketch.cpp Sprite.cpp Textu
src > MotorApplication.cpp > ...
MOTOR
  AnimatedSprite.cpp 54
  AnimatedSprite.h 55
  Body.cpp 56
  Body.h 57
  Color.cpp 58
  Color.h 59
  Event.cpp 60
  Event.h 61
  Font.cpp 62
  Font.h 63
  Globals.h 64
  Log.cpp 65
  Log.h 66
  MotorApplication.cpp 67
  MotorApplication.h 68
  Music.cpp 69
  Music.h 70
  PhysicBody.cpp 71
  PhysicBody.h 72
  Rect.cpp 73
  Rect.h 74
  Singleton.h 75
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
begin();
54
55
56 Uint32 tickBegin = SDL_GetTicks();
57 Uint32 tickEnd;
58 int frameCounter = 0;
59 bool deltaCalculated = false;
60
61 while (running)
62 {
63     // EXECUTION BEGIN *****
64     if (deltaCalculated)
65     {
66         event->update();
67         sketch->run();
68         execute();
69         sketch->cleanner();
70
71         sketch->begin();
72         render();
73         sketch->draw();
74         sketch->end();
75     }
76     // EXECUTION END *****
77
78     //SDL_Delay(10); // Just while i dont config and calculate limit
79
80     // FPS COMPUTE
81     frameCounter++;
82     if ((tickEnd = SDL_GetTicks()) - tickBegin > 1000)
```



- OpenGL - *Open Graphics Library* é uma API livre usada para
  - computação gráfica
  - desenvolvimento de aplicativos gráficos
  - ambientes 3D
  - Jogos, etc.
- Produzida com C e C++, mas pode ser utilizada para diversas outras com um alto nível de eficiência
- Minecraft, Adobe Premiere e After Effects, Blender, AutoCAD, 3D Studio Max, etc.
- <https://www.opengl.org>



# Microsoft DirectX

- Coleção de APIs de multimídia com foco em jogos digitais
