



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

Henrique Calaboni Godoy

**DESENVOLVIMENTO DE JOGO ROGUELIKE COM SISTEMA DE PESCA:
Deep Cave Catch**

**Americana, SP
2026**



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

Henrique Calaboni Godoy

**DESENVOLVIMENTO DE JOGO ROGUELIKE COM SISTEMA DE PESCA:
Deep Cave Catch**

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

Área de concentração: Jogos Digitais.

Americana, SP

2026

GODOY, Henrique Calaboni

Desenvolvimento de jogo roguelike com sistema de pesca:
Deep Cave Catch. / Henrique Calaboni Godoy – Americana, 2026.

71f.

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

Orientador: Prof. Esp. José William Pinto Gomes

1. C # – linguagem de programação 2. Desenvolvimento de
software 3. Jogos eletrônicos. I. GODOY, Henrique Calaboni II.
GOMES, José William Pinto III. Centro Estadual de Educação
Tecnológica Paula Souza – Faculdade de Tecnologia de Americana
Ministro Ralph Biasi

CDU: 681.3.061C#

681.3.05

681.6

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

Henrique Calaboni Godoy

**DESENVOLVIMENTO DE JOGO ROGUELIKE COM SISTEMA DE PESCA:
Deep Cave Catch**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Jogos Digitais pelo Centro Paula Souza – Faculdade de Tecnologia de Americana – Ministro Ralph Biasi.

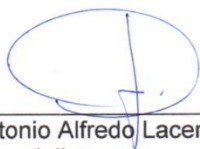
Área de concentração: Jogos Digitais.

Americana, 08 de junho de 2026

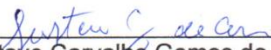
Banca Examinadora:



José William Pinto Gomes (Presidente)
Especialista
Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"



Antonio Alfredo Lacerda (Membro)
Especialista
Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"



Gustavo Carvalho Gomes de Abreu (Membro)
Especialista
Faculdade de Tecnologia de Americana "Ministro Ralph Biasi"

RESUMO

Este trabalho documenta o desenvolvimento de um jogo 2D que mescla mecânicas do gênero *Roguelike* com mini jogos de pesca. O principal objetivo foi averiguar a viabilidade dessa junção de mecânicas para o engajamento do jogador. O projeto foi desenvolvido na game *engine Unity* utilizando programação C#, com artes autorais, geração de mapa procedural através do posicionamento aleatório de salas pré-fabricadas em uma matriz e mini jogo de pesca, que serve como quebra de ritmo do combate e é fonte de melhorias e cura para o jogador, o qual deve explorar salas enquanto tenta encontrar a saída e permanecer vivo ao longo de duas fases implementadas neste projeto. A partir de constantes testes de qualidade durante o desenvolvimento e de respostas adquiridas através do teste beta realizado, é possível afirmar que esse design é válido e atinge os objetivos deste trabalho, resultando em um jogo engajante e divertido para os jogadores.

Palavras-chave: jogo digital; *roguelike*; desenvolvimento de jogos

ABSTRACT

This paper documents the development of a 2D game that merges mechanics from the Roguelike genre with fishing minigames. The main objective was to investigate the viability of this combination of mechanics regarding player engagement. The project was developed in the Unity game engine using C# programming, featuring original artwork, procedural map generation through the random positioning of prefabricated rooms in a matrix, and a fishing minigame that serves as a pacing break from combat, as well as a source of upgrades and healing for the player. The player must explore rooms while attempting to find the exit and stay alive throughout the two stages implemented in this project. Based on constant quality testing during development and feedback acquired through the conducted beta test, it can be stated that this design is valid and achieves the objectives of this work, resulting in an engaging and fun game for the players.

Keywords: digital game; roguelike; game development.

LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxograma de telas.....	30
Figura 2 - Arte conceitual do personagem jogável	33
Figura 3 - Arte conceitual do inimigo	34
Figura 4 – Personagem jogável (principal).....	34
Figura 5 - Inimigo (peixe com lança)	34
Figura 6 - Sala inicial (com exemplo de posicionamento de portas)	35
Figura 7 - Variação 1 de sala de combate	36
Figura 8 - Variação 2 de sala de combate	37
Figura 9 - Variação 3 de sala de combate	38
Figura 10 - Variação 4 de sala de combate	39
Figura 11 - Sala de pesca.....	40
Figura 12 - Sala com porta para a próxima fase	41
Figura 13 - Fase 2 (arena de combate).....	42
Figura 14 - Variação 1 de coral	43
Figura 15 - Variação 2 de coral	43
Figura 16 - Variação 3 de coral	44
Figura 17 – <i>Sprite sheet</i> da animação <i>idle</i> do jogador	44
Figura 18 - <i>Sprite sheet</i> da animação de andar do jogador	45
Figura 19 - <i>Sprite sheet</i> da animação de ataque do jogador.....	45
Figura 20 - <i>Sprite sheet</i> da animação de andar do inimigo	45
Figura 21 - <i>Sprite sheet</i> da animação de atacar do inimigo	46
Figura 22 - <i>Sprite sheet</i> da zona de pesca ativa	46
Figura 23 - <i>Sprite sheet</i> do portão abrindo (a ordem inversa é usada para fechar) ..	46
Figura 24 - <i>Sprite sheet</i> do efeito de ataque	46
Figura 25 - Menu inicial	47
Figura 26 - Tutorial	47
Figura 27 - Opções.....	48
Figura 28 - Créditos.....	48
Figura 29 - Indicador de vida do jogador (vida cheia)	48
Figura 30 - Indicador de vida do jogador (após levar 2 de dano)	49
Figura 31 - Diagrama de estados do jogo	50

Figura 32 – Gráfico de respostas à pergunta sobre gênero	63
Figura 33 - Gráfico de respostas à pergunta sobre idade	63
Figura 34 - Gráfico de respostas à pergunta sobre frequência de jogo.....	64
Figura 35 - Gráfico de respostas de opinião geral sobre o jogo	64
Figura 36 - Gráfico de respostas sobre o design do jogo	65
Figura 37 - Gráfico de respostas sobre a jogabilidade	65
Figura 38 - Gráfico de respostas sobre menu e interação	66
Figura 39 - Gráfico de respostas sobre arte e som	66
Figura 40 - <i>Feedback</i> dos participantes 1	66
Figura 41 - <i>Feedback</i> dos participantes 2	67

LISTA DE QUADROS

Quadro 1 - Comparação entre os jogos.	18
Quadro 2 - Cronograma.	55
Quadro 3 - Análise de riscos.	57

LISTA DE ABREVIATURAS E SIGLAS

FSM: *Finite State Machine* (máquina de estado finito)

RPG: *Role Playing Game* (jogo de interpretação de papéis)

S.N.: Sem nome

SUMÁRIO

1	INTRODUÇÃO	14
2	REVISÃO BIBLIOGRÁFICA	15
2.1	Jogo e Jogo Digital	15
2.2	Gênero	15
2.3	<i>Roguelike e Roguelite</i>	16
3	METODOLOGIA	17
3.1	Características	17
3.2	Jogos similares	17
3.2.1	Quadro comparativo	18
4	CONCEITO	19
4.1	Gênero	19
4.2	Classificação etária	19
4.3	Trama principal	19
4.4	<i>Look & Feel</i>	19
4.5	Visão geral do conceito do jogo	20
5	HISTÓRIA E NARRATIVA	21
5.1	Visão geral da história	21
5.1.1	Mundo do jogo	21
5.2	Personagens	21
5.2.1	Eventos e ações	21
5.3	Visão geral da narrativa	22
5.3.1	Integração	22
5.3.2	Estrutura e progressão	22
5.3.3	Roteiro	22
5.3.4	Tutoriais e manual do jogo	22
5.4	Cenas e sequências cinemáticas	23
6	GAMEPLAY	24
6.1	Imersão	24
6.2	Estrutura de missões e desafios	24
6.3	Objetivos do jogo	24
6.4	Fluxo de jogo	25

6.5	Mecânica de jogo	25
6.5.1	Regras implícitas e explícitas do jogo	25
6.5.2	Física	25
6.5.3	Movimentação dos personagens	26
6.5.4	Objetos.....	26
6.5.5	Gatilhos e ações	26
6.5.6	Mecânica de combate	26
6.6	Projeto de fases (<i>level design</i>)	27
6.7	Projeto de interface	28
6.7.1	Sistema visual	28
6.7.2	Sistema de controle	29
6.7.3	Fluxo de telas.....	29
6.7.4	Opções do jogo	30
6.7.5	<i>Saving e checkpoints</i>	31
6.7.6	Sistema de ajuda	31
6.8	Projeto de som	31
6.8.1	Descrição geral	31
6.8.2	Trilhas sonoras.....	31
6.8.3	Efeitos sonoros	31
6.9	Sistema de inteligência artificial	32
7	ARTE	33
7.1	Guia de cores e estilos gráficos	33
7.2	Arte conceitual.....	33
7.3	<i>Asset List</i>	34
7.3.1	Personagens	34
7.3.2	Ambientes	35
7.3.3	Animações.....	44
7.3.4	Interfaces.....	47
8	DOCUMENTAÇÃO TÉCNICA	50
8.1	Requisitos de sistema	50
8.2	Engenharia de <i>software</i>	50
8.3	<i>Procedures e Patterns</i>	51
8.4	Game engine.....	51
8.5	Programação e <i>Scripting</i>	52

9	GERENCIAMENTO E ANÁLISE DO PROJETO	53
9.1	Organização inicial do projeto	53
9.2	Escopo do projeto	53
9.3	Recursos	54
9.4	Entregas	54
9.5	Premissas.....	54
9.6	Restrições	54
9.7	Cronograma	55
9.8	Estimativa de custos	56
9.9	Limites do projeto	56
9.10	Processos.....	57
9.11	Análise de riscos	57
9.12	Viabilidade técnica	58
9.13	Viabilidade econômica.....	59
9.14	Plano de testes e controle de qualidade	59
9.15	Ferramentas e técnicas.....	60
10	RESULTADOS	61
10.1	Teste beta.....	61
10.2	Análise da loja	67
11	CONSIDERAÇÕES FINAIS	68
	DOWNLOAD DO JOGO.....	70
	REFERÊNCIAS.....	71

1 INTRODUÇÃO

Ao longo dos anos, os jogos digitais passaram por diversas mudanças de *design* que definiram a modificação de gêneros existentes ou criação de gêneros novos, eventualmente mesclando diferentes mecânicas de diversos gêneros a fim de criar experiências cada vez mais complexas para os jogadores. Um gênero que se destaca é o *Roguelike*, marcado pela geração procedural de cenários e morte permanente do jogador. Este projeto buscou mesclar tais mecânicas de *Roguelike* com minijogos de pesca, os quais normalmente estão presentes em jogos de RPG (*Role Playing Games*) ou jogos casuais, com a finalidade de avaliar a efetividade dessa junção no engajamento do jogador.

O objetivo deste trabalho é a criação do jogo intitulado *Deep Cave Catch*, combinando características do gênero *Roguelike* (morte permanente, geração procedural, combate) com minijogos de pesca em uma visão 2D *top-down* em *pixel art*.

Este documento está estruturado em 10 capítulos que contemplam a criação e documentação dos processos de concepção do jogo. O capítulo 1 trata da revisão da literatura existente; o capítulo 2 mostra a metodologia utilizada; o capítulo 3 aborda o conceito geral do jogo; o capítulo 4 expõe a história e narrativa; o capítulo 5 engloba as características de gameplay; o capítulo 6 revela as artes criadas; o capítulo 7 detalha a perspectiva técnica do projeto; o capítulo 8 aborda o gerenciamento do projeto, incluindo tópicos como análise de riscos e viabilidade do jogo; por fim, o capítulo 10 expõe a análise dos resultados obtidos durante a criação do jogo e pelas respostas do teste realizado.

2 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta definições de termos e conceitos que representam a base para o entendimento do projeto e para o desenvolvimento do jogo digital proposto através da pesquisa e análise de obras existentes, como livros e artigos, e exemplos de jogos que se encaixam nas definições estudadas.

2.1 Jogo e Jogo Digital

Um jogo, segundo Huizinga (1950), é definido como uma atividade exercida dentro de certos limites de tempo e espaço, segundo regras próprias, as quais são livremente aceitas, mas também absolutamente seguidas. Essa atividade tem por objetivo ela mesma e possui como característica fundamental a consciência de que é diferente da vida cotidiana. Essa definição evidencia o caráter lúdico que serve como base para os jogos.

Nos jogos digitais, a definição se estende para incluir a interatividade fornecida pelo meio digital. Salen e Zimmerman (2004) definem o jogo digital como um sistema pelo qual os jogadores participam de um conflito artificial com regras definidas e com resultados quantificáveis, enfatizando tanto a estrutura dessas regras quanto o feedback para guiar a experiência do jogador, concordando com a definição de Huizinga (1950). Como complemento a essa perspectiva, Juul (2005) argumenta que os jogos digitais preservam as características essenciais dos jogos tradicionais, como regras e objetivos, porém, se diferenciam pela capacidade de simular ambientes complexos, capazes de se adaptar e responder dinamicamente às ações do jogador.

O conceito de jogo digital adotado para esse projeto está alinhado com essas definições clássicas, as quais enfatizam o caráter interativo, a presença de regras sistematizadas e bem definidas, objetivos claros e feedback contínuo, elementos que fundamentam a experiência do jogador e seu engajamento no jogo.

2.2 Gênero

Os jogos digitais são classificados em gêneros e podem pertencer a mais de um simultaneamente. Para Adams (2014), o gênero reflete os tipos de desafios e

mecânicas presentes neles sem levar em conta seu conteúdo, ao contrário de outras obras, como livros, filmes e séries, os quais são classificados por sua narrativa. O jogo desenvolvido neste projeto pertence ao gênero *Roguelike*.

2.3 *Roguelike e Roguelite*

O gênero *Roguelike* se origina no jogo clássico *Rogue*, lançado em 1980, caracterizado por aspectos como geração procedural de mapas, morte permanente (permadeath), combate em turnos e dificuldade elevada, segundo Cartlidge (2024). Adams (2014) complementa, definindo o gênero por uma estrutura de jogo onde o progresso é perdido e reiniciado a cada vez que o jogador perde, incentivando-o a constantemente aprender, experimentar e se adaptar em tentativas geradas aleatoriamente. Esses elementos geram o sentimento de tensão no jogador a cada ação realizada, pois cada erro pode resultar no fim de seu progresso, ao mesmo tempo em que proporcionam uma experiência sempre renovada com a inserção da aleatoriedade.

