
**FACULDADE DE TECNOLOGIA DE AMERICANA “Ministro Ralph Biasi”
Curso Superior de Tecnologia em Jogos Digitais**

Felipe Gonçalves
Giovanni Marchan Santos

**EQUITIS ADVENTUM
RELATÓRIO DE DESENVOLVIMENTO**

**FACULDADE DE TECNOLOGIA DE AMERICANA “Ministro Ralph Biasi”
Curso Superior de Tecnologia em Jogos Digitais**

Felipe Gonçalves
Giovanni Marchan Santos

EQUITIS ADVENTUM
RELATÓRIO DE DESENVOLVIMENTO

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Jogos Digitais sob a orientação do Prof. Esp. José William Pinto Gomes.

Área de concentração: Jogos Digitais.

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana
Ministro Ralph Biasi- CEETEPS Dados Internacionais de
Catalogação-na-fonte**

GONÇALVES, Felipe

EQUITIS ADVENTUM: RELATÓRIO DE DESENVOLVIMENTO. /
Felipe Gonçalves, Giovanni Marchan Santos – Americana, 2024.

96f.

Relatório técnico (Curso Superior de Tecnologia em Jogos
Digitais) - - Faculdade de Tecnologia de Americana Ministro Ralph
Biasi – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Esp. José William Pinto Gomes

1. Jogos Digitais 2. Role-Playing Game
3. Plataforma 2D. I. GONÇALVES, Felipe, II. SANTOS, Giovanni
Marchan III. GOMES, José William Pinto IV. Centro Estadual de
Educação Tecnológica Paula Souza – Faculdade de Tecnologia de
Americana Ministro Ralph Biasi

CDU: 681.6
681.3.061C#
681.6

Elaborada pelo autor por meio de sistema automático gerador de
ficha catalográfica da Fatec de Americana Ministro Ralph Biasi.

Felipe Gonçalves
Giovanni Marchan Santos

EQUITIS ADVENTUM

Relatório de Desenvolvimento

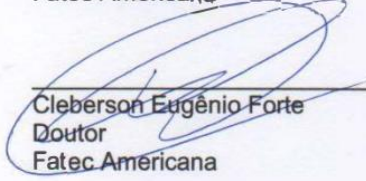
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Jogos Digitais pelo CEETEPS/Faculdade de Tecnologia de Americana "Ministro Ralph Biasi".
Área de concentração: Jogos Digitais.

Americana, 02 de dezembro de 2024.

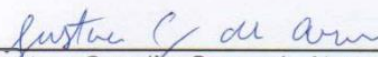
Banca Examinadora:



José William Pinto Gomes
Especialista
Fatec Americana



Cleberson Eugênio Forte
Doutor
Fatec Americana



Gustavo Carvalho Gomes de Abreu
Especialista
Fatec Americana

RESUMO

Esse trabalho apresenta o desenvolvimento de um jogo em plataforma 2D, chamado “Equitis Adventum” com o objetivo de ser um clássico, elegante e simples RPG de aventura. Desenvolvido em três fases, com um tutorial simples e objetivo, cada uma dessas fases tem seu próprio conjunto de desafios, incluindo obstáculos e inimigos bem elaborados. A narrativa leva o jogador, de um campo aberto agradável, até uma montanha escura e fechada, utilizando a metodologia de desenvolvimento ágil para gestão do projeto, desenvolvimento de *pixels arts* 2D, programação de mecânicas de jogo e design de níveis, para proporcionar uma experiência imersiva e desafiadora. Os resultados mostraram um jogo completo e funcional, capaz de entreter e envolver os jogadores com suas mecânicas e jogabilidade simples. Por fim, o projeto conclui que a combinação de mecânicas simples ainda pode funcionar para criar algo envolvente para os jogadores fãs dos jogos clássicos.

Palavras-chave: Jogos digitais, plataforma 2D, RPG.

ABSTRACT

This project presents the development of a 2D platform game, called "Equitis Adventum" with the aim of being a classic, elegant and simple adventure RPG. Developed in three levels, with a simple and objective tutorial, each of these levels has its own set of challenges, including well-crafted obstacles and enemies. The narrative takes the player from a pleasant open field to a dark and closed mountain, using the agile development methodology for project management, 2D pixel art development, game mechanics programming and level design, to provide an immersive and challenging experience. The results showed a complete and functional game, capable of entertaining and engaging players with its simple mechanics and gameplay. Finally, the project concludes that the combination of simple mechanics can still work to create something engaging for players who are fans of the classic games.

Keywords: *Digital games, 2D platform, RPG.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Tela do mapa da fase de tutorial.	30
Figura 2 – Tela do mapa da primeira fase.	31
Figura 3 - Tela do mapa da segunda fase.	32
Figura 4 - Tela do mapa da terceira fase.	34
Figura 5 - Cavaleiro Godfrey.	41
Figura 6 – Morcego.	42
Figura 7 – Camaleão.	42
Figura 8 – Planta.	42
Figura 9 – Fantasma.	42
Figura 10 - Cogumelo ambulante.	43
Figura 11 - Tronco.	43
Figura 12 - Cenário do tutorial.	43
Figura 13 - Cenário da primeira fase.	43
Figura 14 - Cenário da segunda fase.	44
Figura 15 - Cenário da terceira fase.	44
Figura 16 - Escudo do cavaleiro Godfrey.	44
Figura 17 - Espada do cavaleiro Godfrey.	45
Figura 18 - Animação do cavaleiro Godfrey parado.	45
Figura 19 – Animação do cavaleiro Godfrey correndo.	46
Figura 20 - Animação do cavaleiro Godfrey pulando.	46
Figura 21 – Animação do cavaleiro Godfrey caindo.	47
Figura 22 - Animação do cavaleiro Godfrey atacando.	47
Figura 23 – Animação do cavaleiro Godfrey defendendo.	48
Figura 24 - Animação do cavaleiro Godfrey morto.	48
Figura 25 - Animação do morcego parado.	49
Figura 26 - Animação do morcego acordando.	49
Figura 27 - Animação do morcego voando.	49
Figura 28 - Animação do morcego retornando à origem.	50
Figura 29 - Animação do morcego tomando dano.	50
Figura 30 - Animação do camaleão parado.	50
Figura 31 - Animação do camaleão andando.	51

Figura 32 - Animação do camaleão atacando.	51
Figura 33 - Animação do camaleão tomando dano.	51
Figura 34 - Animação da planta parada.	52
Figura 35 - Animação da planta disparando.	52
Figura 36 - Animação da planta se escondendo.	52
Figura 37 - Animação da planta aparecendo.	53
Figura 38 - Animação da planta tomando dano.	53
Figura 39 - Animação do fantasma parado e seguindo o cavaleiro.	53
Figura 40 - Animação do fantasma desaparecendo.	54
Figura 41 - Animação do fantasma aparecendo.	54
Figura 42 - Animação do fantasma tomando dano.	54
Figura 43 - Animação do cogumelo ambulante parado.	55
Figura 44 - Animação do cogumelo ambulante andando.	55
Figura 45 - Animação do cogumelo ambulante tomando dano.	55
Figura 46 - Animação do tronco parado.	55
Figura 47 - Animação do tronco disparando.	56
Figura 48 - Animação do tronco tomando dano.	56
Figura 49 - Efeito de poeira do solo.	56
Figura 50 - Efeito sangramento.	57
Figura 51 - Interface do menu inicial.	57
Figura 52 - Interface opções de volume no menu.	58
Figura 53 - Interface de créditos.	58
Figura 54 - Interface do seletor de níveis.	59
Figura 55 - Interface do HUD.	59
Figura 56 - Interface do jogo pausado.	60
Figura 57 - Interface das opções de volume no jogo pausado.	60
Figura 58 - Interface da tela de morte.	61
Figura 59 - <i>Background</i> do tutorial e da terceira fase.	61
Figura 60 - <i>Background</i> da primeira fase.	62
Figura 61 - <i>Background</i> da segunda fase.	62
Figura 62 - Projétil do inimigo planta.	62
Figura 63 - Projétil do inimigo tronco.	63
Figura 64 - Planejamento de tarefas do Giovanni Marchan Santos.	65
Figura 65 - Planejamento de tarefas do Felipe Gonçalves.	65

Figura 66 - <i>Script</i> PlayerBehaviour variáveis.	68
Figura 67 - <i>Script</i> PlayerBehaviour função de entrada do jogador.	69
Figura 68 - <i>Script</i> FollowPlayerArea variáveis.	70
Figura 69 - <i>Script</i> FollowPlayerArea função de troca de estado.	71
Figura 70 - <i>Prefab</i> do SceneTransition sendo utilizado na fase 3.	72
Figura 71 - <i>Script</i> SceneTransition funções de troca de cena.	73
Figura 72 - Gráfico do gênero dos participantes que participaram dos testes.	82
Figura 73 - Gráfico da faixa etária dos participantes dos testes.	82
Figura 74 – Gráfico de frequência que os participantes dos testes jogam Jogos Digitais.	82
Figura 75 – Gráfico da opinião geral dos participantes sobre os testes.	83
Figura 76 - Gráfico da opinião dos participantes dos testes sobre o <i>desing</i> do jogo.	83
Figura 77 – Gráfico da opinião dos participantes dos testes sobre a jogabilidade. ..	84
Figura 78 - Gráfico da opinião dos participantes dos testes sobre o menu e interação.	84
Figura 79 - Gráfico da opinião dos participantes dos testes sobre a arte e o som. ..	85

LISTA DE QUADROS

Quadro 1 - Quadro de comparação entre os jogos.	18
---	----

LISTA DE ABREVIATURAS E SIGLAS

RPG: Role Playing Game (Jogo de interpretação).

PC: Computador.

TCC: Trabalho de conclusão de curso.

Nintendo EAD: Nintendo Entertainment Analysis & Development Division (Divisão de Análise e Desenvolvimento de Entretenimento da Nintendo).

NPC: non-playable character (personagem não jogável).

SUMÁRIO

INTRODUÇÃO	14
1 REVISÃO BIBLIOGRÁFICA.....	15
1.1 Jogos digitais.....	15
1.2 Jogos 2D.....	15
1.3 <i>Pixel Art</i>	15
1.4 Por que escolhemos plataforma?	15
2 METODOLOGIA.....	17
2.1 Características.	17
2.2 Jogos similares.....	18
2.2.1 Quadros comparativos. -----	18
3 CONCEITO.....	19
3.1 Gênero.....	19
3.2 Classificação etária.	19
3.3 Trama principal.	19
3.4 <i>Look & Feel</i>	19
3.5 Visão geral do conceito do jogo.	19
4 HISTÓRIA E NARRATIVA.....	21
4.1 Mundo do jogo.	21
4.2 Personagens.....	21
4.2.1 Eventos e Ações. -----	23
4.3 Visão geral da narrativa.	23
4.3.1 Integração.-----	23
4.3.2 Estrutura e Progressão.-----	23
4.3.3 Roteiro. -----	23
4.3.4 Tutoriais e manual do jogo.-----	24
4.4 Cenas e Sequências cinemáticas.	24
5 GAMEPLAY	25
5.1 Imersão.	25
5.2 Estrutura de missões e desafios.....	25
5.3 Objetivos do jogo.....	26
5.4 Fluxo de jogo.....	26

5.5	Mecânica do Jogo.....	26
5.5.1	Regras implícitas e explícitas do jogo.	26
5.5.2	Física.	27
5.5.3	Movimentação dos personagens.	27
5.5.4	Gatilhos e ações.	28
5.5.5	Mecânica de Combate.	29
5.6	Projeto de Fases (<i>Level Design</i>).	29
5.7	Projeto de Interface.	35
5.7.1	Sistema Visual.	35
5.7.2	Sistema de Controle.	36
5.7.3	Fluxo de Telas.	36
5.8	Projeto de Som.....	37
5.8.1	Descrição geral.	37
5.8.2	Trilhas sonoras.	38
5.8.3	Efeitos Sonoros.	38
5.9	Sistema de Inteligência Artificial.....	39
5.9.1	Inimigos.	39
6	ARTE	41
6.1	Guia de cores e estilos gráficos.....	41
6.2	<i>Asset List</i>	41
6.2.1	Personagens.	41
6.2.2	Ambientes.	43
6.2.3	Equipamentos.	44
6.2.4	Armas.	45
6.2.5	Animações.	45
6.2.6	Efeitos.	56
6.2.7	Interfaces.	57
6.2.8	<i>Backgrounds</i>	61
6.2.9	Projéteis.	62
7	DOCUMENTAÇÃO TÉCNICA	64
7.1	5.1 – Requisitos de sistema.....	64
7.2	Engenharia de <i>Software</i>	65
7.3	<i>Software(s)</i> Secundário(s).	66
7.4	<i>Game Engine</i>	66

7.5	Bibliotecas.....	66
7.6	Programação.....	67
7.7	<i>Scripting</i>	67
7.7.1	<i>Script</i> de movimentação do personagem principal.....	67
7.7.2	<i>Script</i> de movimentação e inteligência artificial do inimigo fantasma. 70	
7.7.3	<i>Script</i> de transição de tela.....	71
8	GERENCIAMENTO E ANÁLISE DO PROJETO	74
8.1	Organização inicial do projeto.	74
8.2	Escopo do projeto.	74
8.3	Recursos.....	74
8.4	Entregas.....	74
8.5	Premissas.	75
8.6	Restrições.....	75
8.7	Cronograma.....	75
8.8	Estimativa de Custos.....	76
8.9	Limites do Projeto.....	76
8.10	Processos.....	76
8.11	Análise de Riscos.	77
8.12	Viabilidade Técnica.	78
8.13	Viabilidade Econômica.	79
8.14	Plano de testes e Controle de qualidade.	79
8.15	Ferramentas e técnicas.....	80
9	RESULTADOS.....	81
9.1	Teste alfa realizado.....	81
9.2	Teste beta realizado.....	81
	CONSIDERAÇÕES FINAIS.....	86
	DOWNLOAD DO JOGO.....	89
	REFERÊNCIAS.....	90
	APENDICE A – QUESTIONÁRIO PARA TESTES NA VERSÃO BETA.....	92

INTRODUÇÃO

Muitos jogos de plataforma fizeram sucesso durante a história dos jogos digitais, e continuam fazendo até nos dias de hoje.

Este trabalho tem o objetivo de desenvolver um jogo de plataforma, onde o jogador controla um avatar que se movimenta e salta entre plataformas e obstáculos, enfrenta inimigos e desafios, e coletando bônus [Smith et. al. 2008].

Utilizando vários jogos como inspiração, sem ter um que seja total referência, surgiram várias ideias, como *level design* de fases, arte gráfica, mecânicas, efeitos e trilha sonoras. As referências para o desenvolvimento desse jogo foram jogos antigos e recentes do mesmo gênero. Também foram executadas pesquisas intensas e aprofundadas e nenhum jogo encontrado é similar ao deste trabalho.

As mecânicas desse gênero podem ser bastante variadas dependendo de qual jogo é utilizado como referência para o seu desenvolvimento. Então para tomar a jogabilidade desse jogo diferente dos demais, é possível bloquear os inimigos e projéteis do jogo com um escudo.

A importância do trabalho é mostrar as habilidades desenvolvidas no curso de jogos digitais na faculdade de tecnologia de Americana e aplicá-las no desenvolvimento do Equitis Adventum na *engine* Unity.

O trabalho está organizado em três capítulos, sendo o capítulo I será a fundamentação teórica, capítulo II é o percurso metodológico e o capítulo III conterà os resultados, análise e discussão dos dados.

1 REVISÃO BIBLIOGRÁFICA

1.1 Jogos digitais.

Os jogos digitais têm se tornado uma influência na vida das pessoas, tanto na forma de entretenimento e diversão quanto como ferramenta educativas e simulações. Mas o que são jogos digitais? Segundo a clássica definição de Huizinga (1938), “um jogo é uma atividade voluntária que se desenrola dentro de limites temporais e espaciais específicos, regida por regras próprias, e desvinculada das seriedades da vida cotidiana, sendo capaz de absorver o jogador de maneira intensa e completa”.

Além de Huizinga, um jogo digital ou jogo eletrônico pode ser definido, segundo a ALVES (2023), “O jogo eletrônico, videojogo ou videogame é aquele que usa a tecnologia de computador. Ele pode ser jogado em computadores pessoais (dentre eles tablets e telefones celulares), em máquinas de fliperama ou em consoles”.

1.2 Jogos 2D.

Segundo Gonçalves (2024), “os jogos 2D são jogos com uma perspectiva bidimensional, onde os objetos do jogo são representados num plano, sem a utilização de modelos tridimensionais. Usam gráficos planos, chamados de sprites, e não possuem geometria tridimensional”.

1.3 *Pixel Art*.

“A definição técnica de *pixel art* pode até ser um pouco complicada, mas a sua aparência é simples: quem a vê numa tela a reconhece na hora. Uma composição digital com baixa resolução, cores limitadas e uma estética simpática, fofinha e agradável, que remete instantaneamente aos videogames mais antigos” (Torres, 2023).

1.4 Por que escolhemos plataforma?

Optou-se pelo desenvolvimento de um jogo no gênero plataforma devido a diversos fatores que alinham praticidade e viabilidade com o prazo disponível para o projeto. Primeiramente, o gênero é reconhecido por sua relativa simplicidade de execução, o que o torna mais rápido para ser desenvolvido do zero. Além disso, o período curto de três meses exigiu escolhas que otimizassem o tempo de produção, priorizando mecânicas e elementos visuais de implementação ágil. Por fim, a abundância de exemplos e recursos disponíveis na internet, como tutoriais e bibliotecas de ativos, facilitou a criação e a resolução de eventuais problemas técnicos, permitindo o cumprimento das metas estipuladas dentro do cronograma.

2 METODOLOGIA

O principal objetivo deste trabalho é desenvolver do zero um jogo de plataforma 2D através da Unity, *engine* estudada ao longo dos semestres na faculdade, com o propósito de botar em prática os conhecimentos obtidos no curso de jogos digitais.

Nesse capítulo, pode-se acompanhar alguns jogos utilizados como referência para a criação da história e definições de conceitos gerais, além de algumas ferramentas necessários para dar vida a este trabalho. Para isso, foram utilizadas a Unity para o desenvolver o jogo e utilizar alguns *assets* gratuitos da comunidade, o Visual Studio para programação de *scripts*, o Libresprite para a criação de *assets* próprios, o GitHub para salvar o projeto na nuvem e o site ITCH.IO para colocar o jogo para *download* gratuito após a finalização do projeto.

O tempo estimado para o desenvolvimento do “Equitis Adventum” foi de 3 meses, sendo atualizado a cada 2 dias. A equipe é composta por dois membros, o membro Giovanni Marchan Santos ficou responsável por fazer a programação, sonorização, *level design* de *gameplay*, animação e desenvolvimento do projeto. Já o membro Felipe Gonçalves ficou responsável por fazer a roteirização, história do jogo e documentação do TCC.

Em relação a amostra do projeto consistiu em jogadores de diversas faixas etárias e com diferentes níveis de habilidade em jogos, sendo aplicado em uma pesquisa de opinião a fim de coletar informações e *feedbacks* sobre a experiência, a arte e jogabilidade do jogo. Com esses dados pode se observar as análises feitas pelos jogadores para apontar possíveis erros que passaram imperceptíveis durante o desenvolvimento do projeto, além de usar os *feedbacks* para melhorar, implementar e consertar mecânica de jogabilidade.

2.1 Características.

As principais características do jogo desenvolvido são:

- Gênero: Plataforma
- Plataforma alvo: PC
- Direcionamento artístico: *Pixel art* 2D
- Público-alvo: Crianças e adolescentes

- Classificação ESRB: Livre

2.2 Jogos similares.

Para o desenvolvimento do TCC foram usados como inspiração e referência os jogos A Lenda do Herói e Super Mario Bros.

A Lenda do Herói é um jogo que homenageia vários clássicos das eras 8 e 16 *bits*, como The Legend of Zelda e Wonder Boy in Monster Land, e ao mesmo tempo tirar sarro de diversos clichês encontrados no mundo dos *games*. Esse jogo foi desenvolvido em 2012 pela Dumativa em parceria com os Castro Brothers.

Já o clássico **Super Mario Bros** é um jogo de plataforma onde Mario deve passar pelo Reino do Cogumelo e salvar a Princesa Toadstool (mais tarde Princesa Peach) da terrível tartaruga Bowser. Ele foi lançado em 1985 pela Nintendo EAD.

Ambos os jogos serviram como referência para desenvolvimento de mecânicas dos personagens, construção de cenários, desenvolvimento do *level design* de fases e como inspiração para artes gráficas e sonorização de efeitos e trilha sonoras.

2.2.1 Quadros comparativos.

Quadro 1 - Quadro de comparação entre os jogos.

Jogos	Plataforma	Fases	Dificuldades	Preço
Jogo do TCC	Desktop	3	Sem configuração	Gratuito
A Lenda do Herói	Desktop	9	Sem configuração	R\$ 29,99
Super Mario	Nintendo	21	Sem configuração	Gratuito

Fonte: Desenvolvido pelos autores (2024).

O “Equitis Adventum” não se destaca em algum quesito específico, ele tem um estilo de aventura clássica com mecânicas de plataforma e aventura, se apega ao estilo clássico de jogos antigos e possui mecânicas similares aos jogos citados. O seu diferencial é a mecânica de rebater os inimigos e os projéteis ao bloquear com o escudo.

3 CONCEITO

“Equitis Adventum” é um clássico jogo de plataforma 2D em *pixel art*.

3.1 Gênero.

Plataforma 2D.

3.2 Classificação etária.

O público-alvo são as crianças e os adolescentes, mas a classificação etária é livre para todos os públicos.

3.3 Trama principal.

A trama principal acompanha a história do cavaleiro Godfrey que foi enviado em uma missão pelo rei Radagon e a rainha Marika, para combater monstros e recuperar itens que foram roubados do palácio. Nessa missão ele passará por muitos desafios, caminhando por desertos áridos, florestas sombrias e escalando montanhas até ele recuperar os itens roubados.

3.4 *Look & Feel*.

O jogo possui uma aparência 2D simples que transmite o sentimento de simplicidade que jogos de plataforma possuem.

3.5 Visão geral do conceito do jogo.

A história começa no reino de Leyndell em sua era de ouro, em um momento que ninguém ousaria desafiar eles, porém vários monstros começaram a aparecer nas fronteiras do reino, mas isso não incomodava o rei Radagon e sua a rainha Marika pois eles estavam no reino mais protegido do mundo, mesmo assim isso não os impediu de serem atacados pelos monstros. Durante o ataque os itens mais preciosos

do castelo foram roubados e levados para o senhor dos monstros. Desesperados por perderem os itens mais importantes do reino, eles chamaram o cavaleiro Godfrey, pois ele havia derrotado a maioria dos inimigos durante a invasão. Sua missão era eliminar os monstros e recuperar os itens roubados. Com promessas de glória e riquezas, o cavaleiro partiu em sua jornada, carregando o destino do reino em sua lâmina.

4 HISTÓRIA E NARRATIVA

Os próximos tópicos mostram um aprofundamento da história do personagem e do cenário onde o jogo se passa.

4.1 Mundo do jogo.

O jogo se passa em um mundo de fantasia com três ambientes distintos em que o cavaleiro precisa superar os perigos iminentes para completar a sua missão.

Nível do deserto árido

O primeiro desafio de Godfrey é enfrentar o calor escaldante do deserto, com inimigos durões onde o escudo do cavaleiro não funciona para bloquear de seus ataques. Então ele precisa ser esperto e tomar cuidado para que a areia quente e o calor do sol não o exaustem em sua missão.

Nível da floresta fechada

A próxima parada do cavaleiro é na densa floresta, dessa vez os inimigos são astutos e utilizam do cenário para se camuflarem e enganarem o grande cavaleiro, mas Godfrey se mostra perceptivo e é bastante cuidadoso para adentrar na floresta e continuar seguindo em frente.

Nível da grande montanha

Por fim, o cavaleiro se depara com uma grande cadeia montanhosa e nessa montanha os inimigos são inteligentes, então eles se escondem para pegar o Godfrey de surpresa. Logo ele precisa escalá-la com cuidado para conseguir avançar em direção ao seu objetivo e recuperar o tesouro roubado.

4.2 Personagens.

4.2.1 Protagonista:

Cavaleiro Godfrey: Esse é o personagem principal, controlado pelo jogador. Ele é o cavaleiro mais habilidoso do reino que recebeu a missão de derrotar os monstros que invadiram e roubaram os tesouros do reino de Leyndell.

Inimigos terrestres:

Cogumelo ambulante: Este é um dos inimigos terrestres encontrado no nível da floresta sombria. Embora ele seja perigoso, seus movimentos são bem lentos.

Camaleão: O inimigo terrestre está localizado no nível do deserto árido. Ele possui um corpo resistente ao escudo e sua língua é bastante afiada, mas seus movimentos também são lentos.

Inimigos voadores:

Morcego: Já este é um dos inimigos voadores que pode ser encontrado tanto no nível da floresta sombria quanto no nível da grande montanha. Ele sempre está descansando em seu cantinho, mas se o cavaleiro chegar perto demais, ele pode se irritar e parte em direção a ele.

Fantasma: O fantasma é um inimigo que se encontra dentro da grande montanha. Ele possui um corpo translúcido e quando ameaçam acabar com a sua existência, ele acaba desaparecendo por um tempo, enquanto volta para o lugar em que estava.

Inimigos atiradores:

Tronco: Este é um dos inimigos atiradores mais encontrados no nível da floresta sombria. Apesar de possuir pernas, o tronco se mantém parado no lugar aproveitando o ar úmido da floresta, mas ele não gosta de companhia. Se chegar perto demais, ele se sentirá ameaçado e como meio de se defender atirá sementes pontiaguda em direção ao cavaleiro para tentar afastá-lo.

Planta: Por fim esse é um dos inimigos mais encontrado no nível da grande montanha. Ele sempre está aproveitando o ar fresco e, parecido com o tronco, caso se sinta ameaçado este inimigo começa a atirar sementes na direção do cavaleiro

para tentar afastá-lo. Mas caso Godfrey chegue perto de mais, ele se esconde na terra de medo, caso ele se afaste da planta, ela perde o medo e sai de dentro da terra.

4.2.1 Eventos e Ações.

Aconteceu um ataque ao reino e o cavaleiro Godfrey foi enviado em uma missão para recuperar o tesouro roubado do reino. Ao chegar no final da 3 fase, o cavaleiro percebe que está apenas começando a sua longa jornada e se prepara para seguir o seu caminho e completar a sua missão. Sua única motivação é eliminar os monstros e recuperar os itens roubados para salvar o seu reino e receber uma recompensa.

4.3 Visão geral da narrativa.

A narrativa acompanha o cavaleiro Godfrey em sua missão para eliminar os monstros que invadiram e roubaram o reino de Leyndell.

4.3.1 Integração.

O estilo 2D da arte do jogo, desenvolvido no Unity, proporciona uma integração harmoniosa entre a transição dos cenários de fundo e a movimentação dinâmica do personagem durante a *gameplay*. A jogabilidade é fluida, alinhando-se perfeitamente as mecânicas estabelecidas, enquanto o fundo se ajusta aos movimentos do jogador, enriquecendo a experiência visual.

4.3.2 Estrutura e Progressão.

A estrutura narrativa é linear, permitindo que o jogador progrida à medida que avança pelas fases do jogo.

4.3.3 Roteiro.

O “Equitis Adventum” não conta com um roteiro escrito para narrar a história, nem possui NPCs com os quais o jogador possa interagir por meio de diálogos.

4.3.4 Tutoriais e manual do jogo.

O jogo apresenta um tutorial para servir como uma fase zero e nela possui algumas explicações de como funcionam as principais mecânicas do jogador e como os inimigos principais se comportam.

4.4 Cenas e Sequências cinemáticas.

O jogo apresenta transições de cenas do personagem entrando e saindo das cenas para mostrar a progressão do jogador em relação a história do jogo.

Caso o jogador acabe perdendo por ter a sua vida zerada, quando ele escolhe a opção de reiniciar, a tela escurece e o jogo recomeça do começo da fase atual, voltando para a cinemática do personagem entrando na cena.

No final da última fase, o personagem antes de progredir para sua próxima aventura, ele olha para trás e percebe que essa missão só está apenas começando, logo em seguida ele dá um pulo em direção ao horizonte e o jogo termina.

5 GAMEPLAY

O “Equitis Adventum” oferece uma jogabilidade desafiadora, combinando elementos de ação e estratégia em um mundo rodeado de monstros perigosos. Os jogadores assumem o papel do cavaleiro Godfrey, o mais habilidoso cavaleiro do reino de Leyndell.

A jogabilidade é composta em combates com espada, onde os jogadores irão enfrentar uma razoável variedade de inimigos, desde cogumelos ambulantes em uma floresta até fantasmas dentro de uma montanha. Utilizando também um escudo resistente, o jogo incentiva o jogador a adquirir um *timing* certo na hora de atacar e de defender os inimigos no caminho da sua missão.

Nos próximos tópicos pode-se acompanhar quais foram as mecânicas utilizadas para o funcionamento da jogabilidade do jogo.

5.1 Imersão.

Os cenários dinâmicos que se mexe em relação a movimentação do personagem dão a sensação de um mundo vivo e interativo.

A arte gráfica em *pixel art* também proporcionam uma integração visualmente cativantes para o jogador.

As mecânicas de gameplay precisas e desafios progressivos o mantêm engajado e concentrado para passar de fase.

A escolha das trilhas sonoras dinâmicas e responsivas se encaixam com o ambiente.

Todos esses elementos ajudaram a criar uma experiência imersiva e satisfatória.

5.2 Estrutura de missões e desafios.

Todas as fases utilizam a mesma estrutura, o jogador precisa ir da esquerda para a direita enquanto ele elimina os monstros e passa por obstáculos no caminho. Caso consiga chegar até a extrema direita do nível atual com pelo menos uma vida, ele pode passar para a próxima fase.

Se o jogador perder todas as três vidas ele morrerá, tendo que recomeçar a fase atual do começo até conseguir passar com vida.

5.3 Objetivos do jogo.

O objetivo principal do jogador é passar pelos três diferentes níveis enquanto elimina os monstros que encontrar em seu caminho.

5.4 Fluxo de jogo.

A medida com que o jogador vai avançando nas fases 1,2 e 3, respectivamente, o nível de dificuldade dos inimigos aumenta, fazendo com que o jogador tenha que evoluir suas habilidades.

5.5 Mecânica do Jogo.

O jogo “Equitis Adventum” possui mecânicas de combate que desafiam o jogador e testam constantemente suas habilidades e estratégias.

Tanto o personagem principal Godfrey quanto os inimigos do jogo têm mecânicas básicas de movimentação, de ataque e de defesa.

5.5.1 Regras implícitas e explícitas do jogo.

Durante o desenvolvimento do “Equitis Adventum”, as regras implícitas e explícitas foram pensadas de forma que o combate garantisse uma experiência divertida para os jogadores.

Regras explícitas:

Movimentação: O jogador pode se mover o personagem usando as teclas direcionais Seta para a esquerda e Seta para a Direita ou A e D e pular na barra de espaço.

Ataques: Para realizar os ataques, o jogador precisa pressionar o botão esquerdo do mouse enquanto estiver no chão ou enquanto estiver no ar após pular.

Defesa: Para defender os ataques dos inimigos, o jogador precisa segurar o botão direito do mouse enquanto estiver no chão.

Regras implícitas:

Regeneração da estamina do escudo: Se o jogador diminuir a barra de estamina do escudo usando a mecânica de defesa, enquanto o jogador não segurar o botão direito do mouse para defender, a estamina do escudo começa a se regenerar.

Invulnerabilidade após tomar dano: Se o jogador tomar um dano do inimigo, ele entrará em um estado de invulnerabilidade durante um curto período em que o personagem consegue passar diretamente pelos inimigos e não tomará dano.

Invulnerabilidade após inimigo tomar dano: Se o inimigo tomar um dano do jogador, ele entrará em um breve estado de invulnerabilidade onde o personagem consegue passar diretamente pelos inimigos e não tomará dano.

Mecânica de rebote: Sempre que algum personagem sofre um dano, ele aplica, ou não, uma força para a direção oposta do ataque para afastar o inimigo do personagem e vice e versa.

5.5.2 Física.

Em “Equitis Adventum”, a física é um elemento bem predominante e está ligado com a movimentação de todos os personagens e com a mecânica de rebote dos ataques. Ela também é utilizada para rebater o projétil dos inimigos atiradores, mudando sua direção para eles quando o personagem os bloqueia com o escudo.

5.5.3 Movimentação dos personagens.

Personagem principal:

Movimentação: A movimentação do personagem pode ser realizada de duas maneiras, ou pelas teclas A e D ou pelas setas esquerda e direita do teclado.

Ataque: Para atacar, o jogador só precisa pressionar o botão esquerdo do mouse toda vez que ele precisar atacar. Para direcionar o ataque, ele precisa estar virado para a direção do inimigo.

Defesa: Para defender, o jogador só precisa segurar o botão direito do mouse toda vez que ele precisar defender. Para bloquear o ataque do inimigo, o personagem precisa estar virado em direção do inimigo.

Inimigos:

Movimentação: Os inimigos terrestres e voadores permanecem parados enquanto o jogador está longe da distância do alcance do inimigo. Caso o personagem entre nessa distância, o inimigo se move em direção a ele. Os inimigos atiradores permanecem parados o tempo todo, mesmo se o jogador estiver na linha de visão do ataque deles.

5.5.4 Gatilhos e ações.

Mecânica de seleção: Ao iniciar o jogo, a tela de menu inicial é exibida, exigindo interação do jogador para avançar e começar o jogo ou alterar as opções do volume. Essa mecânica também acontece quando o jogador pausa o jogo ou o personagem acaba morrendo, mostrando uma tela exigindo interação do jogador para recomeçar o jogo, voltar para o menu principal ou alterar as opções do volume.

Mecânica de dano: Ao sofrer algum dano, o personagem muda de cor para sinalizar ao jogador que tal personagem sofreu algum dano.

Mecânica de morte: Caso o inimigo zere sua quantidade de vidas, ele é destruído e explode em partículas vermelhas para sinalizar ao jogador que o inimigo está morto. Caso o personagem zere sua quantidade de vidas, ele inicia uma animação de morte e uma tela de você morreu é exibida, exigindo interação do jogador para recomeçar o jogo ou voltar para o menu principal.

Mecânica de troca de fase: Quando o personagem principal consegue chegar na parte mais à direita da fase, acontece uma cinemática onde ele anda para a direita até estar fora do mapa enquanto a tela vai escurecendo para trocar de fase.

5.5.5 Mecânica de Combate.

Quando as fases começam, os inimigos ficam parados e eles só começam a atacar o jogador caso ele chegue perto demais. O primeiro que zerar o valor de vida de seu oponente ganha o combate.

5.6 Projeto de Fases (*Level Design*).

Fase tutorial: O tutorial.

Sinopse: Antes do cavaleiro Godfrey começar a se aventurar para dar início a sua missão, ele precisa praticar um pouco suas habilidades em uma área com os três tipos de inimigo que o jogador irá deparar ao longo da jornada.

Material introdutório: Uma *cut-scene* do cavaleiro entrando na cena pelo lado esquerdo da fase.

Objetivos da fase:

Fazer o jogador se familiarizar com os comandos de movimentação do personagem.

Derrotar os inimigos que estão no caminho do cavaleiro.

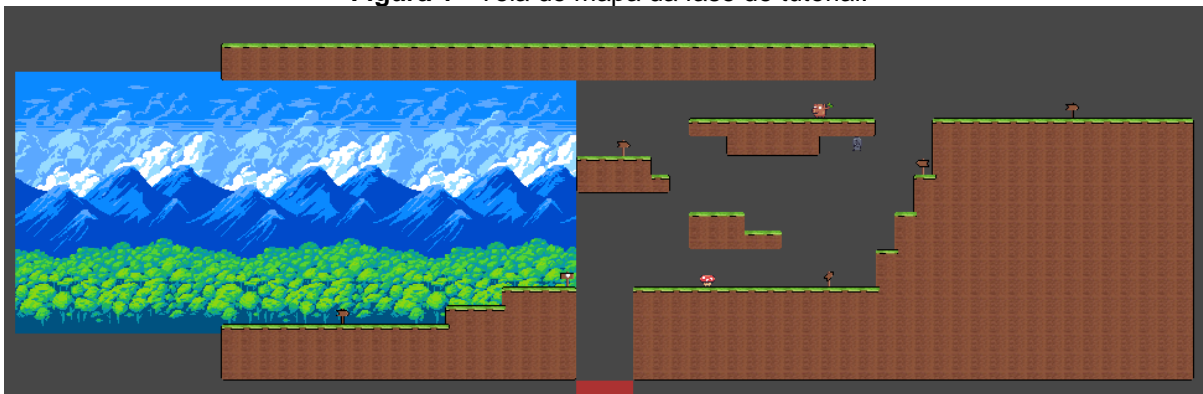
Ir para a direita até o personagem progredir para a próxima fase.

Descrição visual: O tutorial é um lugar tranquilo e cheio de placas com avisos que ajudam o cavaleiro a se lembrar de como suas habilidades físicas funcionam. No caminho, ele encontra um pequeno morro e três inimigos perdidos que não apresentam nenhuma dificuldade, fazendo que o cavaleiro se recorde e utilize de suas artimanhas para superar esse pequeno desafio e começar a sua aventura.

Descrição física: Terreno aberto, um chão de grama, um céu azul, três inimigos, um pequeno morro e alguns pedaços de chão flutuantes.

Mapa da fase:

Figura 1 - Tela do mapa da fase de tutorial.



Fonte: Autoria própria (2024).

Localização de gatilhos: Quando o personagem chegar na extrema direita do mapa desencadeia uma *cut-scene* do cavaleiro saindo da cena pelo lado direito da fase, indicando que ele está indo em direção para a próxima fase.

Walkthrough:

Início: Após a *cut-scene* inicial, o jogador está livre para andar pelo mapa e derrotar os inimigos que estão no seu caminho.

Conclusão: Ao chegar na extrema direita do mapa, o jogador avança automaticamente para a próxima fase do jogo.

Material de Fechamento: Ao completar a fase, o jogador desencadeará uma *cut-scene* e irá para a próxima fase automaticamente.

Fase 1: O deserto árido.

Sinopse: O cavaleiro Godfrey começa a se aventurar para dar início a sua missão, onde ele precisa derrotar os primeiros inimigos que aparecem em seu caminho para começar a fazer sua missão.

Material introdutório: Uma *cut-scene* do cavaleiro entrando na cena pelo lado esquerdo da fase.

Objetivos da fase:

Derrotar os inimigos que estão no caminho do cavaleiro.

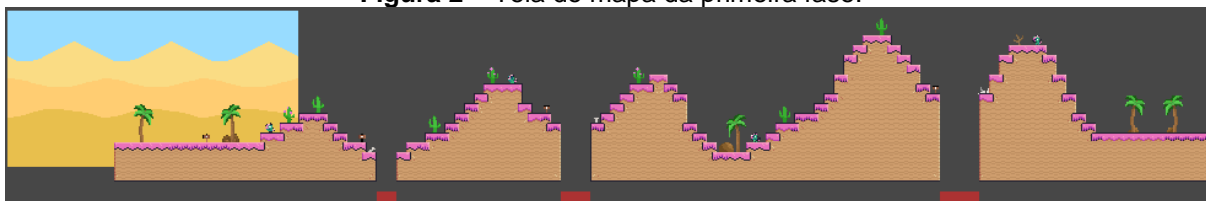
Ir para a direita até o personagem progredir para a próxima fase.

Descrição visual: O deserto árido é um lugar quente e árido, com uma variedade de plantas espinhosas e palmeiras altas. No meio da areia o cavaleiro nota vários crânios de possíveis criaturas que não aguentaram com o calor e a seca do local.

Descrição física: Terreno arenoso e chão rosado, várias montanhas, cactos com e sem flores, além de palmeiras verdes para mostrar a beleza do lugar.

Mapa da fase:

Figura 2 – Tela do mapa da primeira fase.



Fonte: Autoria própria (2024).

Localização de gatilhos: Quando o personagem chegar na extrema direita do mapa desencadeia uma *cut-scene* do cavaleiro saindo da cena pelo lado direito da fase, indicando que ele está indo em direção para a próxima fase.

Walkthrough:

Início: Após a *cut-scene* inicial, o jogador está livre para andar pelo mapa e derrotar os inimigos que estão no seu caminho.

Conclusão: Ao chegar na extrema direita do mapa, o jogador avança automaticamente para a próxima fase do jogo.

Material de Fechamento: Ao completar a fase, o jogador desencadeará uma *cut-scene* e irá para a próxima fase automaticamente.

Fase 2: A floresta sombria.

Sinopse: Depois de passar pelo deserto árido, o cavaleiro avança para a floresta sombria, uma floresta úmida e densa, cheia de árvores grandes que podem esconder a presença de inimigos sorrateiros. Mas isso não é problema para o Godfrey, pois sua visão aguçada fará com que ele jamais seja pego de surpresa pelos inimigos furtivos.

Material introdutório: Uma *cut-scene* do cavaleiro entrando na cena pelo lado esquerdo da fase.

Objetivos da fase:

Derrotar os inimigos escondidos que estão no caminho do cavaleiro.

Ir para a direita até o personagem progredir para a próxima fase.

Descrição visual: A floresta sombria é um lugar escuro e cheio de inimigos sorrateiros, com uma variedade de musgos e árvores grandes que escondem a presença dos inimigos.

Descrição física: Uma floresta fechada, o terreno de terra com o chão cheio de musgo, vegetação rasteira, várias árvores grandes que mostram um clima sombrio e com um ar de sempre ter alguém te observando na escuridão da mata.

Mapa da fase:

Figura 3 - Tela do mapa da segunda fase.



Fonte: Autoria própria (2024).

Localização de gatilhos: Quando o personagem chegar na extrema direita do mapa desencadeia uma *cut-scene* do cavaleiro saindo da cena pelo lado direito da fase, indicando que ele está indo em direção para a última fase do jogo.

Walkthrough:

Início: Após a *cut-scene* inicial, o jogador está livre para andar pelo mapa e derrotar os inimigos escondidos que estão no seu caminho.

Conclusão: Ao chegar na extrema direita do mapa, o jogador avança automaticamente para a próxima fase do jogo.

Material de Fechamento: Ao completar a fase, o jogador desencadeará uma *cut-scene* e irá para a próxima e última fase automaticamente.

Fase 3: A grande montanha.

Sinopse: A terceira e última fase da aventura do cavaleiro Godfrey onde ele terá que escalar uma grande montanha para chegar no topo e ter uma visão clara do reino para encontrar o covil do senhor dos monstros e recuperar o tesouro roubado.

Material introdutório: Uma *cut-scene* do cavaleiro entrando na cena pelo lado esquerdo da fase.

Objetivos da fase:

Derrotar os inimigos que estão no caminho do cavaleiro.

Subir a montanha até o personagem chegar no topo dela.

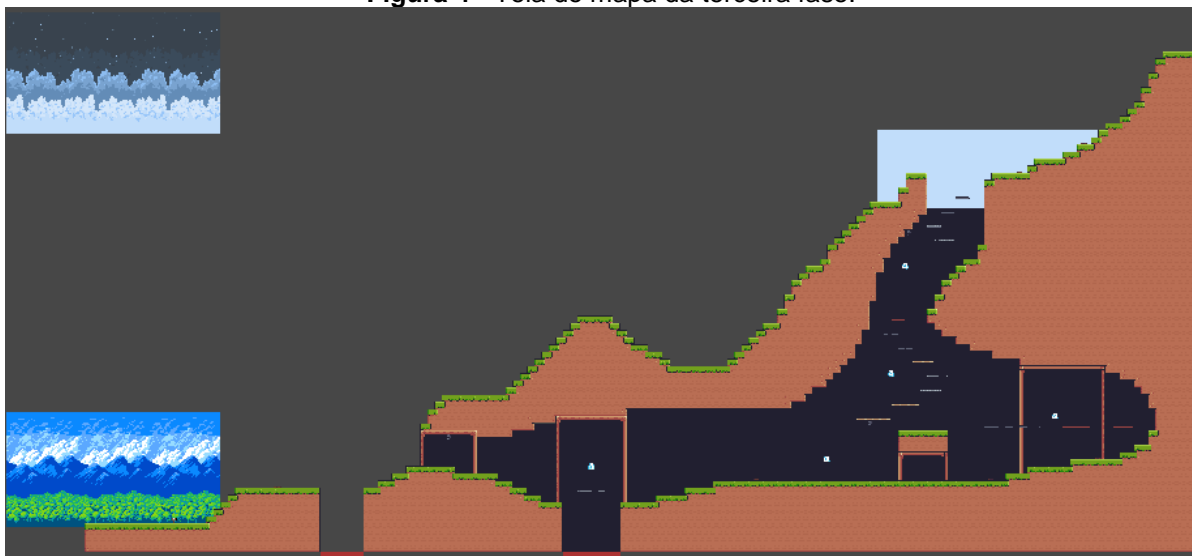
Descrição visual: A grande montanha é um lugar aberto e cheio de caminhos escondidos, com a presença de inimigos perigosos. Ela apresenta uma variedade de plataformas flutuantes e paredes falsas que escondem o interior sombrio e assustador da montanha.

Descrição física: Uma combinação das duas fases anteriores, um terreno montanhoso com uma vegetação verde, do lado de fora da montanha há um céu claro

e logo depois uma noite estrelada. Já do lado de dentro da montanha há uma escuridão e uma sensação de mistério paira sobre o ar, com uma variedade de plataformas flutuantes e inimigos desafiadores.

Mapa da fase:

Figura 4 - Tela do mapa da terceira fase.



Fonte: Autoria própria (2024).

Localização de gatilhos: Quando o personagem chegar no topo da montanha na parte mais à direita do mapa, ele desencadeia uma *cut-scene* do cavaleiro indo para a beira da montanha, ele olha para trás por um breve período e logo em seguida Godfrey salta em direção ao abismo, saindo da cena pelo lado direito da fase, indicando que ele está indo em direção ao covil do senhor dos monstros para completar a sua missão.

Walkthrough:

Início: Após a *cut-scene* inicial, o jogador está livre para andar pelo mapa e derrotar os inimigos que estão no seu caminho.

Conclusão: Ao chegar no topo da montanha, o jogador avança automaticamente para a *cut-scene* final do jogo.

Material de Fechamento: Ao completar a fase, o jogador desencadeará uma

cut-scene e irá levar o jogador para o menu inicial automaticamente, finalizando o jogo.

5.7 Projeto de Interface.

5.7.1 Sistema Visual.

a) HUD (*Head-Up Display*):

Sistema de vida: Localizado no canto superior esquerdo da tela, representado por três corações vermelhos. Esses corações ficam vazios conforme o jogador recebe dano dos inimigos, oferecendo uma indicação visual imediata da saúde do personagem.

Barra de estamina do escudo: Posicionada um pouco abaixo do sistema de vida, é um retângulo marrom que representa o tempo em que o personagem consegue usar o escudo para se defender dos ataques dos inimigos. Conforme o jogador mantém o botão direito do mouse pressionado, a barra vai diminuindo. Se a barra diminuir por completo, o personagem ficará incapaz de defender enquanto a barra não aumenta por completo. Caso o jogador solte o botão antes da barra diminuir por completo, ele ainda será capaz de defender com o escudo e a barra aumentará conforme o tempo passa.

b) Menu *in game*:

Tela de *pause*: O jogo possui uma tela tradicional de *pause*, quando o jogador pressiona a tecla ESC do teclado, o jogo congela o tempo e aparece uma tela de *pause* com as seguintes opções:

- **Botão de continuar:** Caso o jogador selecione o este botão, a tela de *pause* desaparece, o jogo descongela o tempo e o jogo segue normalmente.

- **Botão de recomeçar:** Caso o jogador selecione o este botão, a tela de *pause* escurece e o jogo recomeça a fase atual desde o começo, desde a *cut-scene* do personagem entrando na cena.

- **Botão de voltar ao menu:** Caso o jogador selecione o este botão, a tela de *pause* escurece e o jogador é levado de volta para a tela de menu inicial do jogo.

- **Engrenagem de opções de volume:** Caso o jogador selecione o este botão, a tela de *pause* desaparece e é revelada a tela de opções.

Tela de opções: O jogo possui uma tela de opções onde o jogador pode alterar as seguintes configurações de jogo:

- **Volume das músicas:** O jogador pode ajustar o volume das músicas no jogo.

- **Volume dos efeitos sonoros:** O jogador pode ajustar o volume dos efeitos sonoros no jogo.

- **Botão de voltar:** Caso o jogador selecione o este botão, os volumes que foram alterados são salvos e a tela de opções desaparece, retornando para a tela de *pause*.

Tela de morte: O jogo possui uma tela de morte que só aparece caso o personagem zere a sua quantidade de vida. Nesta tela o jogador possui a opção de escolher pressionar um dos dois botões já apresentados na tela de opções: **Botão de recomeçar** ou **Botão de voltar ao menu**.

c) Câmera:

A câmera do jogo é em terceira pessoa e ela acompanha os movimentos do personagem, tendo uma visão ampla do que está acontecendo na fase. Dessa maneira o jogador consegue ver quais são os perigos vêm pela frente, facilitando a execução de suas ações durante a jogatina.

5.7.2 Sistema de Controle.

Interação com o Mouse: Em todas as telas, incluindo menu, tela de *pause* e de morte, a interação é feita exclusivamente com o *mouse*. O uso desse sistema de controle garante uma experiência de usuário intuitiva e fácil de usar.

5.7.3 Fluxo de Telas.

Descrições de telas

Menu principal

Jogar: Mostra a tela de seleção de nível.

Opções: Mostra as opções para aumentar ou diminuir o volume da música e do efeito sonoro no menu principal. O jogador usa o *mouse* para deslizar o *slider* do volume e deixar o som na altura que mais lhe agrada.

Créditos: Mostra os nomes e as contribuições da equipe de desenvolvimento.

Sair: Fechar o jogo e retornar para a área de trabalho.

Tela de seleção de nível

Esta tela mostra quais são os níveis que o jogador pode escolher para jogar, ao escolher uma o jogador é levado para a fase escolhida.

Tela de jogo

Mostra o ambiente do jogo onde o jogador controla o cavaleiro Godfrey em sua missão. Inclui o HUD com o sistema de vida e a barra de estamina do escudo.

Tela de pause

Esta tela aparece quando o jogador aperta a tecla ESC do teclado, pausando o jogo e mostrando as opções de continuar, recomeçar e voltar ao menu principal. O jogador usa o *mouse* para selecionar a opção que ele deseja.

Tela de opções

Mostra as opções para aumentar ou diminuir o volume da música e do efeito sonoro já descritos em opções no menu inicial.

Tela de morte

Ela aparece quando o personagem morre no jogo, mostrando as opções de recomeçar o jogo ou de voltar para o menu principal. O jogador usa o *mouse* para selecionar a opção que ele deseja realizar.

5.8 Projeto de Som.

5.8.1 Descrição geral.

O sistema de áudio em “Equitis Adventum” consiste em trilhas sonoras e efeitos sonoros produzidos por membros das comunidades do Unity, do Pixabay e do OpenGameArt. Elas foram utilizadas de forma retratada no momento, no caso dos efeitos sonoros, e coesas com a fase atual, no caso dos loops de áudio da trilha sonora.

5.8.2 Trilhas sonoras.

- a) Ambiente (“8-bit-game-158815”, tocada no menu inicial / “8-bit-hero-251426”, tocada na fase tutorial / “8Bit Music – Track 8”, tocada na fase 1 / “8Bit Music – Track 5”, tocada na fase 2 / “8Bit Music – Track 10”, tocada na fase 3).

5.8.3 Efeitos Sonoros.

- a) Movimentação (“stepdirt_1”, “stepdirt_2”, “stepdirt_3”, “stepdirt_4”, “stepdirt_5”, “stepdirt_6”, “stepdirt_7”, “stepdirt_8”, são tocados aleatoriamente quando o personagem anda / “cartoon-jump-6462”, “sfx_jump_07-80241”, são tocados de forma aleatória quando o personagem pula).
- b) Armas (“mixkit-dagger-woosh-1487”, tocado quando o jogador utiliza a espada durante o ataque / “steampunk-mechanical-gadget-188052”, tocado quando o jogador utiliza o escudo durante a defesa / “swing-whoosh-metal-hit-4-234618”, “swing-whoosh-metal-hit-10-234626”, são tocados de forma aleatória quando o inimigo é bloqueado pelo escudo / “bow-and-arrow-shoot-sound-effect-2-239699”, “bow-and-arrow-shoot-sound-effect-1-239700”, são tocados de forma aleatória quando o inimigo atirador dispara seus projéteis).
- c) Dano (“Male-Ouch-Sound”, “Male_Hurt-Sound”, são tocados de forma aleatória quando o personagem toma algum dano / “Hurt_Sound”, tocado quando qualquer inimigo sofre algum dano).
- d) Morte (“Male-Death-Sound”, é tocado quando o personagem zera a sua quantidade de vida / “Explode-Enemy-Sound” é tocado quando qualquer inimigo zera a sua quantidade de vida).

- e) Interface (“click-button-140881”, é tocado quando o jogador passa com o mouse por cima de qualquer opção que tenha em qualquer tela / “mouse-click-153941” é tocado quando o jogador clica com o botão esquerdo do mouse em alguma opção no menu inicial, na tela de *pause* ou na tela de morte / “game-start-6104”, é tocado quando o jogador seleciona com o botão esquerdo do mouse algum nível na tela de seleção de níveis).

5.9 Sistema de Inteligência Artificial.

5.9.1 Inimigos.

Os inimigos no jogo “Equitis Adventum” são controlados por uma Inteligência Artificial (IA) desenvolvida para movimentar os três tipos diferente de inimigos. Utilizando os *scripts* do Unity, cada inimigo tem uma forma de detecção diferente do alcance entre o inimigo e o personagem para trocar as animações e atacar o jogador. O jogo também utiliza a tecnologia da máquina de estados finitos para controlar a movimentação dos inimigos.

Cogumelo Ambulante:

Movimentação: O Cogumelo utiliza uma área circular invisível que detecta se o jogador entrou nessa área, caso isso aconteça o inimigo se move em direção ao personagem, caso contrário ele se mantém parado no lugar.

Camaleão:

Movimentação: O Camaleão utiliza uma área circular invisível que detecta se o jogador entrou nessa área, caso isso aconteça o inimigo se move em direção ao personagem até uma certa, caso contrário ele se mantém parado no lugar.

Ataque: Quando ele está a uma certa distância do jogador, o camaleão para de se mover e executa um ataque. Após o ataque, o inimigo volta a se movimentar.

Pós-ataque: Após o ataque, ele espera um tempo antes de poder atacar novamente.

Morcego:

Movimentação: O Morcego utiliza um raio invisível que detecta se o jogador está a uma certa distância do inimigo, caso isso aconteça o inimigo se move em

direção ao personagem, caso contrário ele para de se mover em direção ao personagem e se move voltando para o lugar em que estava.

Fantasma:

Movimentação: O Morcego utiliza um raio invisível que detecta se o jogador está a uma certa distância do inimigo, caso isso aconteça o inimigo se move em direção ao personagem, caso contrário ele para de se mover em direção ao personagem e se move voltando para o lugar em que estava.

Tronco:

Movimentação: Apesar do Tronco ter pernas, ele se mantém parado no lugar.

Deteccção para atacar: O inimigo utiliza um raio invisível que detecta se o jogador está a uma certa distância do inimigo, caso isso aconteça ele executa o ataque.

Ataque: Quando ele está a uma certa distância do jogador, o tronco dispara um projétil em direção ao jogador.

Pós-ataque: Após o ataque, ele espera um tempo antes de poder atacar novamente.

Planta:

Movimentação: A planta não se move como os demais inimigos.

Deteccção para se esconder: este inimigo utiliza uma área circular invisível que detecta se o jogador entrou nessa área, caso isso aconteça o inimigo se esconde do personagem entrando debaixo da terra, ficando invulnerável aos ataques do jogador. Caso o personagem saia dessa deteccção, a planta sai da terra e volta para a superfície.

Deteccção para atacar: O inimigo utiliza um raio invisível que detecta se o jogador está a uma certa distância do inimigo, caso isso aconteça ele executa o ataque.

Ataque: Quando ele está a uma certa distância do jogador, a planta dispara um projétil em direção ao jogador.

Pós-ataque: Após o ataque, ele espera um tempo antes de poder atacar novamente.

6 ARTE

Nesse capítulo, pode-se acompanhar algumas artes que foram utilizadas para a criação do personagem, dos inimigos e do cenário do jogo.

6.1 Guia de cores e estilos gráficos.

O ambiente do jogo possui um tom mais claro nas primeiras fases, dando uma sensação de alívio e tranquilidade para os jogadores. Nas próximas fases o jogo possui tons mais escuros, passando a sensação de angústia, com o cenário feitos para deixar o jogador tenso e que ajudam a esconder a presença dos inimigos, esperando a hora de atacar quando o jogador estiver desprevenido.

O estilo artístico é o 2D clássico se encaixa bem com o cenário apresentado, tendo cores vivas nos ambientes abertos, e depois ele apresenta cores mortas, com cenários fechados, representando um ambiente preso e sufocante.

6.2 *Asset List*.

6.2.1 Personagens.

Figura 5 - Cavaleiro Godfrey.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 6 – Morcego.



Fonte: Frog (2024).

Figura 7 – Camaleão.



Fonte: Frog (2024).

Figura 8 – Planta.



Fonte: Autoria própria (2024).

Figura 9 – Fantasma.



Fonte: Frog (2024).

Figura 10 - Cogumelo ambulante.



Fonte: Frog (2024).

Figura 11 - Tronco.



Fonte: Frog (2024).

6.2.2 Ambientes.

Figura 12 - Cenário do tutorial.



Fonte: adaptado pelo autor (2024).

Figura 13 - Cenário da primeira fase.



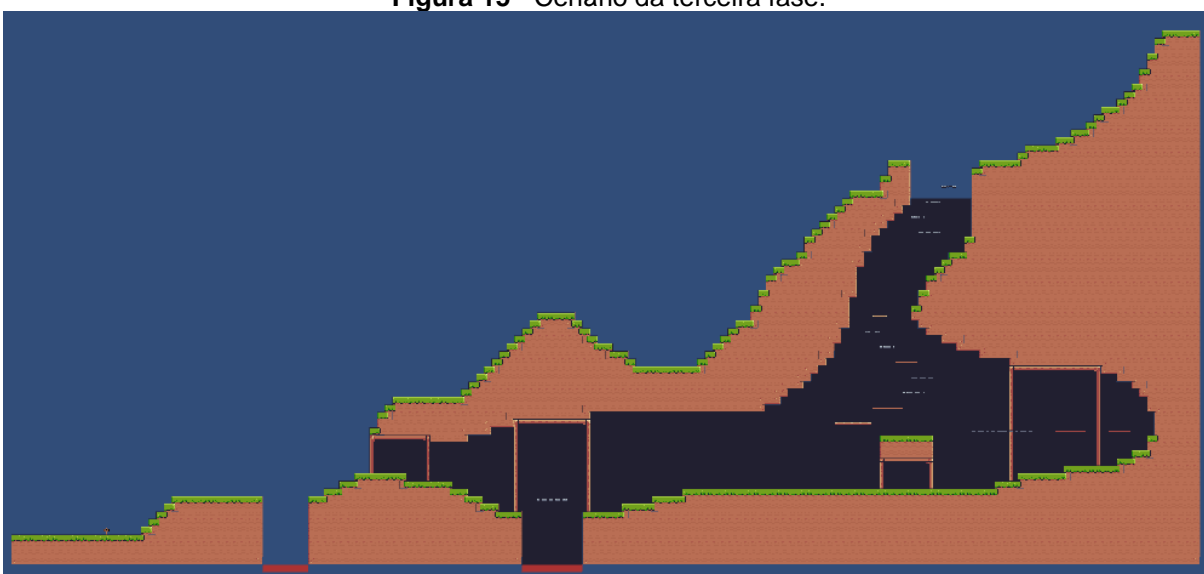
Fonte: Lifty (2021), Frog (2024), adaptado pelo autor (2024).

Figura 14 - Cenário da segunda fase.



Fonte: CraftPix (2019), adaptado pelo autor (2024).

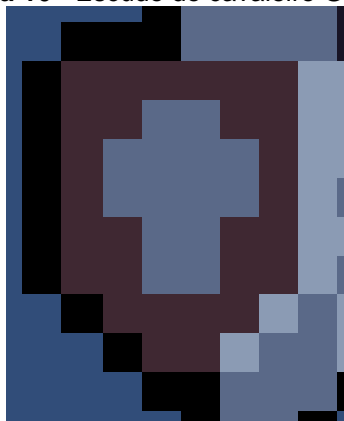
Figura 15 - Cenário da terceira fase.



Fonte: Frog (2024), adaptado pelo autor (2024).

6.2.3 Equipamentos.

Figura 16 - Escudo do cavaleiro Godfrey.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

6.2.4 Armas.

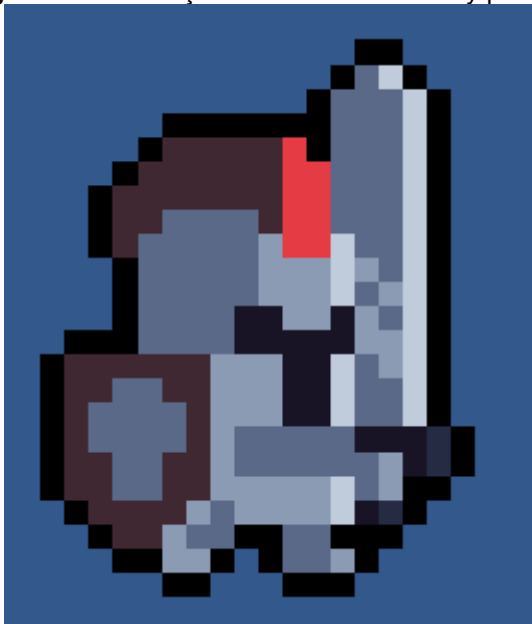
Figura 17 - Espada do cavaleiro Godfrey.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

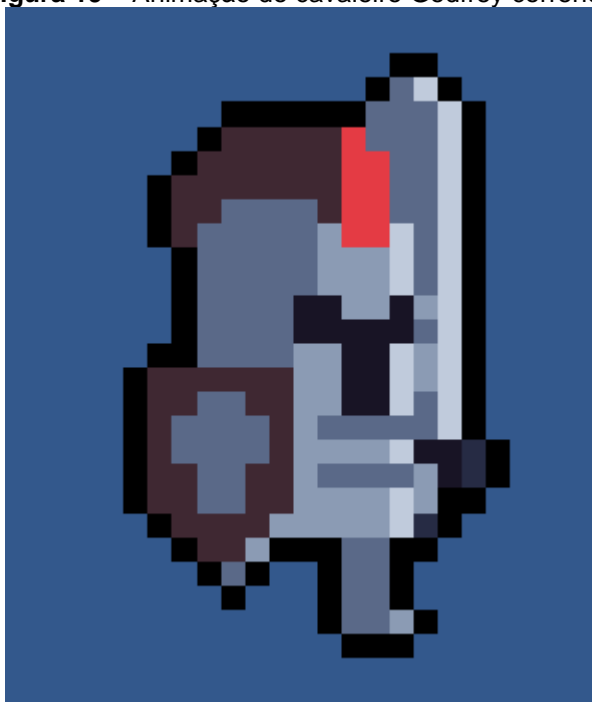
6.2.5 Animações.

Figura 18 - Animação do cavaleiro Godfrey parado.



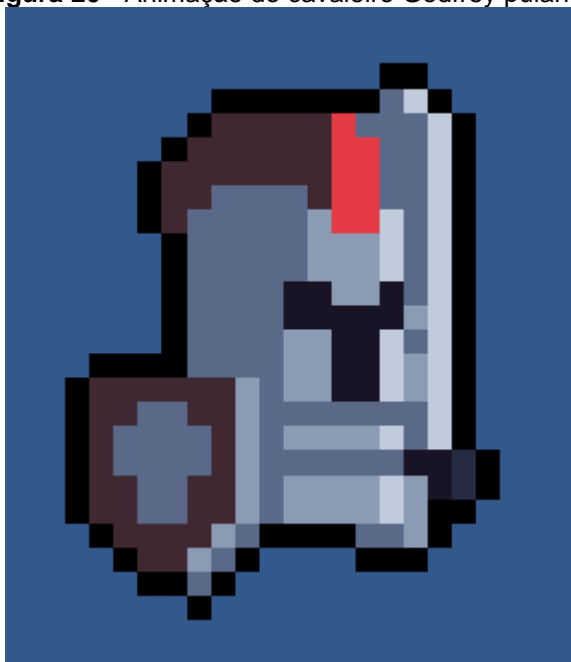
Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 19 – Animação do cavaleiro Godfrey correndo.



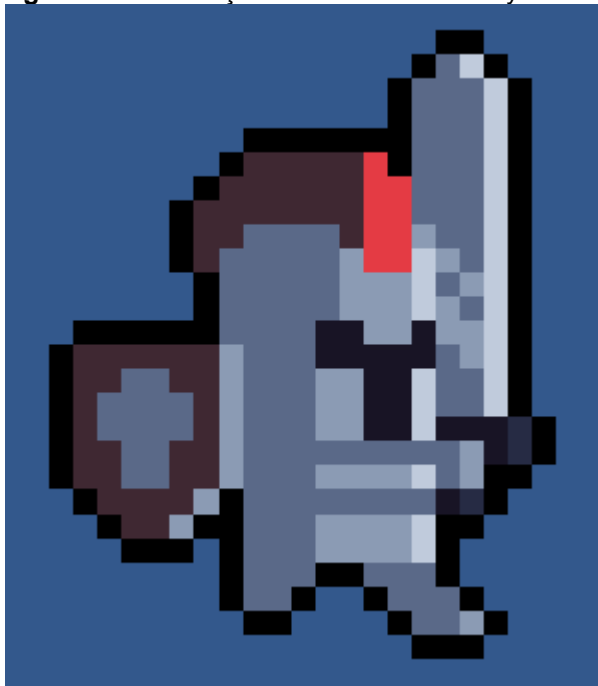
Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 20 - Animação do cavaleiro Godfrey pulando.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 21 – Animação do cavaleiro Godfrey caindo.



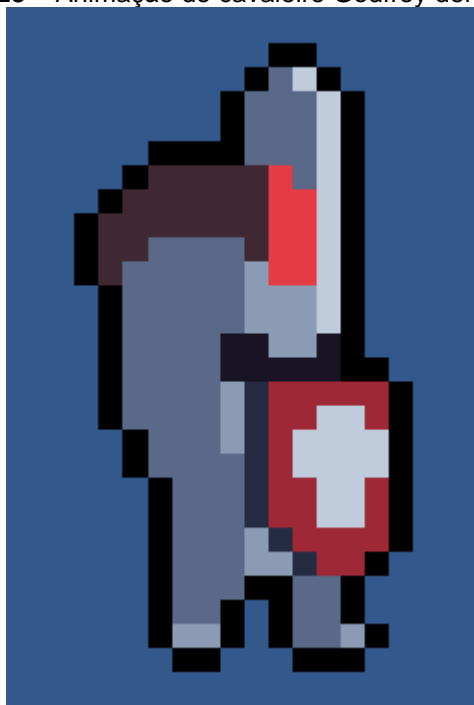
Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 22 - Animação do cavaleiro Godfrey atacando.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 23 – Animação do cavaleiro Godfrey defendendo.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 24 - Animação do cavaleiro Godfrey morto.



Fonte: RgsDev (2020), adaptado pelo autor (2024).

Figura 25 - Animação do morcego parado.



Fonte: Frog (2024).

Figura 26 - Animação do morcego acordando.



Fonte: Frog (2024).

Figura 27 - Animação do morcego voando.



Fonte: Frog (2024).

Figura 28 - Animação do morcego retornando à origem.



Fonte: Frog (2024).

Figura 29 - Animação do morcego tomando dano.



Fonte: Frog (2024).

Figura 30 - Animação do camaleão parado.



Fonte: Frog (2024).

Figura 31 - Animação do camaleão andando.



Fonte: Frog (2024).

Figura 32 - Animação do camaleão atacando.



Fonte: Frog (2024).

Figura 33 - Animação do camaleão tomando dano.



Fonte: Frog (2024).

Figura 34 - Animação da planta parada.



Fonte: Autoria própria (2024).

Figura 35 - Animação da planta disparando.



Fonte: Autoria própria (2024).

Figura 36 - Animação da planta se escondendo.



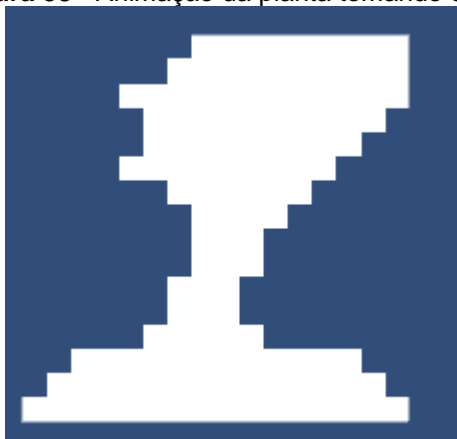
Fonte: Autoria própria (2024).

Figura 37 - Animação da planta aparecendo.



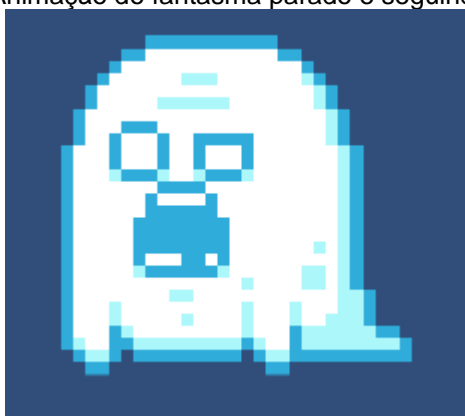
Fonte: Autoria própria (2024).

Figura 38 - Animação da planta tomando dano.



Fonte: Autoria própria (2024).

Figura 39 - Animação do fantasma parado e seguindo o cavaleiro.



Fonte: Frog (2024).

Figura 40 - Animação do fantasma desaparecendo.



Fonte: Frog (2024).

Figura 41 - Animação do fantasma aparecendo.



Fonte: Frog (2024).

Figura 42 - Animação do fantasma tomando dano.



Fonte: Frog (2024).

Figura 43 - Animação do cogumelo ambulante parado.



Fonte: Frog (2024).

Figura 44 - Animação do cogumelo ambulante andando.



Fonte: Frog (2024).

Figura 45 - Animação do cogumelo ambulante tomando dano.



Fonte: Frog (2024).

Figura 46 - Animação do tronco parado.



Fonte: Frog (2024).

Figura 47 - Animação do tronco disparando.



Fonte: Frog (2024).

Figura 48 - Animação do tronco tomando dano.



Fonte: Frog (2024).

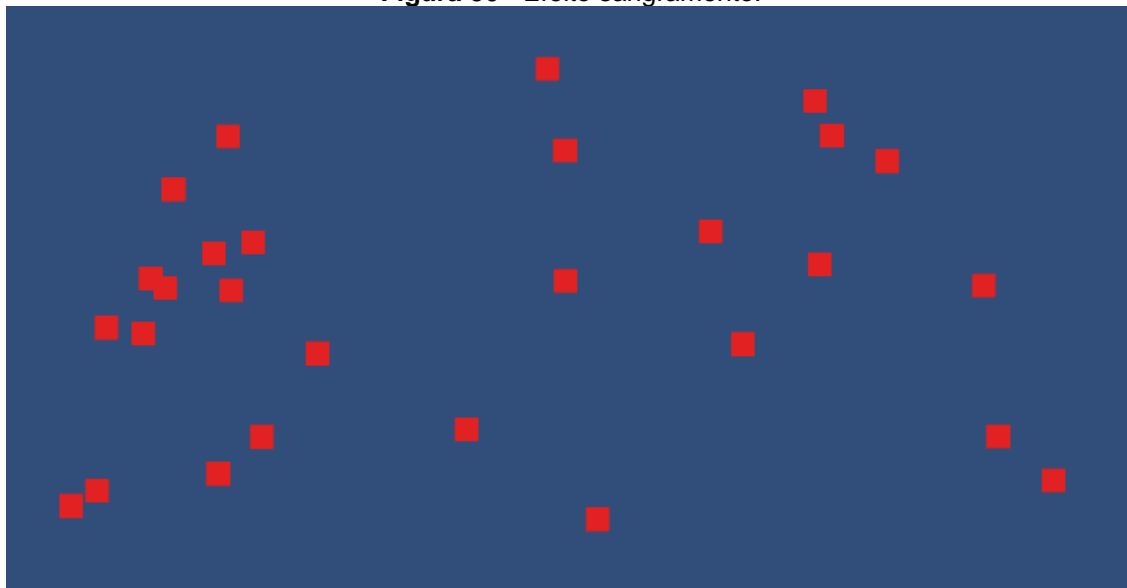
6.2.6 Efeitos.

Figura 49 - Efeito de poeira do solo.



Fonte: Frog (2024).

Figura 50 - Efeito sangramento.



Fonte: Autoria própria (2024).

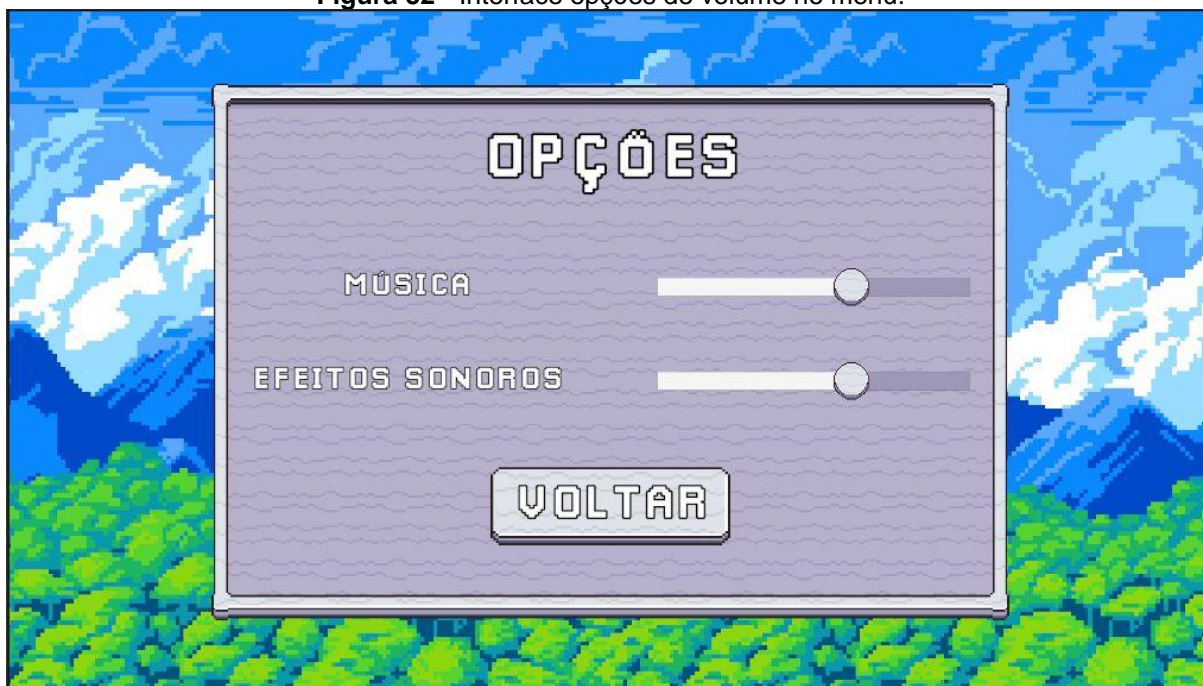
6.2.7 Interfaces.

Figura 51 - Interface do menu inicial.



Fonte: Narik (2020), adaptado pelo autor (2024).

Figura 52 - Interface opções de volume no menu.



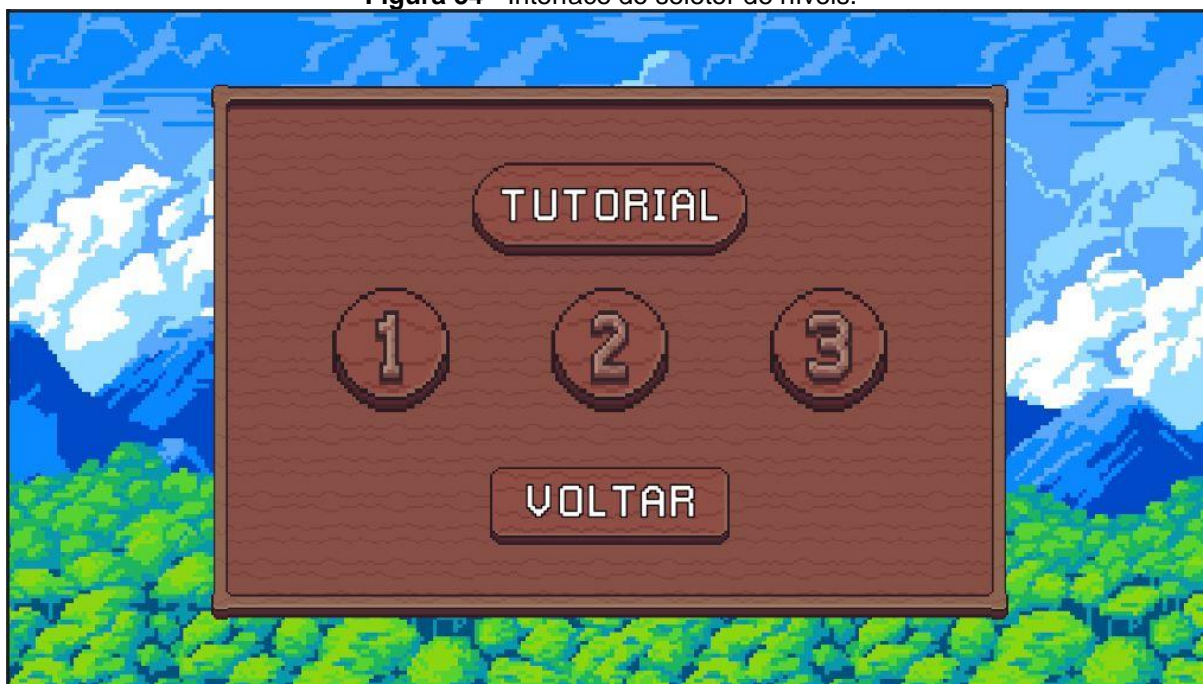
Fonte: Narik (2020), adaptado pelo autor (2024).

Figura 53 - Interface de créditos.



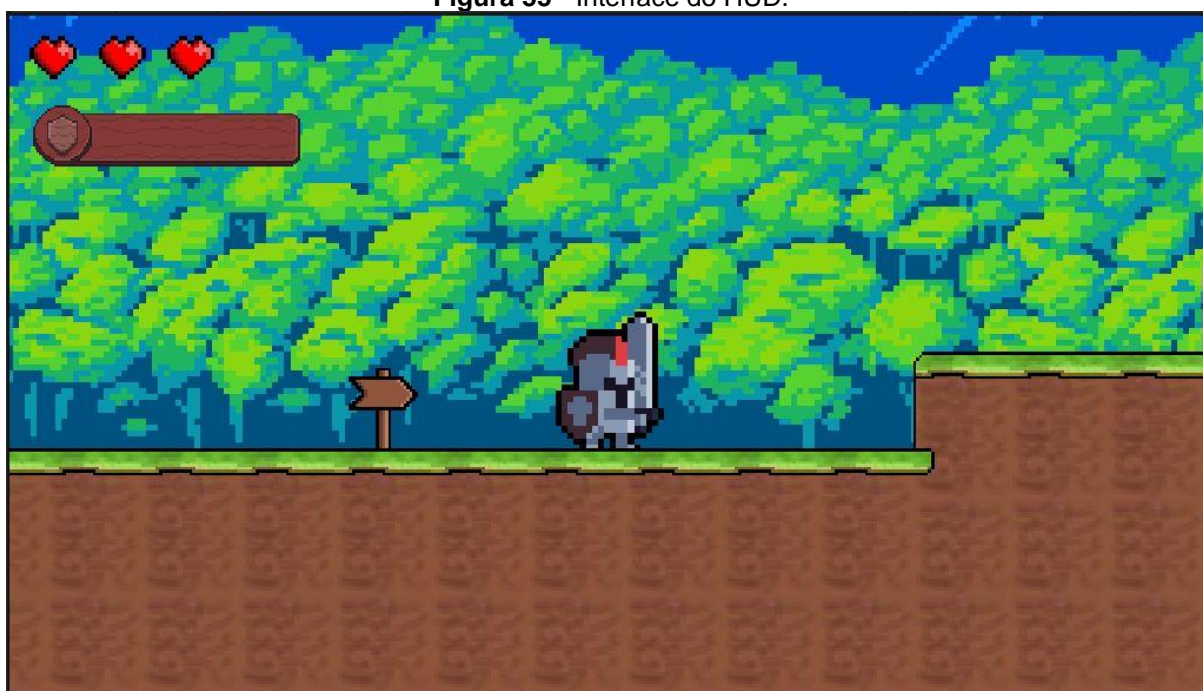
Fonte: Narik (2020), Fatec Americana (2024), adaptado pelo autor (2024).

Figura 54 - Interface do seletor de níveis.



Fonte: Narik (2020), adaptado pelo autor (2024).

Figura 55 - Interface do HUD.



Fonte: Narik (2020), adaptado pelo autor (2024).

Figura 56 - Interface do jogo pausado.



Fonte: Narik (2020), adaptado pelo autor (2024).

Figura 57 - Interface das opções de volume no jogo pausado.



Fonte: Narik (2020), adaptado pelo autor (2024).

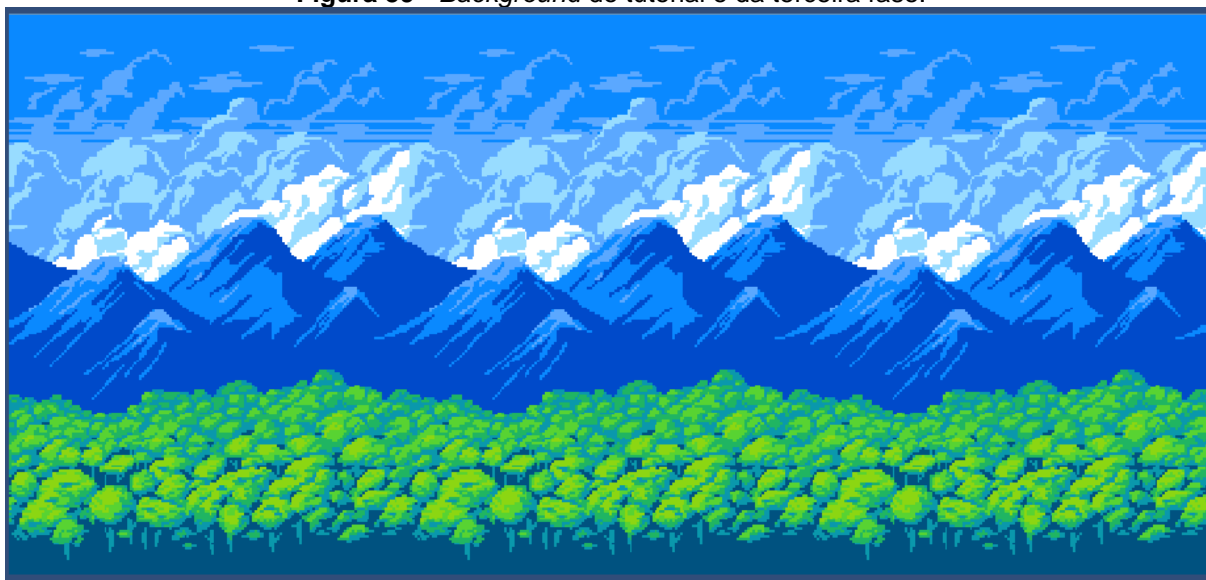
Figura 58 - Interface da tela de morte.



Fonte: Narik (2020), adaptado pelo autor (2024).

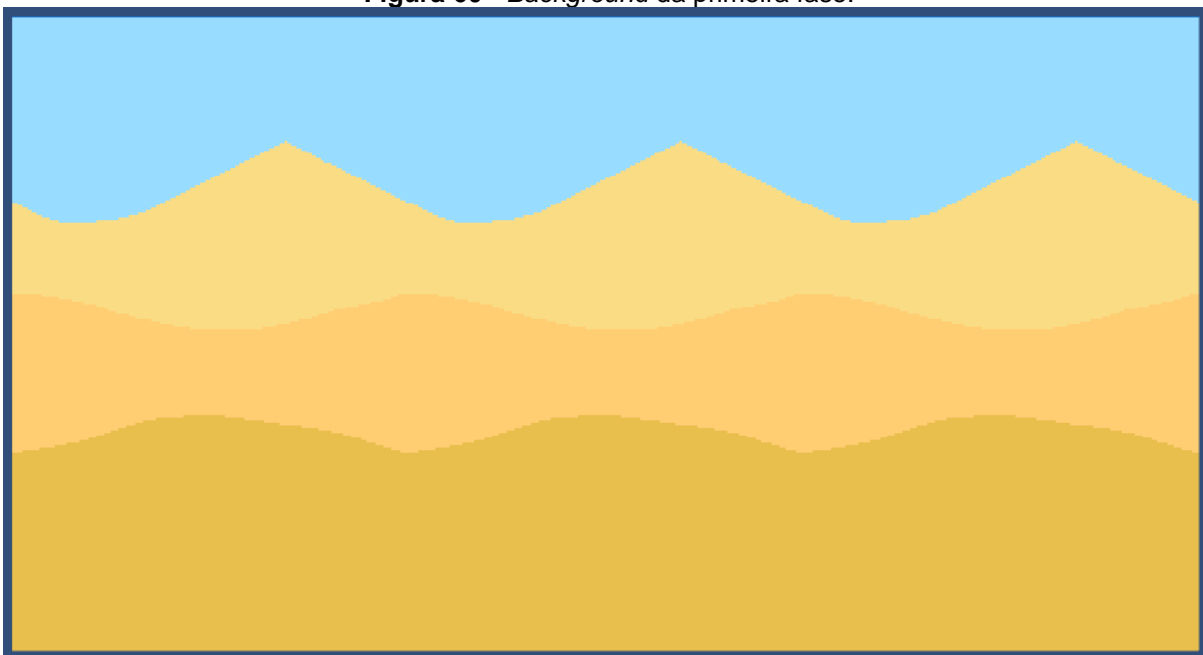
6.2.8 Backgrounds.

Figura 59 - Background do tutorial e da terceira fase.



Fonte: Ansimuz (2018).

Figura 60 - *Background* da primeira fase.



Fonte: Autoria própria (2024).

Figura 61 - *Background* da segunda fase.



Fonte: CraftPix (2019).

6.2.9 Projéteis.

Figura 62 - Projétil do inimigo planta.



Fonte: Autoria própria (2024).

Figura 63 - Projétil do inimigo tronco.



Fonte: Frog (2024).

7 DOCUMENTAÇÃO TÉCNICA

7.1 5.1 – Requisitos de sistema.

O jogo foi desenvolvido para ser otimizado e funcionar em uma ampla gama de máquinas, incluindo aquelas com *hardware* mais modesto. Abaixo estão as especificações recomendadas para rodar o jogo com uma performance aceitável e utilizando o máximo dos recursos visuais:

Sistema Operacional

Windows 10 ou superior.

Processador (CPU)

Intel Core i5 ou equivalente.

Clock de pelo menos 2,8 GHz.

Memória RAM

8 GB de RAM.

Placa Gráfica (GPU)

NVIDIA GeForce GTX 750 Ti ou equivalente, com suporte a DirectX 11.

Armazenamento

Disco rígido (HDD) ou unidade de estado sólido (SSD).

Pelo menos 2 GB de espaço livre para instalação do jogo.

Placa Mãe

Placa mãe compatível com o processador escolhido e suporte a PCIe para a placa gráfica.

Monitor

Monitor com resolução de pelo menos 1920x1080 (1080p) para exibir os gráficos do jogo de forma adequada.

Verificar a compatibilidade das entradas do monitor (HDMI, DisplayPort ou VGA) com a placa gráfica.

Periféricos

Teclado e mouse padrão para interação com os menus e interface do jogo.

Fones de ouvido ou alto-falantes para uma melhor experiência de áudio.

Requisitos Adicionais

Acesso à internet para baixar atualizações e *patches*, se necessário.

7.2 Engenharia de Software.

Figura 64 - Planejamento de tarefas do Giovanni Marchan Santos.

<input type="checkbox"/>	Tipo	# Chave	Resumo	Status	Sprint	Responsável	Data de entrega
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-5	Personagem principal	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	4 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-6	Sistema de vida	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	22 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-7	Barra de estamina do escudo	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	17 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-8	Primeiro cenário	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	6 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-9	Criar três inimigos diferentes	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	8 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-10	Detectar colisões	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	21 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-11	Animações do personagem principal	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	10 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-12	Animações dos inimigos	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	28 de out. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-17	Sonorização	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	5 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-13	Tela de menu inicial	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	30 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-14	Segundo cenário	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	4 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-15	Terceiro cenário	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	11 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-16	Tutorial do jogo	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	1 de out. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-18	Opções de controlar o volume do jogo	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	1 de nov. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-19	Tela de créditos	CONCLUÍDO	TCC - Giovanni e Felipe	Giovanni_MS "Giova...	20 de nov. de 2024

Fonte: Adaptado pelo autor (2024).

Figura 65 - Planejamento de tarefas do Felipe Gonçalves.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-1	Ideia do jogo	CONCLUÍDO	TCC - Giovanni e Felipe	felipe.goncalves34	1 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-2	Roteiro da história	CONCLUÍDO	TCC - Giovanni e Felipe	felipe.goncalves34	1 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-3	Estilo do jogo	CONCLUÍDO	TCC - Giovanni e Felipe	felipe.goncalves34	1 de set. de 2024
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SCRUM-4	Escolher as mecânicas do jogo	CONCLUÍDO	TCC - Giovanni e Felipe	felipe.goncalves34	1 de set. de 2024

Fonte: Adaptado pelo autor (2024).

7.3 Software(s) Secundário(s).

Libresprite: um programa gratuito e *open-source* desenvolvido pelo David Capello no ano de 2021, foi utilizado para criar e animar alguns sprites presentes no projeto.

7.4 Game Engine.

Uma game engine é um ambiente de desenvolvimento integrado que oferece um conjunto de ferramentas visuais e componentes de *software* reutilizáveis, facilitando a criação de jogos. Ela simplifica as tarefas complexas de desenvolvimento, proporcionando uma camada de abstração que torna muitas partes difíceis mais acessíveis, enquanto lida com as fórmulas e codificações mais pesadas. Essencialmente, uma game engine é um *framework* projetado para a construção e desenvolvimento de jogos digitais, permitindo a criação em consoles, dispositivos móveis e computadores (Tecnoblog, 2021).

Unity

A Unity proporciona um conjunto poderoso de ferramentas gráficas, físicas e de áudio, além de suporte extensivo para *scripts* e conectividade online. Sua capacidade de importar uma variedade de formatos de ativos e seu sistema robusto de física e renderização a tornam uma escolha ideal para desenvolvedores que buscam criar experiências interativas de alta qualidade. (Unity User Manual, 2022)

A Unity é uma *engine* de desenvolvimento de jogos e aplicativos interativos bastante utilizada por uma grande quantidade de desenvolvedores de jogos profissionais ou indies. Ela oferece uma plataforma unificada para criar projetos em 2D, 3D, realidade virtual (VR) e realidade aumentada (AR). Essa *engine* se destaca das demais pela facilidade de uso, suporte multiplataforma e pela grande quantidade de *assets* gratuitas feitos exclusivamente pela comunidade da Unity.

7.5 Bibliotecas.

Na programação do jogo “Equitis Adventum” foram utilizadas as bibliotecas padrões do Unity para desenvolvimento em plataforma 2D.

7.6 Programação.

Para o desenvolvimento do “Equitis Adventum”, foi utilizado a linguagem C# ou (C Sharp). Além do fato dela ser a principal linguagem de programação aceita pela Unity, essa é uma das melhores linguagens disponíveis para a criação de jogos digitais. C# possui uma perfeita integração entre os *scripts*, variáveis estáticas e muitas funções pré-prontas voltadas para a criação de experiências interativas entre o jogador e o jogo. A flexibilidade que o C# dispõe facilita a implementação de mecânicas de jogos simples e complexos, garantindo uma experiência de jogo bastante fluida.

7.7 Scripting

Utilizando o C# nos scripts do jogo obteve-se excelentes resultados para a programação da lógica de movimentação dos personagens, para a detecção de dano, na transição de tela dos próximos níveis, entre outras funções desenvolvidas. Neste tópico será mostrado apenas três exemplos de como foram utilizados alguns *scripts* para a movimentação do personagem principal, para a inteligência artificial de um dos inimigos do jogo e para a transição de telas quando o jogador passa de nível.

7.7.1 Script de movimentação do personagem principal.

O *script* PlayerBehaviour utiliza diversas variáveis para verificar o estado do movimento e atualizar as animações do personagem. Além de verificar se houve alguma entrada do jogador no teclado ou no mouse para alterar essas mesmas variáveis e verificar se o ele está tocando o chão para poder realizar alguns movimentos específicos, como pular ou defender.

Figura 66 - Script PlayerBehaviour variáveis.

```

21 references | 0 scene usages | 1 prefab usage
public class PlayerBehaviour : MonoBehaviour

[Header("Player Info")]
1 reference
[SerializeField] private float moveSpeed;
1 reference
[SerializeField] private float jumpPower;
1 reference
[SerializeField] private float footstepInterval;
1 reference
[SerializeField] private float reboundVelocity;

[Header("Layer Info")]
1 reference
[SerializeField] private LayerMask whatIsGround;
1 reference
[SerializeField] private ParticleSystem dust;

17 references
private Rigidbody2D rb2d;
8 references
private Animator anim;
3 references
private CapsuleCollider2D capsuleCollider2D;
6 references
private PlayerLife playerLife;
2 references
private PauseScript pauseGame;

5 references | 4 references
public bool isFacingRight, canMove;
5 references | 8 references | 8 references
private bool isGrounded, attacking, blocking;
8 references
private float horizontalInput;
3 references
private float footstepTimer;

1 reference
private float attackCooldown = 0.5f;
2 references
private float lastAttackTime;
1 reference
private float attackBufferTime = 0.3f;
4 references
private float attackBufferCounter;
1 reference
private float coyoteTime = 0.2f;
4 references
private float coyoteTimeCounter;
1 reference
private float jumpBufferTime = 0.2f;
4 references
private float jumpBufferCounter;

[Header("Stamina Settings")]
4 references
public Slider staminaSlider;
6 references
public float maxStamina;
1 reference
public float staminaDepletionRate;
1 reference
public float staminaRecoveryRate;
1 reference
public float recoveryDelay = 1f;

5 references
public bool isDefending = false;
4 references
public bool canDefend = true;
11 references
private float currentStamina;
3 references
private float recoveryTimer;

```

Fonte: A autoria própria (2024).

Figura 67 - Script PlayerBehaviour função de entrada do jogador.

```

1 reference
private void PlayerInput()
{
    if(!playerLife.takingDamage)
    {
        // Movimento horizontal
        horizontalInput = Input.GetAxis("Horizontal");

        // Pular
        if(IsGrounded())
        {
            coyoteTimeCounter = coyoteTime;
        }
        else
        {
            coyoteTimeCounter -= Time.deltaTime;
        }

        if(Input.GetButtonDown("Jump"))
        {
            jumpBufferCounter = jumpBufferTime;
        }
        else
        {
            jumpBufferCounter -= Time.deltaTime;
        }

        if(coyoteTimeCounter > 0f && jumpBufferCounter > 0f && !attacking && !blocking)
        {
            Jump();
            jumpBufferCounter = 0f;
        }

        if(Input.GetButtonUp("Jump") && rb2d.velocity.y > 0f)
        {
            rb2d.velocity = new Vector2(rb2d.velocity.x, rb2d.velocity.y * 0.5f);
            coyoteTimeCounter = 0f;
        }

        //Atacar
        if(Input.GetButtonDown("Fire1"))
        {
            attackBufferCounter = attackBufferTime;
        }
        else
        {
            attackBufferCounter -= Time.deltaTime;
        }

        if(attackBufferCounter > 0f && !attacking)
        {
            Attack();
            attackBufferCounter = 0;
            horizontalInput = 0;
        }

        // Bloquear
        if (Input.GetButtonDown("Fire2") && IsGrounded() && !blocking && canDefend)
        {
            Block();
            isDefending = true;
            horizontalInput = 0;
        }

        if (Input.GetButtonUp("Fire2") && blocking)
        {
            DisableBlock();
            isDefending = false;
        }

        // Pausar
        if (Input.GetButtonDown("Cancel"))
        {
            pauseGame.SetPauseMenu(!PauseScript.paused);
        }
    }
}

```

Fonte: Autoria própria (2024).

7.7.2 Script de movimentação e inteligência artificial do inimigo fantasma.

O script FollowPlayerArea utiliza diversas variáveis para verificar o estado de movimentação e atualizar as animações do inimigo. Ele também verifica se o personagem entrou na área de detecção do fantasma para trocar do estado atual para outro estado caso o jogador entre na área de alcance do fantasma para persegui-lo. Além de retorná-lo para a origem caso algumas condições da área de detecção sejam verdadeiras ou se o cavaleiro estiver morto.

Figura 68 - Script FollowPlayerArea variáveis.

```

0 references | 0 scene usages | 1 prefab usage
public class FollowPlayerArea : MonoBehaviour
{
    [Header("Enemy Status")]
    [SerializeField] private float searchRadius;
    [SerializeField] private float maxDistance;
    [SerializeField] private float movemetVelocity;
    [SerializeField] private float reboundPower;
    [SerializeField] private LayerMask whatIsPlayer;
    [SerializeField] private Transform playerTransform;
    [SerializeField] private MovimentState actualState;
    public enum MovimentState
    {
        Waiting,
        Following,
        Returning,
        TakingDamage,
    }
    private Vector3 initialPoint;
    private Rigidbody2D rb2d;
    private Animator anim;
    private EnemyLife enemyLife;
    private PlayerLife playerLifeScript;
    private PlayerBehaviour playerMovementScript;
}

```

Fonte: Autoria própria (2024).

Figura 69 - Script FollowPlayerArea função de troca de estado.

```

0 references | Unity Message
private void Start() ...

0 references | Unity Message
private void Update()
{
    switch (actualState)
    {
        case MovimentState.Waiting: ...
        case MovimentState.Following: ...
        case MovimentState.Returning: ...
        case MovimentState.TakingDamage: ...
    }

    AnimationState();
}

1 reference
private void WaitingState() ...

1 reference
private void FollowingState() ...

1 reference
private void ReturningState() ...

1 reference
private IEnumerator TakingDamage() ...

2 references
private void FlipSprite(Vector3 target) ...

0 references | Unity Message
private void OnCollisionStay2D(Collision2D collision) ...

0 references | Unity Message
private void OnTriggerEnter2D(Collider2D collision) ...

1 reference
private void AnimationState() ...

0 references | Unity Message
private void OnDrawGizmos() ...

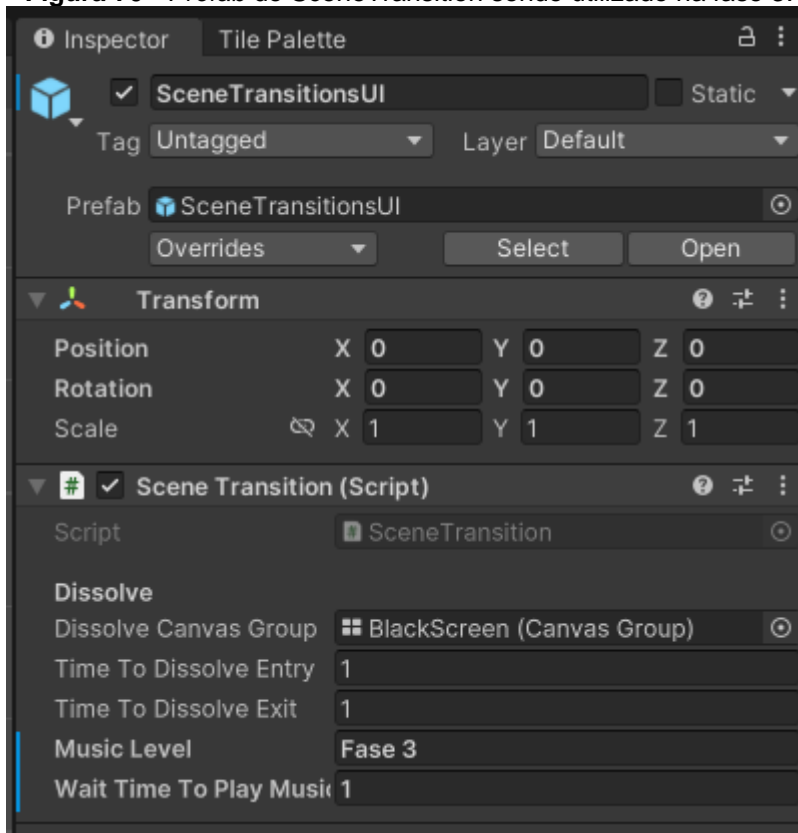
```

Fonte: Aatoria própria (2024).

7.7.3 Script de transição de tela.

O *script* SceneTransition utiliza algumas variáveis e funções para controlar a entrada e a saída de transições durante as *cut-scenes*. Ele também está encarregado de tocar ou parar a música da fase atual após um certo tempo, neste caso a música toca assim que as transições acabam.

Figura 70 - Prefab do SceneTransition sendo utilizado na fase 3.



Fonte: Aatoria própria (2024).

Figura 71 - Script SceneTransition funções de troca de cena.

```

1  using System.Collections;
2  using UnityEngine;
3  using UnityEngine.SceneManagement;
4
5  9 references | 0 scene usages | 1 prefab usage
6  public class SceneTransition : MonoBehaviour
7  {
8      10 references
9      [Header("Dissolve")]
10     12 references
11     public CanvasGroup dissolveCanvasGroup;
12     1 reference
13     public float timeToDissolveEntry;
14     3 references
15     public float timeToDissolveExit;
16     1 reference
17     public string musicLevel;
18     1 reference
19     public float waitTimeToPlayMusic;
20
21     0 references | Unity Message
22     private void Awake()
23     {
24         if(Instance == null)
25         {
26             Instance = this;
27         }
28         else
29         {
30             Destroy(gameObject);
31         }
32     }
33
34     0 references | Unity Message
35     private void Start()
36     {
37         StartCoroutine(DissolveEntry());
38     }
39
40     1 reference
41     private IEnumerator DissolveEntry() ...
42
43     6 references
44     public void DissolveExit(int indexScene) ...
45
46     1 reference
47     public void DissolveExitGame() ...
48
49     1 reference
50     public void DissolveNextLevel() ...
51
52 }

```

Fonte: Autoria própria (2024).

8 GERENCIAMENTO E ANÁLISE DO PROJETO

8.1 Organização inicial do projeto.

Gerente de projetos: Felipe Gonçalves e Giovanni Marchan Santos

Desenvolvedores: Felipe Gonçalves e Giovanni Marchan Santos

Designer: Felipe Gonçalves e Giovanni Marchan Santos

Game Designer: Felipe Gonçalves e Giovanni Marchan Santos

Respondendo ao Coordenador do TCC: José William Pinto Gomes

8.2 Escopo do projeto.

Menu (Jogar, Opções, Créditos, Sair).

- 1 História do jogo.
- 3 Fases diferentes.
- 6 tipos diferentes de inimigos.

Resumo do escopo do game para se ter uma ideia do seu tamanho (quantidade de setores, fases, ambientes, NPCs, armas e outros itens relevantes).

8.3 Recursos.

O projeto foi planejado em executado em 5 computadores diferentes que utilizavam o sistema da Unity 2022.3.36f1 (64-bit); Visual Studio 2022 (Ambiente de desenvolvimento Integrado);

8.4 Entregas.

Ao longo do desenvolvimento do projeto foram feitas versões jogáveis para que testes fossem realizados para podermos encontrar erros no projeto junto com uma coleta de dados para podermos corrigir e melhorar de acordo com o que os tentadores avaliaram, para darmos uma melhor experiencia do público.

8.5 Premissas.

O cronograma do desenvolvimento do projeto foi dividido em três meses, com metas estabelecidas para monitorar o progresso e garantir avanços consistentes. No primeiro mês, foram definidas as ideias iniciais, com o foco em uma proposta simples para o início do projeto. No segundo mês, o projeto já se encontrava em um estado jogável, ainda utilizando o título provisório “Knight’s Adventure”. No terceiro mês, o projeto foi finalizado com o título “Equitis Adventum”, e a documentação foi iniciada após sua conclusão. Como resultado de tal medida, quase todas as metas semanais foram executadas dentro do prazo estabelecido, corroborando com a produção no dado tempo.

8.6 Restrições.

Devido ao curto espaço de tempo para o desenvolvimento que foram apenas 3 meses o projeto não pode chegar ao seu estado máximo sendo apenas um simples e curto plataforma 2D.

8.7 Cronograma.

Mês 1

Estipulado: Definir as bases do novo projeto.

Alcançado: bases do projeto foram estabelecidas como Fases, inimigos, história etc.

Mês 2

Estipulado: personagem, inimigos, cenários, fases.

Alcançado: personagens, inimigos fases e cenários feitos ou pegos de terceiros.

Mês 3

Estipulado: mecânicas pedidas pelos testadores, melhor polimentos

Alcançado: mecânicas pedidas pelos jogadores, jogo polido com novos inimigos adicionados.

8.8 Estimativa de Custos.

Toda a produção do projeto foi executada utilizando em sua maioria por trabalhos com direitos autorais encontrados na internet que foram devidamente referenciados.

Além disso, os *softwares* utilizados são oriundos de/ou possuem licenças gratuitas. Logo, o único custo presente no projeto, fora o tempo depositado nele.

8.9 Limites do Projeto.

No projeto chamado “Equitis Adventum” inclui-se: uma tela de menu com 4 botões sendo elas jogar, opções, créditos, sair. No botão jogar tem acesso ao tutorial e as 3 fases do jogo. No botão opções tem acesso a mudar o volume do jogo. Nos créditos tem os créditos mostrando os desenvolvedores e o coordenador. No botão sair fecha o jogo. Atribui-se como resultado do projeto: Jogo; chamado “Equitis Adventum” os modelos 2D, programação, trilhas sonoras, efeitos visuais e elementos gráficos. São resultados do trabalho sem autoria de terceiros, que estão mencionados na tela de “créditos” do produto final.

8.10 Processos.

Definição de *Game Engine*: Devido ao cronograma compacto, o projeto percorreu em constante alternância entre prototipagem e implementação no projeto final, em que caso classificado como trabalhoso ou inviável, a concepção definida no escopo do projeto era reimaginada ou descartada visando a fluidez do desenvolvimento.

Desenvolvimento da arte e gráficos: Sendo frutos de terceiros, que dispõem seus trabalhos para livre uso na internet.

Programação e jogabilidade: foi feito primeiramente a movimentação do jogador implementando as mecânicas e controles lógicos básicos tangidos pelo *design* de jogabilidade, para então desenvolver a IA dos inimigos presentes no jogo

Desenvolvimento de áudio: Análogo ao desenvolvimento de arte e gráficos, também é fruto de terceiros.

Testes e qualidade: Processo restringido aos testes funcionais realizados pelo desenvolvedor e os testes de usabilidade da primeira exposição do projeto ao público. Testes estes, que foram fundamentais na melhoria e garantia da qualidade do produto.

8.11 Análise de Riscos.

Descrição dos Níveis de Risco do Projeto:

Possíveis Riscos do Projeto:

Leve: Subestimação do tempo de desenvolvimento: Dificuldade em estimar corretamente o tempo necessário para completar certas funcionalidades ou ajustes.

Leve: Problemas com integração de componentes: Dificuldades na integração de diferentes partes do jogo, como mecânicas de jogo, gráficos e áudio.

Leve: Atraso na produção de arte e *assets*: Dificuldades na criação de *assets* visuais e sonoros dentro do prazo estabelecido.

Leve: Doenças ou imprevistos pessoais: Problemas de saúde ou outros imprevistos pessoais que possam afetar o desenvolvimento do jogo.

Médio: Escassez de recursos de modelagem: Limitações de recursos financeiros ou técnicos para a criação de *assets* 2D.

Médio: Mudanças imprevistas nos requisitos de *gameplay*: Alterações nas especificações de jogo que possam exigir ajustes significativos no *design* e na programação.

Médio: Dificuldades técnicas com mecânicas específicas: Problemas na implementação de mecânicas de jogo mais complexas, como física avançada ou IA sofisticada.

Médio: Compromissos pessoais que afetam o desenvolvimento: Compromissos acadêmicos, profissionais ou pessoais que possam interferir no tempo dedicado ao projeto.

Médio: Problemas de comunicação com a equipe (sendo você sozinho): Dificuldades em receber *feedback* ou informações críticas de terceiros devido à falta de uma equipe estruturada.

Grave: Falhas técnicas graves durante o desenvolvimento: Problemas severos com *hardware*, *software* ou infraestrutura que impeçam o progresso no desenvolvimento do jogo.

Grave: Desempenho insatisfatório de recursos limitados: Limitações de

recursos, como poder de processamento ou memória, que afetam o desempenho geral do jogo.

8.12 Viabilidade Técnica.

Este tópico abordará a viabilidade técnica do desenvolvimento do jogo “Equitis Adventum” destacando as tecnologias necessárias, os desafios técnicos a serem enfrentados e estratégias para gerenciar o cronograma e o escopo do projeto.

Tecnologias

- **Motor de Jogo:** O jogo “Equitis Adventum” foi desenvolvido utilizando o motor de jogo Unity. A escolha do Unity permitiu criar um ambiente 2D robusto e interativo, facilitando a implementação de mecânicas de jogo complexas e a integração de recursos visuais e de áudio.
- **Linguagem de Programação:** A Unity utiliza principalmente a linguagem C# para programação, escolhida por sua compatibilidade com a *engine* e por ser amplamente suportada na indústria de jogos. Isso facilitou o desenvolvimento das mecânicas de jogo e a implementação de sistemas interativos.
- **Ferramentas de Arte:** Para a criação dos gráficos e animações 2D, foram utilizadas bibliotecas de *assets* disponíveis na Unity, como modelos de personagens, ambientes e objetos. Isso permitiu uma rápida prototipagem e desenvolvimento sem a necessidade de criar ativos do zero.
- **Bibliotecas de Áudio:** Para a trilha sonora e efeitos sonoros, foram utilizadas bibliotecas de áudio disponíveis na Unity, além de *softwares* externos que disponibilizam efeitos sonoros gratuitos e personalizados.

Desafios Técnicos

- **Gerenciamento de Recursos:** Foi necessário otimizar o jogo para garantir um desempenho estável, com atenção especial ao gerenciamento de memória e ao uso eficiente de recursos gráficos e de processamento.
- **Balanceamento de Jogo:** Ajustar a dificuldade e o ritmo do jogo para oferecer uma experiência desafiadora, mas equilibrada, foi essencial. Isso envolveu testes extensivos e ajustes nas mecânicas de combate, progressão e IA.

Cronograma e Escopo

- **Planejamento Iterativo:** O desenvolvimento do jogo foi estruturado em 3 meses, sendo diariamente atualizado no GitHub, facilitando o acesso para fazer os ajustes finais.
- **Estimativa de Tempo:** Cada aspecto do jogo foi cuidadosamente estimado em termos de tempo e esforço, garantindo que o projeto permanecesse dentro do prazo.
- **Priorização de Recursos:** Recursos essenciais foram identificados e priorizados para manter o foco nos elementos fundamentais para a jogabilidade e a experiência do jogador, mesmo com recursos limitados.

8.13 Viabilidade Econômica.

Não houve gastos.

8.14 Plano de testes e Controle de qualidade.

Durante o desenvolvimento do projeto, foram realizados testes sistemáticos para garantir a qualidade e a funcionalidade do jogo. Abaixo estão alguns dos resultados dos testes e as soluções adotadas:

Dificuldade: foi relatado muita dificuldade pela ordem das fases.

Solução: mudar as ordens das fases

Dificuldade: mecânicas que os jogadores de teste falavam que faltou no jogo.

Solução: implementamos essas mecânicas.

Teste de Sons e Músicas:

As músicas e efeitos sonoros foram bem aceitos pelos jogadores.

Teste de Colisões e Física:

os sistemas de física foram bem-feitos e receberam elogios

Teste de Jogabilidade e Controles:

Os sistemas de jogabilidade e controle estão no padrão para um jogo de plataforma, são bem-feitos e qualquer jogador consegue se adaptar.

8.15 Ferramentas e técnicas.

GitHub: utilizamos o GitHub para salvar e atualizar os arquivos na nuvem.

9 RESULTADOS

Com os primeiros protótipos do projeto, os resultados alcançados foram melhores que o esperado, tendo em sua maioria avaliações perfeitas por parte das pessoas que testaram o Equitis Adventum. Eles também trouxeram *feedbacks* positivos e sugestões de melhorias para o projeto que foram implementadas na versão beta e final do jogo.

9.1 Teste alfa realizado.

Durante o primeiro teste realizado do Equitis Adventum, os resultados obtidos foram, em sua maior parte, positivos. Isso possibilitou a continuação do projeto e providenciou melhorias técnicas que os jogadores aclamavam para facilitar e melhorar a mecânica de combate do jogo.

9.2 Teste beta realizado.

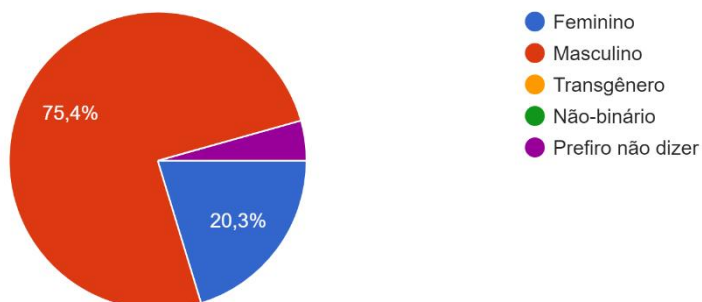
Durante o segundo teste realizado do Equitis Adventum, já com as melhorias do primeiro teste implementadas, os resultados obtidos foram excelentes. Isso possibilitou a finalização do projeto.

Em resumo, o teste beta do "Equitis Adventum" validou a direção geral do desenvolvimento do jogo, destacando a eficácia das mecânicas, a qualidade da arte e a integração do som. As sugestões dos participantes ajudaram a esclarecer o que faltava para aperfeiçoar a experiência do jogo, especialmente em termos de mecânica de combate para o personagem principal.

Figura 72 - Gráfico do gênero dos participantes que participaram dos testes.

Qual o seu Gênero?

69 respostas

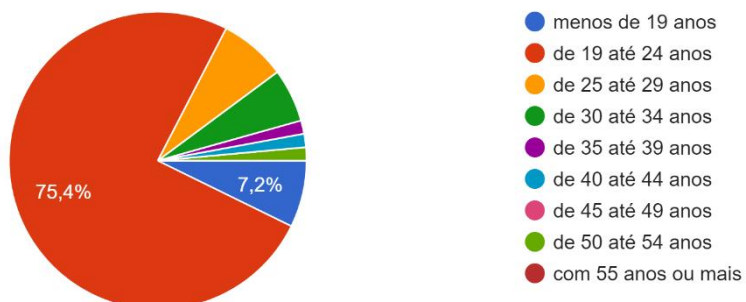


Fonte: Autoria própria (2024).

Figura 73 - Gráfico da faixa etária dos participantes dos testes.

Em qual faixa etária você está?

69 respostas

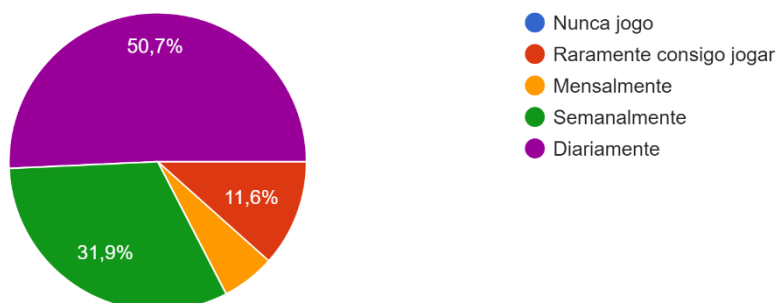


Fonte: Autoria própria (2024).

Figura 74 – Gráfico de frequência que os participantes dos testes jogam Jogos Digitais.

Com que frequência você costuma jogar Jogos Digitais?

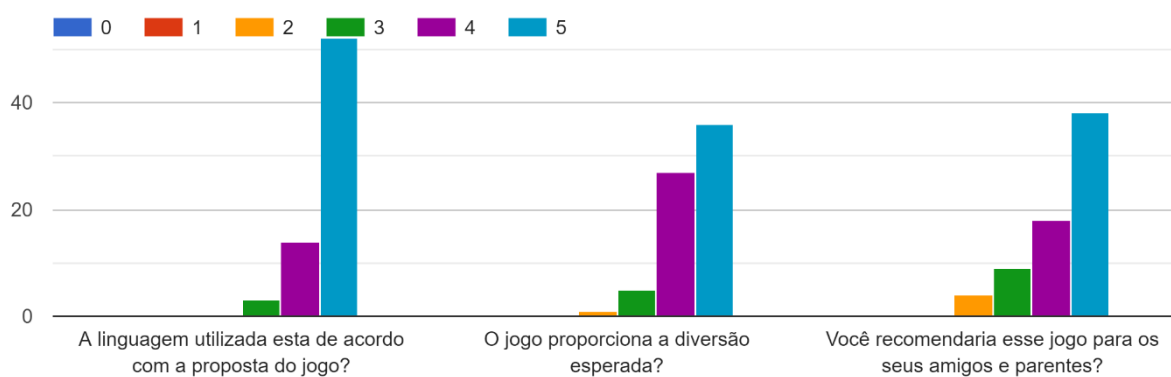
69 respostas



Fonte: Autoria própria (2024).

Figura 75 – Gráfico da opinião geral dos participantes sobre os testes.

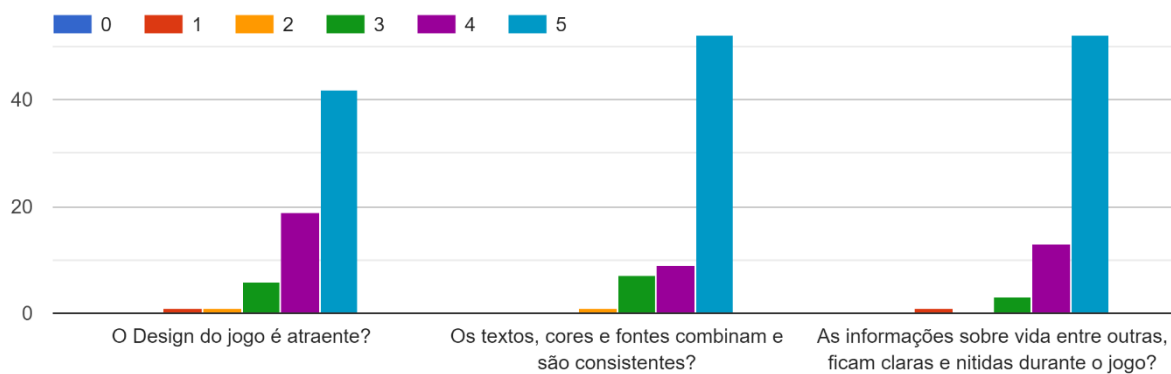
Opinião geral sobre o Jogo



Fonte: Autoria própria (2024).

Figura 76 - Gráfico da opinião dos participantes dos testes sobre o *desing* do jogo.

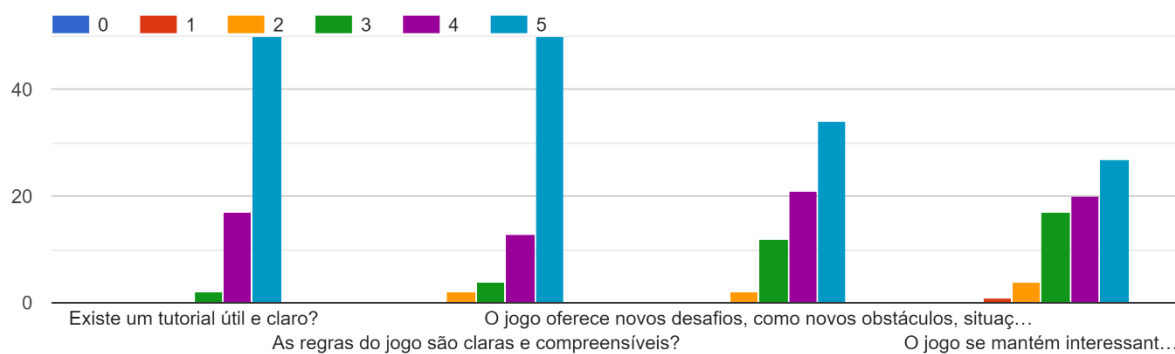
Design do Jogo



Fonte: Autoria Própria (2024).

Figura 77 – Gráfico da opinião dos participantes dos testes sobre a jogabilidade.

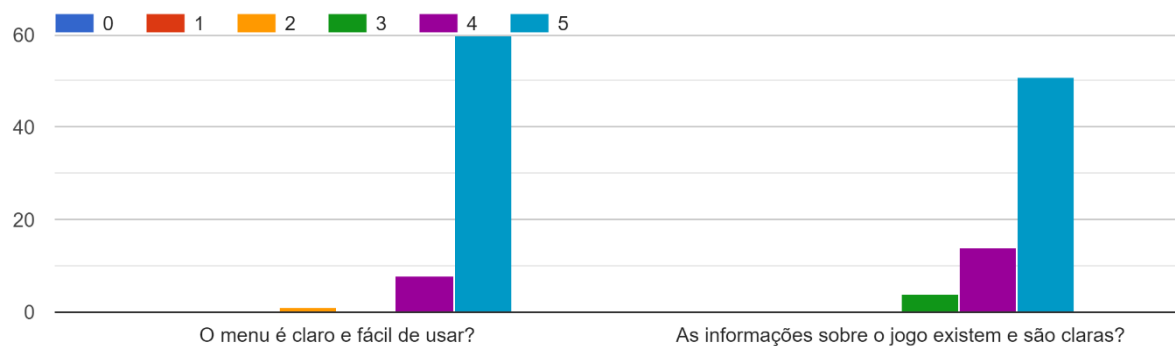
Jogabilidade



Fonte: Autoria própria (2024).

Figura 78 - Gráfico da opinião dos participantes dos testes sobre o menu e interação.

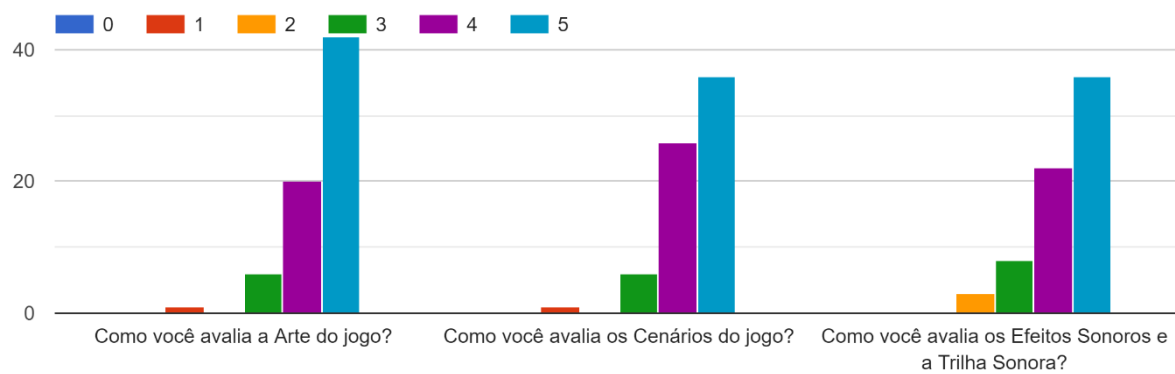
Menu e interação



Fonte: Autoria própria (2024).

Figura 79 - Gráfico da opinião dos participantes dos testes sobre a arte e o som.

Arte e Som

**Fonte:** Autoria própria (2024).

CONSIDERAÇÕES FINAIS

O que deu certo?

- a) **Implementação dos *Feedbacks*:** as mecânicas pedidas pelos jogadores nos *feedbacks* dos testes iniciais para melhorar a jogabilidade do projeto foram implementadas com sucesso.
- b) **Ferramentas escolhidas:** As escolhas das ferramentas para o desenvolvimento do jogo, como Unity para criação de jogo e Libresprite para a criação de *sprites*, foram essenciais. Elas proporcionaram os recursos necessários para implementar as ideias para o projeto de maneira eficaz.
- c) **Desenvolvimento em dupla do jogo:** A realização do projeto por dois desenvolvedores foi um dos principais acertos de "Equitis Adventum". Enquanto um dos membros desenvolvia o jogo, o outro focava totalmente na documentação, agilizando o processo da finalização tanto do projeto quanto do trabalho de conclusão de curso.
- d) **Escolha da arte do jogo:** A escolha do estilo artístico em *pixel art* 2D foi de grande ajuda para manter a beleza de um clássico jogo de plataformas, deu um estilo simples e que agradou os jogadores.
- e) **Escolha do orientador:** a escolha do orientador foi de extrema importância devido a experiência dele tanto na área como professor e orientador de diversos outros trabalhos de conclusão de curso quanto como especialista na parte de desenvolvimento de jogos.

O que deu errado?

- a) **Demora para começar o projeto:** Ter iniciado o projeto com apenas 3 meses de antecedência antes da data da entrega final, o que incomodou os integrantes da equipe devido ao pouco tempo que foi utilizado para o desenvolvimento do projeto.
- b) **Falta de comunicação da equipe:** Ocorreram poucas comunicações entre os membros do grupo e o coordenador, apenas o básico foi dito e as tarefas

que foram separadas entre os integrantes da equipe quase não foram cumpridas como era devido.

- c) **Gestão do tempo:** A demora para começar o projeto foi um grande problema sobre a divisão das tarefas com cada membro da equipe, obrigando-os a correrem contra o tempo e se apressarem para conseguir finalizar tudo o que faltava dentro desse curto prazo.
- d) **Ausência de novos desafios:** Pela demora para o início do projeto, não houve tempo para novos desafios serem implementados no jogo, podendo tornar as fases monótonas e deixando um pensamento de que faltou algo para completar.
- e) **Problemas no final do projeto:** Ao final do projeto ocorreram imprevistos como: inimigos desaparecendo durante os testes, problemas de colisão entre os objetos do jogo e mecânicas parando de funcionar como deveriam.
- f) **Teste final e *Feedback*:** Os testes finais foram feitos durante as últimas semanas antes da entrega, então os *feedbacks* recebidos foram menores do que os feitos durante o primeiro teste. Logo, utilizamos todas as análises disponíveis para implementar no projeto final.

O projeto “Equitis Adventum” atingiu todos os objetivos, conquistando os corações dos jogadores que se dedicaram para finalizar todas as fases, deram *feedbacks* positivos e foram de suma importância para dar continuidade ao projeto.

Os desafios como a gestão de tempo, a falta de comunicação da equipe e o prazo curto para a entrega do projeto evidenciaram áreas para a melhoria e forçaram a tomada de decisões para retirar mecânicas pensadas na ideia do projeto.

A continuação deste projeto e a exploração de novas técnicas podem trazer mecânicas mais complexas e novos desafios para aperfeiçoar o jogo e criar fases únicas e memoráveis.

Trabalhos Futuros.

- a) **Novas mecânicas para o personagem principal:** Desenvolver mais mecânicas para expandir as possibilidades de *level design* das fases e

aumentar as opções de movimentação do personagem, como subir escadas e agachar para passar em buracos.

- b) **Novos personagens:** Introduzir novos personagens, tanto aliados quanto inimigos, com histórias de fundo e motivações, enriquecendo a narrativa da história e trazendo mais missões para o jogo.
- c) **Implementar sistema de recuperar vida:** Desenvolver um sistema que recupera a vida do jogador após ele ter tomado algum dano.
- d) **Adicionar itens coletáveis:** Após algum inimigo ser morto pelo jogador, ele poderia soltar um item no chão que o personagem seja capaz de coletar, aumentando o leque de possibilidades podendo variar desde moedas de ouro e vida extra até itens raros e chaves para abrir portas trancadas.
- e) **Criar fases novas e dividi-las em capítulos:** Continuar a desenvolver e refinar os ambientes do jogo, trazendo biomas diversos com características únicas e mecânicas adicionais para aumentar a diversidade dos desafios.
- f) **Colocar obstáculos diferentes para as fases:** Além dos buracos sem fim e dos três tipos de inimigos, o jogo possuirá armadilhas escondidas para pegar o jogador desprevenido, portas trancadas que só se abrem caso o personagem encontre um item ou mate algum personagem e teria os clássicos espinhos afiados para aumentar um pouco a dificuldade do jogo.
- g) **Criar chefes de fase:** No final de cada fase, haverá um chefe de fase com um sistema de combate diferente, ele terá uma barra representando sua vida e o jogador só poderá passar de fase após derrotar o chefe da fase.
- h) **Adicionar um meio de contar a história do jogo:** Refinar as *cut-scenes* de modo que contassem a história desenvolvida neste documento, aumentando a motivação do jogador para continuar jogando o jogo e enriquecendo a trama principal.

Download do Jogo.

O jogo pode ser baixado de forma gratuita acessando o QR-Code abaixo ou indo no *link* do Itch.io.



<https://giovannims.itch.io/equitis-adventum/>

Disponível em: [https://giovannims.itch.io/equitis-adventum.](https://giovannims.itch.io/equitis-adventum/) Acesso em: 21 de nov de 2024.

REFERÊNCIAS

- ALVES, Maria Mikaele Rodrigues. **Sugestão 6 e 7 anos jogos eletrônicos**. 2023. Disponível em: <https://www.cerquilha.sp.gov.br/public/admin/globalarg/uploads/files/SUGESTÃO%206%20e%207%20anos%20jogos%20eletrônicos.pdf/>. Acesso em: 4 dez. 2024.
- ANSIMUZ. **Sunny land - pixel game art collection**. 2018. Disponível em: <https://ansimuz.itch.io/sunny-land-pixel-game-art/>. Acesso em: 22 nov. 2024.
- CRAFTPIX. **Free Swamp Game Tileset Pixel Art**. 2019. Disponível em: <https://craftpix.net/freebies/free-swamp-game-tileset-pixel-art/>. Acesso em: 22 nov. 2024.
- DUMATIVA. **A Lenda do Herói**. 2012. Disponível em: <https://alendadoheroi.com.br/>. Acesso em: 20 nov. 2024.
- FROG, Pixel. **Pixel adventure 1**. 2024. Disponível em: <https://pixelfrog-assets.itch.io/pixel-adventure-1/>. Acesso em: 22 nov. 2024.
- FROG, Pixel. **Pixel adventure 2**. 2024. Disponível em: <https://pixelfrog-assets.itch.io/pixel-adventure-2/>. Acesso em: 22 nov. 2024.
- GONÇALVES, R. M. **Qual foi o primeiro jogo em plataforma 2D?** 2024. Disponível em: <https://olhardigital.com.br/2024/04/01/games-e-consoles/qual-foi-o-primeiro-jogo-em-plataforma-2d/>. Acesso em: 4 dez. 2024.
- HUIZINGA, Johan. **Homo Ludens: A Study of the Play-Element in Culture**. Routledge & Kegan Paul Ltd, 1938.
- LIFTY. **Pixel Art Environment/Platformer Pack**. 2021. Disponível em: <https://assetstore.unity.com/packages/2d/pixel-art-environment-platformer-pack-195739/>. Acesso em: 22 nov. 2024.
- MORAES, Cristine do C. S. B. **Template para trabalho de conclusão de curso da Faculdade de Tecnologia de Americana**. Americana, SP: FATEC, 2013.
- NARIK. **Pixel Art Wooden GUI v1**. 2020. Disponível em: <https://soulofiran.itch.io/pixel-art-wooden-gui-v1/>. Acesso em: 22 nov. 2024.
- RGSDEV. **Animated Knights character Pack**. 2020. Disponível em: <https://rgsdev.itch.io/pixel-art-animated-knight-character-pack-rgsdev/>. Acesso em: 22 nov. 2024.
- SMITH, G.; CHA, M.; WHITEHEAD, J. **A framework for analysis of 2D platformer levels**. In: SANDBOX '08: *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*. Nova York: ACM, 2008.

TECNOBLOG. **O que é uma engine de jogos.** 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-uma-engine-de-jogos/>. Acesso em: 21 nov. 2024.

TORRES, Bolívar. **Pixel art: impulsionada pela nostalgia e pelos games, técnica de baixa resolução está em alta na cultura pop.** 2023. Disponível em: <https://oglobo.globo.com/cultura/noticia/2023/06/pixel-art-impulsionada-pela-nostalgia-e-pelos-games-tecnica-de-baixa-resolucao-esta-em-alta-na-cultura-pop.ghtml>. Acesso em: 4 dez. 2024.

UNITY Technologies. **Unity User Manual.** 2022. Disponível em: <https://docs.unity3d.com/Manual/index.html/>. Acesso em: 21 nov. 2024.

APENDICE A – Questionário para testes na versão Beta.

Questionário sobre a qualidade do Jogo Digital "Equitis Adventum"

Agradeço a sua participação.

O Game que você jogou foi desenvolvido como parte de um Trabalho de Conclusão de Curso e eu gostaria muito de saber a sua opinião.

Por favor, responda as questões apresentadas.

jwpgomes@gmail.com [Mudar de conta](#)



Não compartilhado

* Indica uma pergunta obrigatória

Qual o seu nome?

(ele será mantido em segredo, essa informação serve para controle)

Coloque apenas um nome se preferir.

Sua resposta

Qual o seu Gênero? *

- Feminino
- Masculino
- Transgênero
- Não-binário
- Prefiro não dizer

Em qual faixa etária você está? *

- menos de 19 anos
- de 19 até 24 anos
- de 25 até 29 anos
- de 30 até 34 anos
- de 35 até 39 anos
- de 40 até 44 anos
- de 45 até 49 anos
- de 50 até 54 anos
- com 55 anos ou mais

Com que frequência você costuma jogar Jogos Digitais? *

- Nunca jogo
- Raramente consigo jogar
- Mensalmente
- Semanalmente
- Diariamente

Arte e Som *

	0	1	2	3	4	5
Como você avalia a Arte do jogo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Como você avalia os Cenários do jogo?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Como você avalia os Efeitos Sonoros e a Trilha Sonora?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Deixe sua opinião e sugestões de melhorias.

Sua resposta _____