



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO TOCANTINS-CAMPUS ARAGUAÍNA
CURSO SUPERIOR EM TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

EDIMONES FILHO ALVES BEZERRA ALMEIDA

PRINCIPAIS ENGINES PARA DESENVOLVEDORES DE JOGOS INDIE

**ARAGUAÍNA– TO
2022**

EDIMONES FILHO ALVES BEZERRA ALMEIDA

PRINCIPAIS ENGINES PARA DESENVOLVEDORES DE JOGOS INDIE

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do grau de Tecnólogo em análise e desenvolvimento de sistemas no Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, concentrado na área de Educação.

Orientadora: Esp. Maríly Lima da Conceição

ARAGUAÍNA - TO

2022

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

A447p Almeida, Edimones Filho Alves Bezerra de
Principais Engines para Desenvolvedores de Jogos Indie /
Edimones Filho Alves Bezerra de Almeida. – Araguaína, TO, 2022.
49 f. : il. color.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e
Desenvolvimento de Sistemas) – Instituto Federal de Educação,
Ciência e Tecnologia do Tocantins, Campus Araguaína, Araguaína,
TO, 2022.

Orientadora: Esp. Maríly Lima da Conceição

1. Desenvolvedor Indie. 2. Engine. 3. Jogos. I. Conceição, Maríly
Lima da. II. Título.

CDD 004.2

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins
de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos
pelo(a) autor(a).

EDIMONES FILHO ALVES BEZERRA ALMEIDA

PRINCIPAIS ENGINES PARA DESENVOLVEDORES DE JOGOS INDIE

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do grau de Tecnólogo em análise e desenvolvimento de sistemas no Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, concentrado na área de Educação.

Orientadora: Esp. Maríly Lima da Conceição

Data da Aprovação: ____/____/2022

BANCA EXAMINADORA

Marily Lima Da Conceição

Prof. Esp Marily Lima da Conceição– **ORIENTADORA**
Instituto Federal do Tocantins – Campus Araguaína

Prof. M.e. Katiane Ribeiro Lopes
Instituto Federal do Tocantins – Campus Araguaína

Jonathon Viana Silva
Prof. Esp Jonathon Viana Silva
Instituto Federal do Tocantins – Campus Araguaína

RESUMO

Desenvolvedor Indie de Games, geralmente é quem tem uma ideia de um jogo, e por conta própria fica responsável pelo o desenvolvimento do mesmo, sem contar com uma equipe, ou grandes estúdios muitas vezes usados por muitas empresas maiores. É um desafio, que pode se tornar o caminho mais difícil, porém é algo que muitos desenvolvedores estão entrando neste nicho. Algumas das principais questões sobre esse mercado foram abordadas neste trabalho a fim de esclarecer, principais engines que se encaixam nesse tipo de desenvolvedor, plataformas suportadas por elas, linguagem comuns de serem utilizadas, alguns exemplos de jogos produzidos pelas engines citadas, e meios de distribuir o jogo produzido pela engine. Como resultados, é aparente que seja um meio de negócio que tem tido uma crescente, conforme os jogos evoluem e a engine cada vez mais necessária e a facilidade em termos de ter acesso a IDE para desenvolver um jogo, é notório que vários desenvolvedores optam por esse meio, seja ela se juntar a uma grande empresa ou mesmo ser um desenvolvedor Indie.

Palavras-Chave: Desenvolvedor Indie; Engine; Godot; Unity 3d; Unreal engine.

ABSTRACT

Indie game developers are usually the ones who have an idea for a game and are responsible for developing it on their own, without relying on a team or large studios that are often used by many larger companies. It's a challenge, which can be difficult, but it's something a lot of developers are getting into in this niche. In this work, some of the main questions about this market were addressed in order to clarify the main engines that adapt to this type of developer, the platforms they support, the common languages to use, some examples of games produced by the engines. aforementioned. , and means of distributing the game produced by the engine. As a result, it is evident that it is a business medium that has been growing, as games evolve and the engine is more and more necessary and the ease in terms of having access to the IDE to develop a game, it is notorious that several developers They choose this medium, whether it's joining a large company or even being a freelance developer.

Keywords: Engine; Godot; Indie Developer; Unity 3d; Unreal engine.

SUMÁRIO

1 INTRODUÇÃO	8
2 OBJETIVOS	9
2.1 Objetivos Gerais	9
2.2 Objetivos específicos	9
3 REFERENCIAL TEÓRICO	9
3.1 Conceitos e Características de uma Game Engine	9
3.2 Estrutura de uma engine	10
3.2.1 Engine Core:	10
3.2.2 Engine SDK:	10
3.2.3 Level Editors:	10
3.2.4 Conversores/Exportadores:	11
3.2.5 Builders:	11
3.2.6 Linguagens Script:	11
3.3 Godot	12
3.4 Unity	12
3.5 Unreal engine	13
4 PROCEDIMENTOS METODOLÓGICOS	13
5. RESULTADOS E DISCUSSÕES	14
5.1 Godot	14
5.1.1 Plataformas suportadas pela engine	15
5.1.2 Linguagem utilizada	15
5.1.3 Interface do editor Godot	15
5.1.4 Nós	18
5.1.5 Cenas	18
5.1.6 Linguagem principal da Engine	19
5.2 Unity	21
5.2.1 Plataformas suportadas pela engine	21
5.2.2 Linguagem utilizada	21
5.2.3 Interface do editor Unity	22
5.2.4 Scripts e IDEs compatíveis com a Unity	23
5.3 Unreal Engine	24
5.3.1 Plataformas suportadas pela engine	25
5.3.2 Linguagem utilizada	25
5.3.3 Interface da Unreal engine	25
5.3.4 Linguagem padrão da engine	26
6 JOGOS FEITOS PELA ENGINE CITADAS	27

6.1 Godot	27
6.1.1 Kingdoms Of The Dump	27
6.1.2 Gravity Ace	29
6.1.3 The Garden Path	31
6.2 Unity	33
6.2.1 Cuphead	33
6.2.2 Pokémon GO	34
6.2.3 Cities: Skylines	35
6.3 Unreal Engine	37
6.3.1 Fortnite	37
6.3.2 Batman: Arkham City	38
6.3.3 Borderlands 2	40
7 MEIOS DE LANÇAR O JOGO	41
7.1 Godot	41
7.2 Unity	43
7.3 Unreal Engine	44
8 CONSIDERAÇÕES FINAIS	46
REFERÊNCIAS	47

1 INTRODUÇÃO

Em 2022 houve um significativo aumento de dois pontos percentuais em relação ao ano anterior em brasileiros que jogam algum tipo de jogo eletrônico. Segundo levantamento da Pesquisa Game Brasil (PGB), o Brasil atingiu a marca de 5074,5% de pessoas, isso significa que a cada quatro brasileiros três jogam em algum tipo de equipamento eletrônico que seja compatível com jogos.

Com a crescente popularização dos jogos, os celulares são alguns dos grandes influenciadores do aumento da demanda de jogos, segundo a Pesquisa Game Brasil (PGB), responsável pelo levantamento, cerca de 48,3% usam o smartphone para jogar. O sócio da GoGamers, Carlos Silva, explica que, "Computadores e consoles para jogar apresentam custos elevados para os padrões da maioria dos brasileiros. No entanto, os smartphones democratizam este hábito, uma vez que existem em maior quantidade do que a própria população do Brasil".

A partir do crescente aumento do interesse de pessoas por jogos, também aquece ainda mais o mercado de jogos. Segundo um estudo da Accenture (NYSE: ACN) considera que o setor de jogos já supera a soma dos mercados de filmes e música, com valor superior a US\$ 300 bilhões. Segundo Seth Schuler, diretor de Software e Platforms da Accenture, "O surgimento de novas plataformas de games e as mudanças demográficas estão forçando as empresas do setor a se tornarem plataformas orientadas para a experiência e não mais centradas em produto".

Com isso torna-se ainda mais famosas as engines, que basicamente é a alma do jogo, responsável pela criação de cenários gráficos, mecânicas dos personagens, trilha sonora etc. Independente da plataforma escolhida para rodar o jogo, existe alguma engine que já facilita o processo de criação do jogo. Existem inúmeras engines de jogos, tem aquelas que só é utilizada por determinada empresa, pois a mesma a criou para determinado jogo, é existe as mais consolidadas no mercado, seja por empresas desenvolvedoras de jogos ou programadores Indies. Algumas das principais são Unreal Engine e Unity.

Porém com tantas engines fica uma difícil decisão para o desenvolvedor que pretende iniciar com alguma engine, alguns dos fatores escolhido possa ser o quanto a engine é usada, e quanto a sua comunidade é ativa, ou até mesmo se

precisar pagar algo a engine pretendida, outros podem querer encontrar uma engine de código aberto, sendo um dos seus pontos mais atrativos a comunidade.

Existem dois tipos de desenvolvedores mais comuns que tem esta duvida, de por onde ir, seriam, o desenvolvedor que pretende iniciar uma carreira na programação de jogos, a fim de trabalhar para alguma empresa já atuante no mercado, é a outra possibilidade é ser um desenvolvedor Indie, tendo como proposta de desenvolver algo de forma independente, sem inicialmente ter contrato com empresa ou studio, este tem como propósito final colocar em prática alguma ideia de um jogo que tenha em mente, buscando criar e lançar o jogo por conta própria.

Este trabalho aborda este segundo tipo de desenvolvedor, tratando três de algumas das principais engines mais utilizadas e adequadas para este tipo de desenvolvedor, analisando as plataformas que a engine suporta, Linguagens de programação compatíveis e nativas do motor de jogos e explorando a interface do motor da engine.

2 OBJETIVOS

2.1 Objetivos Gerais

Avaliar as principais game engines utilizadas para a criação de jogos indie, mostrar como é o ambiente de um motor de jogos, algumas características é programação.

2.2 Objetivos específicos

- ✓ Identificar as principais engines para desenvolvedor indie
- ✓ Comentar quais plataformas suportadas pela Engine analisada, Linguagem utilizada, editor da Engine e linguagem de programação principal do motor de jogos
- ✓ Demonstrar jogos que foram produzidos por essas Engines.
- ✓ Detalhar meios de lançar um jogo com a proposta indie.

3 REFERENCIAL TEÓRICO

3.1 Conceitos e Características de uma Game Engine

Este capítulo vem discorrer sobre os conceitos e as principais características de uma engine de jogos, Trata-se de conceitos fundamentais de engines, e características importantes das engines citadas, é diferenças que tornam algumas engines úteis a um determinado motivo e/ou contexto.

Uma engine de jogos e/ou conjunto de bibliotecas que são responsáveis pela construção e desenvolvimento dos jogos, que contém um conjunto de cenários, IA, física, sons, texturas, efeitos 3D, basicamente tudo que o jogo precisa para funcionar. Como uma game engine possui já um ambiente que disponibiliza o desenvolvimento de jogos, de forma mais simplificada e usando abstração no desenvolvimento de jogos, fica evidente a sua importância para a criação dos mesmos, um framework tão poderoso feito exclusivamente para a construção do jogo, garantindo, dependendo da engine que o jogo funcione em diferente plataformas, com o mesmo código, o desenvolvedor consegue um ganho de tempo considerável usando uma engine, pois o mesmo não tem a preocupação de ter criar algum recurso fundamental que tem na engine já por padrão.

Dentro da área de jogos, um engine se encarregará por lidar com o hardware gráfico, irá controlar os modelos para serem renderizados, tratará das entradas de dados do jogador, tratará de todo o processamento de baixo nível e outras coisas que o desenvolvedor de jogos normalmente não deseja fazer ou não tem tempo para se preocupar. (CLUA; BITTENCOURT,2005).

De acordo com Battaiola (2000) é definida como o motor do jogo, ou seja, é o mecanismo que controla a reação do game em função das ações do jogador.

3.2 Estrutura de uma engine

Ainda segundo (CLUA; BITTENCOURT,2005), Um jogo 3D é um software especial, pois contém elementos muito variados: módulos de Computação Gráfica, Inteligência Artificial, Redes de Computadores, Multimídia, entre outros. Ele complementa que os componentes mais comuns de uma engine são:

3.2.1 Engine Core:

Consiste no “coração do engine”. Este será um programa que executará a aplicação do jogo, manipular a fase e os objetos, renderizar as cenas, etc. Fazendo-se uma grossa analogia, pode-se dizer que o engine core é o sistema operacional do jogo.

3.2.2 Engine SDK:

É o código fonte do engine core. Através dele pode-se alterar o funcionamento do engine. Normalmente este componente é o mais protegido e para consegui-lo, no caso de engines comerciais, será necessário comprar o pacote que a empresa oferece.

3.2.3 Level Editors:

Através deste componente será possível unificar modelagens feitas em diversos programas, associá-los à programação, inserir códigos em scripts, etc. Em muitos casos, dentro destes editores é possível também criar modelos 3D.

3.2.4 Conversores/Exportadores:

Os resources serão normalmente feitos em diversos softwares comerciais, mais ainda: numa equipe de desenvolvimento grande cada artista poderá criar os elementos no programa de sua preferência. Assim sendo, os engines deverão fornecer instrumentos para importar estes modelos para o formato específico do engine. Estes conversores poderão ser plug-ins instalados nos programas de modelagem 3D ou podem estar incluídos no level editor;

3.2.5 Builders:

Como será discutido algumas operações pré-processamentos sobre os objetos (tais como o BSP, lightmaps, portais, PVS, etc.) precisam ser feitas. Desta maneira, um engine fornecerá as ferramentas para realizar estes pré-processamentos.

3.2.6 Linguagens Script:

Grande parte do desenvolvimento da lógica do jogo e da Inteligência Artificial dos elementos dinâmicos será implementada sobre scripts e não diretamente sobre o engine core. Assim, cada engine possuirá sua linguagem de programação script, sendo comum usar linguagens comuns, tais como JavaScript, Python e LUA.

Utilizando uma engine, o desenvolvedor não precisará reinventar a roda, todos os recursos básicos presentes na maioria dos jogos, estará nativamente no motor de jogos.

Com a evolução dos Motores de jogos o desenvolvedor já não precisa se preocupar em como funcionará o mecanismo de mira, como programar o comportamento dos inimigos (ex. para um jogo de FPS). Mas sim, montar a história com as ferramentas disponíveis, pois hoje as Game Engines já possuem inteligência artificial, ferramentas pré-programadas, mapas prontos e algumas, inclusive, possuem templates prontos (ex. Engine M.U.G.E.N que é específica para jogos de lutas 2D e já vem com nove modos de jogos pré-programados para serem editados). (SCHERER; BATISTA; MENDES, 2020).

Uma engine contém dentro dela uma série de recursos, a fim de facilitar no desenvolvimento do jogo.

A implementação do motor pode envolver conceitos presentes em diversas áreas da computação, tais como, computação gráfica, inteligência artificial, redes de computadores, engenharia de software, etc.(BATTAIOLA, A L. ET AL, 2002).

E ainda é possível dependendo da engine adicionar mais recursos, conforme a necessidade do projeto.

Cada engine possuirá conversores e exportadores, que permitirão que sejam utilizados outros programas para programar seu SDK ou para modelar os objetos 3D. É fundamental que o desenvolvedor analise quais as ferramentas com as quais se pode produzir os para o engine em questão (se a equipe de modeladores conhece apenas o MAYA, é conveniente que o engine a ser adotado suporte o MAYA para os modelos 3D).(CLUA; BITTENCOURT,2005).

Algumas das características mais importantes das engines que programadores Indies devem levar em consideração antes de escolher o motor de jogos, serão, no quesito de custo para uso da engine, plataformas suportadas, tanto para importação como para exportação, conhecer da detentora da engine, se a mesma recebe atualizações necessárias, funcionalidade se é compatível com a plataforma desejada para distribuir o jogo. Algumas das engine que possui a capacidade de distribuir o jogo em várias plataformas, e possui um boa política para o uso da engine, e constantes atualizações na engine e recursos da mesma estão: A Godot, Unity 3D e a Unreal Engine.

3.3 Godot

A Godot utiliza uma arquitetura de nós aninhados, que é baseada na orientação a objetos com foco em composição. Cada entidade na engine é representada por uma cena (Scene), que é composta de nós (Nodes), os quais por sua vez podem ser compostos de outros nós, cada qual tendo seu comportamento definido através de scripts/código.

Uma Scene pode ser pensada como um arquivo que guarda uma fase, uma área específica ou um menu do jogo. Dividir seu jogo em muitas scenes pode ajudar na prototipagem, uma vez que você pode testar suas scenes individualmente e de forma rápida.(SOUSA;2022).

3.4 Unity

A Unity é uma engine capaz de fornecer diversos recursos ao desenvolvedor de jogos, a mesma possui vários recursos gráficos, de rede entre outros, por ser uma engine com bastante recursos, pode ser utilizada por várias aplicações que tem o intuito de utilizar tais recursos.

O Unity pode oferecer uma valiosa contribuição no desenvolvimento de quaisquer aplicações que utilizem recursos como: gráficos 3D, simulações físicas, networking, suporte multimídia e capacidade multiplataforma.(SOUSA;2022).

Porém Desenvolvedores Indie tendem a dar preferência a engine que sejam leves, engine que não exigem muito do computador, ou engine sem custos na sua utilização, a Unity possui uma característica de ser leve, mas tem um custo para a distribuição do jogo feito, conforme foi discutido em resultados e discussões.

A Unity 3D apresenta a melhor curva de aprendizado e é mais leve para computadores menos robustos, embora com escalabilidade menor (não suporta simulações físicas mais realistas, a taxas interativas) e as mensalidades desestimulam desenvolvedores com fins não lucrativos.(Macedo;D V. ET AL, 2015).

3.5 Unreal engine

Assim como na Unity 3d, os recursos presentes na Unreal, poderá ser utilizado em outros projetos, não se limitando a apenas desenvolvimento de jogos.

A abrangência de ferramentas e funcionalidades existentes não as limita somente ao desenvolvimento de jogos, sendo amplamente usadas para a síntese de uma vasta gama de aplicações gráficas. Basicamente, um Game Engine funciona como uma camada entre a aplicação final e os diferentes motores e software sagregados, gerando transparência ao desenvolvedor e

fornecendo mecanismos eficientes de integração entre os seus componentes.(Macedo;D V. ET AL, 2015).

4 PROCEDIMENTOS METODOLÓGICOS

Por se tratar de uma pesquisa com o objetivo de reunir maiores informações a respeito de um tema, Munaretto, Luiz Corrêa e Cunha (2013) classificam este tipo de pesquisa como uma pesquisa exploratória.

Grande parte da pesquisa foi feita com base na documentação das engines citadas ao discorrer dos resultados, utilizando a mesma para o levantamento de definições,funcionalidades, sobre a empresa detentora da engine, dados e informações pertinentes ao tema e imagens necessárias para melhor entendimento.

De acordo com Gil (2019), acrescenta ainda que uma pesquisa exploratória tem por objetivo realizar uma pesquisa cujo fim seja modificar, esclarecer ou desenvolver melhor uma ideia ou tema levantando hipóteses deixadas para estudos futuros. Foi feito um estudo com base na documentação de cada Engine.

5. RESULTADOS E DISCUSSÕES

Segundo a empresa de criação de jogos Danqui Code, Como desenvolvedor indie, provavelmente você não terá acesso a muitos engines games que grandes estúdios de AAA possuem (grandes estúdios que trabalham com grandes editoras como EA, Ubisoft e Sony).

5.1 Godot

De acordo com a documentação da godot, a mesma tem como suas principais características ser leve, e totalmente gratuita na sua utilização e distribuição. Segundo a documentação da própria Godot:

Godot Engine é um mecanismo de jogo multiplataforma repleto de recursos para criar jogos 2D e 3D a partir de uma interface unificada. Ele fornece um conjunto abrangente de ferramentas comuns, para que os usuários possam se concentrar em criar jogos sem precisar reinventar a roda. Os jogos podem ser exportados com um clique para várias plataformas, incluindo as principais plataformas de desktop (Linux, macOS, Windows), bem como plataformas móveis (Android, iOS) e baseadas na web (HTML5).

Sendo uma engine de código aberto, e gratuita se torna um grande atrativo para desenvolvedores que não querem ter que ceder parte do ganho da venda do jogo para a detentora da engine.

