

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS
CÂMPUS ANÁPOLIS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JOÃO VICTOR PEREIRA ZEFERINO

**COLHEITA DO SABER
UM JOGO QUE ENSINA AS CRIANÇAS, SEM ESQUECER DA
DIVERSÃO E RELAXAMENTO**

**ANÁPOLIS-GO
2024**

JOÃO VICTOR PEREIRA ZEFERINO

**COLHEITA DO SABER
UM JOGO QUE ENSINA AS CRIANÇAS, SEM ESQUECER DA
DIVERSÃO E RELAXAMENTO**

Projeto de TCC II do Instituto Federal de Educação,
Ciência e Tecnologia de Goiás, Câmpus Anápolis,
como parte dos requisitos necessários para obtenção do
título de Bacharelado em Ciência da Computação.

Orientador(a): Prof(a). Alexandre Bellezi José

Coorientador(a): David Andrade

**ANÁPOLIS-GO
2024**

Dados Internacionais de Catalogação na Publicação (CIP)

ZEFERINO, João Victor Pereira

Z43c Colheita do saber: um jogo que ensina as crianças, sem esquecer da diversão e relaxamento./ João Victor Pereira Zeferino. – Anápolis: IFG, 2025.
53 f. il. color.

Orientador: Prof. Me. Alexandre Bellezi José.

Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Campus Anápolis – Bacharelado em Ciência da Computação, 2025.

1. Jogos educativos. 2. Gamificação. 3. Alfabetização. 4. Aprendizagem lúdica. 5. Educação.

I. JOSÉ, Alexandre Bellezi (orient.).

II. Título.

CDD 371.33

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Na presente data realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulada **CULTIVANDO O SABER UM JOGO QUE ENSINA AS CRIANÇAS, SEM ESQUECER DA DIVERSÃO E RELAXAMENTO**, sob orientação de Alexandre Bellezi Jose , apresentada pelo aluno **João Victor Pereira Zeferino (20171060140180)** do Curso **Bacharelado em Ciência da Computação (Câmpus Anápolis)**. Os trabalhos foram iniciados às **11:00** do dia **10/09/2025** pelo Professor presidente da banca examinadora, constituída pelos seguintes membros:

- **Alexandre Bellezi Jose** (Presidente)
- **Claudia Helena dos Santos Araujo** (Examinadora Interna)
- **Alessandro Rodrigues e Silva** (Examinador Interno)

A banca examinadora, tendo terminado a apresentação do conteúdo do Trabalho de Conclusão de Curso, passou à arguição do candidato. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo aluno, tendo sido atribuído o seguinte resultado:

☒ Aprovado

☐ Reprovado

Nota : 9,6

Observação / Apreciações:

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu **Alexandre Bellezi Jose** lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Documento assinado eletronicamente por:

- **Alexandre Bellezi Jose** (289.101.538-06), em **06/10/2025 17:06:01** com chave **e1edf27ca2ef11f09c5f005056a537a4**.
- **Alessandro Rodrigues e Silva** (906.121.161-15), em **06/10/2025 18:41:12** com chave **2d6bf6baa2fd11f0a9b4005056a537a4**.
- **Claudia Helena dos Santos Araujo** (881.998.511-04), em **06/10/2025 17:06:40** com chave **f8a8a94ea2ef11f0a9b4005056a537a4**.

Este documento foi emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifg.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Ata de Projeto Final

Data da Emissão: 13/10/2025

Código de Autenticação: 2a761e



TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: João victor Pereira Zeferino

Matrícula: 20171060140180

Título do Trabalho: COLHEITA DO SABER UM JOGO QUE ENSINA AS CRIANÇAS, SEM ESQUECER DA DIVERSÃO E RELAXAMENTO

Autorização - Marque uma das opções

1. (x) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. () Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
3. () Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- () O documento está sujeito a registro de patente.
() O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
() Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Anápolis- GO, 12/11/2025.

Local

Data

A handwritten signature in black ink, appearing to read "Léo Zepherino", written over a horizontal line.

Assinatura do Autor e/ou Detentor dos Direitos Autorais

RESUMO

No Brasil, o analfabetismo entre crianças e jovens ainda representa um desafio significativo, sendo que aproximadamente 29% da população apresenta analfabetismo funcional, conforme dados do UNICEF (2024). O presente trabalho descreve a concepção, planejamento e desenvolvimento do jogo mobile Colheita do Saber, voltado para o incentivo à aprendizagem de crianças e adolescentes em conteúdos de Matemática, Língua Portuguesa e Língua Inglesa. O desenvolvimento do jogo seguiu a metodologia ágil Personal Scrum e foi implementado na plataforma Unity, fundamentando-se em teorias de jogos sérios (serious games), gamificação e abordagens pedagógicas voltadas à alfabetização e ao aprendizado básico. O projeto estruturou o jogo por meio de minigames que promovem a fixação de conceitos de forma lúdica, interativa e motivadora, utilizando elementos de diversão para despertar o interesse dos estudantes. Embora o jogo não substitua a sala de aula, ele se apresenta como um recurso educacional complementar que pode ser capaz de apoiar pais e professores na educação. Além disso, o projeto aponta possibilidades de expansões futuras, incluindo a implementação de aprendizado adaptativo, novos conteúdos e melhorias na experiência do usuário.

Palavras Chave: Jogos sérios, Gamificação, Educação básica, Recurso Educacional, Mobile, 3D, Unity

ABSTRACT

In Brazil, illiteracy among children and young people still represents a significant challenge, with approximately 29% of the population exhibiting functional illiteracy, according to UNICEF (2024). This study describes the conception, planning, and development of the mobile game Colheita do Saber, designed to promote learning among children and adolescents in Mathematics, Portuguese, and English. The game development followed the agile Personal Scrum methodology and was implemented on the Unity platform, based on theories of serious games, gamification, and pedagogical approaches focused on literacy and basic learning. The project structured the game through minigames that encourage the reinforcement of concepts in a playful, interactive, and motivating way, using elements of fun to engage students. Although the game does not replace the classroom, it serves as a complementary educational resource capable of supporting parents and teachers in the learning process. Furthermore, the project highlights potential future expansions, including the implementation of adaptive learning, additional content, and improvements in the user experience.

Keywords: *Serious games, Gamification, Basic education, Educational Resource, Mobile, 3D, Unity*

LISTA DE FIGURAS

Figura 1 - Interface do jogo Manda Letra Remake.....	24
Figura 2 - Interface do jogo Vocabicar.....	25
Figura 3 - Interface do jogo Subtraction - Mini Maths Golf.....	26
Figura 4 - Interface do jogo Sum Stacks.....	27
Figura 5 - Interface do jogo Sushi Monster.....	28
Figura 6 - Visão aérea do mapa principal.....	41
Figura 7 - Visão aérea dos portais.....	42
Figura 8 - Visão do minigame caverna das palavras.....	42
Figura 9 - Visão aérea do mapa do interpreta em ação.....	44
Figura 10 - Visão da interface do jogador.....	45
Figura 11 - Visão da interface do pergaminho com texto.....	46
Figura 12 - Visão aérea do mapa da ponte aritmética.....	47
Figura 13 - Visão da interface da ponte aritmética.....	45
Figura 14 - Visão da interface da ponte aritmética.....	48
Figura 15 - Visão da interface mobile	51
Figura 16 - Estrutura do Button Action..	53
Figura 17 - Estrutura do Button Action..	53
Figura 18 - Mundo criado pelo TERRAIN	56
Figura 19 - Oceano criado com AQUALite	57
Figura 20 - Integração do KinematicCharacterMotor	59
Figura 21 - Estrutura do LandManager	61

Figura 22 - Estrutura do PlayerStaminaManager	63
Figura 23 - Estrutura do futemática	51
Figura 24 - Estrutura do caverna das palavras	51
Figura 25 - Estrutura do interpretAção	51
Figura 26 - Interação entre EnterChallengesManager e SmoothRoundRobinWeighted	51
Figura 27 - Sistema de gerenciamento de salvamento/carregamento de jogo	51

1 INTRODUÇÃO.....	6
2 MOTIVAÇÃO E JUSTIFICATIVA.....	6
2.2 OBJETIVOS.....	9
2.3 OBJETIVOS ESPECÍFICOS.....	9
2.4 ESCOPO.....	9
2.5 ORGANIZAÇÃO DO DOCUMENTO.....	10
3 FUNDAMENTAÇÃO TEÓRICA.....	10
3.1 DESENVOLVIMENTO DE VIDEO GAMES.....	10
3.1.1 Metodologia.....	11
3.1.2 Ferramentas de desenvolvimento.....	12
3.2 GAMIFICAÇÃO.....	15
3.3 LÍNGUA PORTUGUESA E O ENSINO DE LÍNGUAS.....	16
3.3.1 Construção de vocabulário.....	17
3.3.2 Interpretação de texto.....	18
3.4 MATEMÁTICA.....	19
3.4.1 Aritmética.....	20
3.4.2 Sentido numérico.....	21
3.5 JOGOS SÉRIOS COMO RECURSOS EDUCACIONAIS.....	22
3.5.1 Ensino de línguas por meio de jogos sérios.....	22
3.5.1.1 Exemplos de jogos sérios para aprendizado de línguas.....	23
3.5.1.1.1 Manda Letra Remake.....	24
3.5.1.1.2 Vocabicar.....	24
3.5.2 Ensino da Matemática por meio de jogos sérios.....	25
3.5.2.1 Exemplos de jogos sérios para aprendizado de línguas.....	26
3.5.2.1.1 Subtraction - Mini Maths Golf.....	26
3.5.2.1.2 Sum Stacks.....	27
3.5.2.1.3 Sushi Monster.....	28
3.6 TEMPO DE TELA.....	29
3.6.1 Impactos Cognitivos e Educacionais do Uso Excessivo de Telas.....	29
3.6.2 Estratégias para o Controle do Tempo de Tela.....	30
4 PLANEJAMENTO DO JOGO PROPOSTO.....	31
4.2 DESIGN E MODELAGEM.....	31
4.2.1 Requisitos.....	31
4.2.2 Requisitos funcionais.....	32
4.2.3 Requisitos não funcionais.....	33
4.3 PRODUCT BACKLOG.....	33
4.3.1 Navegação pelo mapa.....	34
4.3.2 Sistema de Fazendinha.....	35

4.3.3 Sistema de Stamina.....	36
4.3.4 Jogar mini game caverna das palavras.....	37
4.3.5 Jogar mini game interpreta em ação.....	37
4.3.6 Jogar mini game ponte aritmética.....	38
4.3.7 Jogar mini game futemática.....	39
4.4 PROTÓTIPOS.....	40
4.4.1 Ilha principal.....	41
4.4.2 Caverna das palavras.....	42
4.4.3 Interpreta em ação.....	43
4.4.4 Ponte aritmética.....	47
4.4.5 Futemática.....	49
5 DESENVOLVIMENTO DO JOGO COLHEITA DO SABER.....	51
5.2 INTERFACE DE USUÁRIO.....	51
5.2.1 INTERFACE NOS MINIGAMES.....	54
5.3 ESTÉTICA E GRÁFICOS.....	55
5.4 MECÂNICAS.....	58
5.4.1 MOVIMENTAÇÃO DO PERSONAGEM.....	59
5.4.2 PLANTAÇÃO E COLHEITA.....	60
5.4.3 ESTAMINA.....	63
5.4.4 RELÓGIO INTERNO.....	64
5.4.5 GERADOR ALEATÓRIO DE EXPRESSÕES MATEMÁTICAS.....	66
5.4.6 MINIGAME FUTEMATEMÁTICA.....	66
5.4.7 MINIGAME PONTE DO CÁLCULO.....	68
5.4.8 MINIGAME CAVERNA DAS PALAVRAS.....	70
5.4.9 MINIGAME INTERPRETAÇÃO.....	72
5.4.10 ENTRADA DOS DESAFIOS E APRENDIZADO ESPAÇADO.....	73
5.5 PERSISTÊNCIA DE DADOS.....	75
5.6 EFEITOS SONOROS.....	77
5.6.1 FONTE DE DADOS.....	78
5.6.2 IMPLEMENTAÇÃO.....	78
6 CONSIDERAÇÕES FINAIS.....	79
7 BIBLIOGRAFIA.....	81
8 ANEXO A - LINKS.....	90

1 INTRODUÇÃO

Os jogos digitais são sistemas de software desenvolvidos para proporcionar lazer e entretenimento, além de desempenharem funções importantes no âmbito da saúde e da educação (GRAMMENOS; SAVIDIS; STEPHANIDIS, 2009). Na sociedade contemporânea, imersa em um contexto digital, os dispositivos móveis evoluíram significativamente, deixando de ser simples ferramentas de comunicação para se tornarem microcomputadores conectados, cada vez mais presentes no cotidiano de crianças e adolescentes. No entanto, enquanto a tecnologia avança de maneira acelerada, o progresso no campo educacional tem sido, proporcionalmente, mais discreto. Nesse contexto, reduzir essa disparidade – mesmo que de forma gradual – deve ser uma das prioridades da educação (FADEL et al., 2014).

2 MOTIVAÇÃO E JUSTIFICATIVA

O analfabetismo entre crianças e jovens no Brasil configura-se como um desafio estrutural e multifacetado, que persiste apesar dos avanços nas políticas educacionais. Dados recentes apontam que aproximadamente 40% dos estudantes do 4º ano do Ensino Fundamental não dominam habilidades básicas de leitura, demonstrando dificuldades em localizar e compreender informações explícitas em textos (ALMEIDA, 2023).

A pesquisa *Alfabetiza Brasil*, realizada pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) em 2023, indica que, em 2021, apenas quatro em cada dez crianças do 2º ano estavam alfabetizadas — uma queda significativa em relação a 2019, quando o índice era de seis em cada dez. Embora os dados do SAEB 2023 revelem uma melhora (49,3%), o índice ainda permanece abaixo dos níveis pré-pandêmicos (Agência Brasil, 2025). Esse quadro é agravado por fatores como a formação insuficiente de professores, a escassez de materiais pedagógicos adequados e a evasão escolar. Estudos realizados pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e as metas estabelecidas pelo Plano Nacional de Educação (PNE) indicam que a ampliação da formação dos docentes é condição indispensável para a melhoria dos resultados escolares (BRASIL, 2014; INEP, 2022). Apesar de haver avanços

em alguns indicadores, a persistência de baixos índices de titulação entre os professores constitui um entrave significativo para a qualidade do ensino básico no país.

Paralelamente, o analfabetismo funcional atinge 29% da população brasileira, segundo o Indicador de Alfabetismo Funcional (UNICEF, 2024). Dentre esse grupo, 67% apresentam dificuldades significativas em atividades digitais cotidianas, como realizar compras online, efetuar inscrições ou interpretar formulários eletrônicos (UNICEF, 2024), evidenciando limitações que extrapolam o ambiente escolar e impactam diretamente a inclusão social e digital.

E pensando nessas limitações pedagógicas já em 1958, Skinner (1958) dizia que com o crescimento populacional e o consequente aumento da demanda por educação evidenciaram a limitação do modelo tradicional baseado exclusivamente em professores e salas de aula. Décadas antes, em 1926, Sidney Pressey já havia desenvolvido a primeira "teaching machine", um dispositivo mecânico que apresentava questões de múltipla escolha e só prosseguia após a resposta correta do aluno, demonstrando que o fornecimento imediato de feedback favorecia a aprendizagem (SKINNER, 1958). Inspirado por esse trabalho, Skinner elaborou um modelo diferenciado de máquina de ensino, fundamentado nos princípios do reforço positivo. Sua proposta consistia em apresentar o conteúdo em pequenas etapas, promovendo a participação ativa do aluno e oferecendo reforço imediato sem punições, com adaptação ao ritmo individual. Os elementos centrais do dispositivo incluíam: sequência gradual e controlada de unidades de ensino; exigência de resposta ativa (em vez de simples reconhecimento); reforço imediato para respostas corretas; auxílio gradualmente retirado (fading); controle de estímulos e adaptação ao desempenho individual do aprendiz (SKINNER, 1958).

Em 1998, o professor José Armando Valente, aqui no Brasil, antecipava discussões que hoje são cada vez mais relevantes: ao assumir o papel de ensinar, o computador atuava como uma extensão dos métodos tradicionais de instrução programada, substituindo o papel ou o livro por softwares educativos — como tutoriais, exercícios do tipo *drill-and-practice*, jogos e simulações. Essa abordagem seguia uma lógica de instrução direta, mas, no caso dos jogos e simulações, adotava uma pedagogia baseada na exploração autodirigida, onde o aluno constrói seu conhecimento ativamente. No entanto, o avanço da tecnologia e a aceleração da informação vêm provocando uma mudança de paradigma: o computador deixa de ser apenas uma máquina de ensinar e passa a ser compreendido como uma mídia educacional dinâmica, capaz de

complementar, enriquecer e transformar o processo de aprendizagem. Nesse novo cenário, o papel da escola e do professor também se reinventa — não mais o de transmitir conteúdos prontos, mas o de criar ambientes que favoreçam a autonomia do estudante, sua capacidade de investigar, selecionar informações, resolver problemas e aprender de forma crítica e independente.

Valente (1998) já ilustrava isso com o exemplo do jogo “Rocky’s Boots”, que ensina lógica e circuitos eletrônicos de maneira interativa e significativa. Embora os jogos ofereçam uma abordagem envolvente, ele alertava para os riscos da competição sobrepor-se à aprendizagem, desviando o foco do conteúdo conceitual. Muitos jogos ainda exploram conhecimentos superficiais e não favorecem a reflexão sobre os erros cometidos. Por isso, um uso pedagógico eficaz dos jogos exige que os estudantes não apenas joguem, mas analisem suas ações, reflitam sobre suas falhas e, a partir disso, construam conhecimento. Essas idéias permanecem mais atuais do que nunca, reforçando a importância de uma educação centrada no aprendiz e não apenas na tecnologia.

Diante disso, a utilização de tecnologias como recurso pedagógico surge como uma alternativa para potencializar o aprendizado e transformar a experiência educacional não vem de hoje. A incorporação de tecnologias pode contribuir para melhorar a compreensão dos estudantes e proporcionar novas formas de ensino que promovam o engajamento no ambiente escolar (Mattar, 2014). Para muitos alunos, o celular é a porta de entrada para o conhecimento, para interações sociais e, principalmente, para o entretenimento por meio de jogos. Assim, considerar a gamificação como uma estratégia educativa, pode oferecer uma oportunidade significativa de transformação na educação, tornando-a mais envolvente e menos monótona. Entretanto, seu uso irrestrito em ambientes escolares também levanta preocupações relevantes. Estudos recentes apontam que o excesso de exposição às telas pode comprometer tanto o desempenho acadêmico quanto a saúde mental dos estudantes, ocasionando distrações, ansiedade e dificuldades de concentração (UNESCO, 2023). Nesse sentido, a promulgação da Lei nº 15.100/2025, que proíbe a utilização de celulares nas escolas, reflete uma medida de precaução voltada à proteção do ambiente de aprendizagem e ao bem-estar dos alunos. Paralelamente, as diretrizes do Conselho Nacional de Educação (CNE/MEC) destacam a importância do uso intencional e pedagógico das tecnologias digitais, defendendo que sua integração deva ocorrer de forma planejada, contextualizada e orientada para o desenvolvimento da educação digital e midiática.

Quando bem aplicada, a gamificação tem o potencial de melhorar o engajamento e a motivação dos alunos, aspectos esses que são fundamentais para o sucesso no processo de aprendizagem, especialmente na educação infantil. Pesquisas indicam que os elementos mais eficazes em jogos educacionais são aqueles que estimulam a colaboração, a resolução de problemas e a aplicação prática de conhecimentos (Lôbo, 2024).

2.2 OBJETIVOS

O objetivo geral deste projeto é projetar e desenvolver um jogo mobile que funcione como uma recurso de incentivo à aprendizagem de crianças e adolescentes no Brasil. O jogo tem como foco auxiliar na fixação e no aprendizado de conteúdos de matemática, língua portuguesa (construção de vocabulário e interpretação de texto) e língua inglesa (construção de vocabulário e interpretação de texto), proporcionando uma experiência lúdica e interativa.

2.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho incluíram:

- a) **Projetar um jogo a nível de Game Design:** Realizar a modelagem arquitetural, definição de mecânica, minigames e desafios de acordo com as metas de aprendizado necessárias.
- b) **Sistema de Recompensas:** Um sistema de recompensas que pode ser central para os benefícios educacionais da aprendizagem baseada em jogos (Howard-Jones et al., 2016). Essa abordagem pode contribuir para o aumento da autoestima e estimular o desenvolvimento educacional das crianças.
- c) **Jogo com sincronismo com o tempo real:** O tempo para jogar será finito e obrigará o jogador a estabelecer uma rotina de jogar em intervalos durante o dia, reduzindo o tempo de tela contínuo.

2.4 ESCOPO

Este projeto se restringe ao projeto e desenvolvimento de um jogo digital mobile para ensino dos conceitos fundamentais para crianças e adolescentes. Os conceitos a serem desenvolvidos no jogo serão mais bem detalhados posteriormente, porém, de maneira geral, abrangem os conteúdos do ensino fundamental ministrados, por exemplo:

- Matemática: operações básicas de soma, subtração, multiplicação e precedência de operadores, construção de expressões matemáticas.
- Português e Inglês: Construção de vocabulário e interpretação de texto

2.5 ORGANIZAÇÃO DO DOCUMENTO

O trabalho está dividido em capítulos e estruturado da seguinte forma: Este primeiro capítulo apresenta uma introdução ao projeto, definindo a motivação, os objetivos e o escopo; No Capítulo 3 é feita uma fundamentação teórica contextualizando e explicando os conceitos fundamentais para a compreensão do projeto; No Capítulo 4 é realizado o planejamento do jogo proposto, detalhando o design, os requisitos, o Product Backlog e os protótipos das interfaces; No Capítulo 5 descreve o desenvolvimento, incluindo interfaces, mecânicas, persistência de dados e efeitos sonoros. Por fim, o Capítulo 6 apresenta as considerações finais, seguidas pelas referências (Capítulo 7) e anexos (Capítulo 8).

3 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento deste projeto é necessário conhecer alguns conceitos, como: desenvolvimento de jogos, gamificação/jogos sérios, português e matemática. Neste capítulo serão definidos e detalhados alguns desses conceitos mais importantes.

3.1 DESENVOLVIMENTO DE VIDEO GAMES

Segundo (Battaiola, 2000), um jogo de computador pode ser definido como um sistema interativo que permite ao usuário experimentar uma situação de conflito. Quase sempre, como plano de fundo dessa situação, existe um enredo, que define do que se trata o jogo, as regras que o jogador deve seguir e os objetivos que ele deve se esforçar para atingir.

Para o desenvolvimento de software, de modo geral, é necessário seguir uma série de regras bem definidas, a fim de atender as necessidades para as quais ele foi desenvolvido. O desenvolvimento de jogos foca na diversão e entretenimento de seus usuários, procurando interagir com eles da melhor forma possível, por meio da utilização de uma boa interface.

Esta seção aborda diversos aspectos do desenvolvimento de software com foco específico na criação de jogos. Dentre esses, metodologia de desenvolvimento e ferramentas comumente utilizados no processo de desenvolvimento de jogos.

3.1.1 Metodologia

No desenvolvimento de software, é essencial seguir um conjunto de práticas e processos bem definidos para atender às necessidades específicas para as quais o software foi criado. No entanto, como destacado por Costa (2016), apesar da existência de diversos métodos tradicionais para o desenvolvimento de software e sistemas gerais, o desenvolvimento de jogos exige uma abordagem diferenciada devido às suas particularidades. A principal distinção está no foco da criação de jogos, que visa proporcionar diversão e entretenimento aos usuários, buscando interagir com eles de maneira eficaz por meio de uma interface intuitiva e atraente.

O Scrum, conforme descrito por Schwaber e Sutherland (2020), é uma metodologia ágil que divide o desenvolvimento em ciclos curtos denominados *sprints*, permitindo revisões contínuas e ajustes nas prioridades conforme necessário. A flexibilidade proporcionada pelos ciclos de desenvolvimento ágeis permite uma adaptação constante ao longo do projeto, facilitando o ajuste de demandas de forma eficiente. O Scrum é orientado por equipes multidisciplinares que possuem todas as competências necessárias para realizar as tarefas, compartilhando e adquirindo habilidades de forma colaborativa, conforme a necessidade (SCHWABER; SUTHERLAND, 2020).

No contexto de desenvolvimento de jogos, o Scrum organiza o trabalho por meio da criação do *Product Backlog*, uma lista ordenada e emergente das necessidades do jogo. Essa lista inclui a identificação dos requisitos essenciais e a priorização das funcionalidades a serem desenvolvidas (SCHWABER; SUTHERLAND, 2020). As funcionalidades são definidas como histórias de usuário e o desenvolvimento ocorre por meio de *Sprints*, períodos curtos durante os quais são planejadas e executadas funcionalidades específicas. Cada *Sprint* segue uma sequência estruturada, começando com o planejamento, onde as funcionalidades a serem trabalhadas são selecionadas, seguido pela execução, na qual um protótipo jogável é desenvolvido (SCHWABER; SUTHERLAND, 2020). Ao final de cada *Sprint*, realiza-se uma revisão para avaliar o progresso, testar as funcionalidades desenvolvidas, e uma retrospectiva, que analisa o que funcionou bem e ajusta o planejamento para as próximas etapas (SCHWABER; SUTHERLAND, 2020).