- Sistema operacional Microsoft Windows
- Atualmente na versão 12
  - Windows 10, Xbox One e Windows 7
- Linux gamers usam **Wine** para executar jogos desenvolvidos com DirectX



# Wine – Jogos Windows no Linux

- Wine (WINE Is Not an Emulator)
  - camada de compatibilidade para sistemas operacionais UNIX
  - implementação da API da Microsoft Windows
  - permite a execução de aplicações desenvolvidas para ambientes Windows nativamente em outros sistemas operacionais



# Vulkan®

- API de computação e gráficos 3D multiplataforma de baixo overhead criada pela Khronos
- derivada e construída sobre os componentes da Mantle API da AMD (doada pela AMD para a Khronos)
- Objetiva ser uma API de baixo nível para padronizar a indústria, tal como OpenGL
- <https://www.vulkan.org>

Vulkan Platform Support



# libGDX

- Estrutura de aplicação de desenvolvimento de jogos livre e de código aberto
- Escrito em Java com alguns componentes de C e C+
- Multiplataforma, suportando Windows, Mac, Linux, Android, IOS, BlackBerry e HTML5
- <https://libgdx.com>

# libGDX



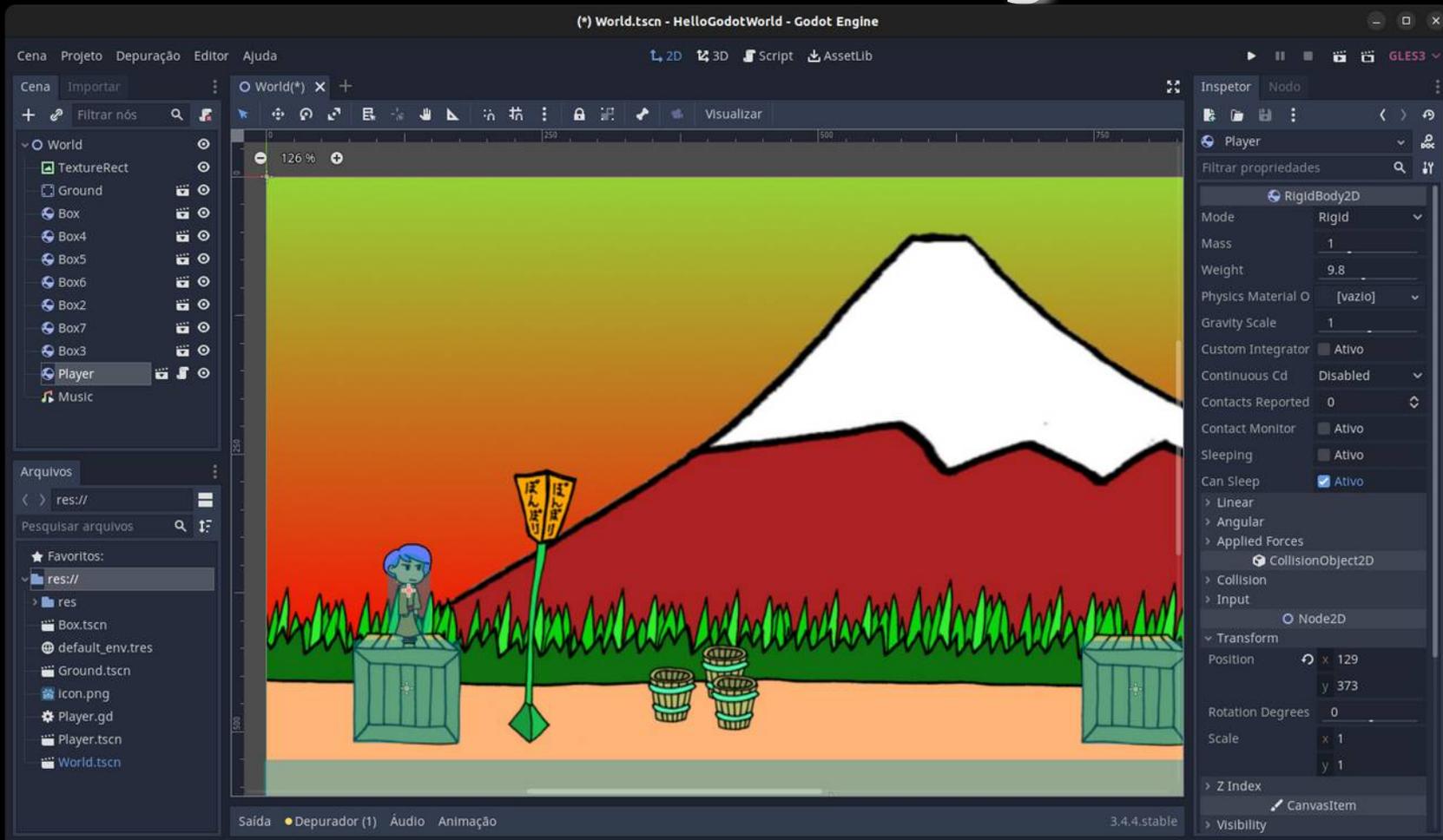


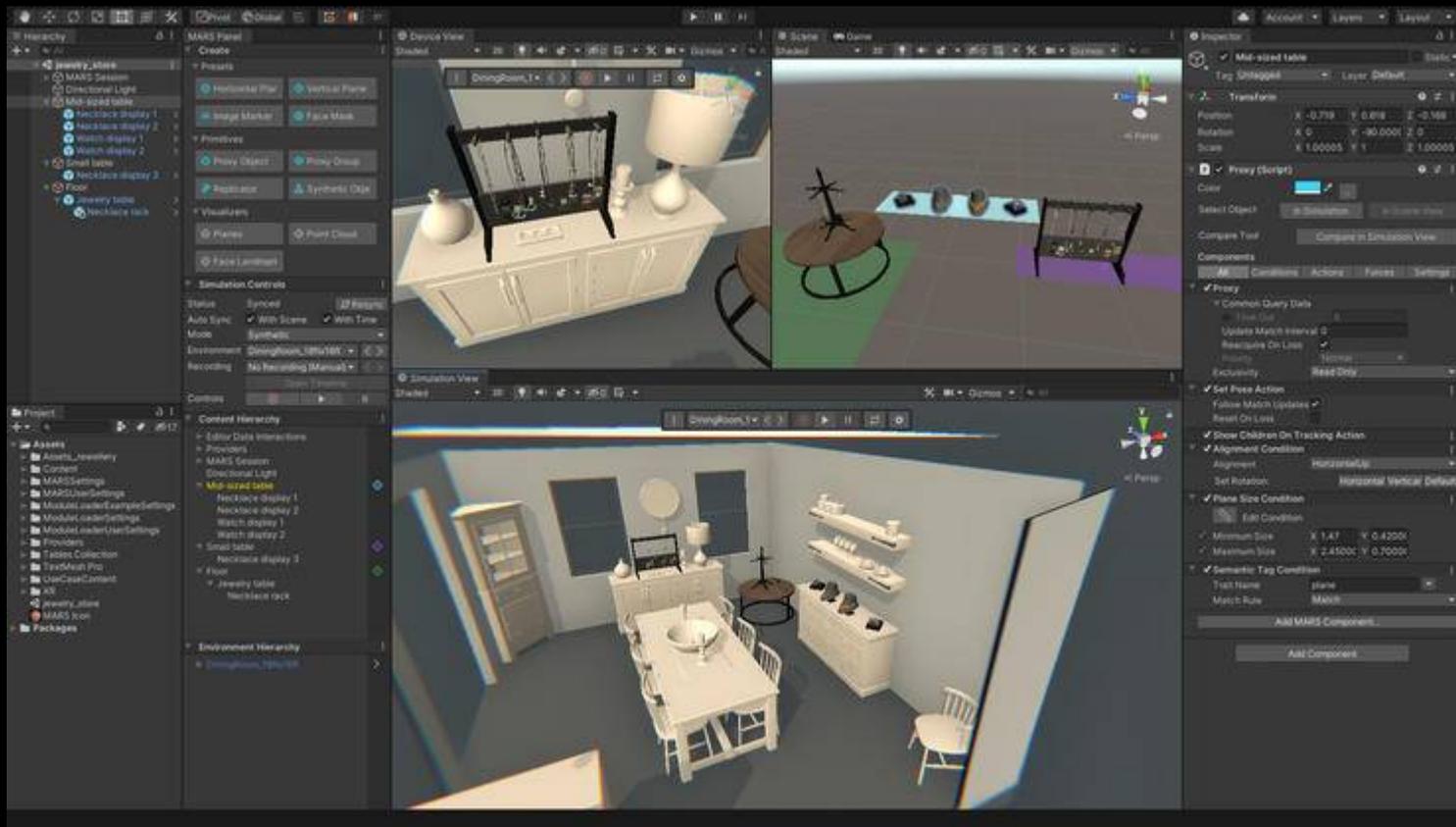
- SFML – *Simple and Fast Multimedia Library*
- Biblioteca multiplataforma, orientada a objetos e livre
- API multimédia escrita em C++ por Laurent Gomila
- Visa fornecer uma interface simples para vários componentes multimédia do computador.
- Compatível com linguagens como C, D, Java, Python, Ruby, .NET, Go, Rust, OCaml, Euphoria e Nim.
- Multiplataforma: Windows, Linux, OS X, Android e iOS
