



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS CORRENTE
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

HÊNDRIL CORDEIRO RIBEIRO

RELATÓRIO TÉCNICO DO DESENVOLVIMENTO DO JOGO ELVEN ESCAPE
UTILIZANDO A FERRAMENTA GAME MAKER STUDIO 2

CORRENTE
2025

HÊNDRIL CORDEIRO RIBEIRO

RELATÓRIO TÉCNICO DE DESENVOLVIMENTO DO JOGO ELVEN ESCAPE
UTILIZANDO A FERRAMENTA GAME MAKER STUDIO 2

Trabalho de Conclusão de Curso apresentado ao
Curso de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal
de Educação, Ciência e Tecnologia do Piauí,
Campus Corrente.

Orientador: Prof. Me. Igor Bezerra Reis.

CORRENTE

2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Ribeiro, Hendril Cordeiro Ribeiro

R484r Relatório técnico do desenvolvimento do Jogo Elven Escape utilizando a
ferramenta Game Maker Studio 2 / Hendril Cordeiro Ribeiro. - 2025.
41p.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento
de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí,
Campus Corrente, 2025.

Orientador : Prof Me. Igor Bezerra Reis

1. Análise e Desenvolvimento de Sistemas - estudo e ensino. 2. Games. 3.
Protótipo. I. Título.

CDD - 004

Elaborado por Tefischer Huanderson Soares e Sousa CRB 3/1251

RESUMO

Este documento possui o intuito de documentar o processo de desenvolvimento de um protótipo de jogo onde a ferramenta a ser utilizada é o Game Maker Studio 2 (GMS2). O jogo possui o gênero de Role Playing Game (RPG) com sistema de batalha em turnos. O relatório abordará as diferentes etapas do projeto incluindo a definição de objetos, sistemas de controle e movimento, inventário, status do personagem, mecânicas de interação com o ambiente, o sistema de batalha em turnos e programas de testagem de classes. Objetos com funções mais complexas dentro do projeto serão abordados em seções próprias para que uma melhor explicação do código seja fornecida.

Palavras-chaves: jogos; rpg; protótipo; batalha em turnos; motor de jogos; desenvolvimento.

ABSTRACT

This document aims to detail the development process of a game prototype using the Game Maker Studio 2 (GMS2) engine. The game belongs to the Role-Playing Game (RPG) genre and features a turn-based battle system. The report will cover the various stages of the project, including the definition of objects, control and movement systems, inventory, character status, environmental interaction mechanics, the turn-based battle system, and class testing programs. Objects with more complex functions within the project will be addressed in dedicated sections to provide a clearer explanation of the code but also the players themselves, making the user experience more immersive.

Keywords: games; RPG; prototype; turn-based combat; game engine; development.

LISTA DE ILUSTRAÇÕES

Figura 1	Imagem da capa do jogo Chrono Trigger	17
Figura 2	Imagem dentro do jogo de Chrono Trigger	18
Figura 3	Imagem da capa do jogo Final Fantasy	19
Figura 4	Imagem da tela de batalha do jogo Final Fantasy	19
Figura 5	Arte promocional do jogo Albion Online	20
Figura 6	Interação entre jogadores no Albion Online	21
Figura 7	Cena de batalha do jogo Baldur's Gate 3	22
Figura 8	Principais etapas do projeto	23
Figura 9	Tela de desenvolvimento em um projeto no GameMaker Studio 2	25
Figura 10	Tela de batalha do jogo Earthbound	27
Figura 11	Sprite feito no programa Libresprite	28
Figura 12	Tela inicial do programa LMMS	29
Figura 13	Batalha em turnos no jogo Golden Sun	30
Figura 14	Tela de <i>debug</i> do GMS2	32
Figura 15	Gráfico de performance do jogo em modo debug	32
Figura 16	Atributos de objetos mostrados em tempo real	33
Figura 17	Tela do console de desenvolvimento	34
Figura 18	Estrutura visual do personagem principal	36
Figura 19	Código do status do personagem principal	38
Figura 20	Exemplo de habilidade dentro da batalha	39
Figura 21	Sprites do personagem	40
Figura 22	Estrutura básica dos inimigos	41
Figura 23	Funções do inimigo	41
Figura 24	Sprites dos inimigos básicos	42
Figura 25	Sprites do chefe final	43
Figura 26	Interface do inventário	44
Figura 27	Estrutura do inventário	44

Figura 28	Estrutura dos itens	45
Figura 29	Imagem do mapa completo	46
Figura 30	Tela final de pontuação	47
Figura 31	Cena de batalha com inimigos	47
Figura 32	Objetos da sala de batalha	48
Figura 33	Imagem do menu inicial finalizado	49

LISTA DE TABELAS

Tabela 1	– Resultados dos testes de funções	35
----------	--	----

LISTA DE ABREVIATURAS E SIGLAS

GML	<i>Game Maker Language</i>
GMS2	<i>Game Maker Studio 2</i>
RPG	<i>Role Playing Game</i>
GBA	<i>Game Boy Advance</i>
NES	<i>Nintendo Entertainment System</i>
LMMS	<i>Linux Multi-Media Studio</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	JUSTIFICATIVA	14
1.2	OBJETIVO GERAL.....	14
1.3	OBJETIVOS ESPECÍFICOS	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	O QUE É UM JOGO?.....	15
2.2	O QUE SÃO JOGOS ELETRÔNICOS?.....	15
2.3	O QUE SÃO JOGOS RPG?.....	16
2.4	EXEMPLOS DE JOGOS RPG	17
2.4.1	<i>Chrono Trigger</i>	17
2.4.2	<i>Final Fantasy</i>	18
2.4.3	<i>Albion Online</i>	20
2.4.4	<i>Baldur's Gate</i>	21
3	METODOLOGIA	22
3.1	IDEALIZAÇÃO DO JOGO	23
3.2	MOTORES DE JOGOS	24
3.2.1	<i>GameMaker Studio 2</i>	24
3.3	PLATAFORMA ESCOLHIDA	25
3.4	ARTE VISUAL DO JOGO	26
3.4.1	<i>O que é pixel art?</i>	26
3.5	MÚSICA E EFEITOS SONOROS	28
4	MECÂNICAS DO JOGO	29
4.1	SISTEMA DE TURNOS	29
4.2	INVENTÁRIO E EQUIPAMENTO	30
4.3	PERSONAGENS NÃO-JOGÁVEIS	31
5	TESTES	31
5.1	DEBUG	32

5.2	VISUALIZAÇÃO DENTRO DO JOGO	33
5.3	CONSOLE DE DESENVOLVIMENTO	34
6	RESULTADOS	35
6.1	PERSONAGEM PRINCIPAL	36
6.1.1	<i>Controle</i>	37
6.1.2	<i>Movimentação</i>	37
6.1.3	<i>Status</i>	38
6.1.4	<i>Habilidades</i>	39
6.1.5	<i>Visual</i>	39
6.2	INIMIGOS	40
6.2.1	<i>Estrutura</i>	41
6.2.2	<i>Sprites</i>	42
6.3	INVENTÁRIO E EQUIPAMENTOS	43
6.3.1	<i>Inventário</i>	44
6.3.2	<i>Estrutura dos itens</i>	45
6.4	CENÁRIO	46
6.5	OBJETIVO	47
6.6	SISTEMA DE COMBATE	47
7	CONCLUSÕES E TRABALHOS FUTUROS	49
	REFERÊNCIAS	53

1 INTRODUÇÃO

Os jogos eletrônicos se tornaram ferramentas de grande importância para a população em um contexto global, adquirindo o segundo lugar entre as formas de entretenimento mais consumidas no mundo inteiro ficando atrás apenas dos serviços de televisão. Mesmo sendo considerados apenas como formas de diversão por uma parte de seus jogadores, os jogos eletrônicos foram responsáveis por um lucro total de aproximadamente US\$193,7 bilhões em 2022 e possuindo previsões para 2025 que flutuam na margem dos US\$206,4 bilhões (CARDOSO; GUSMÃO; HARRIS, 2023).

É necessário entender que os jogos não fazem parte só do dia a dia da população internacional, mas também dos próprios brasileiros. Segundo pesquisas feitas em território brasileiro, notou-se que 79,5% da população considera os jogos eletrônicos como sua principal forma de entretenimento. Além disso, foi possível observar que 52,4% dos brasileiros consideram completamente ou parcialmente o setor de jogos brasileiro como uma forma viável de oferecer boas oportunidades para trabalhar ou seguir carreira (GO GAMERS, 2024).

Os jogos que compõem o mercado atual podem ser separados em gêneros, dentre estes, os jogos *Role Playing Game* (RPG) geraram ao todo US\$46.78 bilhões só no ano de 2022 com projeções apontando para um aumento até o ano de 2029 com valores aproximados de US\$73.46 bilhões (STATISTA, 2024). É possível perceber que os jogos RPG possuem um papel fundamental para a composição do mercado de jogos eletrônicos no mundo inteiro.

O objetivo deste trabalho é documentar o desenvolvimento de um jogo eletrônico utilizando o motor de jogos Game Maker Studio 2, além de apresentar outras ferramentas que foram usadas para criação de recursos visuais e sonoros no jogo, sendo o foco deste o gênero RPG, onde um personagem dentro do jogo é controlado por um jogador, tendo o objetivo final do jogo de escapar de uma prisão subterrânea enquanto enfrenta diversos inimigos e entende um pouco da história daquele lugar. O jogo possui um foco maior em estratégias devido a sua natureza de combate por turnos, onde cada personagem tem sua vez de agir, seja efetuando ataques ou usando outras ações especiais.

1.1 JUSTIFICATIVA

É necessário entender que os jogos não possuem apenas objetivo de entreter aqueles que os experienciam, mas também é utilizada para movimentar a economia local tendo a função de produto ou serviço. É possível notar uma demanda clara por jogos pela população brasileira devido a um aumento de 3% na busca de jogos como forma de entretenimento no Brasil. Este fato revela não só a busca por mais experiências interativas pelos brasileiros, mas também mostra que os jogos têm se tornado formas mais e mais utilizadas para o divertimento e lazer (CARDOSO; GUSMÃO; HARRIS, 2023).

Este relatório além de documentar como foi o processo de criação do jogo em si, também serve como uma indicação de ferramentas, técnicas e mecânicas que poderão ser usadas em outros projetos, explorando ferramentas que possam não ser tão conhecidas mas que fornecem o necessário para indivíduos que busquem o desenvolvimento de jogos eletrônicos mas que precisam considerar fatores como custo, dificuldade de aprender como programas funcionam ou até mesmo a capacidade que esses programas têm de desenvolver um projeto do início até o fim sem que o usuário sinta a falta de alguma função específica, muitas vezes permitindo que o usuário desenvolva suas próprias funções.

1.2 OBJETIVO GERAL

Desenvolver um jogo RPG que possua um sistema de batalhas em turno, utilizando como motor de jogos o programa Game Maker Studio 2 (GMS2).

1.3 OBJETIVOS ESPECÍFICOS

- Criar objetos que carregam os códigos necessários de mecânicas como movimento, colisão e interação com o ambiente.
 - Implementar os gráficos que serão utilizados pelos objetos, estes serão os sprites.
 - Compor efeitos sonoros e a trilha sonora que serão usados para fomentar a experiência do jogador.
- Desenvolver os diferentes ambientes que o jogador vai poder explorar, estas são chamadas de salas pelo Game Maker Studio 2..
- Criar equipamentos, itens consumíveis, inimigos e outros componentes básicos de um RPG.
- Executar e documentar os testes realizados no projeto para adquirir uma compreensão de problemas encontrados e solucionados.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção possui o intuito de fornecer o conhecimento necessário para compreensão dos conceitos básicos da criação de jogos, do gênero RPG e de alguns exemplos famosos de jogos do mesmo gênero.