Entretanto, para o desenvolvimento de jogos em ambientes independentes, que geralmente envolvem equipes pequenas (muitas vezes compostas por uma ou duas pessoas), a aplicação do modelo tradicional de Scrum pode ser excessivamente burocrática e inadequada.

Nesse contexto, como sugere Costa (2016), a adaptação do Scrum por meio do Personal Scrum se torna uma solução mais apropriada. Esse modelo ajustado visa proporcionar uma gestão mais ágil e eficiente das atividades, mesmo em projetos com equipes reduzidas.

No Personal Scrum, o desenvolvedor assume os papéis de *Scrum Master* e *Fly Half*. O Scrum Master é responsável por planejar o processo e garantir que o planejamento seja seguido corretamente, enquanto o *Fly Half* desempenha o papel da equipe de desenvolvimento, sendo responsável pela execução das tarefas estabelecidas. Já o papel de *Product Owner*, tradicionalmente atribuído a um membro da equipe, deve ser compartilhado com uma pessoa externa, como um possível jogador do público-alvo ou um professor orientador, que auxiliará na priorização dos requisitos. Embora o projeto possa ser conduzido por um único desenvolvedor, é essencial que ele compreenda e execute os papéis de forma disciplinada, além de manter uma autogestão eficiente. Costa (2016) também recomenda o uso da ferramenta Kanban para o acompanhamento e monitoramento das atividades realizadas, as reuniões diárias foram substituídas por uma reunião semanal para avaliar o andamento do sprint e ajustar os requisitos caso necessário.

A principal característica do *Personal Scrum* é a realização de *sprints*, períodos curtos (geralmente entre uma e duas semanas) dedicados ao desenvolvimento de requisitos específicos do projeto. Como afirmado por Costa (2016), "a ideia é que o desenvolvedor tenha seu objetivo bem delimitado, mantendo o foco nas atividades do *sprint*, para que, ao seu final, as funcionalidades desenvolvidas sejam testadas". Antes de cada *sprint*, ocorre uma fase de planejamento onde os requisitos a serem abordados são definidos, resultando no *Sprint Backlog*, o qual organiza as tarefas a serem realizadas. Ao término do período estipulado, realiza-se a revisão do *sprint*, testando as funcionalidades planejadas e registrando os resultados. Caso algum requisito não seja atendido, ele retorna para o *Product Backlog* para ser reavaliado e ajustado conforme necessário.

3.1.2 Ferramentas de desenvolvimento

As ferramentas de desenvolvimento desempenham um papel crucial para desenvolvedores independentes de jogos, oferecendo recursos que facilitam a criação, prototipagem e distribuição de jogos de forma eficiente e acessível.

De acordo com o artigo *The history of game engines* (ISPMANAGER.COM, 2024), antigamente o desenvolvimento de jogos era muito prático. Cada linha de código estava vinculada a uma peça específica de hardware e não havia automação, o que tornava o desenvolvimento lento e estabelecia um limite para a complexidade dos projetos. Atualmente, as *Game Engines* se tornaram a espinha dorsal do desenvolvimento de jogos. A proliferação de computadores, consoles e smartphones permitiu a criação de mundos grandes e complexos, enquanto as *engines* simplificaram o desenvolvimento automatizando os processos principais (ISPMANAGER.COM, 2024).

A grande vantagem de um motor de jogo é que ele permite que os desenvolvedores de jogos reutilizem ou adaptem o mecanismo para fazer novos jogos, eles são poupados da tarefa de reinventar a roda (BALASUBRAMANIAN, 2022), fornecendo os recursos necessários para o desenvolvimento de um jogo, como um renderizador 2D ou 3D, um mecanismo de física ou colisão e suporte para som, cinemática, scripts, animação, IA e jogo em rede (BALASUBRAMANIAN, 2022). Para desenvolvedores indies, a principal vantagem é que elas automatizam muitas das tarefas técnicas, simplificando o processo de criação de jogos e permitindo que as equipes criativas foquem em ideias e *design*, em vez de minúcias técnicas (ISPMANAGER.COM, 2024). Conforme avaliado por Christopoulou (2017), atualmente existem diversas opções de *Game Engines* no mercado, sendo a Unreal Engine e a Unity Engine as duas mais predominantes no desenvolvimento de jogos comerciais.

A Unreal Engine, desenvolvida pela Epic Games, destaca-se por sua capacidade de visualização realista, criação de vegetação e terrenos. Embora seja especialmente indicada para projetos de grande escala, não é a escolha ideal para jogos mobile de pequeno porte, apesar de oferecer suporte para iOS e Android (Šmíd, 2017). A engine inclui o sistema Blueprint, uma ferramenta de scripting visual que permite a criação de lógicas complexas por meio de blocos gráficos interconectados, em vez de scripts tradicionais. Além disso, a Unreal oferece código-fonte aberto em C++, o que concede aos desenvolvedores um controle significativo sobre o sistema, uma grande vantagem. Contudo, sua estrutura é complexa e exige um processo de aprendizagem considerável (Šmíd, 2017).

A Unreal também se destaca por seus efeitos de pós-processamento rápidos e pelo suporte a uma ampla gama de recursos, incluindo um editor para criação de materiais personalizados e ferramentas avançadas para otimização e depuração visual. Embora a comunidade da Unreal seja

menor em comparação com a do Unity e sua documentação, em alguns casos, seja insuficiente, esses aspectos não comprometem seu uso em projetos de maior complexidade (Šmíd, 2017). No entanto, para o projeto abordado neste trabalho, a Unreal foi descartada devido ao foco no *design* e à ausência de necessidade de gráficos realistas.

Por outro lado, a Unity, criada pela Unity Technologies em 2002 e com sua primeira versão lançada em 2005, foi inicialmente voltada para o desenvolvimento do OS X. Desde então, a Unity evoluiu e atualmente oferece suporte a 27 plataformas, incluindo VR. Sua principal vantagem está na facilidade de uso, tornando o desenvolvimento, especialmente para plataformas móveis, extremamente ágil. A criação de projetos e jogos de menor escala é simples, com um processo de exportação eficiente. A arquitetura do motor é de fácil compreensão, e o uso de C# para scripting proporciona uma performance rápida e eficaz. A Unity conta com uma grande comunidade, com fóruns repletos de soluções que facilitam a depuração. Além disso, oferece uma loja de *Assets* com preços acessíveis e uma vasta gama de recursos úteis (Šmíd, 2017).

Por todos esses motivos, escolheu-se utilizar a Unity para o desenvolvimento desse projeto. Além das atividades de programação, para o desenvolvimento do jogo será necessário a criação de outros artefatos, como imagens, áudios, e mesmo diagramas e texto para documentação. Para essas tarefas, será priorizado o uso de software livres e gratuitos, para o desenvolvimento de jogos e outras tarefas, uma variedade de software livres e gratuitos se destacam. **GIMP** e **Krita** emergem como ferramentas essenciais para artistas, oferecendo edição de imagens rasterizadas e pintura digital, respectivamente, sendo o **GIMP** comparável ao Photoshop em funcionalidades (GIMP, [s.d.]) e o **Krita** otimizado para criação artística digital e animação 2D (Krita Foundation, [s.d.]). **Audacity** se apresenta como um editor de áudio versátil para criação e edição de sons e músicas (Audacity Team, [s.d.]).

O **Git** vem como um sistema de controle de versão distribuído, que é amplamente utilizado para gerenciar o código-fonte de projetos de software. Ele permite que múltiplos desenvolvedores trabalhem simultaneamente em um mesmo projeto, rastreando todas as alterações feitas no código e facilitando a colaboração. (Git, [s.d.]). Adicionalmente, para diagramação, **draw.io** facilita a criação de fluxogramas online (diagrams.net, [s.d.]), e para documentação, o pacote **LibreOffice** oferece alternativas de escritório completas (LIBREOFFICE, [s.d.]) complementado pela edição colaborativa online do **Google Docs**

(Google, [s.d.]), enquanto o **Notion** auxilia na organização multifuncional de informações e tarefas por meio do quadro de kanban (Notion Labs, Inc., [s.d.]).

3.2 GAMIFICAÇÃO

O termo "gamificação" foi usado pela primeira vez em 2008 por Shirky e Terrill (Deterding, S., 2015). A origem do conceito, no entanto, remonta ao verbo "gamificar", que teria sido utilizado pela primeira vez por Richard Bartle em seu trabalho no primeiro *Multi-User Dungeon* (MUD), um dos primeiros mundos virtuais sociais e interativos gerados por computador (Deterding, S., 2015). Porém, o conceito de gamificação apresenta semelhanças com outros termos e abordagens, como *design* inspirado em jogos, jogos sérios, simulações e os próprios jogos. No entanto, as fronteiras entre esses conceitos nem sempre estão claramente definidas.

A gamificação consiste no uso de elementos, dinâmicas e mecânicas de jogos em contextos que vão além do entretenimento, com o objetivo de aumentar a motivação, o engajamento e influenciar o comportamento do usuário (Marczewski, 2013). Nessa perspectiva, Burke (2015) a definição da gamificação inclui: 1) mecânica de jogos e seus elementos chave como pontos, distintivos e placares; 2) *design* de experiência digital representada pela jornada que os jogadores terão de percorrer, como descrição do jogo, passo a passo, reconhecimento do ambiente e decodificação do roteiro; 3) métodos para engajar indivíduos virtualmente por meio de computadores, tablets, smartphones e outros dispositivos digitais; e 4) motivação de pessoas e comportamentos pró-utilização de tecnologias, estímulo à inovação e desenvolvimento de habilidades. Burke (2015) afirma que a gamificação potencializa, tanto nos jogadores quanto nas organizações, o alcance de objetivos.

Os jogos sérios, por sua vez, são desenvolvidos com propósitos que vão além do mero entretenimento. No contexto educacional, esses jogos se caracterizam como sistemas complexos que combinam jogabilidade com aprendizado, utilizando objetivos, regras e mecânicas específicas para potencializar a aquisição de conhecimento (Christopoulos et al., 2022). Já de acordo com Kapp (2012), a gamificação é "o uso de mecânicas, estética e pensamento de jogos para envolver pessoas, motivar a ação, promover o aprendizado e resolver problemas". Essa abordagem é amplamente empregada em contextos digitais, buscando solucionar desafios,

estimular o engajamento e motivar indivíduos a atingirem seus objetivos. Ao proporcionar uma experiência interativa e lúdica, a gamificação aumenta a percepção de autonomia, competência e conexão social entre os usuários. Como resultado, essa estratégia tem sido amplamente aplicada em áreas como educação, negócios, marketing e serviços, transformando métodos tradicionais de engajamento, especialmente no campo educacional (Christopoulos & Mystakidis, 2023).

No artigo "Gamification in Education", Kiryakova, Angelova e Yordanova (2014) apontam que um dos principais desafios da educação moderna é a falta de engajamento e motivação dos alunos para participarem ativamente do processo de aprendizagem. Para contornar essa questão, professores têm buscado novas abordagens que incentivem a participação ativa dos estudantes. Uma solução eficaz é a implementação de elementos de jogos no ensino, oferecendo recompensas para reconhecer esforços e conquistas, o que contribui para o aumento da motivação (Kiryakova et al., 2014).

A motivação é um dos aspectos afetivos mais relevantes da aprendizagem, pois influencia diretamente os processos cognitivos. Pintrich, Brown e Weinstein (1994) a descrevem como o motor da aprendizagem. Quando os alunos encontram significado e prazer em experiências educacionais online, tendem a se envolver mais, tomar iniciativas e estabelecer metas para alcançar seus objetivos. O nível de habilidade de cada jogador define as ações que ele pode executar, e os desafios lúdicos devem ser progressivamente ajustados para manter um equilíbrio entre dificuldade e competência, acionando o estado de "fluxo". Nesse contexto, gatilhos e estímulos são elementos de gamificação que ajudam a estruturar ciclos de engajamento (Pintrich, Brown & Weinstein, 1994).

Na educação, a motivação é reconhecida pelo modelo de atenção-relevância-confiança-satisfação (ARCS) (Keller, 2010). Esse modelo visa aumentar e sustentar o nível de motivação dos alunos, primeiramente ganhando sua atenção e estabelecendo como o aprendizado é pessoalmente relevante e significativo para eles. Em seguida, o aprendizado deve ser organizado de modo a promover a confiança dos alunos para concluir as tarefas e garantir que sua satisfação geral permaneça alta (Christopoulos & Mystakidis, 2023).

3.3 LÍNGUA PORTUGUESA E O ENSINO DE LÍNGUAS.

Nesta seção, abordaremos o ensino de línguas com foco na língua portuguesa. No entanto, considerando que se trata de um tema abrangente, concentramos nossa discussão nos aspectos relevantes ao projeto, que incluem a construção de vocabulário e a interpretação de texto.

3.3.1 Construção de vocabulário

O aprendizado de uma nova língua exige habilidades específicas e um conjunto de competências que influenciam de maneira geral, como, por exemplo, o processamento visual, a atenção, a memória de trabalho, o conhecimento de vocabulário e as consciências linguísticas – fonológica, sintática, morfológica, semântica, textual, entre outras (MORAIS et al., 2013). Essas competências são essenciais no processo de aprendizagem da leitura e da escrita, interconectando-se e manifestando-se de forma mais ou menos intensa durante o processamento dessas habilidades. Quando há déficits ou dificuldades em alguma dessas habilidades essenciais, isso pode impactar negativamente o processo de alfabetização das crianças (MORAIS et al., 2013).

O vocabulário tem um papel fundamental na transmissão de significados, sendo esse o principal objetivo da comunicação verbal (BIDERMAN, 2001). A informação transmitida por meio de uma mensagem ocorre, principalmente, por meio das palavras, que compõem os enunciados. Além disso, é por meio dos signos linguísticos, ou palavras, que fazemos referência à realidade fora da linguagem, representando elementos desse mundo conforme a forma como são abordados pela língua e pela cultura (BIDERMAN, 2001). Assim, o léxico é o espaço onde se armazenam os significados e os conteúdos da linguagem humana (BIDERMAN, 2001). Para Oliveira e Isquardo (2001), o léxico representa o saber compartilhado presente na consciência dos falantes de uma língua, constituindo-se como "o acervo do saber vocabular de um grupo sociolinguístico-cultural".

Conforme Tonnessen (2009), o vocabulário é considerado um alicerce fundamental para o desenvolvimento das habilidades de alfabetização na escola, sendo amplamente reconhecido na literatura como essencial o ensino e reforço dessa habilidade para as crianças. Betty Hart e Todd R. Risley estimaram que, ao chegarem aos quatro anos de idade, crianças cujas famílias têm condições socioeconômicas mais privilegiadas terão escutado 30 milhões de palavras a mais do

que crianças da mesma idade, de famílias mais pobres (Hart, 1995), o que as coloca em risco de enfrentar dificuldades na aquisição das competências de leitura e escrita (Greenwood et al., 2017). Assim como para Laufer (1986), não há fluência sem uma base sólida de vocabulário; nenhum insumo será abrangente e relevante se o léxico do aprendiz for pobre. Portanto, um vocabulário limitado dificulta o desenvolvimento dessas competências, o que pode repercutir em dificuldades em várias áreas de aprendizagem, uma vez que as crianças terão dificuldade não apenas em compreender o que ouvem, mas também em entender o que leem (Jalongo & Sobolak, 2011).

3.3.2 Interpretação de texto

A leitura é um processo que cada pessoa realiza por si mesma e permite examinar o conteúdo do que está sendo lido, analisando cada parte do material de leitura, destacando o essencial e comparando o conhecimento já existente do leitor com o que acabou de ser adquirido (Perrusquia et al., 2010). Essencialmente, adquirir a habilidade de ler e entender bem é um requisito básico para as demandas sociais e econômicas na sociedade atual (COSTER et al., 2011).

A compreensão de leitura não é apenas a simples decodificação dos significados de um texto, mas a construção de significados que resultam da interação do leitor com o texto (GRAFFIGNA et al., 2008). De acordo com Kate Cain (2009), a compreensão de leitura envolve um conjunto de habilidades cognitivas e linguísticas essenciais para a construção de significados a partir de um texto. Essas habilidades podem ser descritas da seguinte maneira:

1. ***Decodificação das palavras***: a capacidade de identificar palavras de forma precisa e fluente, funcionando como a base para a compreensão mais profunda do conteúdo. Sem a fluência adequada na leitura, o processo de compreender o texto fica comprometido.
2. ***Integração de informações***: permite ao leitor conectar diferentes partes do texto, como frases e parágrafos, a fim de formar uma compreensão coesa. Habilidade essa que envolve tanto a combinação de informações explícitas quanto implícitas, sendo necessário que o leitor associe e relacione ideias ao longo do texto.

3. **Inferência**: habilidade central na compreensão de leitura, pois envolve a capacidade de deduzir informações que não são diretamente apresentadas no texto, mas são fundamentais para um entendimento mais completo do conteúdo.
4. **Monitoramento da compreensão**: onde permite ao leitor perceber quando algo não está claro ou quando a compreensão está falha. Este processo envolve a capacidade de identificar lacunas na compreensão e, se necessário, voltar a reler ou buscar informações adicionais dentro do próprio texto.
5. **Compreensão da estrutura textual**: o conhecimento das diferentes formas de organização de textos facilita a previsão e a organização mental das informações apresentadas, proporcionando uma leitura mais eficiente e coesa.
6. **Memória de trabalho**: permite que o leitor mantenha informações temporariamente enquanto realiza conexões e constrói um entendimento global do texto;
7. **Memória de curto prazo**: necessária para recordar detalhes específicos enquanto a leitura avança.

Essas habilidades funcionam de forma integrada, sendo interdependentes para uma compreensão plena do material lido. A falta de desenvolvimento em uma ou mais dessas áreas pode comprometer a capacidade do leitor de construir um entendimento coeso do texto, mesmo que ele consiga identificar as palavras com precisão (Cain, 2009).

3.4 MATEMÁTICA

Aprender matemática é adquirir ferramentas cognitivas para atuar sobre a realidade. Para Piaget, existe o caráter de continuidade entre as estruturas lógico-matemáticas espontâneas do pensamento infantil e os edifícios formais construídos pelos matemáticos (Piaget, 2002). Ele identifica a matemática como uma espécie de interface entre o espírito humano e o mundo, sendo um instrumento-chave no intercâmbio entre sujeito e universo. A Matemática, assim como o aprendizado de línguas, é um campo extenso que requer um estudo aprofundado para uma

compreensão abrangente do tema. Nesta seção, abordaremos a Matemática no contexto da educação infantil, com foco na aritmética e sentido numérico, mais especificamente na construção e resolução de expressões aritméticas por meio de operações fundamentais, como adição, subtração e multiplicação.

3.4.1 Aritmética

Segundo Brueckner (s.d., p. 11, apud Valente & Ferreira, 2021, p. 11), a aritmética é composta por quatro funções essenciais: a função do cálculo, função informativa, função sociológica e função psicológica.

A função do cálculo refere-se à aprendizagem e à prática das operações e processos de cálculo. Essa função parece constituir o principal objetivo da aritmética e implica a aquisição das seguintes capacidades: a) Compreensão do significado dos números e da linguagem matemática; b) Capacidade e hábito de realizar operações de soma, subtração, multiplicação e divisão envolvendo números inteiros, mistos, frações e decimais; c) Execução dessas operações com clareza, exatidão e rapidez. (BRUECKNER, s.d., p.12, apud VALENTE; FERREIRA, 2021, p. 12).

A função informativa abrange diversos aspectos da aritmética, de modo que essa disciplina adquire uma riqueza de significado e conteúdo que não pode ser alcançada quando se dedica exclusivamente ao cálculo mecânico. Dessa forma, o ensino da aritmética não deve se limitar à aprendizagem das operações e processos de cálculo, mas também deve proporcionar a compreensão do significado social dos fatos e relações quantitativas do meio. Para isso, a aritmética deve: a) Ser aplicada a diversas situações cotidianas que envolvem aspectos quantitativos; b) Utilizar oportunidades interdisciplinares para enriquecer e ampliar a compreensão e utilidade dos números; c) Explicar o papel essencial que o sistema numérico desempenhou no progresso social, econômico e industrial; d) Solucionar problemas matemáticos não apenas para a prática do cálculo, mas também considerando sua relevância social (BRUECKNER, s.d., p.13, apud VALENTE; FERREIRA, 2021, p. 13).

A função sociológica consiste na contribuição da aritmética para a compreensão das atividades econômicas, como produção e consumo (BRUECKNER, s.d., p.12, apud VALENTE;

FERREIRA, 2021, p. 12). Além disso, conforme Valente e Ferreira (2021), a aritmética proporciona algo que nenhuma outra disciplina escolar oferece: um amplo potencial de generalização. Ela permite a organização da experiência por meio dos números, viabilizando a verificação do pensamento quantitativo, o que, de acordo com a autora, nenhuma outra área da matemática possui maior impacto na obtenção desse objetivo. Adicionalmente, a aritmética não apenas fornece meios de generalização por meio da apreensão dos números, mas também possibilita a construção de abstrações, relacionando diferentes conjuntos numéricos. Dessa forma, os alunos desenvolvem a capacidade de perceber, discernir e compreender os aspectos quantitativos das situações cotidianas. Isso contribui para a adoção de uma atitude matemática adequada sempre que necessário (VALENTE; FERREIRA, 2021).

3.4.2 Sentido numérico

De acordo com Nunes e Bryant (1997), a numeralização envolve atribuir um sentido numérico, expressando uma familiaridade com os números, seus usos cotidianos e a capacidade de pensar matematicamente em diversas situações, além do ambiente escolar. Nesse contexto, o sentido numérico refere-se à intuição sobre números, suas aplicações e interpretações, caracterizando-se pela intenção de construir significado para situações numéricas (Sowder, 1995). Spinillo (2006) destaca que o ensino do sentido numérico não pode ser feito de maneira direta, como ocorre com os conceitos matemáticos tradicionais, pois ele representa uma forma de pensar matematicamente, a ser desenvolvido a partir de cada conceito ou tópico do currículo escolar.

Sowder (1995) identifica algumas habilidades essenciais para a construção do sentido numérico: (a) utilizar âncoras, fazer estimativas e realizar cálculos mentais; (b) usar diferentes representações e instrumentos, reconhecendo qual deles é mais adequado para cada situação; (c) perceber a magnitude relativa dos números; (d) compreender os efeitos das operações sobre os números; e (e) atribuir significado aos números, reconhecendo seus usos e funções no cotidiano. Com base nesses aspectos, este projeto busca oferecer às crianças, por meio dos desafios propostos, a oportunidade de exercitar a compreensão dos efeitos das operações sobre os números, com foco no entendimento do número como operador em situações de resolução de problemas envolvendo adição, subtração e multiplicação.

3.5 JOGOS SÉRIOS COMO RECURSOS EDUCACIONAIS

Atualmente, o termo “*serious games*” (jogos sérios) está se tornando cada vez mais popular (SUSI; JOHANNESSON; BACKLUND, 2007). O termo em si já está estabelecido, mas ainda não há uma definição única e consensual do conceito (SUSI; JOHANNESSON; BACKLUND, 2007). Jogos sérios geralmente se referem a jogos utilizados para treinamento, publicidade, simulação ou educação, sendo projetados para rodar em computadores pessoais ou consoles de videogame (SUSI; JOHANNESSON; BACKLUND, 2007). Segundo Corti (2006, p.1), o aprendizado baseado em jogos/jogos sérios “consiste em aproveitar o poder dos jogos de computador para cativar e envolver os usuários finais com um propósito específico, como o desenvolvimento de novos conhecimentos e habilidades”.

Levando em consideração as crianças de hoje, os jogos sérios são ferramentas ideais para atender às suas necessidades. Pesquisas extensas demonstram os efeitos positivos dos jogos digitais na aprendizagem de crianças e adolescentes (Blumberg & Ismailer, 2009). A estrutura dos jogos promove habilidades essenciais para a aprendizagem profunda, como metacognição, atenção seletiva, resolução de problemas, pensamento alternativo, prática contínua e letramento informacional (Blumberg & Ismailer, 2009; Charsky, 2010).

Jogos bem desenvolvidos incluem desafios, fantasia, curiosidade e controle, fatores que, segundo Malone & Lepper (1987), são fundamentais para a motivação intrínseca. Devem incluir tutoriais guiados e ambientes seguros (*sandbox*), facilitando o aprendizado progressivo (Gee, 2004). Gee (2008) também aponta que os jogos sérios insere o conteúdo de aprendizagem em um contexto envolvente, permitindo que os jogadores desenvolvam habilidades cognitivas sem perceber e sem a necessidade de atenção explícita.

3.5.1 Ensino de línguas por meio de jogos sérios.

Alice J. Robinson (2008) argumenta que os videogames podem proporcionar uma estrutura valiosa para refletir sobre o ensino e a prática da escrita. A autora destaca que tanto a teoria quanto a prática da experiência lúdica criam um ambiente projetado, com forte analogia ao *design* de currículos de composição e programas de cursos, o que possui implicações significativas para o aprendizado e a alfabetização. Os jogos de linguagem, segundo Robinson,

podem ser, de fato, estratégias eficazes para estimular a reflexão fonológica, abordando diversas unidades sonoras, como rimas, sílabas, unidades intrassilábicas e fonemas. Esses jogos também favorecem o reconhecimento de palavras, integrando os níveis lexicais e sublexicais da leitura, além de permitir a reflexão sobre as propriedades e a lógica do sistema alfabético e ortográfico. Tais aspectos, frequentemente negligenciados ou tratados de forma superficial ou mecânica, são fundamentais para o desenvolvimento das competências linguísticas (Morais, 2012; Leal, Albuquerque e Leite, 2005).