- <https://www.sfml-dev.org/>



- *SDL – Simple DirectMedia Layer*
- Biblioteca multimídia livre e de código aberto, multiplataforma, escrito em C
- Representa uma interface simples para gráficos, som, e dispositivos de entrada
- Multiplataforma: Android, iOS, Linux, Mac OS X, Windows e outras plataformas não-suportadas oficialmente
- A biblioteca manipula vídeo, eventos, áudio digital, CD-ROM, threads, carregamento de objetos compartilhados, rede e temporizadores
- <https://www.libsdl.org>

# Motores de Jogos







- Ferramenta de criação de jogos em 2D e 3D
- Suporta Direct3D no Windows e Xbox 360; OpenGL no MacOS e Linux; OpenGL ES no Android e iOS; WebGL na Internet
- Utiliza o MonoDevelop para a criação dos scripts (C#, descontinuou as outras)
- Jogos 2D com uso de sprites e um avançado renderizador de mundo 2D



- Jogos 3D
  - compressão de textura
  - mipmaps e configurações de resolução
  - mapeamento de relevo
  - mapeamento de reflexão
  - mapeamento de paralaxe
  - oclusão de ambiente de espaço de tela
  - sombras usando mapas de sombras
  - efeitos de pós-processamento de renderização para textura e tela inteira.



- Suporta
  - criação de vértices
  - fragmentos (ou pixels) personalizados
  - shaders de computação, etc.
- Serviços para desenvolvedores
  - Unity Ads, Unity Analytics, Unity Cloud, Unity Everyplay, Unity IAP, Unity Multiplayer, Unity Performance Reporting, Unity Collaborate e Unity Hub.
- <https://unity.com>





# UNREAL ENGINE

Unreal Engine interface showing a motorcycle model in the viewport, with various toolbars and panels.