2.1 O QUE É UM JOGO?

Os jogos são experiências lúdicas que resultam em um aperfeiçoamento físico e ou mental de um indivíduo ou grupo, podendo ter diversos objetivos como o entretenimento, aprendizado, desafiar capacidades e superar limites, entre outros. Exemplos da prática dos jogos podem ser observados em diferentes épocas e sociedades como por exemplo na Roma antiga onde jogos voltados para gladiadores se enfrentavam em coliseus eram praticados frequentemente, mesmo que por meio da violência direta. Tais jogos resultaram na fomentação na economia local e no entretenimento do povo. (REINOSO; RIOS; TEIXEIRA, 2018).

Os jogos podem ser entendidos como um conjunto de interações realizadas pelo usuário que o permitam ter um senso de controle em um mundo artificial (REINOSO et al., 2012).

2.2 O QUE SÃO JOGOS ELETRÔNICOS?

Uma variação dos jogos tradicionais são os “jogos eletrônicos” ou “jogos digitais” que são uma vertente que utiliza de computadores para controlar e aplicar as regras de como o mundo do jogo funciona e seus objetivos (SCHUYTEMA, 2007).

2.3 O QUE SÃO JOGOS RPG?

Jogos RPG ou jogos do gênero “*Role Playing Game*” são uma categoria de jogos eletrônicos. O termo significa “jogo de interpretação de papéis” e possui o conceito principal de colocar o jogador em um mundo artificial onde o mesmo possui controle de algum personagem em específico. O personagem controlado pelo jogador é capaz de tomar ações que moldam não só a si mesmo, mas também os personagens a sua volta, os ambientes em que o personagem se encontra e até mesmo modificar a história onde o mesmo se passa.

Não existe uma data precisa de quando o primeiro jogo RPG surgiu por ser um conceito ambíguo, no entanto o primeiro jogo do gênero a tomar popularidade é o “*Dungeons and Dragons*”. O jogo utiliza materiais como papel e caneta e não era digital e foi criado no ano de 1974 pelos estudantes de história Gary Gygax e Dave Arneso. Este jogo foi responsável pela popularização de muitas mecânicas como o uso de magias, equipamentos, armas, e outros.

O jogo necessitava de um jogador para assumir o papel de “mestre”, que seria responsável por ditar como o mundo funcionava, como personagens não-controlados pelos jogadores agiriam, criar diferentes perigos e desafios para os jogadores e outras funções as quais os outros jogadores não poderiam ter acesso (GOLDENBOY, 2022).

Em jogos eletrônicos, diferentes dos RPGs clássicos, o papel do mestre é substituído por um computador, permitindo que o jogador possa experienciar diferentes cenários sem necessitar de outras pessoas para que isso seja possível. Esses jogos permitem que o jogador controle seu personagem em diferentes épocas e cenários fazendo com que o jogador possa sentir emoções e ter experiências de forma mais segura do que se o mesmo as vivesse em um contexto da realidade fora do jogo.

2.4 EXEMPLOS DE JOGOS RPG DIGITAIS

Essa seção possui o intuito de apresentar exemplos de alguns jogos digitais populares que se encaixam no gênero RPG e que servem de casos que moldaram a indústria dos jogos até os dias de hoje.

2.4.1 Chrono Trigger

Chrono Trigger é um RPG eletrônico desenvolvido pela Square Co, lançado no ano de 1995 para o console *Super Nintendo*.



Figura 1 - Imagem da capa do jogo *Chrono Trigger*. Fonte: Toriyama, Akira (1995) Disponível em: https://en.wikipedia.org/wiki/Chrono_Trigger

O jogo possui diversas mecânicas que o tornam um modelo de RPG para a época como diversos ambientes exploráveis, sistema de grupos onde o jogador poderia controlar mais de um personagem contanto que estes fizessem parte do grupo principal de personagens jogáveis, inimigos diversos, itens coletáveis e equipáveis, entre outros.



Figura 2 - Imagem dentro do jogo de *Chrono Trigger*. Fonte: Eric Arraché (2018) Disponível em: <https://criticalhits.com.br/games/chrono-trigger-steam-novos-menus-batalha/>

Chrono Trigger se trata da história de Crono, um jovem que decide participar de um festival junto de seus amigos mas que após Marle, sua amiga, se voluntariar para testar uma das invenções e se perder entre dimensões, acaba indo atrás da mesma e se encontra quatro séculos no passado (SQUARE ENIX).

2.4.2 *Final Fantasy*

Final Fantasy é um RPG eletrônico desenvolvido pela Square Co, lançado no ano de 1987 para o console *Nintendo Entertainment System (NES)*. O jogo vendeu aproximadamente 1,3 milhões de cópias em todo mundo, gerando um lucro bruto na época de \$71.232.200,00 (VGSALES, 2024).

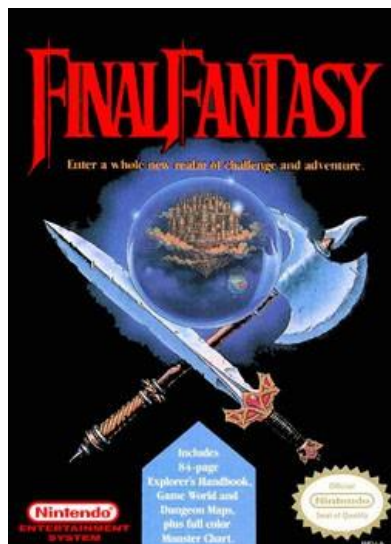


Figura 3 - Imagem da capa do jogo *Final Fantasy*. Fonte: Square Enix (1987) Disponível em: [https://en.wikipedia.org/wiki/Final_Fantasy_\(video_game\)](https://en.wikipedia.org/wiki/Final_Fantasy_(video_game))

O jogo possui diferentes telas que compõem as mecânicas base de exploração, lutas e interações com personagens não-jogáveis, como os diferentes cenários de cidades, lojas, masmorras, etc., telas de luta onde os personagens jogáveis e oponentes aparecem, telas gráficas de interface de usuário, entre outros.



Figura 4 - Imagem da tela de batalha do jogo *Final Fantasy*. Fonte: Square Enix (1987) Disponível em: [https://en.wikipedia.org/wiki/Final_Fantasy_\(video_game\)](https://en.wikipedia.org/wiki/Final_Fantasy_(video_game))

O jogo usava um sistema de grupo onde o jogador possuía controle sobre vários personagens, os mesmos tinham suas próprias características como pontos de vida, magias, equipamento, entre outros atributos que podem sofrer alterações de acordo com a evolução do personagem. Com encontros aleatórios de criaturas que, se derrotadas em batalha, poderiam fornecer pontos de experiência, que eram responsáveis por aumentar o nível do personagem, com níveis mais altos o personagem adquire novas habilidades e se tornava mais forte.

2.4.3 Albion Online

Albion Online é um jogo RPG criado pela desenvolvedora *Sandbox Interactive* e que foi lançado em janeiro de 2021 para as plataformas de computadores mais comuns, Windows, MacOS e Linux e para smartphones.



Figura 5 – Arte promocional do jogo *Albion Online*. Fonte: Steam (2025) Disponível em: https://store.steampowered.com/app/761890/Albion_Online/

Neste jogo diversos jogadores participam juntos utilizando a Internet, fazendo missões juntos em busca de recompensas, derrotando inimigos, indo em busca de equipamento para melhorar o personagem, entre outros. O jogo faz o uso de guildas, que são como associações onde jogadores podem entrar e cooperar para atingir diversos objetivos.



Figura 6 – Interação entre jogadores no *Albion Online*. Fonte: Steam (2025) Disponível em: https://store.steampowered.com/app/761890/Albion_Online/

Os gráficos do jogo são baseados em modelos 3D ao contrário dos jogos anteriores por ser mais atual, isso permite mostrar novas perspectivas do ambiente em volta do jogador por meio da câmera utilizada no jogo em si.

2.4.4 *Baldur's Gate 3*

Baldur's Gate 3 é um jogo eletrônico RPG criado pela desenvolvedora *Larian Studios* que utiliza *Dungeons & Dragons* como principal fonte para seus principais sistemas de batalha, inventário, criação de personagens e outros.



Figura 7 – Cena de batalha do jogo *Baldur's Gate 3*. Fonte: Steam (2025) Disponível em: https://store.steampowered.com/app/1086940/Baldurs_Gate_3/

O jogo possui gráficos em três dimensões, suporte para diversos personagens no mesmo grupo, incremento de níveis de acordo com a experiência ganha pelos personagens e muitas outras mecânicas utilizadas no sistema *Dungeons & Dragons*. *Baldur's Gate 3* não só é exemplo de um jogo RPG com mecânica de turnos para cada personagem, mas também um jogo muito popular no mundo todo, vendendo por volta de quinze milhões de cópias e rendendo a desenvolvedora do jogo €28 milhões apenas no ano de 2023 (YIN-POOLE, 2023).

3 METODOLOGIA

Essa seção é responsável por apresentar as ferramentas e recursos que serão utilizados para o desenvolvimento do projeto em si, além de documentar mecânicas mais específicas que farão parte do jogo. Na Figura 8 é possível ver os principais componentes que deverão ser desenvolvidos para garantir o funcionamento do jogo em si.

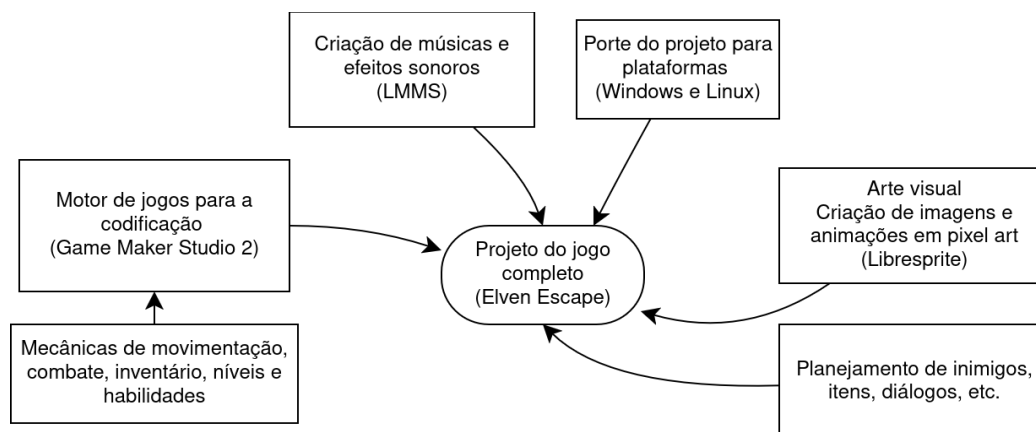


Figura 8 – Principais etapas do projeto. Fonte: Imagem do autor

3.1 IDEALIZAÇÃO DO JOGO

A ideia principal consiste no personagem principal acordar no meio de uma cela subterrânea sem memórias, o local está preenchido por inimigos e o objetivo final é escapar. O nome final deste projeto foi escolhido como “Elven Escape” que pode ser traduzido como “fuga élfica”, fazendo referência ao fato do personagem principal ser um elfo.

Para ajudar durante a aventura do jogador, algumas mecânicas foram criadas para explorar a sua evolução com o passar das lutas:

- **Armas:** Adagas, espadas, alabardas e outras armas possuem um valor de dano normal e mágico, mínimo e máximo que são somadas com a força do personagem. A diferença entre dano normal e dano mágico é que o dano mágico ignora a defesa do alvo, ou seja, é dano direto, não redutível. Cada arma mágica tem um elemento em específico, mas essa mudança é apenas cosmética.
- **Armaduras:** Leve e média, reduzem o dano sofrido de acordo com um valor aleatório numa escala própria de cada armadura que usa o valor mínimo e máximo de defesa.
- **Poção de cura normal e menor,** recuperam pontos de vida para o personagem, as poções normais curam mais porém apenas as poções menores podem ser encontradas ao derrotar inimigos.
- **Poção de energia,** recupera pontos de energia que podem ser usados em diversas habilidades no meio da luta.
- **Mochilas** que aumentam a capacidade de carga do personagem.
- **Chaves** que podem abrir as portas espalhadas pela caverna, permitem que o jogador explore novas áreas e possa terminar o jogo. Não podem ser largadas pois são cruciais para a finalização do jogo.
- **Tomo de fogo,** responsável por garantir uma habilidade de conjuração ao personagem, mas que só pode ser encontrada ao derrotar um inimigo “mago”, tem chances baixas de ser encontrada.
- **Habilidades** que podem ser usadas em batalha como “Bloquear”, que reduz uma parte do dano sofrido, “Foco de Batalha” que aumenta o dano, “Tratar Feridas” que recuperam pontos de vida, entre outros

3.2 MOTORES DE JOGOS

Os motores de jogos podem ser definidos como um conjunto de diferentes funções e ferramentas que permitem o desenvolvimento e a simulação de um jogo, tais funções podem incluir sistemas de física dentro do jogo, gráficos, sons, entre outros. Por meio dos motores de jogos é possível desenvolver o código que será utilizado pelas diferentes entidades que fazem parte do jogo em si. O uso de motores de jogos permite que os desenvolvedores possam se concentrar na criação do jogo sem necessitar de outras ferramentas na maioria das vezes (LEWIS; JACOBSON, 2002; GREGORY, 2018).

O mercado atual de desenvolvimento de jogos possui diversos motores de jogos, alguns exemplos são: *GameMaker Studio 2*, *Unity*, *Scratch* e *Unreal Engine*. O motor de jogos escolhido para o desenvolvimento desse projeto é o *GameMaker Studio 2*. Além de ser um programa que permite o desenvolvimento de jogos em computadores reunindo diversas ferramentas, este pode ser adquirido e utilizado de graça no caso atual, onde o intuito é a criação de um protótipo funcional, sem finalidades comerciais.

3.2.1 *GameMaker Studio 2*

Esse motor de jogos é um programa criado pela empresa YoYo Games que utiliza dois métodos principais de criação de jogos: “pegar e arrastar” e *Game Maker Language* (GML). O primeiro método tem foco visual e consiste em arrastar e conectar blocos com funções específicas dentro do ambiente de desenvolvimento.

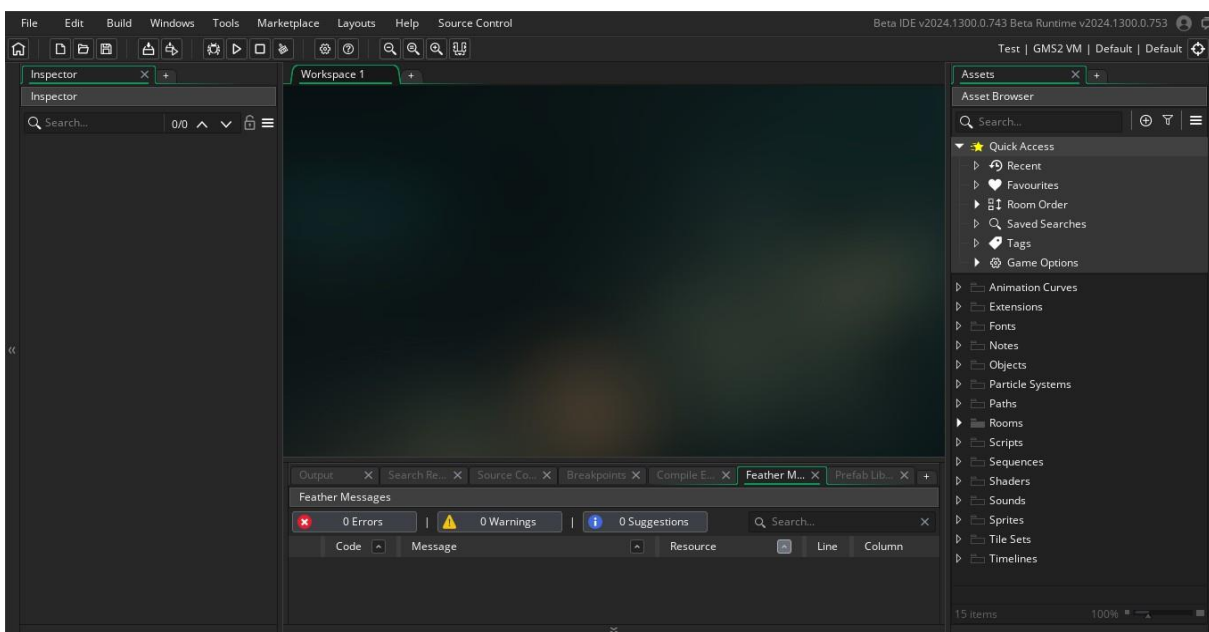


Figura 9 – Tela de desenvolvimento em um projeto no *GameMaker Studio 2*. Fonte: Imagem do autor

O segundo método utiliza uma forma mais tradicional de programação por meio de uma linguagem própria chamada GML onde é possível programar diretamente como cada objeto dentro do jogo vai agir ou quais características estes vão possuir. Utilizando esse método é possível criar e controlar personagens, inimigos, cenários, entre outros elementos que irão compor o protótipo final.

3.3 PLATAFORMA ESCOLHIDA

Existem diferentes plataformas para que um jogo possa ser desenvolvido, desde consoles, computadores e até mesmo celulares. No entanto, pesquisas apontaram uma estimativa de 1,8 bilhões de usuários utilizavam computadores para executar algum tipo de jogo eletrônico no ano de 2024 no mundo inteiro (STATISTA, 2022).

Além disso, é possível justificar a escolha da plataforma analisando o próprio mercado brasileiro, onde uma pesquisa revelou que existiam no ano de 2023 cerca de 215 milhões de computadores em todo o território brasileiro (FGV, 2023). Analisando como o mercado atual de jogos possui uma grande parcela de jogadores que utilizam computadores para jogar, a plataforma de foco para o desenvolvimento deste protótipo foi a dos computadores.

3.4 ARTE VISUAL DO JOGO

Para que os objetos do jogo possam ter uma ilusão de movimento, é necessário utilizar da animação por meio de *sprites*, que são um conjunto de imagens que ao serem trocadas rapidamente fornecem a impressão de movimento contínuo (GOMES, 2014). Com isso, é possível usar diferentes imagens do mesmo personagem para causar a impressão de que o mesmo está correndo, pulando, atacando, etc.

Pesquisas foram realizadas para encontrar quais estilos de arte poderiam ser utilizados no desenvolvimento do jogo. O estilo visual escolhido foi o de *pixel art*, utilizado por jogos da era do *Nintendo Entertainment System* (NES) e do *Game Boy Advance* (GBA).

3.4.1 Pixel Art

Os monitores usados por computadores, telefones e outros dispositivos são constituídos por *pixels*, que são como pequenos pontos que possuem alguma cor específica. Estes podem ser usados em conjunto para formar imagens maiores e mais detalhadas.

O estilo de arte *pixel art* atual é uma remanescente da baixa capacidade dos computadores antigos de mostrar imagens na tela, onde imagens com muitos *pixels* necessitavam de um grande poder de processamento. *Pixel art* é o nome dado ao estilo de imagens onde cada pixel visível na tela é colocado intencionalmente (SILBER, 2015).

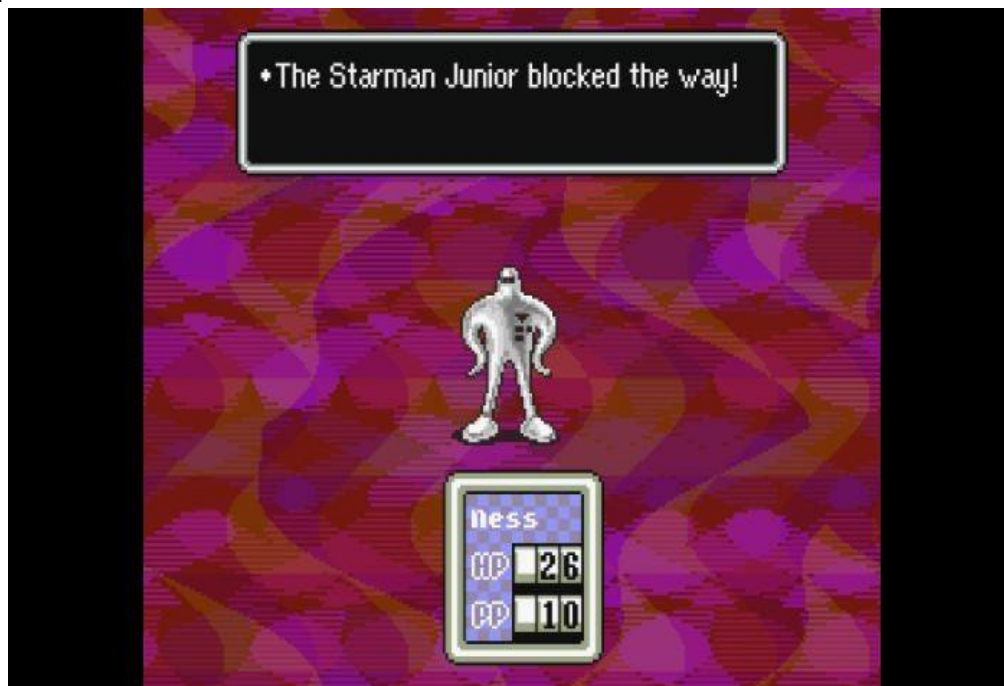


Figura 10 – Tela de batalha do jogo *Earthbound*. Fonte: Thonky (2022). Disponível em: <https://www.thonky.com/earthbound-walkthrough/starman-junior>

Na Figura 10 é possível observar o estilo de *pixel art* aplicado no jogo *Earthbound* onde um dos inimigos é representado no centro da tela. No lado inferior um dos elementos da interface de usuário é utilizado para representar os pontos de vida e pontos de poder do personagem controlado pelo jogador.

Com o aumento de *pixels* na tela, imagens mais detalhadas podem ser representadas, no entanto um custo computacional mais elevado será criado dependendo do tamanho da imagem a ser mostrada, or esse motivo computadores e consoles mais antigos possuem limitações visuais características. Essas limitações são mais raras hoje em dia, porém ainda são utilizadas como um estilo artístico.

Existem diversos programas que podem ser utilizados para criar imagens no estilo *pixel art*, no entanto o programa escolhido para esse projeto foi o Libresprite. Além de ser um programa disponível gratuitamente para o uso, o mesmo é capaz de criar animações, pode ser usado para criar cenários, permite salvar as imagens em alta qualidade, além de outros benefícios.

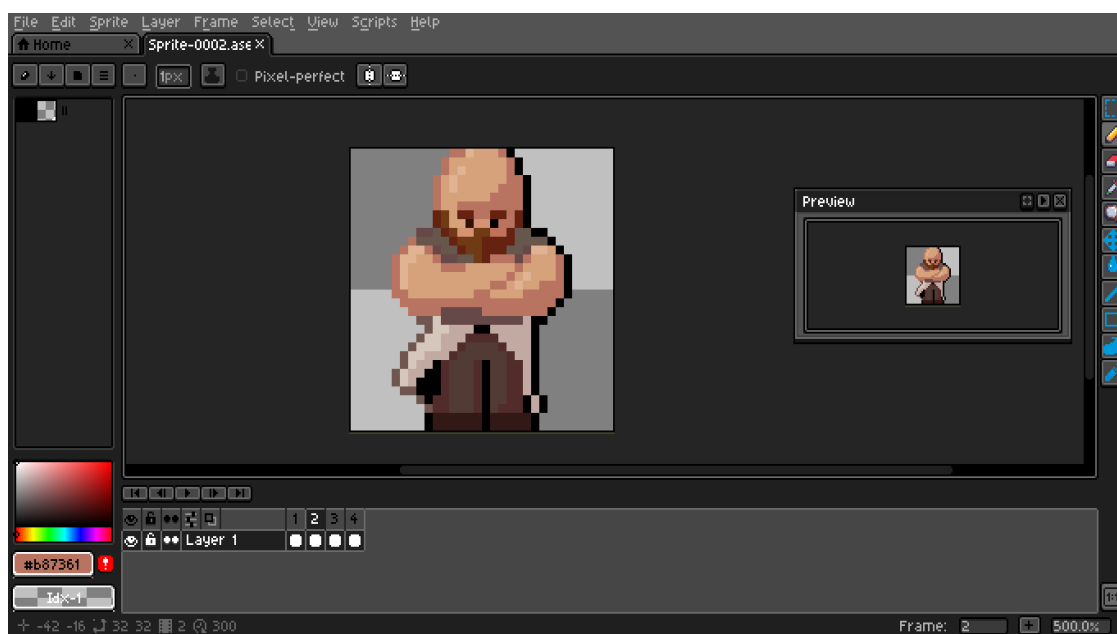


Figura 11 – *Sprite* feito no programa *Libresprite*. Fonte: Imagem do autor

Por meio da Figura 11 é possível ver que este software possui diferentes seções como a escolha de cores do pixel a ser desenhado, permite a utilização de fundos com transparência, possui um conjunto de ferramentas à direita que podem ser usadas para desenhar diversas formas geométricas facilmente.

Abaixo a mesma possui uma linha do tempo com diferentes números que vão de 1 até 4, esses representam os *sprites*, cada um destes possui uma imagem parecida mas com leves alterações do mesmo personagem, quando trocadas em sequência em uma ordem específica uma animação é criada.

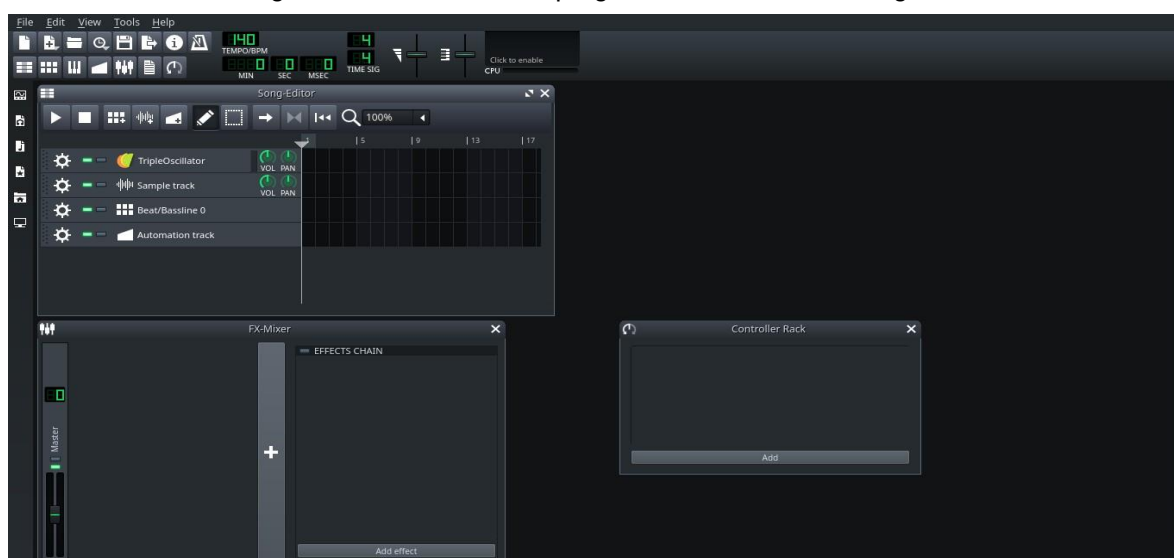
Essa técnica de criação de movimento por meio de imagens pode ser utilizada não só para o personagem controlado pelo jogador mas também por inimigos, itens coletáveis, cenários e até mesmo elementos visuais da interface do usuário.

3.5 MÚSICA E EFEITOS SONOROS

Assim como a arte visual do jogo, a área de efeitos de sons e músicas não pode ser ignorada, sendo muito importante para a experiência geral do jogador. Além de permitir uma maior imersão no mundo do jogo, o design de sons se mostra muito eficaz em fazer com que o jogador expresse ou sinta sentimentos que o criador da obra deseja (GAME, 2020).

O desenvolvimento de sons do jogo fará uso de uma ferramenta conhecida como *Linux Multi-Media Studio* (LMMS), um software utilizado para a criação de músicas por meio do arranjo de notas de diferentes instrumentos. Além de ser uma ferramenta disponível ao público em geral, a mesma não possui custos para ser utilizada e é desenvolvida por uma comunidade que a disponibiliza também como um programa de código aberto, ou seja, qualquer um pode ver qual o código a mesma utiliza

Figura 12 – Tela inicial do programa LMMS. Fonte: Imagem do autor



Na Figura 12 é possível ver que a interface de usuário é composta por janelas menores que possuem diversas funções, como a capacidade de organizar diferentes notas para formar músicas ou efeitos sonoros. O estilo musical escolhido é o de consoles da época do GBA e NES que são mais antigos mas que possuem um estilo característico da época.

4 MECÂNICAS DO JOGO

Essa seção tem o propósito de citar e explicar as principais mecânicas que o jogo deve possuir para que uma jogabilidade seja garantida.

4.1 SISTEMA DE TURNOS

Para que o jogador possa desenvolver estratégias e pensar antes de tomar decisões que possam afetar seu personagem, um sistema de turnos pode ser desenvolvido como uma mecânica principal durante as batalhas. Um sistema de turnos garante que todos os personagens envolvidos em uma cena de luta possam tomar suas próprias ações de forma organizada como se estivessem em uma fila. Em um turno, o jogador pode optar por atacar algum alvo, utilizar alguma magia, acessar seu inventário e consumir um item ou até mesmo pode fugir da luta caso isso faça parte de sua estratégia.



Figura 13 – Batalha em turnos no jogo *Golden Sun*. Fonte: <https://goldensun.fandom.com/wiki/Battle>

A ordem em que cada personagem em uma luta começa a agir será definida por um atributo de velocidade, onde personagens que possuam maior velocidade podem agir primeiro, aqueles que possuem os menores valores terão uma tendência a demorar mais a agir do que os outros.

Após todos os personagens terem agido, uma nova rodada se inicia mantendo a ordem de ação, essa é a ordem de iniciativa que deve ser mantida até o fim do combate caso uma condição de vitória seja alcançada ou os personagens do jogador sejam derrotados.

4.2 INVENTÁRIO E EQUIPAMENTO

Para que o personagem controlado pelo jogador possa guardar os itens encontrados durante suas aventuras, um sistema de inventário precisa ser implementado. Essa mecânica poderia ser facilmente implementada utilizando estruturas de dados capazes de armazenar um valor de identificação de qual é o item em um espaço específico e também o valor de quantidade, que representa quantas unidades de um certo item o personagem possui.

Certos itens são classificados como consumíveis, ou seja, podem ser utilizados mas que são gastos imediatamente, ou equipamentos, itens que podem ser equipados em um personagem para o fortalecer e que não são gastos.

4.3 PERSONAGENS NÃO-JOGÁVEIS

Os personagens não-jogáveis podem ser utilizados para preencher funções específicas dentro do mundo do jogo. Seja como um mercador que vende itens ao personagem controlado pelo jogador, inimigos que podem aparecer durante cenas de batalha, ou até mesmo personagens que interferem na história que o jogador está observando.

Uma das ferramentas que podem ser utilizadas em conjunto com os personagens não-jogáveis são as caixas de texto. Essas permitem que o jogador aprenda mais sobre o mundo em que está jogando, como as mecânicas do jogo funcionam ou até mesmo fornecer detalhes sobre a vida pessoal de cada um que o jogador interaja durante a sua jogatina. Além de caixas de texto, estes podem fazer a utilização de sons, *sprites* e animações para fomentar a experiência do usuário.

5 TESTES

Essa seção tem o intuito de apresentar as principais ferramentas utilizadas no processo de testagem do jogo e quais resultados foram adquiridos. Além da ferramenta principal que o GMS2 dispõe ao desenvolvedor, é possível criar funções customizadas para testar elementos específicos do jogo.

5.1 DEBUG

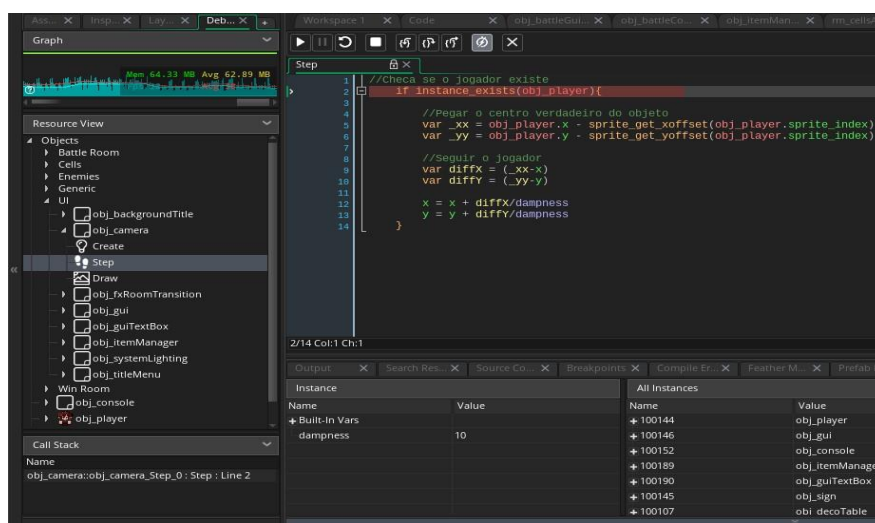


Figura 14 – Tela de debug do GMS2. Fonte: Imagem do autor

Para testar a maioria das funções, interações entre objetos e outras lógicas de programação, o GMS2 oferece um modo chamado de “debug”, onde é possível analisar o jogo mais a fundo, sendo possível ver todos os objetos que estão em uma sala, seus valores de vida, caso possuam, suas posições horizontalmente e verticalmente, entre outras informações que podem ser úteis para identificar problemas que afetem negativamente a experiência do jogador. Além disso é possível alterar suas características em tempo real, permitindo uma análise mais dinâmica de como os objetos interagem entre si.

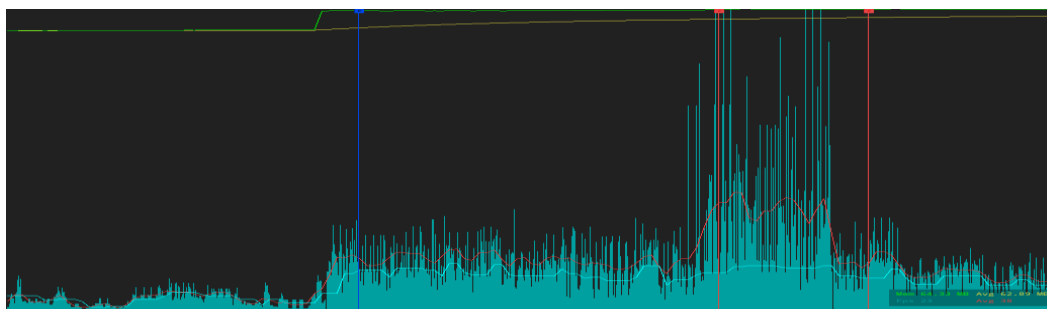


Figura 15 – Gráfico de performance do jogo em modo debug. Fonte: Imagem do autor

A ferramenta oferece um gráfico em tempo real que pode ser utilizado para analisar a performance do jogo. É possível notar que na Figura 15 a utilização de recursos do jogo possui oscilações variadas, sendo a linha azul mais escura a representação do jogo saindo o menu principal para o ambiente das diferentes salas, nesse caso é possível observar um aumento do uso de memória, sendo representada pela linha verde.

Mesmo com muitos objetos na mesma sala, o GMS2 possui ferramentas que realizam automaticamente a limpeza de variáveis que não serão mais utilizadas, ajudando no consumo de memória. Com o jogo sendo executado em uma máquina mais simples, computadores atuais possuem menos dificuldade em processar a lógica do jogo em conjunto com os visuais, já que estes são mais simples devido ao fato do jogo ser em apenas duas dimensões.

5.2 VISUALIZAÇÃO DENTRO DO JOGO

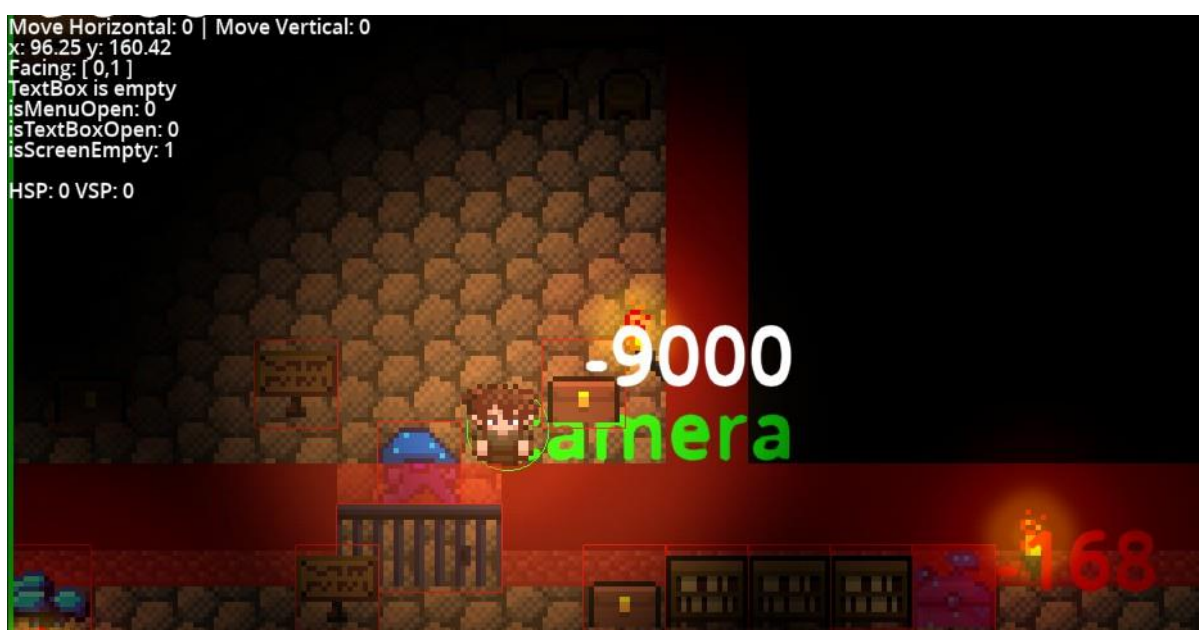


Figura 16 – Atributos de objetos mostrados em tempo real. Fonte: Imagem do autor

Foi desenvolvida uma função responsável por mostrar na tela do jogo informações sobre variáveis importantes como a velocidade vertical e horizontal do personagem, sua posição na sala, o texto que está sendo mostrado para o jogador na tela, retângulos vermelhos em volta de objetos para indicar áreas de interação, colisões com o cenário e outros atributos.

Essas informações não são mostradas ao jogador e são usadas apenas para ajudar na compreensão do resultado de testes e para entender como um problema está pode afetar a movimentação, colisão ou até interação com os objetos na sala.

Essa função foi utilizada em conjunto com uma ferramenta criada especificamente para esse projeto para ajudar a testar diferentes mecânicas básicas de um jogo RPG, sendo o nome da mesma, o “console” de desenvolvimento.

5.3 CONSOLE DE DESENVOLVIMENTO



Figura 17 – Tela do console de desenvolvimento. Fonte: Imagem do autor

Foi desenvolvido um objeto chamado “obj_console”, o mesmo foi posicionado nas salas principais. O objetivo principal de sua inclusão é permitir com que funções essenciais possam ser testadas e o resultado seja mostrado na tela dentro do próprio jogo, sem depender de ter o projeto aberto no GMS2.

Como é possível ver na imagem acima, este objeto possui duas áreas diferentes, a superior que é responsável por mostrar o resultado dos comandos executados, e a parte de baixo que possui um campo de inserção de texto onde comandos podem ser digitados e executados ao pressionar a tecla “enter”.

Este console foi desenvolvido com funções de histórico de comandos e de uma função de ajuda para facilitar o processo de execução de testes, sendo possível adicionar itens, mover objetos para locais diferentes, modificar certos atributos e outras funções que não dependem da execução do projeto dentro da ferramenta GMS2.

As mecânicas principais do jogo possuem uma série de condições que podem retornar valores de 1 ou 0, representando se a ação teve ou não sucesso, em certos casos, é comum uma função retornar um valor 0, como em testes de controle, onde nem sempre o jogador vai estar pressionando algum botão no teclado, e nesse caso, o valor para um botão retornará 0. O resultado de cada função é mostrado apenas durante o desenvolvimento dentro do GMS2.

Tabela 1 – Resultados dos testes de funções

Função testada	Situação	Resultado	Esperado
Adição de itens ao inventário	Com espaços livres	1 (true) Sucesso	1
Adição de itens ao inventário	Já tendo este item no inventário (acumulável)	1 (true) Sucesso	1
Adição de itens ao inventário	Sem espaço no inventário ou já tendo este item no inventário (não-acumulável)	0 (false) Falha	0
Controle	Botão foi pressionado	1 (true) Sucesso	1
Controle	Botão não foi pressionado	0 (false) False	0
Botão da UI	Botão “Z” não pressionado	1 (true) Sucesso	1
Mudar de turno	Nome do turno atual não ser “next”	0 (false) Falha	0
Mudar de turno	Nome do turno atual ser “next”	1 (true) Sucesso	1

6 RESULTADOS

Esta seção tem o propósito de documentar como as principais mecânicas do jogo foram desenvolvidas, focando nos códigos mais relevantes e suas devidas funções.

6.1 PERSONAGEM PRINCIPAL

O personagem principal é o objeto que recebe as principais mecânicas durante todo jogo, ele possui a função de representar o jogador dentro do mundo virtual do jogo, este é capaz de interagir com si mesmo e com o ambiente a sua volta. Para que isso fosse possível, foram desenvolvidas diferentes mecânicas que tinham o foco em dar mais controle e liberdade ao jogador.

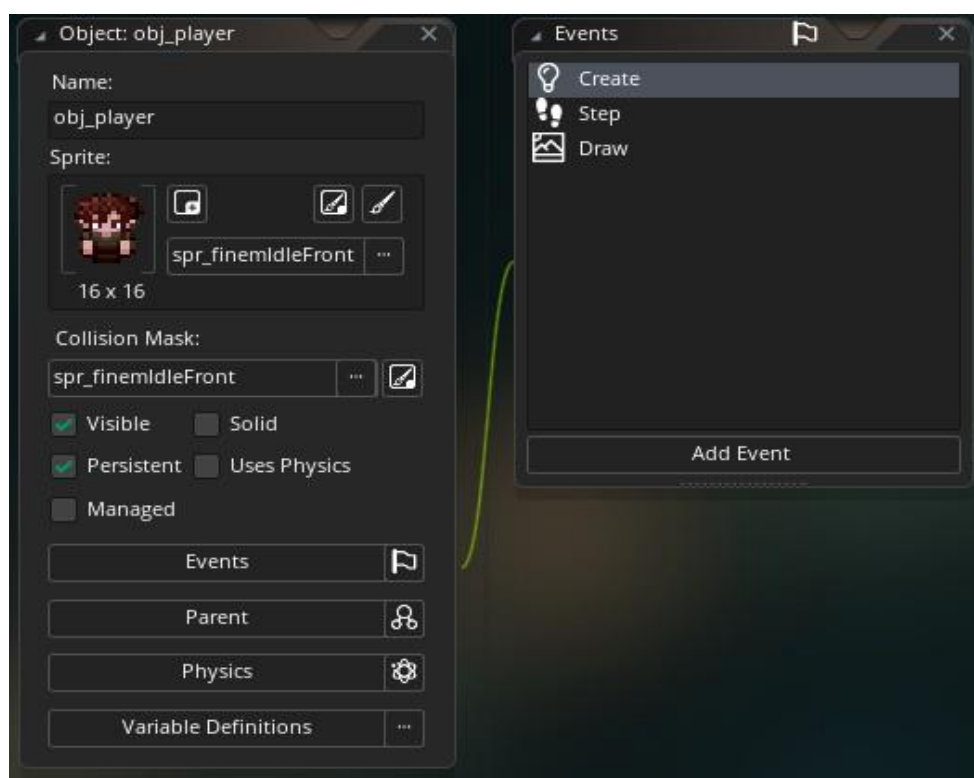


Figura 18 – Estrutura visual do personagem principal . Fonte: Imagem do autor

Na Figura 18 é possível ver como os objetos são representados no GMS2, possuindo atributos como um nome próprio, um sprite que vai definir a imagem do objeto, opções que permitem que o personagem seja afetado por sistemas de física, entre outras funções. No lado direito é possível ver a janela de eventos, uma das partes mais importantes desse motor de jogos, com os eventos é possível fazer um personagem pular, um inimigo causar dano, criar efeitos de iluminação, etc.

Eventos controlam ações e reações que um objeto pode ter, no caso do jogador é possível ver que existem três eventos definidos, um *create event*, que vai executar linhas de código apenas uma vez, que ocorre quando o objeto é criado na cena, um *step event*, que vai ser executado várias vezes por segundo, e por fim um *draw event*, que é responsável por mostrar algo na tela, como a imagem do personagem, um texto, formas geométricas, entre outros.

6.1.1 Controle

O foco principal do jogo é na utilização de um teclado, devido a isso, a maioria dos controles utilizam diferentes teclas para que seja possível acessar menus, interagir com objetos no mapa ou realizar outras ações. Para a movimentação, o jogador tem duas opções principais, usar as teclas de setas ou usar as teclas “wasd”, isso deve-se ao fato de que este jogo foca em plataformas de computadores.

Para confirmar uma ação, interagir com objetos ou personagens e avançar para o próximo diálogo usam a tecla de confirmação “Z”. Para voltar uma ação ou acelerar conversas, o botão “X” foi escolhido. Para acessar o menu principal dentro do jogo deve-se usar o botão “Esc” e por fim, para acessar o inventário deve-se usar o botão “C”. A escolha desses botões possui uma relação com o RPG Maker, um motor de jogos que possui o foco em criação de jogos do gênero RPG e que usava um sistema de controle que foi usado como inspiração para esse.

6.1.2 Movimentação

O personagem principal possui dois tipos diferentes de velocidades, a vertical e a horizontal, por meio destas é possível alterar a posição dele no mapa de acordo com a tecla pressionada no teclado. Para descobrir a direção do movimento foi criada uma variável chamada “move_dir” que é um vetor de duas dimensões, a primeira carrega o movimento horizontal, se for 1 o personagem se move para a direita, -1 se for para a esquerda e 0 se não tiver movimento horizontal, a segunda dimensão do vetor carrega a movimentação vertical, sendo 1 para baixo, -1 para cima e 0 para nenhum movimento.

```
hsp = clamp(hsp,-sqrt(abs(maxSpd/2)),sqrt(abs(maxSpd/2)))
```

Esse código tem três partes, “clamp” que define um limite mínimo e máximo para a velocidade horizontal, a fórmula que utiliza a velocidade máxima geral para adquirir o valor máximo de velocidade diagonal que o personagem vai utilizar e por fim é realizada a re-aplicação da fórmula para adquirir a velocidade negativa. Este código é repetido para o movimento vertical, usando “vsp” como variável.

Para a colisão, o personagem utiliza a função “place_meeting()” para checar se ele está no mesmo espaço que algum outro objeto. Para generalizar os objetos que possuem colisão, todos são tratados como “obj_colission”, ou seja, o personagem checa por uma colisão com qualquer objeto que tenha “obj_collision” como “objeto pai”. No caso de uma colisão detectada, a velocidade do personagem é zerada, impedindo que ele atravesse paredes, baús, e outros objetos físicos.

6.1.3 Status

Além de poder se movimentar e colidir com o cenário, o personagem possui valores que definem a sua força, como os “hp” que são os pontos de vida, estes são reduzidos quando o personagem sofre dano de algum inimigo, também podem ser recuperados usando poções de cura ou descansando na cama que se encontra no canto esquerdo superior da primeira sala.

```
//Status
hpMax = 10
hp = hpMax

spMax = 3
sp = spMax

atribStr = 1
dmg={min:0,max:1} //Min e Max, bônus e quanto do bônus ele perde no começo de cada turno
dmgMagic={min:0,max:0,type:"none"}

atribCon = 1
def = {min:0,max:1}

level = 1
xp = [0,5,0]

name = "Finem"
```

Figura 19 – Código do status do personagem principal. Fonte: Imagem do autor

Como é possível ver na Figura 19, existem outros atributos que compõem o personagem principal, a maioria está relacionada a mecânica de combate, como o dano, dano mágico, defesa, atributo de força, entre outros. Algumas dessas variáveis podem ser aumentadas de acordo com o equipamento que o jogador coletar, equipar uma adaga aumenta o dano mínimo e máximo comparado a não usar nenhum equipamento de ataque.

6.1.4 Habilidades

Durante um combate, o jogador pode optar por executar outras ações além de um ataque contra um oponente, essas são ações especiais e uma delas é a lista de habilidades.

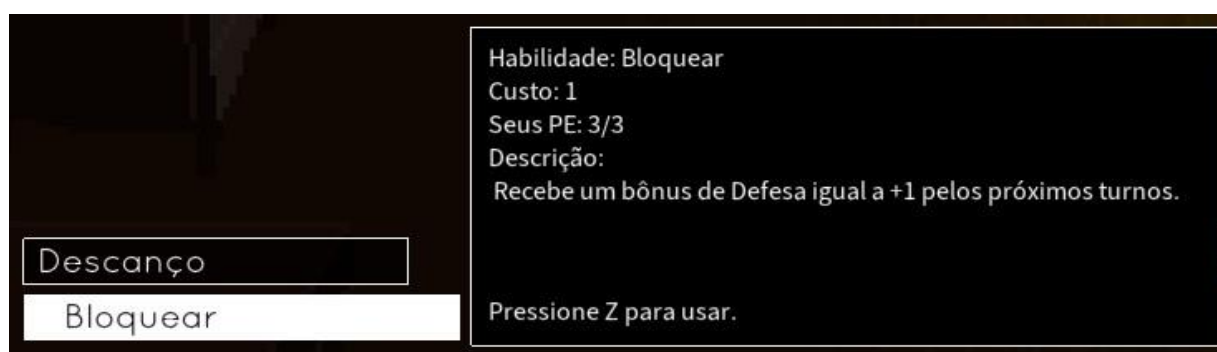


Figura 20 – Exemplo de habilidade dentro da batalha. Fonte: Imagem do autor

Na imagem acima, podemos ver que cada habilidade possui um nome, um custo, que representa quantos pontos de energia serão gastos ao executar essa ação, quantos pontos atuais e máximos o personagem possui, uma descrição breve do que a habilidade faz e por fim uma dica de qual botão apertar para usar a ação, caso o personagem não tenha pontos o suficiente a janela treme e assume um tom de vermelho e avisa que faltam pontos de energia. Novas habilidades podem ser adquiridas ao usar itens específicos ou passar de nível.

6.1.5 Visual

Todas as imagens usadas para representar o personagem, seus ataques e quaisquer outros objetos dentro do jogo foram desenhados com a ferramenta Libresprite.

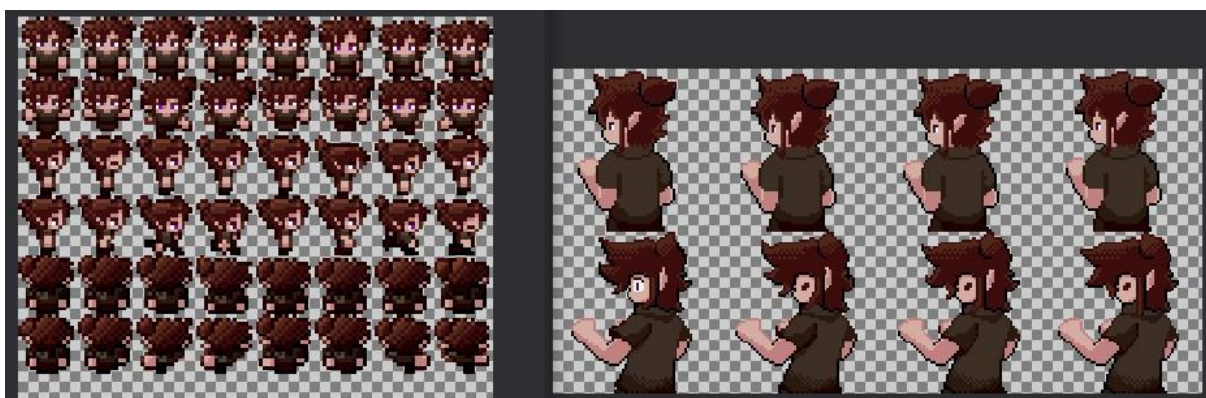


Figura 21 – Sprites do personagem. Fonte: Imagem do autor

Para representar o personagem principal no mundo ou dentro do combate duas imagens foram criadas, a primeira contém todas as imagens do personagem parado ou andando, com variações do mesmo indo para a direita, baixo ou para cima.

Nesse caso é notável que não existem sprites do personagem olhando para a esquerda mesmo que ele faça isso dentro do jogo, para evitar que mais duas variações sejam criadas é possível usar uma técnica que inverte o sprite do personagem quando o mesmo anda para a esquerda usando uma variável chamada “image_xscale”. A mesma controla o comprimento de uma imagem dentro do GMS2. Um valor de 1 faz com que a largura dentro do jogo seja a mesma da imagem original, um valor de 0 faz com que o sprite não tenha comprimento, fazendo-o desaparecer, e por fim, um valor de -1 faz com que um sprite seja invertido.

A segunda imagem possui menos sprites porém estes têm um tamanho maior, a linha de cima é a animação do personagem parado e a linha de baixo representa um ataque que causou dano no personagem, estes sprites são usados durante as batalhas.

6.2 INIMIGOS

Para oferecer um desafio ao jogador, alguns inimigos foram criados para tornar o sistema de combate dinâmico, possuindo inimigos com muitos pontos de vida, ou que causam muito dano, possuem muita defesa ou que oferecem recompensas ao jogador.

Estes funcionam como obstáculos que devem ser derrotados para que seja possível progredir no jogo e eventualmente chegar no final. Como base, todos os inimigos recompensam o jogador com pontos de experiência no final da batalha, com alvos mais fortes tendo valores de pontos mais altos que serão utilizados no cálculo do nível atual do personagem principal

6.2.1 Estrutura

```
function enemy(_id) constructor{
  id = _id
  x = 0
  y = 0
  state = 1
  checkTip = "Nenhuma informação adicional."
  imageIndex = 0
  alpha = 1

  //0- Nome do inimigo 1- Nome do player 2- dano total normal 3- dano magico 4- tipo de dano magico
  msg = {hit:["{0} ataca {1} e causa {2} de dano..."],
        miss:["{0} tentou atacar {1} mas errou..."]}

  //Stats
  hp = 7
  def = {min:0,max:0,bonus:0}
  dmg = {min:0,max:1,bonus:0,type:"normal"}
  dmgMagic = {type:"",min:0,max:0}
```

Figura 22 – Estrutura básica dos inimigos. Fonte: Imagem do autor

Todos os inimigos possuem uma mesma estrutura base, na imagem acima é possível ver quais atributos fazem parte de um inimigo, isso inclui os “hp” que são pontos de vida, “def” que define quanto de resistência a dano um inimigo pode ter, “state” define se o inimigo está vivo ou não, se for 1 ele pode agir em seu turno, se for 0 ele pula a vez e seu sprite não é mostrado na tela, “dmgMagic” indica se um inimigo pode causar dano que não seja só físico já que certos inimigos podem causar dano elemental, como dano de fogo por exemplo. Além dessas variáveis base, todos os inimigos têm a capacidade de realizar e sofrer ataques do jogador.

```

//IA/Pensar na ação
function getAction(){
//Atacar
function actionAttack(_player,_x,_y){
//Sofrer dano
function takeDamage(_dmg){

```

Figura 23 – Funções do inimigo. Fonte: Imagem do autor

A Figura 23 representa as funções principais usadas durante um combate para causar ou receber dano em relação ao personagem principal. A função “getAction” simplesmente recebe o nome do inimigo e retorna a próxima função de ataque. A função “actionAttack” realiza cálculos usando o valor de dano mínimo e máximo do inimigo atual, compara com a defesa do jogador e causa dano.

Por fim, a função “takeDamage” só é executada quando o jogador executar um ataque ou realizar uma habilidade que cause dano no inimigo, caso a vida do inimigo chegue a zero, o jogador recebe um valor de “xp” que representam pontos de experiência, usados para calcular o nível atual do personagem principal, além disso existe uma chance do inimigo deixar um item para trás, como uma poção de cura ou uma arma melhor.

6.2.2 Sprites



Figura 24 – Sprites dos inimigos básicos. Fonte: Imagem do autor

Assim como o personagem principal, cada inimigo tem seu sprite fora de combate, dentro do combate atacando ou sendo atacado, no entanto como os inimigos funcionam como obstáculos que impedem o jogador de progredir, estes não necessitam de sprites de movimento. Cada animação possui quatro sprites diferentes, totalizando três animações para cada inimigo básico, duas para o boneco de treinamento e um conjunto de animações especiais para o chefe final do jogo. Ao sofrer um ataque, além de trocar para a animação de sofrer dano, o inimigo tem seu último sprite travado por mais tempo para ajudar o jogador a perceber que o ataque foi efetivo, em situações onde o personagem principal faz um ataque mas não causa dano, o inimigo não tem sua animação trocada.

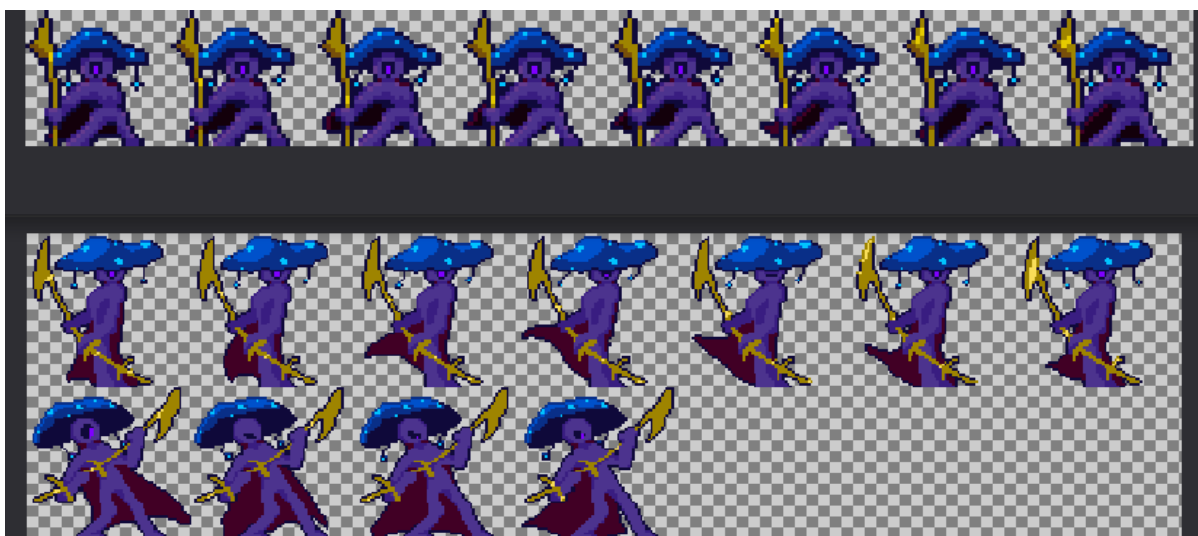


Figura 25 – Sprites do chefe final. Fonte: Imagem do autor

O chefe final possui ao todo sete sprites de animação parado em combate, quatro ao ser atacado e oito sprites em sua versão fora de batalha, seus sprites estão separados dos inimigos mais comuns devido a só existir um chefe no jogo inteiro, que é o último inimigo a ser derrotado para que o jogador consiga ganhar o jogo.

6.3 INVENTÁRIO E EQUIPAMENTOS

Para representar um dos fatores da evolução de um personagem, o jogador pode coletar diferentes itens pelo jogo, como por exemplo as chaves que o permitem mudar de sala e eventualmente chegar ao chefe final, armaduras que ajudam a reduzir o dano sofrido, itens consumíveis que podem recuperar pontos de vida, entre muitos outros.



Figura 26 – Interface do inventário. Fonte: Imagem do autor

Na Figura é possível ver o resultado final da interface do usuário, possuindo uma área que mostra os itens que o personagem possui e se estão equipados, uma área que explica o que o item é e o que ele faz, junto com as possíveis ações que podem ser executadas com ele, e por fim uma janela de “status” que mostra as informações vitais do personagem como seus pontos de vida, o nível, defesa, entre outros.

Este menu pode ser acessado dentro do jogo enquanto o jogador explora as salas ou até mesmo dentro de batalha para conferir como o personagem principal está e quais itens pode usar. Além disso, itens como armas possuem informações extras que o jogador pode analisar por meio da janela do item em si, essas informações podem variar como o dano normal, dano mágico, entre outras.

6.3.1 *Inventário*

```
//Inventário
inventorySize = 5
inventory = ds_map_create()

//Equipamento
equipment = {
    weapon: "",
    armor: ""
}
```

Figura 27 – Estrutura do inventário. Fonte: Imagem do autor

Para que o personagem principal seja capaz de carregar itens, foi optado por utilizar um tipo de variável chamada “map” que no GMS2 possui dois atributos, uma chave e um valor. Com esse tipo de estrutura, o personagem pode carregar um item como por exemplo uma poção de cura, a chave será um identificador único que nesse caso é o nome do item e o valor dele vai ser tudo o que a poção de cura contém, como a quantidade de poções, valor de cura, etc.

6.3.2 *Estrutura dos itens*

```
//Armas
itemAdd({key:"dagger",
    name:"Adaga",
    description:"Uma adaga antiga, pode ser utilizada como arma mesmo enferrujada.",
    type:"weapon",
    dmg:{min:1,max:2,type:"normal"},
    dmgMagic:{min:0,max:0,type:"none"},
    effect:"none",
    actions:["Equipar","Largar"],
    isStackable:false})
```

Figura 28 – Estrutura dos itens. Fonte: Imagem do autor

Como é possível ver na Figura 28, um item possui diferentes atributos que o compõem, nesse caso, armas possuem um valor chamado “dmg” e “dmgMagic” que representam respectivamente o valor de dano normal e dano mágico que uma arma pode causar em um oponente. O atributo “actions” representa as ações que podem ser realizadas com aquele item, quando a opção de “equipar” é executada, a adaga tem sua chave armazenada no espaço de armas nos equipamentos do jogador e pode ser removida posteriormente.

6.4 CENÁRIO

Como objetivo principal do jogo, o personagem principal precisa chegar até a última sala para poder escapar, caso o jogador consiga fazer isso ele consegue terminar o jogo e seus pontos adquiridos serão mostrados na tela final. Para aumentar oferecer mais opções ao jogador, quatro salas principais existem no jogo, que cada uma possui uma porta de cela que deve ser aberta com chaves espalhadas pelo mapa.

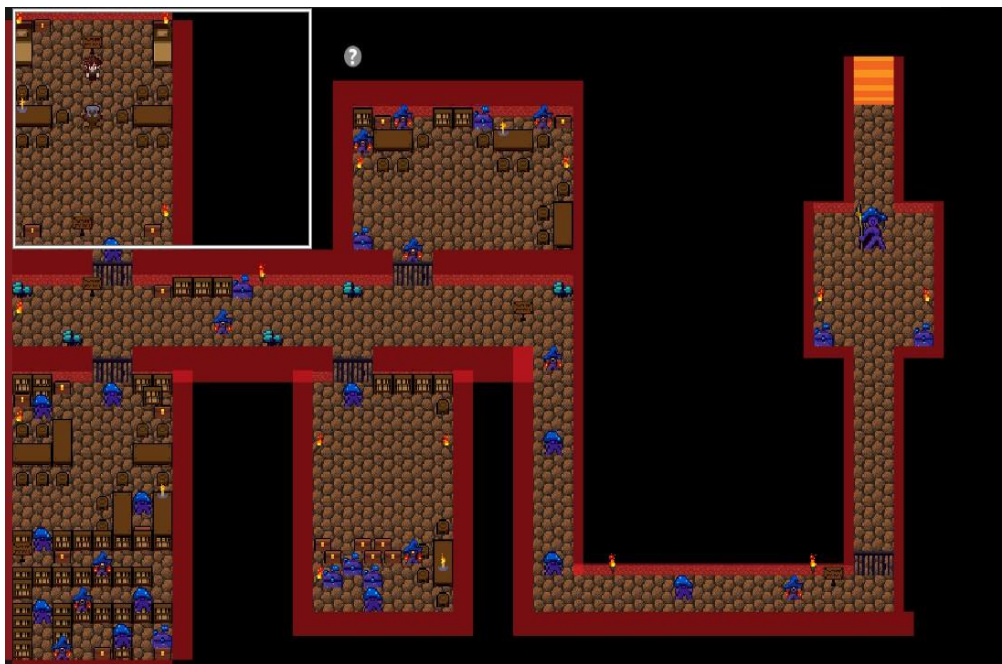


Figura 29 – Imagem do mapa completo. Fonte: Imagem do autor

Como é possível ver na Figura 29, o jogador começa na primeira sala que está representada dentro do retângulo branco, ao sair dessa sala ele tem algumas opções antes de ir direto para o chefe final, já que o mesmo foi criado como um desafio e o jogador precisa tornar o personagem principal forte o suficiente para terminar o jogo. Alguns baús podem ser encontrados no mapa e possuem duas versões, um baú normal que pode conter um item útil ao personagem, e um baú que serve como armadilha e tem um sprite diferente, sendo opcional mas que ao ser derrotado fornece um tesouro raro.

6.5 OBJETIVO

O jogador precisa derrotar os oponentes espalhados pelo mapa para adquirir pontos de experiência, equipamentos, subir de nível, aumentar características como pontos de vida e energia, conseguir novas habilidades entre outras mecânicas. Caso o jogador consiga derrotar o chefe final, sua pontuação será mostrada na tela que utiliza a quantidade de dias gastos, pontos de experiência, nível atual e porcentagem de vida restante para calcular quantos pontos o jogador ganhou.



Figura 30 – Tela final de pontuação. Fonte: Imagem do autor

Na Figura 30 é possível ver um cenário simples em conjunto com uma janela que mostra quantos pontos o jogador ganhou para cada tipo de ação específica. Além de apresentar ao jogador quantos pontos este fez como uma recompensa, ainda indica quais aspectos contribuem mais para sua pontuação.

6.6 SISTEMA DE COMBATE



Figura 31 – Cena de batalha com inimigos. Fonte: Imagem do autor

A mecânica principal que foi desenvolvida para o jogo foi a de combates que utilizam um sistema de turnos para marcar a vez de cada inimigo ou do personagem principal. O jogador sempre começa o combate primeiro e os inimigos tem um turno cada um, após o último inimigo realizar sua ação a rodada termina e o jogador pode jogar novamente, o ciclo se repete até o jogador ganhar por derrotar todos os oponentes, ou caso a vida do personagem principal chegue a zero, neste caso ele é expulso da batalha mas fica com um ponto de vida, neste caso todo o dano causado aos inimigos é ignorado.

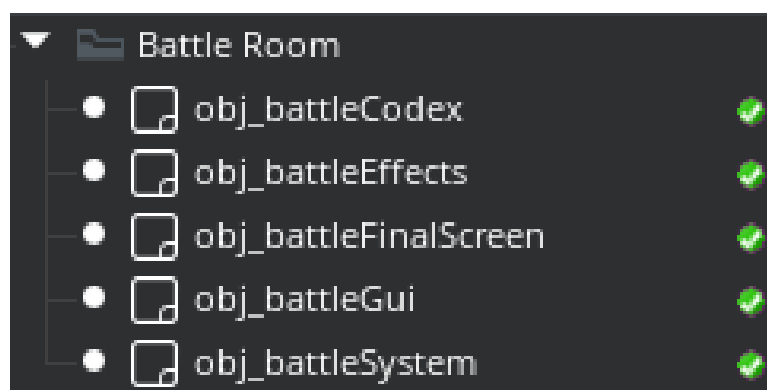


Figura 32 – Objetos da sala de batalha. Fonte: Imagem do autor

A Figura 32 apresenta todos os objetos que são responsáveis por permitir que o combate ocorra, cada um tem suas próprias funções e atributos. O objeto “codex” é responsável por armazenar informações sobre inimigos, como pontos de vida, defesa, dano, recompensas de itens ou pontos de experiência ao serem derrotados, entre outros, é por meio desse objeto que os inimigos podem ser adicionados à luta, caso um inimigo novo seja desenvolvido, suas informações residirem nesse objeto.

O objeto de “effects” representa alguns efeitos visuais que ocorrem durante a batalha, estes incluem os valores de dano que aparecem quando um oponente ou o personagem do jogador causa dano, o símbolo de certas habilidades que aparecem ao serem utilizadas e efeitos de ataque como cortes ou explosões.

O objeto de “finalScreen” representa a tela final que aparece quando todos os inimigos são derrotados, esta possui funções para mostrar os pontos de experiência adquiridos mostra mensagens caso o jogador passe de nível, adquira novas habilidades ou colete itens.

O objeto “gui” representa a maior parte das interfaces de usuário, com o menu de escolha de ações, a lista de habilidades, as imagens do personagem do jogador e dos inimigos, botões, etc. É por meio deste objeto que o jogador consegue ver o resultado de suas ações com objetos no mundo, fora da cena de batalha, como por exemplo ler placas ou interagir com celas.

Por fim, o objeto “system” é responsável por executar a parte lógica das batalhas, como passar de turno, ataques, uso de itens, checagem se todos os inimigos foram derrotados, etc.

7 CONCLUSÕES E TRABALHOS FUTUROS

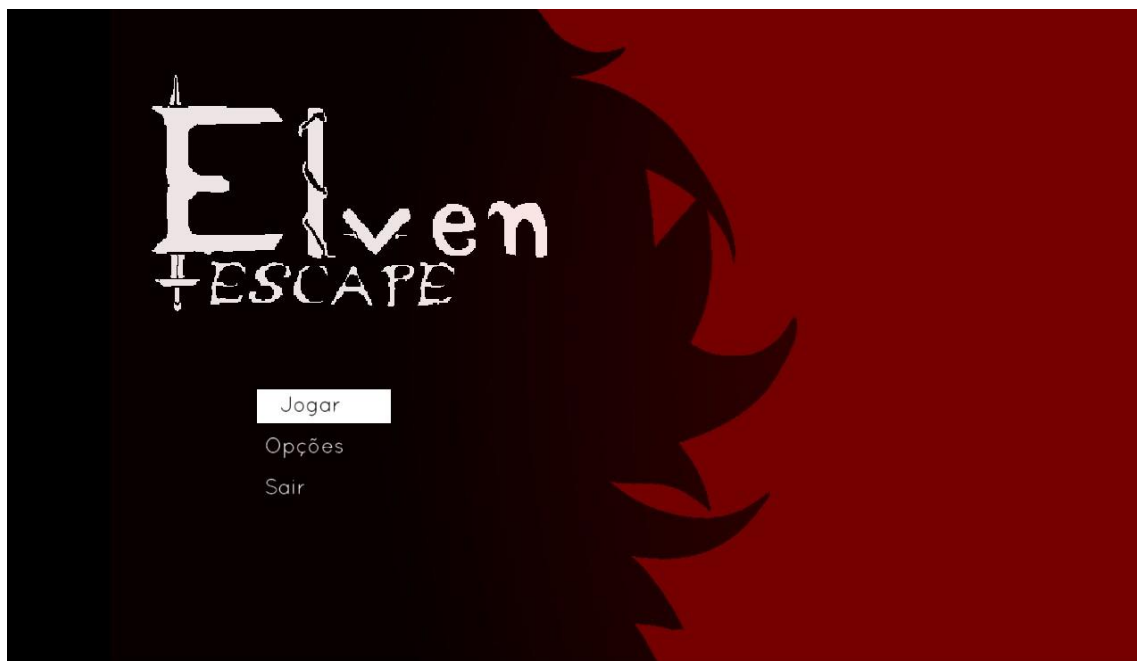


Figura 33 – Imagem do menu inicial finalizado. Fonte: Imagem do autor

Na Figura 33 é possível analisar o resultado final para a interface de usuário que o jogador encontrará pela primeira vez ao iniciar o jogo, a mesma combina elementos como botões animados, transições de tela e a execução de um vídeo de fundo para contribuir na ambientação do jogo. É possível assistir o vídeo completo de como jogar Elven Escape por meio da plataforma YouTube pelo link: <https://youtu.be/StsWHV7mutU>.

Em suma, os objetivos específicos propostos para a conclusão do projeto foram finalizados com sucesso. Criou-se objetos que carregam os códigos necessários de mecânicas como movimento, colisão e interação com o ambiente. Este objetivo fez a utilização da linguagem GML, resultando em uma série de funções que foram essenciais para garantir os requisitos mínimos de um jogo RPG.

Também foram implementados os gráficos utilizados pelos objetos por meio da ferramenta gratuita Libresprite para criação de sprites e animações no estilo *pixel art*. Os efeitos sonoros e trilha sonora que são usadas para fomentar a experiência do jogador foram criados por meio do programa LMMS, que combina uma série de instrumentos musicais que podem ser utilizados para compor trilhas sonoras. Os diferentes ambientes que o jogador vai poder explorar foram criados com o editor de salas do próprio GMS2 em conjunto com sprites de mesas, paredes, portas, camas, entre outros objetos previamente criados no Libresprite.

Por fim, os equipamentos, itens consumíveis, inimigos e outros componentes básicos de um RPG, foram implementados por meio da utilização de *structs* que são variáveis que o GMS2 utiliza para guardar diferentes tipos de atributos em um só objeto, como variáveis de texto, números, etc.

O objetivo geral de desenvolver um jogo RPG que possua um sistema de batalhas em turno, utilizando como motor de jogos o programa GMS2 foi concluído com sucesso, combinando as diferentes mecânicas em um só programa que pode ser executado nas plataformas suportadas sem a necessidade de possuir o GMS2 instalado.

Este projeto revela o resultado da união de mecânicas relativamente simples, mas que em conjunto podem formar um produto final completo e que pode ser utilizado para arrecadação de fundos por meio de sistemas de arrecadação monetária ou até mesmo programas do governo para o incentivo ao desenvolvimento da indústria de jogos brasileira.

Após todo o desenvolvimento deste projeto, é notável que mesmo com a disponibilidade de ferramentas gratuitas no mercado, um dos fatores que mais influenciam não só na qualidade de um jogo mas também no seu tempo total de criação é a experiência dos envolvidos em conjunto com a quantidade de pessoas trabalhando nesse mesmo projeto.

A criação de jogos pode sim ser feita por uma só pessoa, mas este tipo de metodologia deve ser evitada pois a criação de jogos envolve o conhecimento de diferentes áreas, como a programação, que dependerá do motor de jogos a ser utilizado, conhecimentos artísticos para criação de imagens de alta qualidade, experiência com criação de músicas, estudos sobre design de interfaces de usuário, e muitas outras áreas que compõem a criação de uma experiência digital.

No entanto isto não deve ser utilizado como uma forma de afastar as pessoas do desenvolvimento de jogos, mas sim uni-las por meio de iniciativas públicas e privadas, resultando na criação de diferentes estúdios e times de desenvolvimento, eventualmente fomentando o mercado brasileiro de jogos com a criação de novos empregos e na contratação de profissionais.

Como possíveis melhorias que poderiam ser feitas em um futuro trabalho, é notável que mesmo algumas mecânicas do jogo serem funcionais, estas ainda precisam ser polidas para alcançar resultados excepcionais, isso envolveria principalmente o aumento do conhecimento nas áreas de arte visual, sonora, *game design*, criação de interfaces e outras.

O aumento na quantidade de inimigos, criação de mecânicas mais complexas no combate, permitindo que os oponentes possam realizar outras ações além de atacar. Criação de um menu mais complexo, oferecendo mais opções de customização ao jogador, expansão de itens disponíveis ao jogador como novas armas, armaduras, itens especiais, consumíveis, entre outros.

Por fim, outra adição útil seria criar um sistema para salvar o jogo a qualquer momento em conjunto com a pontuação do jogador, o que envolveria explorar como o GMS2 se comunica com o sistema de arquivos de diferentes plataformas, já que cada uma tem uma forma específica de interação com arquivos.

Para executar o jogo e ver o projeto é possível acessar o repositório do GitHub por meio deste link: <https://github.com/Hecori/elvenescape>.

REFERÊNCIAS

CARDOSO, Marcos V.; GUSMÃO, Cláudio, HARRIS, Jonathan J. (Org). **Pesquisa da indústria brasileira de games 2023**. ABRAGAMES: São Paulo, 2023.

GO GAMERS, **Pesquisa Game Brasil**. SX Group, 2024. Disponível em: <<https://www.pesquisagamebrasil.com.br/pt/>>. Acesso em: 10 fev. 2025.

GAME, **User Experience And Player-Centered Design**. [S. l.]: Springer International Publishing, 2020. ISBN 9783030376437, 3030376435. Disponível em: https://www.google.com.br/books/edition/Game_User_Experience_And_Player_Centered/76rbDwAAQBAJ?hl=pt-BR&gbpv=0. Acesso em: 20 fev. 2025.

FGV, **Uso de TI no Brasil: País tem mais de dois dispositivos digitais por habitante, revela pesquisa**. FGV, 2023. Disponível em: <<https://portal.fgv.br/noticias/uso-ti-brasil-pais-tem-mais-dois-dispositivos-digitais-habitante-revela-pesquisa#:~:text=Em%20rela%C3%A7%C3%A3o%20a%20computadores%2C%20o,12%2C4%20milh%C3%B5es%20de%20unidades.>>>. Acesso em: 20 de mar. de 2025.

GOLDENBOY, Saiba o que é RPG e quais são os jogos mais populares - Canaltech. **Canaltech**, 2022. Disponível em: <<https://canaltech.com.br/games/o-que-e-rpg-os-mais-populares>>. Acesso em: 11 de fev. de 2025.

GOMES, Delfin. **Manual Game Maker II**. Lulu.com, 2014. Disponível em: <https://www.google.com.br/books/edition/Manual_Game_Maker_II/NHmwBgAAQBAJ?hl=pt-BR&gbpv=0> Acesso em: 20 fev. 2025.

LEWIS, M.; JACOBSON, J. Game engines. **Communications of the ACM**, v. 45, n. 1, p. 27, 2002.

REINOSO, L. F., et al. Childhood Fears. In: **Games of Global Game Jam**, Califórnia, 2015. Disponível em: <<https://globalgamejam.org/2015/games/childhood-fears>>. Acesso em: 10 fev. 2025.

REINOSO, Luiz Fernando; RIOS, Renan Osório; TEIXEIRA, Giovanny Frossard. **Jogos digitais: princípios, conceitos e práticas**. Edifes, 2018.

SCHUYTEMA, P. **Game design: A practical approach**. [S.l.]: Charles River Media Boston, MA, 2007.

SILBER, Daniel. **Pixel Art for Game Developers**. CRC Press, 2015. Disponível em: <https://www.google.com.br/books/edition/Pixel_Art_for_Game_Developers/OmBECgAAQBAJ?hl=pt-BR&gbpv=0> Acesso em: 20 fev. 2025.

SQUARE ENIX, **SQUARE ENIX - Games - CHRONO TRIGGER**. Disponível em: <https://www.square-enix-games.com/pt_BR/games/chrono-trigger>. Acesso em: 19 fev. 2025.

STATISTA, **Role Playing Games - Worldwide | Statista Market Forecast**. Statista, 2024, Disponível em: <<https://www.statista.com/outlook/amo/app/games/role-playing-games/worldwide>>. Acesso em: 13 fev. 2025.

STATISTA, **Number of PC gamers worldwide 2024 | Statista**, 2021 Disponível em: <<https://www.statista.com/statistics/420621/number-of-pc-gamers/>>. Acesso em: 20 fev. 2025.

VGSALES, **Final Fantasy | Video Game Sales Wiki | Fandom**. Fandom, 2024 ,Disponível em: <https://vgsales.fandom.com/wiki/Final_Fantasy#Final_Fantasy_I%E2%80%93VI_series>. Acesso em: 20 fev. 2025.

YIN-POOLE, Wesley. Baldur's Gate 3 Dev Larian's Huge 2023 Profits Revealed. IGN, 15, nov e 2024. Disponível em: <<https://www.ign.com/articles/baldurs-gate-3-dev-larians-huge-2023-profits-revealed>>. Acesso em: 21, fev. 2025.