Em sua pesquisa, Castro e Sevillano (2022), descrevem que o uso da aprendizagem baseada em jogos digitais resulta em uma melhoria significativa na compreensão de leitura, conforme mensurado pelo programa utilizado. A pesquisa também destaca a eficácia do *design* de jogos voltados para objetivos educacionais, em conformidade com a legislação vigente e os modelos teóricos de jogos sérios.

Jogos destinados ao ensino da interpretação de textos devem incorporar textos escritos ou narrativas orais de forma atraente e lúdica (Castro & Sevillano, 2022). O nível de dificuldade deve ser ajustado conforme o desempenho do aluno, levando em consideração sucessos, erros, dificuldades de atenção e tempo de resposta, criando um percurso de aprendizagem que fortaleça as áreas de maior dificuldade e consolide os pontos em que o aluno apresenta melhor desempenho. Nesse contexto, os alunos devem ser desafiados a extrair as ideias principais dos textos, além de testar a memória de trabalho ao recordar imagens ou textos sobre os quais terão que resolver perguntas ou enigmas. Além disso, é importante incluir exercícios que envolvam a ordenação das ideias ou a interpretação do significado apresentado (Castro & Sevillano, 2022). Para incentivar os alunos, deve-se implementar um sistema de recompensas, como o desenho de um avatar ou o cuidado de um pet virtual, e uma narrativa que possibilite ao aluno avançar conforme atinge os objetivos educacionais (Castro & Sevillano, 2022).

3.5.1.1 Exemplos de jogos sérios para aprendizado de línguas

Nessa seção, apresenta-se exemplos de jogos educativos que combinam conceitos linguísticos e que foram utilizados como inspiração para a elaboração dos minigames deste trabalho.

3.5.1.1.1 Manda Letra Remake

Manda Letra Remake é um jogo educativo voltado para o aprendizado e aprimoramento do vocabulário e da ortografia da língua portuguesa. No jogo, os jogadores precisam formar palavras corretamente a partir das letras disponíveis. O objetivo é utilizar as letras fornecidas de maneira estratégica para formar o maior número de palavras possível, respeitando as regras gramaticais e ortográficas da língua portuguesa (PORTAL LUDO, 2025).

Ao longo do jogo, os participantes são desafiados a pensar rapidamente e a aplicar seus conhecimentos linguísticos para formar palavras e vencer o tempo, promovendo o desenvolvimento das habilidades de leitura e escrita de forma divertida e interativa. O jogo também visa reforçar a compreensão do significado e da estrutura das palavras, estimulando o raciocínio lógico e o conhecimento ortográfico (PORTAL LUDO, 2025).

Figura 1 – Interface do jogo Manda Letra Remake.



Fonte: Portal Ludo (2025).

3.5.1.1.2 Vocabicar

Vocabicar é um jogo educativo interativo focado no desenvolvimento de vocabulário e habilidades linguísticas. O objetivo do jogo é ajudar os jogadores pilotando um carrinho com o objetivo de expandir seu vocabulário por meio de atividades de reconhecimento de palavras e

associação de imagens com seus significados. O jogo utiliza um método visual para reforçar o aprendizado de novos termos, tornando-o uma ferramenta eficaz tanto para iniciantes quanto para aprendizes avançados de idiomas (Vocabicar, 2025).

Os jogadores devem combinar palavras e imagens corretamente, sendo desafiados por diferentes níveis de dificuldade. Cada acerto aumenta a pontuação, enquanto a dificuldade cresce à medida que se avança no jogo (Vocabicar ,2025).

Figura 2 – Interface do jogo Vocabicar.



Fonte: Vocabicar (2025).

3.5.2 Ensino da Matemática por meio de jogos sérios.

Segundo Oldfield (1991) para o ensino da Matemática, alguns jogos são mais adequados do que outros, pois o jogo matemático é uma atividade, individual ou coletiva, que envolve desafio contra uma tarefa ou adversários, pautada por um conjunto de regras e objetivos bem definidos, com começo, meio e fim, apresentando objetivos cognitivos matemáticos. Grandó (2000) ressalta sete momentos do jogo considerados relevantes durante a intervenção que favorecem na prática pedagógica com o jogo digital: a familiarização com o

material, o reconhecimento das regras, o jogar para garantir regras, o registro do jogo, a intervenção escrita e o jogar com competência.

A importância dos jogos como recurso educacional da matemática é ainda mais evidente, pois o jogo instiga o aluno, disposto a vencer, a elaborar uma boa estratégia. Dessa maneira, os jogos sérios estimulam o estudante a pensar, analisar, observar, verificar e presumir, de modo a alcançar seu objetivo. Todo esse processo está diretamente relacionado ao raciocínio lógico, princípio fundamental da Matemática (Poeta & Geller, 2014).

3.5.2.1 Exemplos de jogos sérios para aprendizado de línguas

Nessa seção, apresenta-se exemplos de jogos educativos que combinam conceitos matemáticos e que serviram de inspiração para elaborar os minigames deste trabalho.

3.5.2.1.1 Subtraction - Mini Maths Golf

Subtraction - Mini Maths Golf é um jogo educativo de matemática que combina conceitos matemáticos com mecânicas de minigolfe, proporcionando uma experiência interativa e envolvente. O objetivo do jogador é calcular corretamente operações de subtração para conseguir energia para direcionar a bola até o buraco com o menor número de jogadas possível (Fractions of numbers, 2025).

Com desafios progressivos, o jogo estimula o raciocínio lógico e a agilidade mental, tornando o aprendizado da subtração dinâmico e divertido (Fractions of numbers, 2025).

Figura 3 – Interface do jogo Subtraction - Mini Maths Golf.



Fonte: Fractions of numbers (2025).

3.5.2.1.2 Sum Stacks

Sum Stacks é um jogo educativo que desafia os jogadores a combinar números para atingir valores-alvo pré-determinados. A mecânica do jogo envolve a manipulação de blocos numéricos, onde o primeiro bloco representa o número inicial e o segundo bloco contém uma operação matemática e um segundo número. O objetivo do jogador é aplicar as operações corretamente para alcançar os resultados desejados (Sum stacks, 2025).

O jogo promove o desenvolvimento de habilidades essenciais, tais como: Resolução de Problemas: Os jogadores devem analisar as operações disponíveis e encontrar soluções estratégicas. A estrutura lógica do jogo exige que os jogadores utilizem conceitos de decomposição e reconhecimento de padrões (Sum stacks, 2025).

Figura 4 – Interface do jogo Sum Stacks.



Fonte: Sum stacks | Math playground (2025).

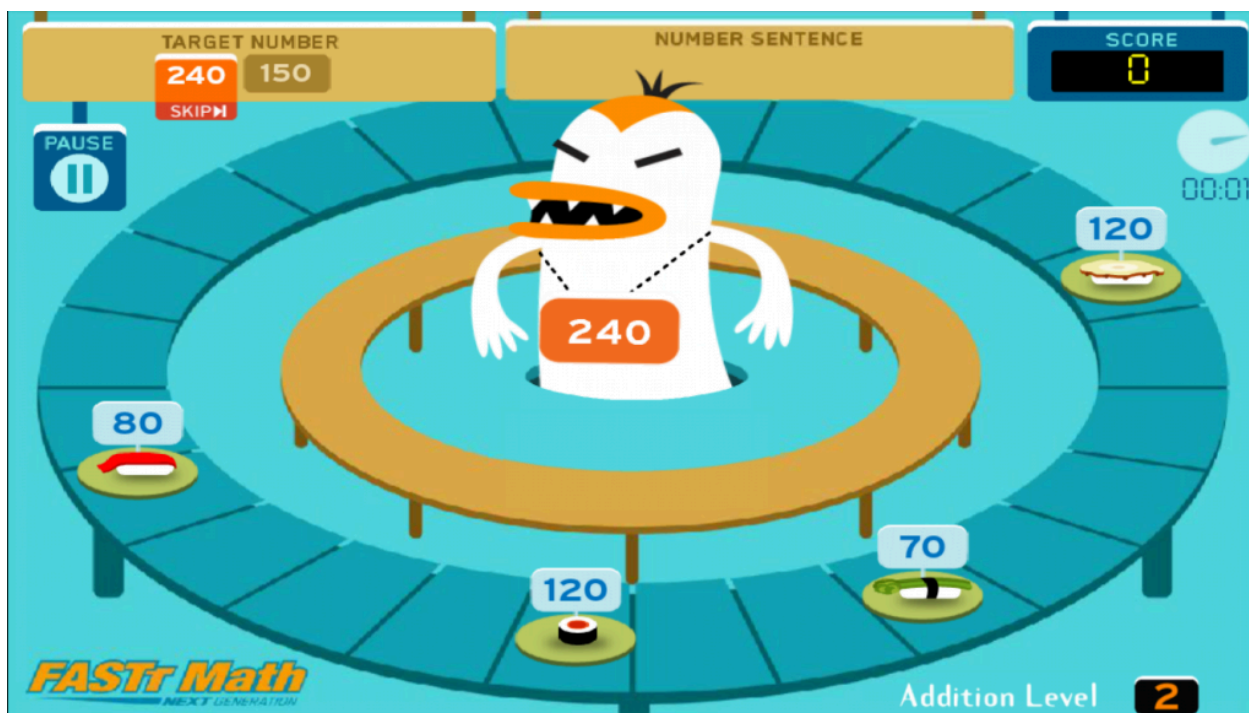
3.5.2.1.3 Sushi Monster

Sushi Monster é um jogo educativo que ajuda crianças a praticar operações matemáticas, como adição e multiplicação, de maneira divertida. *Sushi Monster* exige que os jogadores planejem estrategicamente a escolha de números para resolver equações de adição e multiplicação. Eles devem selecionar os números na ordem correta para garantir que todos os pratos de sushi sejam usados sem sobras, completando os desafios com a pontuação máxima (Sushi Monster, 2017).

O jogo também incentiva a gestão do tempo, pois há um cronômetro que permite aos jogadores acompanhar seu progresso e melhorar sua velocidade na resolução de problemas matemáticos (Sushi Monster, 2017).

Além disso, *Sushi Monster* fortalece a memória de trabalho, ajudando os jogadores a reter e aplicar conceitos matemáticos em diferentes contextos, promovendo habilidades úteis para a vida cotidiana, como planejamento financeiro e tomada de decisões rápidas (Sushi Monster, 2017).

Figura 5 – Interface do jogo Sushi Monster.



Fonte: Sushi monster | app-enfant (2025).

3.6 TEMPO DE TELA

Mistry (2007) em seu estudo documentou que o tempo de tela das crianças está negativamente correlacionado com os resultados de saúde mental, como níveis de ansiedade e abstinência. Por exemplo, descobriu-se que o tempo excessivo de tela dos alunos (ou seja, mais de 2 horas por dia) está relacionado aos seus altos níveis de ansiedade (Taheri et al., 2019).

3.6.1 Impactos Cognitivos e Educacionais do Uso Excessivo de Telas

O uso excessivo de dispositivos eletrônicos na infância tem sido amplamente associado a prejuízos cognitivos, educacionais e comportamentais. Um dos principais efeitos observados é o aumento de dificuldades de atenção e concentração. Análises sistemáticas indicam que o tempo prolongado de exposição a telas está correlacionado com sintomas de déficit de atenção e hiperatividade, especialmente em crianças com menos de dez anos de idade (REVISTA FT, 2021; LISSAK, 2018). Essas crianças tendem a apresentar menor autocontrole e capacidade de foco, o

que se reflete em comportamentos impulsivos e queda no rendimento escolar (TWENGE; CAMPBELL, 2018).

No campo educacional, evidências empíricas também revelam impactos negativos no desempenho acadêmico. Um estudo de coorte realizado no Canadá demonstrou que cada hora adicional de televisão diária aos dois anos de idade esteve associada a uma redução de aproximadamente 6% na proficiência matemática ao final da 4ª série (PAGANI et al., 2013). Além disso, a exposição precoce e prolongada às telas está relacionada à menor capacidade de aprendizado e desempenho cognitivo inferior em etapas posteriores da vida escolar (REVISTA FT, 2021; LISSAK, 2018).

O uso exacerbado de dispositivos eletrônicos compromete ainda o engajamento em atividades cognitivas enriquecedoras, prejudicando diretamente a aquisição de habilidades fundamentais. Segundo Lissak (2018), a superexposição a estímulos digitais pode afetar negativamente funções executivas como a memória de trabalho, o controle inibitório e a flexibilidade cognitiva, fundamentais para a aprendizagem formal.

3.6.2 Estratégias para o Controle do Tempo de Tela

Diante desses riscos, diversas instituições de saúde propõem limites diários para o uso de telas por crianças. A Sociedade Brasileira de Pediatria recomenda evitar totalmente o uso de telas em crianças menores de dois anos e limitar o tempo a no máximo uma hora por dia para crianças de 2 a 5 anos e até duas horas para crianças entre 6 e 10 anos, sempre com supervisão (REVISTA FT, 2021). Diretrizes internacionais como as do CDC nos Estados Unidos sugerem tempo ainda mais restrito: entre 30 minutos e 1 hora diária para crianças de 3 a 7 anos (LISSAK, 2018).

A implementação de regras claras no ambiente familiar, como a proibição do uso de telas no quarto ou durante as refeições, contribui para a redução do tempo de exposição e para a melhora da qualidade do sono e da atenção das crianças (PAGANI et al., 2013). Reforça-se também a importância de criar períodos do dia livres de dispositivos, especialmente antes da hora de dormir.

Com base nesses estudos, o jogo foi desenvolvido com um mecanismo de controle de tempo, permitindo que a criança jogue por no máximo uma hora contínua. Após esse período, o acesso é

temporariamente bloqueado, sendo liberado apenas após algumas horas. Essa estratégia visa fracionar o tempo de exposição ao jogo ao longo do dia, prevenindo os efeitos negativos associados ao uso excessivo de telas, conforme apontado na literatura

4 PLANEJAMENTO DO JOGO PROPOSTO

Neste capítulo, seguindo a metodologia já descrita anteriormente na seção 3.1.1, será detalhado o planejamento do jogo proposto, *design*, *Product Backlog* e protótipos.

4.2 DESIGN E MODELAGEM

A primeira etapa para o desenvolvimento do jogo consistiu na modelagem de sua mecânica e jogabilidade, de modo a atender aos requisitos didáticos e fundamentar-se no referencial teórico sobre as bases que os jogos educacionais devem possuir. Para isso, foi concebida uma dinâmica em que os jogadores solucionam desafios que estimulam seu engajamento e motivação. Optou-se pela criação de um jogo no qual o jogador possui uma ilha, onde desenvolve suas atividades diárias por meio de seu personagem. Nessa ilha, estarão distribuídos portais, cada um vinculado a um conteúdo educacional específico (por exemplo, operações de soma). E com isso, exigir que o jogador entenda e utilize este conceito para que consiga completar o desafio apresentado e possa superar o desafio, construindo o conhecimento necessário para a disciplina de maneira gradual.

Ao acessar os portais, o jogador será direcionado a um minigame, no qual um novo conceito didático será introduzido. Esses minigames serão detalhados posteriormente. Além disso, foi planejado um sistema em que, ao completar desafios, o jogador acumula experiência na disciplina correspondente, desbloqueando habilidades que influenciam as ações que seu personagem pode realizar na ilha. Esse aspecto está alinhado com o primeiro princípio da definição de gamificação proposta por Burke (2015), conforme descrito na seção 3.2.

4.2.1 Requisitos

O sistema consiste em um jogo mobile desenvolvido para crianças e adolescentes do ensino fundamental, independentemente do gênero. Parte-se do pressuposto de que o público-alvo já tenha sido previamente introduzido aos conceitos abordados nos minigames, uma

vez que o objetivo do jogo não é ensinar esses conceitos desde o princípio, mas sim atuar como um recurso de incentivo à aprendizagem, auxiliando na fixação e no aprofundamento dos conhecimentos.

O software enquadra-se na categoria de jogos sérios (serious games) e, portanto, deve integrar elementos de entretenimento com conteúdos educacionais das disciplinas de Língua Portuguesa e Matemática. No contexto dos serious games, o projeto incorpora diversos minigames estruturados em formato de desafios, cada um representando um conceito educacional específico e alinhado ao tema da disciplina correspondente. Os minigames devem ser jogados repetidamente pelos usuários, permitindo a progressão do nível do personagem em cada disciplina.

A seleção do mini game a ser jogado será determinada por um algoritmo Round Robin com pesos, no qual os desafios em que os jogadores não obtiveram sucesso terão um peso maior no ciclo de seleção. Essa abordagem introduz um conceito similar ao método de repetição espaçada, proposto por Sebastian Leitner e denominado Sistema de Leitner. Nesse método, a revisão dos conteúdos ocorre em intervalos ajustáveis de acordo com a dificuldade do usuário em recordar a informação (Pham et al., 2016).

A partir do mapa principal, o usuário poderá selecionar o portal correspondente à disciplina desejada. Durante o jogo, o sistema fornecerá orientações em formato textual, garantindo que cada minigame apresente suas respectivas regras, instruções e demais informações relevantes. O jogo recompensará o progresso do jogador por meio da elevação do nível do personagem, permitindo-lhe utilizar, adquirir ou plantar objetos conforme seu desempenho em cada disciplina. No entanto, não haverá penalizações em caso de erros. Além disso, elementos visuais e sonoros serão incorporados ao longo da experiência para aprimorar a interação do usuário com o jogo.

4.2.2 Requisitos funcionais

[RF1]	O jogo deve possuir um mapa principal onde o personagem irá viver e trabalhar.
[RF2]	O jogo deve permitir que o player escolha os portais a partir do mapa principal

[RF3]	Os portais devem ter uma fila circular com pesos para selecionar o desafio e sua dificuldade a ser disponibilizado.
[RF4]	O jogo deve passar em tempo real (sincronizado com o horário do dispositivo)
[RF5]	O sistema deve fornecer feedback imediato ao player em caso de acerto ou erro.
[RF6]	O jogo deve possuir um sistema onde baseado na combinação dos níveis de habilidade em disciplinas, o personagem é aprimorado automaticamente.
[RF7]	O jogo deve ter um gerador de expressões matemáticas com dificuldades para gerar as expressões para os desafios de matemática.
[RF8]	O jogo deve ter um sistema a fim de permitir a plantação, colheita e o crescimento das plantas.
[RF9]	O jogo deve ter um sistema financeiro em que o personagem possa vender frutos e comprar sementes.

4.2.3 Requisitos não funcionais

[RNF1]	O sistema deve ser compatível com os sistemas operacionais Android e iOS
[RNF2]	O sistema deve ser implementado usando a <i>game engine</i> UNITY.
[RNF3]	O sistema deve armazenar dados em arquivos JSON
[RNF4]	O jogo deve passar em tempo real (sincronizado com o horário do dispositivo)

4.3 PRODUCT BACKLOG

A criação dos *product backlog* do sistema foi feita com base em um conjunto de user stories, essas que serviram para delinear os comportamentos esperados do aplicativo do ponto de vista dos usuários finais, junto com descrição, notas adicionais e critérios para aprovação de cada história.

Cada um deles contém os seguintes elementos:

- Título, contendo resumo da funcionalidade descrita.

- Narrativa, que define o usuário representado, sua vontade e seu objetivo ao completar a ação utilizando a seguinte notação: Como <usuário>, eu quero <fazer algo> de forma que eu possa <alcançar algo>.
- Descrição, contendo detalhes e contexto da história.
- Critérios de aceitação, que especificam o comportamento esperado pelo usuário ao realizar determinada ação em determinada situação. Seguem a seguinte notação: Dado que <situação>, quando <realizo ação>, então <acontece resultado>.
- Notas, descrevendo observações pertinentes e decisões de implementação A seguir estão enumerados os *user stories* descritos para o sistema.

4.3.1 Navegação pelo mapa

Narrativa

Como jogador , **eu quero** poder andar pelo mapa, cultivar plantas , interagir com objetos **de forma que eu possa** explorar a ilha, cuidar da minha fazenda, negociar e entrar nos portais de desafios.

Descrição

O jogo deve ter sua estrutura básica da ilha já disponível para o player, e permitir que com a evolução do personagem ele possa evoluir progressivamente a ilha e sua fazenda conforme o critério do player.

Critérios de aceitação

- **Dado que** estou no mapa, **quando** estou interagindo com um objeto, **então** deve aparecer um ícone indicando qual objeto alvo.
- **Dado que** estou no mapa, **quando** entro na casa , **então** devo ser redirecionado para uma cena com o interior da casa.
- **Dado que** estou no mapa, **quando** estou em um mapa, **então** devo ser redirecionado ao desafio correspondente.

Notas

O jogo terá árvores, pedras e outros objetos que podem ser minerados na ilha, mas a princípio a ideia é não serem interativos, dado o escopo do projeto, mas podendo ser implementado um trabalhos futuros.

4.3.2 Sistema de Fazendinha

Narrativa

Como jogador, **eu quero** poder plantar, colher, comprar sementes e vender frutos **de forma que eu possa** cultivar a plantação de modo que eu ganhe dinheiro vendendo os frutos e assim podendo dar upgrade em minha fazenda.

Descrição

O jogo deve permitir o jogador cuidar da sua fazenda e gerar renda no game para progredir no mundo. O cuidar no contexto do jogo consiste em plantar, regar e colher. Comprar sementes e vender frutos também devem ser permitidos.

Critérios de aceitação

- **Dado que** estou na fazenda, **quando** estou com a ferramenta enxada na mão, **então** deve ser permitido preparar o solo para o plantio.
- **Dado que** estou na fazenda, **quando** estou com o pacote de semente na mão, **então** deve ser permitido plantar a semente no solo preparado.
- **Dado que** estou na fazenda, **quando** estou com a ferramenta regador na mão, **então** deve ser permitido regar as plantas.
- **Dado que** estou na fazenda, **quando** estou interagindo com as plantas já crescidas, **então** deve ser permitido colher os frutos.

Notas

O sistema de colheita deve passar em tempo real, visto isso, plantas podem demorar dias para crescerem e serem cultivadas, com isso exigindo cuidados durante o processo de cultivo.

4.3.3 Sistema de Stamina

Narrativa

Como Pai eu quero que o meu filho não fique mais de 1 hora seguida jogando **de forma que ele possa** gerenciar a stamina, para fazer o máximo de coisas no dia.

Descrição

O jogo deve implementar um sistema de **stamina** que limita as ações do personagem. Cada ação realizada consumirá uma quantidade determinada de stamina, e esta diminuirá progressivamente conforme o tempo de jogatina. A recuperação da stamina será baseada no tempo real, ao invés dos métodos tradicionais, como simplesmente fazer o personagem dormir. Isso adiciona um elemento estratégico ao gerenciamento de recursos, incentivando o jogador a planejar suas ações com cuidado. E principalmente diminuindo o tempo de tela dos jogadores.

Crítérios de aceitação

- **Dado que** estou no jogo, **quando** estou explorando a ilha, **então** a stamina deve ser consumida mais rapidamente.
- **Dado que** estou no jogo, **quando** estou realizando desafios, **então** a stamina deve ser consumida mais lentamente.
- **Dado que** estou fora do jogo, **quando** o personagem está dormindo, **então** a stamina deve ser restaurada baseado no tempo passado fora do jogo.

Notas

Como visto na seção 3.6, o tempo de tela pode causar problemas sérios em crianças, portanto, baseado na pesquisa sobre o tema, o tempo no jogo será finito e obrigará o jogador a estabelecer uma rotina de jogatina em intervalos durante o dia, reduzindo o tempo de tela contínuo. Várias atividades, como o crescimento de plantações, a colheita e as compras no mercado virtual, seguirão horários específicos, juntamente com uma mecânica de vigor, que evitará que o jogador fique longos períodos no jogo. Isso incentivará o jogador a distribuir suas atividades ao longo do dia, transformando o tempo gasto no jogo em uma experiência educativa mais eficaz e menos maçante, ao contrário de muitas das formas tradicionais de estudo.

4.3.4 Jogar mini game caverna das palavras

Narrativa

Como jogador, **eu quero** jogar o minigame caverna das palavras **de forma que dado** o tema, eu descubra a palavra **e então** amplie o meu vocabulário.

Descrição

O jogador, ao receber um tema determinado, deverá descobrir a palavra oculta. O jogo exibirá o número de letras da palavra, inicialmente ocultas, projetadas na parede da caverna. No chão da caverna, haverá botões contendo diferentes letras, que poderão ser ativados ao serem pressionados. O jogador deverá selecionar uma letra dentre as opções disponíveis. Caso a escolha esteja correta, a letra será revelada na palavra oculta, e um feedback visual será fornecido: luz verde para acertos e luz vermelha para erros. Quando o jogador completar a palavra corretamente, uma chave será apresentada, permitindo-lhe abrir o portão e concluir o desafio. À medida que o jogador acerta as palavras, a dificuldade do jogo aumentará progressivamente.

CrITÉrios de aceitação

- **Dado que** estou na ilha (mapa principal), **quando** escolho entrar no portal e o algoritmo escolhe a caverna das palavras, **então** sou redirecionado para a entrada da caverna onde terá uma descrição falando o tema da palavra oculta.
- **Dado que** estou no desafio caverna das palavras, **quando** em várias tentativas e não acerto a palavra, **então** perco uma vida e outra palavra é gerada.
- **Dado que** estou no desafio caverna das palavras, **quando** perdi as 3 vidas exibidas na tela, **então** volto à ilha principal.
- **Dado que** estou no desafio caverna das palavras, **quando** interajo com a porta de entrada, **então** volto à ilha principal.