**Viewport:** Perspective view of a blue motorcycle. Statistics: Current Screen Size: 1.670149, #Static Meshes: 601, #Meshes drawn: 2,015, #Triangles Drawn: 4774,419, #Vertices Used: 4,012,289, Approx Size: 87x214x122.

**Content Browser:** Shows a hierarchy of assets including Demo, Front\_plastic\_x\_..., Gas\_tank\_..., and Unreal\_Sportbike\_SLD/.

**Class Browser:** Lists actors and meshes such as Front\_plastic\_x\_..., Gas\_tank\_..., Unreal\_Sportbike, Cable\_1, Clutch\_cab, Clutch\_c, Cooler\_inje, and Cooler\_inje.

**Properties Panel:** Shows inputs, output, transform (Location, Rotation, Scale), static mesh, materials, and physics settings.

**Dataprep Graph:** A workflow graph with nodes: Tesselation (Datasmith Tesselation), Assign Materials (Substitute Material By Table), and Cleanup. Includes a 'Select By' panel with options like Condition, Float, String, and Operations.

**Watermark:** DATAPREP



- Desenvolvido pela Epic Games (nova versão 5)
- Núcleo escrito em C++
- Múltiplas plataformas
  - Microsoft Windows, Linux, Mac OS e Mac OS X (PC) e consoles Dreamcast, GameCube, Nintendo Switch, PlayStation 2, PlayStation 3, PlayStation 4, PlayStation 5, Stadia, Wii, Wii U, Xbox, Xbox 360, Xbox One e Xbox Series X
- <https://www.unrealengine.com>



**UNREAL**  
ENGINE

- **Introduz a Tecnologia LUMEN**
  - Nova tecnologia de iluminação do Unreal Engine 5
  - Visa competir com o ray tracing.
- **Ray Tracing**
  - Executa uma série de cálculos complexos para determinar como a luz se comporta e interage com os objetos presentes na cena:
    - intensidade, ângulo e posição da luz na cena
    - material de cada objeto e como esses refletem a iluminação
    - posição e ângulo da câmera do jogador em relação ao cenário
  - Define e aplica as sombras e os reflexos nos locais apropriados e com a intensidade correta.
  - Exige hardware específico, caro e inacessível para a maioria dos jogadores



**UNREAL**  
ENGINE

- LUMEN