A *Berlin Interpretation of Roguelikes* é a tentativa mais reconhecida de definir o gênero, formulada na *International Roguelike Development Conference* em 2008, apresentada no formato de lista, criada por fãs e desenvolvedores, a qual pontua mecânicas (como os presentes no jogo *Rogue*) com diferentes prioridades para definir o quanto um jogo se encaixa no gênero, segundo Cartlidge (2024). Porém, quando um jogo se afasta parcialmente das características do gênero, surge um subgênero chamado *Roguelite*, composto por diversos elementos presentes nos *Roguelikes*, porém, nunca apresentando todos simultaneamente. Como exemplo de *Roguelite*, em *Hades*, desenvolvido pela Supergiant Games (2020), o combate não é em turnos e o jogo apresenta formas de progressão entre as tentativas enquanto mantém a característica de morte permanente e geração de mapa aleatória dentro de cada tentativa.

3 METODOLOGIA

O objetivo deste projeto é o desenvolvimento e documentação de um jogo digital que mescla o gênero Roguelike com mecânicas de mini jogos de pesca. Para isso, foram utilizadas revisões da literatura existente sobre definições e características de jogos digitais, além de um teste visando o *feedback* e coleta de dados através de questionários online sobre o estado do jogo e possíveis pontos de melhoria.

3.1 Características

As especificações principais deste projeto/jogo são:

- Gênero: *Roguelike* com mecânicas de mini jogo de pesca.
- Plataforma Alvo: Desktop (Executável nativo para sistemas operacionais Microsoft Windows).
- Direcionamento artístico: 2D, *pixel art*, *top-down*.
- Público-alvo: Jogadores que gostam de desafios, jogos do gênero *Roguelike* e que gostam de tornar cada falha em possibilidade de aprendizado.
- Classificação ESRB: Livre para todos.

3.2 Jogos similares

O primeiro similar analisado é ***The Binding of Isaac***, desenvolvido por Edmund McMillen e Florian Himsl (2011), uma das referências fundamentais para o subgênero *Roguelike* de ação. O título implementa a estrutura de navegação por salas fechadas interconectadas em formato de matriz e valida o sistema de combate baseado em quatro direções espaciais via teclado. A dissociação entre o eixo de movimento do avatar e o eixo de disparo dos ataques, operada de forma similar pelo uso das setas direcionais em *Deep Cave Catch*, demonstra eficiência na criação de desafios baseados em posicionamento espacial e leitura de cenários.

Para o segundo similar, analisou-se o subsistema de pesca contido em ***Stardew Valley***, desenvolvido por ConcernedApe (2016). O mini jogo baseia-se em

um indicador visual oscilante de pressão e ritmo. Essa mecânica de retenção e precisão foi aplicada no jogo desenvolvido, atuando como forma de fortalecer o jogador e funcionando como uma ferramenta de quebra controlada de ritmo (*pacing*) entre os combates.

3.2.1 Quadro comparativo

Quadro 1 - Comparação entre os jogos.

Jogos	Plataforma	Fases	Dificuldades	Preço
Deep Cave Catch	Desktop (Windows)	2 fases	Progressiva com as fases	Gratuito
The Binding of Isaac	Desktop / Consoles	10 fases	3 dificuldades	R\$8,49
Stardew Valley	Desktop / Consoles / Mobile	Mapa fixo	Modificada pelo jogador	R\$24,99

Fonte: Desenvolvido pelo autor (2026).

O diferencial do *Deep Cave Catch* está na fusão de *The Binding of Isaac* e *Stardew Valley*, ao unir a tensão severa e a exploração procedural de um sistema *Roguelike* com uma jogabilidade focada em mini jogos de pesca. Essa arquitetura de design quebra a fadiga cognitiva do jogador ao alternar organicamente momentos de alta tensão com momentos de descanso ativo para a coleta de upgrades.

4 CONCEITO

Deep Cave Catch é um jogo eletrônico de exploração de masmorras que une a tensão do combate corpo a corpo com o relaxamento de minijogos de pesca em um mapa gerado proceduralmente através do posicionamento de salas pré fabricadas.

4.1 Gênero

O jogo classifica-se como um *Roguelike*. Suas mecânicas se baseiam na navegação por salas geradas de forma procedural em *grid* e no sistema de morte permanente. O *loop* principal de jogabilidade é quebrado intencionalmente pela inclusão da atividade de pesca, utilizada para aumentar atributos do jogador, como dano e velocidade de ataque.

4.2 Classificação etária

A audiência-alvo é composta por jogadores entusiastas de desafio e se enquadra na categoria livre, abstendo-se de conteúdos de teor adulto ou violência visual gráfica.

4.3 Trama principal

Um jovem, buscando provar que é o melhor pescador do mundo, parte em uma jornada atrás da lenda de um peixe capaz de realizar qualquer desejo de quem o capturar. O pescador precisará utilizar de suas habilidades de combate e pesca para explorar a caverna à beira do mar.

4.4 Look & Feel

O jogo adota a estética em *Pixel Art* com perspectiva superior (*Top-Down*), evocando o charme e a legibilidade clássica dos jogos de aventura das eras 16-bits e 32-bits. A atmosfera busca transmitir um sentimento de mistério por meio de tons

escuras e frias das paredes rochosas, contrastados por corais coloridos que indicam a vida na caverna.

4.5 Visão geral do conceito do jogo

Ambientado nas profundezas de uma caverna à beira do mar em um período atemporal, o jogador controla um pescador minimalista sem nome. A progressão se dá andar por andar, onde o jogador deve explorar salas interconectadas geradas aleatoriamente em busca de zonas de pesca, para melhorar seus atributos, e da saída para o próximo nível. O jogador irá encontrar salas com inimigos, os quais precisam ser derrotados antes de prosseguir. Nesta iteração do jogo, após encontrar a saída para o próximo nível, o jogador deve sobreviver a três ondas de inimigos em uma arena.

5 HISTÓRIA E NARRATIVA

5.1 Visão geral da história

O enredo acompanha um jovem pescador focado em provar que é o melhor de todo o mundo. A história ocorre a partir da existência da lenda de um peixe capaz de realizar qualquer desejo de quem o pescar, habitando o ponto mais profundo e inexplorado das cavernas subterrâneas a beira do mar. Seguindo essa lenda, o jovem pescador parte em sua jornada.

5.1.1 Mundo do jogo

O universo do jogo é composto pelo complexo subterrâneo à beira do mar, um ambiente caracterizado pelo isolamento, pela umidade e por uma geografia labiríntica verticalizada. O espaço divide-se em andares compostos por salas interconectadas. À medida que o jogador desce, as salas geradas em formato de grid ficam mais complexas e possuem criaturas mais perigosas, além de peixes mais difíceis de pescar, mas que podem melhorar os atributos do jogador, gerando a sensação de isolamento e aprofundamento em um ecossistema hostil e intocado.

5.2 Personagens

O protagonista do jogo é um jovem que busca provar que é o melhor pescador do mundo.

Os antagonistas do jogo incluem peixes antropomórficos que habitam a caverna na qual o jogo ocorre. Nesta iteração do jogo, o inimigo é um peixe que utiliza uma lança contra o jogador pois este está invadindo seu lar.

5.2.1 Eventos e ações

Os eventos que levam ao momento do jogo são o pescador (protagonista) buscando provar que é o melhor de todo o mundo e a descoberta da lenda do peixe

que pode realizar qualquer desejo de quem for capaz de pescá-lo, partindo em sua jornada para uma caverna à beira do mar.

5.3 Visão geral da narrativa

5.3.1 Integração

A arte em pixel remete a jogos clássicos e reforça uma jogabilidade que foca em expandir interações através de mecânicas simples, como o combate e a pesca, que sustenta a narrativa do pescador e serve como forma de melhorar os atributos do jogador ao longo da partida.

5.3.2 Estrutura e progressão

A estrutura narrativa é não-linear no microcosmo tendo em vista que cada tentativa apresenta uma disposição de salas e encontros diferente devido à aleatoriedade procedural, mas possui simultaneamente uma progressão linear no macrocosmo, como o objetivo final de alcançar o fundo se mantém. A progressão da narrativa se dá através de textos conforme o jogador se aprofunda na caverna, que, em versões futuras, culminará no encontro com o peixe lendário.

5.3.3 Roteiro

Devido ao foco na simplicidade, o jogo não apresenta diálogos entre o personagem principal e personagens não jogáveis, sendo a história transmitida diretamente para o jogador através de textos entre níveis.

5.3.4 Tutoriais e manual do jogo

O menu inicial do jogo apresenta os controles ao jogador, o qual deve descobrir como utilizá-los durante as partidas, adquirindo conhecimento e maestria sobre as mecânicas.

No momento em que o jogador encontra uma área de pesca, um ícone de botão aparece sobre o personagem, indicando ao jogador uma área de interação que leva

ao minijogo de pesca. Durante a pesca, um ícone de botão indica ao jogador como interagir.

As ações são: W, A, S e D para se mover, setas para atacar, E para interagir e BARRA DE ESPAÇO para mover a barra durante o minijogo de pesca.

5.4 Cenas e sequências cinemáticas

O jogo não utiliza animações pré-renderizadas. As cenas referentes a história se apresentam na forma de texto para o jogador.

6 GAMEPLAY

6.1 Imersão

A imersão em *Deep Cave Catch* é obtida através da alternância do estado de atenção do jogador, sempre requerendo seu foco total. O design não depende de múltiplos elementos complexos, mas sim da maestria em duas interações refinadas: o combate contra os inimigos e a precisão no minijogo de pesca. Durante o combate, o jogador deve focar em vários inimigos simultaneamente para atacar e desviar de golpes em momentos de tensão, já durante a pesca, mesmo representando um momento de relaxamento, o jogador deve manter o foco no minijogo para não falhar e perder a possibilidade de se tornar mais forte

6.2 Estrutura de missões e desafios

Por se tratar de um Roguelike focado na ação direta, o jogo adota uma estrutura de desafios emergentes por sala. Para cada andar, o jogador deve navegar pela masmorra em busca da passagem para o próximo andar. Em cada andar, existem dois tipos de desafios: salas de combate, nas quais o jogador deve derrotar todos os inimigos para prosseguir, e salas de pesca, nas quais o jogador pode interagir com a área no centro para iniciar o minijogo de pesca que deixa o jogador mais forte, caso seja completado com sucesso.

A segunda fase consiste em uma variação na sala de inimigos: o jogador deve derrotar todos os inimigos que aparecem em ondas para prosseguir.

6.3 Objetivos do jogo

O jogador deve navegar a masmorra em busca de locais de pesca para ficar mais forte e buscar a saída para o próximo andar. Durante a segunda fase, o jogador deve derrotar todos os inimigos que surgem em ondas. Caso o jogador morra, ele recomeça desde o início.

6.4 Fluxo de jogo

Conforme o jogador avança pelos andares da caverna (masmorra), ele chega mais perto do peixe lendário (narrativa), cada andar se torna mais difícil que o anterior (dificuldade) e mais forte o jogador pode ficar caso tenha sucedido nas pescas de cada nível.

Dessa forma, avançar no jogo resulta em chegar mais perto do objetivo do personagem, tornar-se mais forte e enfrentar desafios maiores.

6.5 Mecânica de jogo

6.5.1 Regras implícitas e explícitas do jogo

As regras explícitas do jogo são limitadas às opções de ação do jogador, indicadas no tutorial presente no menu inicial. Essa escolha promove a exploração e descoberta de mecânicas pelo jogador durante suas várias tentativas de vencer, recompensando a percepção e aprendizado.

As regras implícitas incluem: As portas se fecham em salas de inimigos até que todos sejam derrotados, o funcionamento dos inimigos, o minijogo de pesca (até que seja descoberto pelo jogador), as possíveis melhorias provenientes da pesca, os tipos de salas e a arena de batalha durante o segundo nível.

Ao longo das partidas, o jogador deve descobrir como os inimigos funcionam e como derrotá-los minimizando o dano sofrido, se adaptando ao arranjo procedural das salas e reconhecendo tipos de sala e pontos de surgimento de inimigos, de forma que o aprendizado do jogo ocorre de forma suave e transmite para o jogador que sua capacidade de reconhecer e se adaptar o torna progressivamente melhor no jogo ao longo de diversas tentativas.

6.5.2 Física

O jogo não apresenta usos específicos de física além de aplicar bordas que nem o jogador nem os inimigos podem atravessar, como as paredes e a zona de pesca.

6.5.3 Movimentação dos personagens

O personagem principal é controlado através dos comandos W, S, A e D para se mover para cima, baixo, esquerda e direita, respectivamente. Pode, também, atacar com as setas, indicando a direção de seu golpe.

O inimigo possui uma movimentação direta, seguindo o jogador até chegar perto o suficiente para iniciar seu ataque. O inimigo então pausa por um breve momento antes de desferir seu golpe na direção do jogador.

6.5.4 Objetos

Através da interação com zonas de pesca, o jogador pode adquirir aumento de atributos que o deixam mais forte. Nesta iteração do jogo, as melhorias de atributos são aplicadas diretamente ao jogador. Para versões futuras, essas melhorias serão representadas por itens.

6.5.5 Gatilhos e ações

O jogo possui dois gatilhos/ações principais. Primeiro, o jogador pode interagir com zonas de pesca para iniciar o minijogo. Segundo, ao entrar em uma sala com inimigos, as portas se fecham até que todos os inimigos presentes sejam derrotados, depois, as portas se abrem novamente, permitindo que o jogador continue sua exploração.

6.5.6 Mecânica de combate

O combate foca em timing e posicionamento, no qual tanto o jogador quanto os inimigos possuem pontos de vida. Dessa forma, “vence” o combate quem levar os pontos de vida do oponente para zero primeiro acertando golpes.

6.6 Projeto de fases (*level design*)

Fase 1 (S.N.)