Notas

O jogo busca as palavras em arquivos JSON que são separados por temas.

4.3.5 Jogar mini game interpreta em ação.

Narrativa

Como jogador, **eu quero** jogar o minigame interpretação em ação **de forma que dado** um texto contendo pistas sobre qual é o tesouro, a localização de onde ele está e a ordem correta de coleta **para que então** eu baseado na minha interpretação de texto colete os itens conforme o descrito.

Descrição

O jogador, dado um texto que será apresentado a ele, deverá buscar os tesouros citados no texto respeitando a ordem descrita e a quantidade. O mapa terá os tesouros espalhados pelo mapa que terá pontos de referências para facilitar a sua localização. A cada objeto encontrado ele será apresentado na tela como encontrado. Em caso de coleta de um objeto incorreto ou fora de ordem, um feedback visual aparecerá na tela com uma dica sobre o tesouro. Ao coletar todos os itens descritos na história, uma chave aparece para o player e o desafio conquistado.

CrITÉRIOS de aceitação

- **Dado que** estou na ilha (mapa principal), **quando** escolho entrar no portal e o algoritmo escolhe o desafio, **então** sou redirecionado para o mapa em uma localização próxima de onde o primeiro tesouro está localizado.
- **Dado que** estou no desafio, **quando** após um tempo não consegui encontrar os tesouros, **então** um botão para voltar deve me mandar para a minha ilha.
- **Dado que** estou no desafio, **quando** eu coletar um tesouro no mapa, **então** o slot do tesouro vai interagir de acordo com o status, verde se certo, vermelho se errado.
- **Dado que** estou no desafio, **quando** procuro pelos tesouros, **então** eles devem apresentar um destaque para facilitar a visualização.

Notas

O jogo busca as histórias em arquivos JSON, onde contém o ID, o texto contendo a história, as dicas e a ordem dos tesouros apresentados no texto.

4.3.6 Jogar mini game ponte aritmética.

Narrativa

Como jogador, **eu quero** jogar o minigame ponte aritmética **de forma que dado** expressões aritméticas mostradas na tela, devo às resolver mentalmente **para que então** ao selecionar a resposta correta na plataforma e assim me permitindo progredir pelo percurso até alcançar o final do desafio. Caso eu erre eu caio da plataforma e volto do início.

Descrição

O jogador, dado uma expressão aritmética que aparecerá na tela, deve escolher entre as 2 opções de respostas que estão nas plataformas, uma correta e outra errada. Se pular na certa ele não cai e pode avançar, caso escolha a errada ele cai da plataforma e volta para o início, assim perdendo uma vida e mantendo as plataformas que ele acertou.

CrITÉrios de aceitação

- **Dado que** estou no minigame , **quando** caio do da ponte, **então** retorno para a entrada da ponte e perco uma vida.
- **Dado que** estou no minigame, **quando** perco 3 vidas, **então** devo retornar para a minha ilha.
- **Dado que** estou no minigame, **quando** eu chegar ao final da ponte, **então** vai ser permitido a interação com pergaminho que determina a conclusão do desafio.

Notas

As expressões aritméticas devem ser geradas baseadas em um algoritmo que cria essas expressões levando em conta a quantidade de operadores, precedência de operadores e número de operações. Para que não seja preciso armazenar as expressões em arquivos.

4.3.7 Jogar mini game futemática.

Narrativa

Como jogador, **eu quero** jogar o futemática **de forma que dado** várias bolas de futebol espalhadas pelo campo associadas a números ou operadores matemáticos e um placar de fácil visualização exibindo um número-alvo gerado aleatoriamente **para que então** eu chute bolas ao gol de forma estratégica, construindo uma expressão matemática cujo resultado corresponda ao número exibido no placar.

Descrição

O jogador terá à disposição bolas espalhadas pelo campo, cada uma associada a um número de 1 a 9 ou a um dos operadores matemáticos: adição, subtração e multiplicação. Para conquistar um troféu, o jogador deverá construir uma expressão aritmética cujo resultado seja igual ao número exibido no placar. Ao acumular três troféus, o jogador alcança o sucesso no desafio.

CrITÉRIOS de aceitação

- **Dado que** estou no minigame , **quando** chuto uma bola ao gol, **então** o valor da expressão é exibido dentro do gol.
- **Dado que** estou no minigame , **quando** chuto uma bola ao gol, **então** a bola deve sumir.
- **Dado que** estou no minigame , **quando** entro no banco de reserva, **então** reinicio a tentativa, gerando um novo número, gerando novas bolas, assim perdendo uma vida.
- **Dado que** estou no minigame , **quando** chutei todos números ou todas operações disponíveis, **então** perco uma vida e um novo número é gerado e novas bolas são repostas no campo.

Notas

As expressões aritméticas devem ser geradas utilizando o mesmo algoritmo empregado no outro minigame, garantindo consistência na lógica de criação. As expressões serão retornadas em formato de texto para facilitar o processamento, permitindo o mapeamento dos operadores utilizados e a exibição do resultado da expressão no placar. Dessa forma, assegura-se que sempre haverá uma solução disponível para o jogador.

4.4 PROTÓTIPOS

Para a simulação dos minigames e da ilha principal, foram desenvolvidos protótipos diretamente no Unity, utilizando *assets* disponíveis na *Unity Asset Store*. Esses protótipos serviram como artes conceituais. Segundo Schell (2008), em seu livro *The Art of Game Design*, a arte conceitual pode fornecer a visão unificadora da experiência que um jogo está tentando alcançar. Outras vezes, uma ilustração pode deixar claro se uma ideia de interface é possível ou não. Os *designs* de jogos são abstratos, porém as ilustrações são concretas. No doloroso processo

de converter seu *design* abstrato em um jogo concreto, as ilustrações podem servir como uma maneira simples e eficaz de fundamentar seu *design* na realidade logo no início de um projeto (Schell, 2008).

4.4.1 Ilha principal

A ideia do mapa principal é ser uma ilha onde o jogador “vive” com o personagem, pensando nisso o mapa deve ter um ambiente de ilha, cheio de árvores, rochas e o mar ao redor. Apesar de aparentar ser grande, o *design* é pensado para que o personagem possa ter tudo perto. Na figura 6, o item A mostra o local onde a fazenda do personagem estará disponível em seu nível básico, para assim ele conseguir a partir do desenvolvimento em jogo e sucesso nos desafios evoluir a fazenda; o item B e c mostra onde os portais serão localizados e o item C onde ficará uma pequena cidadezinha.

Figura 6 – Visão aérea do mapa principal.



Fonte: Própria, 2025. (a)fazenda; (b) portais; (c)vila

Na Figura 7 – A e B, estão indicados os locais onde os portais serão posicionados. A ideia é que eles fiquem próximos à fazenda do personagem, situados na saída da fazenda ou na direção onde a cidade será construída. Essa localização foi escolhida para facilitar o acesso, evitando que o jogador perca tempo ao se deslocar até os portais.

Figura 7 – Visão aérea dos portais.

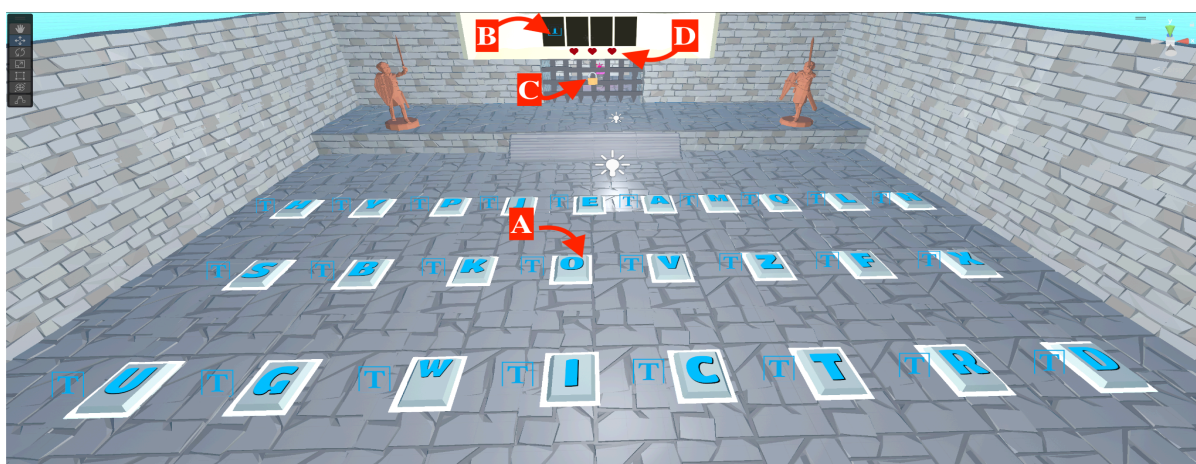


Fonte: Própria, 2025. (a) portal de matemática; (b) portal de português

4.4.2 Caverna das palavras

O *design* do mapa desse mini game foi pensando para o desenvolvimento do vocabulário do jogador e levando em conta a importância do vocabulário e a importância para os alunos, teoria já discutida na seção 3.3.1.

Figura 8 – Visão do minigame caverna das palavras.



Fonte: Própria, 2025. (a) botão; (b) painel; (c) portão; (d) vidas

A Figura 8 apresenta o mapa da caverna e a estrutura do minigame. O Item A representa o botão que, quando pressionado, verifica se a letra escolhida está presente na palavra oculta. Se a letra existir, a palavra será atualizada no painel correspondente (Item B), exibindo os textos de

forma atraente e lúdica na parede, conforme descrito na Seção 3.5.1. Caso a letra não esteja na palavra, a luz do botão muda para vermelho, indicando o erro. O Item D exibe a quantidade de tentativas disponíveis para o personagem, fornecendo feedback imediato sobre os erros. Já o Item C representa um portão com um ícone de cadeado, sinalizando que está trancado e só poderá ser aberto após a descoberta da palavra oculta. A chave necessária para destravá-lo aparecerá apenas quando o jogador acertar a palavra.

4.4.3 Interpreta em ação

O mini game desenvolvido foi pensado visando trabalhar as habilidades cognitivas e linguísticas descritas por Kate Cain (2009), conforme discutido na Seção 3.3.2. A mecânica do jogo foi projetada para estimular e fortalecer a **compreensão de leitura**, promovendo a construção de significado a partir do texto por meio da interação ativa do jogador.

E pensando em *level design*, o livro *The Art of Game Design: A book of lenses*, Schell, J. (2008) descreve que um dos segredos para um bom *level design* em jogos é fazer com que os olhos do jogador o conduzam pelo cenário de maneira natural e sem esforço. Isso faz com que o jogador se sinta no controle e imerso no mundo do jogo. Compreender o que atrai o olhar do jogador pode dar um enorme poder sobre as escolhas que ele deseja fazer. Para o *design* do mapa, foi utilizado um conceito de *level design* chamado “*weenie*”, criado por Walt Disney, refere-se ao uso de elementos visuais estratégicos para guiar o olhar e a movimentação do público de forma intuitiva. Ainda no livro, ele cita que um exemplo clássico é o castelo no centro da Disneylândia, que atrai imediatamente a atenção dos visitantes ao entrarem no parque, conduzindo-os naturalmente para o centro antes de se dispersarem para outras áreas. Esse princípio é amplamente utilizado por cenógrafos, ilustradores, arquitetos e diretores de fotografia para direcionar o foco e influenciar a experiência do espectador (Schell, 2008).

Para o mapa do desafio pensou-se em utilizar um mapa aberto utilizando o conceito de *level design weenie*, no mapa é onde se passará as histórias e pensado de forma que tudo possa ser identificado pelo jogador sem ambiguidades, de forma a não confundir os locais o mapa possui ambientes distintos uns dos outros conforme a Figura 9.

Figura 9 – Visão aérea do mapa do interpreta em ação.



Fonte: Própria, 2025. (a) comércio; (b) pirâmides; (c) acampamento; (d) oriente; (e) faroeste; (f) inverno; (g) ilha caveira; (h) castelo final.

A Figura 9 ilustra claramente os diferentes ambientes do jogo, cada um com referências visuais distintas que auxiliam na orientação do jogador, fazendo uso do conceito de level design “weenie”, conforme descrito anteriormente. Esses elementos foram projetados para garantir uma identidade visual única para cada localidade, facilitando a imersão e a navegação no mapa, trabalhando assim a habilidade linguística de **integração de informações**, explicada anteriormente na Seção 3.3.2. O **Item A** representa uma área comercial, composta por barracas de cores distintas, cada uma vendendo diferentes tipos de itens. O **Item B** destaca-se pela presença de três pirâmides de tamanhos variados, acompanhadas de uma esfinge e uma árvore com formato e coloração diferenciados, reforçando a temática do ambiente e utilizando-se do “weenie”, conforme descrito anteriormente. O **Item C** apresenta um acampamento, no qual barracas coloridas e uma vegetação exclusiva criam um cenário diferenciado, posicionado próximo a uma casa azul isolada. O **Item D** representa um palácio chinês, cercado por um sakura lilás e envolvido por um corpo d’água. No protótipo da imagem, não foi possível adicionar um rio, mas essa característica está prevista para a versão final e também faz uso do “weenie”. O **Item E** remete ao Velho Oeste, contendo elementos característicos como cactos, pedras exclusivas e uma estação de trem, além de ser rodeado por cavalos, carroças e um grande foguete que se destaca à distância. O **Item F** corresponde a uma região de inverno, caracterizada por

árvores cobertas de neve e decorações natalinas espalhadas pelo ambiente e pelas casas. O **Item G** apresenta uma ilha em formato de caveira, com barcos naufragados ao seu redor, criando uma atmosfera de mistério e aventura e o conceito “*weenie*”. Por fim, o **Item H** representa o castelo final, que pode ser visto de longe devido utilizando-se do conceito “*weenie*”, onde está guardado o pergaminho que encerra o jogo. Após o personagem coletar todos os itens conquistados por meio da interpretação de textos ao longo da jornada, uma chave surge, permitindo a abertura do portão do castelo e o acesso à conclusão do desafio.

O posicionamento do personagem no mapa sempre ocorre próximo à localização do primeiro item, garantindo um percurso linear e reduzindo a necessidade de deslocamentos excessivos. No entanto, o nível de dificuldade do desafio pode influenciar essa progressão, exigindo maior exploração conforme a complexidade do jogo aumenta.

Figura 10 – Visão da interface do jogador.



Fonte: Própria, 2025. (a) player; (b) ícones de tesouros; (c) botão que abre a história; (d) tesouros coletáveis.

A Figura 10 apresenta a interface do jogador no minigame, destacando os principais elementos interativos que o *design* foi projetado para se alinhar com o conceito de **monitoramento da compreensão** descrito na Seção 3.3.2.

O **Item A** representa o personagem principal do jogo, responsável por explorar o ambiente e interagir com os desafios propostos. O **Item B** exibe os campos onde as atualizações ocorrem conforme os itens são coletados. Quando um item é adquirido corretamente, o ícone azul muda para verde, indicando sucesso. Caso o jogador colete um tesouro incorreto, o ícone se torna vermelho, acompanhado de um texto na tela que fornece uma dica para auxiliar na interpretação e na tomada de decisão. O **Item C** corresponde ao botão que, quando pressionado, abre um pergaminho contendo o texto a ser interpretado, fornecendo informações essenciais para a progressão no jogo. Já o **Item D** destaca os itens coletáveis no mapa, facilitando sua identificação e auxiliando na navegação do jogador dentro do ambiente do minigame.

Figura 11 – Visão da interface do pergaminho com texto.



Fonte: Própria, 2025.

A Figura 11 apresenta a interface de exibição do texto para o personagem, permitindo uma interação dinâmica com as informações do jogo e seu *design* pensado para a **inferência** tenha um aspecto central da mecânica do mini game, uma vez que o jogador precisa deduzir informações que não estão diretamente expostas no texto, como destacado na Seção 3.3.2 e exigindo constantemente a **memória de trabalho** e a **memória de curto prazo**.

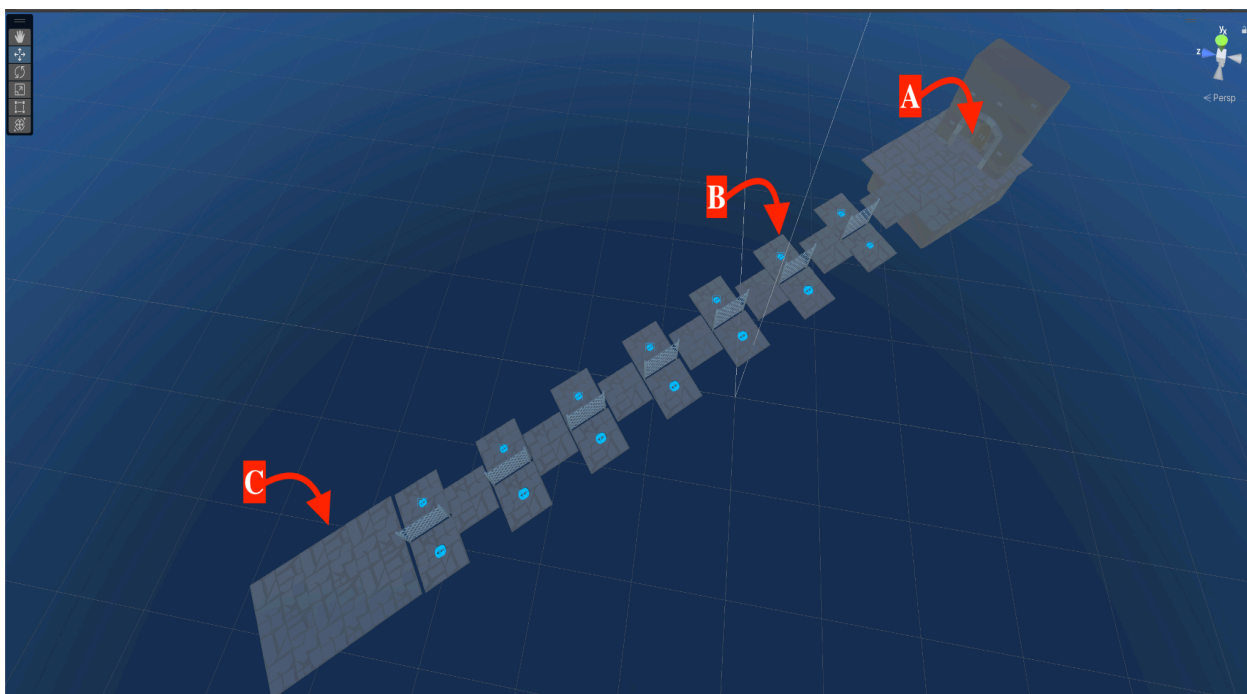
O jogador tem a opção de pressionar um botão para abrir ou fechar o pergaminho conforme necessário, proporcionando maior flexibilidade na consulta do texto durante a

gameplay. Além disso, o sistema de feedback visual destaca os itens já coletados ao longo do jogo. Sempre que um item for adquirido, ele será riscado no texto, facilitando a identificação dos elementos já encontrados e contribuindo para uma melhor compreensão do progresso dentro do minigame. Assim, auxiliando o jogador na organização das informações, na tomada de decisões estratégicas ao longo da experiência e trabalhando as habilidades cognitivas e linguísticas da **decodificação das palavras e compreensão da estrutura textual**, descritas na Seção 3.3.2.

4.4.4 Ponte aritmética

O primeiro minigame de matemática desenvolvido, denominado Ponte Aritmética, teve seu level *design* e protótipo elaborados com o objetivo de reforçar no jogador as habilidades fundamentais para uma aritmética eficiente, conforme já discutido na Seção 3.4.1.

Figura 12 – Visão aérea do mapa da ponte aritmética .

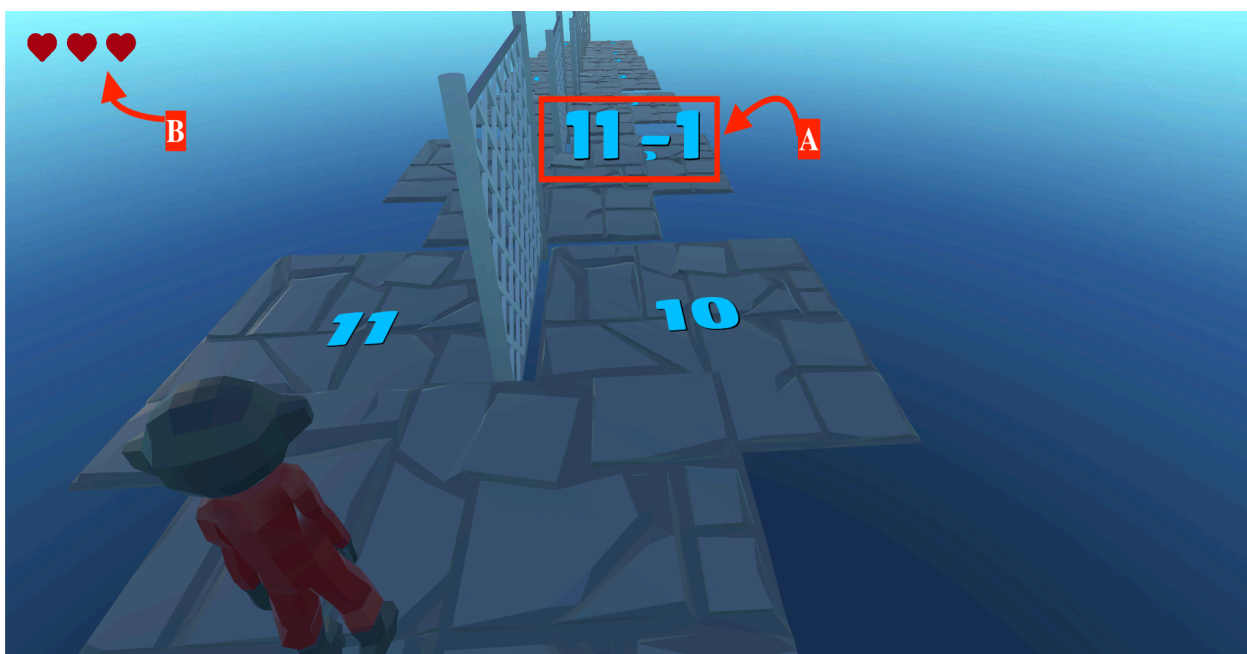


Fonte: Própria, 2025. (a) entrada; (b) plataformas com as respostas; (c) chegada

A **Figura 12** apresenta a estrutura do minigame Ponte Aritmética, destacando seus principais elementos. O **item A** indica o ponto de início do personagem, que também serve como local de reinício caso ele caia da ponte durante o desafio, garantindo que o jogador possa tentar novamente sem comprometer a experiência. O **item B** representa os degraus flutuantes, cada um

exibindo um número correspondente ao possível resultado de um cálculo matemático. O jogador deve analisar a equação proposta e escolher a plataforma correta para avançar com segurança. Por fim, o **item C** marca a chegada do desafio, onde deve ter um altar contendo o Pergaminho do Conhecimento que aguarda o jogador. Ao alcançar esse ponto, o desafio é concluído com sucesso.

Figura 13 – Visão da interface da ponte aritmética .



Fonte: Própria, 2025. (a) expressão a ser respondida; (b) vidas.

A **Figura 13** ilustra a interface projetada para o minigame Ponte Aritmética, destacando os elementos essenciais para a interação do jogador com o desafio. O **Item A** exibe a expressão aritmética que o jogador deve resolver para escolher a plataforma correta. Essa expressão é gerada automaticamente assim que o personagem alcança o degrau imediatamente anterior às plataformas de escolha, garantindo uma experiência dinâmica e imprevisível a cada tentativa. O **item B** representa o contador de tentativas, indicando a quantidade de chances disponíveis para o jogador concluir o desafio, incentivando o jogador a pensar com rapidez e precisão para evitar erros que possam comprometer seu progresso.

4.4.5 Futemática

Neste desafio, o objetivo é reverter a mecânica proposta no Ponte Aritmética, onde o jogador deveria escolher a resposta correta com base em uma expressão aritmética. Neste mini game, o jogador recebe um número gerado aleatoriamente e deve criar uma expressão cujo resultado seja igual ao número apresentado. A estrutura do jogo foi projetada para viabilizar a verificação do pensamento quantitativo e do sentido numérico, conceitos discutidos nas Seções 3.4.1 e 3.4.2, respectivamente.

Figura 14 – Visão da interface da ponte aritmética.



Fonte: Própria, 2025. (a) bola com multiplicação; (b) bola com 9; (c) vidas; (d) resultado alvo; (e) gol.

A Figura 14 apresenta o protótipo completo do jogo, com todos os elementos projetados para facilitar a experiência do usuário e otimizar a jogabilidade. O primeiro detalhe importante é que o campo foi dividido ao meio, permitindo que o jogador não precise se deslocar de um lado a outro, facilitando a navegação e tornando a experiência mais fluida. O **item A** exibe uma bola, que indica qual operador matemático está associado a ela, elemento visual ajuda o jogador a identificar rapidamente as operações envolvidas no desafio. O **item B** representa o número que a bola está associada, fornecendo o valor a ser utilizado na construção da expressão matemática. O **item C** mostra o contador de tentativas em forma de corações, indicando quantas chances o jogador tem para solucionar o desafio corretamente. O **item D** é o número alvo, que é o resultado

desejado para o cálculo, orientando o jogador sobre o valor final que deve ser alcançado ao combinar corretamente os números e operadores apresentados. O **item E** mostra o gol, onde as bolas espalhadas pelo campo devem ser chutadas para que o resultado preliminar seja exibido dentro do gol, permitindo que o jogador acompanhe em tempo real o progresso e o impacto das escolhas feitas ao montar a expressão matemática.