- Cálculos avançados de iluminação e reflexos a um custo computacional bem baixo, em comparação com as outras tecnologias.
- Simulação realista do comportamento da luz no cenário virtual mesmo em hardwares mais básicos.
- Desenvolvedores não precisam se preocupar com técnicas como pré-renderização de sombras pois a Lumen cuida da iluminação do jogo, processada em tempo real, resultando em um ambiente muito mais realista e natural.



**UNREAL**  
ENGINE



EPIC  
GAMES

UNREAL ENGINE 5

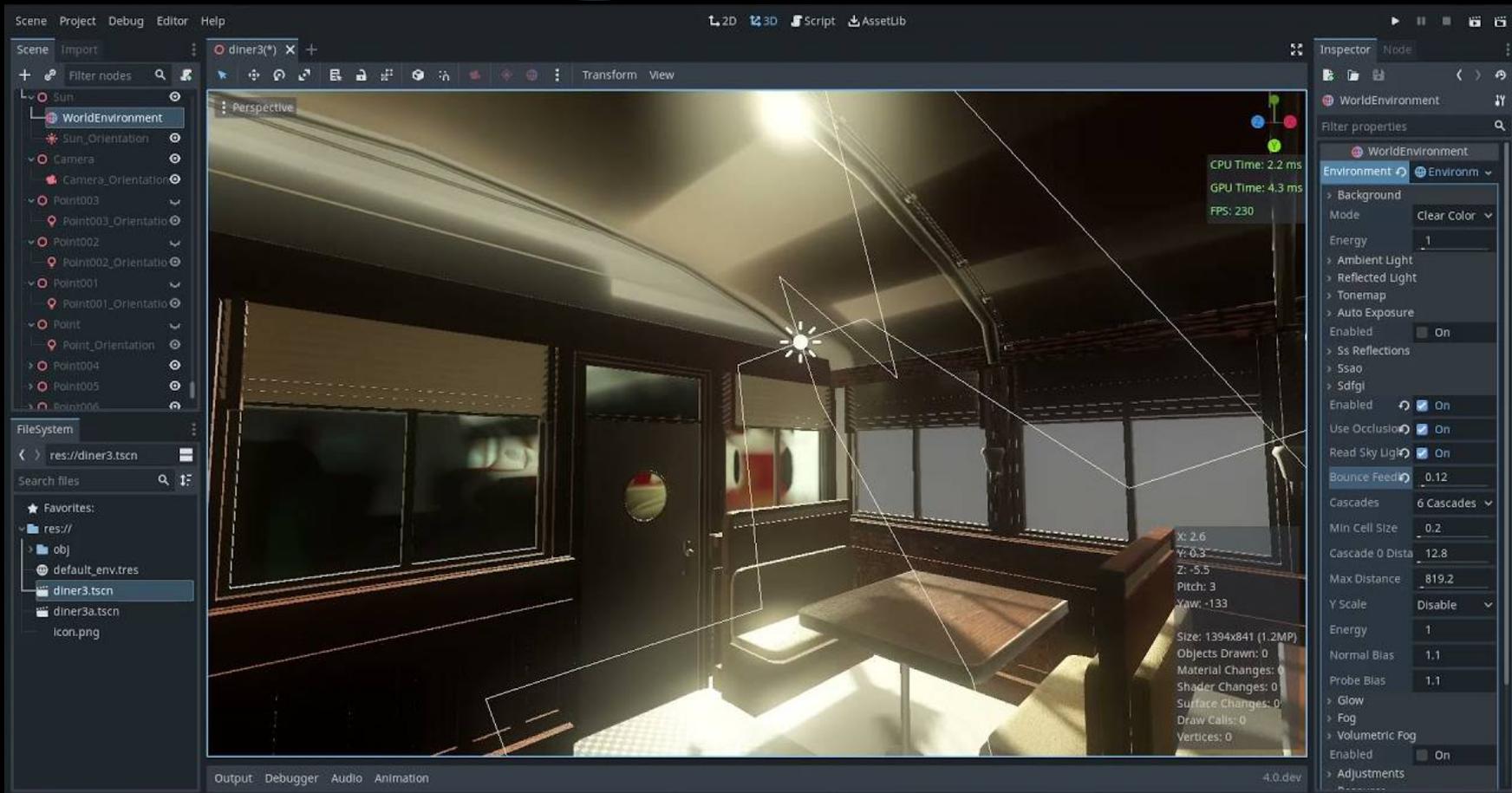


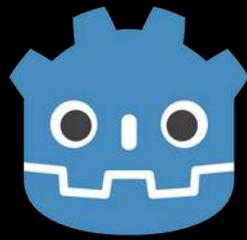
**UNREAL**  
ENGINE





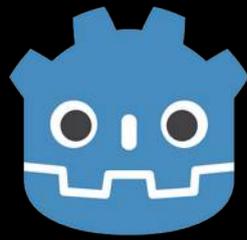
# GODOT





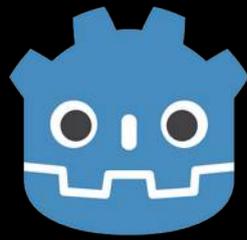
# GODOT

- Motor de jogo de código aberto
- Desenvolvimento iniciado por Juan Linietsky e Ariel Manzur em conjunto com a OKAM Studio em 2001 na Argentina.
- Fevereiro de 2014: código fonte foi publicado no GitHub sob a licença MIT
- 15 de dezembro de 2014: Primeira versão estável
- Atualmente com a versão 4 em desenvolvimento
- Visa fornecer um ambiente de desenvolvimento de jogos totalmente integrado
- Apenas um binário de execução bastante leve



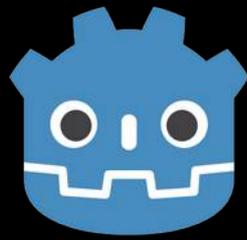
# GODOT

- Permite que os desenvolvedores criem um jogo, sem precisar de outras ferramentas
  - além das usadas para a criação de conteúdo (recursos visuais, música, etc.)
- A arquitetura do motor é construída em torno do conceito de uma árvore de "nós".
  - Os nós são organizados dentro de "cenas", que são grupos de nós reutilizáveis, instáveis, herdáveis e aninhados.
  - Todos os recursos do jogo, incluindo scripts e ativos gráficos, são salvos como parte do sistema de arquivos do computador
  - Esta solução de armazenamento visa facilitar a colaboração entre equipes de desenvolvimento de jogos usando sistemas de controle de versão de software



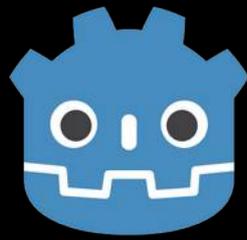
# GODOT

- Trabalha com quatro tipos de linguagens:
  - GDScript
    - Linguagem principal do programa.
    - Sintaxe semelhante ao Python.
  - C#
    - Adicionado na versão 3.0.
    - Visa atender o padrão da indústria.
  - VisualScript:
    - Programação via interface gráfica para pessoas que não sabem programar ou programadores que desejam expor comportamentos do jogo para projetistas, artistas, etc.
  - GDNative:
    - Semelhante ao C++.
    - utilizada para modificação da própria engine, criar extensões e novas funções.



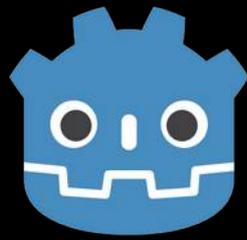
# GODOT

- Permite a especificação de compressão de textura e configurações de resolução para cada plataforma.
- Executa no Linux, macOS e Microsoft Windows, mas permite outras plataformas.
- Oferece suporte a games para multi plataformas:
  - Mobile Android, iOS
  - Desktop Linux, macOS, Microsoft Windows, BSD
  - Web HTML5, WebAssembly
  - Realidade virtual / estendida HTC Vive, Valve Index, Oculus Rift, Oculus Go, Oculus Quest, todos os fones de ouvido Microsoft MR, ARKit da Apple, etc.



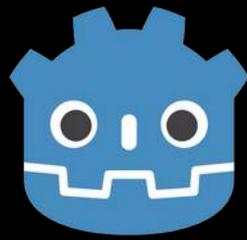
# GODOT

- Sistema de animação sofisticado com a capacidade de
  - esqueletos (skeletons)
  - inversão cinética (inverse kinematic)
  - deformação (morphing).
- Sistema de física integrado para 2D e 3D que suporta
  - colisões
  - objetos rígidos
  - cinética de corpos, etc.



# GODOT

- Sistema de animação sofisticado com a capacidade de
  - esqueletos (skeletons)
  - inversão cinética (inverse kinematic)
  - deformação (morphing).
- Sistema de física integrado para 2D e 3D que suporta
  - colisões
  - objetos rígidos
  - cinética de corpos, etc.



# GODOT

<https://wccftech.com/steam-deck-wont-have-any-exclusive-games-valve-confirms/>

**You mentioned that you're talking with Unity and Epic, are you also talking to Godot?**

Yes, we're talking to Godot as well and are actively supporting them and want their engine to work well with Steam Deck.

**Sponsored by**



Heroic Labs

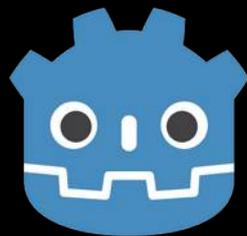


GAMBLIFY

*spiffcode*



**ASIFA**  
HOLLYWOOD



# GODOT



# Empresas de Jogos Digitais em Florianópolis





- Desenvolvedora e publicadora de jogos fundada em 2004
- Localizada no Bairro Santa Mônica em Florianópolis
- <http://www.hoplon.com>
- Jogos:
  - Heavy Metal Machines
  - Taikodon
  - Apocalypse Party's Over



# Hydra Systems



- Desenvolvedora de Jogos para Cassinos
- 7 anos no mercado, com mais de 70 jogos lançados e mais de 8.000 devices ativos.
- <https://www.hydrasystems.com.br>



# Crazy Minds

- Estúdio Independente de Jogos para Android, iOS, Web e PC



# Casthalia Studio

- *Serious Games*
- Criação de jogos e outros tipos de tecnologias educacionais



# Cat Nigiri



- Jogos Independentes
- Trindade – Florianópolis
- <http://www.catnigiri.com>



# MAQNA

- Desenvolvedora de games pra Web, Mobile, Console e PC.
- <http://www.magna.com.br>



# WARP LEMON

- Desenvolvimento e Design de Games
- <https://warplemon.com>



**Marcelo T. Rosa**  
CEO & Programador Chefe

**André T. Rosa**  
CEO & Designer Chefe



# Jogos Digitais

Saulo Popov Zambiasi  
[saulopz@gmail.com](mailto:saulopz@gmail.com)