- a. Sinopse: O jogador começa sua jornada e deve encontrar uma forma de se aprofundar na caverna enquanto luta contra inimigos.
- b. Material introdutório: Texto que contextualiza o jogo (motivação do personagem principal e local).
- c. Objetivos: Encontrar a saída da fase, a qual é representada por uma sala específica com uma porta no centro, gerada em algum ponto da masmorra.
- d. Descrição visual e física: Ambiente escuro e úmido composto por pedras e corais, além de portas que levam para outras salas e eventuais aberturas no chão para pescar.
- e. Mapa da fase: Salas pré-fabricadas são posicionadas e conectadas de forma procedural ao carregar o nível, gerando inúmeras possibilidades de mapas.
- f. Detalhamento de caminhos críticos: O mapa é gerado proceduralmente, portanto, os caminhos mudam toda nova tentativa.
- g. Encontros relevantes: Salas de inimigos (o jogador deve derrotar todos para continuar), salas de pesca (o jogador pode interagir com a área no centro da sala para iniciar o minijogo) e sala final (contém uma zona de pesca e a porta para o próximo nível).
- h. Gatilhos: Entrar em uma sala de inimigos fecha as portas até que todos sejam derrotados, encostar nas portas transporta o jogador para a próxima sala (ou para o próximo nível, caso seja a porta da sala final) e interagir com zonas de pesca inicia o minijogo.
- i. Walkthrough: Tentar minimizar o dano tomado e encontrar zonas de pesca para ficar mais forte enquanto procura por uma sala em formato de “mais” (+) com uma porta no centro e uma zona de pesca. O mapa é procedural, então não existe forma de memorizar o caminho de tentativas anteriores.
- j. Material de fechamento: O jogador é levado diretamente para a próxima fase.

Fase 2 (S.N.)

- a. Sinopse: o jogador é emboscado e se encontra cercado de ondas de inimigos que devem ser derrotados.

- b. Material introdutório: Nenhum, a fim de testar o foco do jogador quando a transição de fase ocorrer e imergi-lo na ideia da emboscada.
- c. Objetivos: Sobreviver e derrotar todos os inimigos.
- d. Descrição visual e física: Ambiente mais claro e com coloração avermelhada, composto apenas por pedras, lembrando uma arena.
- e. Mapa da fase: Arena quadrada e cercada por parede.
- f. Detalhamento de caminhos críticos: Não existem caminhos, apenas uma sala aberta.
- g. Encontros relevantes: Três ondas de inimigos aparecem tentando derrotar o jogador. A primeira onda é composta por três inimigos, a segunda é composta por cinco e a terceira é composta por sete.
- h. Gatilhos: Imediatamente, a primeira onda se inicia até que todos os inimigos sejam derrotados. Quando uma onda finaliza, a próxima começa. Ao finalizar a terceira onda, a fase será concluída.
- i. Walkthrough: Faça o máximo para evitar ficar preso nos cantos da arena e foque em atacar os inimigos mais próximos enquanto se movimenta em grandes círculos na fase. Eventualmente, todos serão derrotados e a fase finalizada.
- j. Material de fechamento: Um texto agradecendo ao jogador e o parabenizando por completar esta versão do jogo.

6.7 Projeto de interface

6.7.1 Sistema visual

- a. HUD (*Head-Up Display*): A vida do jogador é representada por corações no canto superior esquerdo. Ao interagir com zonas de pesca, o mini game aparece à direita da tela, representado por duas barras: uma em que o jogador deve manter a área verde junta do peixe e outra que representa o progresso. Em versões futuras, será adicionado um minimapa no canto superior direito, para auxiliar na navegação, e uma área que mostra os peixes/artefatos coletados pelo jogador durante a pesca.
- b. Menu *in game*: Em seu estado atual, o jogo não possui menus durante a partida. Em versões futuras, será adicionado um menu, acessível pela tecla ESC, no qual o jogador poderá configurar seu áudio, por exemplo, ou abandonar a tentativa e voltar para o menu inicial.

- c. Câmera: Terceira pessoa, *top down*, focando o jogador no centro.

6.7.2 Sistema de controle

O mouse é utilizado exclusivamente para interagir com os elementos do menu inicial e não apresenta nenhuma funcionalidade durante as partidas. Em versões futuras, o mouse será utilizado também para interagir com menus *in game* e visualizar efeitos de itens e minimapa. O teclado é utilizado para controlar todo o restante do jogo. Não existe suporte para outros esquemas de controle.

6.7.3 Fluxo de telas

Menu principal: tela inicial na qual o jogador pode iniciar uma partida, verificar os controles, ajustar opções, visualizar créditos ou fechar o jogo.

Tutorial: Parte do menu principal, apresenta para o jogador os controles do jogo.

Opções: Parte do menu principal, permite que o jogador ajuste opções do jogo. Nesta versão, é possível ajustar os volumes geral (master), efeitos e/ou música.

Créditos: Parte do menu principal, apresenta informações do jogo, como nome, desenvolvedor, orientador e *assets* de terceiros, como as músicas, efeitos sonoros e fonte dos textos.

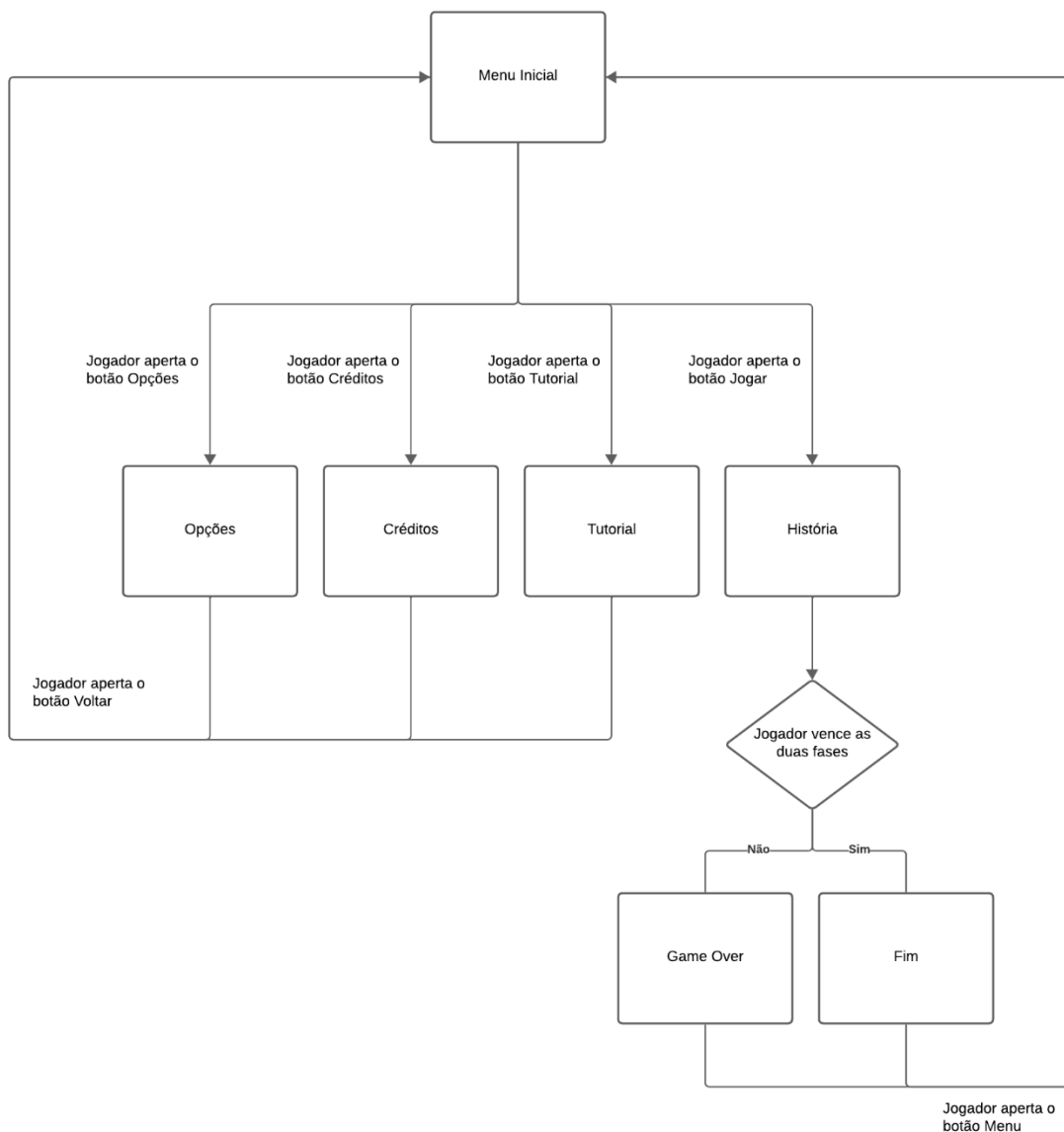
História: É mostrada entre o menu inicial e a primeira fase quando o jogador seleciona o botão de jogar. Serve para contextualizar os eventos e localização ao jogador.

Game Over: Mostrada quando o jogador perde todas as vidas durante qualquer fase. Indica a necessidade de recomeçar e possui um botão para o menu inicial.

Fim: Mostrada quando o jogador completa a segunda fase. Parabeniza e agradece o jogador por jogar. Possui um botão para o menu inicial.

Figura 1 - Fluxograma de telas

Fluxograma de Telas Deep Cave Catch



Fonte: Desenvolvido pelo autor (2026).

6.7.4 Opções do jogo

O jogo não apresenta escolha de dificuldade. A dificuldade depende do progresso do jogador na masmorra: quanto mais o jogador avança, mais complicados serão os combates.

6.7.5 *Saving e checkpoints*

O jogo, em sua versão atual, não possui pontos de salvamento da partida (*run*). Em versões futuras, serão adicionados pontos de salvamento entre andares, permitindo que o jogador possa salvar e continuar sua tentativa mais tarde.

6.7.6 Sistema de ajuda

O único sistema de ajuda presente é a tela de tutorial, acessível através do menu inicial, reforçando o incentivo ao jogador explorar e descobrir mecânicas e padrões por conta própria enquanto melhora no jogo através de seu aprendizado.

6.8 Projeto de som

6.8.1 Descrição geral

O jogo apresenta, em sua maioria, sons e músicas que remetem a jogos clássicos e que combinam com a estética *pixel art*, escolhidos através de pesquisas em bibliotecas de som online, portanto criados por terceiros e disponíveis para todos.

6.8.2 Trilhas sonoras

- a. Menu e tela de vitória: *Calm background Music* by Bertsz, disponível em: <https://freesound.org/s/671900/>
- b. Ambiente das fases: *8-Bit Bossfight* by Volvion, disponível em: <https://freesound.org/s/265308/>

6.8.3 Efeitos sonoros

- a. Seleção no menu: *Menu Select* by pumodi, disponível em: <https://freesound.org/s/150222/>
- b. Ataques: *Melee Weapon Swings* by deleted_user_3656686, disponível em: <https://freesound.org/s/215160/>

- c. Derrota: *Negative_beeps.wav by themusicalnomad*, disponível em: <https://freesound.org/s/253886/>
- d. Falha na pesca: *Retro video game sfx - Fail by OwlStorm*, disponível em: <https://freesound.org/s/404743/>
- e. Sucesso na pesca e vitória no jogo: *Completed.wav by Kenneth_Cooney*, disponível em: <https://freesound.org/s/609336/>
- f. Dano: *Short, Retro Attacking Sound by IndyArts*, disponível em: <https://freesound.org/s/777567/>
- g. Pesca: *Fishing reel winding by 1bob*, disponível em: <https://freesound.org/s/831928/>

6.9 Sistema de inteligência artificial

A Inteligência Artificial, no desenvolvimento de jogos digitais, define o comportamento dos personagens não-jogáveis. Segundo Millington (2019), o propósito da IA nesse espaço é a criação de comportamentos capazes de gerar a ilusão de inteligência e tomada de decisões.

O inimigo presente (peixe com lança) foi desenvolvido para representar o desafio mais básico. Suas ações se restringem a perseguir o jogador até chegar perto o suficiente e atacar. Programado utilizando máquina de estados finitos, esse inimigo começa em um estado de perseguir o jogador até que se aproxime o suficiente, quando troca para o estado de ataque. Nesse estado, o peixe se torna estático por um momento antes de desferir um golpe na direção do jogador, verificando então, baseado em sua distância, se deve continuar desferindo golpes (jogador próximo) ou se deve voltar ao estado de perseguição (jogador distante).

7 ARTE

7.1 Guia de cores e estilos gráficos

O estilo gráfico adotado foi *pixel art* buscando transmitir a sensação de jogos *retrô*.

A primeira fase utiliza uma paleta de cores voltadas ao cinza e azul, refletindo o isolamento da caverna à beira do mar, e algumas cores vibrantes destacando corais, o que indica vida no local. A segunda fase utiliza as mesmas *tiles* de ambiente que o nível anterior, porém, adotando tonalidade avermelhada, representando a situação de perigo imediato que o jogador se encontra.

Os inimigos compartilham da escolha de cores da primeira fase, transmitindo a ideia de que aquele é seu ambiente, seu lar, enquanto o jogador possui uma paleta de diversas cores vibrantes, se destacando do ambiente e evidenciando que não pertence àquela caverna.

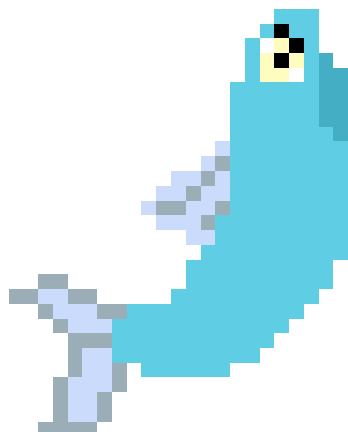
7.2 Arte conceitual

Figura 2 - Arte conceitual do personagem jogável



Fonte: Desenvolvido pelo autor (2026).

Figura 3 - Arte conceitual do inimigo



Fonte: Desenvolvido pelo autor (2026).