Embora não esteja indicado na Figura 14, o jogo final contará com ícones de troféus que serão concedidos ao jogador conforme ele monta a expressão correta. O objetivo é que o jogador possa ganhar até três troféus, que representam o sucesso no desafio.

A conclusão do minigame ocorre quando o jogador coleta o Pergaminho do Conhecimento, que será instanciado ao final da missão, simbolizando a vitória e a finalização do desafio. Esses elementos visuais e interativos foram pensados para garantir uma experiência envolvente, educativa e motivadora para o jogador, ao mesmo tempo que promovem o desenvolvimento das habilidades matemáticas essenciais discutidas nas seções anteriores.

5 DESENVOLVIMENTO DO JOGO COLHEITA DO SABER

Como proposta de solução, este trabalho teve como objetivo projetar e desenvolver um jogo mobile voltado ao incentivo da aprendizagem, a ser elaborado no âmbito da disciplina de Trabalho de Conclusão de Curso II. Embora os detalhes do jogo já tenham sido previamente apresentados, cabe destacar que se trata de um Farm Game educativo — um tipo de jogo no qual o jogador assume o controle de um personagem capaz de plantar, colher e gerir uma economia interna. Essa economia possibilita a aquisição de novos itens e a construção personalizada da fazenda.

O jogo não contempla elementos de combate e adota um sistema de stamina, destinado a limitar o tempo de tela e, conseqüentemente, incentivar o uso equilibrado. O progresso do jogador está atrelado à superação de desafios educacionais, de forma que, a cada nível alcançado nas habilidades de Português e Matemática, novos conteúdos e recursos são desbloqueados. Entre esses recursos, destacam-se ferramentas aprimoradas, sementes de maior valor e culturas que geram produtos mais lucrativos. Essa dinâmica permite a implementação de um level design evolutivo, no qual a complexidade e as recompensas aumentam gradativamente conforme o avanço do jogador.

Neste capítulo, apresentam-se de forma detalhada as etapas de desenvolvimento do jogo proposto, fundamentadas na metodologia Scrum e organizadas a partir dos respectivos *backlogs*. As fases contemplam os seguintes aspectos: interface de usuário, estética e gráficos, mecânicas, persistência de dados e efeitos sonoros.

5.2 INTERFACE DE USUÁRIO

A interface do jogo foi desenvolvida especificamente para dispositivos móveis, com foco na usabilidade e acessibilidade. Os botões possuem dimensões ampliadas e tipografia em negrito, a fim de facilitar a leitura e a interação por parte do jogador. Considerando tratar-se de um jogo voltado para plataformas *mobile*, foi implementado um sistema de *joysticks* virtuais na tela, permitindo o controle da movimentação do personagem, além de botões dedicados para ações específicas, como pular, interagir e acessar o inventário, conforme ilustrado na Figura 15.

Além disso, a interface exibe informações essenciais ao progresso do jogador, como a barra de estamina e o saldo de recursos financeiros. No inventário, são também apresentados os níveis de habilidade de conhecimento, permitindo ao jogador acompanhar sua evolução nas áreas trabalhadas pelo jogo.

Figura 15 – Visão da interface mobile



Fonte: Própria, 2025.

O botão de ação, em específico, foi desenvolvido com ícones dinâmicos, permitindo que o jogador identifique de forma intuitiva qual ação será executada no momento, sem a necessidade de inserir textos na tela, evitando, assim, a poluição visual da interface. Essa abordagem favorece a experiência do usuário, pois reduz a sobrecarga cognitiva e acelera a compreensão, aspecto relevante em jogos mobile, onde o espaço da tela é limitado e a interação precisa ser ágil.

Para implementar essa funcionalidade, foram utilizados *prefabs* e *ScriptableObjects*. No Unity, prefabs são objetos de jogo pré-configurados, salvos como ativos no projeto, que podem ser reutilizados em diversas cenas ou instanciados dinamicamente em tempo de execução. Eles funcionam como modelos, permitindo a criação de múltiplas cópias de um objeto mantendo as mesmas configurações, como componentes, transformações e dados. Já um *ScriptableObject* é uma classe especial utilizada para o armazenamento e gerenciamento de dados de forma independente dos *GameObjects* e Componentes presentes nas cenas. Ao contrário das classes derivadas de *MonoBehaviour*, que necessitam estar associadas a um objeto na hierarquia, eles

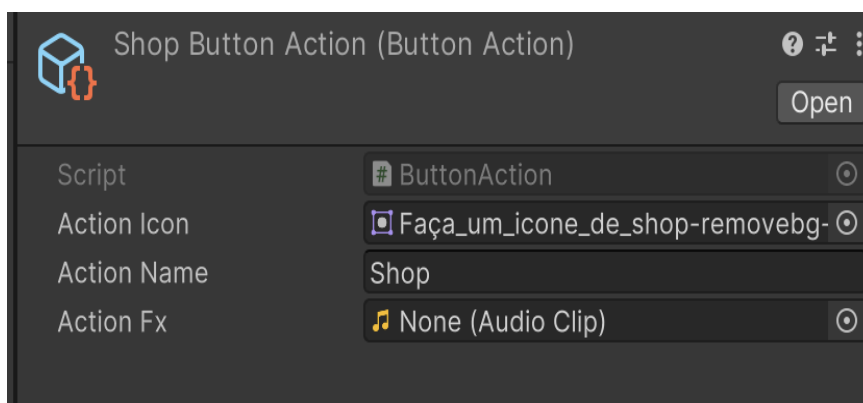
podem existir como assets no projeto, permitindo a criação de instâncias reutilizáveis e centralizadas. De acordo com a documentação oficial do Unity (Unity Technologies, 2025), o uso de *ScriptableObjects* proporciona vantagens como a redução do consumo de memória, uma vez que diversos objetos podem compartilhar uma mesma instância de dados, e a melhoria na organização do código e do projeto, ao separar a lógica de execução da representação dos dados. Além disso, são amplamente empregados no desenvolvimento de jogos para o armazenamento de informações que necessitam ser reaproveitadas em diferentes contextos, como atributos de personagens, parâmetros de configuração, inventários e habilidades.

No projeto, foi criado um *ScriptableObject* denominado *Button Actions*, composto por três elementos principais:

- **ActionIcon** – ícone correspondente à ação;
- **ActionName** – utilizado para identificar internamente a ação que será executada;
- **ActionFX** – um *SoundClip* destinado à reprodução de efeitos sonoros, caso a ação possua áudio associado.

A estrutura desse componente pode ser observada na Figura 16.

Figura 16 – Estrutura do Button Action..



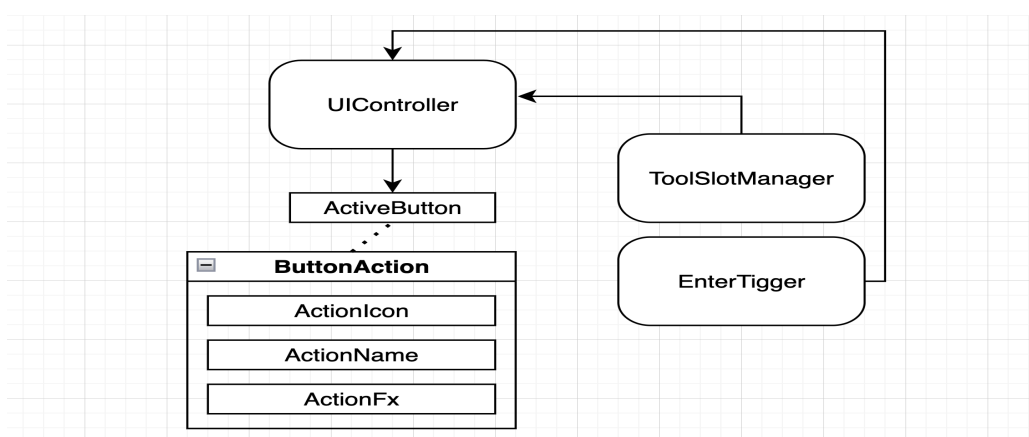
Fonte: Própria, 2025.

Esse componente é utilizado e gerenciado por um *UIController*, um componente global responsável pelo controle de toda a interface do jogo. Em sua função denominada *ActiveButton*, o *UIController* recebe um objeto do tipo *ButtonAction* como parâmetro de entrada e, a partir dele,

realiza a manipulação dos ícones, ativando o botão já com o ícone da ação correspondente disponível para o jogador.

A ativação do *UIController* pode ocorrer por meio de um Trigger ou quando uma ferramenta é selecionada no *ToolSlotManager* — outro componente global, cuja função é carregar e descarregar as ferramentas na mão do personagem de acordo com as interações realizadas durante o jogo. Segue o Diagrama ilustrando a interação entre os componentes.

Figura 17 – Diagrama de interação do botão de ação.



Fonte: Própria, 2025.

5.2.1 INTERFACE NOS MINIGAMES

Assim como no mapa principal, os minigames também possuem interfaces exclusivas, adaptadas conforme o tipo de desafio proposto. Ao entrar em um desses mapas, o sistema realiza uma limpeza na HUD (*Heads-Up Display*), tornando-a mais limpa e exibindo apenas as informações pertinentes à mecânica específica do minigame.

Conforme apresentado anteriormente nos protótipos descritos na Seção 4.4., alguns mapas incluem textos e botões essenciais para a execução das mecânicas. Por exemplo, no mapa da Ponte do Cálculo (Figura 13), é exibida a equação que deve ser resolvida. Já no mapa do InterpretAção (Figura 10), são apresentadas dicas textuais que auxiliam o jogador na realização das tarefas propostas e a quantidade de tesouros a serem encontrados..

A adoção desse formato dinâmico de interface garante que cada minigame ofereça um ambiente focado e livre de distrações, contribuindo para que o jogador concentre-se nos objetivos centrais e tenha maior imersão no desafio proposto.

5.3 ESTÉTICA E GRÁFICOS

A estética de um jogo desempenha papel fundamental, tanto para a criação de uma identidade visual única e uma conexão emocional com o jogador, quanto como recurso para otimização de desempenho, especialmente em títulos desenvolvidos para dispositivos móveis. Jogos podem adotar uma ampla variedade de estilos visuais; neste projeto, optou-se por mesclar dois estilos amplamente consolidados no mercado mobile e bem aceitos por crianças e adolescentes: o estilo *low poly* em 3D, que utiliza baixa contagem de polígonos para gerar modelos mais leves, com influências de anime e *cel-shading*, incorporando elementos de fantasia e um mundo inspirado em *The Legend of Zelda*. Os personagens apresentam designs vibrantes e detalhados, inspirados na estética dos animes, enquanto o cenário exibe paisagens amplas e visualmente exuberantes.

Considerando que o jogo foi concebido para a plataforma mobile, a proposta estética prioriza a leveza, evitando modelos excessivamente complexos e de alta contagem poligonal. Foram utilizadas texturas simples, mas que mantêm apelo visual satisfatório, garantindo que a experiência do jogador não seja comprometida.

Grande parte dos modelos e texturas empregados no projeto foram obtidas por meio da loja oficial de assets do Unity e do site *PolyPizza*, que disponibiliza modelos 3D open source para uso não comercial.

Para a construção do mundo do jogo, foi utilizado o *Terrain*, um asset nativo do Unity que possibilita a criação de terrenos de forma prática e visualmente detalhada. Essa ferramenta permite modelar vales, montanhas, crateras e outros relevos de maneira intuitiva, utilizando recursos de edição diretamente na cena.

O *Terrain* também oferece suporte para aplicação de texturas 3D — como gramados, folhagens e rochas — bem como para a inserção de modelos tridimensionais, como árvores, pedras e flores, por meio de um sistema de pincel (brush). Esse recurso possibilita ao desenvolvedor “pintar” o ambiente de forma personalizada, ajustando a composição do cenário de acordo com a proposta estética do jogo.

Além disso, o sistema permite a sobreposição de diferentes camadas de textura, possibilitando, por exemplo, pintar áreas de grama, regiões arenosas, superfícies rochosas ou encostas montanhosas, garantindo maior riqueza visual e coerência ambiental dentro do mundo virtual.

Para os modelos de elementos que integram o meio ambiente do jogo, foram selecionados assets que conciliam leveza gráfica e qualidade estética, de forma a garantir desempenho otimizado em dispositivos móveis sem comprometer o apelo visual. Entre os recursos utilizados, destacam-se:

- **Cartoon FX Remaster Free** – pacote de efeitos visuais estilizados, ideal para adicionar elementos animados e partículas com baixo custo de processamento;
- **Low Poly Atmospheric Locations Pack** – conjunto de cenários e estruturas em estilo low poly, projetados para criar ambientes imersivos com uso reduzido de polígonos;
- **POLYGON – Sampler Pack (Synty Studios)** – coleção de modelos low poly variados, amplamente utilizados em jogos de mundo aberto pela sua versatilidade e otimização;
- **Simple Low Poly Nature Pack (NeutronCat)** – pacote voltado para vegetação e elementos naturais em baixo polígono, com foco em simplicidade e legibilidade visual;
- **Lowpoly Environment – Nature Free (Medieval Fantasy Series) (Polytope Studio)** – conjunto de modelos gratuitos que inclui árvores, rochas e outros elementos naturais, com estética cartoon e baixo impacto no desempenho.

Figura 18 – Mundo criado pelo TERRAIN



Fonte: Própria, 2025.

Para compor o cenário de praia no jogo, tornou-se essencial a implementação de uma água visualmente atrativa e otimizada, uma vez que superfícies aquáticas dinâmicas, por envolverem cálculos de reflexão, movimentação, profundidade e outros efeitos gráficos, tendem a demandar alto processamento, o que pode comprometer o desempenho em dispositivos móveis. No projeto, foi utilizado o asset AQUA Lite, desenvolvido pela Dogmatic e disponibilizado na biblioteca da Unity.

Esse recurso oferece um conjunto de funcionalidades que equilibram qualidade visual e desempenho, incluindo:

- Absorção de cor realista (Realistic Color Absorption);
- Transparência baseada em profundidade (Depth-Based Transparency);
- Reflexões em tempo real (Realtime Reflections);
- Reflexões via Reflection Probe;
- Distorção de reflexos (Reflection Distortion);

- Bordas suaves na costa (Soft Edges at Shore);
- Personalização da linha costeira (Customisable Shoreline);
- Entre outros recursos adicionais.

A adoção desse asset possibilitou a criação de superfícies aquáticas visualmente imersivas, sem comprometer a performance do jogo. Isso foi possível graças à otimização nativa do *AQUA Lite*, que reduz a carga de processamento em comparação com soluções de água em alta fidelidade, atendendo assim às limitações de hardware comuns em jogos mobile, como demonstrado na Figura 19.

Figura 19 – Oceano criado com AQUALite



Fonte: Própria, 2025.

5.4 MECÂNICAS

Nesta seção, serão apresentadas e explicadas de forma sucinta as principais mecânicas implementadas no jogo, a saber: **movimentação do personagem, sistema de plantação e colheita, stamina, funcionamento dos portais e minigames** — *Ponto do Cálculo*, *Futemática*, *ImaginAção* e *Caverna das Palavras*. Cabe destacar que esta descrição tem caráter ilustrativo, buscando oferecer uma visão geral sobre o funcionamento de cada mecânica. Para uma compreensão mais aprofundada dos aspectos técnicos e de implementação, todo o código-fonte encontra-se disponibilizado no **repositório oficial do projeto no GitHub**.

5.4.1 MOVIMENTAÇÃO DO PERSONAGEM

A mecânica de movimentação do personagem constitui uma das principais funcionalidades de um jogo digital, visto que influencia diretamente a experiência do jogador e sua interação com o ambiente virtual. No caso de jogos desenvolvidos para plataformas mobile, essa mecânica requer atenção especial, pois a movimentação não pode ser excessivamente complexa, em razão das limitações de hardware e do espaço restrito da interface de toque. Conforme discutido anteriormente na seção de interface, não é recomendável sobrecarregar a tela com muitos comandos ou informações.

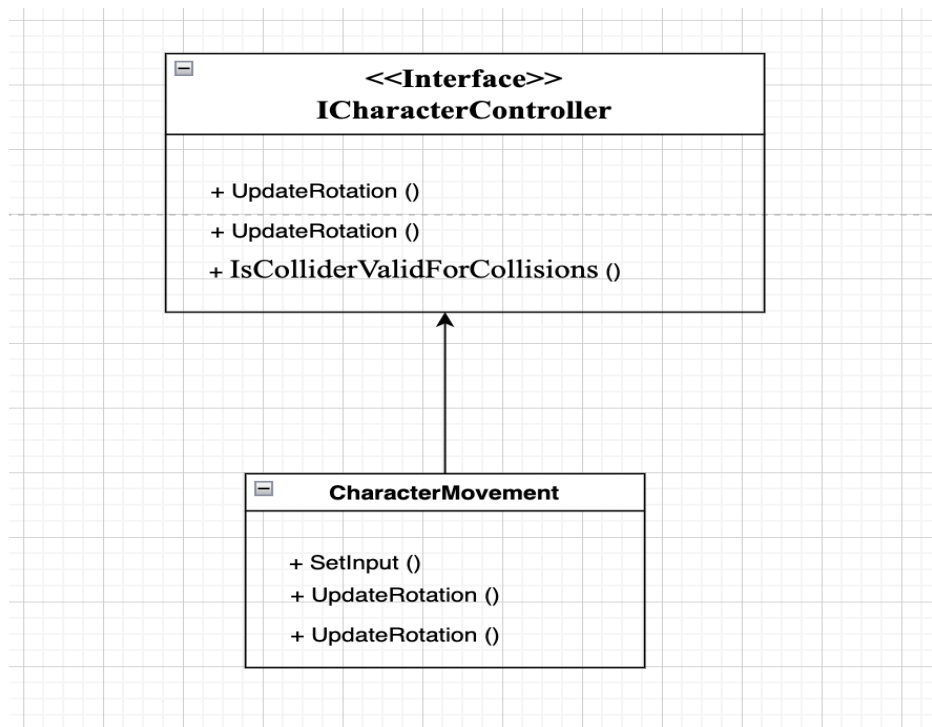
Dessa forma, a mecânica de movimentação foi inspirada no jogo *Genshin Impact*, adotando uma câmera posicionada acima do ombro do personagem, em uma distância previamente determinada. Esse sistema permite que o jogador se mova em todas as direções, ao mesmo tempo em que a câmera acompanha a rotação do personagem, proporcionando uma experiência fluida, intuitiva e condizente com as características de jogos em dispositivos móveis.

Para implementação da movimentação do personagem, foi utilizado um asset chamado *Kinematic Character Controller* (KCC), que é um recurso desenvolvido pela Unity que fornece uma solução de controle de personagens em jogos digitais. Diferente do uso tradicional de *rigid bodies* e da física padrão do motor unity, o KCC adota o algoritmo *collide and slide*, que permite movimentos mais fluidos, precisos e responsivos. Por ser um controlador cinemático, não reage automaticamente a forças externas, sendo necessário programar explicitamente tais interações (LIGHTNINGBOLT SOLUTIONS, 2025).

Entre suas principais características estão: colisões baseadas em cápsula, movimentação interpolada suavemente, alta performance sem alocações de *garbage collector*, suporte a degraus, rampas e cantos, integração com plataformas móveis e *rigidbodies* dinâmicos, além da possibilidade de aplicar gravidade arbitrária ou restringir o movimento a planos específicos (LIGHTNINGBOLT SOLUTIONS, 2025).

A aplicação ocorreu pela adição do componente *KinematicCharacterMotor* ao objeto do *CharacterMovement*, vinculado a uma classe que implemente a interface *ICharacterController*, a qual fornece callbacks como *UpdateVelocity*, *UpdateRotation* e *IsColliderValidForCollisions*, conforme demonstra a Figura 20.

Figura 20 – Integração do KinematicCharacterMotor



Fonte: Própria, 2025.

A função **UpdateVelocity** é responsável por integrar os inputs do jogador com o motor gráfico, controlando o movimento do personagem de acordo com a posição do joystick. Essa função verifica se o personagem está em contato com o chão, parando a movimentação quando não está, o que permite a implementação do salto, além de gerenciar a aceleração do personagem.

Já a função **UpdateRotation** controla a rotação do personagem, ajustando sua orientação conforme os comandos recebidos pelo joystick, garantindo que os movimentos e a direção do personagem sejam consistentes com a perspectiva da câmera e a intenção do jogador.

5.4.2 PLANTAÇÃO E COLHEITA

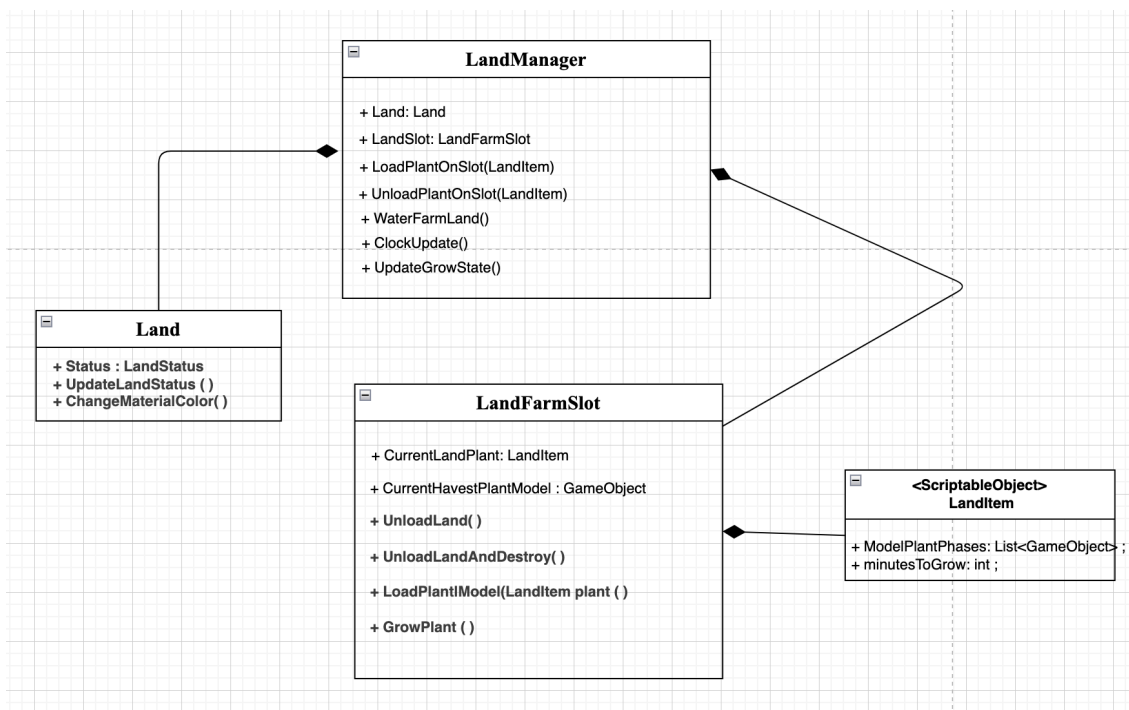
O sistema de plantação e colheita consiste em uma combinação de mecânicas que permitem ao personagem plantar, regar, acompanhar o crescimento das plantas e colher os frutos. Além disso, o sistema registra informações relevantes, como a verificação de irrigação e o tempo decorrido desde a última irrigação

O primeiro componente é o **LandManager**, responsável por gerenciar individualmente cada planta e controlar seu ciclo de crescimento. Cada **LandManager** possui um objeto denominado **LandFarmSlot**, que recebe o **ScriptableObject LandItem**. Este, por sua vez, contém todos os modelos da planta, representando suas diferentes fases de crescimento — desde a semente até o fruto —, bem como os intervalos de tempo entre cada estágio.

O **LandManager** também é responsável pelo gerenciamento do objeto **Land**, que controla o status do solo, indicando se este se encontra molhado, seco ou saudável, além de registrar o tempo desde a última irrigação. O **Land** possui ainda uma função que possibilita modificar o material do solo quando utilizado no modo *preview*, recurso que permite ao jogador visualizar a posição em que o solo será colocado ao interagir com a enxada. A Figura 21 ilustra de maneira simplificada a estrutura do componente **LandManager**, destacando a relação entre seus elementos: **LandFarmSlot**, **LandItem** e **Land**

Assim, a interação com o **LandManager** ocorre por meio de diferentes ferramentas que desempenham funções específicas. A enxada é responsável por criar o solo cultivável; a forquilha, por sua vez, permite remover esse solo. Já o pacote de sementes é utilizado para carregar a lista de modelos que serão instanciados no **LandFarmSlot**, definindo assim o ciclo de crescimento da planta. Por fim, o regador possibilita a irrigação do solo, garantindo sua manutenção adequada e influenciando diretamente no desenvolvimento da plantação.

Figura 21 – Estrutura do LandManager



Fonte: Própria, 2025.

Um componente fundamental para complementar o sistema de plantar e colher é a **classe RayManager**, que atua como um sistema centralizado de interação dentro do jogo. Sua principal função consiste em identificar os objetos com os quais o jogador pode interagir no cenário. Para cumprir essa função, o **RayManager** utiliza a técnica de detecção conhecida como *BoxCast*, que projeta uma caixa invisível a partir da posição do jogador. Esta caixa funciona como um sensor capaz de verificar a presença de objetos interativos em seu percurso, permitindo ao sistema determinar com precisão o ponto de impacto e o objeto selecionado.