Segundo a Godot. Godot é totalmente gratuito e de código aberto sob a licença permissiva do MIT. Sem amarras, sem royalties, nada. Os jogos dos usuários são deles, até a última linha do código do motor.

Outra característica segundo a Godot muito forte dessa engine é a comunidade onde a mesma recebe ajuda de colaboradores ao redor do mundo, através da sua comunidade, sendo possível atualizações, correção de bugs, adição de funcionalidades, etc. Documentação da Godot: O desenvolvimento da Godot é totalmente independente e orientado pela comunidade, capacitando os usuários a ajudar a moldar seu mecanismo para atender às suas expectativas. É suportado pela Software Freedom Conservancy sem fins lucrativos.

5.1.1 Plataformas suportadas pela engine

A Godot é capaz de suportar diversas plataformas sendo elas: Windows a plataforma mais popular entre todas, macOS para usuários de Mac, X11 sendo (Linux, *BSD), Web editor próprio da Godot, atualmente em estágio preliminar, Android (experimental). Tem capacidade para exportar o jogo para as plataformas: Windows (and UWP), macOS, X11 (Linux, *BSD), Android, iOS, Web. Segundo a documentação oficial da Godot: Ambos os binários de 32 e 64 bits são suportados onde faz sentido, sendo 64 o padrão.

5.1.2 Linguagem utilizada

De acordo com a documentação. As linguagens oficialmente suportadas para Godot são GDScript, Visual Scripting, C# e C++.

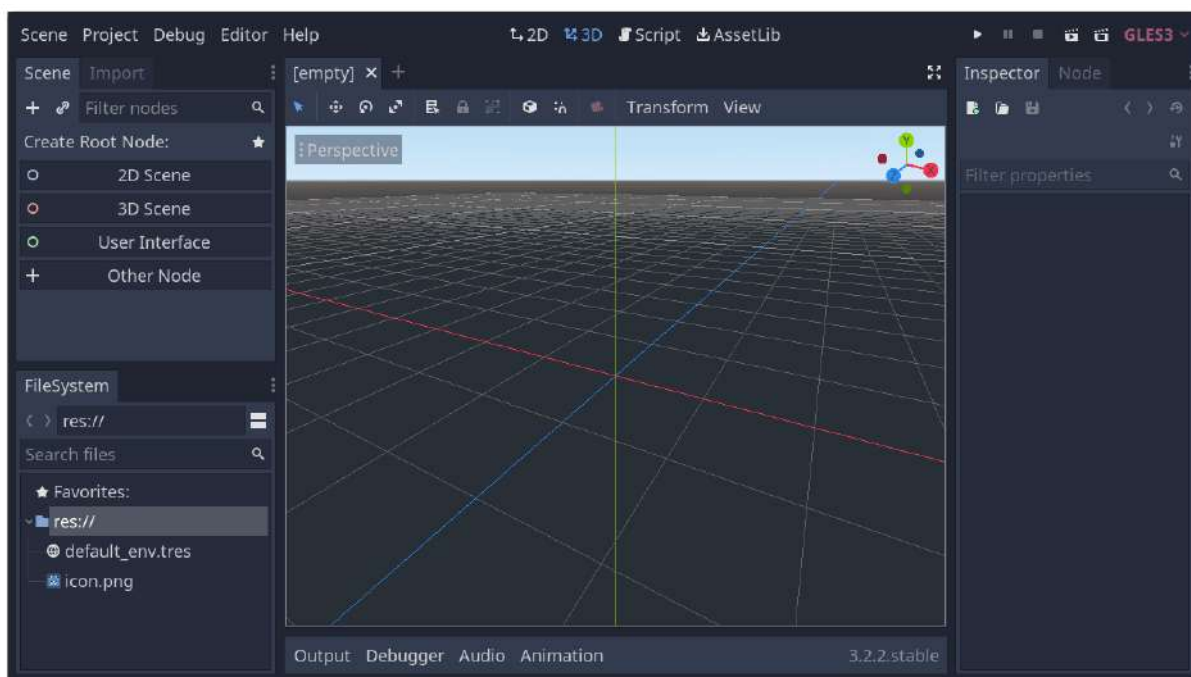
A mesma recomenda que seja usada por padrão a GDScript, principalmente se for programador de primeira viagem, pois a mesma é nativa da própria Godot, evitando dificuldades no primeiro contato com a engine. GDScript se torna uma linguagem amigável, a fim de trazer uma rápida solução na hora de desenvolver os jogos.

Existem várias razões para usar GDScript - especialmente quando você está prototipando, em fases alfa/beta do seu projeto, ou não está criando o próximo título AAA - mas a razão mais importante é a redução geral da complexidade .

5.1.3 Interface do editor Godot

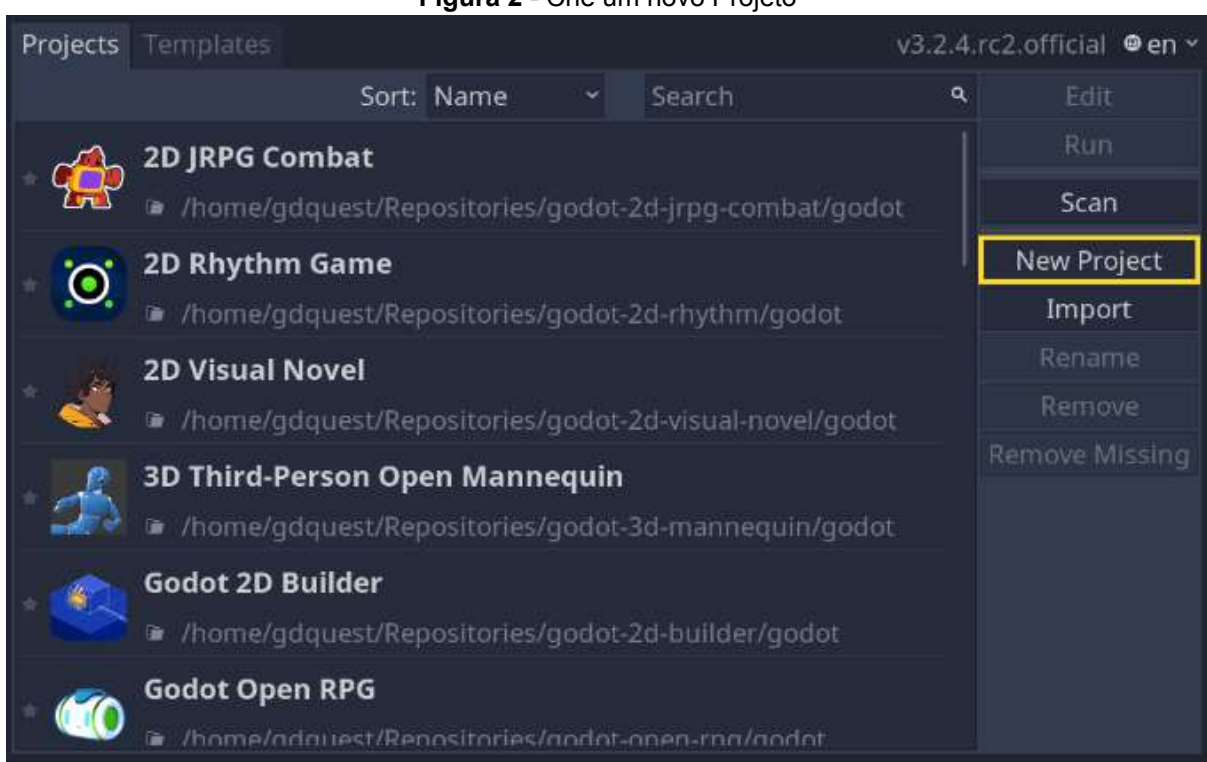
A engine godot possui um editor de cena, poderoso capaz de proporcionar uma boa interface de usuário, conter diversas cenas, opções de configurações de projetos, depuração, mudança de layout, edição de cenas, gerenciamento na árvore de arquivos do projeto, reproduzir a cena, área específica para Script, importação de bibliotecas, entre outras características.

Figura 1 - Interface Padrão da Godot



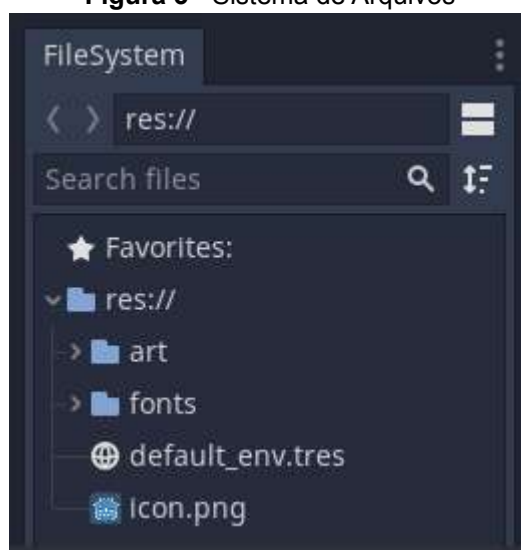
https://docs.godotengine.org/en/stable/getting_started/introduction/first_look_at_the_editor.html

Para começar um novo projeto basta iniciar a godot engine, baixada diretamente no site oficial da Engine. <https://godotengine.org/download>. É então criar o seu projeto.

Figura 2 - Crie um novo Projeto

Fonte: https://docs.godotengine.org/en/stable/getting_started/first_2d_game/01.project_setup.html

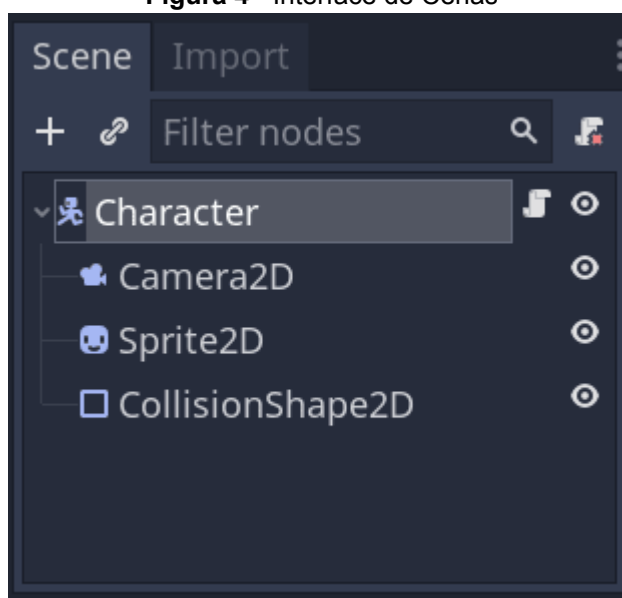
Com uma simples janela de árvores de arquivos do projeto é possível ter uma noção do corpo da aplicação e organização do mesmo.

Figura 3 - Sistema de Arquivos

Fonte: https://docs.godotengine.org/en/stable/getting_started/first_2d_game/01.project_setup.html

5.1.4 Nós

A Engine funciona à base de um conceito em que um jogo Godot é uma árvore de cenas e que cada cena é uma árvore de nós. Segundo a Documentação, os nós são os blocos de construção fundamentais do seu jogo . Eles são como os ingredientes de uma receita. Existem dezenas de tipos que podem exibir uma imagem, reproduzir um som, representar uma câmera e muito mais.

Figura 4 - Interface de Cenas

https://docs.godotengine.org/en/stable/getting_started/first_2d_game/02.player_scene.html

5.1.5 Cenas

A partir do momento em que se tem nós, hierarquicamente organizados, formando uma árvore, é chamado de construção de cenas. Segundo a documentação, Depois de salvas, as cenas funcionam como novos tipos de nó no editor, onde você pode adicioná-las como filhas de um nó existente. É completa que, as cenas permitem que você estruture o código do seu jogo como quiser. Você pode compor nós para criar tipos de nós personalizados e complexos, como um personagem de jogo que corre e pula, uma barra de vida, um baú com o qual você pode interagir e muito mais.

5.1.6 Linguagem principal da Engine

Segundo a Documentação da Godot em relação a Script, Scripts são arquivos contendo código que você anexa a um nó para estender sua funcionalidade.

GDScript é a opção nativa do Engine, recomendada pela mesma. Na Godot um nó tem anexado consigo um script, caso seja criado, sendo capaz de complementar o comportamento do nó. Um exemplo seria todas vez em que um personagem morrer a câmera fique borrada de sangue. Anexando este script ao nó, criaria um recurso que não está por padrão na engine. Abaixo um exemplo de Script, que define a animação de movimentação do Personagem, quando presionado as teclas direita e esquerda.

```
extends KinematicBody2D

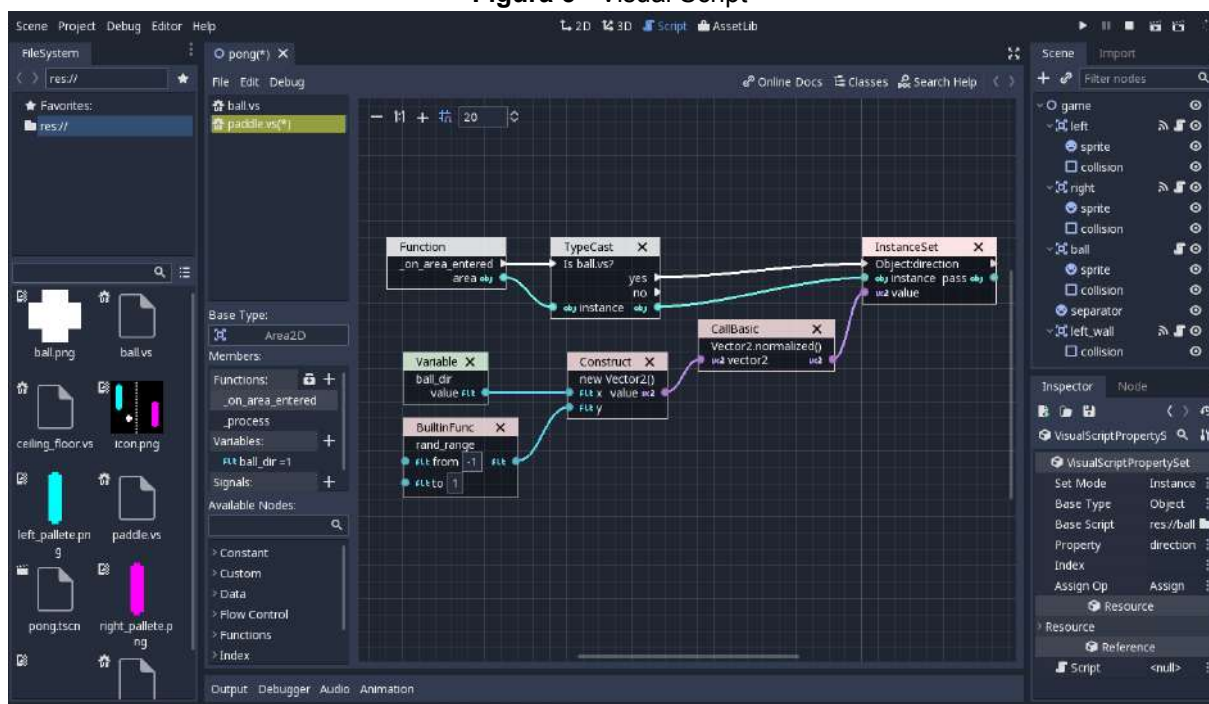
onready var _animated_sprite = $AnimatedSprite

func _process(_delta):
    if Input.is_action_pressed("ui_right"):
        _animated_sprite.play("run")
    else:
        _animated_sprite.stop()
```

Havia outra opção na Godot, a linguagem de programação, chamada Visual Scripting que permitia, o usuário visualmente conectar blocos graficamente, como alternativa para designer de jogos e artistas que não possuem conhecimento prévio

de programação. Porém será descontinuada, segundo a Godot, Godot 4.0 removerá totalmente o VisualScript do núcleo.

Figura 5 - Visual Script



https://docs.godotengine.org/en/stable/tutorials/scripting/visual_script/getting_started.html?highlight=visual%20script

Como resultado, a criação de novos projetos usando scripts visuais no Godot não é recomendada. Versões futuras do Godot 4.x podem ter o VisualScript reimplementado como uma extensão.

5.2 Unity

Unity também conhecido como Unity3D ou UnityEngine, é um motor de jogos criado pela Unity Technologies, em 8 de junho de 2005, após ser anunciado na Apple Worldwide Developers Conference em 2005.

Tendo como características, ser versátil, contribuindo para desenvolvedores iniciantes e experientes, atendo diversos tipos de gráficos e mecânicas, multiplataformas.

Segundo a Documentação da Unity.

Unity capacita designers de jogos a criar jogos. O que é realmente especial no Unity é que você não precisa de anos de experiência com código ou um diploma em arte para criar jogos divertidos. Há um punhado de conceitos

básicos de fluxo de trabalho necessários para aprender Unity. Uma vez entendido, você se verá fazendo jogos em pouco tempo.

Segundo a própria companhia da Unity, em 2021 a Unity teve 5B de downloads por mês de aplicativos desenvolvidos com Unity, 72 % dos 1.000 principais jogos para dispositivos móveis foram feitos com Unity, 50% dos jogos para dispositivos móveis, PC e console foram feitos com Unity, 3.9 de usuários ativos mensais que consumiram conteúdo criado ou operado com soluções da Unity, 20+ diferentes plataformas executam as criações do Unity, 190+ países e territórios têm criadores do Unity.

A ferramenta contém três planos são eles Individual, que consistem em estudante e pessoal, sendo o de estudante totalmente gratuito, mas com algumas limitações na plataforma, o outro contém uma limite de elegibilidade para seu uso, segundo a plataforma, o Unity Personal é para indivíduos, amadores e pequenas empresas com menos de US\$ 100 mil em receita ou recursos arrecadados nos 12 meses anteriores. Tem ainda planos de Equipe é empresa, ambos pagos. Também contém recursos mais específicos para este tipo de nicho, seja uma equipe maior ou uma grande empresa ou estúdio.

5.2.1 Plataformas suportadas pela engine

Segundo a Documentação da Unity ela oferece suporte à maioria das principais plataformas de desktop, web e móveis da seguinte forma: Plataformas independentes: macOS , Windows e Linux, tvOS, iOS, Android e WebGL.

Consegue exportar para:playstation, xbox, nintendo switch,google stadia,apple app store, google play, desenvolvimento de conteúdo de ar e vr, windows, mac e linux.

5.2.2 Linguagem utilizada

A linguagem adotada pela engine é o C#. Segundo a documentação:

O Unity C# Job System permite escrever código multithread simples e seguro que interage com o Unity Engine para melhorar o desempenho do jogo. Você pode usar o C# Job System com o Unity Entity Component System (ECS), que é uma arquitetura que simplifica a criação de código de máquina eficiente para todas as plataformas.

5.2.3 Interface do editor Unity

A Unity possui um editor completo, capaz de criar um ambiente interativo, para trabalho conjunto, isso desde de animação, edição de cenas do jogo, construção de mecânicas entre outras funcionalidades, permite ainda o adição de outras ferramentas para a extensão de recurso dentro da plataforma, como exemplo o Blender, ferramenta de modelagem 3D.

Segundo a Documentação oficial da Unity, algumas das posições mais comuns usadas são:

Figura 6 - Interface Padrão da Unity



Fonte: <https://docs.unity3d.com/Manual/UsingTheEditor.html>

(A) A barra de ferramentas fornece acesso aos recursos de trabalho mais essenciais. À esquerda, contém as ferramentas básicas para manipular a visualização Scene e os GameObjects dentro dela. No centro estão os controles de reprodução, pausa e passo. Os botões à direita dão acesso ao Unity Collaborate, Unity Cloud Services e sua conta Unity, seguidos por um menu de visibilidade da camada e, finalmente, o menu de layout do Editor (que fornece alguns layouts alternativos para as janelas do Editor e permite que você salve suas próprios layouts personalizados).

(B) A janela Hierarchy é uma representação de texto hierárquico de cada GameObject na cena. Cada item na Cena tem uma entrada na hierarquia,

então as duas janelas estão inerentemente ligadas. A hierarquia revela a estrutura de como os GameObjects se conectam uns aos outros.

(C) A visualização do jogo simula a aparência do seu jogo renderizado final por meio de suas câmeras de cena. Ao clicar no botão Play, a simulação começa.

(D) A visualização da cena permite que você navegue visualmente e edite sua cena. A visualização da cena pode mostrar uma perspectiva 3D ou 2D, dependendo do tipo de projeto em que você está trabalhando.

(E) A Janela do Inspetor permite visualizar e editar todas as propriedades do GameObject atualmente selecionado. Como diferentes tipos de GameObjects têm diferentes conjuntos de propriedades, o layout e o conteúdo da janela do Inspetor mudam sempre que você seleciona um GameObject diferente.

(F) A janela Projeto exibe sua biblioteca de ativos que estão disponíveis para uso em seu projeto. Quando você importa ativos para seu projeto, eles aparecem aqui.

No mundo de desenvolvimento da engine unity, é possível desenvolver praticamente qualquer parte do jogo, ou seja criar os seus próprios componentes com programação pode oferecer algo a mais do que somente os componentes integrados da Unity, que embora sejam versáteis, a capacidade de criar algum componente específico para o seu jogo, o torna uma ferramenta mais poderosa, alterando o comportamento de GameObjects, que é basicamente todos os objetos em um cena, com isso permite alterar características dela, como rotação, audio, cores etc, acrescentando seus próprios eventos no jogo e outras propriedades.

5.2.4 Scripts e IDEs compatíveis com a Unity

Os Scripts podem ser criados direto na engine. Só é preciso acessar o canto superior esquerdo do painel Projeto ou pode selecionar: Ativos > Criar > Script C# no menu principal.

Figura 7 - Janela de Criação de Script

<https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>

A Unity suporta as seguintes IDEs para código C#: Visual Studio (com o plug-in Ferramentas do Visual Studio para Unity), Visual Studio para Mac, Piloto JetBrains, Visual Studio Code (experimental). Porém algumas linguagens .NET podem ser usadas, desde que compilem um DLL compatível.

5.3 Unreal Engine

Unreal Engine é o motor de jogos mais utilizado para jogos 3d, criado pela Epic Games, foi usado pela primeira vez em 1998, na época foi desenvolvido um jogo FPS(f first-person shooter), ou tiro em primeira pessoa. Pela fama, se tornou base para muitos jogos. Seu núcleo é escrito em C++. Segundo a empresa. A Unreal Engine é a ferramenta de criação 3D em tempo real mais avançada do mundo. Com ela, você dará vida a experiências incríveis.

A Unreal sem dúvidas é uma das engines mais populares, sendo uma engine que sempre está envolvida em grandes lançamentos, desde o início da mesma, tem sido um das pioneiras do mercado, isso a torna uma das favoritas para uso, sendo adotado por muitos estúdios de desenvolvimento, fortalecendo ainda mais ao longo do tempo a variedade de desenvolvedores que utilizam a engine. Em entrevista ao portal online de notícias e mídia, the verge, o vice-presidente de engenharia da Epic games, Nick Penwarden afirmou que, Mais da metade de todos os jogos anunciados para consoles de última geração estão sendo desenvolvidos usando o conjunto de ferramentas de desenvolvedor Unreal Engine da Epic Games.

Atualmente o motor está na sua versão 5.1, trazendo integridade visual e desempenho, na criação de jogos de 60 fps para consoles de próxima geração e PCs compatíveis, construções de mundos maiores com World Partition e colaboração para facilitar o desenvolvimento do jogo, Produção virtual e efeitos visuais na câmera, é melhorias de renderização.

Segundo a própria documentação da Unreal Engine. O 5.1 está se baseando na estrutura do 5.0, fornecendo atualizações e melhorias para facilitar ainda mais a criação de conteúdo e experiências 3D de próxima geração em tempo real.

Também é possível modificar o motor de jogos de acordo com a necessidade do desenvolvedor, para isso, basta ser parceiro da Epic Games, para ter acesso.

Para ter acesso a plataforma, basta baixar, seu download é gratuito e com todos os recursos disponíveis, segundo a plataforma. A Unreal Engine é gratuita para criação de conteúdo linear como filmes, e para projetos personalizados e internos. Para desenvolver jogos, ela também é gratuita em muitos casos. Um royalty de 5% é cobrado apenas se, e quando seu jogo faturar mais de US\$1 milhão.

Tem ainda uma licença chamada programa empresarial, que segundo os mesmos, Para profissionais fora da área de jogos que buscam suporte premium, treinamento particular e/ou termos de licenciamento personalizados, sendo um total de us\$ 1.500 por lugar / ano.

E também o plano de licença personalizada, ainda segundo eles. Para profissionais e estúdios de desenvolvimento de jogos que buscam suporte premium, treinamento privado e/ou termos de licenciamento personalizados.]

5.3.1 Plataformas suportadas pela engine

É capaz de suportar múltiplas plataformas, tais como, WINDOWS, LINUX, MAC OS E MAC OS X. É consoles PLAYSTATION do 2 ao 5, DREAMCAST. GAMECUBE, NINTENDO SWITCH, XBOX, XBOX 360, XBOX ONE E XBOX SERIES X, STADIA, WII.

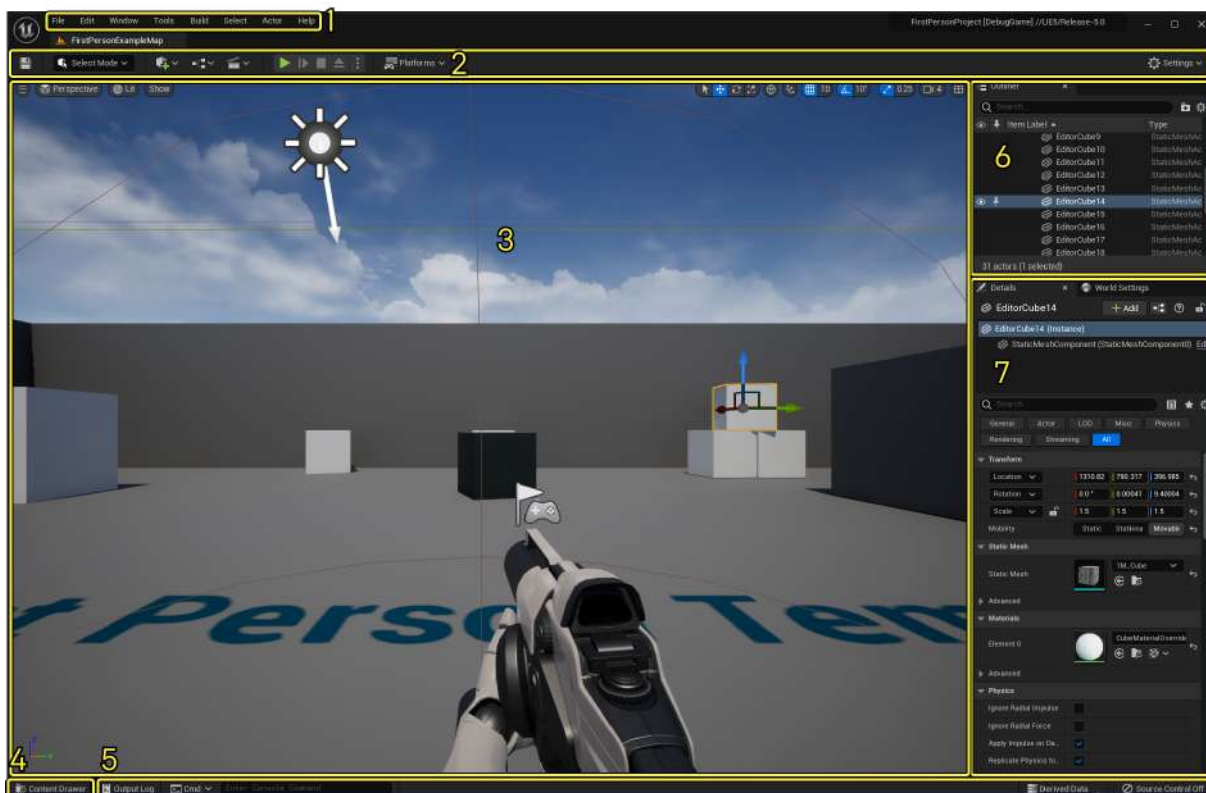
5.3.2 Linguagem utilizada

Seu núcleo é escrito em C++, possibilitando a portabilidade. Suporta múltiplas plataformas, incluindo MICROSOFT WINDOWS, LINUX, MAC OS E MAC OS X em computadores pessoais, e os consoles DREAMCAST, GAMECUBE, NINTENDO SWITCH, PLAYSTATION 2, PLAYSTATION 3, PLAYSTATION 4, PLAYSTATION 5.

5.3.3 Interface da Unreal engine

A interface da engine é bastante completa, contém um editor de níveis. Quando você abre o Unreal Engine 5 pela primeira vez, o Editor de níveis é aberto. Você verá a seguinte janela:

Figura 8 - Interface Padrão da Unreal Engine



<https://docs.unrealengine.com/5.1/en-US/unreal-editor-interface/>

1-Barra de menu, Use esses menus para acessar funções e comandos específicos do editor.

2-Barra de ferramentas principal, Contém atalhos para algumas das ferramentas e editores mais comuns no Unreal Engine, bem como atalhos para entrar no modo Play (executar seu jogo dentro do Unreal Editor) e para implantar seu projeto em outras plataformas.

3-Janela de visualização de nível, Exibe o conteúdo do seu nível, como câmeras, atores, malhas estáticas e assim por diante.

4-Botão da gaveta de conteúdo, Abre a Gaveta de Conteúdo , da qual você pode acessar todos os Ativos do seu Projeto.

5-Barra de Ferramentas Inferior, Contém atalhos para a funcionalidade Console de Comando, Log de Saída e Dados Derivados. Também exibe o status do controle de origem.

6-Delineador, Exibe uma exibição em árvore hierárquica de todo o conteúdo em seu nível.

5.3.4 Linguagem padrão da engine

A Unreal Engine, consegue dar acesso a funções diretamente nos personagens e objetos dentro do jogo, a mesma fornece dois conjuntos de ferramentas, que permite programadores além de desenvolver, fornece um fluxo de trabalho veloz.

Segundo a documentação da Unreal Engine. Novas classes de jogabilidade, elementos de interface do usuário Slate e Canvas e funcionalidade do editor podem ser escritos com C++, e todas as alterações serão refletidas no Unreal Editor após a compilação com Visual Studio ou XCode.

6 JOGOS FEITOS PELA ENGINE CITADAS

6.1 Godot

6.1.1 Kingdoms Of The Dump

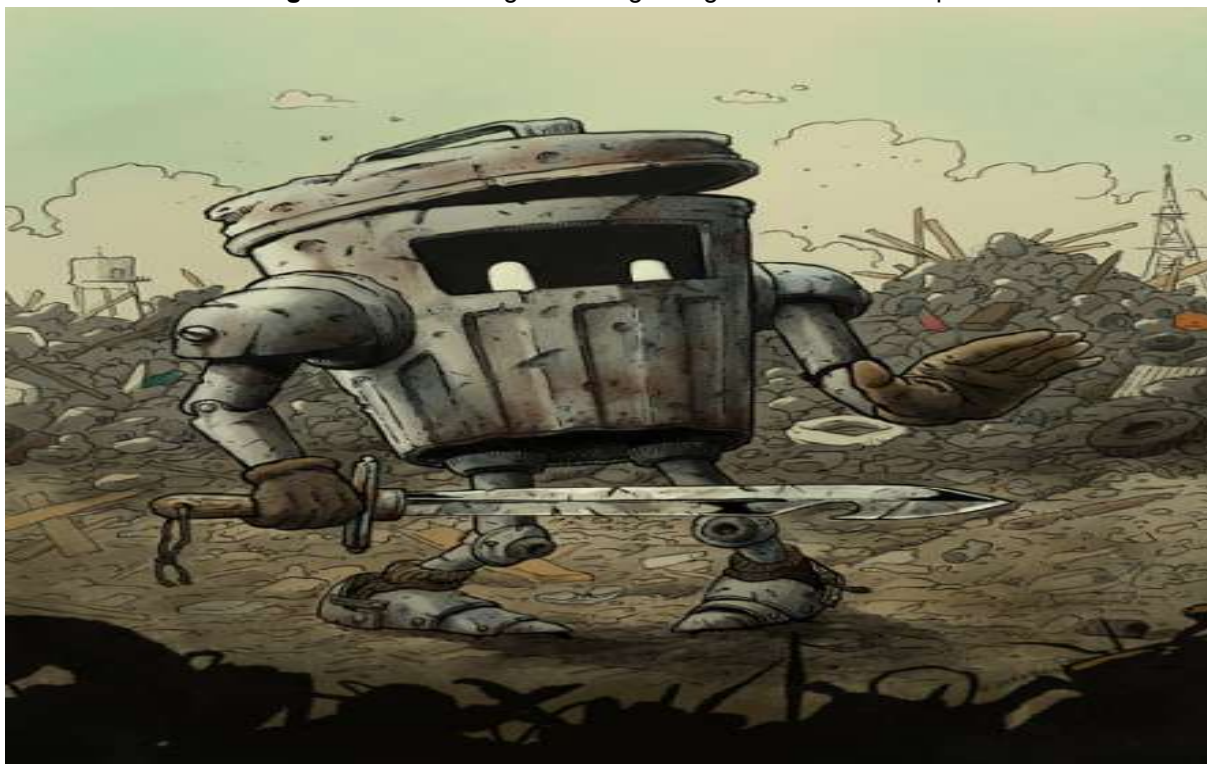
Disponível na Steam, Software esse que fornece jogos, criados por outras empresas, servindo como um software de gestão de direitos digitais, criado em 2002, pela Valve Corporation, criadora do Counter-Strike, é um jogo desenvolvido pelo estúdio Roach games do gênero RPG eletrônico, compatível com as plataformas MICROSOFT WINDOWS, LINUX, MACOS, ANDROID, STEAMOS, IOS, MAC OS CLASSIC.

Figura 9 - Jogo Kingdoms Of The Dump



Fonte: <https://kingdomsofthedump.com/>

Figura 10 - Personagem do Jogo Kingdoms Of The Dump



Fonte: <https://kingdomsofthedump.com/>

Figura 11 - Cenário do jogo Kingdoms Of The Dump



Fonte: <https://kingdomsofthedump.com/>

6.1.2 Gravity Ace

Também disponível na Steam e também na itch.io que é similar a steam, sendo um site que permite a hospedagem, deixando disponível para venda e downloads de jogos eletrônicos com a proposta Indie, foi lançado por Leaf Corcon, em março no ano de 2013. Um jogo 2D eletrônico independente, desenvolvido por John Watson, esse jogo é um perfeito exemplo de um jogo feito por um desenvolvedor Indie.

Segundo o próprio estúdio:

Gravity Ace é um shooter de gravidade multidirecional 2D. É uma carta de amor para Thrust e Gravitator e todos os jogos incríveis que eles inspiraram. Desça em cavernas escuras, voe e abra caminho pelos inimigos, recupere o núcleo do reator e escape, enquanto luta contra o abraço mortal da gravidade esmagadora.

Em uma entrevista para a Godot, em um evento de ShowCase de jogos, o desenvolvedor relata por qual motivo escolheu a Godot para seu jogo:

Acho que sem Godot eu estaria muito mais atrás de onde estou agora. Godot apenas trabalha de uma maneira que combina com a maneira como gosto de pensar e trabalhar. Eu seria muito menos produtivo, com certeza. O fluxo de trabalho 2D é muito bom. Basta soltar algumas imagens e

arrastá-las para a tela, anexar um script, escrever algumas linhas de GDScript e você terá o início de um jogo. A modificação de ativos é perfeita. E a árvore de nós é super poderosa. A reutilização de cenas em outras cenas me permite fazer muitas coisas de forma rápida e visual que seriam muito mais complicadas em outras estruturas.

Figura 12 - Jogo Gravity Ace



Fonte <https://godotengine.org/showcase/gravity-ace>

Figura 13- Cenário do jogo Gravity Ace



Fonte <https://gravityace.com/>

6.1.3 The Garden Path

Disponível na Steam, é um jogo de jardinagem, o tempo no jogo, é semelhante ao tempo real, sendo assim um tipo de simulador. Segundo a Godot que disponibilizou esse jogo na sua plataforma como Showcase:

O tempo no The Garden Path está vinculado ao tempo real, mas com as estações passando a cada semana e os anos passando a cada mês. A cada estação surgem novos eventos, clima e itens colecionáveis que podem ser transformados em chás ou negociados com um elenco de animais viajantes.

O desenvolvedor relato por qual motivo escolheu a engine Godot:

Sou um defensor do software de código aberto desde que me lembro, então Godot parecia o ajuste certo. É uma sensação boa saber que meu trabalho não está sob nenhum tipo de licenciamento restritivo, também gostei da ideia de ser um dos primeiros a adotar na época. Houve um burburinho em torno do motor, mas ele ainda precisa daqueles títulos emblemáticos que as pessoas podem apontar. E isso é realmente muito importante, porque se as pessoas perguntarem 'como isso foi feito' e a resposta for algo de código aberto, isso é uma grande conquista. Algo para eu aspirar, de qualquer maneira.

Figura 14- Jogo The Garden Path



Captura de tela de The Garden Path por carrotcake.studio

Fonte https://store.steampowered.com/app/1638500/The_Garden_Path/

Figura 15 - Cenário do jogo The Garden Path



Fonte https://store.steampowered.com/app/1638500/The_Garden_Path/

6.2 Unity

6.2.1 Cuphead

Cuphead é um jogo mundialmente famoso, inspirado em desenhos animados, como por exemplo os estúdios da Warner Bros. Foi criado pelos irmãos canadenses Chad e Jared Moldenhauer, por meio do Studio MDHR.

Figura 16 - Jogo Cuphead



Fonte <https://store.steampowered.com/app/268910/Cuphead/>

Figura 17 - Cenário do jogo Cuphead



Fonte: <https://www.xbox.com/pt-BR/games/store/cuphead/9NJRX71M5X9P>

6.2.2 Pokémon GO

Pokémon GO outro jogo feito pela Unity, bastante famoso, chegando a quebrar recorde vendas na plataforma switch segundo a própria The Pokémon Company. Possui ainda uma tecnologia de realidade aumentada para smartphones. Desenvolvido por Niantic, Inc.

Figura 18 - Jogo Pokémon GO



Fonte <https://pokemongolive.com/>

Figura 19 - Cenário e Interatividade do jogo Pokémon GO



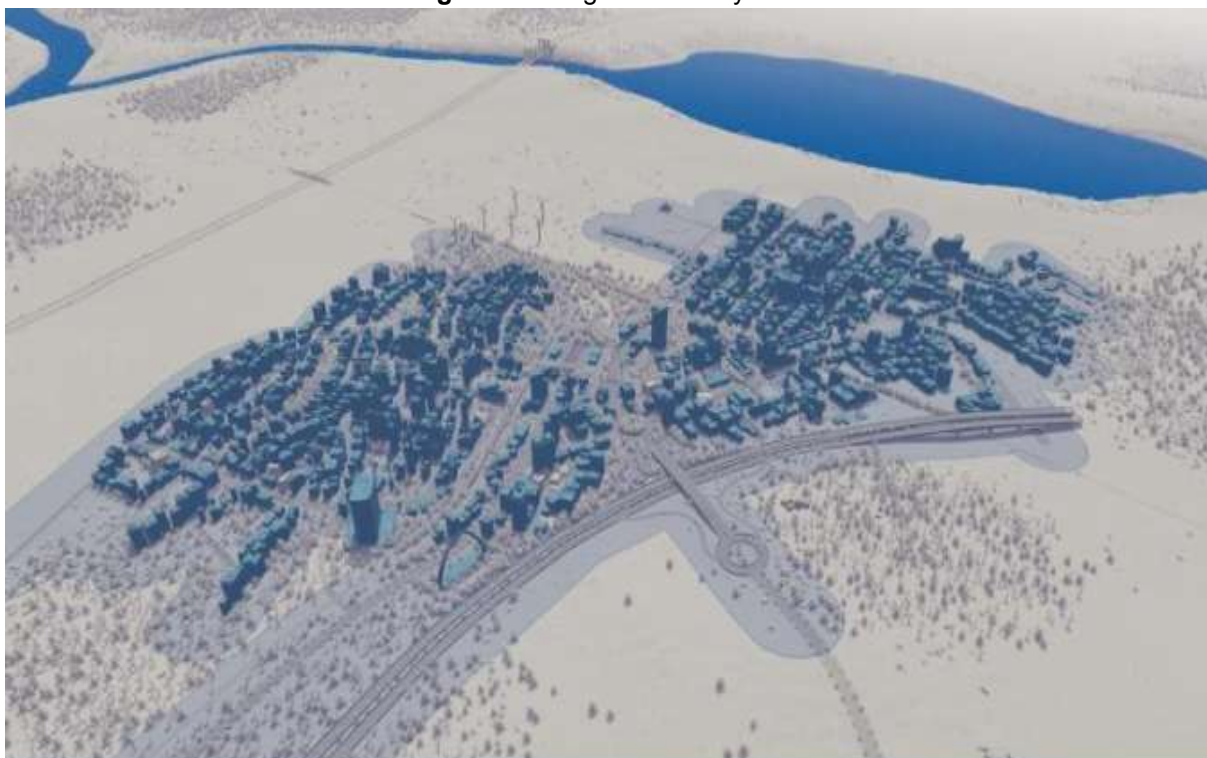
Fonte <https://pokemongolive.com/>

6.2.3 Cities: Skylines

Jogo criado pela Colossal Order e estúdio Paradox Interactive, em 10 de março de 2015, consiste em um jogo de construção de cidade. Disponível para PC, SWITCH, XBOX E PS4. Disponível para baixar na Steam e Epic games. Segundo a empresa do jogo:

Cities: Skylines é uma versão moderna da clássica simulação de cidade. O jogo apresenta novos elementos de jogo para perceber a emoção e as dificuldades de criar e manter uma cidade real enquanto expande alguns tropos bem estabelecidos da experiência de construção da cidade.

Figura 20 - Jogo Cities: Skylines



Fonte <https://www.paradoxinteractive.com/games/cities-skylines/about>

Figura 21 - Cenário do jogo Cities: Skylines



Fonte <https://www.paradoxinteractive.com/games/cities-skylines/about>

6.3 Unreal Engine

6.3.1 Fortnite

O principal jogo da Epic Games e mais famoso da Unreal, é um jogo multiplayer battle royale. Faz também parte dos Studios Warner Bros. Interactive Entertainment. Suporta as plataformas WINDOWS, MACOS, PLAYSTATION®4, PLAYSTATION®4 PRO E PLAYSTATION 4 SLIM, PLAYSTATION 5, NINTENDO SWITCH E SWITCH LITE, XBOX ONE, XBOX ONE S, XBOX ONE S ALL-DIGITAL EDITION, XBOX ONE X, E XBOX ONE SERIES X|S.

Figura 22 - Jogo Fortnite



Fonte: <https://store.epicgames.com/pt-BR/p/fornite>

Figura 23 - Personagens do jogo Fortnite



Fonte: <https://store.epicgames.com/pt-BR/p/fornite>

6.3.2 Batman: Arkham City

Figura 24 - Jogo Batman: Arkham City



Fonte: <https://store.epicgames.com/pt-BR/p/batman-arkham-city>

Baseado na série de quadrinhos da DC Comics Batman é um jogo eletrônico de Ação-Aventura e Stealth produzido pela a Rocksteady, e publicado pelos os Estúdios Warner Bros. Interactive Entertainment e DC Entertainment. Disponível nas plataformas PS3,XBOX 360 E WINDOWS.

Figura 25 - Cenário do jogo Batman: Arkham City



Fonte:

https://store.steampowered.com/app/200260/Batman_Arkham_City__Game_of_the_Year_Edition/?l=portuguese

6.3.3 Borderlands 2

Figura 26 - Jogo Borderlands 2



Fonte: <https://store.epicgames.com/pt-BR/p/borderlands-2>

É um jogo do gênero ficção científica, e tiro em primeira pessoa, RPG eletrônico de ação. Disponível na Epic Games. Produzido pela Gearbox Software, Aspyr Media. Compatível com Microsoft Windows, PlayStation 3, Xbox 360, PS Vita e Nvidia Shield TV.

Figura 27 - Cenário do Jogo Borderlands 2

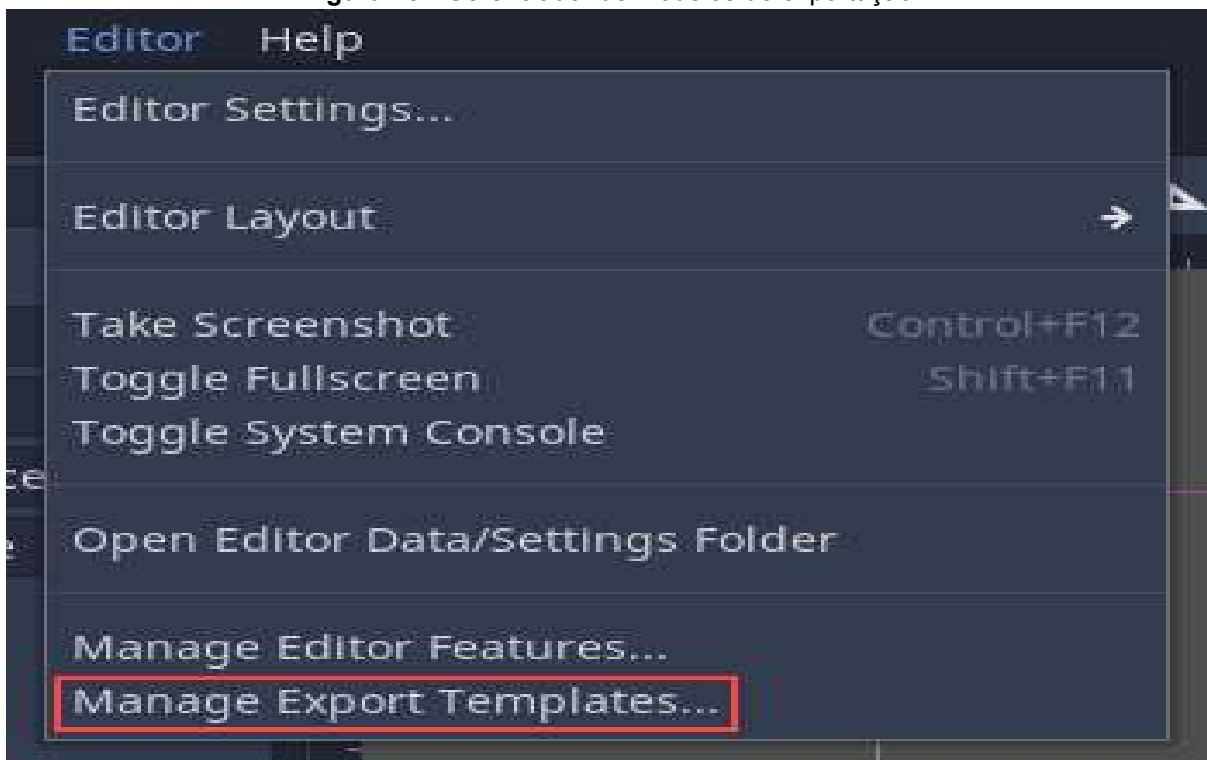


Fonte: <https://store.epicgames.com/pt-BR/p/borderlands-2>

7 MEIOS DE LANÇAR O JOGO

7.1 Godot

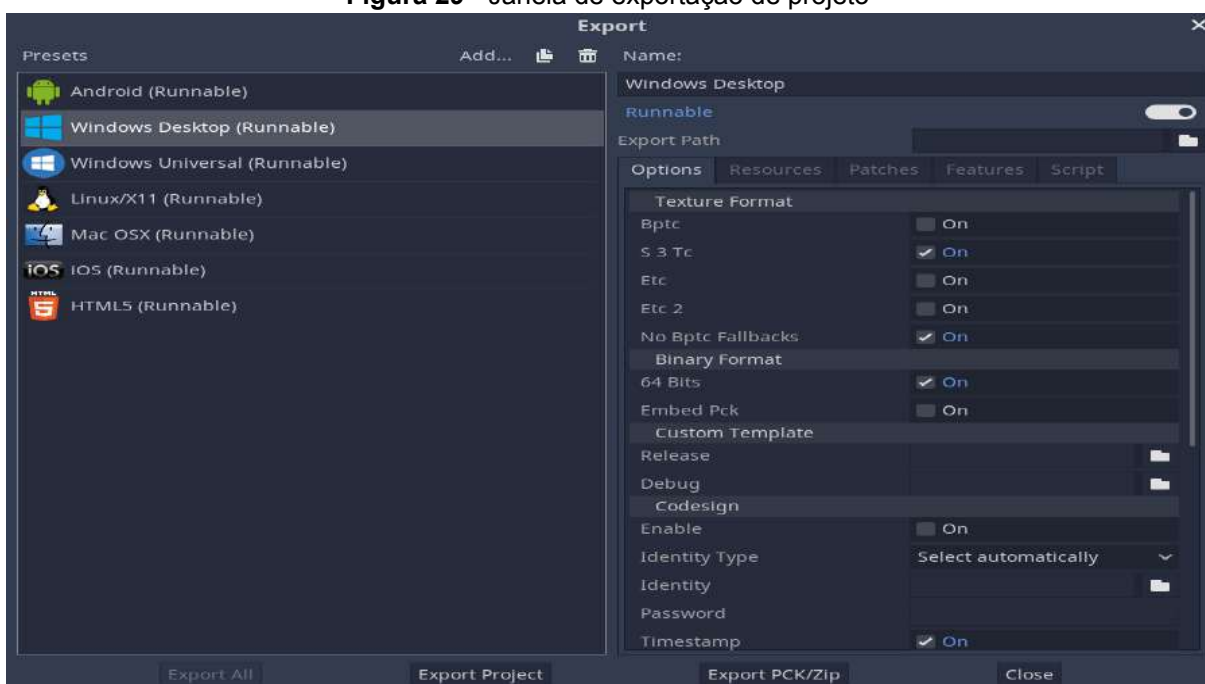
A exportação na Godot se passa pelo recurso de modelos de exportação, disponível através do site <http://godotengine.org/download>, oficial da engine. Ou baixar diretamente da Ferramenta de motor de jogos Godot:

Figura 28 - Gerenciador de modelos de exportação

Fonte: https://docs.godotengine.org/en/stable/tutorials/export/exporting_basics.html

Após baixar é possível predefinir as exportações a fim de predefinir várias plataformas, e outra configuração a fim de ganho de tempo é objetivo. Porém é possível definir como o jogo será exportado para cada plataforma separadamente na guia Resources.

Figura 29 - Janela de exportação de projeto



https://docs.unity3d.com/en/stable/tutorials/export/exporting_basics.html

Para criar uma versão executável completa para a plataforma desejada, basta clicar em Export Project.

7.2 Unity

A Unity consegue distribuir conforme a plataforma de escolha do seu projeto, ou seja, plataformas diferentes terão formas de compartilhar o jogo diferente.

Por exemplo, a Unity possui um portal de distribuição, UDP(Unity Distribution Portal), que tem como objetivo facilitar a distribuição de jogos para diversas plataformas que possuem uma loja de aplicativos. Segundo a Documentação. O UDP reembala sua compilação do Android com o SDK de compra no aplicativo dedicado de cada loja para tornar seu jogo compatível com as lojas de aplicativos separadas. Você pode gerenciar todos os envios da sua loja no console UDP.

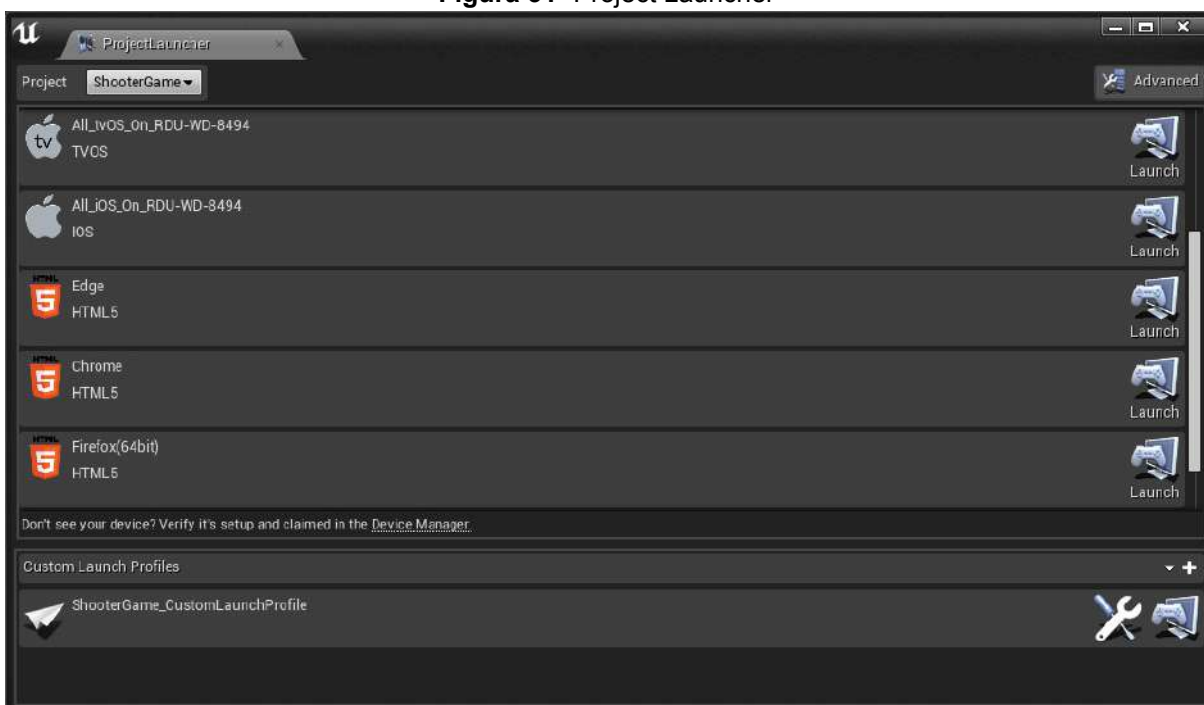
Figura 30 - Processo de compilação

<https://docs.unity3d.com/Manual/udp.html>

Disponibiliza também a opção de usar um editor para isso ou auto-hospedagem em servidores próprios, porém tem também seu próprio serviço ,Cloud Content Delivery, que já vem integrado a plataforma de desenvolvimento Unity, garantindo um ganho de atualizações de conteúdo regularmente.

7.3 Unreal Engine

Para distribuir o jogo na Unreal é preciso acessar o Project Launcher no editor ou então ir na guia Project Launcher no Unreal Frontend, ambas possui o mesmo destino e funcionalidades, porém poderia ser diferente conforme o jogo seja do windows e o objetivo seja criar um instalador com destino a uma plataforma móvel ou outra plataforma, as etapas serão diferentes.

Figura 31 -Project Launcher

<https://docs.unrealengine.com/5.1/en-US/using-the-project-launcher-in-unreal-engine/>

Segundo a Documentação da Unreal Engine, A interface do Project Launcher pode ser dividida em duas áreas principais:

Figura 32 - Interface do Project Launcher

<https://docs.unrealengine.com/5.1/en-US/using-the-project-launcher-in-unreal-engine/>

1. Perfis de inicialização padrão
2. Perfis de lançamento personalizados

8 CONSIDERAÇÕES FINAIS

Conclui-se que, é mais vantajoso um programador indie, usar uma engine já adotada por outros desenvolvedores similares, engines já difundidas no mercado já tem um grande avanço em sua engine, consegue se manter ativa e com atualização necessárias, e tem uma comunidade, que ajuda em possíveis dúvidas, parcerias, sugestões de melhorias para própria engine e até mesmo dependendo da engine uma participação em uma nova feature adotada pela engine.

Também é preciso identificar qual nicho será escolhido de atuação, saber se a engine tem um bom suporte para a plataforma e mecânicas do jogo escolhida é primordial, pois existem engine que são especialista em determinada plataforma ou modo de jogo. Um exemplo disso são games engines que tem como foco plataformas 2D, como é o caso da engine GameMaker Studio 2, é ainda consegue exportar para PLAYSTATION 5, XBOX SERIES X/S, NINTENDO SWITCH, GX.GAMES E STEAM. Mesmo uma engine que não seja a referência em alguns casos, é preciso estudar a possibilidade de utilizá-la, pois não existe uma engine superior em todos os aspectos, não é apenas as engines citadas por este trabalho que vale a pena analisar se suporta sua ideia, é saber que a melhor engine é aquela que atende a necessidade do seu jogo.

REFERÊNCIAS

CLUA, Esteban Walter Gonzalez; BITTENCOURT, João Ricardo. **Desenvolvimento de jogos 3D: concepção, design e programação**. In: Anais da XXIV Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação. 2005. p. 1313-1356.

DA SILVA FILHO, Gerônimo Oliveira et al. Unity: **Criando jogos e outras aplicações multi-plataforma**.

Disponível em 2022< <https://blog.dankicode.com/engines-para-jogos-2d/>>

Acesso em 04 de outubro-2022

Disponível em 2022< <https://docs.godotengine.org/en/stable/index.html>> Acesso em 10 de outubro-2022

Disponível em 2022< <https://docs.unity3d.com/Manual/udp.html>> Acesso em 04 de novembro-2022.

Disponível em 2022<

<https://docs.unrealengine.com/5.1/en-US/unreal-engine-cpp-quick-start/>>

Acesso em 15 de novembro-2022.

Disponível em 2022<<https://kingdomsofthedump.com/>> Acesso em 06 de novembro-2022.

Disponível em 2022<<https://godotengine.org/showcase/gravity-ace>> Acesso em 04 de novembro-2022.

Disponível em 2022<<https://gravityace.com/>> Acesso em 04 de novembro-2022.

Disponível em

2022<https://store.steampowered.com/app/1638500/The_Garden_Path/> Acesso em 10 de novembro-2022.

Disponível em 2022<<https://store.steampowered.com/app/268910/Cuphead/>> Acesso em 09 de novembro-2022.

Disponível em

2022<<https://www.xbox.com/pt-BR/games/store/cuphead/9NJR71M5X9P>>

Acesso em 02 de novembro-2022.

Disponível em

2022<<https://br.ign.com/pokemon-lets-go-pikachu-e-eevee/68589/news/pokemon-lets-go-quebra-recorde-de-vendas-no-switch>> Acesso em 04 de novembro-2022.

Disponível em 2022<<https://pokemongolive.com/>> Acesso em 05 de novembro-2022.

Disponível em

2022<<https://www.paradoxinteractive.com/games/cities-skylines/about>> Acesso em 15 de novembro-2022.

Disponível em 2022<<https://store.epicgames.com/pt-BR/p/fornite>> Acesso em 18 de novembro-2022.

Disponível em

2022<https://store.steampowered.com/app/200260/Batman_Arkham_City_Game_of_the_Year_Edition/?l=portuguese> Acesso em 18 de novembro-2022.

Disponível em 2022<<https://store.epicgames.com/pt-BR/p/borderlands-2>> Acesso em 20 de novembro-2022.

Disponível em 2022<<https://blog.dankicode.com/engines-para-jogos-2d/>> Acesso em 20 de novembro-2022.

Disponível em 2022<<https://godotengine.org/download>> Acesso em 27 de novembro-2022.

Disponível em 2022<<http://godotengine.org/download>> Acesso em 29 de novembro-2022.

Disponível em

2022<https://docs.godotengine.org/en/stable/tutorials/export/exporting_basics.html> Acesso em 26 de novembro-2022.

Disponível em 2022<<https://docs.unity3d.com/Manual/udp.html>> Acesso em 05 de novembro-2022.

Disponível em 2022<<https://www.unrealengine.com/pt-BR/license>> Acesso em 05 de dezembro-2022.

Disponível em 2022 <<https://store.unity.com/pt/#plans-individual>> Acesso em 08 de dezembro-2022.

SCHERER, Daniel; BATISTA, Daniele Ventura; DE CANTALICE MENDES, Aline. Análise da Evolução de Engines de Jogos. In: **Anais do V Congresso sobre Tecnologias na Educação**. SBC, 2020. p. 425-434.