7.3 *Asset List*

7.3.1 Personagens

Figura 4 – Personagem jogável (principal)



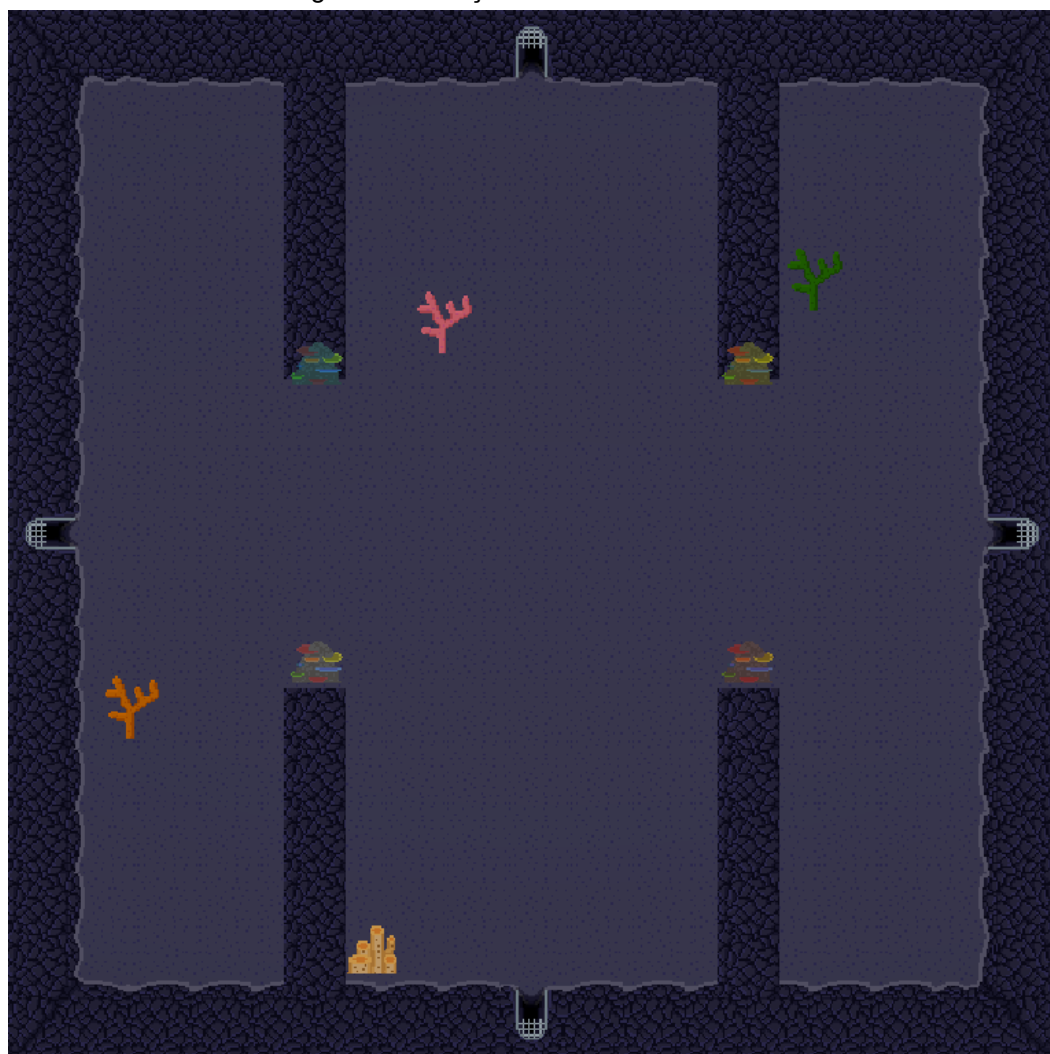
Fonte: Desenvolvido pelo autor (2026).

Figura 5 - Inimigo (peixe com lança)



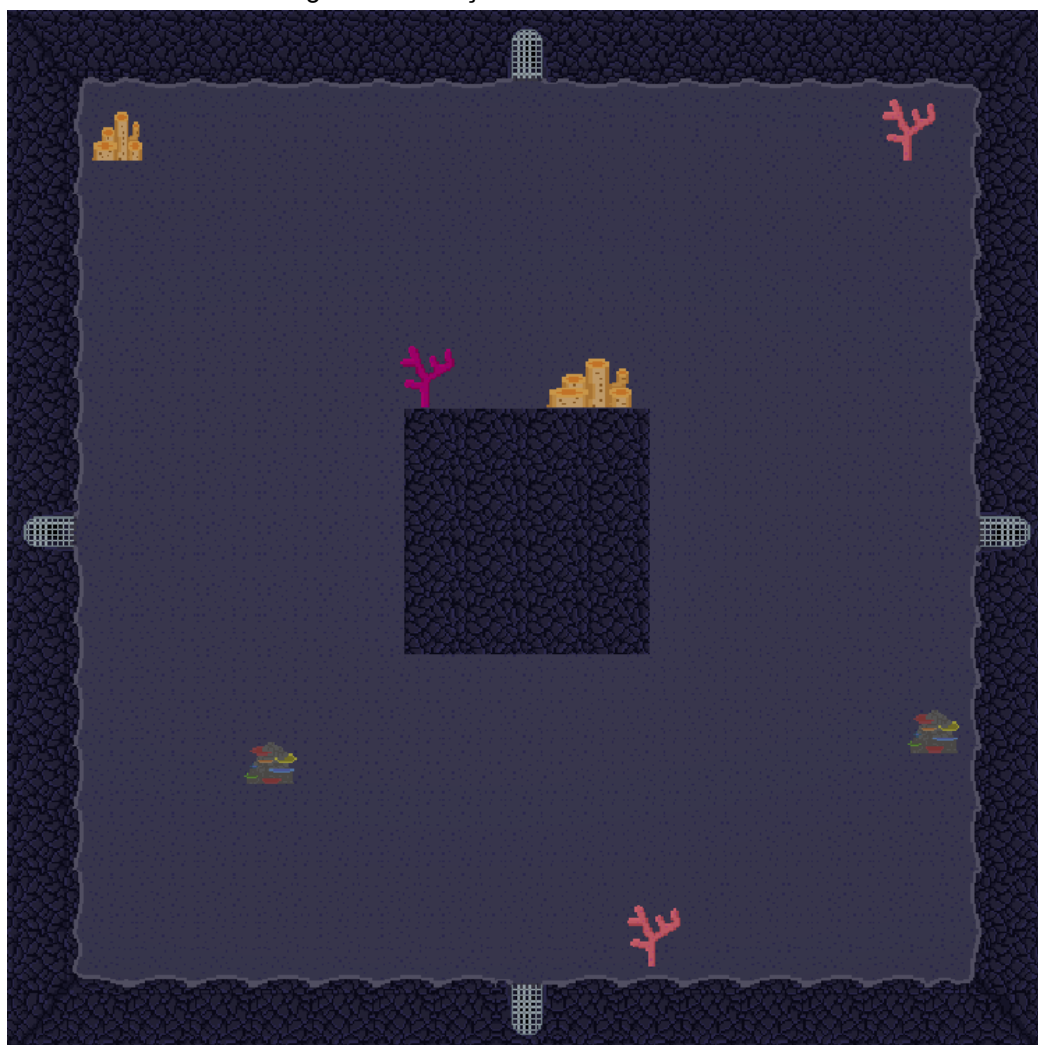
Fonte: Desenvolvido pelo autor (2026).

Figura 7 - Variação 1 de sala de combate



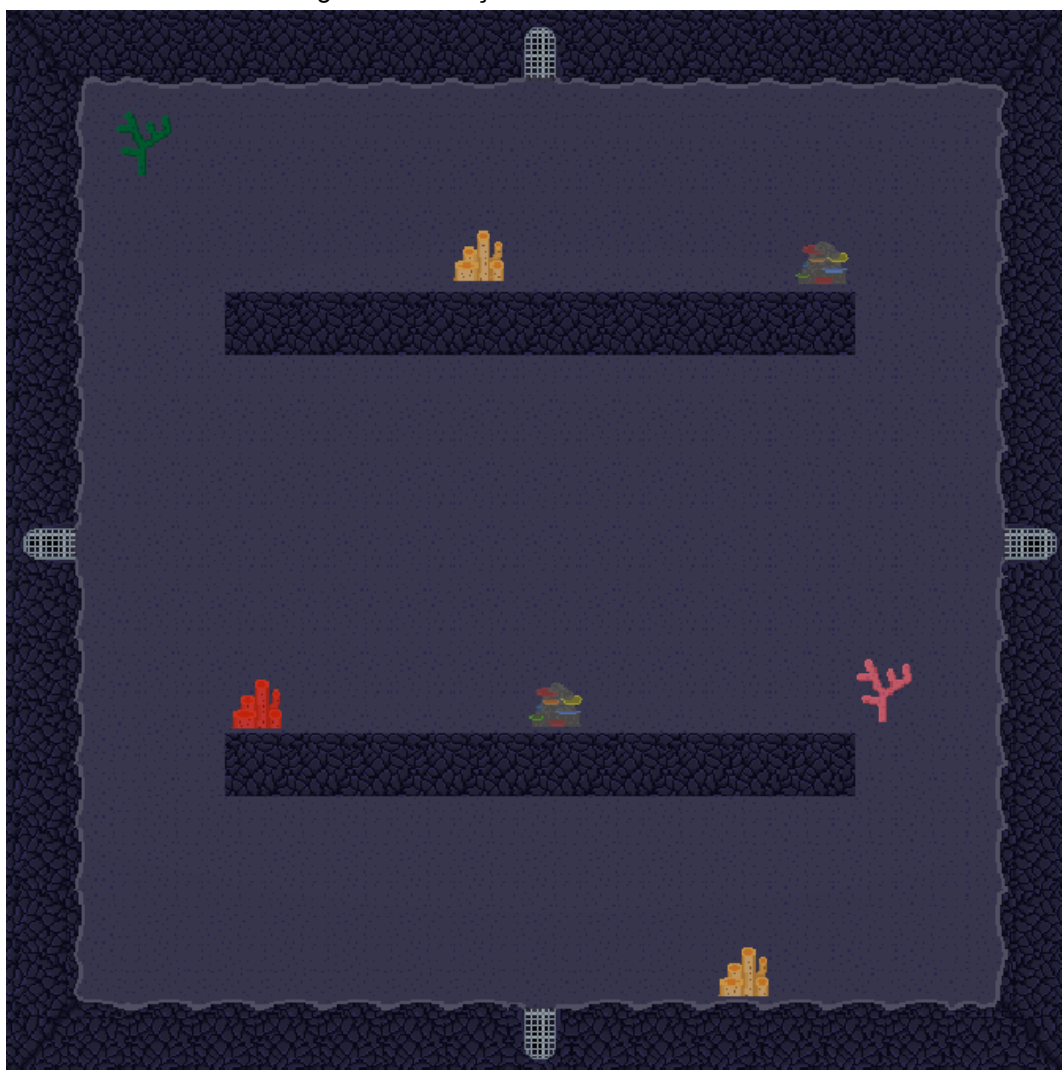
Fonte: Desenvolvido pelo autor (2026).

Figura 8 - Variação 2 de sala de combate



Fonte: Desenvolvido pelo autor (2026).

Figura 9 - Variação 3 de sala de combate



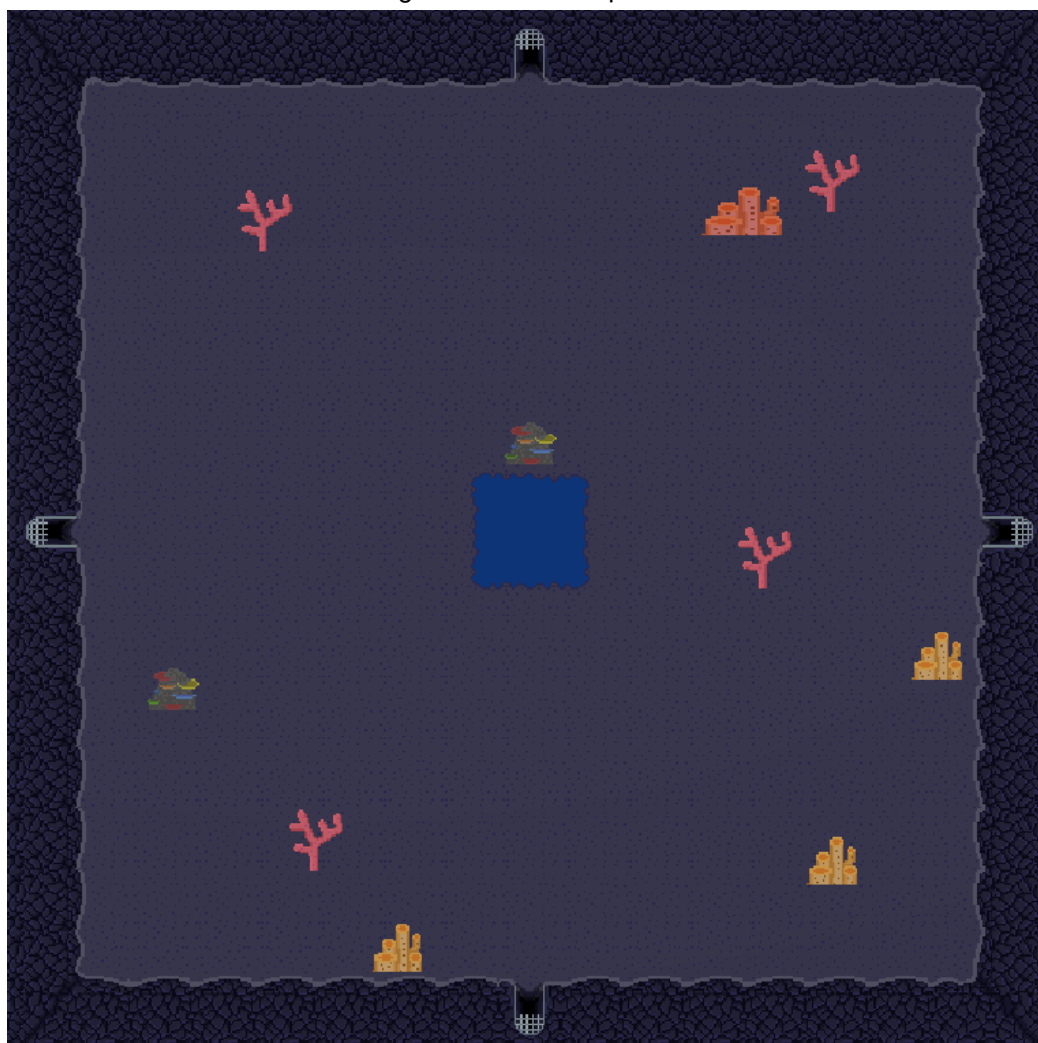
Fonte: Desenvolvido pelo autor (2026).