O funcionamento do **RayManager** envolve a projeção contínua da caixa do *BoxCast* a cada quadro do jogo, garantindo que o sistema identifique corretamente a direção da mira do jogador e os objetos disponíveis para interação. Uma vez detectado um objeto, ele armazena as informações pertinentes, como a posição exata do impacto, permitindo que o jogo interprete de forma precisa onde o jogador está mirando, viabilizando a coleta de itens e a utilização de ferramentas. Além disso, a classe estabelece comunicação com o sistema de interface do usuário, exibindo ícones ou marcadores visuais que indicam a possibilidade de interação com os objetos

detectados. Esses elementos visuais funcionam como uma confirmação da mira, aprimorando a experiência do jogador ao fornecer feedback imediato sobre suas ações potenciais.

O **RayManager** também gerencia a interação com as ferramentas disponíveis, oferecendo pré-visualizações das ações antes que estas sejam efetivadas no cenário. Por exemplo, ao selecionar uma pá, o sistema apresenta a área em que o solo será aberto; ao utilizar um regador, indica o ponto que será irrigado. Quando a ação é confirmada pelo jogador, ele aplica o efeito correspondente de forma precisa sobre o objeto mirado.

Em síntese, a classe desempenha um papel de mediadora entre a mira do jogador e a interação com o mundo do jogo, simplificando a interpretação das intenções do usuário e garantindo que as ações sejam executadas de maneira eficiente e intuitiva.

5.4.3 ESTAMINA

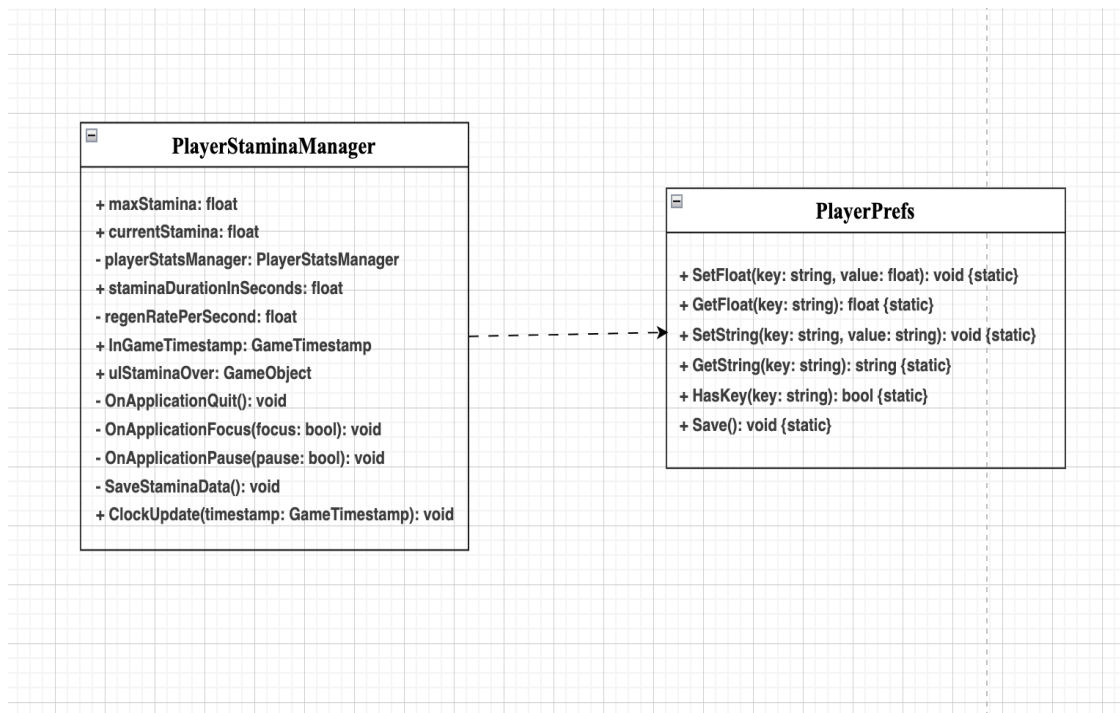
O componente **PlayerStaminaManager** é um componente de software desenvolvido para gerenciar a estamina (ou energia) do jogador. Sua função principal consiste em garantir que a estamina seja regenerada enquanto o jogador estiver fora do jogo e consumida a uma taxa predefinida durante a jogabilidade. Adicionalmente, o componente assegura a persistência do valor da estamina entre diferentes sessões, preservando o progresso do jogador.

Para garantir a persistência de dados, ele faz uso do sistema *PlayerPrefs*, oferecido pelo motor de jogo Unity, para armazenar e recuperar informações críticas, tais como a estamina atual e o registro de data e hora do último encerramento do aplicativo. Essa abordagem permite a manutenção do estado do jogador mesmo após o fechamento do jogo, caracterizando uma forma de persistência local de dados.

Na fase de inicialização do componente, o componente realiza a verificação da existência de valores previamente salvos. Na ausência desses dados, a estamina é automaticamente configurada para seu valor máximo, garantindo um estado inicial consistente. Posteriormente, o componente calcula o tempo decorrido desde a última sessão de jogo. Com base nesse intervalo, aplica-se um algoritmo de regeneração da estamina a uma taxa predefinida, simulando a recuperação contínua da energia do jogador mesmo em períodos nos quais o aplicativo não está ativo. A estamina regenerada é acumulada ao valor atual, respeitando o limite máximo permitido pelo design do

jogo. Durante a execução do jogo, a cada atualização do relógio interno, que será detalhado posteriormente, a estamina é decrementada de forma proporcional ao tempo de atividade do jogador. O componente também realiza o monitoramento do estado do aplicativo. Nos eventos de encerramento, perda de foco e pausa, os valores atuais de estamina, juntamente com os registros de data e hora, são salvos.

Figura 22 – Estrutura do PlayerStaminaManager



Fonte: Própria, 2025.

5.4.4 RELÓGIO INTERNO

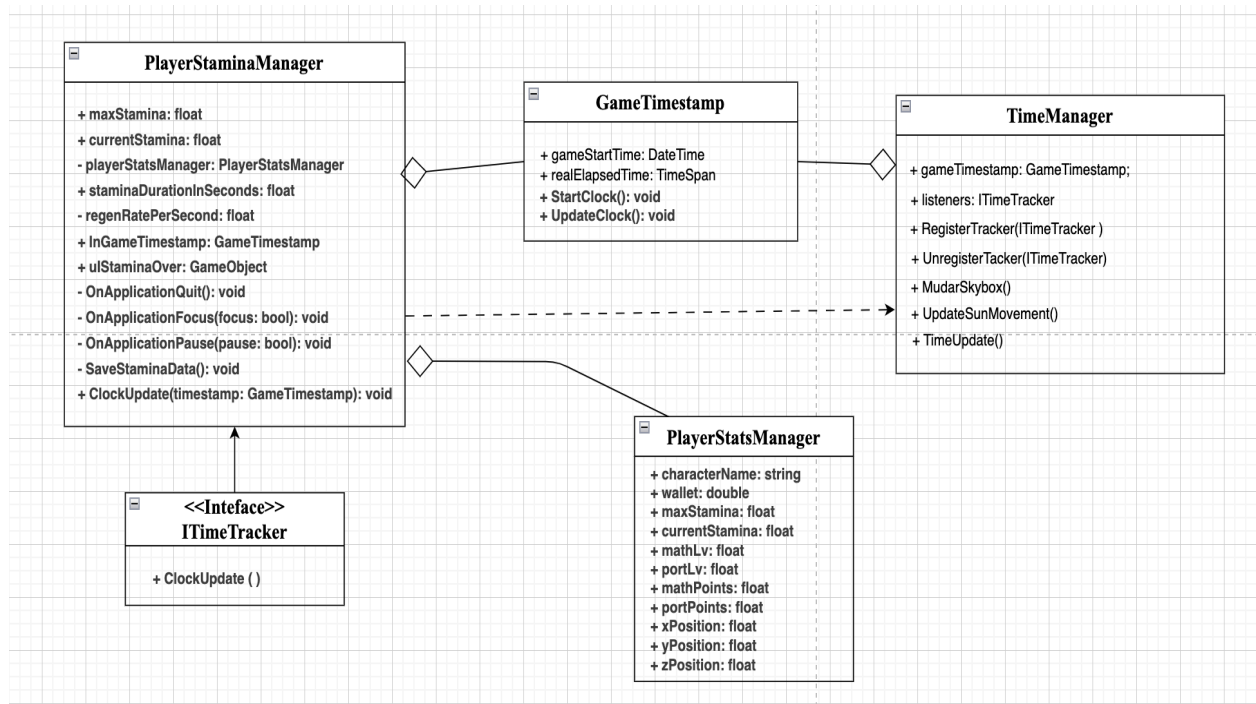
Os componentes, **ITimeTracker**, **GameTimestamp**, e **TimeManager**, constituem o sistema central de gerenciamento de tempo do projeto. Esses componentes foram projetados para sincronizar a passagem do tempo com o mundo real, de forma coesa e influenciando outros elementos da jogabilidade.

A interface **ITimeTracker** atua como um contrato, definindo um único método, *ClockUpdate*, com o objetivo de permitir que qualquer componente do jogo que necessite de atualizações periódicas de tempo se "inscreva" nesse sistema. Ao aderir a esta interface, o componente pode receber dados de tempo atualizados sem criar uma dependência direta e rígida

com a classe de gerenciamento principal, promovendo uma arquitetura modular e de baixo acoplamento. Assim, temos a classe **GameTimestamp** que é o principal objeto de dados que encapsula todas as informações relevantes sobre o tempo no jogo. Ela armazena e processa dados como a hora atual, o dia da semana, o mês e a estação do ano. A classe converte a hora atual do sistema em tempo de jogo e vice-versa.

A classe **TimeManager** constitui a espinha dorsal do sistema de tempo e é implementada como um singleton, assegurando que apenas uma instância esteja ativa durante toda a execução do jogo. Essa classe gerencia a simulação temporal por meio de uma corrotina, que periodicamente atualiza o objeto **GameTimestamp**, mantendo a evolução do tempo de forma contínua e sincronizada. Além disso, ela notifica todos os componentes registrados que implementam a interface **ITimeTracker**, invocando seus respectivos métodos **ClockUpdate** sempre que ocorre uma atualização temporal. Paralelamente, o **TimeManager** realiza a sincronização visual do jogo, ajustando dinamicamente elementos como a posição do sol e o material do céu, de modo a refletir de maneira precisa a passagem do tempo no mundo do jogo, reproduzindo alterações correspondentes ao dia, tarde ou noite.

Figura 23 – Uso do do TimeManager no PlayerStaminaManager



Fonte: Própria, 2025.

5.4.5 GERADOR ALEATÓRIO DE EXPRESSÕES MATEMÁTICAS

O componente **MathGen** atua como um utilitário central para a geração de desafios matemáticos, sendo empregado em minigames como **Futemática** e **Ponte do Cálculo**. Sua função principal consiste em produzir expressões matemáticas dinâmicas, de complexidade variável, ajustando-se a diferentes níveis de habilidade do jogador.

O **MathGen** foi estruturado como uma classe estática, o que possibilita que suas funcionalidades sejam acessadas diretamente, sem a necessidade de instanciação de objetos. Essa característica favorece a eficiência e simplifica a integração com outros módulos do sistema. A funcionalidade central da classe é o método *GerarExpressão*, responsável pela construção das equações matemáticas. Esse método recebe como parâmetros o conjunto de operações disponíveis, o número de operadores a serem utilizados, o valor máximo permitido para os números da expressão e o nível de dificuldade do desafio. A expressão é construída de forma incremental, por meio da seleção aleatória de números e operadores, dentro dos limites definidos. Um aspecto relevante desse processo é a inserção de parênteses, conforme o nível de dificuldade, o que amplia a complexidade do problema e exige do jogador a correta aplicação da ordem das operações matemáticas. O método retorna tanto a expressão completa em formato textual quanto uma lista contendo os números utilizados, possibilitando que outros componentes realizem validação e controle da resposta.

A resolução das expressões é realizada pelo método *Calculate*, que utiliza a classe *DataTable* do *framework* .NET para avaliar a expressão representada como string. Essa estratégia constitui uma solução prática e confiável, pois transforma a representação textual em um resultado numérico exato, o qual pode ser comparado diretamente com a resposta fornecida pelo jogador ou utilizada como entrada para as UI dos minigames.

5.4.6 MINIGAME FUTEMATEMÁTICA

O funcionamento do minigame **Futemática** é estruturado a partir da interação coordenada de três componentes principais, os quais, em conjunto, definem a lógica de progressão, as regras de interação e o sistema de *feedback* ao jogador. Esses elementos atuam de maneira integrada para assegurar uma experiência dinâmica, desafiadora e coerente. Para tanto, serão analisados os

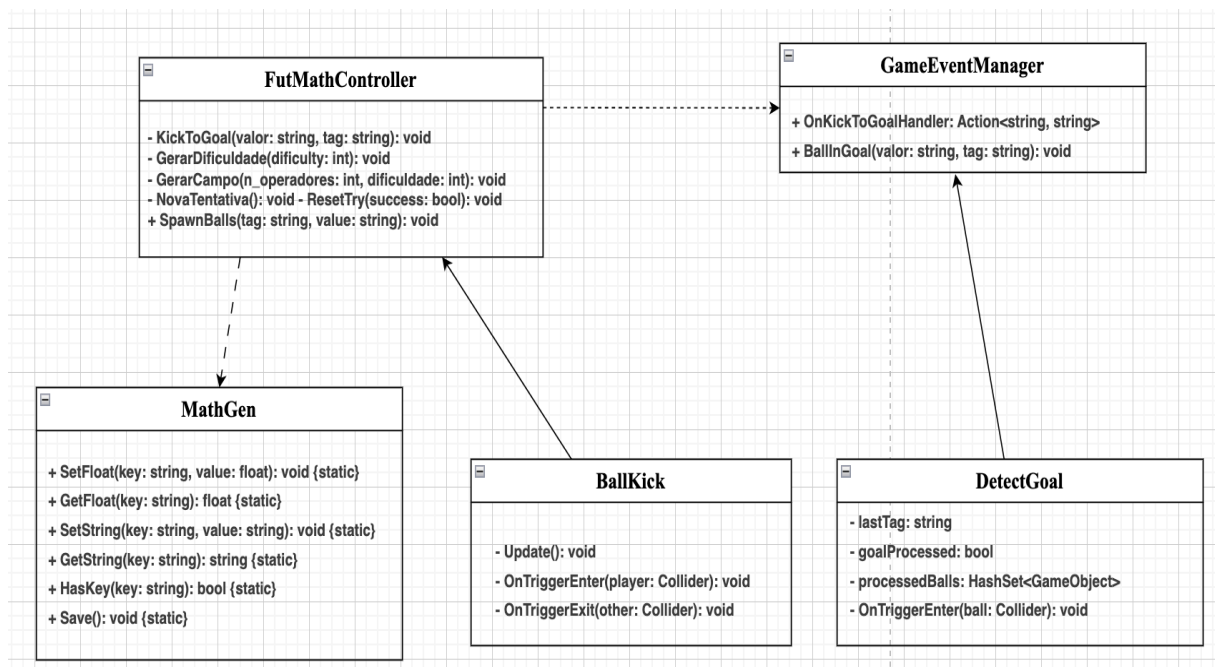
componentes **FutMathController**, **BallKick** e **DetectGoal**, que constituem as mecânicas do minigame, desde a geração dos desafios matemáticos até a validação das ações do jogador.

O **FutMathController** constitui o componente central, sendo responsável por orquestrar a lógica e o fluxo da jogabilidade. Sua principal atribuição consiste na geração dinâmica de desafios matemáticos, assegurando que a experiência do jogador seja progressiva e coerente. Inicialmente, a classe define o nível de dificuldade com base em dados persistentes de sessões anteriores, garantindo a continuidade da evolução do jogador. A partir dessa definição, uma expressão matemática aleatória é gerada por meio do utilitário **MathGen** — mencionado anteriormente na Seção 5.4.5 — e seu resultado é exibido em um painel visível no cenário, estabelecendo o objetivo a ser atingido durante a partida.

Para que o jogador resolva a equação, o sistema instância uma série de objetos de jogo — as bolas — que contêm números e operadores matemáticos. Essas bolas são posicionadas de forma aleatórias no campo de jogo, apresentando o desafio de selecionar e chutar os elementos corretos na ordem exata para replicar a expressão e atingir o resultado esperado. O componente monitora a interação do jogador, avaliando a expressão construída no gol a cada chute e comparando-a com o resultado pré-calculado. Com base nessa avaliação, o sistema concede um ponto por acerto ou subtrai uma vida por erro, fornecendo um *feedback* imediato e gerenciando o progresso do jogador dentro do minigame. A interação do jogador com as bolas é mediada pela classe **BallKick**. Este componente detecta a proximidade do jogador com uma bola e, ao receber o comando de chute, aplica uma força física, impulsionando o objeto em direção ao gol. Esse processo é acompanhado por *feedback* visual e sonoro, que aprimora a experiência do jogador ao confirmar a ação realizada.

Finalmente, o **DetectGoal** é o componente que atua como o ponto de recepção da interação. Acoplado ao objeto que representa o gol, ele detecta a colisão de uma bola e notifica o **FutMathController** sobre o evento. O componente garante que cada bola seja processada apenas uma vez, evitando erros lógicos e duplicidade de eventos, o que assegura a integridade da jogabilidade.

Figura 23 – Estrutura do futemática



Fonte: Própria, 2025.

5.4.7 MINIGAME PONTE DO CÁLCULO

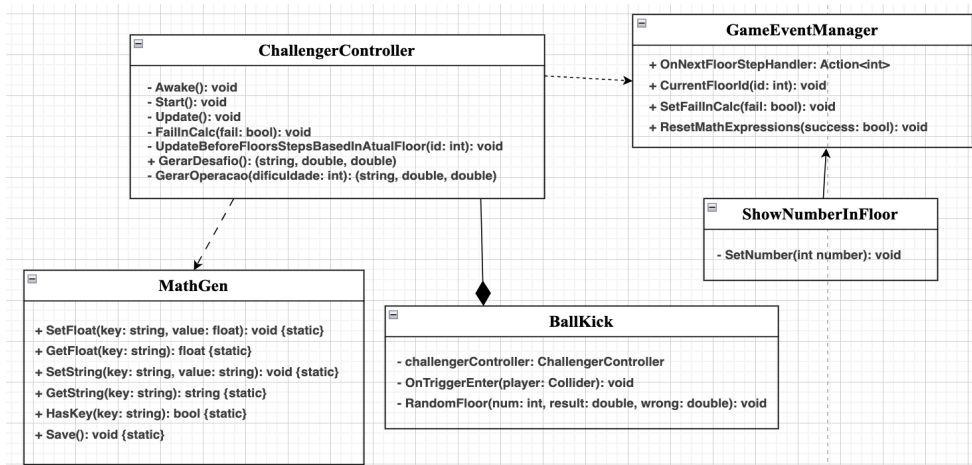
O núcleo da mecânica que constitui este minigame é formado por três componentes centrais, compreendidos como entidades interativas que, de maneira integrada, orquestram a experiência do usuário. Para tanto, serão analisados o **ChallengerController**, **GenerateChallenge** e **ShowNumberInFloor**, os quais desempenham papéis complementares na lógica de funcionamento do sistema de desafios.

O componente **ChallengerController** atua como o componente central do sistema de desafios, sendo responsável por gerenciar o estado do jogador e a progressão da dificuldade. Em sua fase inicial, conecta-se a eventos cruciais do jogo, como a detecção de queda do jogador ou a transição para a etapa seguinte do cenário. Além disso, recupera o nível de dificuldade armazenado em sessões anteriores, garantindo a persistência do progresso. Durante a execução, por meio do método `Update`, o controlador monitora constantemente a barra de vida do jogador. Quando as vidas se esgotam, o jogo é encerrado e o jogador é redirecionado ao mapa principal. Caso ocorra uma queda do cenário, uma vida é deduzida e um efeito sonoro é acionado, reforçando a consequência da ação e intensificando o feedback imersivo.

O método **GerarDesafio** constitui o núcleo da lógica de progressão. Ele invoca o procedimento **GerarOperacao**, que constrói expressões matemáticas adaptadas ao nível de dificuldade atual utilizando também o utilitário **MathGen**, no qual a complexidade é ajustada de forma incremental: a cada nível, novos operadores e operandos são introduzidos, ampliando gradualmente o desafio. O método garante a criação de um resultado correto e outro incorreto, assegurando que este último seja sempre distinto do correto, de modo a validar a proposta de escolha binária ao jogador nas plataformas. O **GenerateChallenge** funciona como o gatilho inicial do processo. Sua ativação ocorre quando o jogador entra em contato com uma área delimitada do cenário por meio de um Trigger. Nesse momento, o componente solicita ao **ChallengerController** o desafio recém-gerado — composto pela expressão matemática e por duas possíveis respostas. Em seguida, distribui a resposta correta assim como a incorreta de forma aleatória entre duas plataformas, utilizando o método **RandomFloor**. A aleatoriedade impede que o jogador memorize padrões de posição, exigindo que a solução seja efetivamente calculada. Após a apresentação do desafio, o componente desativa seus próprios *colliders* e os do objeto pai, prevenindo múltiplas ativações e garantindo a fluidez da experiência de jogo.

O componente **ShowNumberInFloor** desempenha um papel simples, porém fundamental, na comunicação visual com o jogador. Sua função é exibir valores numéricos nas plataformas. Ele recebe do **GenerateChallenge** o número correspondente (correto ou incorreto) e atualiza dinamicamente o texto associado ao objeto, assegurando que a informação esteja sempre visível e alinhada ao desafio em andamento, como descreve a Figura 23.

Figura 23 – Estrutura do ponte do cálculo



Fonte: Própria, 2025.

5.4.8 MINIGAME CAVERNA DAS PALAVRAS

O funcionamento desse minigame, assim como os anteriores, é organizado a partir da cooperação entre os diferentes componentes, que, em conjunto, estabelecem as regras de progressão, controlam as interações do jogador e fornecem os mecanismos de retorno visual. Nesse contexto, serão descritos **ForcaController**, **OnPressTermoButtonChallenge** e **EnableGlowButton**, que abrangem desde a definição das palavras e monitoramento das tentativas até a apresentação do *feedback* ao usuário.

O componente **ForcaController** desempenha o papel de controlador central do minigame. Sua função consiste em gerenciar o estado do jogador e a progressão da partida, atuando como o componente responsável por regular a dificuldade, o número de vidas e as interações do jogador com o ambiente. Na sua inicialização, o controlador recupera o nível de dificuldade previamente salvo, assegurando a continuidade do progresso do jogador. A mecânica principal é iniciada no método **IniciarLevel**, que seleciona uma palavra aleatória a partir de um arquivo de texto adequado ao nível de dificuldade, ofusca a palavra para exibição na interface e define o limite de erros permitidos. Em seguida, o método **InitButtons** assegura a adaptação da jogabilidade ao nível de dificuldade configurado. Esse processo inicia-se preenchendo os botões com as letras ou sílabas que compõem a palavra-desafio, garantindo que os elementos necessários para a resolução estejam disponíveis ao jogador. Após, os demais botões são completados de forma aleatória com letras ou sílabas distintas, introduzindo variabilidade ao desafio e aumentando o nível de complexidade da atividade. Durante a sua execução, o componente **ForcaController** monitora constantemente o número de tentativas incorretas. Caso o limite de erros seja atingido, o nível é reiniciado, uma vida é deduzida e a interface de vidas é atualizada. Se as vidas chegarem a zero, a partida é encerrada e o jogador retorna ao mapa principal. Em contrapartida, quando a palavra é completada corretamente, um *GameObject* representando a chave é ativado, sinalizando a conclusão bem-sucedida do desafio.

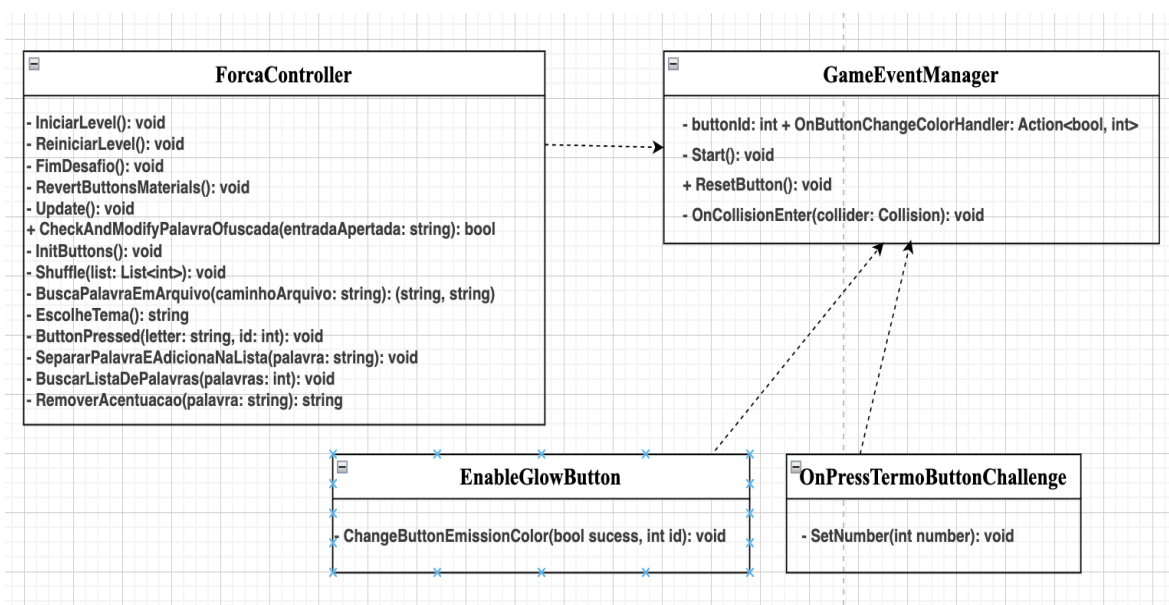
O método **CheckAndModifyPalavraOfuscada** representa o núcleo da mecânica de progressão. Ele processa a entrada do jogador (letra ou sílaba) quando ele pisa em um botão e verifica se corresponde a algum caractere da palavra-desafio. Em caso positivo, a palavra ofuscada que utiliza o caractere especial “■”, é substituída nas posições corretas, e os demais componentes do jogo são notificados. Já o método **ButtonPressed** registra qual botão foi

pressionado e qual letra ou sílaba ele contém, em seguida comunica os resultados aos componentes vinculados.

