



**SERVIÇO PÚBLICO FEDERAL - MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
UNIDADE CONTAGEM
ENSINO MÉDIO**

ISABELA MAGESTE DE ANDRADE
LUAN SETA RAMOS
MIGUEL DOS SANTOS MIRANDA

OS DESAFIOS DO DESENVOLVIMENTO DE JOGOS INDIE NO BRASIL

CONTAGEM
2025

ISABELA MAGESTE DE ANDRADE

LUAN SETA RAMOS

MIGUEL DOS SANTOS MIRANDA

OS DESAFIOS DO DESENVOLVIMENTO DE JOGOS INDIE NO BRASIL

Trabalho de Conclusão de Curso
apresentado ao Curso Técnico em
Informática do Centro Federal de Educação
Tecnológica de Minas Gerais (CEFET-MG)
como requisito parcial para a obtenção do
diploma de Técnico em Informática.

Orientador: Matheus Luz Rocha

Co-Orientador: Ivan Fontainha Alvaredo

Co-Orientador: Thiago Augusto Alves

CONTAGEM

2025

AGRADECIMENTOS

A realização deste Trabalho de Conclusão de Curso só foi possível graças ao apoio e incentivo de várias pessoas e instituições, às quais expressamos nossos mais sinceros agradecimentos:

Primeiramente, agradecemos aos nossos orientadores Matheus, Thiago e Ivan, pela paciência, orientação e dedicação ao longo de todas as etapas deste trabalho. Suas contribuições foram essenciais para o nosso desenvolvimento e para a qualidade deste projeto.

Aos nossos amigos André Victor, Isabelle Tavares, Isaqué, João Vitor Lima, Sarah Rabelo, João Gabriel Bicalho, Gabriel Agostinho, Davi Augusto, Pedro Guimarães de Deus, Arthur Henrique Brum, Daniel Pacheco, Clarissa Sousa, Miguel Ferraz, Otto Emanuel, Vinícius Domingos e Rômulo Ruggiero, que estiveram conosco nos momentos de dificuldades e compartilharam suas experiências, apoio e amizade durante toda a jornada.

Ao CEFET-MG, instituição de ensino que nos proporcionou uma educação de qualidade e um ambiente de aprendizado e crescimento, oferecendo todas as condições para a realização deste trabalho.

Ao professor Alisson, por seu apoio, incentivo e ensinamentos ao longo do curso, que foram fundamentais para nosso desenvolvimento acadêmico e profissional.

A todos, nosso muito obrigado.

RESUMO

O desenvolvimento de jogos independentes (*indie*) no Brasil enfrenta desafios significativos, incluindo a limitação de recursos financeiros e a falta de apoio institucional. Este trabalho tem como objetivo principal explorar e demonstrar esses desafios por meio da criação de um jogo *indie* utilizando a plataforma Unity e a linguagem C#. Especificamente, o trabalho busca analisar o cenário atual do desenvolvimento de jogos *indie* no Brasil, desenvolver um jogo que enquanto parte dessa realidade, listar os recursos necessários e dificuldades encontradas, e validar a viabilidade da Unity como ferramenta acessível para desenvolvedores de baixa renda. A metodologia consistiu em duas fases: na primeira, realizou-se uma pesquisa teórica sobre o cenário de jogos *indie* no Brasil, incluindo entrevistas com desenvolvedores e coleta de dados *on-line*; na segunda, foi desenvolvido um jogo como estudo de caso para exemplificar na prática os desafios identificados. Este processo teve como prerrogativa, apresentar as dificuldades enfrentadas por desenvolvedores independentes, ao mesmo tempo que ilustra o potencial criativo e a resiliência exigida para superar essas barreiras.

Palavras-chave: Jogos *Indie*; Brasileiros; desenvolvimento ; Unity.

ABSTRACT

The development of independent (indie) games in Brazil faces significant challenges, including limited financial resources and a lack of institutional support. This study aims to explore and demonstrate these challenges by creating an indie game using the Unity platform and the C# programming language. Specifically, the project seeks to analyze the current state of the indie game development scene in Brazil, develop a game that reflects this reality, list the necessary resources and encountered difficulties, and validate Unity's viability as an accessible tool for low-income developers. The methodology was divided into two phases: the first involves theoretical research on the Brazilian indie game landscape, including interviews with developers and online data collection; the second entails developing a game as a case study to practically illustrate the identified challenges. This process aims to highlight the obstacles faced by independent developers while showcasing the creative potential and resilience required to overcome these barriers.

Keywords: Indie Games; Brazilian; Development; Unity.

LISTA DE FIGURAS

FIGURA 1 - Tela de jogo “Tennis for Two” em um osciloscópio.....	13
FIGURA 2 - Stardew Valley na loja da Steam.....	15
FIGURA 3 - Projeto de Mapa.....	25
FIGURA 4 - Trecho do mapa da escola.....	26
FIGURA 5 - Personagens Jogáveis.....	27
FIGURA 6 - Personagem Alice.....	28
FIGURA 7 - Personagem Laura.....	29
FIGURA 8 - Personagem Gabriela.....	30
FIGURA 9 - Personagem Ronaldinha.....	31
FIGURA 10 - Personagem Otto.....	32
FIGURA 11 - Personagem Bruno.....	33
FIGURA 12 - Imagem do Slime.....	34
FIGURA 13 - Imagem do Pássaro Kroka.....	35
FIGURA 14 - Imagem da Caveira Flamejante.....	36
FIGURA 15- Imagem da Gárgula.....	37
FIGURA 16 - Imagem do Torrão.....	38
FIGURA 17 - Imagem da Abelha Operária.....	39
FIGURA 18 - Imagem da Abelha Soldado.....	40
FIGURA 19 - Imagem da Abelha Rainha.....	41
FIGURA 20 - Imagem do Golem guardião.....	42
FIGURA 21 - Imagem do Diretor.....	43
FIGURA 22 - Café.....	44
FIGURA 23 - Pudim.....	45
FIGURA 24 - Coroa da rainha abelha.....	45
FIGURA 25 - Núcleo do Golem.....	46
FIGURA 26 - Crânio de dragão.....	46
FIGURA 27 - Amuleto do Piromante.....	47
FIGURA 28 - Pilares da Programação orientada a objetos.....	49
FIGURA 29 - Ataque.....	54
FIGURA 30 - Corrida.....	56
FIGURA 31 - Esquiva.....	56
FIGURA 32 - HUD.....	58

FIGURA 33 - Partículas de chuva.....	61
FIGURA 34 - Partículas de Pétalas.....	61
FIGURA 35 - Partículas de Poeira.....	62
FIGURA 36 - Diálogo dentro do jogo.....	62
FIGURA 37 - Animação de dano.....	67
FIGURA 38 - Outline de item Interagível.....	67
FIGURA 39 - Inimigos de curto alcance.....	73
FIGURA 40 - Inimigos de longo alcance.....	73
FIGURA 41 - Chefe - Abelha Rainha.....	74
FIGURA 42 - Menu principal.....	75
FIGURA 43 - Menu de pausa.....	75
FIGURA 44 - Ícone das missões.....	76
FIGURA 45 - Inventário Vazio.....	77

LISTA DE QUADROS

QUADRO 1 - E-mails Enviados.....	21
QUADRO 2 - Ferramentas Selecionadas.....	24
QUADRO 3 - Sistema de Diálogo.....	64
QUADRO 4 - Árvore de Animações da Unity.....	66
QUADRO 5 - Diagrama de Classes do Sistema de Sons.....	69
QUADRO 6 - Opinião sobre o design do jogo.....	97
QUADRO 7 - Opinião sobre a movimentação do jogador.....	98
QUADRO 8 - Opinião sobre o mapa e ambientação.....	99
QUADRO 9 - Opinião sobre a dificuldade do jogo.....	99
QUADRO 10 - Opinião sobre a interatividade do jogo.....	100
QUADRO 11- Opinião sobre o jogo.....	101
QUADRO 12 - Opinião sobre o design do jogo.....	102
QUADRO 13 - Opinião sobre a movimentação do jogador.....	103
QUADRO 14 - Opinião sobre o mapa e ambientação.....	103
QUADRO 15 - Opinião sobre a dificuldade do jogo.....	104
QUADRO 16 - Opinião sobre a interatividade do jogo.....	104
QUADRO 16 - Opinião sobre o jogo.....	105

SUMÁRIO

1. INTRODUÇÃO.....	10
1.1. Objetivos.....	12
1.1.1. Objetivo Geral.....	12
1.1.2. Objetivos Específicos.....	12
2. REFERENCIAL TEÓRICO.....	13
2.1. A História dos jogos digitais.....	13
2.2. O que são jogos indie.....	15
2.3. Desafios enfrentados pelos desenvolvedores.....	16
2.3.1. Apoio financeiro.....	16
2.3.2. Acesso ao Mercado Internacional.....	17
2.3.3. Qualificação da Mão-de-Obra.....	18
2.4. Estratégias e ferramentas usadas.....	18
2.4.1. O que é a Unity?.....	19
2.4.2. O que é o Aseprite?.....	20
2.5. Efeito Psicológico.....	20
3. METODOLOGIA.....	21
3.1. Entrevistas.....	21
3.2. Produção.....	24
3.2.1. Personagens.....	27
3.2.2. Inimigos.....	34
3.2.3. Inimigos Especiais.....	41
3.2.4. Itens.....	44
3.3. Estruturação de código.....	48
3.3.1. Pilares da Orientação à objetos.....	48
3.3.2. Nomes significativos.....	49
3.3.3. Tratamento de erros e testes constantes.....	49
3.3.4. Padrões de projeto aplicados.....	50
3.3.5. Refatoração Contínua.....	50
3.3.6. Versionamento do código.....	51
3.4. Mecânicas do jogo.....	52

<i>3.4.1. Programação do player.....</i>	52
<i>3.4.2. Sistema de Ataque.....</i>	53
<i>3.4.3. Sistema de input.....</i>	54
<i>3.4.4. Sistema de vida.....</i>	54
<i>3.4.5. Sistema de movimentação.....</i>	55
<i>3.4.6. Interação e objetos interagíveis.....</i>	57
<i>3.4.7. HUD.....</i>	57
<i>3.4.8. Mapas.....</i>	58
<i>3.4.9. Efeitos de pós-processamento.....</i>	59
<i>3.4.10. Partículas.....</i>	60
<i>3.4.11. Diálogo.....</i>	62
<i>3.4.12. Shaders.....</i>	65
<i>3.4.13. Sistema de sons.....</i>	67
<i>3.4.14. Game Feel.....</i>	71
<i>3.4.15. Persistência de dados.....</i>	71
<i>3.4.16. Comportamento dos inimigos.....</i>	72
<i>3.4.17. Menus e UI.....</i>	74
<i>3.4.18. Objetivos e Missões.....</i>	76
<i>3.4.19. Itens.....</i>	76
4. RESULTADOS.....	78
5. CONSIDERAÇÕES FINAIS.....	80
6. REFERÊNCIAS.....	81
7. APÊNDICES.....	84
7.1. APÊNDICE A - Entrevista com Wennes Soares.....	84
7.2. APÊNDICE B - Entrevista com Jair Bortolucci.....	92
7.3. APÊNDICE C - Avaliações da primeira demo.....	97
7.4. APÊNDICE D - Avaliações da segunda demo.....	102

1. INTRODUÇÃO

O desenvolvimento de jogos independentes tem ganhado destaque no cenário global, especialmente pela capacidade de inovar em mecânicas, narrativas e estilos artísticos. Esse mercado, um dos mais lucrativos do setor de tecnologia, teve um faturamento global estimado em US\$227,6 bilhões em 2024, apresentando crescimento em relação a anos anteriores e consolidando os jogos como um dos setores mais dinâmicos do entretenimento. A maior parte desse faturamento foi impulsionada por jogos móveis e casuais, que representaram cerca de 75% do total, incluindo receitas provenientes de publicidade em aplicativos (OBEDKOV, 2024).

O governo brasileiro iniciou em 2024, projetos de incentivo a essa área, conhecido como “Marco Legal dos Games” (BRASIL, 2024). Essa iniciativa visa incentivar o desenvolvimento de jogos no Brasil, oferecendo benefícios fiscais, programas de capacitação e acesso facilitado a linhas de crédito para desenvolvedores independentes. Contudo, sua implementação revelou uma série de limitações, como a burocracia para acessar os benefícios, a falta de clareza nos critérios de elegibilidade e o foco excessivo em grandes estúdios, que já possuem maior estrutura e recursos financeiros e não precisam realmente desse auxílio, proporcionando desvantagens para pequenos desenvolvedores que podem, em diversas situações, não conseguir atender aos requisitos para se beneficiar do programa. Esses obstáculos evidenciam a necessidade de ajustes nas políticas públicas para o setor, com maior atenção às necessidades específicas dos desenvolvedores independentes. Afinal, são estes que frequentemente trazem inovação e diversidade para o mercado, mas que permanecem como os mais vulneráveis no contexto atual.

A relevância da indústria de jogos se evidencia ainda mais quando comparada a outros setores de entretenimento, sendo maior que a indústria da música e do cinema combinadas. A indústria musical, que gerou US\$32 bilhões, e o cinema, com US\$95 bilhões no mesmo ano (PwC, 2024). Apesar do destaque financeiro, o sucesso não é garantido, especialmente para os desenvolvedores independentes, que enfrentam diversos desafios para se destacar em um mercado saturado e competitivo, tendo acesso a pouco incentivo ou visibilidade. Um fator que demonstra isso é que apenas cerca de 20% dos jogos independentes recuperam o investimento inicial, e uma parcela ainda menor consegue alcançar lucros

significativos. Os desafios incluem a alta competição, falta de visibilidade, limitações financeiras e dificuldades em alcançar o público-alvo (GameLab HQ, 2024). Além disso, fatores como a necessidade de múltiplas habilidades técnicas, restrições de marketing e as rápidas mudanças tecnológicas aumentam as barreiras para esses desenvolvedores.

Além da questão monetária, os jogos digitais podem tomar duas direções, trazer benefícios melhorando vários aspectos da vida do jogador ou trazer outros problemas (MARQUES *et al*, 2023). Frustrações em suas vidas normais podem fazer com que passem mais tempo jogando, assim causando-os a ficarem ausentes de suas próprias vidas ou emocionalmente desvinculados. Além disso, os jogos digitais também podem servir de escape de pensamentos ruins e desafios na vida real. Nesse cenário, jogadores podem estar tão envolvidos no mundo virtual que esquecem de suas preocupações reais, preferindo “viver” no mais desejável. Embora o escapismo da vida real por meio de jogos possa parecer uma opção atrativa, ele pode ter efeitos adversos no bem-estar de jogadores isolados com baixa autoestima . Para jogadores emocionalmente sensíveis, os espaços virtuais devem ser mediados por espaços sociais para prevenir impactos negativos no bem-estar no mundo real . Além disso, quando o mundo virtual começa a se confundir com a realidade, os jogadores podem, inadvertidamente, começarem a negligenciar a si mesmos, seus relacionamentos e atividades essenciais de autocuidado. Há também evidências de experiências dissociativas aumentadas, como despersonalização e desrealização.

Diante desse cenário, este trabalho tem como objetivo principal analisar os desafios enfrentados pelos desenvolvedores independentes no Brasil, compreendendo o contexto social, econômico e tecnológico em que estão inseridos. A pesquisa busca identificar os fatores determinantes para o sucesso nesse mercado competitivo, considerando aspectos como acesso a recursos financeiros, estratégias de marketing, visibilidade nas plataformas de distribuição e utilização de ferramentas de desenvolvimento. Além disso, propõe-se explorar como a criatividade e a inovação podem ser empregadas como diferenciais para superar as dificuldades. A partir dessa análise, o trabalho visa propor abordagens práticas que ajudem os desenvolvedores a enfrentar as barreiras mais comuns, contribuindo para o fortalecimento do setor *indie* no país.

A pesquisa adota uma abordagem mista para abordar as questões propostas, combinando análise e coleta de dados com a exploração prática de um projeto desse tipo. Primeiramente, serão realizadas entrevistas com desenvolvedores independentes, permitindo uma compreensão aprofundada das experiências práticas, desafios enfrentados e estratégias utilizadas no mercado brasileiro. Em paralelo, será conduzida uma análise de dados secundários, utilizando relatórios de mercado, artigos acadêmicos e estudos de caso para enriquecer a base teórica do trabalho. Como parte prática, será desenvolvido um jogo digital que funcionará como estudo de caso, possibilitando a validação de ferramentas e métodos utilizados por desenvolvedores independentes. O projeto será documentado em todas as suas etapas, desde a concepção até a publicação, a fim de detalhar as experiências criativas e produtivas. O objetivo final é oferecer uma visão abrangente e prática sobre as ferramentas e estratégias mais eficazes, contribuindo para o avanço da comunidade *indie* no Brasil e destacando possíveis soluções para os desafios enfrentados.

1.1. Objetivos

1.1.1. *Objetivo Geral*

Explorar e demonstrar os desafios do desenvolvimento de jogos independentes no Brasil, com foco na criação de um jogo indie como estudo de caso.

1.1.2. *Objetivos Específicos*

- Desenvolver um jogo enquanto equipe inserida nesse contexto;
- Listar dificuldades enfrentadas e recursos necessários para a execução desse tipo de projeto;
- Validar ferramentas que podem ser usadas no desenvolvimento de um jogo;
- Obter relatos reais e agregar sua experiência ao projeto.

2. REFERENCIAL TEÓRICO

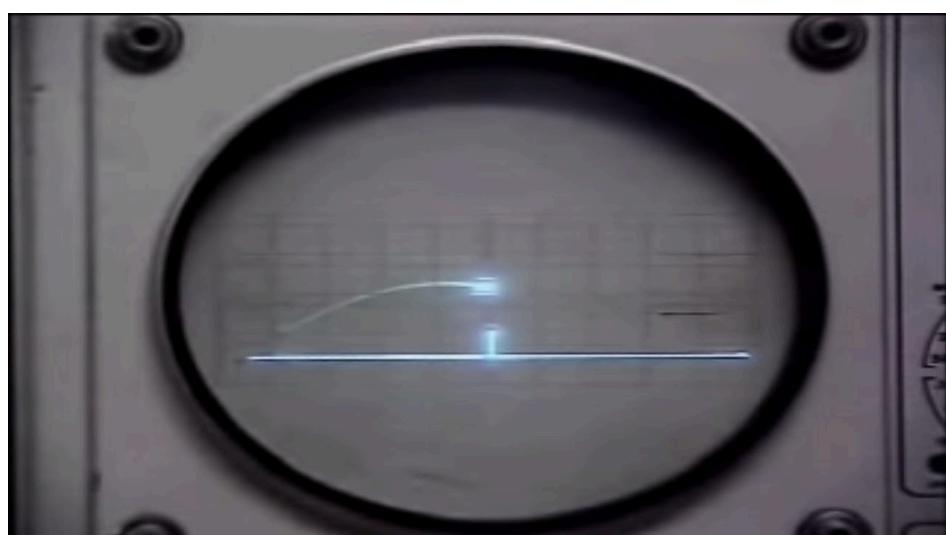
A pesquisa do tema se baseou em artigos científicos, dissertações, sites populares e livros teóricos para analisar e organizar as informações relevantes, bem como dados relacionados. Ao longo dos tópicos levantados, pretende-se explicar a origem dos jogos digitais, o que são jogos *indie*, quais desafios eles trazem consigo, assim como o mercado do qual participam.

2.1. A História dos jogos digitais

Desde seu surgimento inicial como ferramenta para aplicar a tecnologia de maneira visual, os jogos digitais passaram por transformações em sua essência, ampliando seu alcance e relevância em diversas esferas sociais, sendo atualmente uma versão com objetivo e significado diferente.

Dentre esses, se destaca o primeiro jogo criado, “tennis for two”, desenvolvido em 1958 por William Higinbotham, um físico americano. O sistema foi projetado como uma demonstração para atrair visitantes ao laboratório de pesquisas nucleares Brookhaven National Laboratory durante um dia aberto ao público, e seu objetivo principal era mostrar como a tecnologia poderia ser usada de forma divertida e interativa.

FIGURA 1 - Tela de jogo “Tennis for Two” em um osciloscópio



Fonte: BROOKHAVEN NATIONAL LABORATORY. BNL: First Video Game. Disponível em: <https://www.bnl.gov/about/history/firstvideo.php>. Acesso em: 14 dez. 2024.

Na década de 1970, os primeiros jogos digitais comerciais, como os arcades (ou fliperamas), marcaram o surgimento de uma indústria voltada para a produção e consumo em larga escala (REIS, 2014). A popularização, no entanto, ocorreu nos anos 1980, quando os videogames começaram a se consolidar como uma forma de entretenimento acessível e difundida. Esse crescimento demonstra como os jogos evoluíram rapidamente para atender a diferentes públicos e mercados, tendo variantes disponíveis para todas as faixas etárias, embora menos recebidos entre as gerações mais velhas.

O mercado de jogos digitais é um dos setores mais lucrativos da indústria do entretenimento, superando até mesmo os mercados de música e cinema combinados em receita (WAKKA, 2021). Esse cenário reflete o impacto dos jogos digitais na economia global, gerando milhões de empregos diretos e indiretos em áreas como desenvolvimento, *design*, marketing e esportes eletrônicos. Além disso, os jogos desempenham um papel estratégico na inovação tecnológica, uma vez que permitem testar novas tecnologias impulsionando avanços em áreas como inteligência artificial e realidade aumentada. No Brasil, por exemplo, a indústria de jogos contribui para a economia local, especialmente com o crescimento de desenvolvedores independentes, que colocam o país em destaque no cenário global de produção criativa (GAMELAB HQ, 2024). Esse desenvolvimento mostra como o mercado de jogos digitais faz parte do futuro do progresso econômico e tecnológico em diversos países.

Esse desenvolvimento demonstra que os jogos digitais não são apenas artefatos de lazer, mas também produtos culturais que refletem e influenciam os comportamentos e valores contemporâneos. Sua trajetória, marcada por inovações constantes, continua a moldar a maneira como experimentamos o entretenimento e a interação social. Durante atividades de jogo em família ou entre amigos, eles criam oportunidades para o compartilhamento de momentos de diversão e colaboração, promovendo comunicação e entendimento mútuo (VOIDA; CARROLL, 2013). Além disso, o ambiente imersivo dos jogos permite que pessoas se conectem emocionalmente, mesmo à distância, seja por meio de plataformas multiplayer ou de encontros para jogatinas presenciais. No que diz respeito à saúde mental, os jogos atuam como ferramentas valiosas para alívio do estresse e da ansiedade, proporcionando um espaço seguro onde os jogadores podem relaxar e se expressar criativamente (PRZYBYLSKI *et al.*, 2010). Assim, os jogos digitais não apenas

entretêm, mas também fomentam conexões significativas e oferecem benefícios emocionais e psicológicos.

2.2. O que são jogos indie

Os jogos *indie*, abreviação de *independent games*(jogos independentes), são produzidos por pequenos desenvolvedores ou estúdios sem apoio financeiro ou logístico de grandes publicadoras. Diferentemente dos jogos feitos por grandes empresas, os jogos *indie* se destacam pela liberdade criativa, pois muitas vezes seus criadores exploram ideias, estilos artísticos e narrativas de maneira única. Isso possibilita o desenvolvimento de experiências que fogem dos padrões comerciais tradicionais, muitas vezes abordando temas complexos ou experimentais, ou a mesmo críticas sociais pertinentes e pouco destacadas. Apesar de terem orçamentos e equipes reduzidos, os jogos *indie* conseguem competir no mercado global graças a plataformas de distribuição digital, como Steam, Itch.io e consoles modernos, que democratizam o acesso ao público. Títulos como *Stardew Valley*, *Minecraft* e *Undertale* exemplificam o impacto cultural e comercial que os jogos *indie* têm alcançado, mostrando que é possível superar as limitações financeiras. Além disso, esses jogos fortalecem comunidades locais de desenvolvedores, especialmente em países emergentes como o Brasil, contribuindo para o crescimento e a diversidade da indústria global.

FIGURA 2 - Stardew Valley na loja da Steam



FONTE: Steam (2024).

O jogo “Stardew Valley” é um exemplo de sucesso de um jogo *indie*. O jogo lançado em 2016 demorou quatro anos para ser desenvolvido em sua versão mais

simples e menos popular e sua equipe de desenvolvimento é composta por apenas uma pessoa, Eric Barone. Da arte aos códigos, mesmo após o jogo ter explodido no final de 2017, Barone manteve sua equipe única, se recusando a vender o jogo ou formar um estúdio, se comprometendo sempre com a integridade de seu produto. Atualmente, é um dos jogos mais populares em escala global, com mais de 20 milhões de cópias vendidas pela Steam (Steam, 2024).

2.3. Desafios enfrentados pelos desenvolvedores

Embora o Brasil possua um mercado crescente e uma comunidade ativa de desenvolvedores, as dificuldades ainda são significativas, especialmente quando comparadas com o mercado competitivo dos jogos. Sendo assim, é de grande importância evidenciar quais são os desafios enfrentados pelos pequenos desenvolvedores de jogos.

2.3.1. Apoio financeiro

Um dos maiores obstáculos para o desenvolvimento de jogos *Indie* no Brasil é a baixa quantidade de investimentos e apoio financeiro. Muitos criadores não têm acesso a recursos que permitam o desenvolvimento de jogos com qualidade competitiva em nível internacional. Segundo pesquisa realizada pela ABRAGAMES (Associação Brasileira das Desenvolvedoras de Jogos Digitais), apenas 9% dos desenvolvedores de jogos *Indie* no Brasil tem acesso a financiamento público ou privado, sendo que a maioria dos estúdios depende de recursos próprios ou de empréstimos pessoais para viabilizar seus projetos (ABRAGAMES, 2022). Tendo em vista como o mercado de jogos pode ser lucrativo, essa situação denuncia um problema grave de infraestrutura no brasil, que é a insuficiênciа de investimento no mercado de tecnologias e inovações.

Algo potencializado por esse problema é o elevado custo de produção de jogos no Brasil, principalmente devido à alta carga tributária e à escassez de incentivos fiscais. Segundo dados do Portal Gameplan (2023), o Brasil possui uma das maiores taxas de impostos sobre a produção de conteúdo digital no mundo, o que torna os jogos mais caros de serem produzidos localmente. Além disso, a aquisição de ferramentas e licenças de software não é disponível para um grande

público , sendo que muitas dessas ferramentas possuem preços elevados, o que dificulta o acesso de pequenos estúdios.

2.3.2. Acesso ao Mercado Internacional

A falta de visibilidade e o acesso limitado a mercados internacionais são questões que impactam o sucesso dos jogos *Indie* brasileiros. A competição global é feroz, com países como os Estados Unidos, Japão e países da Europa dominando as principais plataformas de distribuição digital como Steam e Epic Games Store. De acordo com um levantamento da Brazilian Games Show (BGS), 70% dos desenvolvedores de jogos *Indie* brasileiros têm dificuldades em conquistar um público internacional, principalmente devido à barreira linguística e à falta de investimento em marketing (BGS, 2021). Além disso, eventos internacionais ainda são inacessíveis para muitos desenvolvedores brasileiros devido aos custos de viagem e hospedagem, além da dificuldade de estabelecer contatos com investidores e *publishers* estrangeiros.

Isso é evidenciado na pesquisa do artigo “A Indústria e o Mercado de Jogos Digitais no Brasil”, que analisa o mercado brasileiro de jogos digitais, destacando seu potencial e suas discrepâncias em relação ao cenário global (AMÉLIO,2018). Apesar de o Brasil ter se consolidado como o quarto maior mercado consumidor de jogos em 2017, ele enfrenta desafios significativos no que diz respeito ao faturamento, ocupando apenas a 12^a posição mundial naquele ano, e caindo para o 13º lugar em 2018. Essa diferença reflete a lacuna entre o consumo em larga escala e a capacidade produtiva do país. A autora investiga os fatores que contribuem para essa disparidade, incluindo a ausência de políticas públicas robustas, dificuldades no acesso a financiamento privado, limitações na formação de profissionais especializados e os hábitos culturais da população. O estudo aponta que o fortalecimento dessas áreas é essencial para que o Brasil amplie sua competitividade no mercado global de jogos digitais.

Embora o Brasil tenha um mercado de jogos em crescimento, ainda há desafios culturais e de aceitação de jogos produzidos localmente, uma vez que existe uma tendência dos consumidores a desvalorizarem os produtos produzidos por brasileiros e preferirem jogos internacionais, criando uma vulnerabilidade. Isso é causado por uma percepção de que jogos brasileiros são sinônimo de qualidade

inferior, o que dificulta o reconhecimento dos jogos . A falta de validação e a escassez de canais de divulgação dedicados aos jogos desenvolvidos no país são desafios adicionais que impactam a competitividade dos micro estúdios.

2.3.3. Qualificação da Mão-de-Obra

Uma equipe de jogos *indie* é por costume formada por poucos integrantes que participam das diversas funções necessárias para a produção de um jogo. Dessa maneira, essas pessoas têm uma grande necessidade de qualificação prática em âmbitos como música, design, arte, programação, marketing entre outros. A formação de profissionais especializados nessas áreas ainda é uma barreira no Brasil, especialmente para formações que precisem ser rápidas e focadas na prática, a baixo custo. Embora as universidades brasileiras estejam cada vez mais oferecendo cursos voltados para o desenvolvimento de jogos, a maioria das opções de educação ainda não são amplamente acessíveis, especialmente em regiões mais afastadas dos grandes centros urbanos. Muitos estúdios de jogos *Indie*, especialmente os menores, enfrentam dificuldades para atrair e reter talentos devido às limitações financeiras e à competição com empresas maiores, que podem oferecer salários mais altos e mais estabilidade no mercado, além de ter dificuldade para auxiliar na formação de seus próprios funcionários. Adicionando ao problema, o Brasil possui uma legislação trabalhista e tributária complexa, que torna a gestão de uma empresa de desenvolvimento de jogos um processo desafiador. A carga tributária no setor de tecnologia no Brasil, além da complexidade na gestão de contratos com freelancers e colaboradores, pode ser um entrave ao sucesso, considerando os orçamentos apertados e uma equipes reduzidas, o que se torna mais um fator agravante na contratação da mão de obra necessária para projetos.

2.4. Estratégias e ferramentas usadas

Para enfrentar essas dificuldades, os desenvolvedores independentes têm adotado estratégias e utilizado ferramentas tecnológicas que ampliam suas possibilidades de criação e distribuição de jogos.

Em resposta aos problemas financeiros, os desenvolvedores têm adotado modelos de financiamento coletivo (*crowdfunding*) por meio de plataformas como Catarse e

Kickstarter. Além disso, a busca por editais de incentivo cultural, como os promovidos pela Ancine e pela Secretaria de Cultura, tem sido uma alternativa eficiente para viabilizar projetos. Essas iniciativas ajudam a reduzir a dependência de grandes investidores e mantêm a independência criativa dos desenvolvedores, além permitir à comunidade dar suporte aos seus jogos preferidos.

Para contornar o problema de capacitação, muitos profissionais recorrem a plataformas online de aprendizado, como a Udemy e a Alura, além de materiais educativos disponibilizados gratuitamente em canais do YouTube, como “Crie Seus Jogos”.

Na questão dos softwares, muitos desenvolvedores optam por ferramentas gratuitas ou de código aberto, como Unity, Unreal Engine e Godot, que oferecem funcionalidades robustas para o desenvolvimento de jogos. Além disso, o uso de softwares como Blender para modelagem 3D e GIMP para edição de imagens reduz os custos de produção sem comprometer a qualidade do produto final (RAMOS; MIRANDA, 2024). Na arte 2D, os programas Aseprite, Procreate, Photoshop são muito populares, devido a sua qualidade e custo benefício.

2.4.1. O que é a Unity?

A Unity é uma plataforma de produção de jogos gratuita que permite desenvolver jogos 2D e 3D para diversos aparelhos diferentes, permitindo alcançar diversos usuários. Ela renderiza as imagens, gerencia os acontecimentos do jogo, as entradas do usuário, colisão, memória, entre outras funções. O ambiente de programação permite um grande avanço em períodos curtos de tempo. Sua interface é complexa à primeira vista, mas com a prática, ela se torna uma ferramenta poderosa na produção de jogos, muito usada por desenvolvedores indie.

A escolha da Unity foi influenciada pelo artigo “Game engines: a survey”, que analisa e compara as principais ferramentas disponíveis para o desenvolvimento de jogos digitais, destacando suas funcionalidades, vantagens e limitações (ANDRADE, 2015). O artigo oferece um panorama detalhado das game engines, considerando critérios como acessibilidade, curva de aprendizado, recursos técnicos e compatibilidade com diferentes plataformas, e com base nisso, a Unity se destacou por sua versatilidade, suporte a múltiplas plataformas, extensa documentação e comunidade ativa, fatores que a tornam uma das ferramentas mais

utilizadas por desenvolvedores independentes e grandes estúdios. Essa análise foi essencial para ter uma comparação comprovada e confiável das diversas plataformas e selecionar a Unity como mais viável para esse projeto.

2.4.2. O que é o Aseprite?

Aseprite é um programa gratuito de animações e produção de *sprites* em estilo *pixel art*, amplamente usado para artes de jogos, devido a sua grande quantidade de atalhos que permitem uma produção rápida e de qualidade. Além disso, há uma lista de comandos disponíveis para facilitar o processo criativo.

No campo do marketing e mercado, desenvolvedores têm explorado redes sociais, como Twitter, Instagram e TikTok, para divulgar seus projetos, criar comunidades engajadas e atrair potenciais consumidores. Além disso, plataformas de distribuição digital, como Steam e itch.io, têm sido amplamente utilizadas para acessar mercados globais, ampliando a visibilidade dos jogos brasileiros (SANTOS, 2023).

2.5. Efeito Psicológico

Por último, mas não menos importante, é conveniente destacar o efeito psicológico dos jogos digitais, uma vez que suas capacidades podem ajudar imensamente na recuperação da saúde mental e descanso da psique em situações de dificuldade. Em uma visão positiva, durante o período de isolamento decorrido da pandemia da COVID-19, esses jogos se tornaram um meio eficaz de manter o bem-estar social através de interações digitais, além de oferecer um escape de uma rotina monótona e solitária. Eles não só incentivam a cooperação e o trabalho em equipe, mas também permitem que os jogadores experimentem diferentes estilos de vida e revelam aspectos de si mesmos que talvez não consigam expressar no cotidiano. Dessa maneira, o mundo virtual dos jogos estabelece uma dinâmica social inclusiva, proporcionando a oportunidade de vivenciar múltiplas identidades e contribuindo para a preservação da saúde mental em tempos de isolamento. Essa forma de interação permitiu também a interação social entre membros de diversas culturas diferentes, fortalecendo laços humanos em sua forma mais pura: a amizade.

3. METODOLOGIA

A seção prática do trabalho foi dividida em duas partes, uma para entrevistas com desenvolvedores *indie* e a outra sendo o desenvolvimento de um jogo a fim de estudo de caso e validar ferramentas que podem ser usadas para tentar contornar os desafios encontrados.

3.1. Entrevistas

O interesse do grupo em entrevistar profissionais na área envolvia principalmente ter fontes confiáveis sobre a experiência da criação de um jogo *indie*. Com esse objetivo em mente, selecionamos alguns criadores de conteúdo relacionado a jogos *indie*, entre eles, desenvolvedores e canais de tutoriais relacionados ao desenvolvimento de jogos. Para contatar os convidados, usamos o gmail, e enviamos um email padrão editado para cada um . O email usado pode ser visto no QUADRO 1.

QUADRO 1 - E-mails Enviados

(continua)

<p>Assunto: Convite para Entrevista Sobre Desenvolvimento de Jogos <i>Indie</i> 🎮</p> <p>Olá [Nome do Desenvolvedor],</p> <p>Espero que o E-mail te encontre bem!</p> <p>Meu nome é Isabela Mageste de Andrade, e estou no ensino médio técnico em Informática no CEFET-MG. Junto com meus colegas Luan Seta Ramos e Miguel dos Santos Miranda, estamos trabalhando em um TCC que explora os desafios e as estratégias do desenvolvimento de jogos <i>indie</i> no Brasil.</p> <p>Estamos desenvolvendo a pesquisa do nosso projeto e achamos que a sua experiência como desenvolvedor <i>indie</i> seria incrível para nos ajudar a entender melhor o que está por trás das dificuldades e sucessos nesse universo.</p>

(conclusão)

Gostaríamos de convidá-lo(a) para uma entrevista online na plataforma Discord, no horário que for mais conveniente para você. Será um prazer ouvir suas experiências e desafios enquanto desenvolvia seu jogo [nome do jogo]!

Se estiver interessado(a), é só nos avisar e ficaremos felizes em organizar tudo. Se precisar de mais informações ou tiver alguma dúvida, estamos disponíveis.

Desde já, agradecemos de coração por considerar nosso convite e esperamos poder conversar em breve!

Cordialmente,

Isabela Mageste de Andrade

Luan Seta Ramos

Miguel dos Santos Miranda

CEFET-MG

FONTE: Autoria própria (2024).

Dentre a lista de convidados, houveram duas respostas e duas entrevistas, cuja transcrição consta nos apêndices A e B, com os dados do entrevistado correspondente.

As entrevistas tiveram grande importância no progresso do grupo, uma vez que as boas práticas aconselhadas foram de grande ajuda. Por exemplo, na questão do uso de *assets* prontos, como códigos ou algoritmos específicos, permitiu salvar muito tempo , garantindo que a nossa atenção fosse dedicada às partes mais sensíveis do projeto.

“Uma das estratégias foi a de comprar *assets* , essa sem dúvidas é uma dica que vou dar para contornar esse problema de falta de recursos tanto financeira quanto técnica. Porque ao comprar um *asset* , por exemplo, você comprou um sistema de *behaviour three* , você não tem que programar aquilo e não perde seu tempo, e se tivesse que contratar uma pessoa para fazer aquilo ia perder tempo e dinheiro. Outra estratégia também foi enxugar o escopo, por exemplo, o CyberWar teria

um escopo muito maior que esse que foi lançado, e aí eu fui enxugando isso, vamos cortar chefões, itens, cenários, senão a gente ia engordurando e engordurando e não ia terminar o jogo.” (SOARES, 2024)

Outra recomendação muito pertinente foi em relação a organização do projeto. A experiência de Wennes ajudou o grupo a perceber problemas de organização fatais que poderiam colocar em risco a integridade do produto final em questão de qualidade. Após a recomendação do desenvolvedor, a documentação dos passos da idealização e do desenvolvimento do jogos se tornaram mais presentes, o que permitiu fechar o escopo e fazer uma etapa de cada vez, garantindo a integridade de cada uma.

“W: Então, a primeira etapa é a documentação básica do jogo para ter um Norte, Porque às vezes o cérebro engana né? [...] Então é bom documentar uma base, para você montar um protótipo para ver se a ideia é realmente divertida, mandar ele pros amigos, pra perguntar se eles acham que a ideia é legal e tal. A partir dali você valida o produto, se o protótipo com quadradinhos e bolinhas já foi divertido, é um sinal que você tá indo em uma boa direção, e aí você segue para a etapa de desenvolvimento, onde você já começa desenvolvendo o jogo, mas jamais utilizando o código do protótipo, aquele dali você esquece. E aí desenvolvendo o projeto do 0, aí você é mais questão de se organizar, porque depende né, se você tá fazendo sozinho ou em equipe, se for uma equipe recomendo ali uma reunião semanal ou quase isso, para estar alinhado do que tá acontecendo e do que não tá, colocar tarefas, metas, importantíssimo colocar as metas. Ter uma ideia do escopo do seu jogo e tentar manter a ideia do escopo, não sair muito dela, e por isso é muito importante a documentação do início.” (SOARES, 2024)

3.2. Produção

A etapa de produção do jogo foi a mais trabalhosa, e a que colocou em prática os aprendizados do curso de informática do CEFET-MG. Essa parte do trabalho foi dividida em pedaços menores, cada um com sua importância. O primeiro passo foi decidir o estilo de projeto, seguido por decidir as melhores ferramentas para realizá-lo e por fim a execução.

Tendo isso em vista, a TABELA 1 destaca quais as ferramentas escolhidas com qual propósito.

QUADRO 2 - Ferramentas Selecionadas

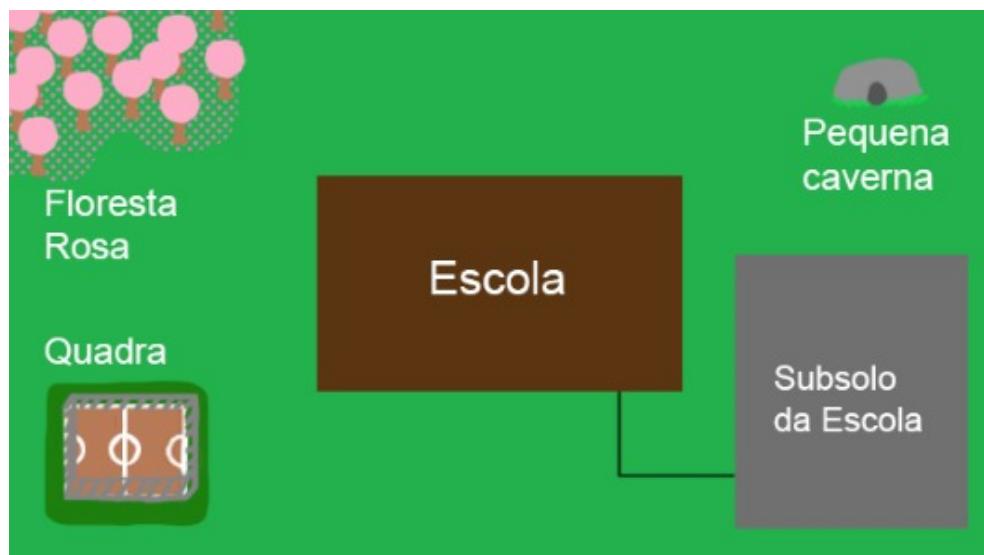
Função	Software	Logo	Versão	Fonte
<i>Engine</i>	Unity		2022.3.20.F1	https://unity.com/pt/download
<i>Sprites</i>	Aseprite		13.10	https://www.aseprite.org/download/
Edição de código/ Programação	Visual Studio		2022	https://visualstudio.microsoft.com/pt-br/downloads/
Organização de Código e versionamento	Git Hub		2.47.1	https://desktop.github.com/download/
Som e música	Unity Asset store		**	https://assetstore.unity.com

FONTE: Autoria própria (2024).

Após isso, iniciou-se a fase de definições da história e mecânicas do jogo, que após múltiplas reuniões foi decidido em uma estrutura baseada em diferentes

áreas . A FIGURA 3 mostra um esquema de como foram organizados as diferentes seções do jogo, em sua idealização. Foi decidido que haveria um cenário interior e um exterior, se passando em uma escola. No interior, haveriam npcs de estudantes e itens relacionados a uma escola. Seria possível fazer compra e venda de itens conversando com os personagens, e fazer missões em troca de outros itens ou habilidades. Foi determinado também que no Subsolo e na área exterior haveria inimigos a serem enfrentados pelo jogador. A categoria de jogo escolhida se chama RPG (*Role Playing Game*), e para o estilo, o selecionado foi a perspectiva *top-down* com *pixel arts 16 bits*.

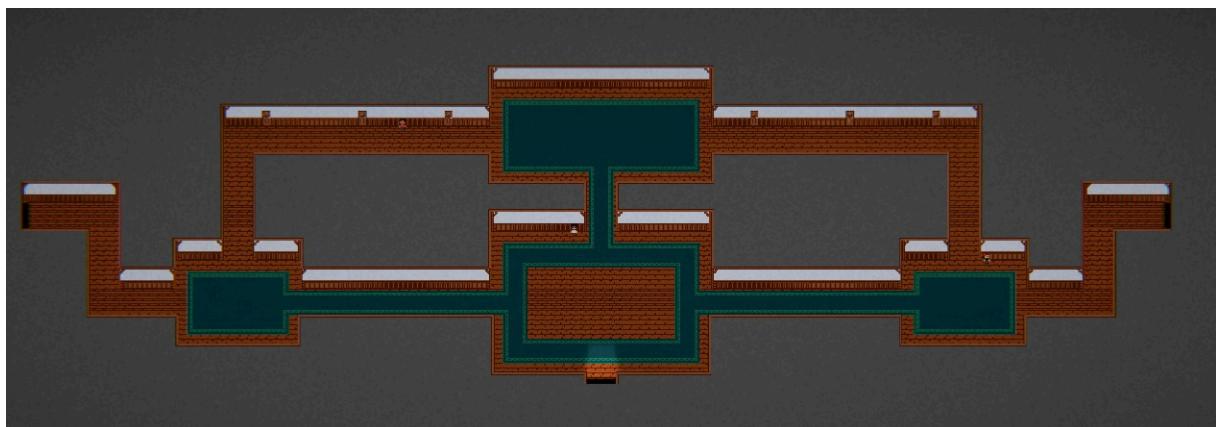
FIGURA 3 - Projeto de Mapa



FONTE: Autoria própria (2024).

Nessa imagem, é possível ver onde cada área ficaria no planejamento inicial. O mapa final apresenta diferenças estéticas no formato e caminhos, mas em essência, a ideia continuou a mesma ao longo do projeto. Após decidir o mapa, iniciou-se a produção de *sprites* (artes) do jogo. A FIGURA 4 destaca algumas das artes feitas para o mapa da escola.

FIGURA 4 - Trecho do mapa da escola



FONTE: Autoria própria (2024).

3.2.1. Personagens

Os personagens presentes no jogo foram divididos em dois tipos: NPCs e Jogadores.

Para os jogadores, dois personagens estão disponíveis, um menino e uma menina, chamados de André e Sofia. No início do jogo, o usuário pode escolher com qual deles irá jogar pelo resto da história daquele arquivo salvo. Na FIGURA 5, constam André, na esquerda, e Sofia, na direita.

FIGURA 5 - Personagens Jogáveis



FONTE: Autoria própria (2024).

A arte do jogo se inspirou em jogos como “Enter the Gungeon”, “Stardew Valley” e pacotes de assets gratuitos, como “Mystic Woods”, disponíveis no site Itch.io de maneira gratuita para usar e alterar em projetos.

Já os personagens não jogáveis foram definidos um por um, com suas aparências, nomes e missões, além de histórias curtas para aumentar a imersão do jogo, dando a cada um dos personagens alguma profundidade, ainda que pouca. Esses personagem tem como propósito dar instruções ao jogador, entregar itens e tornar o jogo mais interativo.

Foi determinado, para facilitar a organização, as seguintes classificações para NPCs:

- Utilitário: Entrega item para o jogador;
- Missão: Passa uma tarefa para o jogador em troca de alguma recompensa;
- Figurante: tem falas, mas nenhum uso.

- Alice

FIGURA 6 - Personagem Alice



FONTE: Autoria própria (2024).

Nome : Alice;

Função no jogo : Missão;

História : Alice é uma aluna do 3º ano, ela iria ajudar o jogador a entender as mecânicas e um pouco da história;

Falas:

- Ei! que tal atacar aquela coisinha verde com essa espada? Te dou um negócio bem legal !
- Você conseguiu! Boa!
- Como vai, [André/Sofia]?

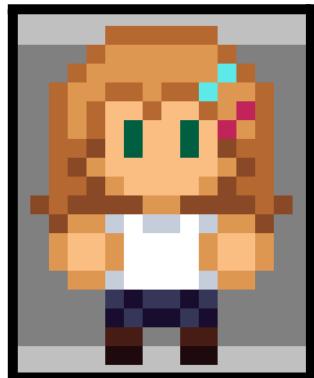
Filiação(aliado, inimigo, neutro) : Aliado;

Missão: básicas para explorar as mecânicas básicas do jogo;

Item: espada.

- **Laura**

FIGURA 7 - Personagem Laura



FONTE: Autoria própria (2024).

Nome : Laura Barreto;

Função no jogo : Missão;

História : Laura adora Balas, e comeu tantas que se tornou um doce de pessoa;

Falas:

- Você pode recuperar meu saco de balas pra mim? Eu perdi ele na floresta mas estou com medo de ir lá de novo...
- Você encontrou? Obrigada!
- Oi! Quer uma bala?

Filiação(aliado, inimigo, neutro) : neutro;

Missão: Encontrar o saco de doces perdido;

Recompensa: Copo de café.

- **Gabriela**

FIGURA 8 - Personagem Gabriela



FONTE: Autoria própria (2024).

Nome : Gabriela ;

Função no jogo : Utilitário;

História : Primeiro ano, gosta de estudar e tem muitos amigos;

Falas:

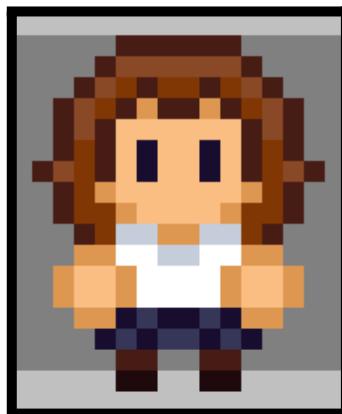
- Você é o aluno novo? Toma, vai te ajudar um pouco!
- Oi! O que tem feito?

Filiação(aliado, inimigo, neutro) : neutro;

Item: Pudim;

- **Ronalda**

FIGURA 9 - Personagem Ronalda



FONTE: Autoria própria (2024).

Nome : Ronalda;

Função no jogo : npc de quest;

História : não é muito popular apesar dos seus esforços constantes. Gosta de estudar toxinas e faz experimentos com elas;

Falas:

- Ei cara... Você pode me dar uma ajudinha? Eu precisava de umas 15 gosmas pra um experimento... Claro que pago bem!

Valeu! Ei... que tipo de lixo você me deu? Isso deu completamente errado! Toma isso!

- *Resmungos* estragou tudo...

Filiação(aliado, inimigo, neutro) : neutro;

Missão: Coletar 15 gosmas matando gosmas(Monstro);

Recompensa: pudim.

- Otto

FIGURA 10 - Personagem Otto



FONTE: Autoria própria (2024).

Nome : Otto;

Função no jogo : Figurante;

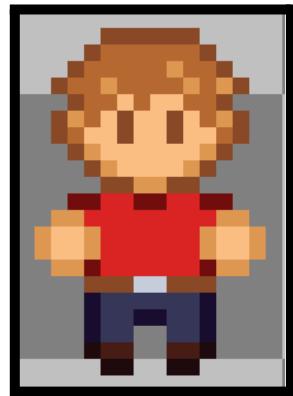
História : Premiado após ser o primeiro a conseguir completar a demo 2.1 do jogo durante a mostra de cursos do CEFET-MG;

Falas:

- Dê o seu 100%! Dificuldades abrem oportunidades! Não desista cara!

- **Bruno**

FIGURA 11 - Personagem Bruno



FONTE: Autoria própria (2024).

Nome : Bruno;

Função no jogo : Figurante;

História : Premiado após ser o primeiro a conseguir completar a primeira demo do jogo;

Falas:

- Só queria um pudim...

3.2.2. Inimigos

Os inimigos compõem uma parte essencial da mecânica do jogo, formando os componentes do combate. A intenção é que cada inimigo tenha mecânicas de combate e movimento únicos entre os outros. Dessa maneira, foram criados alguns monstros, baseados em animais, mitos populares e raças dos jogos de "Dungeons and Dragons" (WIZARDS OF THE COAST, 2014).

- **Slime**

FIGURA 12 - Imagem do Slime



FONTE: Autoria própria (2024).

Essas geleias habitam frequentemente espaços abandonados onde há presença de água bastante suja e parada. São considerados os monstros mais fracos que existem. Podem ter variações em sua coloração, variando entre verde, azul e vermelho.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Segue fixamente o jogador;

Habitat: Todos.

- Pássaro Kroka

FIGURA 13 - Imagem do Pássaro Kroka



FONTE: Autoria própria (2024).

Pássaro Kroka se tornam agressivos quando sentem que seu território está sendo ameaçado por seres humanos. São criaturas inteligentes, que usam armas de guerreiros derrotados para atacar seus inimigos.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Segue o jogador até certo ponto, se mantendo próximo ao seu ninho;

Habitat: Leste do Câmpus.

- **Caveira Flamejante**

FIGURA 14 - Imagem da Caveira Flamejante



FONTE: Autoria própria (2024).

Restos mortais de fantasmas que não conseguiram escapar, são amaldiçoados a vagar pelo local de sua morte, atacando sem consciência. Porém, matá-los, ou melhor, livrá-los da maldição, os ajuda a seguir para o pós-morte, dando ao jogador um efeito positivo chamado “Gratidão das Almas”, que lhe recupera 50% da vida.

Tipo de Ataque: Distância;

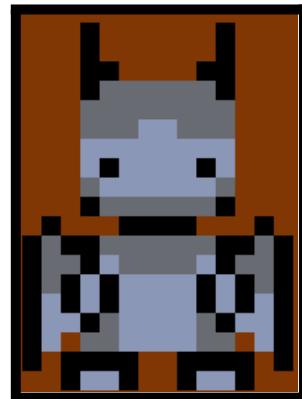
Item dropado: Ouro;

Comportamento: Mantém-se a uma distância do jogador e o golpeia com bolas de fogo. Consegue atravessar objetos;

Habitat: Subsolo.

- **Gárgula**

FIGURA 15- Imagem da Gárgula



FONTE: Autoria própria (2024).

Ninguém sabe exatamente como eles surgiram, porém acreditam que são esculturas antigas feitas por artistas incríveis que acabaram criando vida, contagiados pela magia do ambiente. São bem bizarros.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Segue o jogador lentamente, e possui um ataque que empurra levemente para frente. Não pode ser empurrado;

Habitat: Subsolo.

- **Torrão**

FIGURA 16 - Imagem do Torrão



FONTE: Autoria própria (2024).

As plantas do jardim se conectam com a magia do local, formando tufos de grama e terra ambulantes. Contaminadas e perdidas, sem saber o que está acontecendo de fato, elas apenas procuram se proteger dos seres humanos, já que as únicas vagas memórias que possuem, são deles cortando suas “pontinhas”.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Segue o jogador, sua velocidade para bater e fugir é impressionante;

Habitat: Leste do Câmpus.

- **Abelha Operária**

FIGURA 17 - Imagem da Abelha Operária



FONTE: Autoria própria (2024).

Afinal, como está em seu nome, seu propósito de vida é apenas trabalhar. Porém, não pense em incomodar o serviço delas, pois elas podem demonstrar que são mais do que só trabalhadores esforçados.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Neutro, até serem atacadas. Elas batem e fogem rápido, porém o principal é que chamam suas irmãs abelhas soldados;

Habitat: Oeste do Câmpus.

- **Abelha Soldado**

FIGURA 18 - Imagem da Abelha Soldado



FONTE: Autoria própria (2024).

“Eu devo proteger a colmeia!”. As abelhas soldados sempre foram maiores que as outras abelhas, por isso foram designadas a proteger a colmeia e sua população. Sempre que uma irmã é atacada, honram seu dever e correm para protegê-la.

Tipo de Ataque: Corpo a corpo;

Item dropado: Ouro;

Comportamento: Segue o jogador, priorizando ficar perto de sua rainha e proteger as abelhas operárias. Empurra o jogador longe;

Habitat: Oeste do Câmpus.

3.2.3. Inimigos Especiais

Os inimigos especiais são mais fortes que os mais comuns, sendo chefões únicos com comportamentos de ataque e movimento diferentes. O jogo apresentará um monstro especial por área, uma vez que eles são de produção complexa e são os inimigos mais difíceis do jogo.

- **Abelha Rainha**

FIGURA 19 - Imagem da Abelha Rainha



FONTE: Autoria própria (2024).

Rainha das Abelhas, sempre rodeada de soldados, é a líder suprema da colmeia, soberana da colônia.

Tipo de Ataque: Curto e longo alcance;

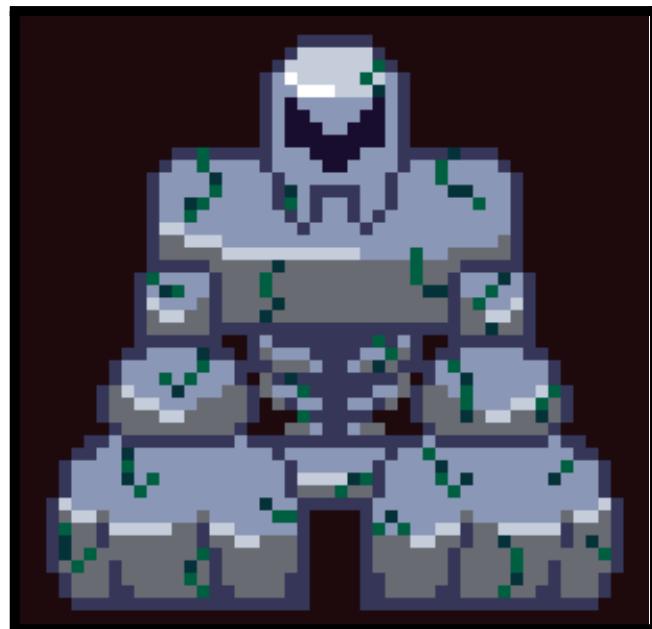
Item dropado: Coroa de favos;

Comportamento: Ataca o jogador com golpes físicos e dispara ferrões no jogador, invoca Abelhas Soldado;

Habitat: Floresta Rosa.

- **Golem guardião**

FIGURA 20 - Imagem do Golem guardião



FONTE: Autoria própria (2024).

Guardião da caverna, há muito tempo esquecido. Ninguém sabe mais o que ele guarda, pois todos que entraram, nunca saíram de seu domínio.

Tipo de Ataque: Curto e longo alcance;

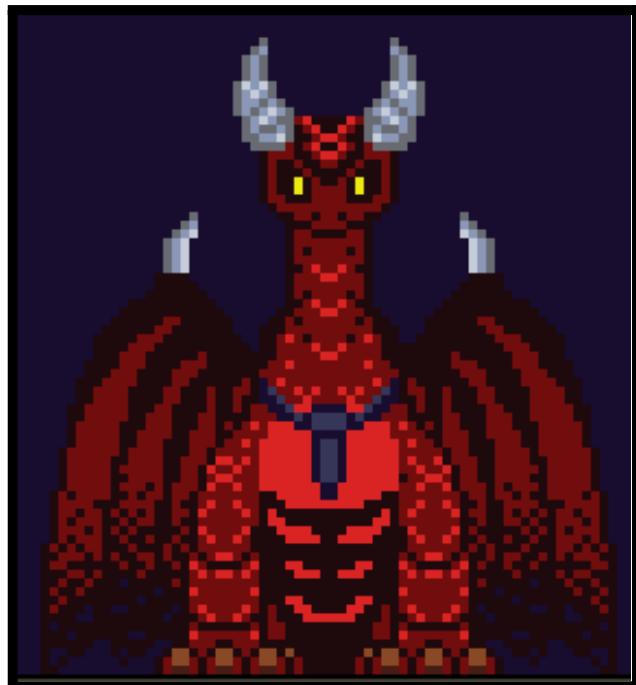
Item dropado: Núcleo do Golem;

Comportamento: Ataca o jogador com golpes físicos, de grande intensidade, em movimentos lentos e atira pedras no jogador;

Habitat: Caverna.

- O Diretor

FIGURA 21 - Imagem do Diretor



FONTE: Autoria própria (2024).

O Diretor da escola, o chefe dos monstros e o responsável pelo caos no campus.

Tipo de Ataque: Curto e longo Alcance;

Item dropado: Crânio do dragão;

Comportamento: Ataca o jogador com suas garras e dispara bolas de fogo, invoca Caveiras Flamejantes para ajudá-lo na batalha;

Habitat: Subsolo.

3.2.4. Itens

Os itens são responsáveis por trazer mais dinamicidade ao jogo, aumentando o número de ações que podem ser realizadas em combate, ou servindo como itens chave, que desbloqueia certas áreas e recursos dentro do jogo, eles melhoram a experiência do jogo, trazendo mais variedade de recursos e estimulando a criatividade do jogador.

O item de cura por exemplo, é representado por uma xícara de café, que por padrão tem uma carga e vai aumentando com o passar do jogo, as cargas são recuperadas ao salvar o jogo, no ponto de *spawn*, elas tem inspiração no item frasco de Estus da franquia Dark Souls, que funciona de maneira semelhante. Os demais itens seguem o mesmo princípio, porém realizando suas respectivas funções, como desbloquear mecânicas do jogo.

- Café

FIGURA 22 - Café



FONTE: Autoria própria (2025).

Nome: Caneca de café;

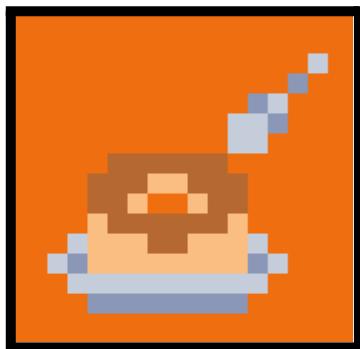
Descrição: O santo suco dos estudantes, te permite lutar mais um pouco quando as coisas ficam difíceis ;

Efeito: cura a vida do jogador;

Fonte: Entregue por Laura Sabelo.

- **Pudim**

FIGURA 23 - Pudim



FONTE: Autoria própria (2025).

Nome: Prato de pudim ;

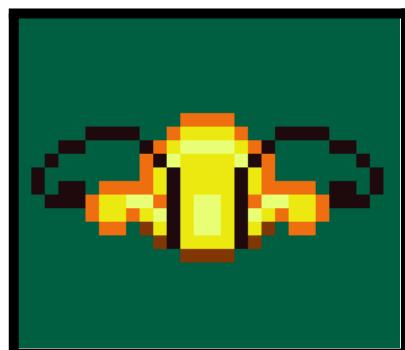
Descrição: Uma sobremesa lendária;

Efeito: Seu grande nível de açúcar aumenta permanentemente a velocidade do jogador;

Fonte: Entregue por Gabriela e Ronaldinha.

- **Coroa de favos**

FIGURA 24 - Coroa da rainha abelha



FONTE: Autoria própria (2025).

Nome: Coroa de Favos;

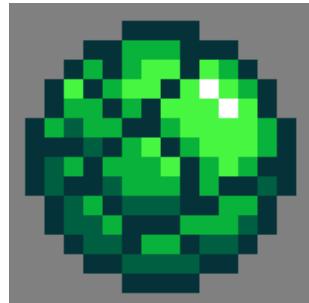
Descrição: Coroa uma vez usada pela vossa magestade, ainda contém alguns feromônios;

Efeito: Recupera a vida do jogador a cada inimigo derrotado;

Fonte: Derrotar a Abelha Rainha .

- Núcleo do Golem

FIGURA 25 - Núcleo do Golem



FONTE: Autoria própria (2025).

Nome: Núcleo do Golem (Prêmio);

Descrição: Dizem que ao segurá-lo, o usuário sofre um forte aumento em sua força, permitindo empurrar pedras e liberar áreas antes inacessíveis;

Efeito: Permite empurrar pedras;

Fonte: Derrotar O Golem.

- Crânio de dragão

FIGURA 26 - Crânio de dragão



FONTE: Autoria própria (2025).

Nome: Crânio de Dragão (Prêmio);

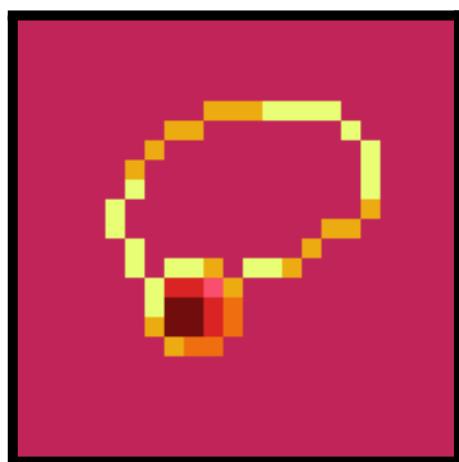
Descrição: O portador emana um odor heroico;

Efeito: Nenhum;

Fonte: Derrotar O Diretor.

- **Amuleto do Piromante**

FIGURA 27 - Amuleto do Piromante



FONTE: Autoria própria (2025).

Nome: Amuleto;

Descrição: Perdido por um aluno piromante chamado Laércio;

Efeito: Desbloqueia o ataque especial da liberação das chamas;

Fonte: Entregue pelo Mochileiro.

3.3. Estruturação de código

A estruturação do jogo teve como princípio a adoção de práticas de “código limpo” para garantir que a manutenção e a expansão do projeto fossem realizadas de forma eficiente e colaborativa. Inspirados nos conceitos apresentados por Robert C. Martin e abordados no artigo CÓDIGO LIMPO: a importância da refatoração (2022), a equipe buscou seguir um conjunto de boas práticas que visam não apenas a funcionalidade do código, mas também sua legibilidade e simplicidade, pautados nos pilares da orientação à objetos.

3.3.1. Pilares da Orientação à objetos

Ao criar o Código-fonte do jogo, o grupo teve por objetivo explorar ao máximo, a programação orientada a objetos, ou POO, estrutura comumente utilizada em linguagens de programação de alto nível como o caso da linguagem C#, utilizada no jogo. A POO se baseia em quatro pilares fundamentais: encapsulamento, abstração, herança e polimorfismo, que permitem criar um código mais modular, reutilizável e de fácil manutenção, evitando duplicidades desnecessárias de trechos do código, embora nem sempre seja possível, é aconselhável que isso ocorra.

- **Encapsulamento:** O encapsulamento é o princípio que consiste em agrupar dados e comportamentos relacionados em uma classe. Ele oculta os detalhes de implementação internos, permitindo que o acesso aos dados seja feito apenas por meio de métodos específicos, chamados de *getters* e *setters*, garantindo maior controle e segurança;
- **Abstração:** A abstração é o princípio de ocultar os detalhes internos de implementação de um objeto, expondo apenas os aspectos essenciais e relevantes para o uso, simplificando a interação com o sistema;
- **Herança:** A herança permite que uma classe herde características (atributos) e comportamentos (métodos) de outra classe, criando uma hierarquia e possibilitando a reutilização de código e a generalização de conceito;
- **Polimorfismo:** O polimorfismo é o princípio que permite que diferentes objetos, pertencentes a diferentes classes, respondam a uma mesma

mensagem ou chamada de método de maneiras diferentes, aumentando a flexibilidade e a extensibilidade do sistema.

FIGURA 28 - Pilares da Programação orientada a objetos



FONTE:<https://pt.linkedin.com/pulse/programa%C3%A7%C3%A3o-orientada-objetos-diego-rocha>

3.3.2. Nomes significativos

A escolha de nomes significativos para variáveis, métodos, classes e outros elementos do código é essencial para a legibilidade e manutenção do projeto. Nomes bem definidos facilitam a compreensão do propósito de cada parte do código, mesmo para membros que não participaram diretamente do seu desenvolvimento. Durante o projeto, evitamos abreviações confusas ou genéricas, optando por descrições mais claras e objetivas que seguem convenções de nomenclatura, como o padrão *PascalCase* para nomes de classes e *camelCase* para métodos e variáveis. Essa prática contribuiu para uma melhor comunicação entre a equipe e reduziu o tempo gasto para localizar e entender trechos de código.

3.3.3. Tratamento de erros e testes constantes

O tratamento de erros foi uma prioridade para evitar falhas inesperadas que poderiam comprometer a experiência do jogador. Utilizamos estruturas de *try-catch* para capturar exceções e tratá-las adequadamente, apresentando mensagens de

erro úteis para a depuração. Além disso, realizamos testes a cada nova alteração no algoritmo, impedindo que erros fossem acumulados e que pudessem comprometer o jogo.

3.3.4. Padrões de projeto aplicados

Os padrões de projeto desempenharam um papel fundamental na organização do código, viabilizando projetos maiores e mais escaláveis. Alguns dos padrões aplicados foram:

- **Singleton:** Utilizado para gerenciar instâncias únicas de classes, como o controlador de jogo (*GameManager*), garantindo que exista apenas uma instância durante a execução do jogo;
- **Observer:** Implementado para criar sistemas de eventos que notificam os objetos sobre mudanças de estado, como a interação entre o *input* do jogador e as ações que ele realiza, por exemplo;
- **Factory:** Aplicado para criar instâncias de personagens e itens de forma dinâmica, permitindo maior controle sobre o processo de inicialização e simplificando a inclusão de novos elementos no futuro;
- **State Machine:** Incorporado de forma nativa na engine no sistema de animações para gerenciar diferentes animações de um personagem, como "andar", "correr", "atacar" ou "morrer", com base no estado atual do personagem. Isso promove uma separação lógica dos comportamentos e evita a criação de códigos com alta complexidade ciclomática, isto é, uma quantidade excessiva de ramificações nas estruturas lógicas que dificultam o entendimento do código.

3.3.5. Refatoração Contínua

A refatoração contínua é uma prática central durante o desenvolvimento de qualquer projeto. Sempre que o código apresentar níveis desnecessários de complexidade e acoplamento devem ser realizadas melhorias que tornem o sistema mais claro e eficiente, como refazer sistemas e mecânicas inteiras. Por meio deste processo foi possível reduzir a ocorrência de *bugs*, aumentar a qualidade do código

e possibilitar a incorporação de novas funcionalidades de forma rápida e segura, sem afetar outras partes do código.

3.3.6. Versionamento do código

O versionamento do código foi uma prática essencial para o desenvolvimento do projeto, e o uso do *GitHub* teve um papel muito importante na organização e colaboração da equipe. As funcionalidades que permitiram isso foram:

- **Ramificações no código:** A criação de uma *branch* dedicada para cada nova funcionalidade, correção de bug ou melhoria foi fundamental para evitar conflitos entre os desenvolvedores e garantir que as alterações não interferissem nas funcionalidades já implementadas;
- **Resolução de conflitos:** Em alguns casos, quando duas *branches* modificavam o mesmo trecho de código, surgiam conflitos que precisavam ser resolvidos antes de realizar a integração ao código principal. O *GitHub* forneceu ferramentas intuitivas para identificar e resolver esses conflitos, evitando a perda de trabalho e garantindo que todas as modificações fossem integradas corretamente;
- **Propostas de alteração:** A ferramenta de *pull requests* no *GitHub* foi essencial para a revisão de código. Antes de mesclar qualquer *branch* ao código principal, foram realizadas análises para verificar a qualidade das alterações, identificar possíveis problemas e garantir que as novas funcionalidades estivessem alinhadas aos requisitos do projeto.

3.4. Mecânicas do jogo

Neste tópico, serão descritas algumas das estratégias utilizadas na programação do jogo, permitindo uma compreensão básica do funcionamento das principais mecânicas implementadas. Serão abordados aspectos como a lógica por trás dos sistemas de interação, movimento, ataque, inteligência artificial dos inimigos e outros elementos essenciais ou estéticos. Além disso, serão apresentadas as referências utilizadas no desenvolvimento, que guiaram o processo de criação. Dessa forma, oferecendo um panorama detalhado das soluções aplicadas no projeto.

3.4.1. Programação do player

A programação do jogador no jogo é organizada de forma modular, ou seja, o personagem principal é representado por um objeto (como um *GameObject* na Unity) que recebe diferentes *scripts* como componentes. Cada um desses scripts tem a função de gerenciar uma ação ou comportamento específico do personagem, tornando o código mais organizado e fácil de gerenciar. Esse sistema é semelhante ao funcionamento da composição na programação orientada a objetos (POO), ou seja, a classe final, ou nesse caso, o objeto *player*, é formada por várias outras classes menores, cada uma resolvendo um problema específico. Esse tipo de estruturamento é necessário ao produzir jogos na medida em que permite a escalabilidade deles, em caso de expansão futura.

Esses códigos funcionam de forma independente, mas interagem entre si, seguindo o princípio da responsabilidade única, conceito também abordado por Robert C. Martin em *Clean Code: A Handbook of Agile Software Craftsmanship* (2009). Dessa forma, cada classe realiza uma tarefa específica. Isso facilita a manutenção do código, pois, se for necessário modificar uma parte do comportamento do jogador, você pode editar ou substituir apenas o código responsável por aquela função, sem afetar outras partes do jogo. Além disso, permite que novos comportamentos sejam adicionados ao jogador de forma simples, bastando apenas adicionar uma nova classe ao objeto do personagem.

3.4.2. Sistema de Ataque

Um sistema de ataque em um jogo é responsável por gerenciar como o personagem interage com inimigos ou outros objetos ao realizar ações ofensivas. Ele envolve a execução de comandos, como a ativação de animações, a detecção de colisões e a aplicação de efeitos no alvo, como redução de vida ou destruição. O sistema de ataque precisa ser eficiente, preciso e expansível, permitindo que novas mecânicas ou tipos de ataques sejam facilmente integrados ao jogo.

Esse sistema interage com duas partes principais, o objeto *Player* e seu objeto filho *ActiveWeapon* (arma ativa). O *Player* contém um *script* chamado *PlayerAttack.cs* (ataque do player), responsável por gerenciar os ataques do jogador. O *script* mantém uma referência ao objeto *ActiveWeapon*, que controla qual arma está equipada no momento. Os ataques são realizados por meio de animações, sincronizadas com eventos específicos. Durante a execução da animação de ataque, um objeto com uma área de colisão (*hitbox*) é ativado por um curto período de tempo, geralmente em um único frame, caso a arma atual seja uma de longo alcance, um projétil com a colisão será instanciado.

Quando essa *hitbox* colide com um objeto que pode sofrer dano, como inimigos ou objetos destrutíveis, o sistema verifica as condições e aplica uma redução na vida ou integridade do alvo, seguindo as regras definidas para cada tipo de interação. Esse processo é essencial para garantir precisão e impacto nos ataques, ao mesmo tempo em que preserva o desempenho do jogo, ativando a colisão apenas nos momentos necessários. O sistema também é projetado para ser expansível, permitindo a adição de diferentes tipos de armas, cada uma com comportamentos e animações específicas, sem a necessidade de alterar significativamente a lógica central do ataque.

FIGURA 29 - Ataque



FONTE: Autoria própria (2024).

3.4.3. Sistema de input

A recepção de comandos do jogador utiliza o sistema de Input da Unity, que gera um asset configurável com as informações das teclas e os eventos associados a cada uma. Para gerenciar esse sistema, um *ScriptableObject* (objeto programável) chamado *InputReader.cs*, (leitor de entrada) responsável por gerenciar a subscrição dos eventos de entrada foi criado. Isso permite que qualquer código que precise acessar os comandos do jogador ou ativar/desativar o mapa de ações o faça facilmente, apenas recebendo como parâmetro uma instância do *InputReader*.

3.4.4. Sistema de vida

A vida do jogador é um dos aspectos fundamentais que determinam sua sobrevivência no jogo. À medida que o jogador interage com o ambiente ou enfrenta inimigos, sua vida pode ser reduzida por danos, o que exige gestão cuidadosa para evitar a morte e continuar a jornada.

O script *PlayerLife.cs* (vida do player) é responsável por gerenciar a vida do jogador, controlando o valor atual de saúde e realizando as ações necessárias quando o jogador sofre dano. O jogador possui um objeto contendo uma *hitbox* com um colisor, que é responsável por detectar colisões com objetos que infligem dano. Quando o colisor da *hitbox* entra em contato com um inimigo ou objeto destrutível, o sistema verifica a colisão e, se o objeto causador do dano for válido, a vida do jogador é reduzida.

Além disso, para manter o HUD (*Heads-Up Display*) atualizado, o sistema usa um padrão de design *Observer* para notificar outros componentes do jogo sobre as mudanças na vida. O script *HealthBar.cs* (barra de saúde) se inscreve para escutar os eventos gerados pelo *PlayerLife.cs*, recebendo notificações sempre que há uma alteração na saúde do jogador. Com isso, a barra de vida do jogador é atualizada em tempo real, refletindo a perda de vida de forma visual, para que o jogador possa monitorar seu estado durante o jogo. Esse sistema garante que a visualização das alterações na vida seja imediata e eficiente, proporcionando uma experiência mais interativa e imersiva.

3.4.5. Sistema de movimentação

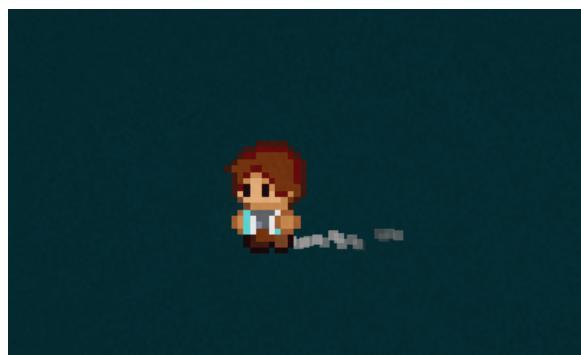
O sistema de movimentação é implementado no script *PlayerMovement.cs* (movimento do player), que atua como o núcleo responsável por processar os comandos recebidos do sistema de *input* e traduzi-los em ações no jogo. Esse script gerencia a movimentação do jogador de forma eficiente, garantindo que os comandos sejam interpretados corretamente e aplicados em tempo real, proporcionando uma experiência de controle fluida e responsiva.

- **Movimentação:** A movimentação do jogo é realizada em oito direções, pois funciona com base em um vetor de direção, que é definida com base nas teclas pressionadas pelo player, por exemplo caso as teclas pressionadas sejam, W(cima) e D(direita) essa ação irá resultar em um deslocamento para a diagonal nordeste, além disso este vetor é normalizado utilizando a biblioteca de matemática da Unity, garantindo que sua magnitude seja sempre 1, tal fato ocorre para impedir que o jogador se movimente mais rápido nas diagonais, após a definição da direção, enquanto o evento é ativado, o jogo toca a animação de caminhada do *player* na direção correspondente ao movimento, utilizando uma *blend tree*, que recebe o vetor de direção citado anteriormente;
- **Stamina:** A *stamina*, assim como a vida, é dividida em duas partes, o código principal “*PlayerStamina.cs*”(resistência do player), que gerencia a quantidade de *stamina* disponível para o player e é instanciada no script da movimentação através da técnica de composição, e o código que gerencia a

barra de *stamina* "StaminaBar.cs"(barra de stamina), que mostra a quantidade disponível na tela;

- **Corrida:** O sistema de corrida verifica se o *player* não está realizando alguma ação que o impossibilite de correr, e caso ele esteja apto a correr, reduz os pontos de *stamina* do *player*, ao mesmo tempo que aumenta sua velocidade;

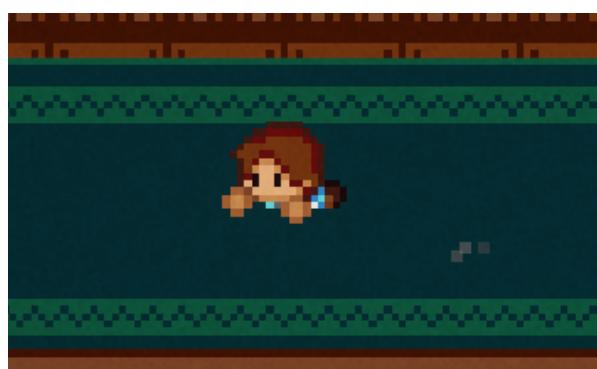
FIGURA 30 - Corrida



FONTE: Autoria própria (2024).

- **Rolamento/Esquiva:** O rolamento ocorre em um *IEnumerable*, tipo de função usada para realizar corrotinas, ações em segundo plano no código que neste caso são usadas para, executar a animação do rolamento, desativar os comandos do *player*, e desativar sua *hitbox*, para impedir que ele toma dano ao rolar. O rolamento consome um quarto da *stamina* disponível do *player* e só pode ser executado caso ele tenha mais que essa quantidade.

FIGURA 31 - Esquiva



FONTE: Autoria própria (2024).

3.4.6. Interação e objetos interagíveis

O sistema de interação utiliza eventos para detectar quando o jogador interage com objetos, o *player* contém um *script* chamado *PlayerInteraction.cs* (interações do player), que recebe o *input* da tecla usada para a interação do *player*, e quando essa tecla é acionada dispara um evento de interação, e os próprios objetos interagíveis verificam se o player está em seu alcance e se estiver disparam sua função interagir, presente na interface *IInteractable.cs* (interagível), os objetos têm um *script* próprio filho de *InteractableObject.cs*, que por sua vez implementa a interface *IInteractable.cs*, e eles tem como filho a instância de um *prefab*, que contém o código *Interactor.cs*, que notifica o código específico do objeto interagível que a interação pode ocorrer.

- **NPCs:** O código específico dos *NPCs* é implementado em *NPCInteraction.cs*, caso o player possa interagir, dispara um caixa de diálogo correspondente ao diálogo do *NPC*, armazenado em *scriptable objects*;
- **Baú:** O baú, diferente dos *NPCs*, dispara uma caixa de novo item, dá um item ao player e reproduz uma animação de abertura;
- **Porta:** A porta apenas reproduz uma animação de abertura e permite que o player passe por ela.

3.4.7. HUD

O HUD (*Heads-Up Display*) é a interface visual do jogo que fornece informações cruciais ao jogador de maneira clara e imediata durante a jogabilidade. Ele é projetado para manter o jogador informado sobre o status do personagem, recursos disponíveis e outros elementos essenciais sem interromper a imersão ou a ação do jogo.

FIGURA 32 - HUD



FONTE: Autoria própria (2024).

- **Stamina:** Barra verde que representa a energia do personagem, geralmente usada para ações como correr, esquivar ou realizar ações especiais. Um indicador de stamina permite ao jogador gerenciar seus recursos para evitar ficar vulnerável durante o combate ou exploração.
- **Vida:** Barra vermelha que exibe o estado de saúde do personagem. Uma barra ou associada à vida permite ao jogador monitorar o dano sofrido e ajustar sua estratégia, como buscar itens de cura ou adotar uma postura defensiva.
- **Ícone do personagem:** Esfera à esquerda do HUD que contém uma representação visual do personagem controlado.
- **Poder especial:** Esfera à direita do HUD que indica a disponibilidade e o progresso de habilidades especiais, como a bola de fogo.
- **Esferas:** Pequenas esferas azuis que indicam a disponibilidade das esferas, que geram pequenas explosões afastando os inimigos.

3.4.8. Mapas

O mapa do jogo foi realizado com a utilização de diversas técnicas de renderização e ferramentas disponibilizadas pela própria *engine*, que facilitam o fluxo de trabalho e tornam a produção e otimização do jogo mais eficientes, dentre elas ocorrem a utilização de:

- **Tilemap:** Os *Tilemaps* são sistemas da Unity projetados para criar e gerenciar mapas 2D de forma eficiente. Eles permitem o uso de *tiles*, que são pequenos *sprites* reutilizáveis, para compor cenários e ambientes do jogo. O sistema

oferece ferramentas que facilitam o posicionamento e a edição de *tiles* diretamente no editor. Além disso, os *Tilemaps* podem ser integrados com componentes como colisores e sistemas de iluminação, permitindo que os mapas não apenas representam visualmente o ambiente, mas também influenciam a jogabilidade, como definir áreas acessíveis ou barreiras físicas;

- **Auto Tiling:** para automatizar o processo de produção de novos mapas, o jogo utiliza uma ferramenta de *auto-tiling* disponível no acervo da Unity conhecida como *rule tiles*, que permite que conjuntos de regras sejam criados para definir a posição de *sprites* do *tileset* com base na posições já existentes, evitando a construção 100% manual do mapa;
- **Cenas:** No funcionamento da Unity, o projeto é organizado em cenas, que representam ambientes individuais contendo instâncias específicas de objetos. Cada cena funciona como um espaço isolado onde os elementos do jogo, como personagens, cenários e interações, são configurados e gerenciados;
- **Transição entre mapas:** A mudança dos mapas ocorre substituindo com a colisão do player em um objeto que contém o *script* de transição de mapas, em suma, este *script* recebe a cena ao qual o player deve ser transferido e o seu ponto de *spawn* em alguma cena, com isso, o player aparecerá na posição correta na próxima localidade, além disso, o jogo contém uma lista de objetos que são preservados ao carregar uma nova cena, pois estão contidos na região *DontDestroyOnLoad* (não destruído no carregamento), nativa da Unity, que impede que os objetos sejam destruídos ao trocar de cena.

3.4.9. Efeitos de pós-processamento

Os efeitos de pós-processamento são ferramentas amplamente utilizadas para melhorar a estética e a imersão visual de um jogo. Na Unity, esses efeitos são aplicados por meio de um sistema de pós-processamento que utiliza componentes chamados *Global Volumes* para definir áreas onde os efeitos são ativados ou desativados. Um *Global Volume* é um componente que controla os efeitos de pós-processamento em uma cena. Ele funciona como um recipiente para um perfil de pós-processamento, que contém as configurações dos efeitos visuais aplicados à câmera. Além do *Global Volume*, é possível criar volumes que se aplicam a áreas

específicas, permitindo transições suaves de efeitos baseados na posição do jogador.

Algumas dos efeitos de pós processamento utilizados no jogo foram:

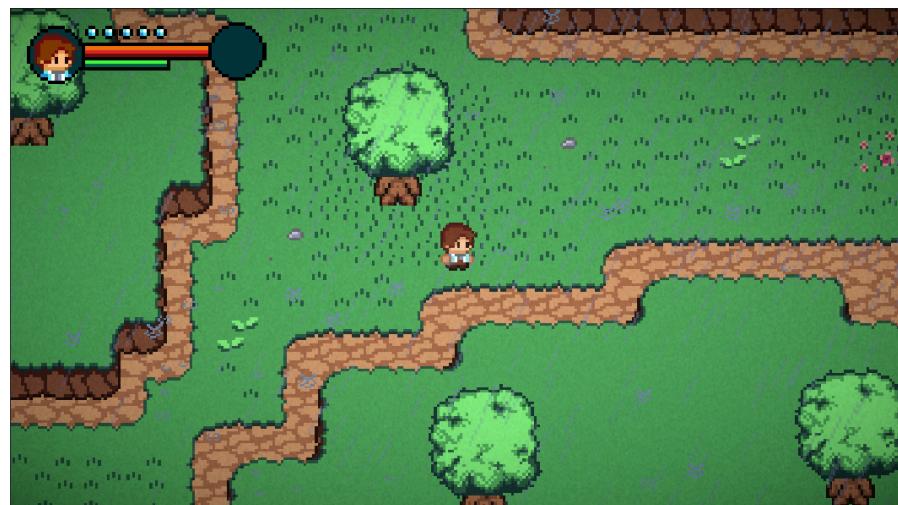
- **Correção de cor:** A correção de cor ajusta as tonalidades, saturação e brilho da cena, permitindo criar atmosferas específicas, como um tom quente para cenários aconchegantes ou tons frios para situações tensas;
- **Bloom:** Bloom adiciona um brilho difuso ao redor de áreas muito iluminadas, simulando o comportamento natural da luz ao atingir a retina ou lentes de câmera;
- **Vinheta:** A vinheta escurece as bordas da tela, direcionando o foco do jogador para o centro. É um efeito útil para criar atmosferas mais íntimas ou sombrias;
- **Chromatic Aberration:** Este efeito simula a separação das cores (vermelho, verde e azul) nas bordas da tela, como acontece em lentes de câmeras reais;
- **Film Grain:** A granulação de filme adiciona uma textura de ruído sutil na tela, simulando a aparência de filmes analógicos antigos, pode ser usada para enriquecer a atmosfera e uma cena sem grandes esforços.

3.4.10. Partículas

Um sistema de partículas no Unity é uma ferramenta usada para criar efeitos visuais dinâmicos, como fogo, fumaça, chuva, explosões ou faíscas. Ele funciona emitindo e manipulando pequenos *sprites* ou objetos chamados partículas, que são controlados por propriedades como velocidade, cor, escala, e tempo de vida. Exemplos de partículas de ambientação presentes no jogo:

- **Chuva:** As partículas de chuva utilizam dois sistemas de partículas combinados, um sistema principal responsável por renderizar os pingos, e um emissor, executado ao final do ciclo de vida das partículas ele gera um impacto com o solo após o pingo cair;

FIGURA 33 -Partículas de chuva



FONTE: Autoria própria (2024).

- **Pétalas:** As pétalas são simples, e apenas se deslocam para a direita, variando apenas a velocidade, tempo de vida e rotação das partículas;

FIGURA 34 -Partículas de Pétalas



FONTE: Autoria própria (2024).

- **Poeira:** As partículas de poeira são as mais menos complexas, apenas *pixels* que variam levemente sua posição e velocidade de modo que pareçam estar apenas flutuando devagar;

FIGURA 35 - Partículas de Poeira



FONTE: Autoria própria (2024).

3.4.11. Diálogo

Um sistema de diálogo em um jogo é uma mecânica que permite a interação entre personagens por meio de conversas, escolhas e respostas. Ele é essencial para a narrativa, ajudando a desenvolver a trama, os personagens e as relações entre eles

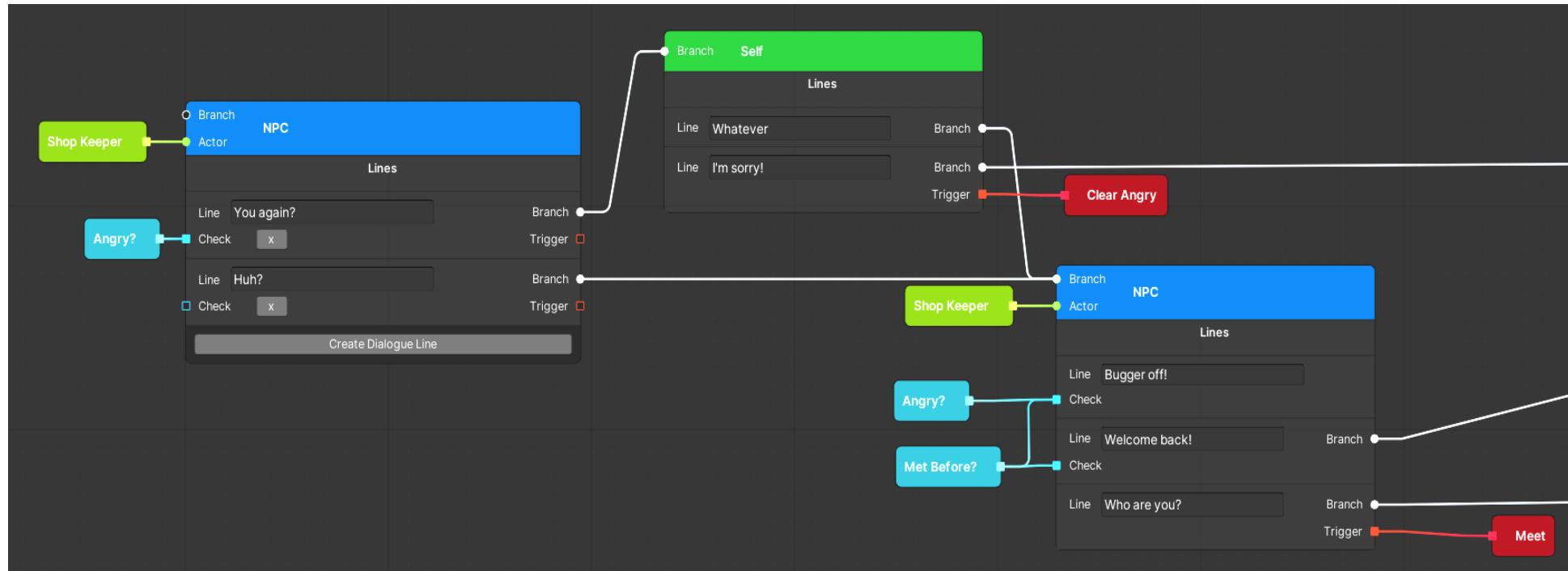
FIGURA 36 - Diálogo dentro do jogo



FONTE: Autoria própria (2025).

Com o fim de facilitar a implementação de tal sistema no jogo foi utilizado o *Dialogue Graph* (gráfico de diálogo), um pacote para Unity que permite criar conversações através de nós e estruturas lógicas, permitindo tanto diálogos sequenciais quanto diálogos ramificados dependendo de condições específicas, devido a sua versatilidade a ferramenta se encaixa em diversos tipos de projetos, além disso sua estrutura de gráficos facilita a criação destes e permite armazená-los de maneira modular com o uso de *ScriptableObjects*. O sistema foi criado por Teodor Vecerdi e está disponível em um repositório público no GitHub (VECERDI, 2019).

QUADRO 3 - Sistema de Diálogo

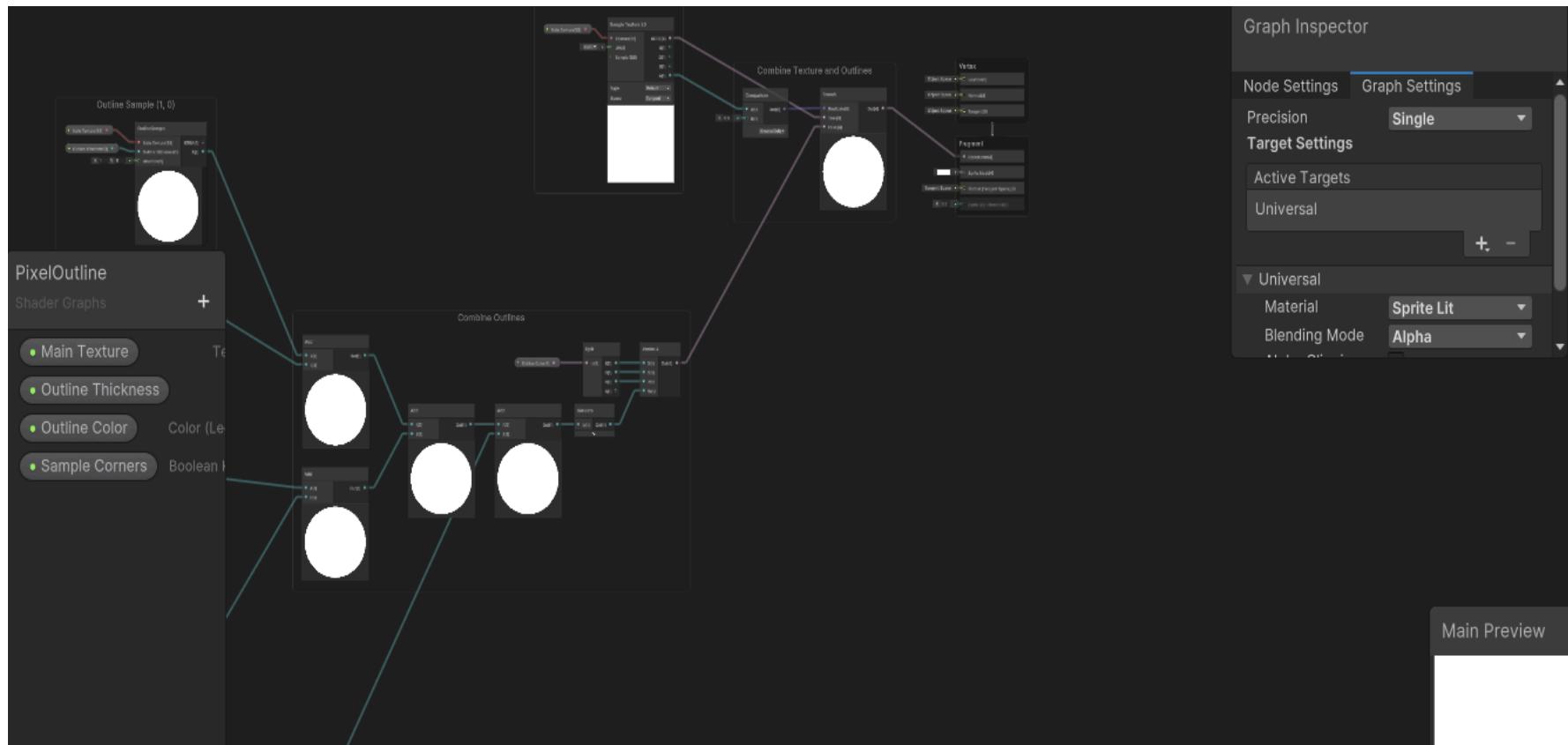


FONTE: <https://github.com/TeodorVecerdi/DialogueGraph>

3.4.12. Shaders

Shaders são classes que controlam como os gráficos de um objeto são processados e renderizados na *GPU*, definindo efeitos visuais como cores, texturas, iluminação e transparência. Na Unity, eles são usados em materiais para definir a aparência de objetos, permitindo criar desde superfícies simples até efeitos avançados, como reflexos, sombras dinâmicas e partículas realistas. Além disso na Unity estes *scripts* podem ser criados através de gráficos, que facilitam a criação e visualização dos *shaders*, como no exemplo abaixo:

QUADRO 4 - Árvore de Animações da Unity



FONTE: Autoria própria (2024).

- **HitEffect:** Dispara um brilho branco, usado em objetos que sofreram algum dano, para dar mais imersão ao jogo.

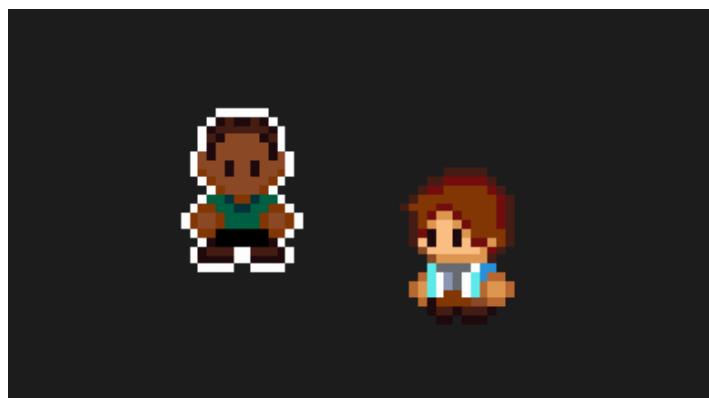
FIGURA 37 - Animação de dano



FONTE: Autoria própria (2024).

- **Pixel outline:** cria uma linha externa de *pixels* para indicar que o jogador está no alcance de algum objeto interagível. A versão utilizada no jogo foi feita com base no vídeo “Give Your Sprites Outlines in Unity Shader Graph”, do canal Daniel Ilett.

FIGURA 38 - Outline de item Interagível



FONTE: Autoria própria (2024).

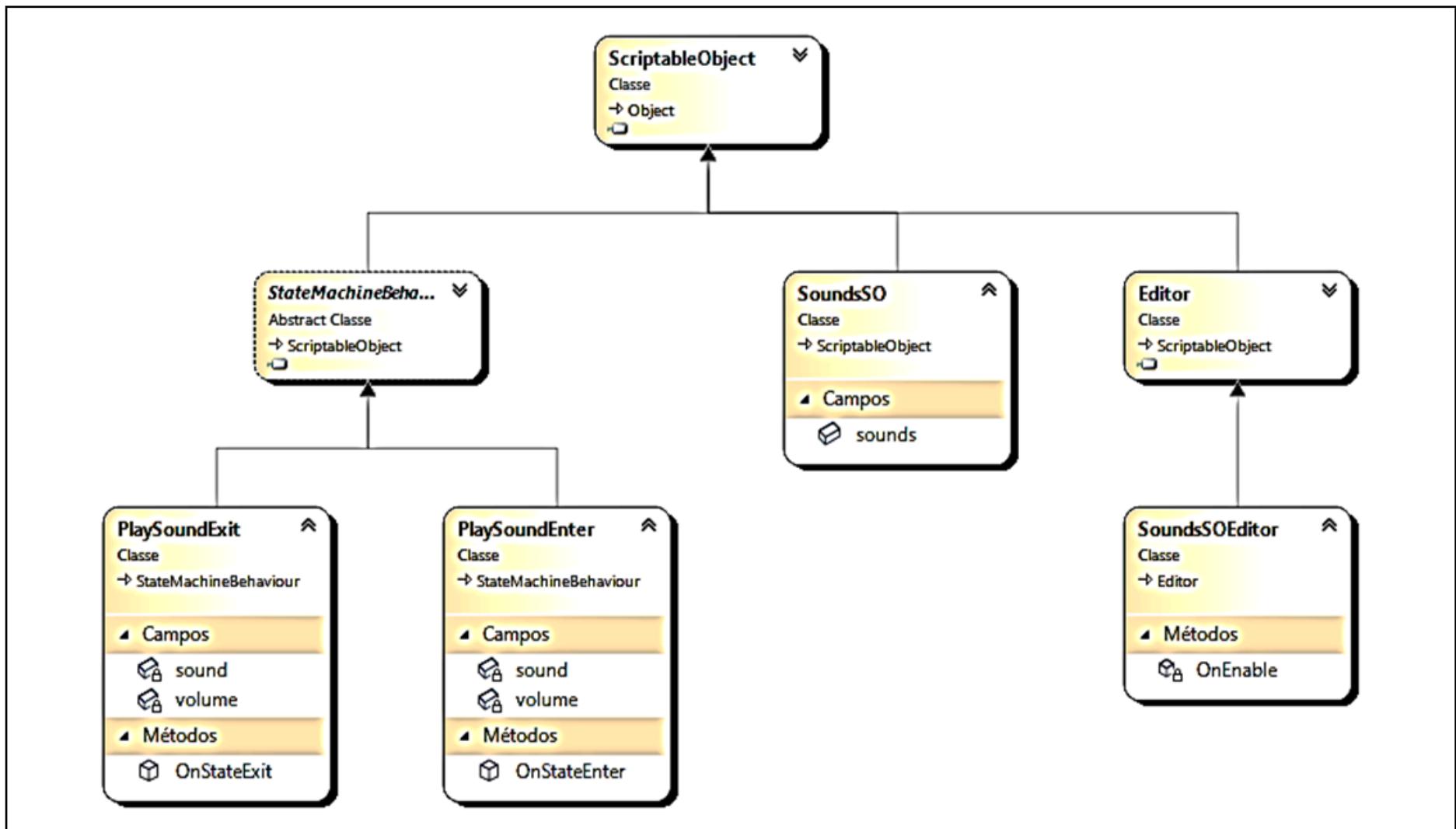
3.4.13. Sistema de sons

O sistema de sonorização do jogo foi desenvolvido com base no sistema apresentado pelo canal Small Hedge Games, em um vídeo de seu canal e disponível em seu repositório público no GitHub. Esse sistema utiliza um objeto de instância única (*Singleton*) para gerenciar os sons do jogo de forma centralizada. No núcleo do sistema, há um *Scriptable Object* que recebe a definição de um *Enum*

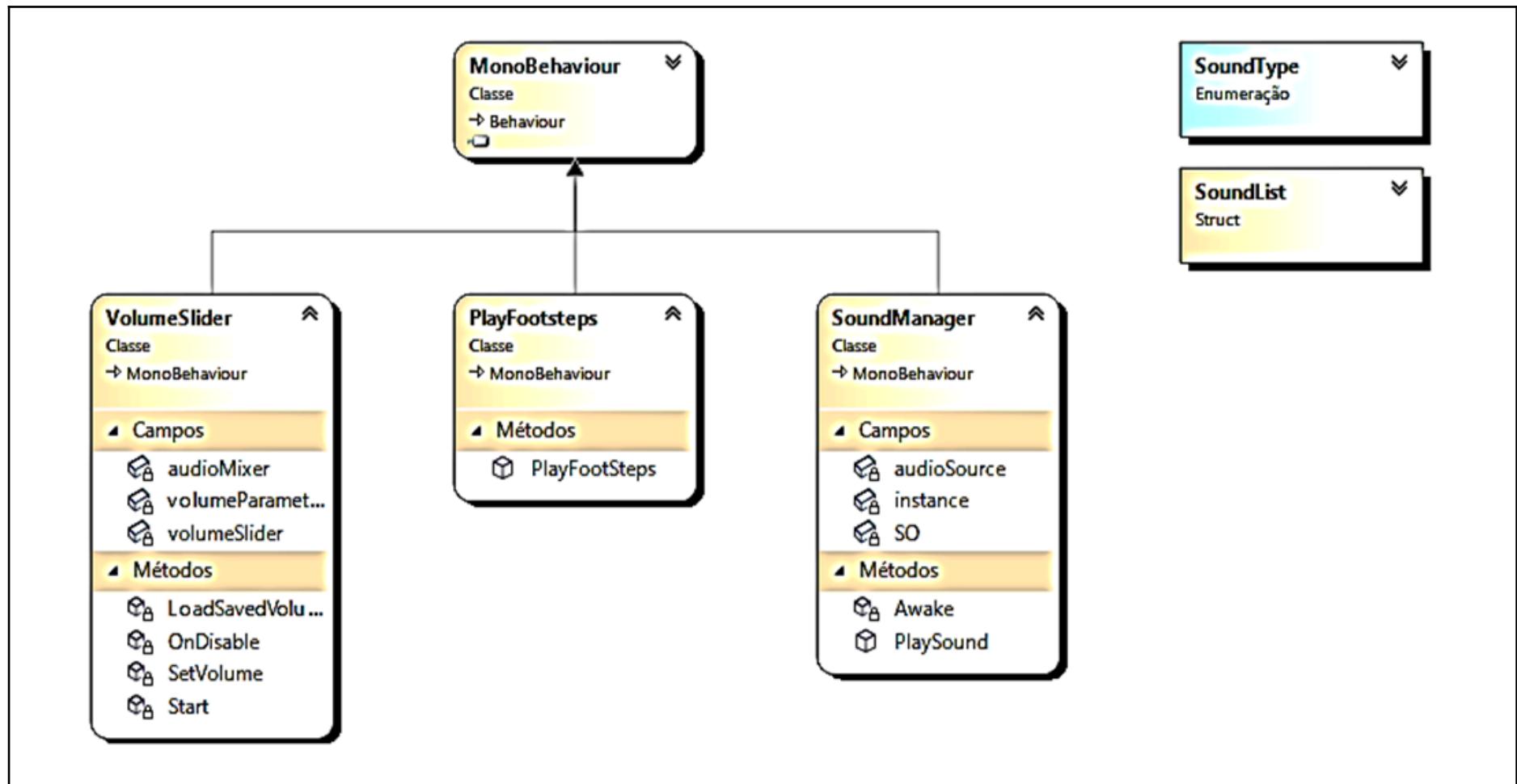
(uma estrutura que categoriza valores fixos e nomeados) representando diferentes tipos de sons. Cada item do *Enum* é associado a uma lista de sons, permitindo que esses sejam organizados e acessados de forma eficiente. Os sons são reproduzidos de forma dinâmica por meio de *scripts* em momentos específicos do jogo, como em eventos de animação ou interações do jogador. Por exemplo, efeitos sonoros podem ser disparados durante transições de animações, utilizando os *Animation Events* (eventos de animação) para sincronizar sons de passos ou ações específicas, ou diretamente através de *scripts* para outros contextos, como interações e retornos sonoros gerais.

QUADRO 5 - Diagrama de Classes do Sistema de Sons

(continua)



(conclusão)



FONTE: Autoria própria (2024).

3.4.14. Game Feel

O termo *Game Feel* é o nome dado à aplicação de respostas sensoriais que aprimoram a experiência do jogador, aumentando a imersão e a satisfação durante o *gameplay*. Essas sensações podem ser visuais, auditivas ou táteis, e são utilizadas para tornar as interações no jogo mais dinâmicas e envolventes.

No desenvolvimento do jogo, algumas técnicas de *Game Feel* implementadas foram:

- Retorno Visual:

1. **Screen Shake:** Efeito de tremor na câmera para enfatizar ações impactantes, como ataques ou explosões, no caso do jogo a classe é uma referência estática que pode ser convocada em qualquer lugar do código;
2. **Flash de Dano:** Breve mudança de cor ou brilho em inimigos e personagens ao serem atingidos, indicando claramente que sofreram dano;
3. **Efeitos de Partículas:** Utilização de partículas para representar eventos como ataques, magias, ou destruição de objetos.

- Feedback Auditivo:

1. **Efeitos Sonoros Sincronizados:** Sons precisos para ações específicas, como passos, ataques, interações ou transições, reforçando as conexões com o mundo do jogo.
2. **Variações de Som:** Alterações sutis no pitch ou volume para evitar repetição excessiva de efeitos sonoros e manter a experiência dinâmica.

3.4.15. Persistência de dados

O sistema de salvamento serve para manter informações que não podem ser perdidas ao fechar o jogo, como o player escolhido ou seus itens, por exemplo. Esse sistema de salvamento do jogo consiste em uma classe *PlayerData.cs* (dados do player) armazena as informações do jogador, como o nome e se ele pegou a espada

ou não. Quando o jogo começa, o script *PlayerLoader.cs* (carregador do player) carrega esses dados usando o *SaveManager.cs* (gerenciador de salvamento).

O *SaveManager.cs* é uma classe estática, o que significa que não precisa ser instânciada para ser utilizada e é responsável por salvar e carregar os dados. O método *SavePlayerData()* recebe um objeto *PlayerData.cs*, atualiza os dados existentes e os salva em um arquivo JSON. O método *LoadPlayerData()* tenta ler esse arquivo e, se encontrado, converte o conteúdo para um objeto *PlayerData.cs*. Se o arquivo não for encontrado, ele retorna um valor nulo. O caminho do arquivo é gerado pela função *GetSaveFilePath()*, que define onde o arquivo de save será armazenado de forma persistente no dispositivo, permitindo que o progresso do jogador seja mantido entre sessões de jogo.

3.4.16. Comportamento dos inimigos

No jogo, os inimigos possuem sua lógica separada em módulos distintos para movimentação, ataque e vida, garantindo uma estrutura de código mais organizada e modular. A movimentação utiliza o sistema de *NavMesh* em conjunto com o plugin *NavMeshPlus*, que permite suporte a malhas de navegação em modo 2D, adequando-se ao estilo do jogo. A inteligência artificial dos inimigos utiliza um algoritmo de *pathfinding* conhecido como *A** ou *AStar*, que é responsável por calcular o caminho mais curto entre dois pontos dentro da malha de navegação, contornando obstáculos definidos por modificadores de terreno, como barreiras ou obstáculos. Essa abordagem garante que o comportamento de navegação seja eficiente e responsivo, proporcionando interações melhores dos inimigos com o cenário.

- **Curto alcance:** Os inimigos de curto alcance são aqueles que precisam se aproximar ou entrar em contato direto com o jogador para executar seus ataques. Um exemplo dessa categoria é o slime, que inflige dano ao encostar diretamente no jogador;

FIGURA 39 - Inimigos de curto alcance



FONTE: Autoria própria (2025).

- **Longo Alcance:** Os inimigos de longo alcance são aqueles que se utilizam de projéteis para causar dano ao jogador, quando chegam a uma distância pré estabelecida do jogador eles param de se aproximar;

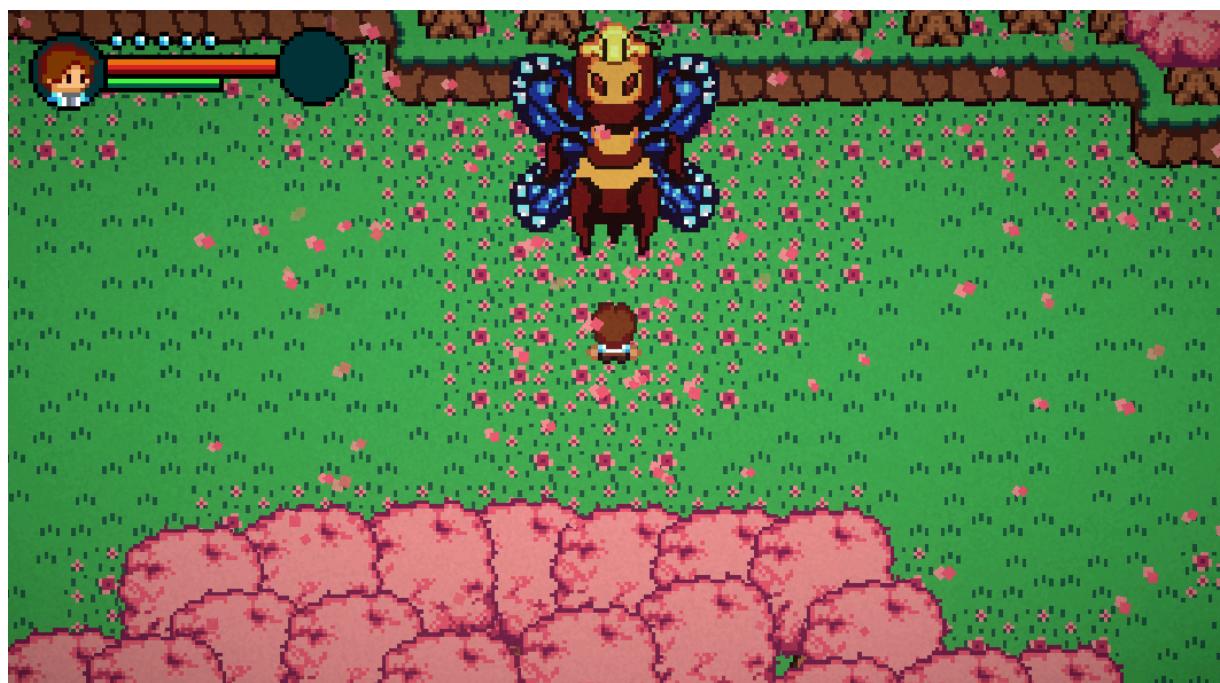
FIGURA 40 - Inimigos de longo alcance



FONTE: Autoria própria (2025).

- **Chefes:** Ao contrário dos inimigos tradicionais, os chefes não utilizam inteligência artificial para controlar seus ataques e movimentos. Em vez disso, as batalhas contra eles são roteirizadas, seguindo padrões pré definidos que proporcionam uma experiência mais imersiva para o jogador, reduzindo a dependência da sorte e aumentando a necessidade de habilidade do jogador.

FIGURA 41 - Chefe - Abelha Rainha



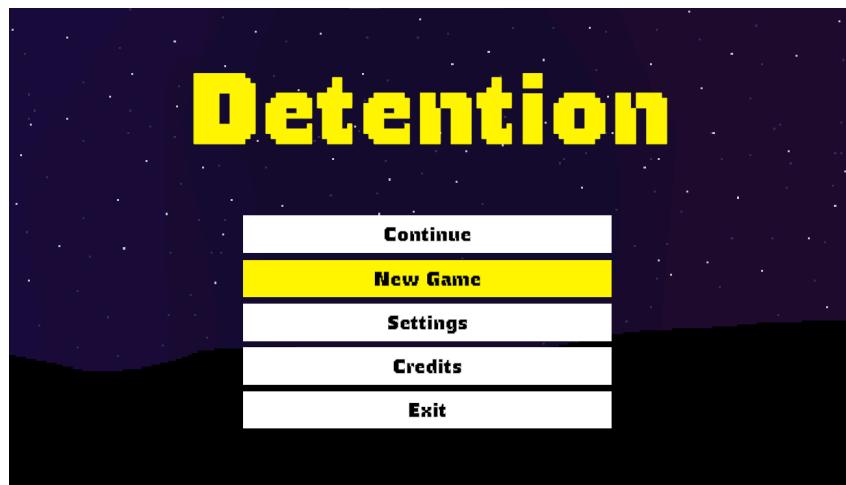
FONTE: Autoria própria (2025).

3.4.17. Menus e UI

Os menus interativos garantem ao jogador o controle da aplicação, é por meio deles que o usuário recebe uma interface clara e intuitiva para acessar funcionalidades importantes. Dessa forma, o jogador pode navegar entre diferentes opções, personalizar configurações de acordo com suas preferências e interagir diretamente com os sistemas do jogo.

- **Menu principal:** O Menu principal do jogo conta com 5 botões principais, ele oferece a opção de começar um jogo novo, continuar um jogo que já existe, acessar as opções de configurações do jogo, ver os créditos do jogo e fechar a aplicação;

FIGURA 42 - Menu principal



FONTE: Autoria própria (2025).

- **Menu de pausa:** O menu de pausa é uma versão reduzida do menu principal, feita para ser acessada a qualquer momento, oferecendo uma gama mais limitada de configurações, apenas para ajustar coisas simples que podem ser alteradas durante a *gameplay*.

FIGURA 43 - Menu de pausa



FONTE: Autoria própria (2025).

3.4.18. Objetivos e Missões

A progressão da gameplay e da história ocorre por meio de objetivos, armazenados nos dados salvos do jogo, a cada novo objetivo cumprido, o jogo registra e apresenta ao usuário seu próximo objetivo. Esse sistema não apenas evita que o jogador se perca, mas também enriquece sua experiência, aumentando o envolvimento com o universo da obra, o que torna a jornada mais satisfatória e divertida. Além disso, a cada objetivo concluído ocorre alguma alteração no mundo do jogo, fazendo com que o cenário mude levemente ao longo do tempo.

FIGURA 44 - Ícone das missões

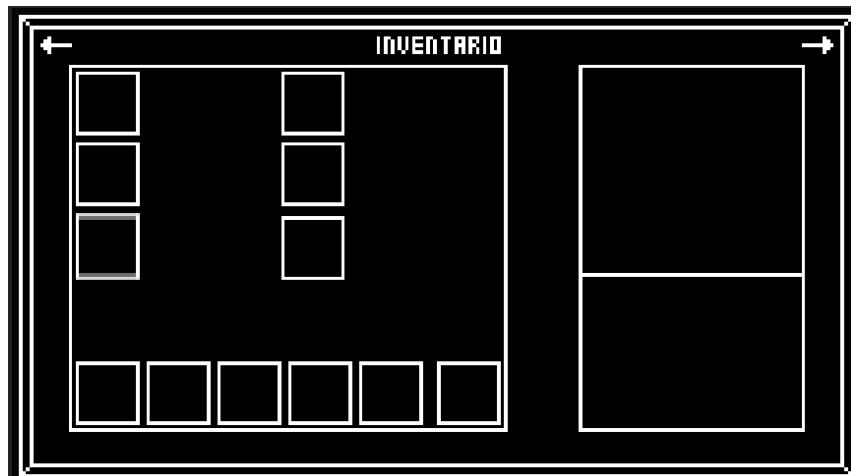


FONTE: Autoria própria (2025).

3.4.19. Itens

Os itens são organizados de forma fixa em vez de serem armazenados em um inventário tradicional. Quando um item é recebido, ele é associado a um *Scriptable Object* (objeto programável) que contém as informações necessárias, como o ícone e as propriedades do item. Esses itens, ao serem adquiridos, são alocados em variáveis booleanas e aparecem em um local pré-determinado no inventário do jogador. Esse tipo de organização é útil para controlar de forma mais contida a progressão no jogo, liberando determinados itens e funcionalidades após eventos específicos. Dessa forma o inventário não necessita de alterações dinâmicas, tornando a programação mais simples e leve.

FIGURA 45 - Inventário Vazio



FONTE: Autoria própria (2025).

4. RESULTADOS

Como resultado, tivemos um jogo funcional que atendia a todos os objetivos iniciais. Por meio desse projeto, foi possível validar com sucesso as ferramentas selecionadas como viáveis para desenvolver um jogo *indie* no Brasil. Além disso, tivemos avaliações de diversos estudantes, sendo esses, alunos visitantes durante a Mostra de Cursos do CEFET-MG, os alunos do CEFET-MG durante a 33º META e a avaliação dos nossos colegas convidados . Sobre isso, obtivemos dados ao entrevistá-los e registramos o retorno, adaptando as sugestões pertinentes.

Além disso, pudemos reunir recomendações de programadores experientes na área de jogos, permitindo reunir as partes mais cruciais para o sucesso de um projeto pequeno. As informações reunidas nas entrevistas permitiram ao grupo ter um olhar crítico em relação aos problemas com o projeto, que sofreu uma importante reestruturação após essas conversas.

Com isso, formamos uma comunidade ativa que acompanhou o jogo desde seus estágios mais iniciais de desenvolvimento e que aguarda o lançamento final com grandes expectativas. Além disso, o engajamento dos estudantes nos permitiu perceber nossas falhas e aprimorar nosso trabalho, comprovando as recomendações dos nossos desenvolvedores convidados como muito úteis e pertinentes em relação à primeira versão do jogo. Graças a essas opiniões, foi possível melhorar a qualidade do produto final. Os dados completos do questionário estão no apêndice C. Dentre as mensagens que recebemos, as mais relevantes foram:

“O design e arte do jogo está muito linda mas o chão por uma questão de cor não combina muito com as paredes do jogo além disso o personagem está meio estranho principalmente na parte das mãos que parecem só duas bolas sem conexão isso é mais evidente quando o personagem está de lado ou de costas” - Daniel Pacheco Monteiro

“O jogo em uma masmorra é meio padrão no gênero RPG, poderia ter uma variação maior nos cenários” - Arthur Maximus

“A ambientação é muito bonita, mas falta elementos do cenário (tochas e ossos espalhados por exemplo) o mapa também tem áreas bem memoráveis como a sala com 20 slimes de uma vez, a sala da cruz e a sala com os 4 pilares, além disso o mapa se sai muito bem no quesito de ser um labirinto, além disso acredito que o mapa devia tomar uma direção mais catacumbas de paris onde há áreas do mapa iguais as que tem agora e passagens estreitas onde não há pedras nas paredes apenas terra” - Daniel Pacheco Monteiro

5. CONSIDERAÇÕES FINAIS

A presente pesquisa baseou-se na análise dos desafios enfrentados no desenvolvimento de jogos *indie* no Brasil, destacando aspectos como dificuldades financeiras, qualificação de mão de obra, acesso a ferramentas e inserção no mercado. Como estudo de caso, foi desenvolvido um jogo, cuja produção serviu para validar as barreiras identificadas e testar as estratégias e ferramentas sugeridas por desenvolvedores *indie* brasileiros entrevistados durante o processo de pesquisa.

A partir de um referencial teórico baseado em Amélio(2018), Andrade(2015), Ramos e Miranda (2024), Santos (2023), e com apoio de relatórios da ABRAGAMES (2022) e IGDA (2020), foi possível estruturar um conjunto de estratégias aplicáveis ao contexto brasileiro. Essas estratégias foram incorporadas ao processo de criação do jogo estudado, desde o planejamento inicial até o lançamento.

O desenvolvimento do jogo teve como premissa a utilização de ferramentas acessíveis e gratuitas, como Unity e Aseprite, e a criação de uma comunidade fiel de jogadores, por meio do Discord e do Whatsapp. O uso dessas ferramentas poderosas e acessíveis permitiu que a equipe otimizasse recursos e enfrentasse os desafios de tempo e orçamento limitados. Durante o processo, os principais desafios encontrados incluíram a ausência de financiamento para o projeto, a necessidade de qualificação técnica da equipe e a complexidade de equilibrar o trabalho em grupo com os estudos e a vida social dos integrantes. Contudo, com o auxílio dos orientadores, a análise das opções e do diálogo, foi possível contorná-los e chegar a um resultado satisfatório dentro do que foi objetivado.

Por fim, a criação do jogo como produto final da pesquisa mostrou-se um experimento bem-sucedido de aplicação prática das estratégias levantadas. Além de validar as propostas teóricas, o trabalho contribui com um recurso útil para desenvolvedores *indie* brasileiros, fornecendo uma gama de ferramentas e experiência que podem ser usadas para superar os desafios do setor. Dessa maneira, podemos concluir que o nosso trabalho atingiu o seu principal objetivo, dentro dos limites de nossa realidade .

6. REFERÊNCIAS

ABRAGAMES. Relatório Anual 2022: Panorama do Desenvolvimento de Jogos Digitais no Brasil. Associação Brasileira das Desenvolvedoras de Jogos Digitais, 2022.

ANDRADE, Antônio. **Game engines: a survey.** Porto: Virtual Campus Lda., 2015. Acesso em: 9 jan. 2025.

AMÉLIO, Camila de Oliveira. **A Indústria e o Mercado de Jogos Digitais no Brasil.** Belo Horizonte: Universidade Federal de Minas Gerais, 2018.

BGS. **Estudo sobre o Mercado de Jogos no Brasil 2021.** Brazilian Games Show, 2021.

BRASIL. Lei nº 14.852, de 03 de maio de 2024. Cria o marco legal para a indústria de jogos eletrônicos; e altera as Leis nºs 8.313, de 23 de dezembro de 1991, 8.685, de 20 de julho de 1993, e 9.279, de 14 de maio de 1996. **Diário Oficial da União:** seção 1, Brasília, DF, n. 86, 06 de maio de 2024.

COLE, H.; GRIFFITHS, M. D. **Social interactions in massively multiplayer online role-playing gamers.** *Cyberpsychology & Behavior*, v. 10, n. 4, p. 575-583, 2007.

GAMELAB HQ. **Jogos indie: a crescente cena dos desenvolvedores independentes.** [S. l.]: Gamelab HQ, 2023.

GAMES, Small Hedge. **UNITY SOUND MANAGER TUTORIAL.** Disponível em: <https://www.youtube.com/watch?v=q5WT91Sn3hg&t=543s>. Acesso em: 4 jan. 2025.

ILLET, Daniel. **PLEASE use a Unity SOUND MANAGER! - Full Tutorial.** Disponível em: <https://www.youtube.com/watch?v=84rZ-rCRsZk>. Acesso em: 4 jan. 2025.

INTERNATIONAL GAME DEVELOPERS ASSOCIATION (IGDA). **Developer satisfaction survey 2020.** Disponível em: <https://www.igda.org/2020-survey/>. Acesso em: 28 dez. 2024.

MACHADO, Ronney. **Um estudo sobre a história dos jogos eletrônicos.** Acesso em: 3 dez. 2024.

MARQUES, Lucas M. et al. **Escaping through virtual gaming—what is the association with emotional, social, and mental health? A systematic review.** São Paulo: Faculdade de Medicina, Instituto de Medicina Física e Reabilitação, Hospital das Clínicas HCFMUSP, Universidade de São Paulo, 2023.

MARTIN, Robert C. **Código limpo: habilidades práticas do Agile Software.** Tradução de Camila Helena Nazareth Aguiar. São Paulo: Alta Books, 2022.

OBEDKOV, Evgeny. **Games market revenue to surpass \$330 billion in 2028, with mobile gaming accounting for over 75% of total.** [S. l.]: Game World Observer, 2024.

PRICEWATERHOUSECOOPERS (PwC). **Global Entertainment & Media Outlook 2024–2028.** PwC, 2024. Disponível em: <https://pwc.com>. Acesso em: 3 dez. 2024.

PRZYBYLSKI, A. K. et al. **The ideal self at play: The appeal of video games that let you be all you can be.** Psychological Science, v. 21, n. 4, p. 531-537, 2010.

RAMOS, L.; MIRANDA, M. **Desafios do Desenvolvimento Indie no Brasil.** Rio de Janeiro: Editora GameLab, 2024.

REIS, Leoncio José de Almeida; CAVICHIOLLI, Fernando Renato. **Dos Single aos Multiplayers.** Licere, Belo Horizonte, v. 17, n. 2, p. 1-20, jun. 2014.

SANTOS, J. **Marketing Digital para Jogos Indie.** São Paulo: TechPress, 2023.

SETZER, P. **Financiamento Colaborativo e Jogos Digitais**. Belo Horizonte: IndieBooks, 2023.

SOUNDMANAGER. **Sound Manager**. Disponível em:
<https://github.com/SmallHedge/SoundManager>. Acesso em: 4 jan. 2025.

SOARES, W. **Crie Seus Jogos: Guia Prático para Desenvolvedores**. Disponível em: <https://youtube.com/crieseusjogos>. Acesso em: 4 jan. 2025.

STEAM. **Stardew Valley**. Disponível em:
https://store.steampowered.com/app/413150/Stardew_Valley/. Acesso em: 26 dez. 2024.

STEAM SPY. **Steam game sales and player data 2020**. Disponível em:
<https://steamspy.com>. Acesso em: 28 dez. 2024.

VECERDI, Teodor. **DIALOGUE GRAPH**. Disponível em:
<https://github.com/TeodorVecerdi/DialogueGraph>. Acesso em: 4 jan. 2025.

VOIDA, A.; CARROLL, J. M. **The iPad for the family: stories of interaction**. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2013. p. 1257-1266.

WAKKA, Wagner. **Mercado de games agora vale mais que indústrias de música e cinema juntas**. Canaltech, 2021. Disponível em:
<https://canaltech.com.br/games/mercado-de-games-agora-vale-mais-que-industrias-de-musica-e-cinema-juntas-179455/>. Acesso em: 9 ago. 2024.

WIZARDS OF THE COAST. **Manual dos monstros**. 5. ed. Renton, WA: Wizards of the Coast, 2014.

7. APÊNDICES

7.1 APÊNDICE A - *Entrevista com Wennes Soares*

Data: 16 /10/ 2024

Q: QCG ESTUDIOS | W: WENNES SOARES

Canal: COMO CRIAR UM JOGO | CRIE SEUS JOGOS

Jogo: Economize 29% em Cyberwar: Neon City no Steam

Site: Crie Seus Jogos - O seu portal sobre desenvolvimento de jogos.

Sobre o convidado: Wennes Soares é o fundador do portal e canal Crie Seus Jogos, bem como é co-fundador do Cyber Monkey Studios. Desenvolvedor de jogos e professor de cursos de programação, além de ter se consagrado na área dos jogos por meio do “CyberWar Neon City”, que já ganhou prêmios dedicados a jogos *indie* em escala internacional.

Q: Como você começou sua entrada no mercado de desenvolvimento de jogos?

W: É... perfeito. Eu comecei nesse mercado depois que eu descobri que existia desenvolvedores de game *indie*, mas especificamente eu descobri a existência do RPG Maker, então eu comecei a estudar o RPG Maker há uns 11 anos atrás, e nisso aí comecei a tomar gosto pela ferramenta, pelo trabalho de desenvolver jogos, mas a partir daquele momento, como eu era muito jovem, não tinha levado aquilo como uma decisão de uma profissão. Então, fui fazendo várias outras coisas na vida até que eventualmente pensei que eu deveria voltar para a área de desenvolvimento de jogos porque foi uma área que sempre gostei, desde quando mexia no RPG Maker, então minha entrada nesse mercado foi meio controversa, eu comecei a mexer com o desenvolvimento de jogos, mas larguei por muitos anos, e só depois eu voltei, só que quando voltei foi para valer e comecei a desenvolver os jogos usando a Unity, que acho que vocês devem conhecer. E dali para frente, foi a engine que eu resolvi me aprimorar e desenvolver meus jogos, desenvolver o meu conteúdo e o CyberWar, no qual fui o único programador do jogo.

Q: O que exatamente te inspirou a criar o canal Crie Seus Jogos?

W: Perfeito. Foi a falta de conteúdo na área em português, porque quando eu comecei a estudar, eu não encontrava tutorials, assim hoje em dia você encontra vários, mas quando comecei não tinha Instagram, TikTok, tinha nada disso, então a internet era quase um “mato”, e a gente tinha o Youtube e alguns blogs, alguns fóruns também, mas esses vídeos no Youtube eram bem básicos, porque a internet ainda tava começando e principalmente a falta de conteúdo em português, já que tudo que encontrava era em inglês. Então falei, pô vou criar um blog no qual vou falar sobre o desenvolvimento de jogos só que em português, aí comecei a criar artigos e o site crieseusjogos.com.br (Crie Seus Jogos - O seu portal sobre desenvolvimento de jogos.) foi a primeira coisa que criei, e lá comecei a escrever artigos de tutoriais, artigos mais técnicos e até alguns falando da minha própria experiência. Com o tempo, aquilo foi evoluindo até que chegou num ponto que falei “pô deveria ter um canal também, já que tenho um blog e escrevo”.

Q: Você já enfrentou alguma dificuldade que te deu vontade de desistir do projeto ou do canal?

W: Sim, primeiramente do canal a vergonha mesmo. Eu tinha muita vergonha, então várias vezes quis desistir, mas tem uma coisa que pegou que foi o seguinte, quando foi passando o tempo consegui uma consistência de não sentir mais vergonha, não se sentir mais intimidado diante da câmera, então a coisa foi fluindo mais. Em relação a carreira em si, a maior dificuldade foi no início, onde eu tive que tirar um tempo para criar um portfólio sem receber nada por aquilo, sabe? Porque eu fiquei praticamente um ano inteiro só trabalhando para construir um portfólio e sem receber nada por isso. Mas no final valeu a pena, já que consegui o meu primeiro emprego a partir desse projeto que trabalhei por mais ou menos um ano. Então acho que essas foram as duas maiores dificuldades que encontrei que me fizeram pensar em desistir do projeto e do canal.

Q: Como surgiu a ideia do jogo CyberWar Neon City?

W: Então, na verdade a ideia base do jogo surgiu de um brainstorm de um possível Planeta dos Macacos CyberPunk. E a partir dali começamos a fazer o jogo,

e com essas referências, cara isso daí é tudo liberdade criativa do nosso diretor de arte, que é bem “doido da cabeça”, ele é muito fã de muitos animes, como One Piece, aí ele acaba misturando um monte de coisa e referência, e One Piece é meio assim né, muita coisa maluca, meio nada haver, aí acaba que ele leva a sério demais e fica essa maluquice toda de referências no jogo. Então basicamente é isso.

Q: Quais foram as maiores influências para o desenvolvimento do jogo?

W: Planeta dos Macacos igual eu falei, e Cyberpunk 2077, mas não desse jogo que conhecemos hoje, mas sim de um teaser que teve anos atrás. E na época eu lembro que eu falei que até no final do desenvolvimento desse jogo, o CyberPunk já teria sido lançado e se ele tivesse um impacto pelo menos como uma parte ali do GTA, a gente já estaria com um tema muito quente e a gente vai lançar um jogo com um tema que vai estar ali na galera. Então essas duas temáticas foram as maiores influências do jogo. Mas fora isso teve várias outras, a que eu acho mais legal é a jornada do herói, que a gente utilizou bastante do conceito do livro para montar a história do personagem né, de fechar aquele ciclo narrativo. E agora, de referências são tantas que fica até difícil até de citar elas, tipo tem One Piece, Dragon Ball, vários animes e até coisas da cultura Pop como Star Wars, e é, várias outras.

Q: Como que você elabora os conteúdos para os tutoriais do seu Canal?

W: Eu busco pela demanda do que a galera tá precisando, tipo do que é mais necessário. Tanto que se você ver no canal, tem os tutoriais considerados mais básicos, como como mexer o personagem ou como fazer uma plataforma, porque é aquele conteúdo que o pessoal tá procurando mais para dar o primeiro passo de aprender uma ferramenta. Tem alguma coisa ou outra que é mais avançada, mas eu foco muito mais nesses conteúdos para aquelas pessoas que são iniciantes ou intermediárias, já que é o que o pessoal procura mais. Mas também analiso muita coisa que o pessoal pede nos comentários e tento fazer daquele conteúdo que julgo ser mais relevante naquele momento.

Q: Quais são as maiores dificuldades na hora de desenvolver o jogo?

W: Então o inglês ele é interessante, porque realmente se a pessoa não sabe o básico do inglês, eu recomendo ela a usar um tradutor para colocar os códigos em inglês porque isso é muito importe. Se você precisar contratar uma pessoa depois, o padrão do mercado é que as variáveis estejam em inglês e etc. Então sim, para algumas pessoas essa parte do inglês é uma dificuldade. Outras dificuldades técnicas que um desenvolvedor pode ter, é criar sistemas modulares no código dele no sentido seguinte: eu fiz uma mecânica de árvore de habilidade, mas por algum motivo eu tenho que expandir ela, mas se o código estiver muito amarrado, eu não vou conseguir fazer isso fácil. Então isso queria dizer que o meu jogo não está modular o suficiente, e essa seria também uma dificuldade. E falando sobre a parte do *Game Design*, dificuldade em deixar o jogo divertido, que é uma dificuldade geral de todo desenvolvedor de jogos, que é deixar o jogo tão divertido quanto ele parece ser. E tem também umas dificuldades mais técnicas, de artes, que por exemplo que a gente passou muito com o Cyber, por causa da criação de shaders , já que é uma parte que você tem que saber lidar com matemática e com a sintaxe dos shaders, então cara é uma loucura isso aí. Também tem a questão do *Pixel Art*, de que tamanho vai ser os *sprites*, aí a gente teve que testar vários formatos, como 8x8,16x16,32x32, até chegar no formato que a gente queria, e ainda tivemos que mudar alguns *sprites* antigos, então cara tem muitas dificuldades. E além dessas mais técnicas, tem a dificuldade financeira, porque a gente tirou tudo do nosso bolso, então os gastos foram todos da equipe. E uma coisa que a gente não teve dificuldade, mas acho que é comum, é a gestão de equipe, como ter problema de equipe, não saber de alocar as pessoas, às vezes encontra uma equipe que não é prometida. E graças a Deus não tivemos esse problema com o desenvolvimento do jogo. Mas acho que é isso minha listinha.

Q: Quais foram suas estratégias para contornar a falta de recursos como um desenvolvedor indie?

W: Uma das estratégias foi a de comprar assets , essa sem dúvidas é uma dica que vou dar para contornar esse problema de falta de recursos tanto financeira quanto técnica. Porque ao comprar um asset , por exemplo você comprou um sistema de *behaviour tree* , você não tem que programar aquilo e não perde seu

tempo, e se tivesse que contratar uma pessoa para fazer aquilo ia perder tempo e dinheiro. Outra estratégia também foi enxugar o escopo, por exemplo, o CyberWar teria um escopo muito maior que esse que foi lançado, e aí eu fui enxugando isso, vamos cortar chefões, itens, cenários, senão a gente ia engordurando e engordurando e não ia terminar o jogo. E como eu disse, o jogo saiu do nosso bolso, então não consegui contratar uma equipe realmente grande, então tive que contratar pessoas realmente técnicas e aqui foi uma maneira que achei para driblar outro problema, que era de colocar profissionais muito bons como sócios do jogo, ai não precisava de pagar salário para eles, já que não conseguia tirar esse dinheiro todo do meu bolso. Então foi também uma maneira de driblar a dificuldade financeira. E em algumas características cruciais para o jogo, que é a música , que considero uma das partes principais do jogo, tem gente que subestima a música mas eu não, conversei com um amigo meu freelancer muito competente, mas que fez um preço especial, então acabou que a gente combinou com ele e talz, até para ele ter o Cyber no seu portfólio.

Q: No geral você acha que o mercado brasileiro é receptivo com os jogos *indie* ou ainda existe aquele leve preconceito com a indústria brasileira?

W: Eu acho que melhorou muito em relação com o que era a um tempo atrás, porque tem lançado muito jogo bom brasileiro e teve quebrando essa barreira de que jogo brasileiro não presta. Então acaba que esses jogos bons tem quebrado esse lance aí. E recentemente acabei passando por isso no pré-lançamento do CyberWar que tinha um comentário pejorativo ou outro alí só pelo jogo ser brasileiro sabe. eve até alguns comentários falando tipo “o jogo é tão bom que nem parece ser brasileiro”. É vei, a galera é maldosa, não generalizando, mas algumas são. Então realmente ainda existe um preconceito em relação aos jogos brasileiros, mas teve uma evolução bem grande assim no nosso mercado.

Q: Como você fez para equilibrar a liberdade criativa com a falta de financeiro e de pessoas no seu projeto?

W: Eu fiz isso limitando o escopo do jogo, como eu falei. Tem liberdade criativa? Tem, mas até certo ponto, porque não dá para ir fazendo tudo sem limite

né, tem que ter algum limite. Até porque como um estúdio independente tem que conseguir se manter, como eu disse, a gente não tava recebendo nada durante o desenvolvimento do jogo, então não podíamos brincar com o tempo, tem que terminar o quanto antes. Então enxugando o escopo nós conseguimos manter a liberdade criativa ao mesmo tempo, mas a liberdade criativa sempre existiu e sempre vai ser o pilar do nosso estúdio, o artista, a pessoa tá ali tem que ser criativa e fazer o que quiser, contanto que esteja dentro do escopo.

Q: Como foi o processo de divulgação do jogo e quais foram os métodos de marketing que funcionaram melhor para você?

W: Bacana. A gente tem uma comunidade, tanto do estúdio como a minha pessoal. Então, essa comunidade foi criada com o tempo certo, então quando lançou o jogo, já tinha uma galera que acompanhava a 4 anos atrás o desenvolvimento do jogo, por exemplo. Um dos nossos pilares foi o fato da gente ter essa comunidade, então por a gente ter construído ela ao longo dos anos, acabou que a gente teve uma galera muito fiel, teve pessoal que comprou o jogo ali no primeiro minuto que lançou, já que né, tava ansiosa, já tava acompanhando o desenvolvimento a muito tempo. Então um dos pilares da divulgação do jogo foi essa comunidade, através das redes sociais. E além disso recebemos e enviamos muito e-mails para influencers, produtores de conteúdo. E uma terceira estratégia que a gente fez e até um pouco inovadora, foi de divulgar o jogo em grupos de WhatsApp e etc, com um brinde para quem comprasse ele, ai meio que a gente tira um dinheiro do nosso bolso para comprar esses brindes mas a gente ia fazer isso sorrindo e feliz, porque íamos tirar do nosso bolso, mas íamos né presentear aqueles que acompanharam o desenvolvimento do jogo desde o início. Outras estratégias que a gente não quis aplicar agora, mas que o jogo está em acesso antecipado, que a gente vai estar aplicando mais pra frente quando a gente lançar a 1.0 do jogo que seria a gente contratar uma empresa de assessoria lá no exterior, para fazer uma divulgação do nosso jogo lá fora.

Q: Você utilizou uma publisher ou alguma coisa?

W: Então, não, apesar de que hoje o Cyber está numa publisher atualmente que é a GoGo Games e que eu sou sócio. Então, a publisher não se envolveu diretamente com o lançamento do CyberWar.

Q: Você acha que o feedback positivo que teve no lançamento do jogo e no desenvolvimento dele dentro da comunidade foi importante?

W: Cé ta doido, até hoje é muito importante. Se não é a comunidade o jogo não vai para frente, sem o feedback do pessoal a gente não iria conseguir balancear o jogo direito, fazer a estética dele legal, e se não fosse a comunidade, por exemplo, vamos supor que não lançamos um acesso antecipado e lançamos o jogo direto, sem esse feedback da comunidade o jogo poderia ser um fiasco. Porque a gente ia lançar um jogo com vários problemas por não ter tido um feedback do pessoal antes. Então assim, a comunidade é muito importante para ajudar a guiar o jogo de forma que ele fique ideal, é um pilar muito importante para polir o jogo.

Q; Na sua opinião, qual a visão do futuro dos jogos *indie* no Brasil?

W: A mais otimista possível. Acho que em poucos anos vamos ter muito mais estúdios e publishers adentrando ao mercado nacional, e até com a entrada de novos profissionais na área, como por exemplo vocês (entrevistadores), teremos pessoas cada vez mais técnicas então a gente vai ter estúdios e pessoas técnicas que é o que a gente precisa para crescer a indústria. Sem falar do aumento de apoio do governo, agora com a aprovação do Marco Legal dos Jogos, então vai chegar mais investimento, etc. Então acaba que minha visão é a mais otimista possível e daqui alguns anos quero ver vários estúdios brasileiros de sucesso aí, concorrendo a Game Awards e tudo mais.

Q: Como você fez para organizar as etapas do desenvolvimento do jogo? Tipo, existe aquele horizonte de coisas para fazer e a gente nunca sabe por onde exatamente começar.

W: Então, a primeira etapa é a documentação básica do jogo para ter um Norte, Porque às vezes o cérebro engana né? Você pensa, pô essa é a melhor ideia

que já tive na minha vida nunca vou esquecer ela, e passa um tempo alí você distrai e acaba esquecendo. Então é bom documentar uma base, para você montar um protótipo para ver se a ideia é realmente divertida, mandar ele pros amigos, pra perguntar se eles acham que a ideia é legal e tal. A partir dali você valida o produto, se o protótipo com quadradinhos e bolinhas já foi divertido, é um sinal que você tá indo em uma boa direção, e ai você segue para a etapa de desenvolvimento, onde você já começa desenvolvendo o jogo, mas jamais utilizando o código do protótipo, aquele dali você esquece. E aí desenvolvendo o projeto do 0, ai você é mais questão de se organizar, porque depende né, se você tá fazendo sozinho ou em equipe, se for uma equipe recomendo ali uma reunião semanal ou quase isso, para estar alinhado do que tá acontecendo e do que não tá, colocar tarefas, metas, importantíssimo colocar as metas. Ter uma ideia do escopo do seu jogo e tentar manter a ideia do escopo, não sair muito dela, e por isso é muito importante a documentação do início.

Q: Quais padrões de projeto você utilizou no desenvolvimento do seu jogo?

W: Olha, a gente trabalhou com um MVC adaptado, porque em um desenvolvimento de jogos é impossível aplicar ele literalmente, mas a gente adaptou ele para o projeto. Vamos dizer que o *back end* é a parte mais chata do projeto, as IA dos npcs e tudo mais, e no view control tem a interface da Unity e a ligação do código mais pesado com o que está sendo visto pelo jogador. Normalmente se eu tivesse um outro programador da equipe, um ficaria com a parte pesada e a parte mais visual com o outro programador. Também usamos o scrum adaptado, porque não acho que essas arquiteturas de Software consigam se encaixar muito bem com o desenvolvimento de jogos, ela tem que ser adaptada e não feita ao pé da letra. Então acho que seriam mais essas duas arquiteturas durante o desenvolvimento do jogo.

7.2. APÊNDICE B - Entrevista com Jair Bortolucci

Data: 22/10/2024

Q: QCG ESTUDIOS | J: JAIR BORTOLUCCI

Canal: <https://www.youtube.com/@ProgramandoGames>

Sobre o Convidado: Jair Bortolucci é um youtuber que produz conteúdo de tutoriais para jogos em Unity, se destacando pela qualidade de seu material. Além disso, ele trabalha em uma empresa de jogos de médio porte como desenvolvedor.

Q: Como você começou sua entrada no mercado de desenvolvimento de jogos?

J: Faz tempo.. Em 2012 fiz um curso na faculdade de tecnologia aqui em São Paulo, e ela oferecia o curso de desenvolvimento de jogos e foi por aí que comecei, não esperava ser programador de jogos, gostava mais da parte de animação 3d e desenho, enfim, e a faculdade era mais voltada ao desenvolvimento, era um curso novo, e por aí comecei a gostar de programação mas antes programava jogos em powerpoint.

Q: O que exatamente te inspirou a criar o canal

J: Antes de começar o canal, quando estava no mestrado, eu via que faltava conteúdo de desenvolvimento de jogos da perspectiva de programação, tinha muitos tutoriais que eram cópias de outros tutoriais gringos e sentia falta desses tutoriais mais elaborados focados mais na matemática e na física, e acabou que começou aí, com essa motivação

Q: Você já enfrentou alguma dificuldade que te deu vontade de desistir do projeto ou do canal?

J: Particularmente, eu peguei uma época muito boa da área de desenvolvimento, não só de jogos mas também na área num geral, então não tive dificuldade de entrar no mercado de trabalho, tanto pra fazer freelancer de jogos quanto pra trabalhar com programação, trabalhei a maior parte da minha vida como programador de simulações computacionais, mas trazendo mais para a perspectiva

de hoje, hoje imagino que esteja mais concorrido, por mais que as ferramentas estejam mais evoluídos, hoje em dia para fazer um jogo ter visibilidade é bem mais difícil que no passado

Q: Quais foram as maiores influências para começar a desenvolver jogos?

J: Foi um youtuber sul-africano chamado Sebastian Lague, ele faz alguns videos de desenvolvimento chamados de coding challenge e ele pega várias implementações e desenvolve, ele vai programando e explicando o que ele estava fazendo, e para jogos tive muitas referências de jogos como Towerfall e Celeste, e jogos iniciantes como Knights of Pen and Paper, minhas referências sempre foram jogos *indie*.

Q: Como você elabora/escolhe o conteúdo que você vai fazer para seu canal?

J: Cara na época eu pesquisava assim conteúdo que eram assim bem relevantes que estavam em inglês e tentava trazer eles pra português, tipo o vídeo mais famoso do canal é um de câmera de FPS e eu vi que tinha poucos vídeos em português que falavam de forma detalhada como fazer isso, então eu meio que fazia isso, ia pegando referências lá de fora e ia trazendo para a nossa língua. E outros era por mais interesse próprio mesmo, os que eu achava mais interessante a implementação deles.

Q: Na sua opinião quais são os principais desafios que os produtores *indie* enfrentam no Brasil?

J: Eu tava pensando inclusive sobre isso, e acho que hoje em dia não sei como é direito, mas sei que tem algumas empresas por trás que incentivam em algumas ideias/projetos, só que isso venho em um momento que o mercado ta muito cheio, com muitos desenvolvedores, por exemplo, a pouco tempo atrás teve um edital aqui em São Paulo, escolhendo alguns desenvolvedores, e teve um monte de inscrição. Então, fora isso a maior dificuldade é conseguir financiar o

desenvolvimento do jogo, porque se você não for financiado, você vai ter que fazer o jogo ou no tempo livre e com recursos próprios, aí isso já vira empreender como qualquer outro tipo de empreendimento e vocês já devem saber que empreendimento no Brasil não é fácil, em nenhum tipo de empreendimento. Então eu colocaria esses dois cenários, se você quer fazer o jogo e tem tempo livre e determinação, é até que relativamente fácil, já que você tem as ferramentas para isso de forma simples de achar, mas se você quer fazer isso de forma financeira, aí fica difícil, porque o dinheiro vai ter que sair do próprio bolso.

Q: Você acha que o mercado brasileiro é receptivo para jogos *indie*?

J: Olha, difícil dizer sem dados, mas eu acredito que hoje em dia os públicos existem para determinados produtos, os jogos *indie* e os que são de empresas mesmo, mas acho que está mudando essas coisas como por exemplo com a “qualidade de gráficos”, que esse preconceito já está mudando. Mas no geral, ainda acho que esse preconceito com jogos *indie* existe sim, e é só analisar as premiações, que muitas vezes não premiam jogos excepcionais só pelo fato de serem *indie*.

Q: Referente a programação qual foi a maior dificuldade que você teve?

J: Difícil falar um porque a gente faz tanta coisa assim que é meio difícil escolher um. Falando assim sobre recentemente mesmo, nessa empresa que estou trabalhando agora, onde a gente usa a Unity, e eu acredito que o que deu mais trabalho foi em relação a algoritmos de busca, de formar de caminhos, é realmente difícil, até para fazer o vídeo de tutorial no canal foi tão difícil, que eu até descartei a ideia. Então acho que essa questão de IA para se movimentar num ambiente 2D, sem ficar batendo em parede, pulando plataforma, foi a coisa mais chata que tive que mexer.

Q: Você acredita que as redes sociais são importantes para um desenvolvedor indie hoje em dia?

J: Com certeza, eu acho que muitos jogos só tiveram sucesso graças ao Youtube e outras redes sociais. Porque, tipo assim, hoje em dia você não consegue construir um jogo sozinho, antigamente tinha uma certa romantização do desenvolvedor *indie*, que era um cara que ficava trancado no quarto e programando e soltava o jogo e o jogo explodia e de fato era assim que acontecia. Só que não é mais assim porque, primeiro porque mudou muito o mercado e segundo porque existem muitos jogos no mercado agora, e se você não deixar as pessoas participarem do desenvolvimento do jogo, elas não vão ter interesse, por mais incrível que seja seu jogo, no fim para o público vai ser só mais um jogo. Então sim, trabalhar esse acompanhamento e participação do público com o desenvolvimento no jogo através das redes sociais é uma parte fundamental hoje em dia.

Q: Qual é sua visão para o futuro dos jogos *indie* no Brasil?

J: Eu acho que tem uma perspectiva de melhoria, a partir do momento que nosso Governo perceber que a área de desenvolvimentos de jogos é bem lucrativa, que nem outros governos de fora já perceberam, eles vão começar aí a melhorar as coisas, é uma coisa que não tem como parar porque cada vez mais as pessoas vão estar desenvolvendo e mostrando que é lucrativo, e aí as coisas vão melhorando com o tempo.

Q: Como você enxerga algumas iniciativas como por exemplo o Marco Legal dos Jogos?

J: Eu acho que é bem importante e benéfico, eu não li exatamente tudo que estava ali, mas acho que é bem importante o início da discussão né? Já começar a dar uma atenção para o mercado e ver as pessoas que querem entrar nele, já é um primeiro passo né. Se existem problemas nas leis ou questões que você vê muita gente falando, sobre algumas coisas que não tá legal, mas acho que o importante mesmo é o pontapé inicial e com o tempo vai mudando e se adequando melhor.

Q: Quais conselhos você daria para alguém que está adentrando agora no mercado e quer se destacar?

J: Acho que o conselho principal é: faça jogos. Comece fazendo um jogo, mas claro, que existe toda a formação técnica, o estudo, mas com a grande quantidade de informação que existe hoje em dia, você consegue estudar enquanto desenvolve um jogo. Então acho que minha dica é: defina um jogo simples e faça isso até o final. E depois repita o processo, de novo e de novo, que com isso você vai aprender as coisas muito rápido.

Q: Quais padrões de projeto você usa com mais frequência em seus projetos?

J: Cara eu uso muito o *state machine* e quando to com preguiça uso o *singleton*, também uso muito o *observer* e o *fracture*. Mas acho que principalmente o *observer* e o *state machine*, acho que são indispensáveis em jogos, porque quando você pensa em *state machine* você pensa em estados inteligentes de um agente , mas não só isso, o jogo tem seus próprios estados, como o game loop e tudo mais, então ele basicamente está presente em todos os cantos do código.

Q: Você está trabalhando em algum projeto novo no momento?

J: Eu estou trabalhando em um projeto da empresa, que é um aplicativo que está tentando trazer a sensação de dançar para um jogo, e estou como um desenvolvedor *indie* senior na empresa, cuidando tanto da parte do front e do *back end*. E pessoalmente eu estou trabalhando em projetos de *assets* agora e inclusive uma coisa que tem sido bem rentável e também sugeria para quem está começando agora é a venda de *assets*, então eu estou trabalhando agora nesse sentido de criar mais *assets* e é isso.

7.3. APÊNDICE C - Avaliações da primeira demo

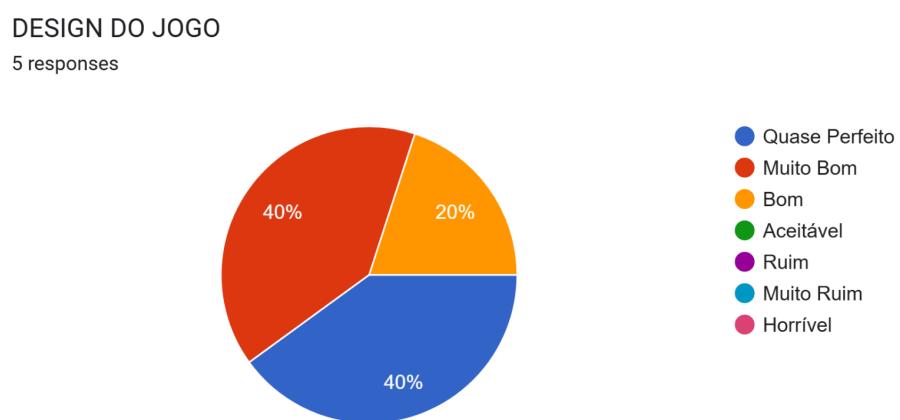
Segue abaixo o conteúdo das avaliações da primeira demo do jogo, apresentada na mostra de cursos do CEFET-MG.

Pergunta: Qual a sua opinião sobre o design do jogo?

André Victor Gonçalves Nascimento: "Gostei bastante do estilo de arte, realmente impecável! A UI do jogo também é legal e o menu muito promissor, a paleta de cores me agradou. Outro ponto que achei legal foi o design da mira. Porém, o personagem está em um plano superior ao da parede(quando encosto nela a minha mão aparece em cima)."

Daniel Pacheco Monteiro: "o design e arte do jogo está muito linda mas o chão por uma questão de cor não combina muito com as paredes do jogo além disso o personagem está meio estranho principalmente na parte das mãos que parecem só duas bolas sem conexão isso é mais evidente quando o personagem está de lado ou de costas

QUADRO 6 - Opinião sobre o design do jogo



FONTE: Autoria própria (2024).

Pergunta: Qual a sua opinião sobre a movimentação?

Bruno de Jesus Anacleto: "Só senti falta de movimentar enquanto ataca"

André Victor Gonçalves Nascimento: "Gostei bastante da mecânica de movimentação, porém acredito que tenha duas outras que podem servir bem

também, mesmo não tendo certeza. Uma é movimentação em point & click (seria mais como uma opção extra pro jogador) e a movimentação baseada em onde o player está olhando (se tiver pra baixo, o w andaria para baixo, e não para cima)."

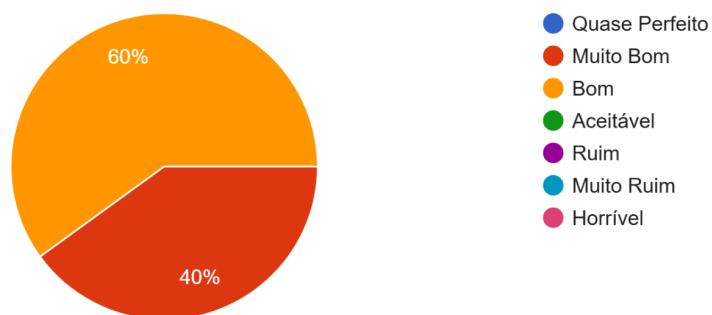
y

Daniel Pacheco Monteiro: "Quando o personagem ataca parece que tem um mini delay pra voltar a se movimentar fazendo o personagem parecer meio travado em combate"

QUADRO 7 - Opinião sobre a movimentação do jogador

MOVIMENTAÇÃO DO PLAYER

5 responses



FONTE: Autoria própria (2024).

Pergunta: Qual a sua opinião sobre o mapa?

André Victor Gonçalves Nascimento: "O mapa é muito promissor, porém achei meio vazio e escuro, para uma beta, está incrível!"

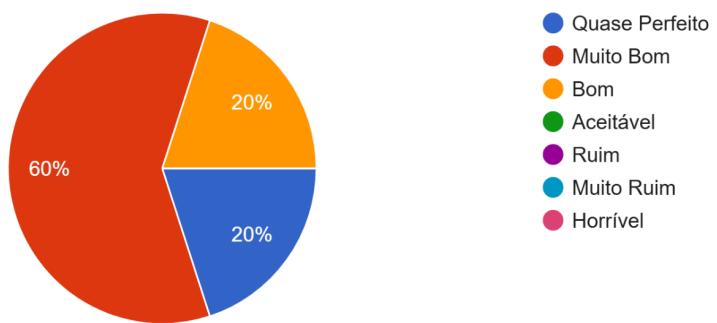
Daniel Pacheco Monteiro: "A ambientação é muito bonita, mas falta elementos do cenário (tochas e ossos espalhados por exemplo) o mapa também tem áreas bem memoráveis como a sala com 20 slimes de uma vez, a sala da cruz e a sala com os 4 pilares, além disso o mapa se sai muito bem no quesito de ser um labirinto, além disso acredito que o mapa devia tomar uma direção mais catacumbas

de paris onde há áreas do mapa iguais as que tem agora e passagens estreitas onde não há pedras nas paredes apenas terra”

QUADRO 8 - Opinião sobre o mapa e ambientação

AMBIENTAÇÃO/MAPA DO JOGO

5 responses



FONTE: Autoria própria (2024).

Pergunta: Qual a sua opinião sobre a dificuldade?

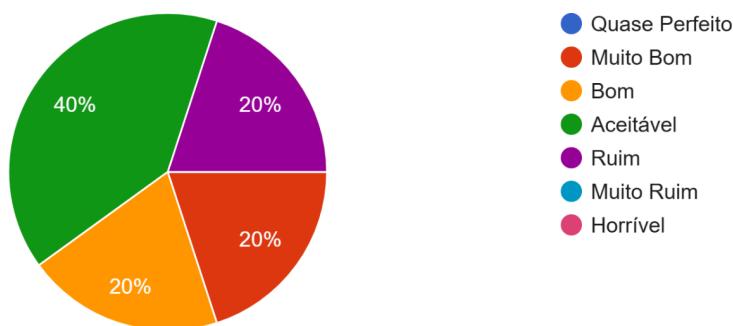
André Victor Gonçalves Nascimento: “Achei as slimes muito fortes pra mobs iniciais, já que o player não inicia com nenhum item de aumento de dano (sofri muito para abrir os baús kkkk).”

Daniel Pacheco Monteiro: “O jogo está atualmente muito fácil por causa do pathing dos inimigos que toda vez ficam presos em quinas das paredes e em baús fazendo assim muito fácil de ataca-los pelas paredes, mas fora isso o combate está bom apenas precisa de mais armas para o personagem “

QUADRO 9 - Opinião sobre a dificuldade do jogo

DIFICULDADE DO JOGO

5 responses



FONTE: Autoria própria (2024).

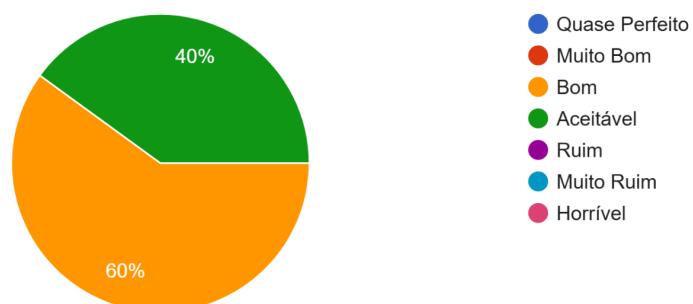
Pergunta: Qual a sua opinião sobre a interação com o ambiente?

Daniel Pacheco Monteiro: “não há muita interação com o ambiente ainda além de 3 baús mas seria bom adicionar coisas como pinturas ou detalhes nas paredes e celas de prisioneiros para fazer o storytelling pelo ambiente que o personagem está inserido

QUADRO 10 - Opinião sobre a interatividade do jogo

INTERAÇÃO COM O AMBIENTE

5 responses



FONTE: Autoria própria (2024).

Pergunta: Qual a sua opinião sobre o jogo?

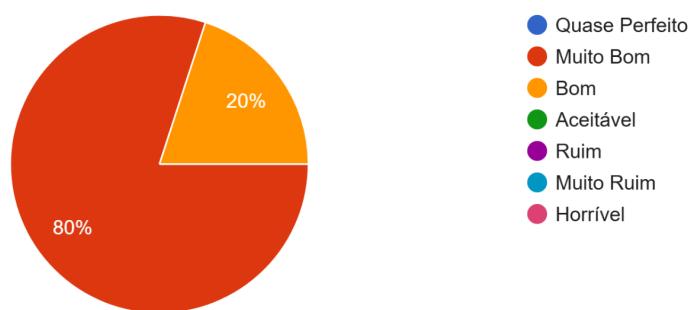
André Victor Gonçalves Nascimento: “Jogo bem viciante e legal, me senti desafiado a achar os baús e encontrar métodos de derrotar as slimes. A ambientação é boa e a arte impecável, gostei demais, mal posso esperar para as versões futuras.”

Daniel Pacheco Monteiro: “o jogo em geral é um beta bastante sólido tendo sua arte já bastante bonita e mecânicas básicas já implementadas porém algumas mecânicas mais do que as outras precisam de polimento.”

QUADRO 11 - Opinião sobre o jogo

O JOGO EM GERAL

5 responses



FONTE: Autoria própria (2024).

7.4 APÊNDICE D - Avaliações da segunda demo

Segue abaixo o conteúdo das avaliações da segunda demo do jogo, apresentada na 33º META.

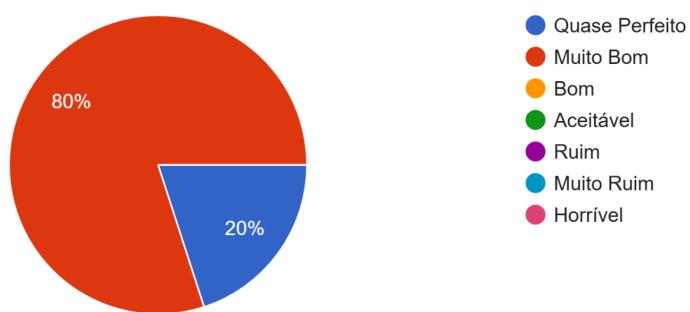
Pergunta: Qual a sua opinião sobre o design do jogo?

Hiane Coelho Matos : "O design é lindo. A pixel art ficou muito bem feita em todos os elementos do jogo."

QUADRO 12 - Opinião sobre o design do jogo

DESIGN DO JOGO

5 responses



FONTE: Autoria própria (2024).

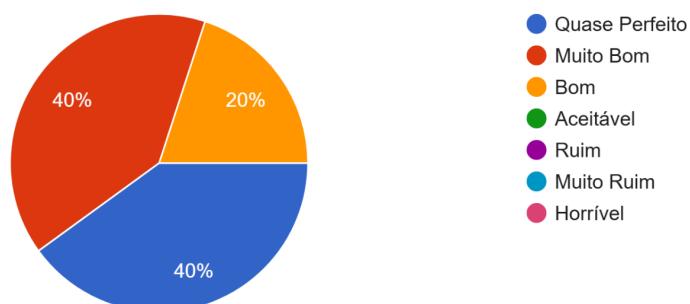
Pergunta: Qual a sua opinião sobre a movimentação?

Hiane Coelho Matos : "A movimentação é completamente eficiente e suave."

QUADRO 13 - Opinião sobre a movimentação do jogador

MOVIMENTAÇÃO DO PLAYER E COMBATE

5 responses



- Quase Perfeito
- Muito Bom
- Bom
- Aceitável
- Ruim
- Muito Ruim
- Horrível

FONTE: Autoria própria (2024).

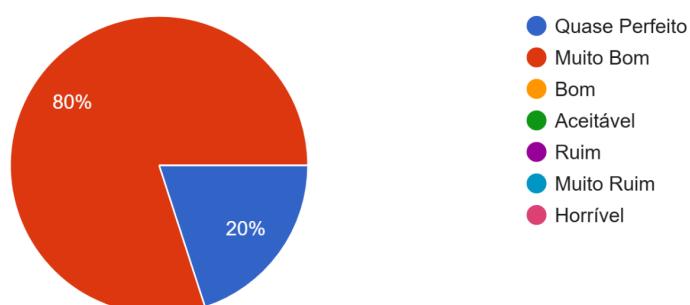
Pergunta: Qual a sua opinião sobre o mapa?

Hiane Coelho Matos : “Os mapas são bonitos e interativos. Um pouco vazio, mas ta bonito”

QUADRO 14 - Opinião sobre o mapa e ambientação

AMBIENTAÇÃO/MAPA DO JOGO

5 responses



- Quase Perfeito
- Muito Bom
- Bom
- Aceitável
- Ruim
- Muito Ruim
- Horrível

FONTE: Autoria própria (2024).

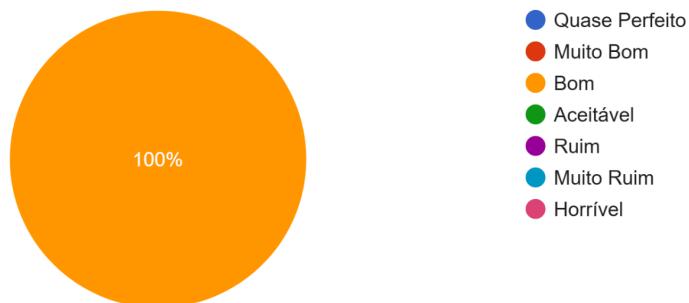
Pergunta: Qual a sua opinião sobre a dificuldade?

Hiane Coelho Matos : “Eu acredito que o nível de dificuldade não seja muito elevado mas para pessoas leigas talvez pudesse ser difícil.”

QUADRO 15 - Opinião sobre a dificuldade do jogo

DIFICULDADE DO JOGO

5 responses



FONTE: Autoria própria (2024).

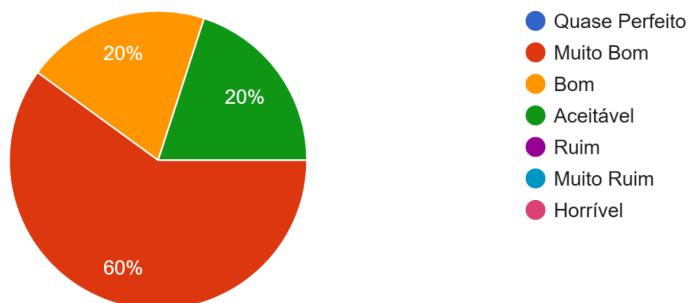
Pergunta: Qual a sua opinião sobre a interação com o ambiente?

Hiane Coelho Matos : “A interação é muito boa e talvez futuramente mais elementos poderiam ser incrementados.”

QUADRO 16 - Opinião sobre a interatividade do jogo

INTERAÇÃO COM O AMBIENTE

5 responses



FONTE: Autoria própria (2024).

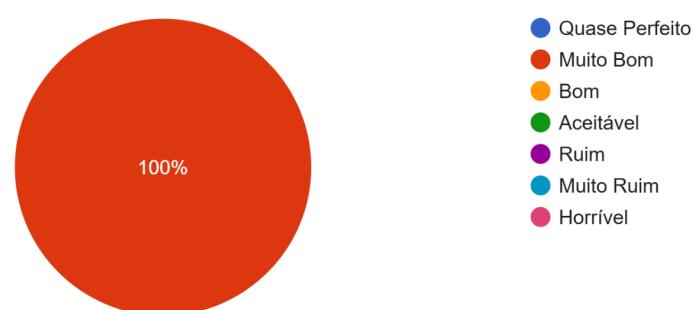
Pergunta: Qual a sua opinião sobre o jogo?

Hiane Coelho Matos : “Em geral o jogo é muito bonito e interativo e com certeza a versão final vai ser muito satisfatória.”

QUADRO 17 - Opinião sobre o jogo

O JOGO EM GERAL

5 responses



FONTE: Autoria própria (2024).