Figura 10 - Variação 4 de sala de combate



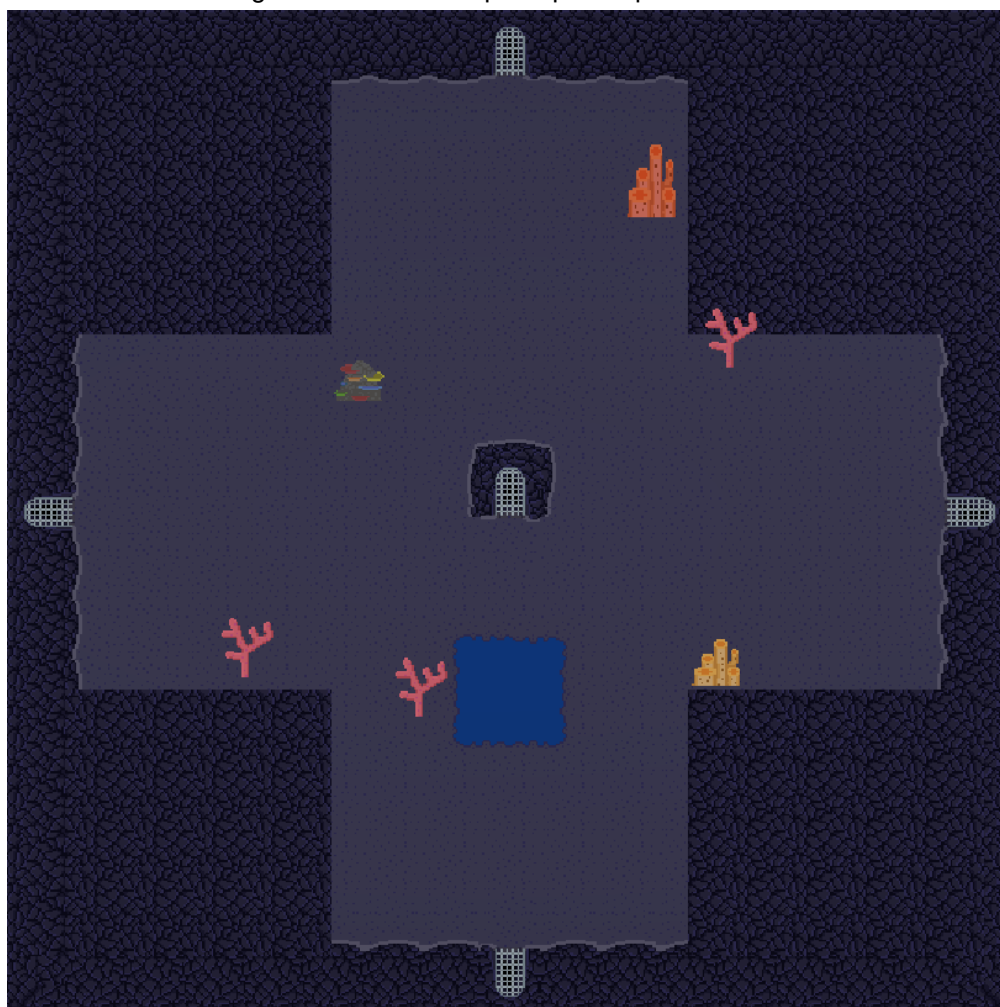
Fonte: Desenvolvido pelo autor (2026).

Figura 11 - Sala de pesca



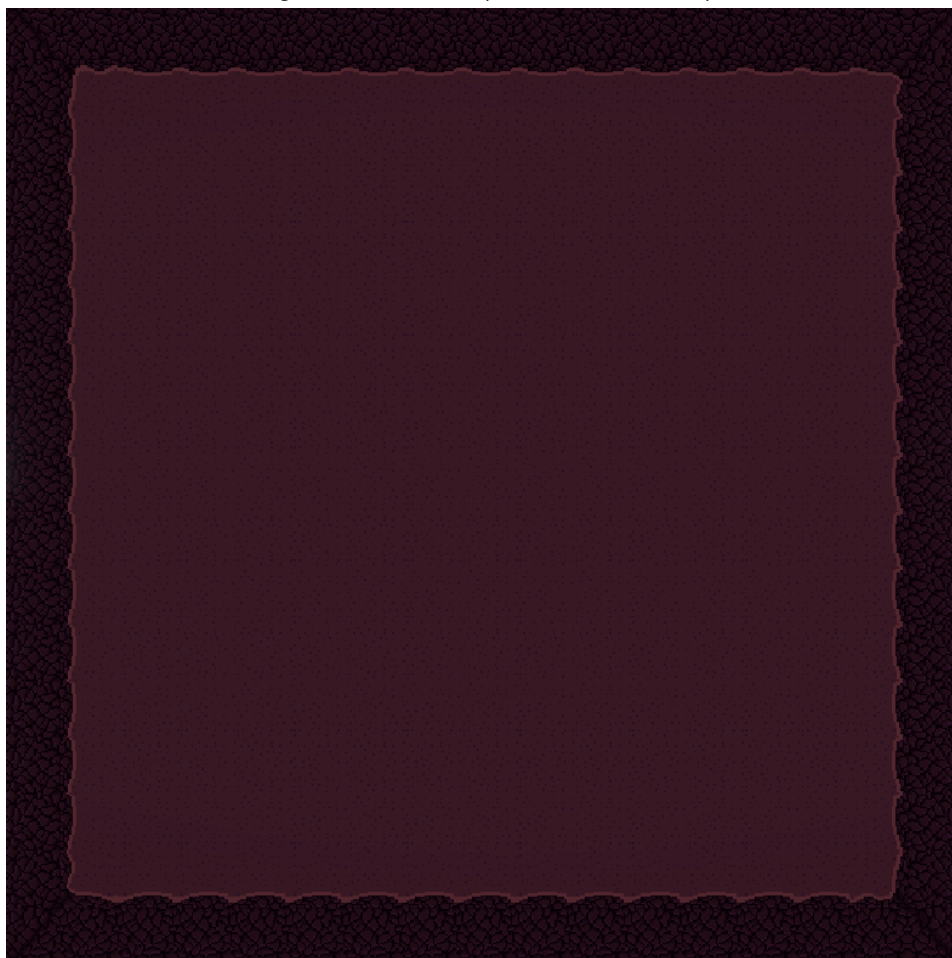
Fonte: Desenvolvido pelo autor (2026).

Figura 12 - Sala com porta para a próxima fase



Fonte: Desenvolvido pelo autor (2026).

Figura 13 - Fase 2 (arena de combate)



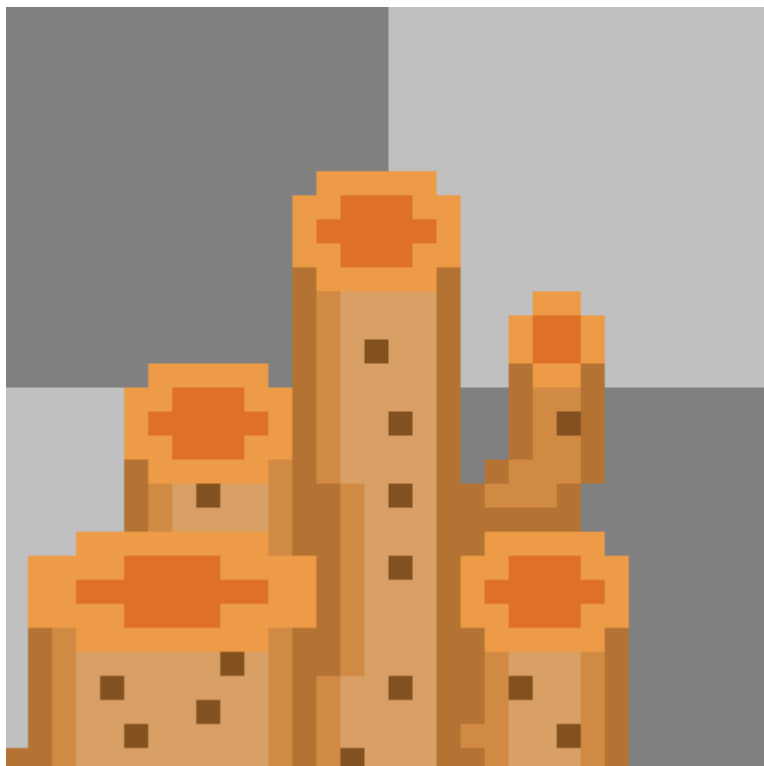
Fonte: Desenvolvido pelo autor (2026).

Figura 14 - Variação 1 de coral



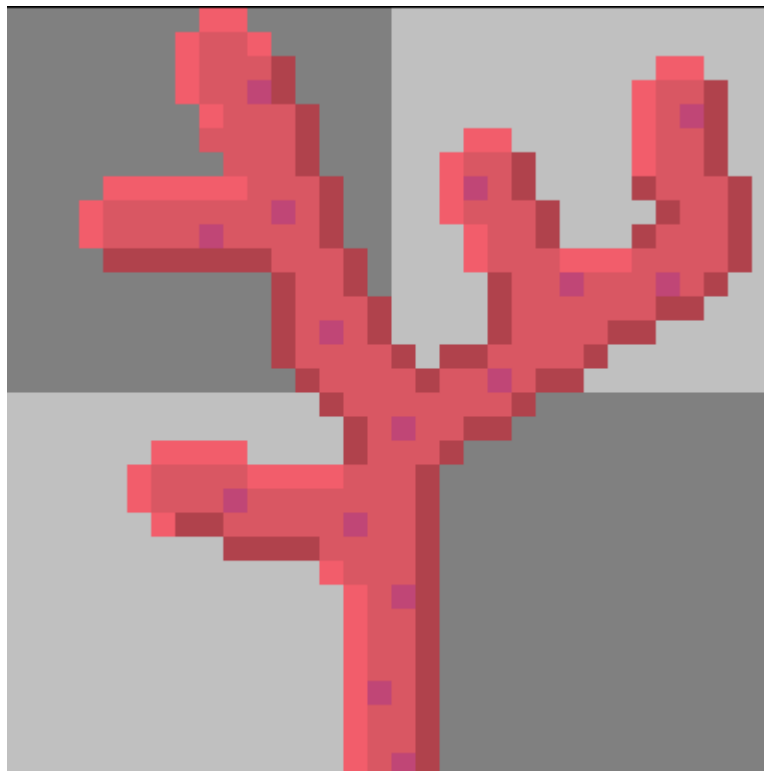
Fonte: Desenvolvido pelo autor (2026).

Figura 15 - Variação 2 de coral



Fonte: Desenvolvido pelo autor (2026).

Figura 16 - Variação 3 de coral



Fonte: Desenvolvido pelo autor (2026).

7.3.3 Animações

Figura 17 – *Sprite sheet* da animação *idle* do jogador



Fonte: Desenvolvido pelo autor (2026).

Figura 18 - *Sprite sheet* da animação de andar do jogador



Fonte: Desenvolvido pelo autor (2026).

Figura 19 - *Sprite sheet* da animação de ataque do jogador



Fonte: Desenvolvido pelo autor (2026).

Figura 20 - *Sprite sheet* da animação de andar do inimigo



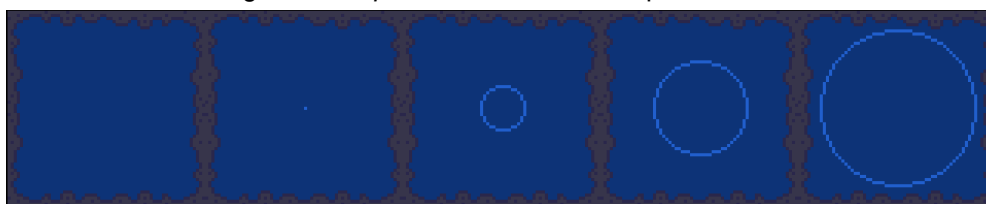
Fonte: Desenvolvido pelo autor (2026).

Figura 21 - *Sprite sheet* da animação de atacar do inimigo



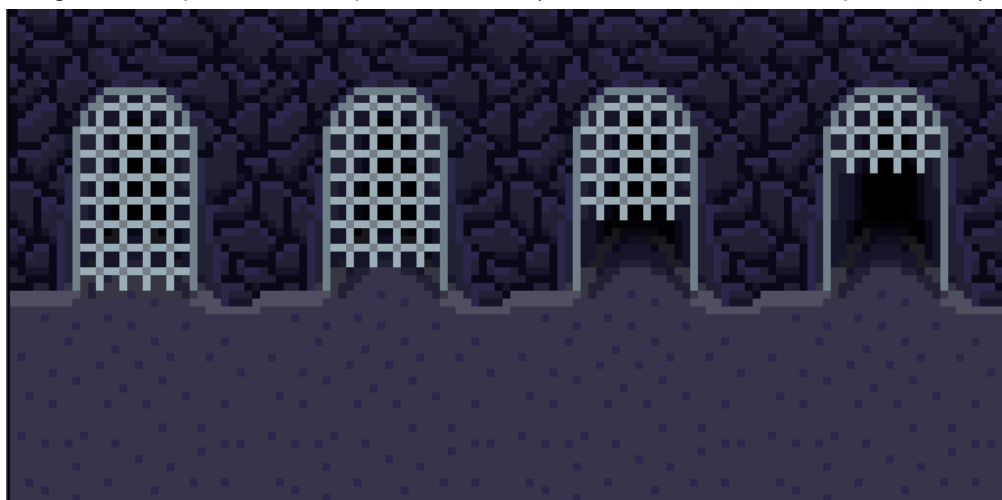
Fonte: Desenvolvido pelo autor (2026).

Figura 22 - *Sprite sheet* da zona de pesca ativa



Fonte: Desenvolvido pelo autor (2026).

Figura 23 - *Sprite sheet* do portão abrindo (a ordem inversa é usada para fechar)



Fonte: Desenvolvido pelo autor (2026).

Figura 24 - *Sprite sheet* do efeito de ataque



Fonte: Desenvolvido pelo autor (2026).

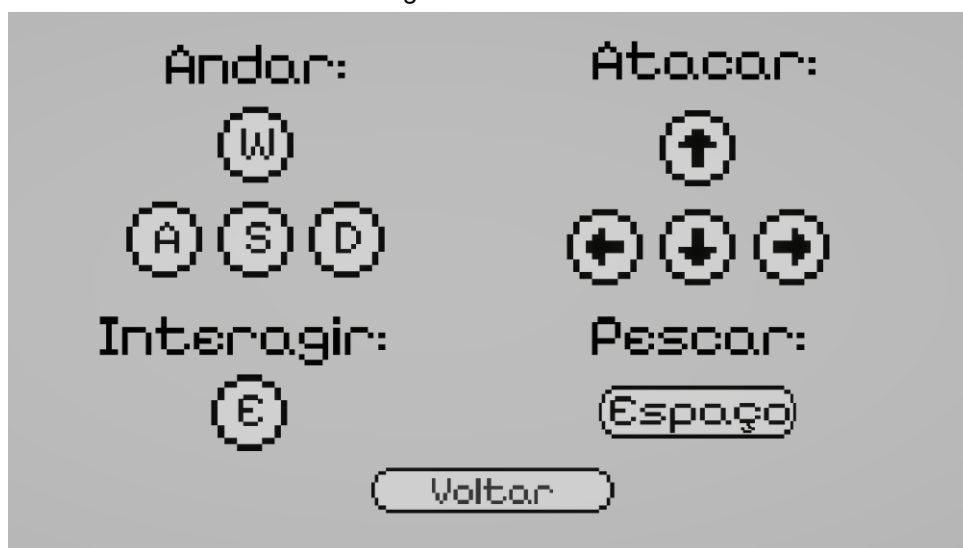
7.3.4 Interfaces

Figura 25 - Menu inicial



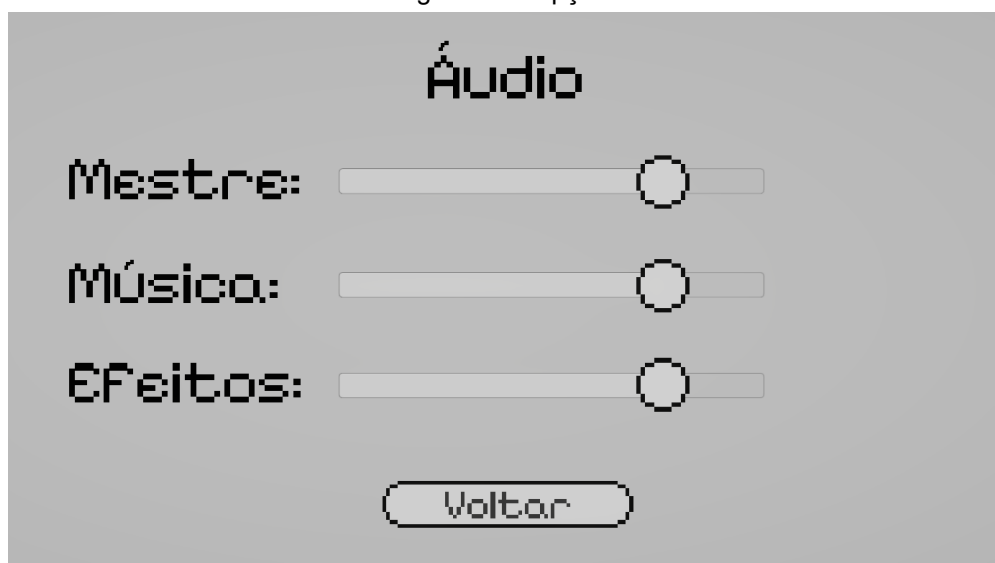
Fonte: Desenvolvido pelo autor (2026).

Figura 26 - Tutorial



Fonte: Desenvolvido pelo autor (2026).

Figura 27 - Opções



Fonte: Desenvolvido pelo autor (2026).

Figura 28 - Créditos



Fonte: Desenvolvido pelo autor (2026).

Figura 29 - Indicador de vida do jogador (vida cheia)



Fonte: Desenvolvido pelo autor (2026).

Figura 30 - Indicador de vida do jogador (após levar 2 de dano)



Fonte: Desenvolvido pelo autor (2026).

8 DOCUMENTAÇÃO TÉCNICA

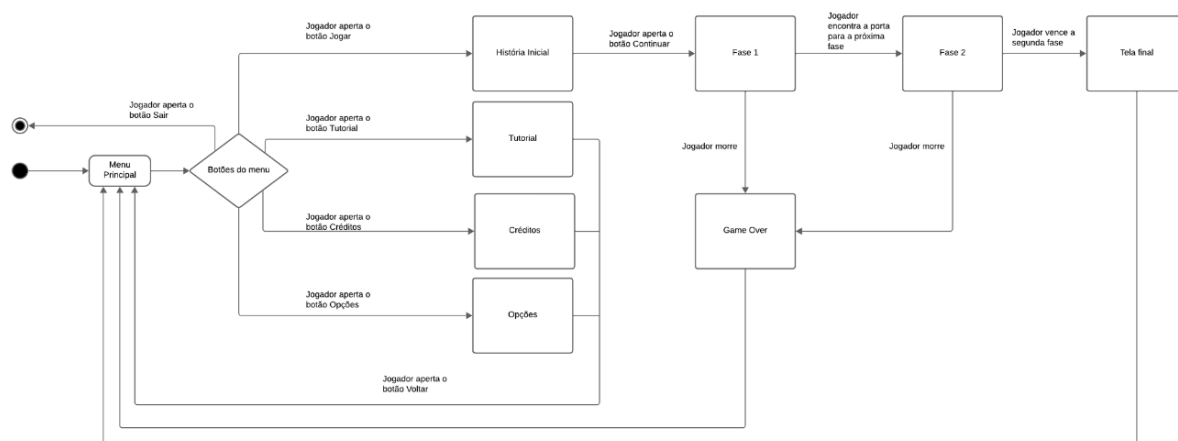
8.1 Requisitos de sistema

- Os requisitos mínimos de sistema para a execução do jogo são:
- Sistema Operacional: Windows 7 ou superior.
- Processador: Intel Core 2 Duo E8400 (3.0 GHz) / AMD Athlon 64 X2 6000+ (3.0 GHz) ou qualquer processador de arquitetura dual-core equivalente ou superior.
- Memória RAM: 2 GB.
- Placa de vídeo: Qualquer solução de vídeo com suporte a Direct X12.
- Armazenamento: 150 MB de espaço disponível.

8.2 Engenharia de *software*

Figura 31 - Diagrama de estados do jogo

Estados principais



Fonte: Desenvolvido pelo autor (2026).

8.3 *Procedures e Patterns*

Além da orientação a objetos, padrão da *game engine* Unity (pela qual o projeto foi desenvolvido), os códigos utilizam dois padrões principais: *Singletons* e Máquinas de Estados Finitos.

O *singleton* garante que certos scripts, como o *Game Manager* (responsável por manter referências e variáveis importantes ao funcionamento do jogo), permitam seu acesso de forma fácil e global por outras partes do código e garante que apenas uma instância exista por vez, impedindo duplicidade de informações, as quais podem resultar em erros durante a execução.

Já a máquina de estados finitos é utilizada para definir a execução de diferentes estados mutuamente exclusivos no código. Usado principalmente na programação do jogador e do inimigo, garantindo ações como "andar" ou "atacar" sejam independentes e não interfiram umas com as outras.

8.4 *Game engine*

A Unity é uma *engine* de jogos multiplataforma desenvolvida pela *Unity Technologies* que fornece um ambiente completo para o desenvolvimento de inúmeros tipos de jogos por implementar nativamente componentes e sistemas base, como física e renderização 2D e 3D.

O fluxo de trabalho dentro dessa *engine* se baseia na arquitetura orientada a componentes. Nesse modelo, qualquer elemento presente na cena do jogo é um objeto vazio base (chamado *GameObject*). O comportamento desse objeto é determinado pela associação de componentes específicos a ele, por exemplo, acoplando o componente *RigidBody* a fim de ser incluído pelos cálculos de física da própria *engine*.

O controle detalhado desses objetos ocorre por meio de *scripts* escritos na linguagem C#. Ao criar um *script* personalizado derivado da classe base (*MonoBehaviour*), a lógica que dita como suas variáveis e opções de personalização dos componentes devem se comportar pode ser programada. Esse *script*, por sua vez, passa a funcionar como um novo componente modular, o qual pode ser acoplado a qualquer objeto, permitindo que execute ações em resposta a eventos específicos,

como a leitura de comandos do teclado, a colisão entre elementos, a passagem do tempo etc.

8.5 Programação e *Scripting*

A linguagem de programação utilizada foi C# (*C sharp*): linguagem orientada a objetos e fortemente tipada desenvolvida pela Microsoft. Essa escolha se deu por ser a linguagem nativa de *scripting* utilizada na *game engine* Unity.

9 GERENCIAMENTO E ANÁLISE DO PROJETO

9.1 Organização inicial do projeto

Por se tratar de um trabalho de conclusão de curso individual, a equipe de desenvolvimento do projeto é composta exclusivamente pelo autor, que assume o papel de desenvolvedor principal, portanto se concentram as funções de programação, arte, *game design*, e som.

Apesar da execução centralizada, o projeto conta com partes interessadas que atuam de forma direta e indireta no direcionamento do projeto. A principal parte interessada interna é o professor orientador, José William Pinto Gomes, responsável pela supervisão metodológica, validação do cronograma e direcionamento das decisões de *design* técnico perante os critérios acadêmicos da instituição.

Como partes interessadas externas, identificam-se a banca examinadora e os jogadores que participaram da fase de teste beta. A banca examinadora atua como o órgão avaliador final, validando o rigor científico e a maturidade tecnológica do projeto. Já os jogadores do teste desempenham o papel de validadores da experiência de usuário, cujos *feedbacks* coletados via questionário norteiam o refinamento do jogo.

9.2 Escopo do projeto

O jogo desenvolvido, *Deep Cave Catch*, é um *Roguelike* composto de duas fases jogáveis, sendo a primeira construída utilizando geração procedural através do posicionamento de salas pré fabricadas em uma matriz, totalizando 15 salas por geração de dungeon. As possíveis salas são divididas em quatro grupos: uma sala inicial, quatro *layouts* de salas contendo inimigos, uma sala contendo a zona de pesca e uma sala final com a porta para o segundo nível e uma zona de pesca. A segunda fase é construída como uma arena de combate, na qual três ondas de inimigos surgem. Um tipo de inimigo foi desenvolvido para antagonizar o jogador em combate.

Para a pesca, foi desenvolvido um mini jogo, o qual cura o jogador no momento da interação e aumenta seus atributos de dano, velocidade de ataque ou velocidade de movimento, aleatoriamente, ao suceder.

Além do jogo, foi desenvolvido este relatório a fim de documentar todas as etapas de sua criação, além de detalhar processos e resultados.

9.3 Recursos

Por se tratar de um projeto desenvolvido individualmente, a equipe foi composta pelo autor. Foram utilizados: um computador pessoal operando windows, a *game engine* gratuita Unity, a ferramenta paga Aseprite (para o desenvolvimento da arte) e bibliotecas gratuitas de som para as trilhas e efeitos sonoros.

9.4 Entregas

Duas entregas principais resultaram desse projeto: este relatório de desenvolvimento técnico, que abrange as características e etapas de desenvolvimento detalhadamente, e o jogo eletrônico *Deep Cave Catch*.

9.5 Premissas

Foram considerados verdadeiros, reais ou certos durante a elaboração do cronograma:

Disponibilidade da Infraestrutura: Assumiu-se que o computador pessoal de desenvolvimento e o fornecimento de energia elétrica e internet permaneceram estáveis e operacionais durante o período de janeiro até maio.

Acesso às Ferramentas: Considerou-se que a *game engine* Unity e o *software* Aseprite manteriam suas licenças de uso ativas e sem atualizações de software que quebrassem a compatibilidade do código.

Proficiência Técnica Prévia: Assumiu-se que o conhecimento prévio do autor na linguagem C# e na lógica de componentes da Unity seria suficiente para implementar os sistemas lógicos sem a necessidade de passar por curvas longas de aprendizado inicial.

9.6 Restrições

Os fatores limitadores foram:

Prazo Improrrogável: O projeto possui uma restrição de tempo vinculada ao calendário de entregas acadêmicas da instituição, impossibilitando extensões de prazo e restringindo o desenvolvimento desse projeto no limite de 5 meses.

Limitação de Recursos Humanos: Por se tratar de um desenvolvimento individual, a capacidade de trabalho diária ficou restrita às horas disponíveis do autor, impossibilitando a execução de tarefas paralelas.

Restrição Orçamentária: O orçamento financeiro total foi limitado ao capital próprio preexistente do autor.

Escopo Fechado: Qualquer funcionalidade que demandar tecnologias externas a base da Unity (como servidores de rede ou APIs de terceiros) foi desconsiderada para não comprometer a entrega.

9.7 Cronograma

Quadro 2 - Cronograma.

Atividade	Início (Est.)	Fim (Est.)	Início (Real)	Fim (Real)
Pesquisa Bibliográfica	Mês 1 / Sem 1	Mês 1 / Sem 3	Mês 1 / Sem 1	Mês 1 / Sem 3
<i>Game Design</i>	Mês 1 / Sem 3	Mês 2 / Sem 1	Mês 1 / Sem 3	Mês 2 / Sem 2
Programação da Movimentação do Jogador	Mês 2 / Sem 1	Mês 2 / Sem 2	Mês 2 / Sem 2	Mês 3 / Sem 1
Programação do Mapa Procedural	Mês 2 / Sem 2	Mês 3 / Sem 2	Mês 3 / Sem 1	Mês 3 / Sem 4
Programação do Sistema de Combate e UI	Mês 3 / Sem 2	Mês 4 / Sem 1	Mês 3 / Sem 3	Mês 4 / Sem 1
Programação do Inimigo	Mês 4 / Sem 1	Mês 4 / Sem 2	Mês 4 / Sem 1	Mês 4 / Sem 3

Criação da Segunda Fase (Arena)	Mês 4 / Sem 2	Mês 4 / Sem 4	Mês 4 / Sem 3	Mês 4 / Sem 4
Teste Beta (Questionários)	Mês 4 / Sem 4	Mês 5 / Sem 2	Mês 5 / Sem 4	Mês 5 / Sem 4
Análise de Resultados e Conclusão	Mês 5 / Sem 2	Mês 5 / Sem 4	Mês 5 / Sem 4	Mês 5 / Sem 4

Fonte: Desenvolvido pelo autor (2026).

9.8 Estimativa de custos

O projeto foi realizado utilizando ferramentas e fontes de *assets* gratuitos, como a *engine* Unity e a bibliotecas online de sons, com exceção da arte, para qual foi utilizado o *software* previamente adquirido Aseprite, totalizando um custo de produção aproximado de R\$40,00.

9.9 Limites do projeto

Estão explicitamente incluídos no projeto: o desenvolvimento da base do jogo Deep Cave Catch, o que abrange sua compatibilidade apenas com a plataforma Windows, controles limitados ao mouse e teclado, jogabilidade *singleplayer*. Além disso, foram desenvolvidos a geração procedural da primeira fase e a forma fixa da segunda, um tipo de inimigo, o personagem principal e sua programação, um menu que contempla o tutorial, as opções de áudio, os créditos e os botões de jogar e sair. Também estão inclusos trilhas e efeitos sonoros conforme necessário às ocasiões do jogo.

A fim de limitar o escopo do projeto para refletir o período limite desenvolvimento, estão excluídos do projeto: suporte para múltiplas plataformas e esquemas de controle, além de qualquer jogabilidade *multiplayer*. Além disso, foram excluídos personagens não jogáveis neutros ou amigáveis, sistemas de diálogo e cinemáticas.

9.10 Processos

Para garantir a entrega do jogo *Deep Cave Catch* dentro do prazo limite, foram selecionados processos de gerenciamento de projetos baseados no ciclo de vida clássico (Iniciação, Planejamento, Execução, Controle e Encerramento), adaptados para um desenvolvimento individual. O nível de implementação de cada processo foi ajustado de forma a priorizar a agilidade, focando no controle do cronograma e na qualidade do jogo.

A aplicação prática desses processos ocorreu da seguinte forma:

Início e Planejamento: Nesta fase, foram definidos o escopo do produto, os limites do projeto, as premissas e o cronograma de atividades. Esta etapa foi crucial para controlar o projeto contra o aumento do escopo além do possível.

Execução: Foco na produção técnica modular das mecânicas, das artes em *pixel art*, da coleta de sons etc.

Controle: Utilização de lista de tarefas e cronograma para identificar e monitorar o progresso das atividades planejadas.

Encerramento: Este processo concentrou-se na execução concentrada do teste beta, na coleta e tabulação dos dados do questionário eletrônico e na elaboração deste relatório técnico.

9.11 Análise de riscos

Quadro 3 - Análise de riscos.

Identificação do Risco	Impacto	Probabilidade
Aumento de escopo	Alto	Média
Dificuldades na produção de arte	Médio	Alta
Falhas críticas no código procedural	Alto	Média

Falha de <i>hardware</i> ou perda de dados	Alto	Baixa
Falhas na implementação de mecânicas básicas	Médio	Média

Fonte: Desenvolvido pelo autor (2026).

9.12 Viabilidade técnica

A viabilidade técnica para o desenvolvimento do jogo *Deep Cave Catch* é sustentada pela robustez das ferramentas de *software* escolhidas, alinhadas à experiência prática documentada na literatura de desenvolvimento de jogos. O uso da *game engine* Unity constitui a principal base técnica, uma vez que implementa, nativamente, sistemas base de física e renderização, por exemplo. Essa integração elimina a necessidade de construir subsistemas básicos do zero, permitindo o foco na lógica do *loop* de jogabilidade, otimizando o tempo de produção dentro do cronograma acadêmico.

De forma complementar, a linguagem C# oferece ao projeto estabilidade por meio de sua característica fortemente tipada e do gerenciamento automático de memória via *Garbage Collector*, o que reduz drasticamente a incidência de falhas críticas em tempo de execução (*crashes*). A modularidade da arquitetura orientada a objetos da Unity foi potencializada pela aplicação prática de *design patterns* como *Singletons* para o gerenciamento global de estados (exemplo, *Game Manager*) e Máquinas de Estados Finitos (FSM) para o controle dos comportamentos de movimentação e ataque do protagonista e dos inimigos. A separação lógica fornecida por esses padrões impede o acoplamento excessivo de código, facilitando a depuração e garantindo que novas mecânicas, como o mini jogo de pesca, pudessem ser acopladas aos sistemas pré implementados.

Por fim, a viabilidade técnica é ratificada pelo controle de escopo e pela ausência de dependências tecnológicas complexas. Ao excluir recursos de alta complexidade de implementação, como arquiteturas de rede para multijogador, o projeto eliminou os principais gargalos técnicos que poderiam inviabilizar sua execução.

9.13 Viabilidade econômica

A viabilidade econômica do projeto é demonstrada pelo modelo de desenvolvimento de baixo custo financeiro direto. O projeto foi planejado para minimizar investimentos de capital, concentrando os fatores de produção na força de trabalho do próprio autor. A escolha de ferramentas que oferecem licenças gratuitas, como a *game engine* Unity, eliminou custos de *software* de desenvolvimento visando a sustentabilidade financeira do projeto desde a fase de concepção.

O único custo material direto foi a prévia aquisição do software Aseprite para a criação e animação dos elementos visuais em *pixel art*, totalizando um valor aproximado de R\$40,00. Esse investimento substituiu a necessidade de contratação de artistas externos ou a compra de pacotes de *assets* comerciais genéricos em lojas digitais. Adicionalmente, as necessidades de áudio foram supridas integralmente por meio do uso de bibliotecas online de sons distribuídos sem custos financeiros.

Do ponto de vista de infraestrutura, não houve custos com *hardware* (uso de computador pessoal do autor) ou com hospedagem de servidores ou serviços de nuvem complexos, uma vez que o escopo delimitou o jogo como uma experiência estritamente *singleplayer* e local. Por fim, considerando que o jogo/projeto foi produzido como um trabalho de conclusão de curso sem fins de publicação comercial imediata, a ausência de orçamentos para *marketing*, taxas de publicação em lojas digitais (como Steam) e manutenção pós-lançamento ratifica a viabilidade econômica do projeto, provando que um produto de *software* funcional pode ser entregue com baixo investimento financeiro.

9.14 Plano de testes e controle de qualidade

O controle de qualidade e os testes de mecânicas do jogo foram estruturados por meio de abordagem prática e iterativa, focando na experiência do usuário e no funcionamento dos sistemas. Por se tratar de um projeto individual com cronograma delimitado, o controle de qualidade concentrou-se em testes funcionais contínuos pelo autor e um teste beta para a coleta de *feedbacks*.

O processo de verificação foi executado em duas etapas principais: Testes Funcionais Iterativos e teste beta.

Os testes funcionais foram conduzidos de forma contínua pelo próprio autor a cada nova mecânica implementada, simulando repetidamente tais mecânicas e suas interações com sistemas previamente implementados em ambientes de teste e, posteriormente, em situações reais durante um teste completo iniciado desde o menu até o fim do jogo.

O controle de qualidade durante o teste beta ocorreu por meio da aplicação subsequente de questionários eletrônicos. O foco foi avaliar a experiência dos usuários como um todo, verificando a forma e eficácia que diversos sistemas, como mecânicas de jogo, UI e trilha sonoras, interagem entre si.

9.15 Ferramentas e técnicas

As ferramentas utilizadas durante o projeto e que se mostraram adequadas foram a *game engine* Unity e o *software* Aseprite. A Unity apresenta licença gratuita e robustez excelente para a realização do desenvolvimento do jogo, fornecendo toda a base necessária. Já o Aseprite, embora não gratuito, foi adquirido previamente pelo autor e se tornou crucial para o desenvolvimento do *pixel art* como software especializado.

10 RESULTADOS

O desenvolvimento do projeto cumpriu com êxito os objetivos propostos neste trabalho, resultando na criação e documentação de um jogo digital funcional chamado *Deep Cave Catch*. A pesquisa e a execução prática demonstraram a viabilidade de combinar o gênero *Roguelike*, fundamentado em exploração de masmorras, geração procedural de mapas e morte permanente, com as dinâmicas de precisão características de mini jogos de pesca. Essa união de mecânicas opostas se provou eficiente no *game design*, estabelecendo uma alternância orgânica entre momentos de alta tensão e períodos de descanso ativo, o que contribuiu para a redução da fadiga cognitiva e o aumento do engajamento do jogador.

Dessa forma, este projeto atende às exigências científicas e tecnológicas de um trabalho de conclusão de curso, documentando o ciclo de vida do jogo digital desde a sua concepção conceitual até a validação com o público-alvo através do teste realizado.

10.1 Teste beta

O teste beta, realizado em 27 de maio na FATEC Americana, contou com 31 respostas de alunos de diferentes cursos da instituição.

A primeira parte da pesquisa consistiu em identificar o perfil dos participantes. O primeiro dado coletado foi o gênero, contando com 87,1% masculinos, 6,5% femininos, 3,2% não-binários e 3,2% que preferiram não responder. O segundo dado se refere à faixa etária, contemplando resultados desde menores de 19 anos até 34 anos: 41,9% entre 19 e 24 anos, 35,5% menores de 19 anos, 12,9% de 30 até 34 e 9,7% entre 25 até 29 anos. Finalmente, foi questionada a frequência com que os participantes jogam jogos digitais, onde 51,6% dizem jogar diariamente, 41,9% semanalmente e 6,5% mensalmente. Dessa forma, o perfil das pessoas que responderam a pesquisa foi definido.

A segunda parte da pesquisa foi referente a qualidade de diversos aspectos do jogo, com pontuações de 0 a 5. Quanto à opinião geral sobre o jogo, os resultados indicam que a linguagem utilizada está de acordo com a proposta e o jogo proporciona a diversão esperada, com resultados majoritariamente em 5 pontos e mínimos não

inferiores a 3. Aproximadamente $\frac{2}{3}$ dos participantes recomendam este jogo para amigos e parentes, indicando que o jogo atinge seu objetivo como fonte de entretenimento.

Em seguida, foi avaliado o design do jogo. Quanto à sua atratividade, a maioria dos participantes atribuiu notas 5 ou 4, com apenas 3 participantes atribuindo 2 ou 3. Quando questionados sobre a consistência de textos, cores e fontes, a maioria dos participantes pontuaram nota máxima, apenas um atribuiu nota 2 e o restante se dividiu entre 3 e 4. Por fim, nessa categoria, os participantes foram questionados quanto à clareza de informações cruciais, como vida, durante o jogo, e mais uma vez, a maioria dos participantes pontuou 5, com o resto se dividindo entre 3 e 4. Com base nessas informações, fica clara a viabilidade do *design* adotado, necessitando apenas de pequenos ajustes.

O próximo tópico avaliado foi a jogabilidade, no qual a maioria dos participantes atribuiu nota máxima para as primeiras duas perguntas, avaliando a existência e clareza do tutorial e a clareza das regras presentes. Porém, avaliando como o jogo oferece novos desafios e obstáculos, houve o surgimento de pontuações 1, indicando a necessidade do jogo por novos tipos de inimigos e obstáculos. Finalizando a categoria de jogabilidade, é possível perceber a necessidade de adicionar mais fases, ambientes e desafios para manter o jogo atrativo, pois mais da metade dos participantes atribuíram nota 4 ou menor. No geral, a jogabilidade atinge níveis satisfatórios para o projeto.

O quarto tópico avaliado foi referente ao menu e interação, recebendo em grande parte notas máximas quanto a clareza e facilidade do menu e da existência de informações sobre o jogo, indicando que a navegação do menu é intuitiva.

Por fim, a última categoria avaliada foi arte e som. A avaliação da arte e dos cenários apresentou resultados similares, nos quais os participantes atribuíram majoritariamente notas 5, enquanto o restante atribuiu notas entre 3 e 4. Porém, quando questionados sobre a trilha e efeitos sonoros, apesar da maioria ter considerado nota 5, 3 participantes atribuíram notas 1 e 2, indicando necessidade de melhoria de som no jogo.

No fim do questionário, os participantes também deixaram opiniões e sugestões de melhoria, as quais, cruzadas com os resultados da pesquisa, revelam pontos nítidos de trabalho para versões futuras: Adicionar mais variedades de inimigos e cenários, estendendo a duração do jogo, implementar um mini mapa, para auxiliar o

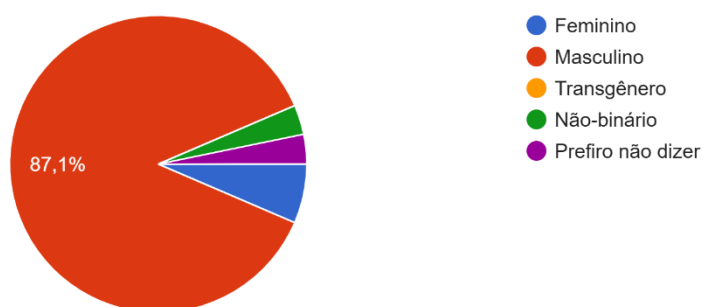
jogador na navegação da *dungeon*, mostrar status do jogador na HUD, como velocidade e dano, e ajustes gerais, como velocidade de ataque do inimigo, tamanho do texto presente na história inicial e ajustar a legibilidade da fonte em *pixel*.

Concluindo, o teste beta realizado se mostrou um sucesso, validando o *design* e execução do jogo e norteando alterações para futuras versões, a fim de alcançar um produto cada vez mais refinado e prazeroso para os jogadores.

Figura 32 – Gráfico de respostas à pergunta sobre gênero

Qual o seu Gênero?

31 respostas

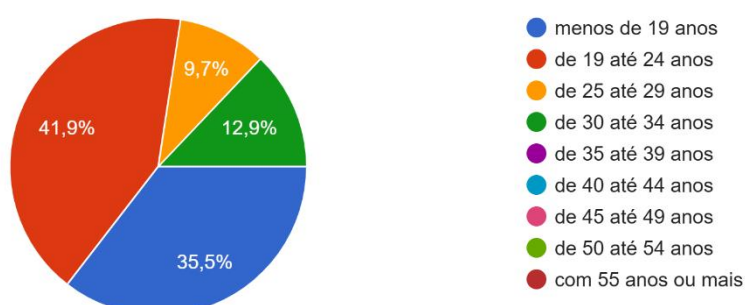


Fonte: Desenvolvido pelo autor (2026).

Figura 33 - Gráfico de respostas à pergunta sobre idade

Em qual faixa etária você está?

31 respostas

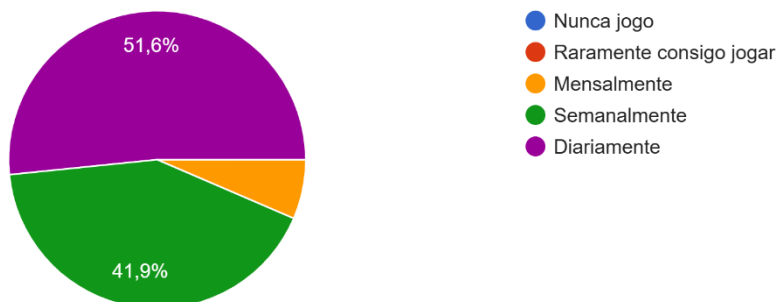


Fonte: Desenvolvido pelo autor (2026).

Figura 34 - Gráfico de respostas à pergunta sobre frequência de jogo

Com que frequência você costuma jogar Jogos Digitais?

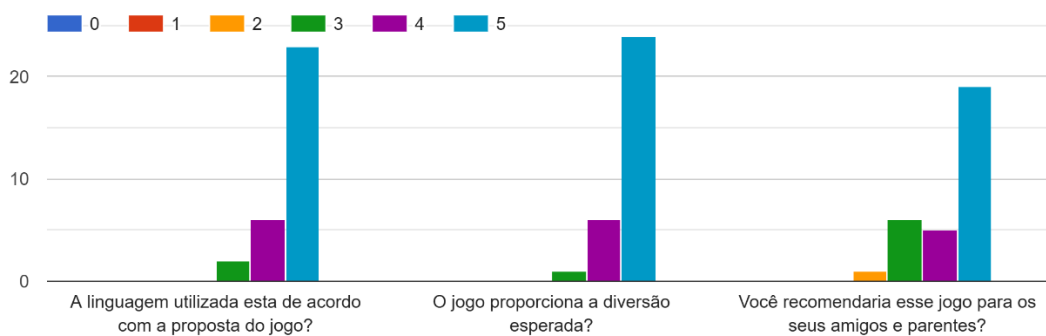
31 respostas



Fonte: Desenvolvido pelo autor (2026).

Figura 35 - Gráfico de respostas de opinião geral sobre o jogo

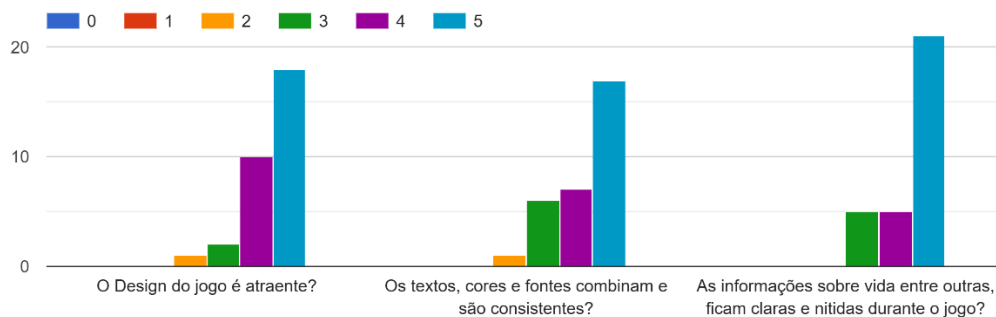
Opinião geral sobre o Jogo



Fonte: Desenvolvido pelo autor (2026).

Figura 36 - Gráfico de respostas sobre o design do jogo

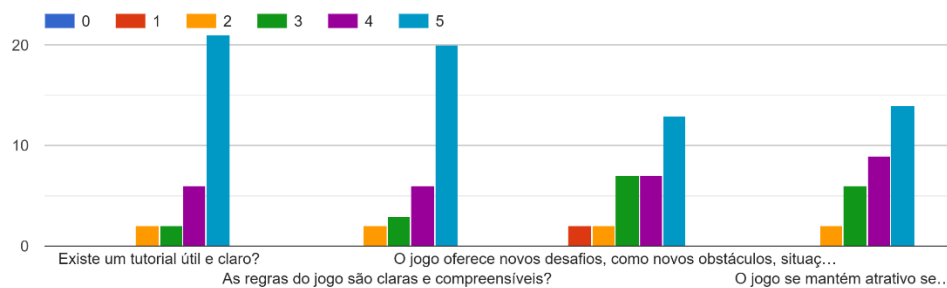
Design do Jogo



Fonte: Desenvolvido pelo autor (2026).

Figura 37 - Gráfico de respostas sobre a jogabilidade

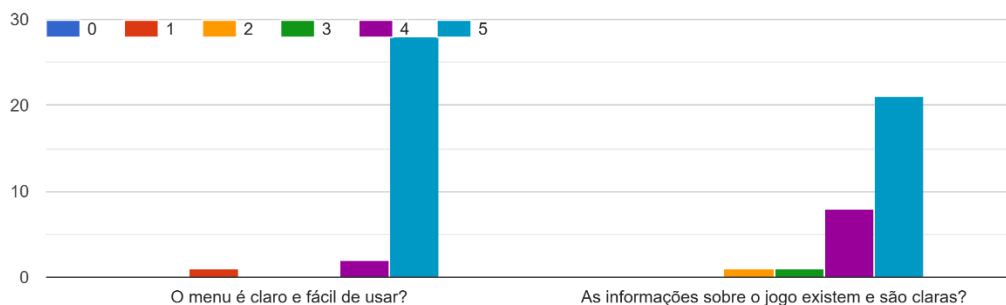
Jogabilidade



Fonte: Desenvolvido pelo autor (2026).

Figura 38 - Gráfico de respostas sobre menu e interação

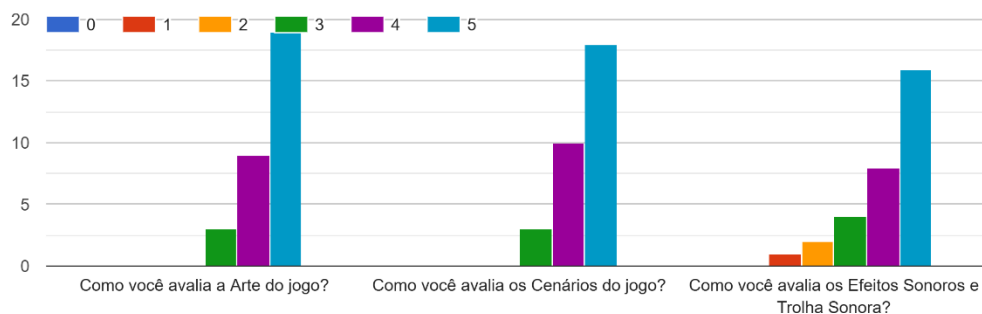
Menu e interação



Fonte: Desenvolvido pelo autor (2026).

Figura 39 - Gráfico de respostas sobre arte e som

Arte e Som



Fonte: Desenvolvido pelo autor (2026).

Figura 40 - *Feedback* dos participantes 1

Talvez seja interessante adicionar colisores nos objetos, mas está excelente!!!!!!

o jogo está bom, apenas alguns bugs e a velocidade de ataque dos inimigos está lenta

o jogo tem uma pixel art e cenários muito bons, mas uma variação a mais de cenários e inimigos como outros tipo de peixes faria o jogo ficar bem mais interessante

variação de inimigos

Fonte: Desenvolvido pelo autor (2026).

Figura 41 - *Feedback* dos participantes 2

Mapas iguais confundem pra saber por onde você passou, talvez um indicativo de que você já terminou a área daquela porta.

Só isso, com certeza jogaria e rejogaria, muito viciante e divertido

A escrita inicial poderia ficar um pouco menor porque ajuda na leitura

Jogo muito interessante e rapido, com uma boa variedade de possibilidade para jogar, mas que poderia se estender muito mais. A mecanica de pesca e os buffs são muito legais, mas ela poderia ser mais intuitiva, talvez aumentando a barra de pesca e dando itens para poder usar entre as fases que possam recuperar a vida ou utilizar mais possibilidades. Mas ainda assim, muito legal.

Não tenho muitas críticas, apenas de que o jogo precisa de maior variedade de inimigos.

Um mini-mapa seria muito útil

É bastante complexo no começo, mas é só pegar o jeito, acho que ser um pouco menos difícil no começo.

Fonte: Desenvolvido pelo autor (2026).

10.2 Análise da loja

O projeto foi disponibilizado publicamente na plataforma Itch.io, mas não apresentou interações que permitissem análises aprofundadas.

O foco das análises do jogo foi o teste realizado localmente na FATEC Americana. Sua publicação na plataforma demonstra a última etapa do desenvolvimento de um jogo e permitirá a coleta aprofundada de informações e *feedback* de um público mais abrangente futuramente.

11 CONSIDERAÇÕES FINAIS

Este projeto proporcionou experiências práticas do desenvolvimento de um jogo digital, desde sua concepção teórica e de *design*, até o desenvolvimento de arte e programação de *scripts* e sistemas complexos e interligados, os quais geraram uma experiência de jogo agradável ao público, conforme resultado do teste beta. Por ter sido desenvolvido de forma individual, o contato direto com todas as partes do projeto se mostrou crucial para o desenvolvimento pessoal das habilidades do autor, principalmente na criação de arte digital e programação de sistemas.

Embora algumas partes importantes não tenham sido implementadas, como maior variedade de fases e inimigos e minimapa, além de pequenos *bugs* e balanceamentos, o projeto sucedeu ao implementar as mecânicas propostas mesclando o gênero *Roguelike* e mini jogos de pesca, resultando em um jogo divertido e engajante que requer o foco do jogador.

Para estudos futuros, é interessante expandir a base criada por este projeto, como a criação de inimigos e fases, além de novos mini jogos de pesca e refinamento da geração procedural de mapas.

Alguns pontos que deram certo durante o desenvolvimento:

- a. **Escolha da engine:** A Unity se provou o ambiente ideal para a elaboração deste projeto por proporcionar ferramentas e sistemas básicos que aceleraram a produção permitindo o foco no desenvolvimento das mecânicas específicas do jogo.
- b. **Escolha da ferramenta Aseprite:** Embora não seja gratuita, o investimento na ferramenta foi uma escolha excelente para o desenvolvimento da arte, mostrando-se uma das ferramentas mais robustas para a criação de *pixel art*, tanto estáticas quanto animadas.
- c. **Tipo de arte:** O estilo artístico *pixel art* rapidamente se mostrou a escolha correta para o projeto. Considerando a falta de familiaridade do autor com criação de arte, este estilo se provou simples o suficiente para a criação de *assets* satisfatórios ao mesmo tempo em que foi desafiador na medida correta para auxiliar no desenvolvimento pessoal quanto a criação de arte.
- d. **Gênero e mecânicas:** A escolha de mesclar mecânicas de *Roguelike* com mini jogos de pesca resultou em um jogo divertido e engajante.

e. **Teste Beta:** A realização do teste beta validou as escolhas de desenvolvimento do jogo, obtendo bons resultados entre os participantes, e auxiliou na conclusão deste projeto por fornecer dados que mostram o sucesso na criação do jogo proposto.

Alguns pontos que deram errado durante o desenvolvimento:

a. **Individualismo:** Como o projeto foi desenvolvido de forma individual, não houve possibilidade de evoluir com base nas interações com outras pessoas e a falta de especialização em pontos do projeto (por exemplo, artista dedicado) certamente afetou a qualidade final do jogo produzido.

DOWNLOAD DO JOGO

O jogo desenvolvido, *Deep Cave Catch*, está disponível publicamente na plataforma Itch.io.



<https://axotlofdoom.itch.io/deep-cave-catch>

Disponível em: <https://axotlofdoom.itch.io/deep-cave-catch>. Acesso em: 30 de maio de 2026.

REFERÊNCIAS

ADAMS, Ernest. **Fundamentals of game design**. 3.ed. San Francisco: New Riders, 2014.

CARTLIDGE, James. **Genre, Prototype Theory and the Berlin Interpretation of Roguelikes**. Game Studies, 2024. Disponível em: <https://gamestudies.org/2403/articles/cartlidge>. Acesso em: 30 de maio de 2026.

CONCERNEDAPE. Stardew Valley. [S.l.]: **ConcernedApe**, 2016. Jogo eletrônico. Disponível em: https://store.steampowered.com/app/413150/Stardew_Valley/. Acesso em: 31 mai 2026.

HUIZINGA, Johan. **Homo Ludens**: a study of the play-element in culture. New York: Roy Publishers, 1950.

JUUL, Jesper. **Half-Real**: Video Games between Real Rules and Fictional Worlds. Cambridge: MIT Press, 2005.

McMILLEN, Edmund; HIMSL, Florian. **The Binding of Isaac**. [S.l.]: Edmund McMillen, 2011. Jogo eletrônico. Disponível em: https://store.steampowered.com/app/113200/The_Binding_of_Isaac/. Acesso em: 31 mai 2026.

MILLINGTON, Ian. **AI for Games**. 3rd ed. Boca Raton: CRC Press, 2019.

SALEN, Katie; ZIMMERMAN, Eric. **Rules of Play**: Game Design Fundamentals. Cambridge: MIT Press, 2004.

SUPERGIANT GAMES. Hades. [S.l.]: **Supergiant Games**, 2020. Jogo eletrônico. Disponível em: <https://store.steampowered.com/app/1145360/Hades/>. Acesso em: 31 mai 2026.