O Componente **OnPressTermoButtonChallenge** gerencia a interação direta do jogador com os botões que contêm letras ou sílabas. Esse componente detecta a colisão do jogador com o botão e, ao ser acionado, simula o clique por meio da movimentação do objeto, acionando visualmente sua função. Em seguida, ele notifica o **GameEventManager**, que repassa a informação ao **ForcaController** para validação da jogada. Além disso, o método **ResetButton** restaura a posição e a aparência original do botão, preparando-o para novas interações. Essa reinicialização garante a repetição consistente do desafio, permitindo que o jogador continue tentando até completar ou falhar no nível.

O **EnableGlowButton** tem como função fornecer feedback visual imediato ao jogador. Ele permanece em escuta dos eventos disparados pelo **GameEventManager**, identificando se a jogada foi correta ou incorreta. A partir dessa informação, altera a cor e ativa o brilho do material do botão. No caso de acertos, o botão adquire a cor azul, sinalizando sucesso; em caso de erros, a cor exibida é vermelha. Esse recurso visual desempenha papel fundamental na experiência do usuário, pois reforça os resultados da ação sem depender exclusivamente de textos explicativos, oferecendo clareza, imersão e responsividade às interações realizadas.

Figura 24 – Estrutura do caverna das palavras



Fonte: Própria, 2025.

5.4.9 MINIGAME INTERPRETAÇÃO

O funcionamento do minigame é sustentado pela interação coordenada de dois componentes principais, que em conjunto definem a lógica de progressão, a validação das ações do jogador e o sistema de feedback imediato. Esses elementos, quando integrados, asseguram uma experiência interativa dinâmica, desafiadora e coerente, fundamentada na interpretação de textos narrativos e na coleta de objetos em sequência predeterminada. Para tanto, serão analisados os componentes **InterpretationChallengeController** e **TreasureObject**, que constituem os principais componentes da mecânica, abrangendo desde a geração dos desafios de interpretação até a resposta do sistema diante das escolhas realizadas pelo jogador.

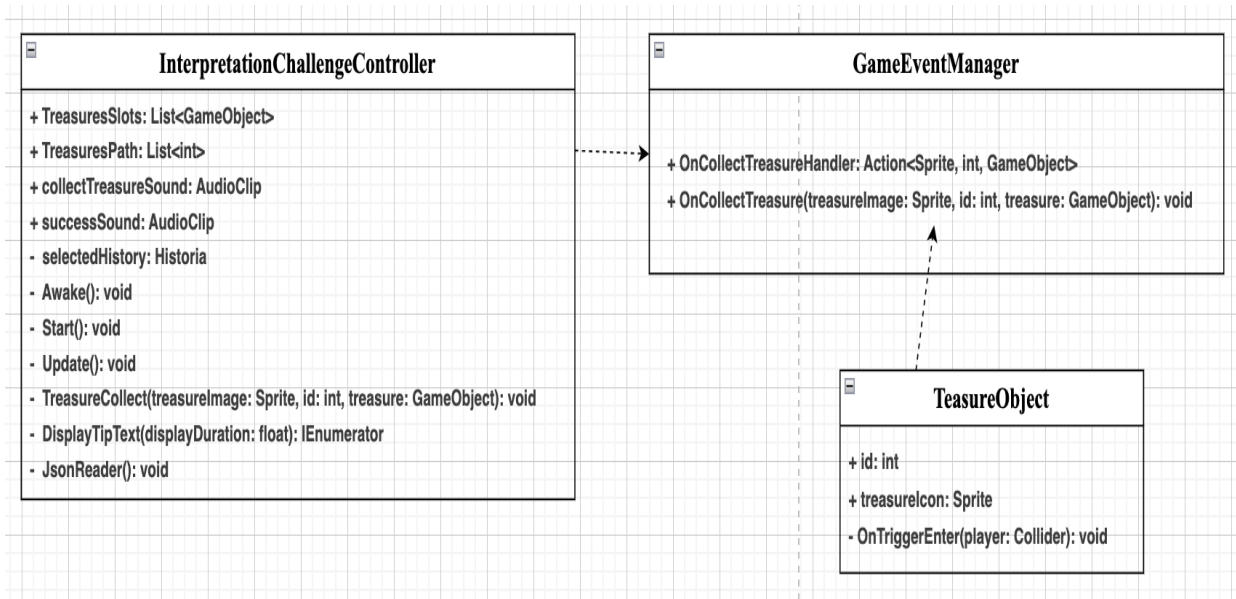
O componente **InterpretationChallengeController** pode ser compreendido como o “cérebro” do minigame. Sua função central é gerenciar o estado do desafio, acompanhar a progressão do jogador e articular a interface com os elementos narrativos do jogo. Ao inicializar, o componente faz a leitura e desserialização de um arquivo JSON, no qual estão armazenadas as histórias, suas respectivas dicas e a ordem correta dos tesouros a serem coletados. A seleção de uma história ocorre de forma aleatória, baseado no nível do jogador nesse desafio, de modo a garantir progressão e imprevisibilidade na experiência de cada sessão. Durante a execução, ele monitora as ações do jogador, sendo possível detectar quando o jogador deseja consultar o painel de história, exibindo ou ocultando o texto e as dicas associadas. Além disso, o **InterpretationChallengeController** é responsável por validar continuamente o progresso na coleta dos tesouros. Quando todos os itens são recolhidos na sequência correta, o sistema disponibiliza uma chave especial, para que o jogador se dirija ao castelo onde realizará a conclusão bem-sucedida do desafio e sinalizando a transição para a próxima etapa do jogo.

O método **TreasureCollect** realiza a lógica de validação. Ao acionar a coleta de um tesouro, ele verifica se o objeto corresponde ao próximo item previsto pela narrativa. Caso positivo, o tesouro é registrado como coletado, a interface gráfica é atualizada com o ícone correspondente e a parte da história associada é marcada como concluída. Por outro lado, se o jogador recolher um item fora da ordem correta, o sistema ativa um mecanismo de *feedback* negativo: uma dica visual temporária, exibida no UI, orientando o jogador sobre o próximo tesouro que deveria ser selecionado, assim exibindo dicas predefinidas no arquivo narrativo, funcionando como suporte ao aprendizado e à correção de erros de maneira não punitiva, mas

orientadora. O **TreasureObject** representa a materialização dos elementos colecionáveis do jogo, funcionando como a interface física de interação entre o jogador e a narrativa. A cada tesouro é atribuído um objeto de jogo independente, ao qual o mesmo é anexado. Sua principal função é detectar colisões entre o jogador e o item no cenário. Ao ser ativado, o componente envia para o **GameEventManager** informações essenciais, como a identificação única (ID) do tesouro, sua imagem de ícone e a referência ao objeto físico coletado.

Esses dados são, então, utilizados pelo **InterpretationChallengeController** para validar a sequência de coleta. Dessa forma, embora seja um componente relativamente simples, o **TreasureObject** possui papel crucial: ele garante a modularidade e a independência dos itens de jogo, permitindo que cada tesouro seja tratado como uma entidade interativa autônoma. Essa arquitetura contribui para a escalabilidade do sistema, possibilitando a inserção de novos tesouros e histórias sem comprometer a coesão do jogo. A Figura 25 ilustra essa estrutura.

Figura 25 – Estrutura do interpretAção



Fonte: Própria, 2025.

5.4.10 ENTRADA DOS DESAFIOS E APRENDIZADO ESPAÇADO

O componente faz o gerenciamento da entrada do personagem em desafios e faz o uso de aprendizado espaçado para aprimorar o aprendizado. A implementação se baseia em um algoritmo de balanceamento de carga conhecido como Round Robin Ponderado, que permite a

seleção de cenas de forma inteligente, dando preferência a aquelas com maior "peso", ou seja, aquelas que precisam ser revisadas com mais frequência.

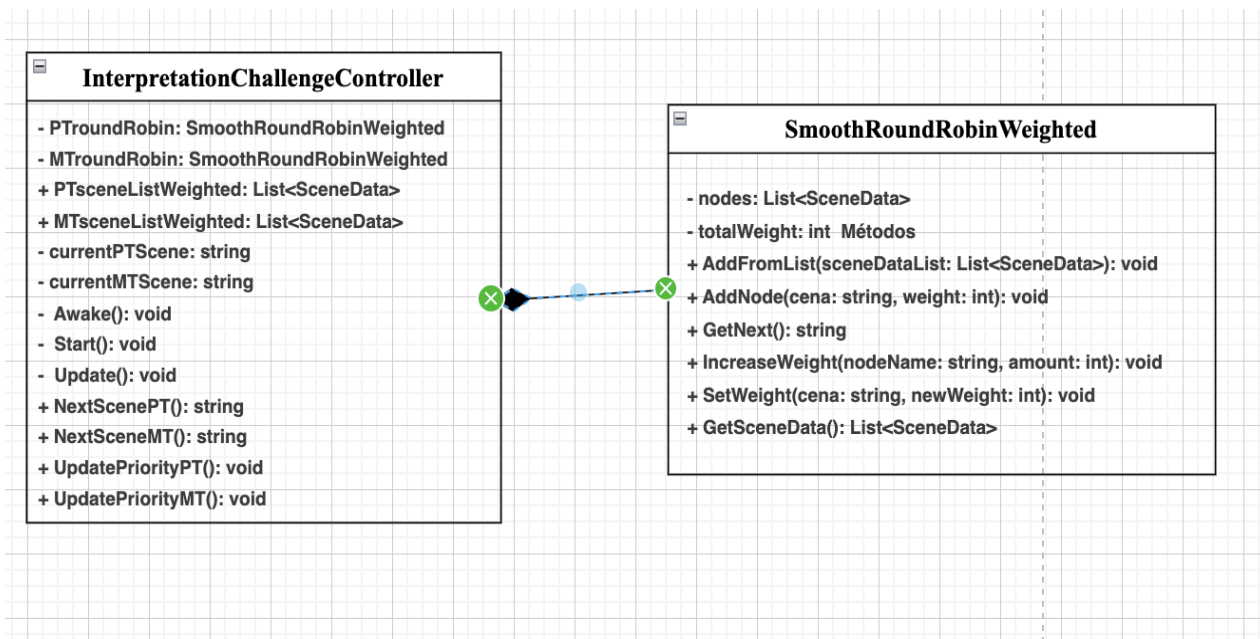
A classe **EnterChallengesManager** atua como um gerenciador central, utilizando o padrão de projeto Singleton para garantir que apenas uma instância exista em toda a aplicação. A classe contém campos públicos para a interface do usuário da tela de carregamento, como um painel e um campo de texto, permitindo que a transição entre cenas seja visualmente representada. Há também campos para armazenar as instâncias dos algoritmos de Round Robin Ponderado, um para os desafios de português e outro para os de matemática, além das listas de cenas que serão usadas para alimentar esses algoritmos. Os campos **currentPTScene** e **currentMTScene** registram a cena que está sendo exibida no momento. O método **Awake** é responsável por implementar o padrão *Singleton*, verificando se já existe uma instância do gerenciador. Se não houver, ele a estabelece e a mantém persistente durante a transição entre cenas com o método **DontDestroyOnLoad**, garantindo que as listas de desafios e o estado do sistema não sejam perdidos. O método **Start** inicializa as listas de cenas dos algoritmos, a menos que dados de salvamento prévios já existam, o que indica que o usuário está retornando ao jogo.

Os métodos **NextScenePT** e **NextSceneMT** são as funções-chave que interagem com os algoritmos de Round Robin. Eles invocam o método **GetNext** do algoritmo correspondente para obter o nome da próxima cena a ser carregada, baseando-se no seu peso, e atualizam a lista de cenas para refletir essa escolha. Esses métodos são acessíveis publicamente para serem chamados por outras classes que precisam iniciar um novo desafio. Para o conceito de aprendizado espaçado, os métodos **UpdatePriorityPT** e **UpdatePriorityMT** são essenciais. Eles permitem que o peso de uma cena seja ajustado dinamicamente, aumentando sua prioridade. Nesse contexto, significa que se o aluno cometeu erros em um desafio, o peso dessa cena pode ser aumentado para que ela seja selecionada novamente mais cedo, promovendo a repetição e a fixação do conteúdo.

A classe **SmoothRoundRobinWeighted** é a implementação do algoritmo do Round Robin ponderado. Ela gerencia uma coleção de objetos do tipo **SceneData**, que contêm o nome da cena e o seu peso. O peso é um valor que determina a "frequência" com que a cena deve ser selecionada. Cenas com peso maior têm mais chances de serem escolhidas.

O método **AddFromList** popula a lista de cenas do algoritmo a partir de uma lista externa. O método **GetNext** é a parte central do algoritmo. Em cada chamada, ele itera por todas as cenas, incrementando um "peso atual" de cada cena com seu peso original. A cena com o maior peso atual é então selecionada. Após a seleção, o peso atual da cena escolhida é reduzido pelo peso total de todas as cenas, garantindo que ela não seja selecionada imediatamente na próxima rodada, mas que o balanceamento ponderado seja mantido suavemente ao longo do tempo. Esse mecanismo é o que permite a seleção inteligente e a distribuição desigual das cenas ao longo do tempo. Os métodos **IncreaseWeight** e **SetWeight** são as funções que permitem a manipulação direta do peso de uma cena, essenciais para a aplicação do aprendizado espaçado. Por exemplo, se um aluno erra uma questão de matemática, a cena correspondente pode ter seu peso aumentado para que o desafio seja reapresentado mais cedo. O método **GetSceneData** retorna a lista de cenas e seus pesos atuais, o que é útil para salvar o estado do jogo

Figura 26 – Interação entre EnterChallengesManager e SmoothRoundRobinWeighted



Fonte: Própria, 2025.

5.5 PERSISTÊNCIA DE DADOS

A persistência de dados constitui um elemento essencial do desenvolvimento de jogos digitais, pois permite ao sistema armazenar e recuperar informações relacionadas ao progresso do jogador, ao estado do ambiente e à configuração de objetos interativos. Essa funcionalidade

assegura que os avanços realizados pelo usuário não sejam perdidos entre sessões de jogo, garantindo continuidade, consistência e uma experiência mais imersiva. Com isso em mente, a seguir será apresentada a arquitetura e a implementação do sistema de gerenciamento do progresso de jogo utilizado neste projeto, estruturado a partir de três componentes principais: o gerenciador de salvamento global, o gerenciador de terrenos e a classe auxiliar destinada à leitura e escrita de arquivos.

A classe **WorldSaveGameManager** atua como o ponto de controle central para as operações de salvamento, carregamento e exclusão do progresso do jogador. Sua implementação segue o padrão de projeto *Singleton*, garantindo que apenas uma instância esteja ativa durante a execução do jogo. O gerenciador mantém referências cruciais, como a instância do controlador do jogador e da classe **SaveDataWriter**, responsável pelas operações de entrada e saída de dados.

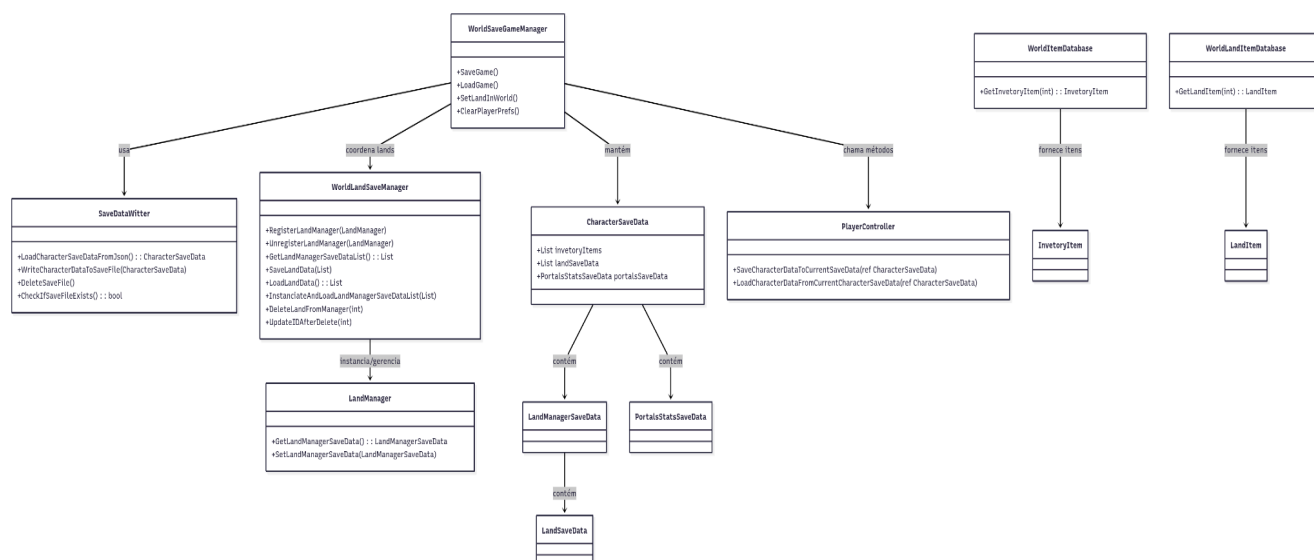
A estrutura de dados do jogador, denominada **CharacterSaveData**, é armazenada de modo a permitir sua manipulação e persistência. O método *SaveGame* coordena o processo de salvamento, inicializando o escritor de dados, definindo o caminho e o nome do arquivo e invocando o método do controlador do jogador que coleta todas as informações relevantes, armazenadas em *currentCharacterSaveData*, que posteriormente são gravadas no arquivo. De forma complementar, o método *LoadGame* recupera o estado do jogo, carregando os dados armazenados e aplicando-os ao controlador do jogador, restaurando as propriedades do personagem.

WorldLandSaveManager é uma classe dedicada à persistência de dados do mundo do jogo, especialmente dos terrenos cultiváveis. Assim como o **WorldSaveGameManager**, ele também adota o padrão *Singleton*, garantindo controle exclusivo sobre o estado de todos os terrenos. O método *InstantiateAndLoadLandManagerSaveDataList* constitui o núcleo da funcionalidade de carregamento, ao iterar sobre uma coleção de dados de salvamento de terrenos, instanciando novos objetos correspondentes e aplicando os dados previamente armazenados, como posição e estado da plantação, restaurando a configuração anterior do ambiente. A classe ainda fornece mecanismos para registro e exclusão de gerenciadores de terreno, essenciais para a administração da coleção de terrenos ativos, enquanto os métodos *GetLandManagerSaveDataList* e *SaveLandData* permitem coletar e persistir os dados de todos os terrenos conforme a preferência do jogador. O sistema também incorpora bases de dados, representadas pelas classes

WorldLandItemDatabase e **WorldItemDatabase**, implementadas como Singletons. Estas classes funcionam como catálogos de itens, itens esses que foram implementados como *Scriptable Objects*, possibilitando que outros componentes do jogo acessem informações sobre itens de terreno ou de inventário por meio de identificadores únicos, promovendo consistência e facilitando a manutenção dos dados.

Por fim, a classe **SaveDataWitter** desempenha papel fundamental na manipulação de arquivos de salvamento, encapsulando a lógica de leitura e escrita em formato JSON, com base nos diretórios definidos pelo gerenciador principal. Seus métodos, como **LoadCharacterSaveDataFromJson** e **WriteCharacterDataToSaveFile**, abstraem a complexidade do manuseio de arquivos e do tratamento de exceções, oferecendo uma interface segura e eficiente para a persistência de dados. A classe também fornece funcionalidades de verificação e exclusão de arquivos de salvamento.

Figura 27 – Sistema de gerenciamento de salvamento/carregamento de jogo



Fonte: Própria, 2025.

5.6 EFEITOS SONOROS

A importância dos efeitos sonoros em jogos sérios educativos reside em sua capacidade de aprimorar a experiência de aprendizagem por meio de estímulos auditivos que reforçam comportamentos desejados, sinalizam eventos e criam atmosferas imersivas. Estudos indicam que o uso adequado de áudio pode aumentar a retenção de informações, melhorar a concentração e

proporcionar feedback imediato ao jogador, facilitando a compreensão de ações e consequências dentro do jogo (RODRIGUES, 2016).

Os efeitos sonoros desempenham um papel relevante na interação do jogador com os conteúdos educacionais. O uso de sons como resposta a ações e eventos no jogo contribui para a navegação e a compreensão do ambiente virtual, favorecendo uma experiência mais acessível e promovendo uma aprendizagem mais equitativa (OLIVEIRA et al., 2021).

Neste contexto, a seguir abordaremos brevemente a implementação e a análise dos efeitos sonoros no jogo desenvolvido para este projeto.

5.6.1 FONTE DE DADOS

Para o desenvolvimento deste projeto, foram utilizados diversos bancos de áudio, entre os quais se destacam Freesound.com, HookSounds.com e Sounds-Resource.com. Todos esses repositórios oferecem efeitos sonoros de uso livre, permitindo sua integração sem restrições legais. Destaca-se, em especial, o Sounds-Resource.com, que disponibiliza uma wiki com efeitos sonoros de jogos consagrados, alguns também de livre utilização, oferecendo uma referência valiosa para a criação de experiências sonoras autênticas e imersivas.

5.6.2 IMPLEMENTAÇÃO

A implementação dos efeitos sonoros neste projeto, explicada de maneira simplificada, foi realizada por meio da classe **UIController**, responsável pelo gerenciamento da interface do usuário no jogo, atuando também como uma classe estática que centraliza a reprodução de efeitos sonoros. Nesse contexto, a mesma possui um campo público do tipo *AudioSource*, que funciona como o componente central para a emissão de sons.

O método público *PlayUIFx* é o principal responsável pela execução dos efeitos sonoros. Ele recebe como parâmetro um *AudioClip* e aciona a função *PlayOneShot*, uma função nativa da Unity, que reproduz o áudio especificado sem interromper sons previamente em execução. Dessa forma, a utilização do *PlayOneShot* permite que efeitos sonoros pontuais, como o clique de um botão, sejam reproduzidos simultaneamente à música de fundo, garantindo uma experiência auditiva contínua, coerente e imersiva para o jogador. O método *PlayUIFx* é invocado a partir de

outros componentes associados a elementos da interface, incluindo botões, ações, animações do personagem e sons de ambientação, acionando a reprodução de clipes de áudio específicos para cada interação.

Além disso, a **UIController** inclui métodos voltados à gestão da música de fundo do jogo, como *PlayMapMusic* e *PlayMainMapMusics*, que se comunicam com o componente **GameMusicManager** para controlar a execução das trilhas sonoras. O método *StopAllSounds*, permite interromper toda a reprodução de áudio por meio de uma função correspondente no *GameMusicManager*. Essa implementação foi desenvolvida visando estabelecer uma clara separação entre os efeitos sonoros da interface, caracterizados por sua curta duração e pontualidade, e a música de fundo, gerenciada por uma classe dedicada, evidenciando uma organização eficaz e modular da arquitetura de áudio do projeto.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o processo de desenvolvimento do jogo mobile Colheita do Saber, desde a etapa de planejamento e elaboração de level design até sua implementação prática. O objetivo central do projeto consistiu em criar um jogo mobile para que seja usado como recurso educacional para incentivar a aprendizagem de crianças e adolescentes no contexto educacional brasileiro. O jogo foi estruturado com foco em conteúdos de Matemática, Língua Portuguesa — abordando tanto a construção de vocabulário quanto a interpretação de textos — e Língua Inglesa, contemplando igualmente vocabulário e compreensão textual.

Para a concepção do jogo, foi necessário o aprofundamento em teorias da educação, no estudo de tecnologias aplicadas ao ensino-aprendizagem e em fundamentos de desenvolvimento de software. Nesse sentido, a proposta buscou integrar elementos educacionais, sobretudo relacionados aos processos de alfabetização e de aprendizado básico, com referenciais teóricos da área pedagógica. Nesse sentido, o objetivo pedagógico principal foi buscar explorar a ludicidade de jogos como estratégia de ensino, utilizando a diversão como recurso para despertar o interesse dos estudantes, promovendo, de forma prática e acessível, a fixação de conceitos essenciais.

Como recurso motivacional, foi desenvolvido um sistema de recompensas, baseado em elementos de gamificação, que inclui novas plantas, aquisição de ferramentas, moedas virtuais e progressão vinculada ao desempenho nos conteúdos de matemática e línguas. Além disso, foi

implementado um sistema de sincronismo em tempo real, com o propósito de limitar o tempo de uso do jogo, mesmo em se tratando de um recurso educacional, evitando, assim, a exposição excessiva às telas.

Os testes realizados durante o desenvolvimento do jogo tiveram como foco principal a análise de usabilidade e estabilidade do sistema, sendo aplicados de forma preliminar apenas com familiares, sem aprofundamento em avaliações voltadas à mensuração direta do impacto no processo de aprendizagem. Tal delimitação se justifica pelo fato de que, embora se trate de um jogo educacional, o Colheita do Saber não tem como objetivo substituir a sala de aula, mas sim atuar como recurso complementar, oferecendo suporte a pais, professores e alunos. Nesse sentido, foram incorporados elementos de gamificação e de jogos sérios como recursos educacionais, já apresentados na Seção 3.2 e 3.5, respectivamente, como estratégia para aprimorar a experiência do usuário, estimulando o engajamento, a motivação e, conseqüentemente, buscar contribuir no processo de aprendizagem.

Para trabalhos futuros, diversas melhorias podem ser implementadas com o intuito de expandir e aprimorar a experiência do jogador. Entre elas, destaca-se a inclusão de novos conteúdos e disciplinas, como Geografia, História e Física, ampliando o escopo educacional do jogo. Também se sugere o desenvolvimento de tutoriais interativos voltados ao ensino da mecânica dos minigames, a inserção de uma narrativa que auxilie na imersão do jogador no universo do jogo e a implementação de interações com NPCs distribuídos pelo ambiente virtual, favorecendo uma dinâmica mais envolvente.

Outra possibilidade de aprimoramento consiste na utilização de dados de telemetria em conjunto com técnicas de Machine Learning e Inteligência Artificial, a fim de personalizar recomendações pedagógicas e adaptar os conteúdos dos minigames ao perfil e às necessidades específicas de cada jogador, reforçando a proposta de aprendizagem adaptativa. Sugere-se, ainda, a realização de estudos de caso em ambientes escolares e com crianças com necessidades educacionais específicas, o que poderia fornecer dados relevantes tanto para o aperfeiçoamento da usabilidade quanto para a avaliação da eficácia do jogo no processo de aprendizagem.

7 BIBLIOGRAFIA

AGÊNCIA BRASIL. Taxa de alfabetização era de 49,3% no Brasil em 2023, diz Inep. Brasília: Empresa Brasil de Comunicação, 16 abr. 2025. Disponível em: <https://agenciabrasil.ebc.com.br/educacao/noticia/2025-04/taxa-de-alfabetizacao-era-de-493-no-brasil-em-2023-diz-inep>. Acesso em: 3 ago. 2025.

ASSESSORIA DE COMUNICAÇÃO SOCIAL DO INEP. Inep publica relatório da pesquisa Alfabetiza Brasil. 2023. Disponível em: <https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/alfabetiza-brasil>. Acesso em: 02 fev. 2025.

AUDACITY(R) | Free, open source, cross-platform audio software for multi-track recording and editing. Disponível em: <https://www.audacityteam.org/FAQ/>. Acesso em: 14 fev. 2025.

BALASUBRAMANIAN, K. Game engines: all you need to know. Disponível em: <https://www.gameopedia.com/>. Acesso em: 08 fev. 2025.

BATTAIOLA, A. L. Jogos por computador – histórico, relevância tecnológica e mercadológica, tendências e técnicas de implementação. In: ANAIS DA XIX JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA (JAI), 2000. Proceedings., Santa Rita do Sapucaí-MG, 2000, p. 83-122. Acesso em: 04 fev. 2025.

BIDERMAN, M. T. C. Léxico e vocabulário fundamental. ALFA: Revista de Linguística, São Paulo, v. 40, 2001. Disponível em: <https://periodicos.fclar.unesp.br/>. Acesso em: 18 fev. 2025.

BLUMBERG, F.C., & ISMAILER, S. S. (2009). Chapter 9: What do Children Learn from Playing Digital Games? In U. Ritterfeld, M. Cody, & P. Vorderer (Eds.). Serious Games Mechanisms and Effects (131-142). New York and London: Routledge. Acesso em: 21 fev. 2025.

BRASIL. Lei nº 13.005, de 25 de junho de 2014. Plano Nacional de Educação 2014-2024. Diário Oficial da União, Brasília, DF, 26 jun. 2014.

BRASIL. Lei nº 15.100, de 13 de janeiro de 2025. Dispõe sobre a utilização, por estudantes, de aparelhos eletrônicos portáteis pessoais nos estabelecimentos públicos e privados de ensino da educação básica. Diário Oficial da União: seção 1, Brasília, DF, 13 jan. 2025.

BRASIL. Conselho Nacional de Educação (CNE). Resolução CNE/CEB nº 2/2025, de 21 de março de 2025. Diretrizes Operacionais Nacionais sobre o uso de dispositivos digitais em espaços escolares e integração curricular da educação digital e midiática. Diário Oficial da União, Brasília, DF, 24 mar. 2025.

BURKE, B. Gamificar: como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DVS Editora, 2015. Acesso em: 18 fev. 2025.

CHARSKY, D. (2010). From Edutainment to Serious Games: A Change in the Use of Game Characteristics. *Games and Culture* 5(2), 177-198. Doi: 10.1177/1555412009354727. Acesso em: 21 fev. 2025.

CHRISTOPOULOU, Eleftheria; XINO GALOS, Stelios. Overview and comparative analysis of game engines for desktop and mobile devices. *International Journal of Serious Games*, v. 4, n. 4, p. 21-36, 14 dez. 2017. Disponível em: <https://journal.seriousgamessociety.org/index.php/IJSG/about>. Acesso em: 09 fev. 2025.

CHRISTOPOULOU, A, MYSTAKIDIS, S. Gamification in Education. *Encyclopedia*. 2023; 3(4):1223-1243. <https://doi.org/10.3390/encyclopedia3040089>. Acesso em: 16 fev. 2025.

CHRISTOPOULOS, A.; MYSTAKIDIS, S.; CACHAFEIRO, E.; LAAKSO, M.-J. Escapando da célula: salas de escape de realidade virtual na educação em biologia. *Comportamento. Inf. Tecnologia*. 2023, 42, 1434–1451. <https://doi.org/10.1080/0144929X.2022.2079560>. Acesso em: 15 fev. 2025.

CORTI, K. (2006) Games-based Learning; a serious business application. PIXELearning Limited. <https://tapggc.org/projects/2024/Fall/PixelArcade/> . Acesso em: 21 fev. 2025.

COSTA, Kaio Rezende Nunes. Personal Scrum: uma alternativa ágil para desenvolvimento de indie games. 2016. Trabalho de Conclusão de Curso (Graduação em Engenharia de Software) – Universidade Federal do Pampa, Campus Alegrete, Alegrete, 2016. Acesso em: 04 fev. 2025.

COSTER, I. D. et al. Teaching Reading in Europe: Contexts, Policies and Practices. Disponível em: https://www.researchgate.net/publication/303022099_Teaching_Reading_in_Europe_Contexts_Policies_and_Practices. Acesso em: 19 fev. 2025.

DETERDING, S. A ambiguidade dos jogos: histórias e discursos de um mundo lúdico. Em *O mundo lúdico* ; Walz, SP, Deterding, S., Eds.; The MIT Press: Cambridge, MA, EUA, 2015; pp. 23–64. ISBN 978-0-262-32571-4. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2463983 Acesso em: 15 fev. 2025.

DIAGRAMS.NET - Online diagram software. Disponível em: <https://en.wikipedia.org/wiki/Diagrams.net>. Acesso em: 14 fev. 2025.

FADEL, L.M.; ULBRICHT, V.R.; BATISTA, C.R.; VANZIN, T. Gamificação na educação. São Paulo: Pimenta Cultural, 2014. FRACTIONS OF NUMBERS - MINI MATHS GOLF -

MATHSFRAME. Disponível em:

<https://mathsframe.co.uk/en/resources/resource/568/Fractions-of-Numbers-Mini-Maths-Golf>.

Acesso em: 2 fev. 2025.

GEE, J. P. (2004). *Situated Language and Learning*. New York and London: Routledge. Acesso em: 21 fev. 2025.

GEE, J.P. (2008). Video Games and Embodiment. *Games and Culture* 3 (3-4), 253-263. DOI: 10.1177/1555412008317309 Acesso em: 21 fev. 2025.

GIMP - GNU Image Manipulation Program. Disponível em: <https://www.gimp.org/>. Acesso em: 14 fev. 2025.

GIT -SCM. Disponível em: <https://git-scm.com/>. Acesso em: 14 fev. 2025.

GOOGLE DOCS: Free Online Documents for Personal Use. Disponível em:

https://en.wikipedia.org/wiki/Google_Docs. Acesso em: 14 fev. 2025.

GRAFFIGNA, M. L. et al. Lectura y comprensión de textos en el nivel superior: un desafío compartido entre alumnos y docentes. *Revista Iberoamericana de Educación*, v. 46, n. 2, p. 1–15, maio 2008. Acesso em: 19 fev. 2025.

GRAMMENOS, Dimitris; SAVIDIS, Anthony; STEPHANIDIS, Constantine. Designing universally accessible games. *Comput. Entertain.*, v. 7, n. 1, p. 1-29, fev. 2009. Disponível em: <https://dl.acm.org/doi/10.1145/1486508.1486516>. Acesso em: 1 fev. 2025.

GRANDO, Regina Célia. O conhecimento matemático e o uso de jogos na sala de aula. 2000. Tese (Doutorado em Educação) – Faculdade de Educação, UNICAMP, Campinas, 2000. Acesso em: 22 fev. 2025.

GREENWOOD, C., CARTA, J., MCCONNELL, S., GOLDSTEIN, H., & KAMINSKI, R. (2009). Center for Response to Intervention in Early Childhood (CRTIEC): Roadmap to Pre-K RTI. National Center for Learning Disabilities. <https://nclld.org/> Acesso em: 18 fev. 2025.

HART, B.; RISLEY, T. *Diferenças significativas na experiência cotidiana de crianças americanas*. Baltimore, MD: Paul H. Brookes Publishing Co., Inc., 1995.

HOWARD-JONES, P. A.; JAY, T.; MASON, A.; JONES, H. Gamification of learning deactivates the default mode network. *Frontiers in Psychology*, v. 6, 2016. Disponível em: <https://www.frontiersin.org/research-topics/30148/gamification-in-education/magazine>. Acesso em: 03 fev. 2025.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA – INEP. Perfil do docente da educação básica: dados do Censo Escolar. Brasília: INEP, 2022.

ISPMANAGER.COM. The history of game engines — from assembly coding to photorealism and AI. Disponível em: <https://dev.to/ispmanager/the-history-of-game-engines-from-assembly-coding-to-photorealism-and-ai-d9h>. Acesso em: 08 fev. 2025.

JALONGO, M. R., & SOBOLAK, M. J. (2011). Supporting young children’s vocabulary growth: The challenges, the benefits, and evidence-based strategies. *Early Childhood Education Journal*, 38(6), 421–429. Acesso em: 18 fev. 2025.

CAIN, Kate. (2009). Making sense of text: Skills that support text comprehension and its development. *Perspectives on Language and Literacy*. Disponível em : <https://onlit.org/wp-content/uploads/2023/08/Kate-Cain-Reading-Comprehension.pdf> . Acesso em: 19 fev. 2025.

KAPP, K.M. *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*; Pfeiffer: San Francisco, CA, USA, 2012; ISBN 978-1-118-09634-5. Acesso em: 15 fev. 2025.

KELLER, John. (2010). *Motivational Design for Learning and Performance: The ARCS Model Approach*. 10.1007/978-1-4419-1250-3. Acesso em: 16 fev. 2025.

KIRYAKOVA, Gabriela & ANGELOVA, Nadezhda & YORDANOVA, Lina. (2014). *GAMIFICATION IN EDUCATION*. Acesso em: 16 fev. 2025.

KRITA | Free Digital Painting Software. Disponível em: <https://krita.org/en/>. Acesso em: 14 fev. 2025.

LAUFER, B. 1997 The lexical plight in second language reading. IN: COADY, J. & T.N. HUCKIN 1997. *Second Language Vocabulary Acquisition*. Cambridge University Press. p. 20-34. Acesso em: 18 fev. 2025.

LEAL, T. F.; ALBUQUERQUE, E. B. C. de; LEITE, T. M. R. Jogos: alternativas didáticas para brincar alfabetizando (ou alfabetizar brincando?). In: MORAIS, A. G. de; ALBUQUERQUE, E. B. C. de; LEAL, T. F. (org.). *Alfabetização: apropriação do sistema de escrita alfabética*. Belo Horizonte: Autêntica, 2005, p. 111-131. Acesso em: 15 fev. 2025.

LIGHTNINGBOLT SOLUTIONS. Kinematic Character Controller. Unity Asset Store, 2025. Disponível em:

<https://assetstore.unity.com/packages/tools/physics/kinematic-character-controller-99131>. Acesso em: 16 ago. 2025.

LISSAK, G. Adverse physiological and psychological effects of screen time on children and adolescents: Literature review and case study. *Environmental Research*, v. 164, p. 149–157, 2018. DOI: 10.1016/j.envres.2018.01.015.

LIBREOFFICE - Free Office Suite - Fun Project - Fantastic People. Disponível em: <https://pt-br.libreoffice.org/>. Acesso em: 14 fev. 2025.

LÔBO, Ítalo Martins. A gamificação na educação infantil: impactos na qualidade do ensino. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 2024. Disponível em: <https://www.periodicos.unirp.br/index.php/rease/article/view/13712>. Acesso em: 02 fev. 2025.

MALONE, T. & LEPPER, M. (1987). Making Learning Fun: a Taxonomy of Intrinsic Motivations for Learning. In R. Snow and M. Farr (Eds.) *Aptitude, Learning & Instruction*. III. Conative & Affective Process Analyses, 223–253. NJ: Erlbaum. Acesso em: 15 fev. 2025.

MARCZEWSKI, A. (2013, 03 11). What's the difference between Gamification and Serious Games? Retrieved from Gamasutra: <https://www.gamedeveloper.com/>. Acesso em: 22 fev. 2025.

MISTRY, K. B.; MINKOVITZ, C. S.; STROBINO, D. M.; BORZEKOWSKI, D. L. G. Children's television exposure and behavioral and social outcomes at 5.5 years: does timing of exposure matter? *Pediatrics*, v. 120, n. 4, p. 762–769, 2007. Disponível em: <https://publications.aap.org/pediatrics/article/120/4/762/30676/Children-s-Television-Exposure-and-Behavioral-and>. Acesso em: 02 fev. 2025.

MORAIS, A. G. de; ALBUQUERQUE, E. B. C. de; LEAL, T. F. (org.). *Alfabetização: apropriação do sistema de escrita alfabética*. Belo Horizonte: Autêntica, 2005. Acesso em: 21 fev. 2025.

MORAIS, J.; LEITE, I.; KOLINSKY, R. Entre a pré-leitura e a leitura hábil: condições e patamares da aprendizagem. In: MALUF, M.R.; CARDOSO-MARTIN, C. *Alfabetização no século XXI: como se aprender a ler e escrever*. Porto Alegre: penso, 2013. p. 49-81. Acesso em: 18 fev. 2025.

NOTION – The all-in-one workspace. Disponível em: <https://www.notion.so/>. Acesso em: 14 fev. 2025.

NUNES, Terezinha, BRYANT, Peter. *Crianças fazendo matemática*. Porto Alegre: Artmed, 1997. Acesso em: 20 fev. 2025.

OLDFIELD, Bernard J. Games in the learning of mathematics. *Mathematics and School*. Artigos 1-5, March 1991 – January 1992. Acesso em: 22 fev. 2025.

OLIVEIRA, Ana Maria Pinto Pires de, ISQUERDO, Aparecida Negri (orgs.). 2 ed. As ciências do léxico: lexicologia, lexicografia, terminologia. Campo Grande:Ed. UFMS, 2001. Acesso em: 18 fev. 2025.

OLIVEIRA, Rhále Nascimento de; BELARMINO, Guilherme Dias; RODRIGUEZ, Carla; GOYA, Denise; ROCHA, Rafaela Vilela da; FERNÁNDEZ VENERO, Mirtha Lina; BENITEZ, Priscila; KUMADA, Kate Mamhy Oliveira. Desenvolvimento e Avaliação da Usabilidade e Acessibilidade de um Protótipo de Jogo Educacional Digital para Pessoas com Deficiência Visual. 2021. Disponível em: https://www.researchgate.net/publication/353592561_Desenvolvimento_e_Avaliacao_da_Usabilidade_e_Acessibilidade_de_um_Prototipo_de_Jogo_Educacional_Digital_para_Pessoas_com_Deficiencia_Visual. Acesso em: 17 ago. 2025.

PAGANI, L. S. et al. Prospective associations between early childhood television exposure and academic, psychosocial, and physical well-being by middle childhood. *Archives of Pediatrics & Adolescent Medicine*, v. 167, n. 5, p. 425–431, 2013. DOI: 10.1001/jamapediatrics.2013.50.

PERFETTI, Ch. & HART, L. (2001) The lexical bases of reading comprehension. In D. S. Gorfien (Ed.), *On the consequences of meaning selection: Perspectives on resolving lexical ambiguity* (pp. 67-86). American Psychological Association. Acesso em: 18 fev. 2025.

PERRUSQUIA, E., CARRANZA, L., VÁZQUEZ, T., GARCÍA, M., MEZA, F. Curso básico de formación continua para maestros en servicio, planeación didáctica para el desarrollo de competencias en el aula 2010. Subsecretaria de Educación Básica, SEP (2010) Acesso em: 19 fev. 2025.

PHAM, X.-L. et al. (2016). Card-based design combined with spaced repetition: A new interface for displaying learning elements and improving active recall. *Computers & Education*, 98(Supplement C):142 – 156. Acesso em: 25 fev. 2025.

PINTRICH, P. R.; BROWN, D. R.; WEINSTEIN, C. E. (Eds.). *Student Motivation, Cognition, and Learning: Essays in Honor of Wilbert J. McKeachie*. 1. ed. Routledge, 1994. Disponível em: <https://doi.org/10.4324/9780203052754>. Acesso em: 18 fev. 2025.

PIAGET, J. Seis estudos de psicologia. 24. ed. Rio de Janeiro: Florence, 2002. 136 p. Acesso em: 20 fev. 2025.

POETA, C. D. and GELLER, M. (2014). Jogos digitais educacionais: Concepções metodológicas na prática pedagógica de matemática no ensino fundamental, *Educação Matemática Revista*. <https://www.sbemrasil.org.br/periodicos/index.php/EMR-RS/article/view/1496> . Acesso em: 22 fev. 2025.

PORTAL LUDO EDUCATIVO - MANDA LETRA - REMAKE. Disponível em:
<https://www.ludoeducativo.com.br/pt/play/manda-letra-remake?tag=lingua-portuguesa>. Acesso em: 7 mar. 2025.

ALMEIDA, Daniella. Quatro em cada dez alunos brasileiros do 4º ano não dominam a leitura. Agência Brasil, Brasília, 16 maio 2023. Disponível em:
<https://agenciabrasil.ebc.com.br/educacao/noticia/2023-05/quatro-em-dez-alunos-brasileiros-do-4-o-ano-nao-dominam-leitura>. Acesso em: 02 fev. 2025.

REVISTA FT. Tempo de tela em crianças: quais os riscos? Revista Fisioterapia e Terapia, v. 5, p. 21–29, 2021. Disponível em: <https://revistaft.com.br/tempo-de-tela-criancas>. Acesso em: 3 ago. 2025.

ROBINSON, Alice J. "The design is the game: Writing games, teaching writing." Computers and Composition 25.3: 359-370, 2008 <https://doi.org/10.1016/j.compcom.2008.04.006>. Acesso em: 21 fev. 2025.

SÁNCHEZ CASTRO, S., & PASCUAL SEVILLANO, M. Ángeles. (2022). Eficacia de un juego serio digital para la mejora de la comprensión lectora y el rendimiento académico. Investigaciones Sobre Lectura, 1(17), 40-66. <https://doi.org/10.24310/isl.vi17.14325> . Acesso em: 21 fev. 2025.

SCHELL, Jesse. The Art of Game Design: a Book of Lenses. Amsterdam; Boston: Elsevier/Morgan Kaufmann, 2008. Acesso em: 25 fev. 2025.

SCHWABER, Ken; SUTHERLAND, Jeff. O guia do Scrum: o guia definitivo para o Scrum, a regra do jogo. 2020. Disponível em:
<https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 8 fev. 2025. Acesso em: 15 fev. 2025.

SCHWABER, K.; SUTHERLAND, J. O guia do Scrum: o guia definitivo para o Scrum, as regras do jogo. [s.l.: s.n.]. Disponível em:
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>. Acesso em: 13 fev. 2025.

ŠMÍD, A. Comparison of Unity and Unreal Engine. Undergraduate thesis. Czech Technical University in Prague, 2017. Disponível em: <https://core.ac.uk/download/pdf/84832291.pdf>. Acesso em: 08 fev. 2025.

SOWDER, J. A compreensão de número na escola e primeiro grau. MEIRA, Luciano & SPINILLO, Alina Galvão (Orgs). Anais da I Semana de Estudos em Psicologia da Educação Matemática. p.19-27. Recife: Universidade Federal de Pernambuco, 1995. Acesso em: 20 fev. 2025.

SPINILLO, A. G. O Sentido de Número e sua Importância na Educação Matemática. BRITO, Márcia Regina Ferreira de (Org.). Soluções de Problemas e a Matemática Escolar. p.83- 111. Campinas: Alínea, 2006. Acesso em: 15 fev. 2025.

SKINNER, B. F. Teaching Machines. Science, Washington, v. 128, n. 3330, p. 969–977, 1958. Disponível em:
<https://www.bfskinner.org/wp-content/uploads/2014/02/teaching-machines-1958.pdf>. Acesso em: 03 ago. 2025.

SUSHI MONSTER - APP-ENFANT. Disponível em:
<https://app-enfant.fr/application/sushi-monster-application-gratuite-addition-multiplication>.
 Acesso em: 7 mar. 2025.

SUSI T, JOHANNESSON M, BACKLUND P. Serious Games: An Overview [Internet]. Skövde: Institutionen för kommunikation och information; 2007. (IKI Technical Reports). Available from:
<https://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-1279>. Acesso em: 21 fev. 2025.

SUM STACKS | MATH PLAYGROUND. Disponível em:
https://www.mathplayground.com/sum_stacks.html. Acesso em: 7 mar. 2025.

TAHERI, E.; HESHMAT, R.; MOTLAGH, M. E.; ARDALAN, G.; ASAYESH, H.; QORBANI, M.; KELISHADI, R. Association of physical activity and screen time with psychiatric distress in children and adolescents: CASPIAN-IV study. Journal of Tropical Pediatrics, v. 65, n. 4, p. 361–372, 2019. Disponível em:
<https://academic.oup.com/tropej/article/65/4/361/5529158?login=false>. Acesso em: 02 fev. 2025.

TONNESSEN, E. S. Potentials for writing. In: COSTA, J. C. de; PEREIRA, V. W. Linguagem e cognição [recurso eletrônico]: relações interdisciplinares. Dados eletrônicos –Porto Alegre : EDIPUCRS, 2009. 331 p. Acesso em: 18 fev. 2025.

TWENGE, J. M.; CAMPBELL, W. K. Associations between screen time and lower psychological well-being among children and adolescents: Evidence from a population-based study. Preventive Medicine Reports, v. 12, p. 271–283, 2018. DOI: 10.1016/j.pmedr.2018.10.003.

UNESCO. *Technology in education: a tool on whose terms?* Paris: Global Education Monitoring Report, 2023. Disponível em:
https://unesdoc.unesco.org/ark:/48223/pf0000386147_por/PDF/386147por.pdf.multi.
 Acesso em: 18 jun. 2025.

UNICEF. Analfabetismo funcional não apresenta melhora e alcança 29% dos brasileiros, mesmo patamar de 2018, aponta novo levantamento do Inaf. Brasília: Fundo das Nações Unidas para a Infância, 16 maio 2024. Disponível em:
<https://www.unicef.org/brazil/comunicados-de-imprensa/analfabetismo-funcional-nao-apresenta>

melhora-e-alcanca-29-por-cento-dos-brasileiros-mesmo-patamar-de-2018-aponta-novo-levantamento-do-inaf. Acesso em: 3 ago. 2025.

UNITY TECHNOLOGIES. ScriptableObject. Unity Manual, 2025. Disponível em: <https://docs.unity3d.com/ScriptReference/ScriptableObject.html>. Acesso em: 16 ago. 2025.

VALENTE, José Armando. Computadores e conhecimento: repensando a educação. Campinas:UNICAMP/NIED, 1998. <https://www.nied.unicamp.br/wp-content/uploads/other-files/livro-computadores-e-conhecimento.pdf>

VALENTE, W. R.; FERREIRA, J. dos S. Como se ensina aritmética? Elementos de uma aritmética para ensinar no manual de Faria de Vasconcelos. Ensino em Re-Vista, v. 28, n. Contínua, p. e027, 11 maio 2021. <https://repositorio.ufsc.br/handle/123456789/224117>. Acesso em: 20 fev. 2025.

VOCABICAR. Disponível em: <https://vocabicar.de/?lang=en>. Acesso em: 7 mar. 2025.

8 ANEXO A - LINKS

- I. GITHUB: [GitHub - Jayzeferino/GameTCC](#)
- II. LINK DOWNLOAD APK PARA ANDROID: