

---

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

Cauê Pazotto Bellan  
Kauan Gabriel Chiaranda da Rosa  
Pedro Carriel de Barros  
Ricardo Moraes Ferreira

**O JULGAMENTO:  
Relatório de desenvolvimento de um jogo Roguelike**

---

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

Cauê Pazotto Bellan

Kauan Gabriel Chiaranda da Rosa

Pedro Carriel de Barros

Ricardo Moraes Ferreira

**O JULGAMENTO:**

**Relatório de desenvolvimento de um jogo Roguelike**

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**

**2025**

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

BELLAN, Cauê Pazotto

O julgamento: relatório de desenvolvimento de um jogo roguelike. / Cauê Pazotto Bellan, Kauan Gabriel Chiaranda da Rosa, Pedro Carriel de Barros, Ricardo Moraes Ferreira – Americana, 2025.

67f.

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

Orientador: Prof. Esp. José William Pinto Gomes

1. Jogos digitais 2. Jogos eletrônicos 3. Narrativa em jogos eletrônicos. I. BELLAN, Cauê Pazotto, II. ROSA, Kauan Gabriel Chiaranda da, III. BARROS, Pedro Carriel de, IV. FERREIRA, Ricardo Moraes V. GOMES, José William Pinto VI. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana Ministro Ralph Biasi

CDU: 681.6

681.6

681.6

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


Cauê Pazotto Bellan  
Kauan Gabriel Chiaranda da Rosa  
Pedro Carriel de Barros  
Ricardo Moraes Ferreira

**O JULGAMENTO:**  
**Relatório de desenvolvimento de um jogo Roguelike**

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

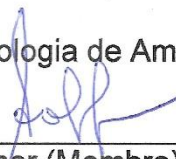
Americana, 05 de dezembro de 2025.

**Banca Examinadora:**




---

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



---

Renato Kraide Soffner (Membro)  
Doutor  
Faculdade de Tecnologia de Americana Ministro Ralph Biasi



---

José Mario Frasson Scaffi (Membro)  
Mestre  
Faculdade de Tecnologia de Americana Ministro Ralph Biasi

## RESUMO

Este trabalho apresenta os processos de criação e de desenvolvimento do projeto O Julgamento, um jogo 2D do gênero *roguelike* situado no Velho Oeste, visando o estudo e prática das teorias de jogos digitais. O jogo proporciona aos jogadores a chance de acompanhar a história do xerife Kane Ropes, que puniu um homem inocente e deverá enfrentar as consequências. O principal objetivo do jogo é integrar a narrativa às mecânicas de repetição e aleatoriedade que compõem o gênero *roguelike*, explorando o conceito de morte permanente, a variação nas fases e a evolução do jogador por meio de escolhas estratégicas a cada tentativa. A estrutura narrativa acompanha propositalmente o ciclo do gênero, fazendo com que a repetição após cada falha componha o desenvolvimento do personagem e reforce elementos temáticos do enredo. Cada nova jogatina permite ao jogador reinterpretar eventos, enfrentar desafios diferentes e avançar gradualmente na compreensão da trama, criando paralelos entre tentativa, erro e aprendizado, fundamentais para o *roguelike*. Sistemas como as cartas de melhoria, a geração de desafios nas fases e a progressão cíclica contribuem para oferecer experiências únicas a cada partida, fortalecendo a integração entre narrativa e as mecânicas. Para o desenvolvimento foram utilizadas as ferramentas *Unity*, *Visual Studio Code* e *LibreSprite*, consolidando o conhecimento adquirido ao longo do curso como parte fundamental da construção do projeto.

**Palavras-chave:** *Unity*; Jogos Digitais; *Roguelike*; Desenvolvimento 2D.

## **ABSTRACT**

*This project presents the creation and development process of O Julgamento, a 2D roguelike game set in the Old West, aiming to study and apply concepts related to digital game design. The game offers players the opportunity to follow the story of Sheriff Kane Ropes, who wrongfully punished an innocent man and must now face the consequences. The main objective of the project is to integrate the narrative with the mechanics of repetition and randomness that define the roguelike genre, exploring elements such as permadeath, variation between stages, and player progression through strategic choices in each attempt. The narrative structure was intentionally designed to accompany the cyclical nature of the genre, allowing each failure to contribute to the protagonist's development and reinforce thematic aspects of the story. Every new playthrough enables players to reinterpret events, encounter different challenges, and gradually deepen their understanding of the plot, drawing parallels with trial, error, and learning, core principles of roguelike design. Systems such as upgrade cards, dynamic challenge generation, and cyclical progression were implemented to provide unique experiences in every session, strengthening the integration between narrative and gameplay mechanics. The development was carried out using Unity, Visual Studio Code, and LibreSprite, consolidating the knowledge acquired throughout the course as an essential part of the creation of the project.*

**Keywords:** Unity; Digital Games; Roguelike; 2D Development.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1:</b> Input System – Categoria Player .....	19
<b>Figura 2:</b> Input System - Categoria UI.....	19
<b>Figura 3:</b> Componente Cinemachine.....	20
<b>Figura 4:</b> Tilemap .....	21
<b>Figura 5:</b> Tilemap Layers.....	22
<b>Figura 6:</b> A* Nodes .....	23
<b>Figura 7:</b> Exemplo ScriptableObject.....	24
<b>Figura 8:</b> Classe StatUpgrade.....	25
<b>Figura 9:</b> Exemplo ScriptableObject 2 .....	26
<b>Figura 10:</b> Classe ActiveUpgrade .....	27
<b>Figura 11:</b> Produção de Sprites .....	29
<b>Figura 12:</b> Kane Ropes .....	31
<b>Figura 13:</b> Kane Ropes - Animações .....	32
<b>Figura 14:</b> O Enforcado.....	33
<b>Figura 15:</b> Inimigo 1 - Zumbi .....	34
<b>Figura 16:</b> Inimigo 2 - Olho Demoníaco .....	35
<b>Figura 17:</b> Inimigo 3 – Mago .....	36
<b>Figura 18:</b> Interface - Menu .....	37
<b>Figura 19:</b> Interface – Ajustes .....	38
<b>Figura 20:</b> Interface – Créditos 1.....	38
<b>Figura 21:</b> Interface – Créditos 2.....	38
<b>Figura 22:</b> Interface – Créditos 3.....	39
<b>Figura 23:</b> Interface – Introdução .....	39
<b>Figura 24:</b> Interface – HUD .....	39
<b>Figura 25:</b> Interface – Escolha de Melhoria .....	40
<b>Figura 26:</b> Interface – Pausado.....	40
<b>Figura 27:</b> Interface – Inventário .....	40
<b>Figura 28:</b> Interface – Tela de Morte.....	41
<b>Figura 29:</b> Interface – Conclusão .....	41
<b>Figura 30:</b> Interface – Tela de Vitória.....	41
<b>Figura 31:</b> Interface – Tutorial 1 .....	42

<b>Figura 32:</b> Interface – Tutorial 2 .....	42
<b>Figura 33:</b> Interface – Tutorial 3 .....	42
<b>Figura 34:</b> Fluxograma – Progressão .....	45
<b>Figura 35:</b> Cenário – Fase 1 .....	45
<b>Figura 36:</b> Cenário – Fase 2 .....	46
<b>Figura 37:</b> Cenário – Fase 3 .....	46
<b>Figura 38:</b> Cenário – Fase 4 .....	46
<b>Figura 39:</b> Cenário – Fase 5 .....	47
<b>Figura 40:</b> Cenário – Fase 6 .....	47
<b>Figura 41:</b> Caso de Uso – Sistema de Melhorias .....	49
<b>Figura 42:</b> Melhoria – Balas de Prata.....	50
<b>Figura 43:</b> Melhoria – Distintivo de Xerife .....	50
<b>Figura 44:</b> Melhoria – Pólvora Especial .....	51
<b>Figura 45:</b> Melhoria – Cinto de Balas.....	51
<b>Figura 46:</b> Melhoria – Gatilho Melhorado.....	52
<b>Figura 47:</b> Melhoria – Tambor Melhorado.....	52
<b>Figura 48:</b> Melhoria – Chapéu de Vaqueiro .....	53
<b>Figura 49:</b> Melhoria – Poncho .....	53
<b>Figura 50:</b> Melhoria – Cantil .....	54
<b>Figura 51:</b> Melhoria – Charuto .....	54
<b>Figura 52:</b> Melhoria – Bandana.....	55
<b>Figura 53:</b> Melhoria – Botas de Vaqueiro .....	55
<b>Figura 54:</b> Pesquisa Teste Beta – Pergunta 1 .....	56
<b>Figura 55:</b> Pesquisa Teste Beta – Pergunta 2.....	57
<b>Figura 56:</b> Pesquisa Teste Beta – Pergunta 3 .....	57
<b>Figura 57:</b> Pesquisa Teste Beta – Seção Opinião Geral .....	58
<b>Figura 58:</b> Pesquisa Teste Beta – Seção Design do Jogo .....	58
<b>Figura 59:</b> Pesquisa Teste Beta – Seção Jogabilidade .....	59
<b>Figura 60:</b> Pesquisa Teste Beta – Seção Menu e Interação .....	60
<b>Figura 61:</b> Pesquisa Teste Beta – Seção Arte e Som .....	60



## LISTA DE ABREVIATURAS E SIGLAS

**2D:** Duas Dimensões

**3D:** Três Dimensões

**IDE:** Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

**GIF:** Graphics Interchange Format (Formato de Intercâmbio de Gráficos)

**PNG:** Portable Network Graphics (Gráficos de Rede Portáteis)

**RPG:** Role-Playing Game (Jogo de Interpretação de Papéis)

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>11</b>
<b>2. REFERENCIAL TEÓRICO .....</b>	<b>12</b>
<b>2.1 Jogo Digital .....</b>	<b>12</b>
<b>2.2 Roguelike.....</b>	<b>12</b>
<b>2.3 Rogue.....</b>	<b>13</b>
<b>2.4 Ferramentas Utilizadas .....</b>	<b>13</b>
2.4.1 <i>Unity</i> .....	13
2.4.2 <i>LibreSprite</i> .....	14
2.4.3 <i>Visual Studio Code</i> .....	14
2.4.4 <i>Audacity</i> .....	15
<b>2.5 Jogos Similares .....</b>	<b>15</b>
2.5.1 <i>The Binding of Isaac</i> .....	15
2.5.2 <i>Enter the Gungeon</i> .....	16
2.5.3 <i>Hades</i> .....	17
<b>3. DESENVOLVIMENTO DO PROJETO .....</b>	<b>18</b>
<b>3.1 METODOLOGIA.....</b>	<b>18</b>
3.1.1 <i>Input System</i> .....	18
3.1.2 <i>Cinemachine</i> .....	20
3.1.3 <i>Tilemap</i> .....	21
3.1.4 <i>Algoritmo A* (A-star)</i> .....	22
3.1.5 <i>Scriptable Objects</i> .....	23
<b>3.2 Game Design Document.....</b>	<b>27</b>
3.2.1 <i>Escolha do Gênero</i> .....	27
3.2.2 <i>Narrativa do Jogo</i> .....	28
3.2.3 <i>Arte</i> .....	28
3.2.4 <i>Fonte</i> .....	30
3.2.5 <i>Personagens do Jogo</i> .....	30
3.2.6 <i>Inimigos</i> .....	34
3.2.7 <i>Sons e Música</i> .....	36
3.2.8 <i>Interfaces</i> .....	37
<b>3.3 Mecânicas de Jogo.....</b>	<b>43</b>

<b>3.4 Progressão do Jogo .....</b>	<b>44</b>
3.4.1 Cenários.....	45
3.4.1 Obtenção de Melhorias.....	48
3.4.2 Melhorias.....	50
<b>4. RESULTADOS .....</b>	<b>56</b>
4.1 Teste Beta Realizado.....	56
<b>5. DOWNLOAD DO JOGO.....</b>	<b>61</b>
<b>CONSIDERAÇÕES FINAIS.....</b>	<b>62</b>
<b>REFERÊNCIAS.....</b>	<b>63</b>

## 1. INTRODUÇÃO

O presente projeto, O Julgamento, é um *roguelike* de ação no estilo *top down* (vista de cima), com visual em *pixel art* (arte baseada em *pixels*). Os elementos do jogo têm como foco a movimentação rápida do personagem e as ondas constantes de inimigos, com o objetivo de prender a atenção do jogador por meio de um combate frenético e desafiador.

O gênero *roguelike* foi escolhido por se adequar à proposta da narrativa, em que o personagem principal retorna ao ponto inicial sempre que é derrotado. A cada nova fase superada, o jogador é recompensado com uma carta especial, que concede melhorias ao personagem, uma mecânica marcante desse estilo de jogo, que valoriza o progresso a partir da tentativa, falha e repetição.

A escolha da câmera em visão *top down* permite uma melhor visualização do cenário e controle das ondas de inimigos, aumentando o nível de desafio. A arma principal do personagem é uma pistola, o que possibilita ao jogador mirar livremente com o *mouse*, tornando o combate mais dinâmico.

As principais referências para o desenvolvimento foram os jogos do mesmo gênero, como *Hades*, *The Binding of Isaac* e *Enter the Gungeon*, títulos que ajudaram a moldar os padrões do gênero ao longo dos anos, especialmente por seus sistemas de morte permanente, progressão e recompensas.

Em O Julgamento, o jogador percorre cidades destruídas do Velho Oeste, acompanhando o xerife Kane Ropes em sua missão de deter o espírito vingativo conhecido como “O Enforcado”.

## 2. REFERENCIAL TEÓRICO

### 2.1 Jogo Digital

Um jogo, seja digital ou analógico, pode ser definido de diversas formas diferentes. Segundo Huizinga (1949), o jogo antecede a cultura, estando presente tanto na vida dos seres humanos quanto nos animais. Ele não se limita apenas a uma função biológica ou instintiva, mas envolve aspectos como diversão, imaginação e criação de sentido. Dessa forma, o jogo transcende necessidades utilitárias, sendo uma manifestação espontânea de liberdade e expressão simbólica.

Por outro lado, Segundo Rogers (2012, p. 26–27), um jogo pode ser definido como uma atividade que requer no mínimo um jogador, possui regras e uma condição de vitória, enquanto um videogame é simplesmente um jogo exibido em uma tela.

### 2.2 Roguelike

*Roguelike* é um gênero de jogos digitais, mais especificamente um sub gênero de RPG, caracterizado por elementos como geração procedural de fases, *permadeath* (morte permanente do personagem), jogabilidade baseada em turnos e alto fator de rejogabilidade.

Os jogos que pertencem a este gênero normalmente apresentam mundos exploráveis, com grande foco em mecânicas e experiências únicas a cada nova partida, devido à aleatoriedade dos desafios e fases. Segundo o Roguebasin, uma das principais comunidades voltadas ao estudo e desenvolvimento de *roguelikes*, essas características estruturam a definição clássica do gênero, herdada do jogo *Rogue* (1980), que lhe deu origem (Roguebasin, 2024).

Para muitos, o apelo dos *roguelikes* está na combinação entre desafio, imprevisibilidade e aprendizado por tentativa e erro, proporcionando ao jogador uma progressão menos guiada e mais orgânica. A dificuldade elevada, aliada à aleatoriedade, contribui para tornar cada sessão única, incentivando múltiplas jogatinas como parte do próprio design do jogo (Mana Pool, 2024).

## 2.3 *Rogue*

O termo *roguelike* vem do inglês *like*, que significa “parecido” ou “semelhante”, enquanto *Rogue* é o nome do jogo que originou o gênero, sendo assim jogos “parecidos com *Rogue*”.

*Rogue* (1980) é um jogo sobre explorar masmorras e labirintos, sub gênero mais conhecido como *dungeon crawling*. Sua interface é baseada em blocos de texto em ASCII ou outros conjuntos de caracteres. O objetivo do jogador é controlar o personagem e explorar diversos níveis de uma masmorra em busca do amuleto de Yendor, localizado no andar mais baixo. Conforme avança, o jogador deve derrotar hordas de monstros e coletar recursos para ajudar em sua aventura.

A mecânica de morte permanente se popularizou com *Rogue*: ao perder toda sua vida pelo combate ou outros meios, o personagem morria e o jogador deveria começar com um novo personagem do zero. Além disso, cada vez que reinicia, o mapa e os encontros seriam totalmente novos, pois era gerado proceduralmente para cada jogatina. Apesar de não ter sido o primeiro a utilizar esses elementos, *Rogue* foi o responsável por popularizá-los, dando origem ao nome do gênero *roguelike*.

Durante os anos 1980, *Rogue* se tornou muito famoso entre os programadores, por vir integrado com o sistema operacional Unix 4.2BSD, inspirando muitos jogos na época por apresentar uma estrutura de jogo baseado em texto contendo elementos como a morte permanente e a geração procedural, elementos que definem o gênero e que permanecem populares até os dias atuais.

## 2.4 Ferramentas Utilizadas

### 2.4.1 *Unity*

A *Unity* apresenta suporte para diversas plataformas, como computadores, consoles, dispositivos móveis e até mesmo *websites*, além de oferecer funções para criação de jogos 3D, 2D, Realidade Virtual e Realidade Aumentada. Foi lançada em 2005, contando com constantes atualizações para todas as suas versões, sejam elas mais antigas ou mais recentes, além de oferecer grande suporte aos desenvolvedores. Também conta com uma biblioteca de *assets* pagos e gratuitos, feitos por outros desenvolvedores, permitindo encontrar grande parte de seus

projetos, como sistemas, modelos e artes acessíveis aos outros usuários. Atualmente, a *Unity* conta com uma grande comunidade, que oferece ajuda e assistência em diversas áreas da *game engine* ou no código em C#, sendo possível encontrar suporte no fórum da própria *Unity*.

Toda a documentação do *software* pode ser encontrada em seu *site*, sendo possível obter as definições, exemplos das classes e funcionalidades disponibilizadas pela *Unity* e sua vasta biblioteca. Graças à sua comunidade, também há uma grande variedade de vídeos e tutoriais de desenvolvimento de jogos de menor escopo, auxiliando durante o desenvolvimento do projeto.

#### 2.4.2 *LibreSprite*

O *LibreSprite* é um *software* de uso gratuito e de código aberto, oferecendo diversas ferramentas para criação de *sprites* no estilo *pixel art*, artes focadas nos *pixels* da imagem, estilo comumente utilizado em jogos antigos (8-bit e 16-bit). Permite acesso ao gerenciamento de camadas, paletas de cores e uma *interface* leve e flexível, bem como exportação para diferentes tipos de arquivo, como PNG e GIF.

Este programa se originou de uma ramificação do *software* pago *Aseprite* em 2016, época em que ainda possuía uma licença gratuita. Por se tratar de uma ramificação do *Aseprite*, o *LibreSprite* conta com uma grande comunidade compartilhada entre eles, oferecendo a assistência de diversos usuários e artistas para referências ou dúvidas quanto ao *software* ou ao desenvolvimento dos *sprites*.

#### 2.4.3 *Visual Studio Code*

*Visual Studio Code* é um editor de código-fonte gratuito e de código aberto. Foi lançado em 2015 e desenvolvido pela *Microsoft*, tornando-se reconhecido pelo suporte a diversas linguagens de forma leve e extensível (Microsoft, 2015). Oferece várias formas de personalização para a *interface*, como temas, atalhos e *layout*, além de um *debugger* embutido. Também possui integração com Git e outras ferramentas de versionamento, facilitando o gerenciamento durante o desenvolvimento do jogo. O *VSCode* também oferece uma grande variedade de extensões, que permitem suporte a diferentes linguagens, ferramentas adicionais e mais personalizações para os usuários.

#### 2.4.4 Audacity

Para a edição dos efeitos sonoros do jogo, foi utilizado o software *Audacity*. Este programa gratuito e de código aberto permite a gravação e edição de áudio. Durante o projeto foi utilizado o recurso de recorte para garantir que os sons se adequassem aos contextos e pudessem ser utilizados precisamente.

O *Audacity* foi essencial para garantir a qualidade sonora do projeto, contribuindo para a ambientação do jogo e a imersão do jogador nos cenários propostos, especialmente durante os combates e interações com os elementos do jogo.

### 2.5 Jogos Similares

A seguir é apresentado alguns dos jogos que foram utilizados como estudo para o desenvolvimento de O Julgamento, sendo jogos escolhidos com base em suas mecânicas originais e marcantes, que fizeram esses títulos se tornarem grandes sucessos em suas épocas ou até mesmo hoje, que são considerados grandes pilares para o gênero de *roguelike*.

#### 2.5.1 *The Binding of Isaac*

*The Binding of Isaac* (2011) é um jogo do gênero *roguelike*, desenvolvido por Edmund McMillen e Florian Himsl, sendo resultado de uma *Game Jam* (maratona de criação de jogos) de uma semana. O jogo tornou-se muito conhecido por reacender o interesse dos jogadores pelo gênero, fazendo o título ficar conhecido como um dos pilares desse tema.

Combinando elementos de *shooter* multidirecional e visão *top down*, é tematizado com elementos bizarros e grotescos, tendo uma narrativa inspirada pela história bíblica do sacrifício de Isaac, além de conter elementos da perspectiva de Edmund sobre a religião (McMillen, 2011), o que tornou o jogo um lançamento arriscado e sem expectativas de vendas. Contrariando as previsões, *The Binding of Isaac* alcançou 3 milhões de cópias vendidas ao longo de três anos após o seu lançamento.



O jogador deve controlar Isaac ou outro dos seis personagens desbloqueáveis para passar pelos diversos andares do porão, que são gerados aleatoriamente, enquanto derrota os inimigos e coleta melhorias a fim de derrotar os chefes e a mãe de Isaac no final. Além das mecânicas de morte permanente e geração aleatória, o jogo conta com diversas melhorias e itens de poder, que ao serem combinados, adicionam mais maneiras de jogar e passar pelos andares.

### 2.5.2 *Enter the Gungeon*

Desenvolvido pela *Dodge Roll* e publicado pela *Devolver Digital*, *Enter the Gungeon* (2016) é um jogo *roguelike*, com elementos de exploração e tiroteio frenético com diferentes tipos de armas, baseado na visão *top down* (Dodge Roll, 2016). Coleciona críticas positivas pela mídia por ter um combate divertido e centenas de armas exóticas e melhorias para o jogador escolher.

A narrativa do jogo segue um bando de personagens que buscam entrar na misteriosa “*Gungeon*” e explorá-la com o grande objetivo de encontrar a lendária arma capaz de matar o passado. Baseado nisso, o jogador deve escolher um desses personagens e entrar na masmorra para avançar pelos andares e mapas, gerados proceduralmente, enquanto coleta novas armas e melhorias, além de negociar com os vários lojistas durante a aventura.

*Enter the Gungeon* tem um visual com estilo mais retrô e bem colorido, sendo considerado difícil por ser um *bullet hell*, subgênero que consiste em muitos projéteis pela tela. Para sobreviver, o jogador deve utilizar da mecânica de esquiva e rolamento, o que torna necessária uma grande precisão e bons reflexos.

Entre as tentativas, os jogadores entram na “Brecha”, um local onde é possível conversar com os personagens não jogáveis, além de ter acesso a melhorias e novos itens para encontrar durante a masmorra nas próximas tentativas. Por ter essa progressão entre as tentativas, *Enter the Gungeon* pode ser considerado um *roguelite*, gênero que conta com a mecânica de morte permanente, mas adiciona a progressão secundária.

### 2.5.3 Hades

Considerado um dos maiores sucessos em 2020, Hades foi desenvolvido e publicado pela *Supergiant Games*, lançado a princípio em dezembro de 2018 como acesso antecipado pela *Epic Games*, e posteriormente lançado oficialmente em 2020. Utiliza como base as mecânicas de um *roguelike* com adição de elementos de *Hack and Slash* e RPG de Ação, em conjunto com uma narrativa dinâmica, que usa dos elementos do jogo a seu favor (Supergiant Games, 2020). Para os fãs do gênero, Hades foi uma inovação pela sua história profunda e progressiva, levando em consideração a mecânica de morte permanente como um elemento narrativo.

O jogador controla Zagreu, filho de Hades, em sua tentativa constante de escapar do submundo. A cada tentativa, ele enfrenta uma jornada por diferentes regiões, enquanto enfrenta inimigos e chefes que bloqueiam seu caminho. O mapa das fases é gerado proceduralmente, e cada sala oferece recompensas variadas, como melhorias de atributos, moedas para usar com mercadores ou melhorias, que auxiliam o protagonista a escapar do submundo.

A narrativa progride com o jogador, conforme suas tentativas, sendo elas sucedidas ou não. Durante o jogo, é possível encontrar diversos personagens e avançar a história por meio dos diálogos, garantindo que o jogador entenda melhor a trama, enquanto fica mais forte e consiga progredir mais a cada tentativa.

Ao longo do *gameplay*, é possível obter novas armas, aprimorar seus atributos base e personalizar o estilo de combate para a próxima tentativa. Durante a fuga, o jogador recebe a ajuda dos deuses do Olimpo, que oferecem bênçãos e melhorias para as habilidades que ele possui, possuindo diferentes níveis de raridade e efeitos variados. Com as melhorias corretas, é possível combinar as bênçãos e criar sinergias entre as habilidades e efeitos, dando um maior impacto para o combate.

### 3. DESENVOLVIMENTO DO PROJETO

#### 3.1 METODOLOGIA

O objetivo deste projeto foi desenvolver um jogo chamado O Julgamento, que visa criar uma experiência que integra os principais elementos do gênero *roguelike* a uma narrativa interativa original.

Inicialmente, foi construída uma base narrativa sólida, definindo os personagens principais e os eventos centrais da trama. A partir dessa estrutura, as mecânicas do jogo foram escolhidas e ajustadas para que se adequassem à proposta narrativa. Esse processo contou com leves ajustes na narrativa e jogabilidade para garantir a coerência entre eles.

As ferramentas utilizadas no desenvolvimento incluem:

- *Game Engine Unity*: software responsável pela criação do jogo em 2D.
- *Visual Studio Code*: editor de código que permite a utilização da linguagem de programação C#.
- *LibreSprite*: software para a produção dos elementos visuais em *pixel art*.

Para documentar e organizar o fluxo de trabalho, foi adotado o método *Kanban*, uma abordagem visual que permite separar as tarefas em diferentes etapas de produção, facilitando sua visualização e priorização.

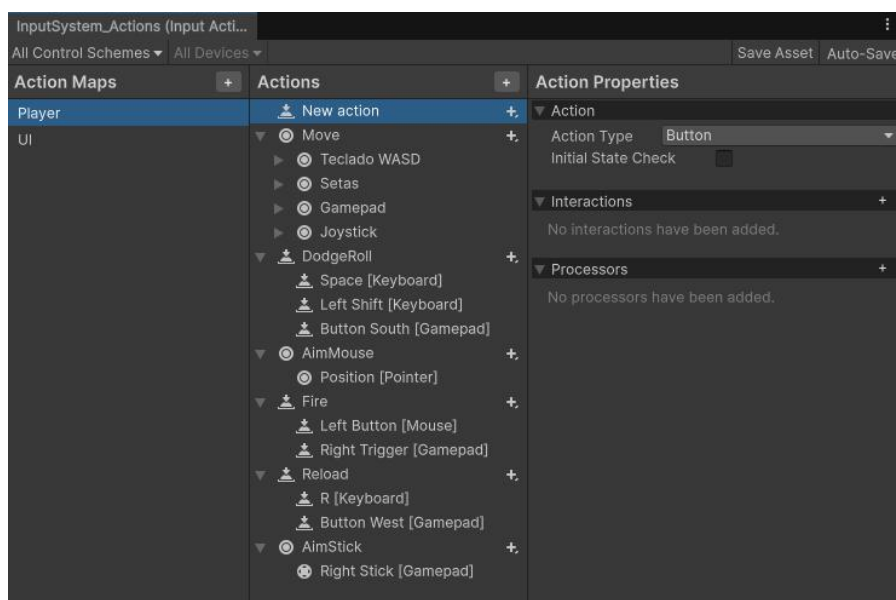
##### 3.1.1 Input System

O Input System é o sistema moderno de gerenciamento de entradas da *Unity*, permitindo maior flexibilidade e suporte multiplataforma. Com ele, é possível mapear diferentes dispositivos de entrada, como teclado, mouse e controle de videogame de forma centralizada e modular. No projeto, ele é utilizado para capturar as ações do jogador (como movimento, ataque e interação), tornando o controle mais responsivo e de fácil personalização.

Como demonstrado nas figuras 1 e 2, os comandos são divididos em duas categorias: jogador e interface. A primeira categoria é dividida entre as diferentes ações que o jogador pode performar durante a *gameplay*, e cada ação é mapeada

para um ou mais botões específicos, que variam de acordo com o dispositivo sendo utilizado. A figura 1 exemplifica algumas das ações desta categoria.

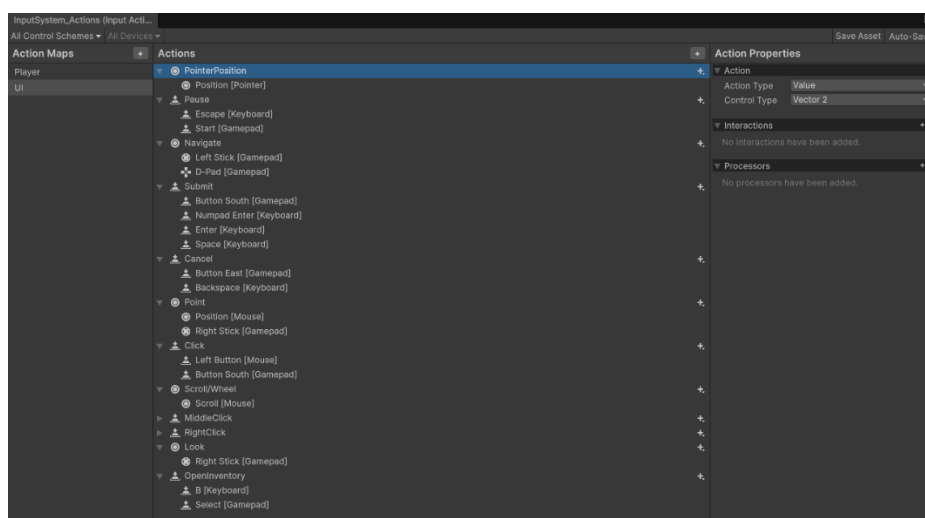
**Figura 1:** *Input System – Categoria Player*



**Fonte:** Desenvolvido pelos autores (2025).

A figura 2 lista as ações e botões referentes a categoria de interface do jogo. Ela é essencial, pois muitas das ações que seriam facilmente executadas com o cursor do mouse, se tornam mais complexas em um controle de videogame, no qual a seleção dos elementos da interface ocorre de forma automática.

**Figura 2:** *Input System - Categoria UI*



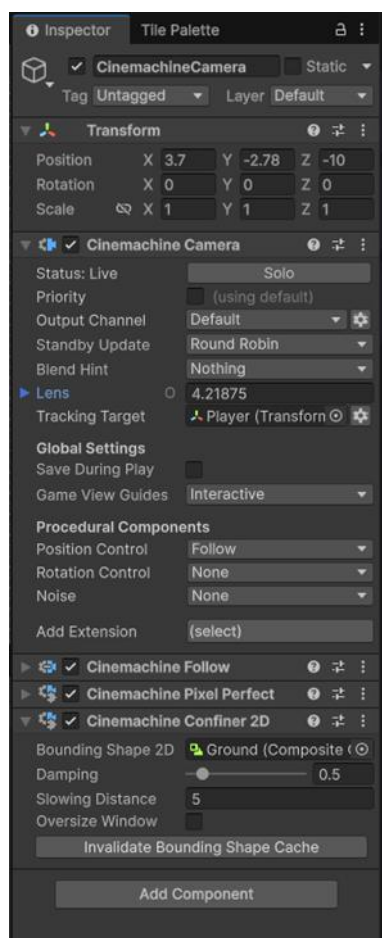
**Fonte:** Desenvolvido pelos autores (2025).

### 3.1.2 Cinemachine

O *Cinemachine* é um sistema de câmera inteligente que permite criar comportamentos automatizados sem a necessidade de *scripts* complexos. Ele ajusta dinamicamente o enquadramento, o zoom e a transição entre câmeras conforme o movimento do jogador ou eventos do jogo. No projeto, o *Cinemachine* é responsável por seguir o personagem principal, facilitando a mudança de cenas e criação de efeitos visuais imersivos.

A figura 3 exibe os componentes do *Cinemachine* no inspetor de elementos da *Unity*. A configuração utilizada no nosso jogo faz com que a câmera siga o jogador a todo momento e esteja confinada num espaço determinando, de modo que seja exibido apenas o cenário desejado.

**Figura 3:** Componente *Cinemachine*

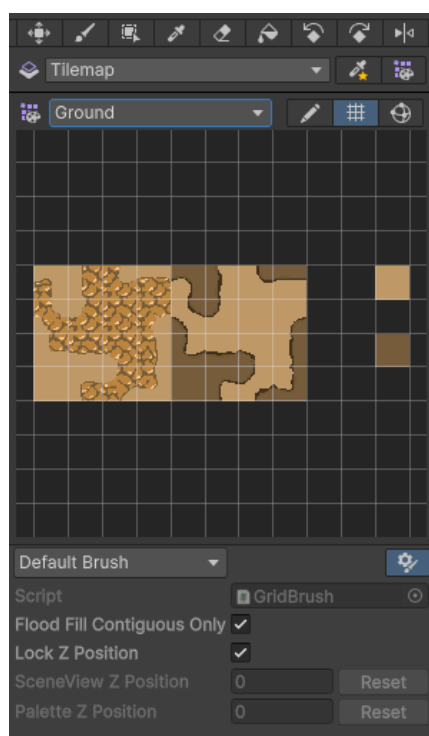


**Fonte:** Desenvolvido pelos autores (2025).

### 3.1.3 Tilemap

O *Tilemap* é uma ferramenta fundamental em jogos 2D, permitindo a criação de mapas complexos a partir de pequenos elementos denominados *tiles*, blocos pixelados que, combinados, compõem a arte visual das áreas. Cada *tile* representa uma unidade mínima no cenário, e a combinação de vários deles possibilita a construção de mapas detalhados e consistentes. Esse sistema facilita tanto o design quanto a manutenção dos mapas, permitindo alterações rápidas sem a necessidade de redesenhar o cenário completo. A figura 4 demonstra o uso da ferramenta.

**Figura 4:** *Tilemap*



**Fonte:** Desenvolvido pelos autores (2025).

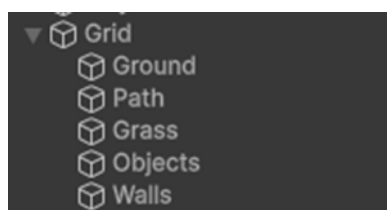
No estudo em foco, o *Tilemap* foi organizado em diferentes *layers* (camadas), nos quais cada uma representa um tipo de elemento e desempenha funções específicas no mapa:

- *Layer Ground*: Corresponde à base do mapa, formando o terreno sobre o qual as entidades do jogo se movimentam.
- *Layer Path*: Indica os caminhos visuais que indicam ao jogador os objetivos no mapa.

- *Layer Grass*: É destinado a elementos visuais sobre o chão, que não apresentam colisão com o jogador ou os inimigos.
- *Layers Obstacles*: Possui *colliders* que interagem com os objetos do jogo e é responsável por distribuir elementos pelo mapa, funcionando como barreiras para o jogador e para projéteis.
- *Layer Walls*: Também possui *colliders* que interagem com os objetos do jogo e define os limites de movimentação, impedindo que o jogador ultrapasse o fim do mapa.

A distinção entre esses *layers* possibilita um desenvolvimento mais organizado e um controle mais preciso do terreno, oferecendo ao jogador desafios variados de movimentação e exploração. É possível ver tal organização na figura abaixo

**Figura 5:** *Tilemap Layers*



**Fonte:** Desenvolvido pelos autores (2025).

A utilização de múltiplos *layers* no *Tilemap* demonstrou-se uma solução eficiente para a construção de mapas complexos e dinâmicos, facilitando a criação visual e a organização do cenário.

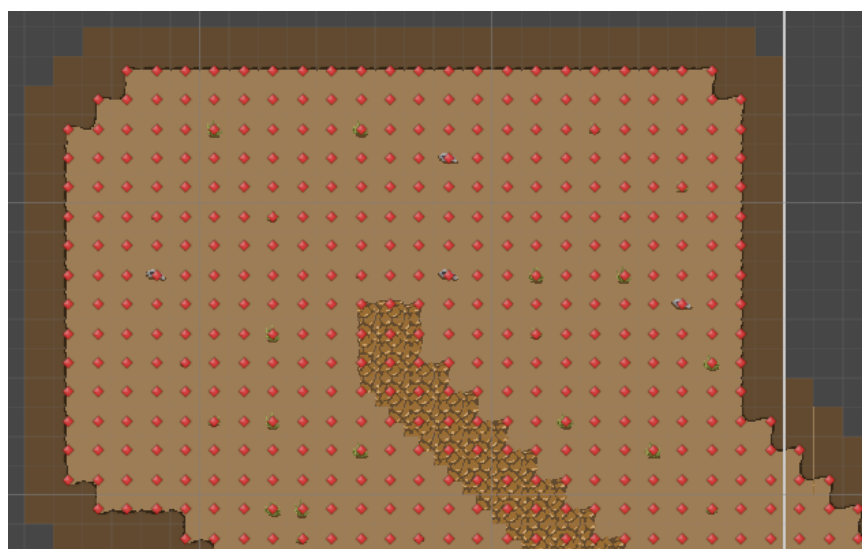
#### 3.1.4 Algoritmo A\* (*A-star*)

O algoritmo A\* (*A-star*) é amplamente utilizado em jogos digitais e sistemas de simulação, com o objetivo de criar inteligências artificiais capazes de determinar o caminho mais eficiente entre dois pontos dentro de um espaço definido. Trata-se de um algoritmo de busca heurística, que combina a avaliação do caminho já percorrido com a estimativa do custo restante até o destino, garantindo resultados precisos e otimizados. No contexto de jogos, o A\* é essencial para permitir que *NPCs* e inimigos se movimentem de maneira eficiente em mapas complexos.

Para a implementação do algoritmo neste estudo, foi adotada uma abordagem baseada em uma grid imaginária, construída automaticamente a partir do *Tilemap* do

jogo. Cada célula da grid contém um *Node*, um *GameObject* vazio que representa uma posição com possibilidade de movimentação. O algoritmo avalia cada *Node*, considerando os custos de movimentação e heurísticas, para determinar o caminho mais curto até o objetivo. Essa estrutura permite que as entidades do jogo naveguem pelo mapa de maneira fluida, independentemente da complexidade do cenário. A figura 6 exibe a disposição dos *Nodes*.

**Figura 6:** A\* *Nodes*



**Fonte:** Desenvolvido pelos autores (2025).

A escolha de uma grid adaptativa trouxe diversas vantagens ao desenvolvimento, permitindo a criação de mapas complexos conforme o *Tilemap*, sem a necessidade de configuração manual dos *Nodes*. Além disso, o sistema suporta atualizações futuras, uma vez que modificações nos tiles são refletidas automaticamente na geração dos *Nodes*, garantindo a consistência do *pathfinding*.

Dessa forma, a utilização do algoritmo A\* demonstrou-se uma solução adequada para a inteligência artificial dos inimigos, permitindo a implementação da Máquina de Estados Finitos com ações como patrulha, perseguição e ataque.

### 3.1.5 *Scriptable Objects*

Os *ScriptableObjects* são uma forma de armazenar dados de maneira independente dos *GameObjects* e das cenas. Eles permitem a criação de “bancos de



dados” reutilizáveis dentro da *Unity*, facilitando o gerenciamento de informações como atributos de personagens, itens, habilidades ou configurações gerais do jogo.

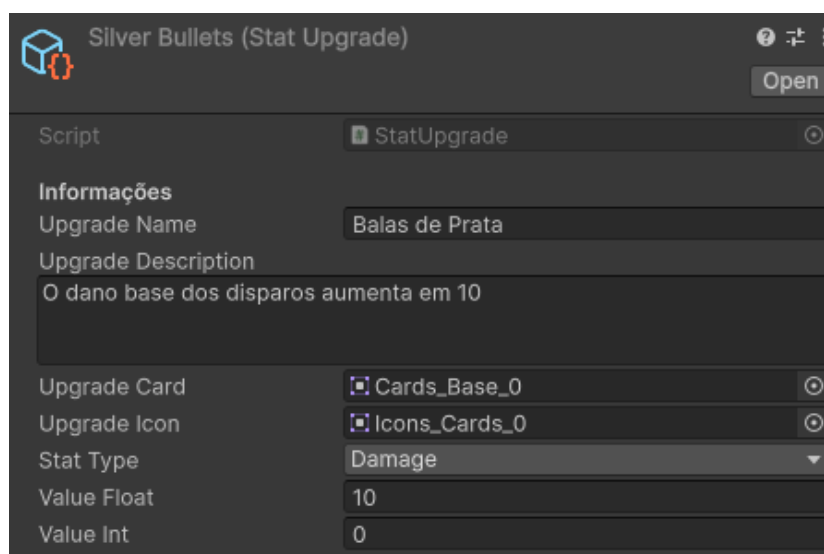
No projeto, os *ScriptableObjects* foram utilizados para criar e definir as melhorias que o jogador obtém durante o jogo, melhorando a organização e evitando a duplicação de código. Estes objetos foram divididos em duas classes principais, que herdam da mesma classe abstrata:

- *StatUpgrade*: Altera diretamente valores que compõem os atributos básicos do jogador.
- *ActiveUpgrade*: Ativa variações no comportamento de outros *scripts*.

A figura 7 demonstra um exemplo de *ScriptableObject* criado durante o desenvolvimento que pertence a classe *StatUpgrade*. Este container de dados irá carregar as seguintes informações:

- *Upgrade Name*: O nome da melhoria que corresponde ao objeto.
- *Upgrade Description*: O efeito concedido ao jogador.
- *Upgrade Card* e *Upgrade Icon*: Os componentes que irão compor a arte do upgrade.
- *Stat Type*: Classificação do efeito a ser aplicado.
- *Value Float* e *Value Int*: Tipo e valor a ser modificado nos atributos do jogador.

**Figura 7:** Exemplo *ScriptableObject*



**Fonte:** Desenvolvido pelos autores (2025).

Na figura 8 é possível ver o *Script* que define esta classe. Cada *ScriptableObject* é classificado de acordo a melhoria correspondente, e então, dependendo desta classificação, são incrementados ou decrementados dos atributos do jogador o valor carregado pelo objeto.

**Figura 8:** Classe *StatUpgrade*

```

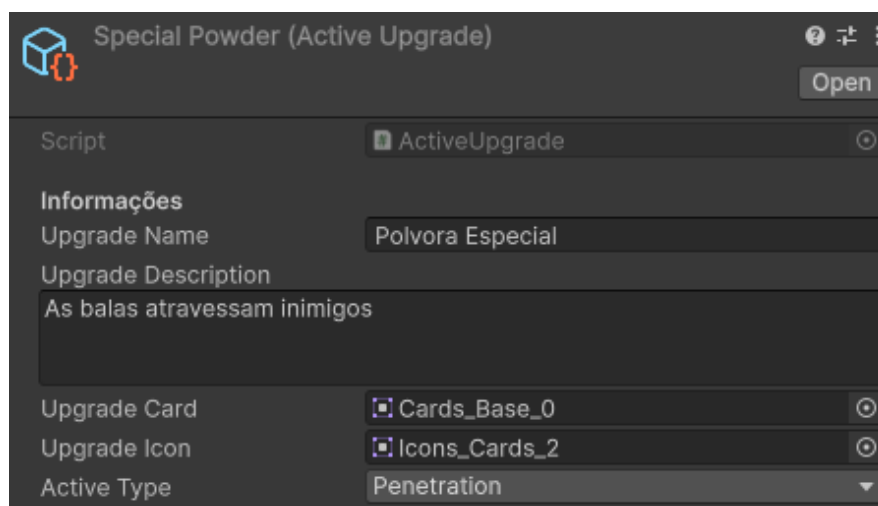
1  using UnityEngine;
2
3  10 references
4  public enum StatType
5  {
6      1 reference | 1 reference | 1 reference
7      Damage, CritChance, MaxAmmo,
8      1 reference | 1 reference | 1 reference
9      FireRate, ReloadTime, RollCooldown,
10     1 reference | 1 reference | 1 reference
11     MoveSpeed, DamageReduction, MaxHealth,
12 }
13
14 [CreateAssetMenu(menuName = "Upgrades/StatUpgrade")]
15 0 references
16 public class StatUpgrade : UpgradeClass
17 {
18     1 reference
19     public StatType statType;
20     9 references
21     public float valueFloat;
22     1 reference
23     public int valueInt;
24     0 references
25     public override void Effect(PlayerStats player)
26     {
27         switch (statType)
28         {
29             case StatType.Damage:
30                 player.baseDamage += valueFloat;
31                 break;
32             case StatType.CritChance:
33                 player.critChance += valueFloat;
34                 break;
35             case StatType.MaxAmmo:
36                 player.ammoMax += valueInt;
37                 break;
38             case StatType.FireRate:
39                 player.fireRate -= valueFloat;
40                 break;
41             case StatType.ReloadTime:
42                 player.reloadTime -= valueFloat;
43                 break;
44             case StatType.RollCooldown:
45                 player.dodgeCooldown -= valueFloat;
46                 break;
47             case StatType.MoveSpeed:
48                 player.moveSpeed += valueFloat;
49                 break;
50             case StatType.MaxHealth:
51                 player.maxHealth += valueFloat;
52                 player.currentHealth += valueFloat;
53                 break;
54             case StatType.DamageReduction:
55                 player.damageReduction += valueFloat;
56                 break;
57         }
58     }
59 }

```

Fonte: Desenvolvido pelos autores (2025).

A figura 9 exibe outro exemplo de *ScriptableObject*, desta vez pertencente a classe *ActiveUpgrade*. Os dados carregados por ele são semelhantes, com a única diferença sendo que este não possui valores a serem alterados.

**Figura 9:** Exemplo *ScriptableObject* 2



**Fonte:** Desenvolvido pelos autores (2025).

Os *ScriptableObjects* da classe *ActiveUpgrade* funcionam de modo que a ativação de seu efeito irá expressar um dado *booleano* (verdadeiro ou falso) como verdadeiro. O dado a ser alterado irá depender de acordo com a classificação do objeto. O funcionamento do *script* desta classe pode ser observado na figura abaixo.

Figura 10: Classe *ActiveUpgrade*

```

1  using UnityEngine;
2
3  4 references
4  public enum ActiveType
5  {
6      1 reference
7      Penetration,
8      1 reference
9      Vampirism,
10     1 reference
11     HealOnKill,
12 }
13
14 [CreateAssetMenu(menuName = "Upgrades/ActiveUpgrade")]
15 0 references
16 public class ActiveUpgrade : UpgradeClass
17 {
18     1 reference
19     public ActiveType activeType;
20     0 references
21     public override void Effect(PlayerStats player)
22     {
23         switch (activeType)
24         {
25             case ActiveType.Penetration:
26                 player.penetration = true;
27                 break;
28             case ActiveType.Vampirism:
29                 player.vampirism = true;
30                 break;
31             case ActiveType.HealOnKill:
32                 player.healOnKill = true;
33                 break;
34         }
35     }
36 }

```

Fonte: Desenvolvido pelos autores (2025).

## 3.2 Game Design Document

### 3.2.1 Escolha do Gênero

O motivo para a escolha do gênero *roguelike* se deu pelo foco em uma jogabilidade que pode ser reaproveitada pelo jogador, proporcionando uma experiência única a cada vez que ele joga. Além disso, estes conceitos podem ser utilizados para ampliar a narrativa a partir do momento que a razão pela qual a história se reinicia e o jogador volta para o início é suportada pelo enredo.

A familiaridade dos desenvolvedores do projeto com jogos semelhantes que compartilham do gênero *roguelike* também foi um fator crucial para sua escolha.

A opção pela câmera *top down* e o estilo visual em *pixel art* foi guiada por sua simplicidade técnica e potencial estilístico, além de facilitar a leitura visual em momentos de combate intenso, comum no gênero.

### 3.2.2 Narrativa do Jogo

A trama do jogo acompanha o renomado xerife Kane Ropes, encarregado de descobrir quem está por trás da morte de um banqueiro influente de sua cidade. Após uma série de investigações, o xerife identificou como principal suspeito um jovem indígena chamado Rayetayah, que era assistente do banqueiro.

Apesar de declarar inocência, Kane Ropes está convicto de que o jovem é culpado, resultando em sua execução por enforcamento. Com o caso dado encerrado, meses se passam até que o xerife começa a receber cartas de cidades vizinhas, relatando ataques de criaturas malignas. Então, ele parte em uma jornada para desvendar o mistério e dar um fim aos ataques.

Conforme o jogador enfrenta os inimigos e coleta as pistas durante as fases, é revelado que por trás dos ataques está uma figura conhecida como “O Enforcado”, que diz estar coletando as almas dos verdadeiros pecadores e deseja se vingar do xerife, o desafiando para um duelo.

Ao vencer o espírito maligno, o xerife descobre que a entidade era, na verdade, o espírito de Rayetayah, buscando justiça por sua morte. A revelação de que o banqueiro havia cometido suicídio e que Rayetayah era inocente leva Kane a refletir sobre seus atos, em sua decisão de se aposentar e na libertação do espírito do jovem.

Entretanto, caso o jogador falhe durante a jornada, fazendo com que Kane Ropes morra antes de concluir seu objetivo, a narrativa sofre uma leve alteração. Antes de o xerife morrer de fato, O Enforcado impede que a sua alma deixe seu corpo, fazendo com que o jogador volte no tempo para o ponto em que começou a sua jornada, fazendo com que a história se repita até o jogador conseguir chegar ao fim do jogo.

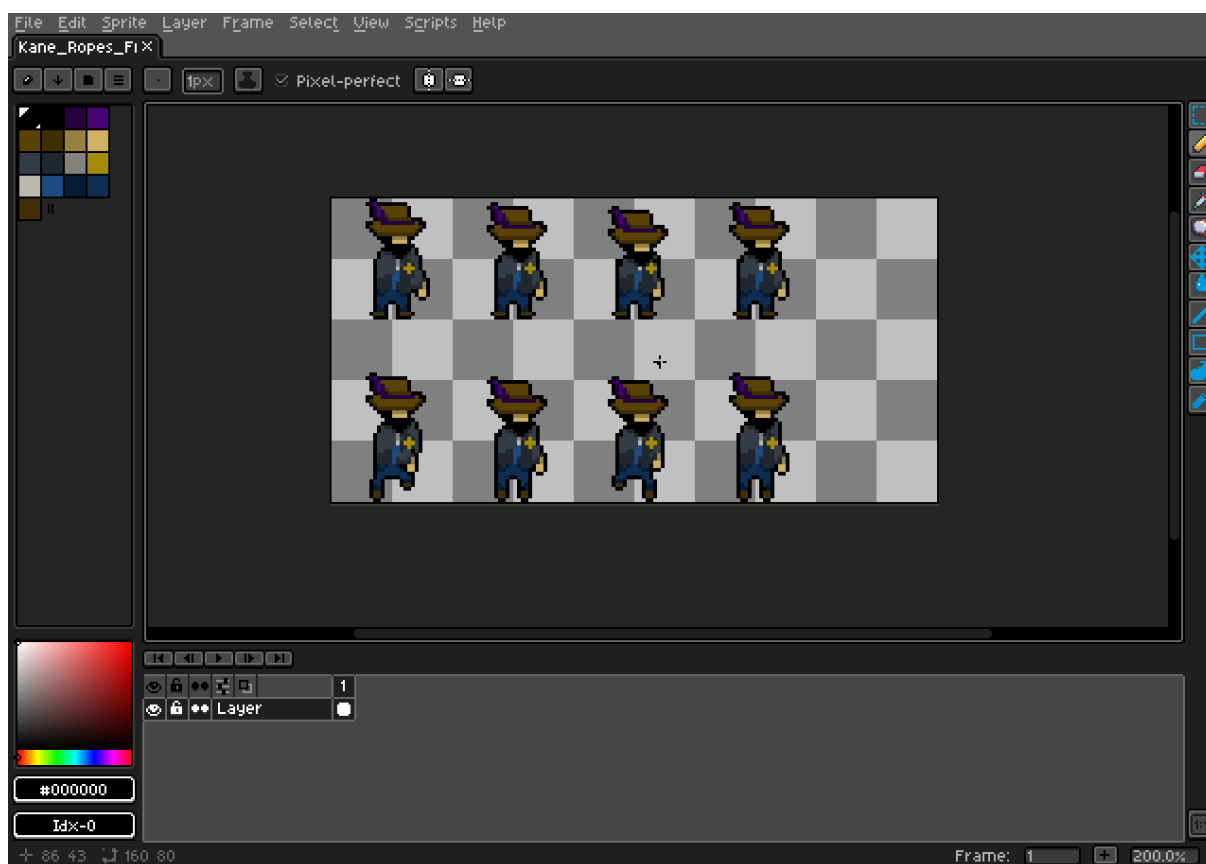
### 3.2.3 Arte

O desenvolvimento dos sprites do projeto foi realizado no *LibreSprite*, uma ferramenta voltada para criação de *pixel art* que oferece recursos simples e eficientes para produção de gráficos em baixa resolução. Para o jogo, adotou-se o formato 32x32 *pixels*, garantindo uma identidade visual retrô e consistente, além de facilitar a integração com a *Unity*. Essa abordagem permitiu criar personagens, inimigos e

objetos com boa legibilidade, mesmo em perspectiva *top-down*, mantendo a clareza visual durante as animações e movimentação no cenário.

A utilização de *pixel art* também favoreceu a expansão do conjunto de *sprites*, permitindo a criação de novas variações de forma rápida e organizada. O *LibreSprite* contribuiu para esse processo ao oferecer paletas personalizadas, camadas separadas e pré-visualização de animações, facilitando ajustes finos durante o desenho. Dessa forma, o conjunto de *sprites* foi desenvolvido de maneira prática e coerente com a proposta estética do projeto, garantindo uma linguagem visual uniforme em todas as fases e elementos do jogo.

**Figura 11:** Produção de *Sprites*



**Fonte:** Desenvolvido pelos autores (2025).

### 3.2.4 Fonte

A fonte *BoldPixels*, do pacote *YUKIPIXELS*, foi integrada ao projeto para reforçar a identidade visual retro do jogo. Trata-se de uma fonte *bitmap* em estilo *pixel art*, ideal para títulos, HUDs e elementos textuais de jogos com estética clássica. O uso dessa fonte contribui para a coesão artística e mantém o visual consistente com o estilo gráfico do projeto.

### 3.2.5 Personagens do Jogo

O Xerife, protagonista da narrativa, é chamado Kane Ropes, o personagem controlado pelo jogador. Renomado e experiente, acaba sendo encarregado de investigar o assassinato de um importante banqueiro local. A partir de suas investigações, Kane acusa e condena à morte um jovem indígena chamado Rayetayah, que sempre alegou ser inocente.

Conforme a narrativa progride, Kane deve lidar com as consequências de suas ações passadas, sendo desafiado a rever sua própria moralidade diante de seu grave erro. O jogador é levado a questionar se o xerife é realmente um herói durante a história, que termina em um confronto contra a entidade responsável pelos ataques.

O nome “Ropes” significa “cordas”, elemento que faz alusão direta ao passado do personagem. As figuras a seguir apresentam uma arte conceitual do personagem e suas animações.

**Figura 12:** Kane Ropes



**Fonte:** Desenvolvido pelos autores (2025).





O Enforcado, antagonista do jogo, é Rayetayah, um jovem injustamente executado por um crime que não cometeu. Após sua morte, ele retorna como uma entidade sobrenatural conhecida como “O Enforcado”, responsável pelos ataques demoníacos que assolam as cidades da região. Sua motivação principal é buscar justiça e confrontar o xerife que o condenou, sendo inicialmente apresentado como um vilão, também serve como uma figura que representa as consequências dos erros do passado.

O nome “Rayetayah” foi escolhido por remeter ao sentido de “boca pendurada”, associando-se ao modo como o personagem morreu dentro da narrativa. A imagem abaixo apresenta uma arte conceitual do personagem.

**Figura 14:** O Enforcado



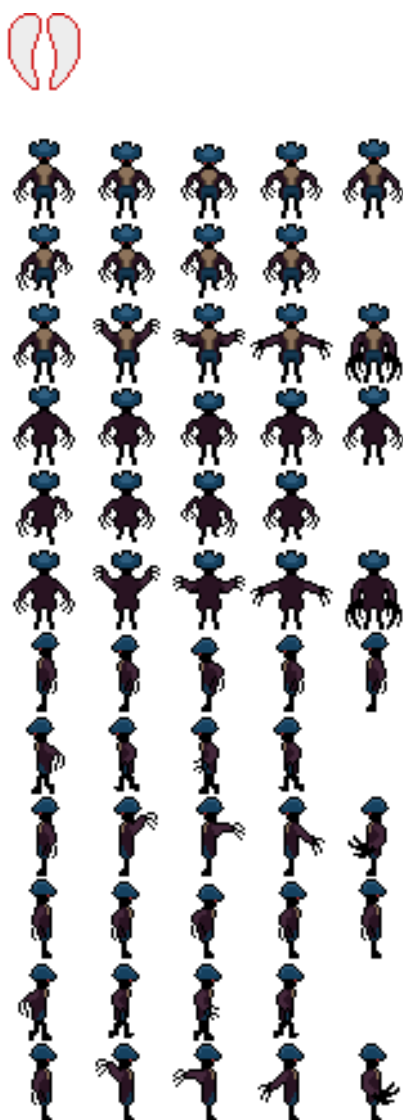
**Fonte:** Desenvolvido pelos autores (2025).

### 3.2.6 Inimigos

No contexto do projeto, foram desenvolvidos três inimigos distintos, cada um com comportamento e padrões de movimentação próprios.

O primeiro inimigo é o zumbi, que se movimenta de maneira direta, com velocidade moderada, possui vida considerável e ataca de perto. Seu comportamento tem como objetivo servir de obstáculo ao jogador, pois, devido à sua resistência, ele consegue suportar mais ataques sem ser derrotado. Além disso, seu ataque corpo a corpo é previsível, sendo acionado sempre que o jogador se aproxima, o que permite que o jogador antecipe suas ações. A imagem abaixo apresenta a aparência desse inimigo e suas animações:

**Figura 15:** Inimigo 1 - Zumbi



**Fonte:** Desenvolvido pelos autores (2025).

O olho demoníaco é um inimigo ágil e veloz, com pouca vida, e também ataca de perto. Ele realiza investidas rápidas em direção ao jogador sempre que este estiver ao seu alcance. Seu objetivo é atuar como distração, dificultando a precisão dos ataques do jogador devido ao seu rápido deslocamento e ao alto dano em proximidade. A figura abaixo demonstra sua aparência e animações no jogo.

**Figura 16:** Inimigo 2 - Olho Demoníaco



**Fonte:** Desenvolvido pelos autores (2025).

Por fim, o mago é um inimigo lento, com vida moderada, mas capaz de atacar à distância. Sua mecânica busca confundir o jogador, disparando projéteis rápidos, porém previsíveis. Combinado aos outros tipos de inimigos, ele permanece protegido a maior parte do tempo, podendo concentrar-se em causar alto dano ao jogador. A figura 16 exemplifica sua aparência e animações.

**Figura 17:** Inimigo 3 – Mago

**Fonte:** Desenvolvido pelos autores (2025).

### 3.2.7 Sons e Música

O design sonoro do projeto O Julgamento foi desenvolvido com o objetivo de reforçar a ambientação do Velho Oeste e dar clareza às ações do jogador durante a jogatina.

Os efeitos sonoros utilizados foram obtidos da plataforma Pixabay, que disponibiliza conteúdo livre de royalties. Posteriormente os sons foram ajustados com a ferramenta de edição *Audacity* melhor se adequarem aos ritmos e contextos do jogo.

Cada efeito sonoro cumpre um papel específico na comunicação com o jogador, contribuindo diretamente para o feedback imediato das ações, tornando a jogabilidade mais responsiva. A trilha musical, por sua vez, ajuda a estabelecer o clima das fases e a sensação de tensão crescente à medida que o jogador avança.

A lista dos arquivos de áudio consiste em:

- *sfx\_shot*: Som de disparo que toca quando o jogador atira. Biblioteca de som utilizada: Pixabay. Autor(a): Universfield.
- *sfx\_damage*: Som de impacto que toca quando o jogador ou inimigo perdem vida. Biblioteca de som utilizada: Pixabay. Autor(a): cabled\_mess.
- *sfx\_roll*: Som de impacto que toca quando o jogador efetua uma esquiva. Autor(a):
- *sfx\_reload*: Som de manuseio de arma que toca quando o jogador recarrega. Biblioteca de som utilizada: Pixabay. Autor(a): Dredile.
- *sfx\_walk*: Som de pegadas na areia que toca conforme o jogador anda. Biblioteca de som utilizada: Pixabay. Autor(a): Lukeo135.
- *sfx\_spawn*: Som de impacto que toca quando inimigos aparecem na cena. Biblioteca de som utilizada: Pixabay. Autor(a): wjl.
- *BGM\_1*: Música de fundo do jogo. Biblioteca de som utilizada: Pixabay. Autor(a): Emmraan.

### 3.2.8 Interfaces

A sequência de figuras abaixo lista as interfaces do jogo:

**Figura 18:** Interface - Menu



**Fonte:** Desenvolvido pelos autores (2025).

Figura 19: Interface – Ajustes



Fonte: Desenvolvido pelos autores (2025).

Figura 20: Interface – Créditos 1



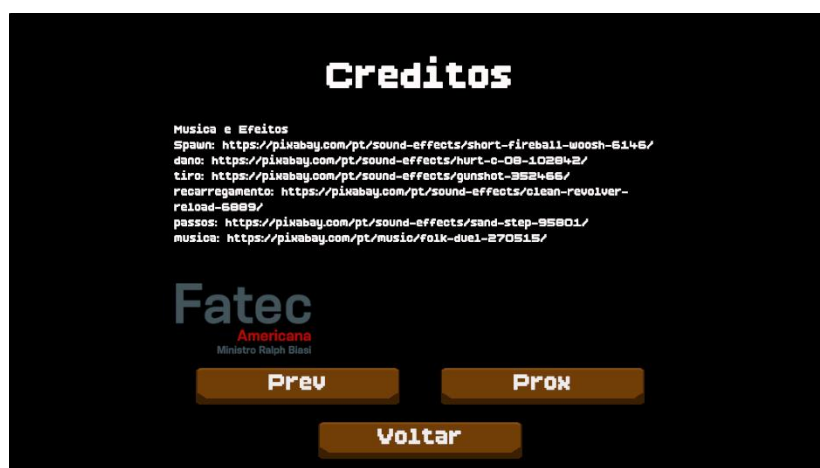
Fonte: Desenvolvido pelos autores (2025).

Figura 21: Interface – Créditos 2



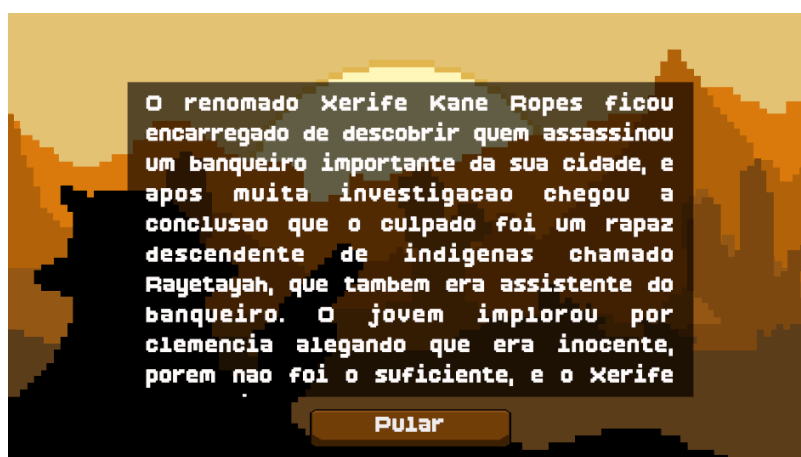
Fonte: Desenvolvido pelos autores (2025).

Figura 22: Interface – Créditos 3



Fonte: Desenvolvido pelos autores (2025).

Figura 23: Interface – Introdução



Fonte: Desenvolvido pelos autores (2025).

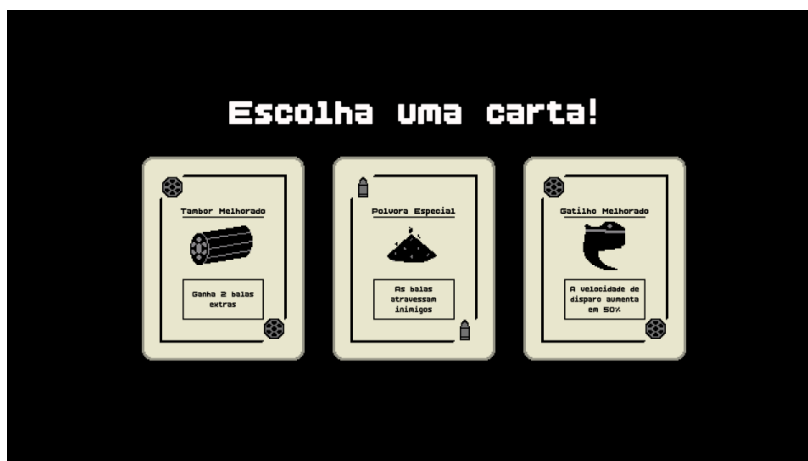
Figura 24: Interface – HUD



Fonte: Desenvolvido pelos autores (2025).



**Figura 25:** Interface – Escolha de Melhoria



Fonte: Desenvolvido pelos autores (2025).

**Figura 26:** Interface – Pausado

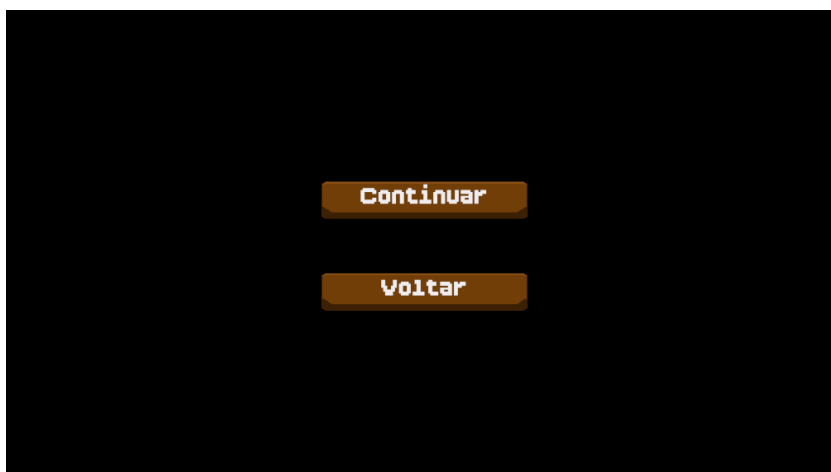


Fonte: Desenvolvido pelos autores (2025).

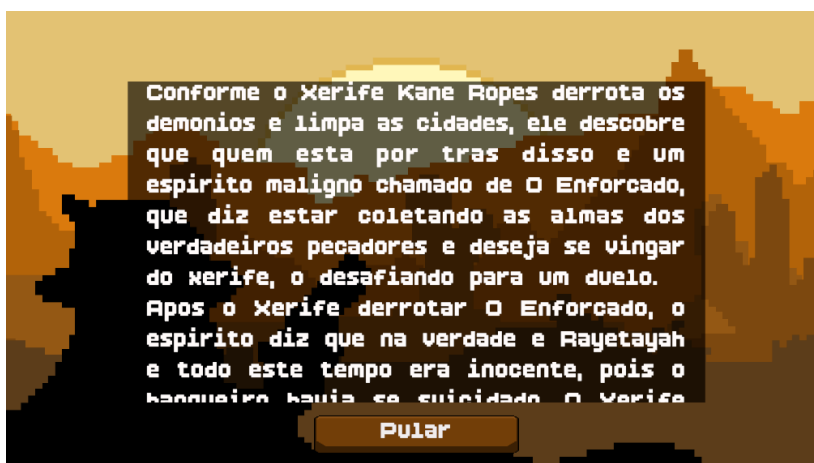
**Figura 27:** Interface – Inventário



Fonte: Desenvolvido pelos autores (2025).

**Figura 28:** Interface – Tela de Morte

Fonte: Desenvolvido pelos autores (2025).

**Figura 29:** Interface – Conclusão

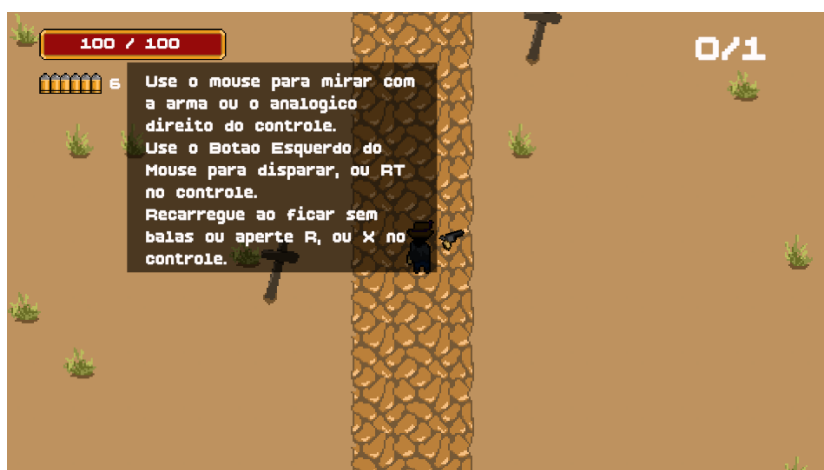
Fonte: Desenvolvido pelos autores (2025).

**Figura 30:** Interface – Tela de Vitória

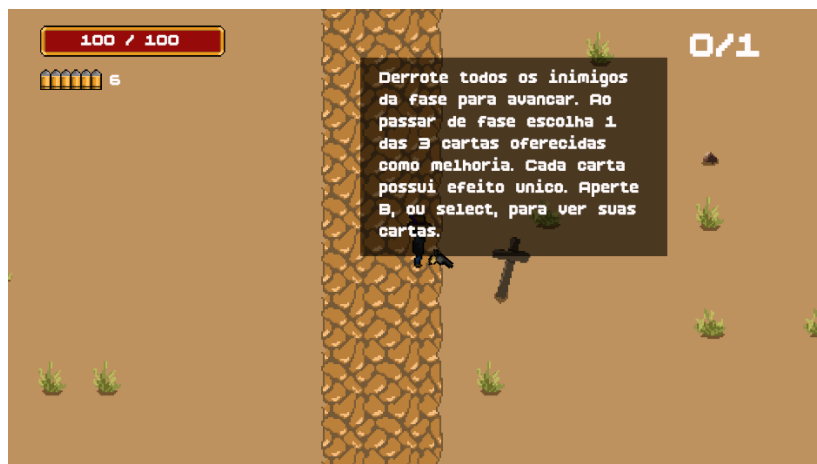
Fonte: Desenvolvido pelos autores (2025).

**Figura 31: Interface – Tutorial 1**

Fonte: Desenvolvido pelos autores (2025).

**Figura 32: Interface – Tutorial 2**

Fonte: Desenvolvido pelos autores (2025).

**Figura 33: Interface – Tutorial 3**

Fonte: Desenvolvido pelos autores (2025).

### 3.3 Mecânicas de Jogo

Por ser do gênero *roguelike*, O Julgamento se baseia em o jogador ter diversas tentativas de passar as fases até conseguir chegar ao final, considerando que cada uma delas será diferente da anterior pelas escolhas feitas, podendo mudar de caminho ou selecionar melhorias distintas.

Diferente de um *roguelite*, onde há uma progressão direta do jogador, com possibilidade de melhorar armas ou habilidades para as próximas tentativas, O Julgamento se enquadra como um *roguelike* tradicional, em que o estado inicial do personagem permanece o mesmo em todas as tentativas, diferenciando-se apenas pelas decisões tomadas durante a partida.

A progressão do jogador acontece de forma indireta, à medida que ele compreende o padrão de movimento dos inimigos, se adapta à precisão e reflexos necessários para desviar dos desafios de cada área e planeja quais melhorias serão mais úteis em cada tentativa.

Ao iniciar o jogo, a primeira fase funciona como um tutorial, ensinando e demonstrando as mecânicas de movimentação e ataque que serão necessárias para superar as próximas áreas.

Baseado em uma câmera *top down* (visão de cima para baixo), o jogador tem acesso a informações durante a jogabilidade que influenciam sua tomada de decisão rápida, como a quantidade de inimigos a enfrentar ou os projéteis a serem desviados, além de visualizar o caminho até a próxima fase ou as hordas de inimigos.

Durante as fases, o personagem pode se movimentar de maneira multidirecional, podendo ir para cima, baixo, esquerda, direita ou na diagonal. Além disso, pode realizar um rolamento também multidirecional, com o objetivo de desviar de ameaças. Para atacar, é possível mirar em um ângulo de 360 graus ao redor do personagem, atirando na direção desejada.

Quando confrontado com um inimigo ou projétil em sua direção, o jogador pode realizar um rolamento e tornar-se brevemente invulnerável para escapar do perigo. Essa invulnerabilidade é garantida pelos chamados *I-frames* (quadros de invulnerabilidade), durante os quais o personagem fica intangível.

Para gerar os diferentes inimigos, são utilizados *spawners*, ou invocadores, que instanciam os *prefabs* dos inimigos de forma aleatória quando o jogador entra na área de ativação correspondente. Esses invocadores estão distribuídos pelas fases, e o

jogador deve localizar cada um e derrotar os inimigos gerados, garantindo o progresso e o equilíbrio do desafio apresentado.

### 3.4 Progressão do Jogo

Antes de adentrar a primeira fase, é exibido ao jogador uma cena introduzindo-o a narrativa através de um breve texto explicando seus eventos. Após essa cena o jogo de fato começa.

O posicionamento das salas e o *layout* interno de cada uma delas, incluindo obstáculos e inimigos, é pré-definido manualmente, com uma estrutura modular que facilita a composição do mapa. À medida que o jogador derrota todos os inimigos de uma fase e alcança seu final, é possível obter uma carta de melhoria para o personagem.

No total, o jogador deve atravessar seis fases distintas, com a primeira delas servindo como tutorial. Cada fase aumenta o número de inimigos para que o jogador possa colocar em uso as melhorias que acumulou. As cinco primeiras fases garantem uma melhoria ao serem superadas, no entanto a sexta e última fase, que contém o maior número de inimigos, não oferece melhorias.

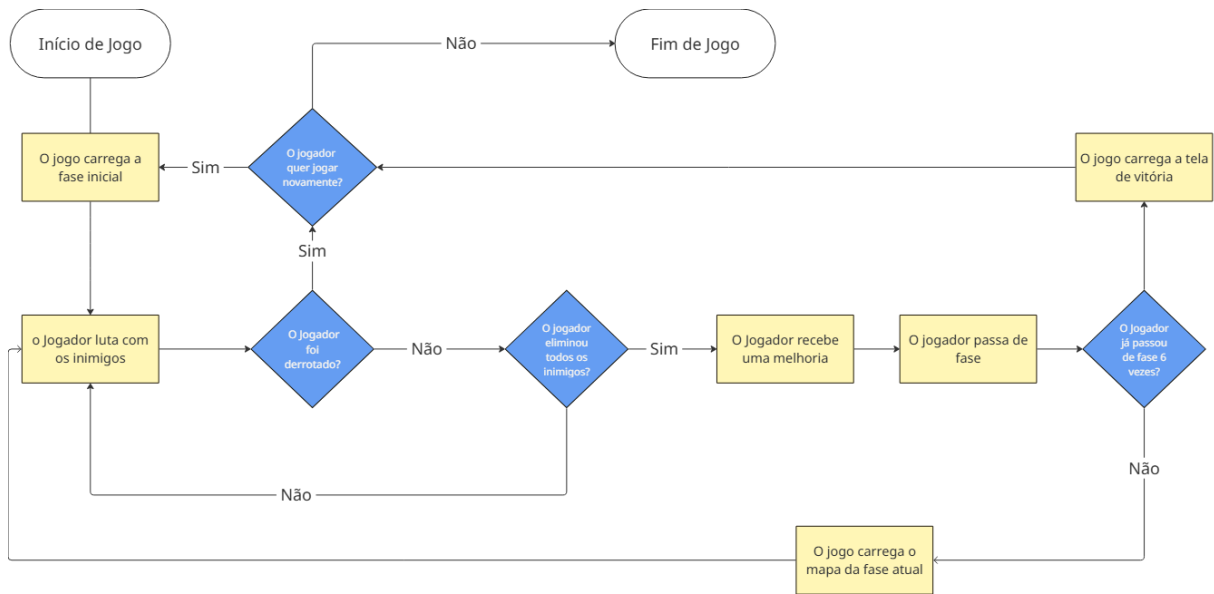
Ao fim da sexta fase, o jogador é levado para uma cena contendo o desfecho da narrativa e então a tela de vitória, onde ele pode escolher jogar novamente.

Se, por erro ou falha, Kane for derrotado durante a jornada, todas as melhorias adquiridas até então são perdidas, e o jogador tem a opção retornar ao início da primeira fase, podendo tentar novamente concluir a história e superar os inimigos. Com uma nova tentativa, o jogador pode experimentar diferentes combinações de melhorias ou focar em corrigir suas fraquezas anteriores.

Assim, ao ter o controle das escolhas de melhorias, o jogador personaliza o estilo de jogo que seguirá durante as fases, tornando cada tentativa única. Apesar da punição com a perda de progresso pela mecânica de *permadeath* (morte permanente), essa dinâmica faz parte do ciclo de aprendizado e progressão característico do gênero, o que não é diferente em O Julgamento.

O fluxograma na Figura 33 demonstra a progressão do jogo.

Figura 34: Fluxograma – Progressão

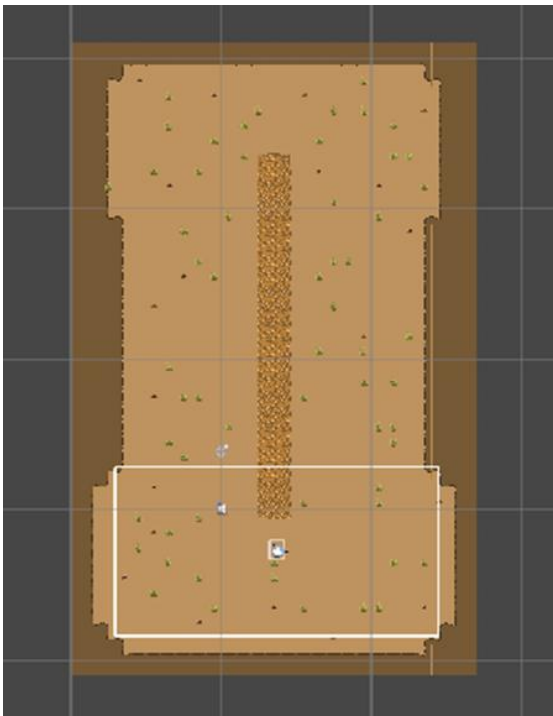


Fonte: Desenvolvido pelos autores (2025).

3.4.1 Cenários

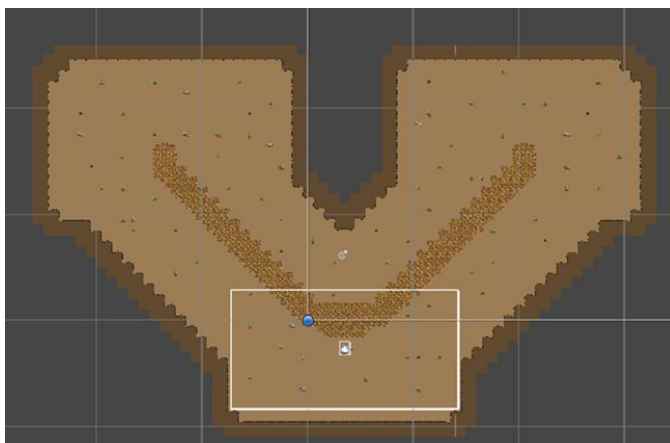
A sequência de figuras abaixo lista os cenários do jogo:

Figura 35: Cenário – Fase 1



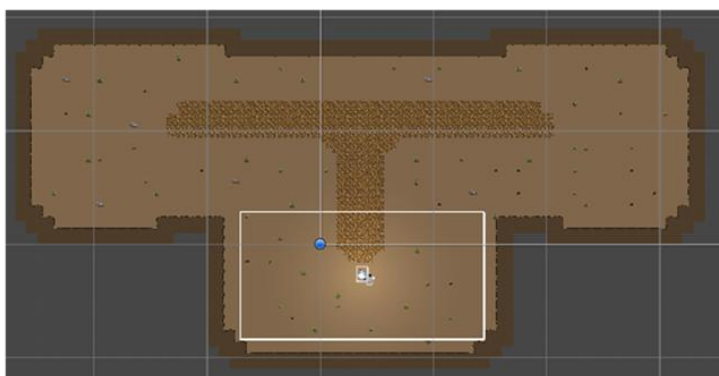
Fonte: Desenvolvido pelos autores (2025).

**Figura 36:** Cenário – Fase 2



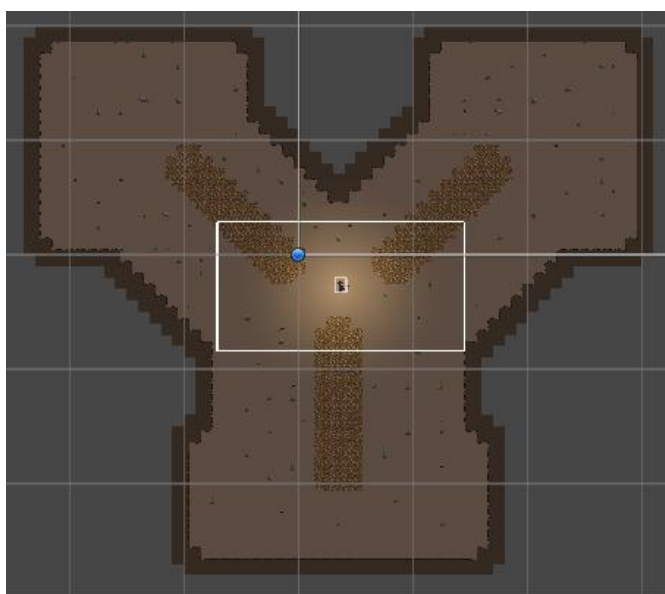
**Fonte:** Desenvolvido pelos autores (2025).

**Figura 37:** Cenário – Fase 3

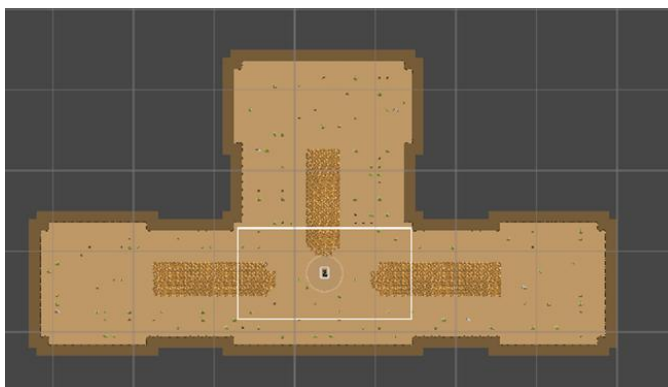


**Fonte:** Desenvolvido pelos autores (2025).

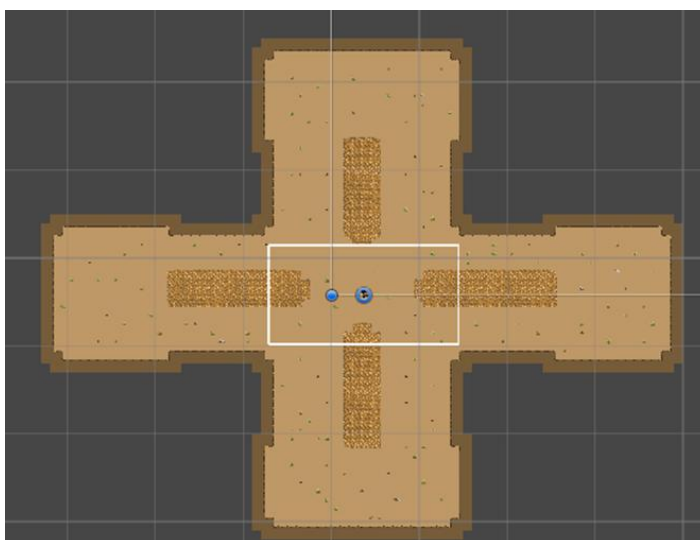
**Figura 38:** Cenário – Fase 4



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 39:** Cenário – Fase 5

**Fonte:** Desenvolvido pelos autores (2025).

**Figura 40:** Cenário – Fase 6

**Fonte:** Desenvolvido pelos autores (2025).

Utilizando o *Universal Render Pipeline da Unity*, em conjunto com a renderização 2D, é possível criar elementos de luz e sombra dentro das *scenes*. Com essa ferramenta, foi desenvolvido um sistema de anoitecer para o projeto, no qual, conforme o jogador avança pelas fases, o cenário vai escurecendo progressivamente. Nesse contexto, a única fonte de luz constante é aquela emitida pelo personagem principal, orientando sua movimentação e proporcionando um controle mais estratégico sobre a exploração do mapa.

Esse sistema tem como objetivo aumentar a imersão e a sensação de desafio, exigindo que o jogador se aproxime mais dos inimigos para identificá-los claramente durante o combate. A limitação da visão também intensifica o suspense e a atenção necessária em cada fase, reforçando o aspecto tático da jogabilidade.

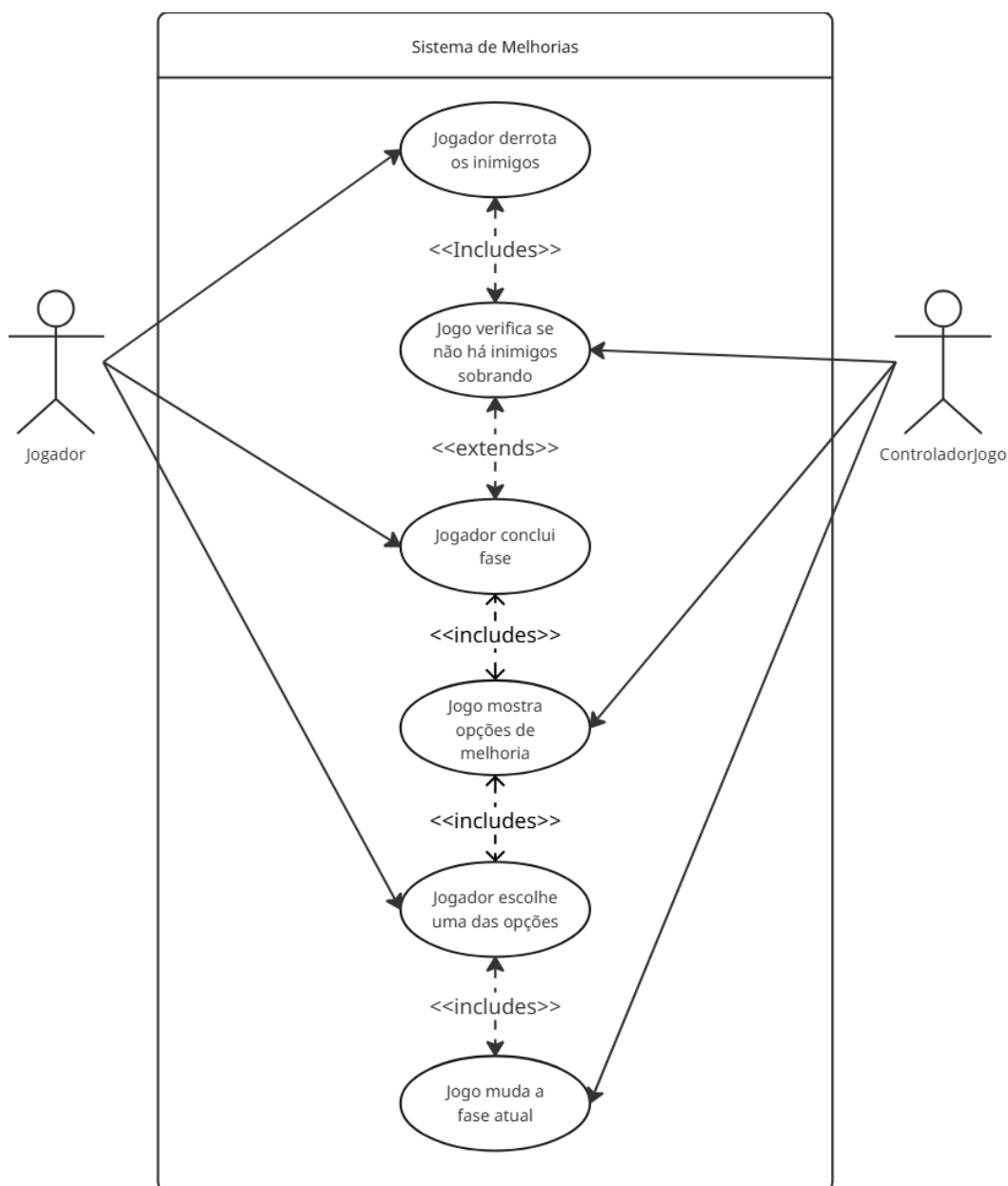


Além disso, os projéteis disparados pelo jogador também emitem luz, permitindo que eles sejam utilizados como ferramenta para revelar perigos à frente, como inimigos ocultos ou obstáculos no cenário. Dessa forma, a implementação do sistema de luz não apenas contribui para a estética visual do jogo, mas também integra diretamente com as mecânicas de combate e exploração, reforçando a experiência do jogador.

#### 3.4.1 Obtenção de Melhorias

O Julgamento conta com um sistema de melhorias progressivas, projetado para incentivar a adaptação estratégica do jogador a cada nova jogatina, reforçando os elementos de rejogabilidade característicos do gênero *roguelike*.

Conforme o jogador avança nas fases e derrota as hordas de inimigos em determinada área, ele é recompensado com a possibilidade de escolher uma melhoria para o personagem. Essa escolha é apresentada em forma de cartas, nas quais um número limitado de opções é disponibilizado ao final de cada fase. O jogador deve, então, selecionar apenas uma das cartas apresentadas, o que o obriga a tomar uma decisão. O diagrama de caso uso da figura 40 exemplifica esse sistema.

**Figura 41:** Caso de Uso – Sistema de Melhorias

**Fonte:** Desenvolvido pelos autores (2025).

Esse sistema de combinação recompensa o jogador por traçar estratégias de longo prazo durante sua jornada, permitindo criar sinergias entre as melhorias obtidas e construir uma configuração única para cada tentativa. Com isso, cada jogatina apresenta uma experiência distinta, onde o jogador equilibra o fator aleatório das cartas sorteadas com suas próprias decisões de combinação e personalização, tornando o progresso dinâmico e imprevisível a cada nova partida.

### 3.4.2 Melhorias

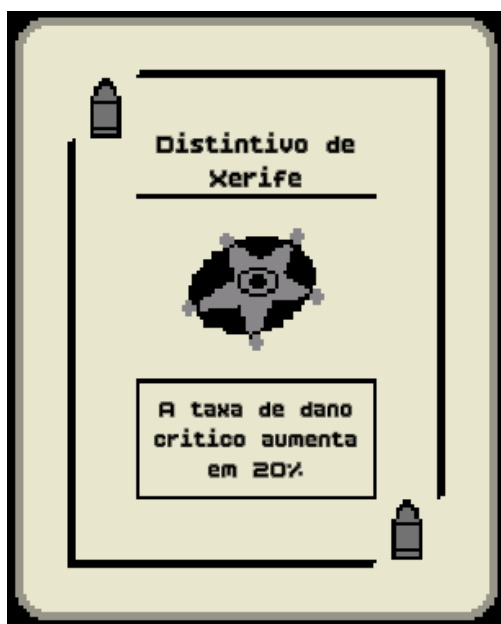
A sequência de figuras abaixo lista todas as melhorias do jogo:

**Figura 42:** Melhoria – Balas de Prata



Fonte: Desenvolvido pelos autores (2025).

**Figura 43:** Melhoria – Distintivo de Xerife



Fonte: Desenvolvido pelos autores (2025).

**Figura 44:** Melhoria – Pólvora Especial



Fonte: Desenvolvido pelos autores (2025).

**Figura 45:** Melhoria – Cinto de Balas



Fonte: Desenvolvido pelos autores (2025).

**Figura 46:** Melhoria – Gatilho Melhorado



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 47:** Melhoria – Tambor Melhorado



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 48:** Melhoria – Chapéu de Vaqueiro



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 49:** Melhoria – Poncho



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 50:** Melhoria – Cantil



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 51:** Melhoria – Charuto



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 52:** Melhoria – Bandana



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 53:** Melhoria – Botas de Vaqueiro



**Fonte:** Desenvolvido pelos autores (2025).



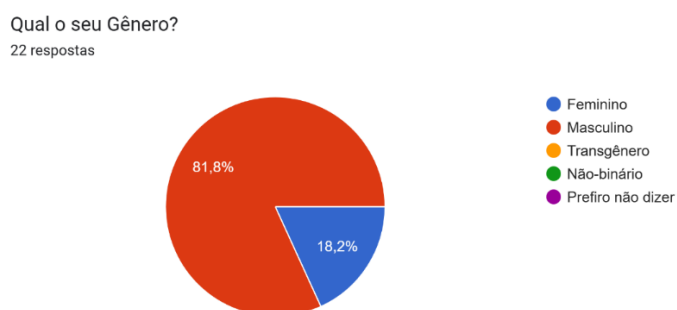
## 4. RESULTADOS

### 4.1 Teste Beta Realizado

O teste beta de O Julgamento foi realizado no dia 14 de novembro de 2025 em um dos laboratórios da Fatec Americana, o qual contou com a participação de 22 alunos. O objetivo desse teste foi obter feedbacks referentes a jogabilidade, desempenho, imersão e as artes do jogo desenvolvido.

Os participantes foram divididos de acordo com o gênero, conforme demonstra a figura 54, e dentre as 22 pessoas que responderam à pesquisa, 18 são do sexo masculino e 4 são do sexo feminino. A figura 55 divide os participantes de acordo com a faixa etária, de modo que 72% dos participantes têm entre 19 e 24 anos, representando um público majoritariamente jovem. Outros 18,2% estão entre 25 e 29 anos, à medida que 9,1% possuem idade entre 30 e 34 anos.

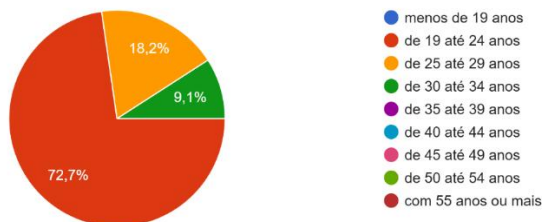
**Figura 54:** Pesquisa Teste Beta – Pergunta 1



**Fonte:** Desenvolvido pelos autores (2025).

**Figura 55:** Pesquisa Teste Beta – Pergunta 2

Em qual faixa etária você está?  
22 respostas

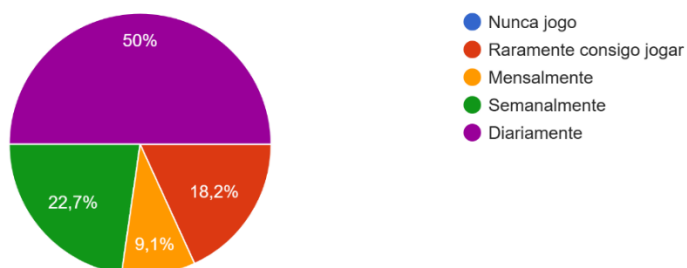


**Fonte:** Desenvolvido pelos autores (2025).

Quanto à frequência com que os participantes jogam, 50% afirmam jogar diariamente, e os demais afirmam jogar com menos frequências. As porcentagens exatas podem ser vistas na figura abaixo.

**Figura 56:** Pesquisa Teste Beta – Pergunta 3

Com que frequência você costuma jogar Jogos Digitais?  
22 respostas



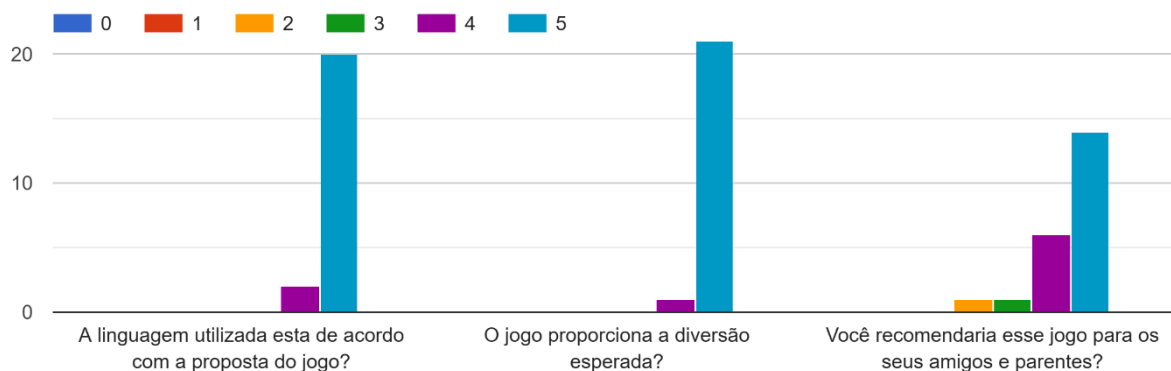
**Fonte:** Desenvolvido pelos autores (2025).

A avaliação da opinião geral do jogo pode ser feita através da figura 57. No quesito uso da linguagem, 20 avaliadores deram nota máxima e 2 atribuíram nota 4, sendo assim, uma grande parte concorda que a linguagem utilizada está de acordo com a proposta. Em relação à diversão esperada, 21 pessoas deram nota 5 e apenas 1 deu nota 4, evidenciando excelente aceitação.

Quanto à pergunta “Você recomendaria este jogo a amigos ou familiares?”, houve maior variação: 14 notas 5, 6 notas 4, 1 nota 3 e 1 nota 2. Portanto, maior parte recomendaria totalmente, enquanto a outra parcela não recomendaria totalmente, mas também não descartaria a possibilidade.

**Figura 57:** Pesquisa Teste Beta – Seção Opinião Geral

Opinião geral sobre o Jogo

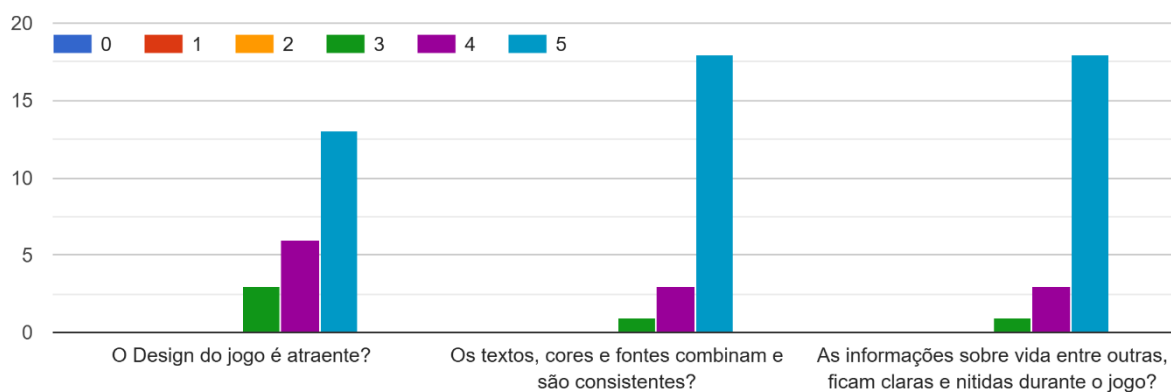
**Fonte:** Desenvolvido pelos autores (2025).

A figura 58 também evidencia avaliações positivas. Em relação ao design do jogo, 12 pessoas deram nota máxima e 6 pessoas deram nota 4, dessa forma gerando respostas mistas, mas ainda otimistas.

Referente as duas perguntas “Textos, cores e fontes combinam e são consistentes?” e “As informações sobre vida entre outras, ficam claras e nítidas durante o jogo?”, grande parte dos participantes demonstrou satisfação.

**Figura 58:** Pesquisa Teste Beta – Seção Design do Jogo

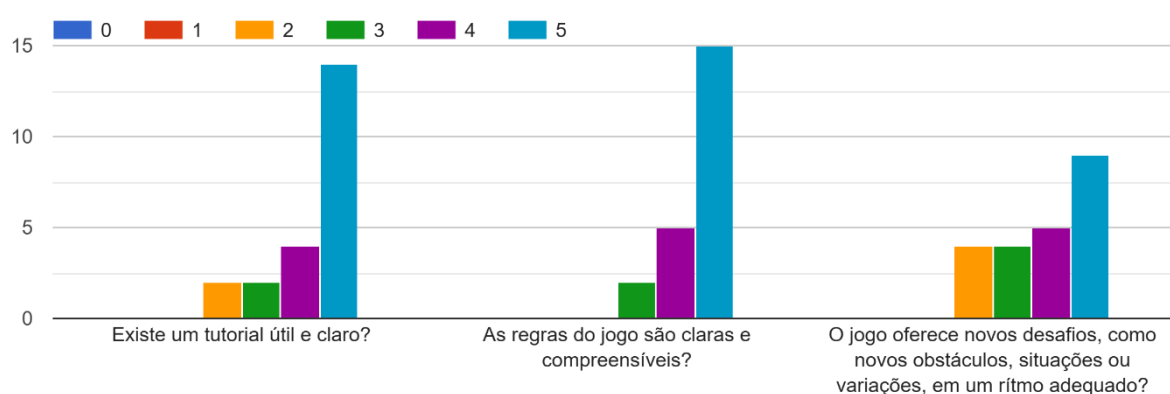
Design do Jogo

**Fonte:** Desenvolvido pelos autores (2025).

Na seção sobre jogabilidade, as perguntas referentes ao tutorial e regras do jogo tiveram respostas extremamente positivas, com a maioria afirmando que as instruções oferecidas são claras compreensíveis. Por outro lado, o ritmo do jogo recebeu avaliações mistas, uma vez que durante a execução do teste beta as fases disponíveis estavam muito similares. Os gráficos podem ser vistos na figura 59.

**Figura 59:** Pesquisa Teste Beta – Seção Jogabilidade

Jogabilidade

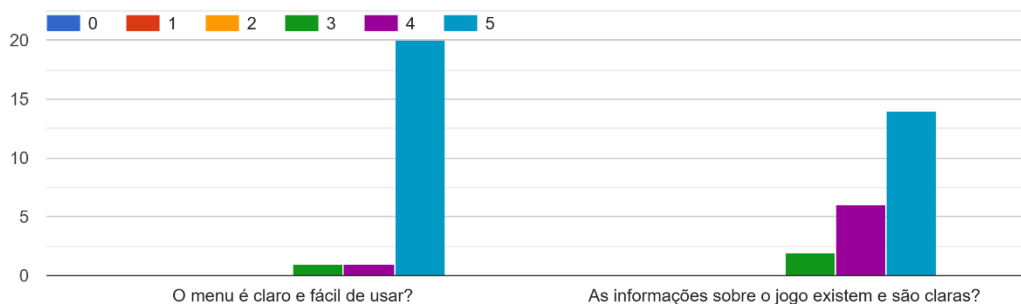


**Fonte:** Desenvolvido pelos autores (2025).

A figura 60 contém os gráficos da seção menu e interação. As avaliações acerca da facilidade de se utilizar o menu e clareza das informações do jogo também foram positivas, com 20 participantes dando nota 5 para a pergunta “O menu é claro e fácil de usar?” e 14 participantes dando nota 5 para a pergunta “As informações sobre o jogo existem e são claras?”.

**Figura 60:** Pesquisa Teste Beta – Seção Menu e Interação

Menu e interação

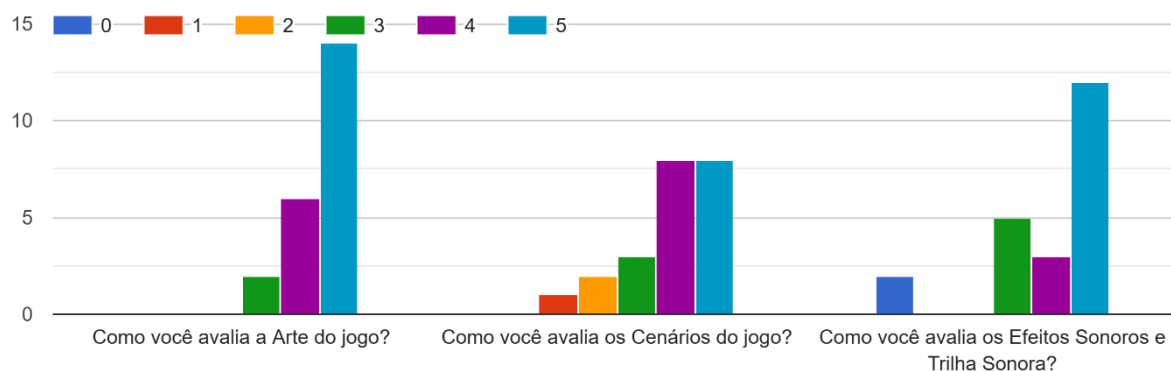
**Fonte:** Desenvolvido pelos autores (2025).

Por fim a seção de arte e som, evidenciada na figura 61. A arte do jogo foi bem recebida, com 14 participantes afirmando estarem satisfeitos. Entretanto, os cenários do jogo receberam avaliações equilibradas, pois como foi dito anteriormente, o teste beta utilizou fases com estruturas muito parecidas, incluindo os cenários.

Quanto aos efeitos sonoros e trilha sonora, houve avaliações majoritariamente positivas, com 12 participantes dando nota 5, mas também houve avaliações negativas, com 2 participantes dando nota 0.

**Figura 61:** Pesquisa Teste Beta – Seção Arte e Som

Arte e Som

**Fonte:** Desenvolvido pelos autores (2025).

## 5. DOWNLOAD DO JOGO

O *download* do jogo pode ser feito através do *link*: <https://cauepazz.itch.io/o-julgamento> no *website* “itch.io”.

## CONSIDERAÇÕES FINAIS

O projeto O Julgamento cumpriu o objetivo de aprofundar nas técnicas de desenvolvimento de um jogo *roguelike* que integra narrativa cíclica e mecânicas de repetição e aleatoriedade. A jornada do xerife Kane Ropes, marcada por temas como culpa e redenção, foi construída de forma a reforçar essas mecânicas, proporcionando uma experiência interativa coerente e imersiva.

Durante o desenvolvimento, foram aplicados conhecimentos técnicos e teóricos adquiridos ao longo da formação, utilizando ferramentas como *Unity*, *LibreSprite* e *Visual Studio Code*. A escolha do estilo visual em *pixel art*, da câmera *top down* e do sistema de progressão reforçou os aspectos de jogabilidade e narrativa.

Além de ser uma realização acadêmica, o jogo demonstrou o potencial dos jogos digitais como meio expressivo. A inspiração em títulos como *Hades* e *The Binding of Isaac* ajudou a consolidar decisões de design, contribuindo para um produto final coeso e funcional.

Para futuras atualizações no jogo, se destacam:

- Maior gama de melhorias: Mais opções para o jogador poder escolher, aumentando o valor de rejogabilidade e reforçando integração da narrativa as mecânicas de jogo estabelecidas.
- Combinação de melhorias: Melhorias que possuem diferentes níveis, se juntam aos efeitos ou só desbloqueiam depois da obtenção de outras. Tais mudanças aprofundariam as mecânicas de jogo e a imersão do jogador ao oferecer escolhas mais complexas.
- Luta contra chefões: Inimigos desafiadores e que testam as habilidades do jogador, permitindo que a progressão do jogo seja transmitida de forma natural.
- Aprofundamento na narrativa: Desenvolver ainda mais a história do jogo ao adicionar cenários diferentes, mais personagens e diálogos, contribuindo para a compreensão e desenrolar da narrativa.

## REFERÊNCIAS

AUDACITY TEAM. **Audacity**: editor de áudio livre e gratuito. Disponível em: <https://www.audacityteam.org>. Acesso em: 14 maio 2025.

CABLED\_MESS. Hurt C-08. Efeito sonoro. Pixabay, 2022. Disponível em: <https://pixabay.com/pt/sound-effects/hurt-c-08-102842/>. Acesso em: 13 nov. 2025.

DODGE ROLL. **Enter the gungeon** [Jogo eletrônico]. PC. Devolver Digital, 2016. Disponível em: [https://store.steampowered.com/app/311690/Enter\\_the\\_Gungeon/](https://store.steampowered.com/app/311690/Enter_the_Gungeon/). Acesso em: 04 maio 2025.

DREDILE. Clean Revolver Reload. Efeito sonoro. Pixabay, 2021. Disponível em: <https://pixabay.com/pt/sound-effects/clean-revolver-reload-6889/>. Acesso em: 13 nov. 2025.

EMMRAAN. Duel. Música. Pixabay, 2024. Disponível em: <https://pixabay.com/pt/music/folk-duel-270515/>. Acesso em: 13 nov. 2025.

GAME DEV GARNET. AStar Pathfinding Tutorial. YouTube, 2025. Disponível em: <https://www.youtube.com/watch?v=UHnOW-OimLQ&t=935s>. Acesso em: 26 out. 2025.

HUIZINGA, Johan. **Homo ludens: o jogo como elemento da cultura**. 5. ed. São Paulo: Perspectiva, 2001.

LUKEO135. *Sand Step*. Efeito sonoro. Pixabay, 2022. Disponível em: <https://pixabay.com/pt/sound-effects/sand-step-95801/>. Acesso em: 26 nov. 2025.

MANA POOL. **Mana pool guide to roguelikes**. Disponível em: <https://www.manapool.co.uk/mana-pool-guide-to-roguelikes>. Acesso em: 15 abr. 2025.

MCMILLEN, Edmund; HIMSL, Florian. **The binding of Isaac** [Jogo eletrônico]. Windows. USA: Edmund McMillen, 2011. Disponível em: [https://store.steampowered.com/app/113200/The\\_Binding\\_of\\_Isaac/](https://store.steampowered.com/app/113200/The_Binding_of_Isaac/). Acesso em: 04 maio 2025.

MICROSOFT CORPORATION. **Visual studio code**: editor de código-fonte. Disponível em: <https://code.visualstudio.com>. Acesso em: 14 maio 2025.

ROGERS, Scott. **Level Up: um guia para o design de grandes jogos**. São Paulo: Blucher, 2012. 494 p.

ROGUEBASIN. **What a roguelike is**. Disponível em: [https://www.roguebasin.com/index.php?title=What\\_a\\_roguelike\\_is](https://www.roguebasin.com/index.php?title=What_a_roguelike_is). Acesso em: 15 abr. 2025.



SUPERGIANT GAMES. **Hades** [Jogo eletrônico]. PC. Supergiant Games, 2020. Disponível em: <https://store.steampowered.com/app/1145360/Hades/>. Acesso em: 04 maio 2025.

TOY, Michael; WICHMAN, Glenn; ARNOLD, Ken; LANE, Jon. Rogue [Jogo eletrônico]. Versão original. Santa Cruz: A.I. Design, 1980. Disponível em: <https://github.com/Davidslv/rogue>. Acesso em: 04 maio 2025.

UNIVERSFIELD. Gunshot. Efeito sonoro. Pixabay, 2025. Disponível em: <https://pixabay.com/pt/sound-effects/gunshot-352466/>. Acesso em: 13 nov. 2025.

UNITY TECHNOLOGIES. Cinemachine. Disponível em: <https://docs.unity3d.com/Packages/com.unity.cinemachine@3.1/manual/index.html>. Acesso em: 26 out. 2025.

UNITY TECHNOLOGIES. Input System. Disponível em: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.15/manual/index.html>. Acesso em: 26 out. 2025.

UNITY TECHNOLOGIES. ScriptableObject. Disponível em: <https://docs.unity3d.com/6000.2/Documentation/Manual/class-ScriptableObject.html>. Acesso em: 26 out. 2025.

UNITY TECHNOLOGIES. Tilemap. Disponível em: <https://docs.unity3d.com/Packages/com.unity.2d.tilemap@1.0/manual/index.html>. Acesso em: 26 out. 2025.

UNITY TECHNOLOGIES. **Unity**: Manual oficial e documentação da engine Unity. Disponível em: <https://unity.com>. Acesso em: 14 maio 2025.

WJL. Short-Fireball-Woosh. Efeito sonoro. Pixabay, 2021. Disponível em: <https://pixabay.com/pt/sound-effects/short-fireball-woosh-6146/>. Acesso em: 13 nov. 2025

YUKIPIXELS. BoldPixels [Fonte pixel]. Unity Asset Store, 2025. Disponível em: <https://assetstore.unity.com/packages/2d/fonts/boldpixels-332078>. Acesso em: 26 out. 2025.

## **APÊNDICE A – Introdução e Conclusão da Narrativa**

### **Introdução:**

O renomado Xerife Kane Ropes ficou encarregado de descobrir quem assassinou um banqueiro importante da sua cidade, e após muita investigação chegou à conclusão que o culpado foi um rapaz descendente de indígenas chamado Rayetayah, que também era assistente do banqueiro. O jovem implorou por clemência alegando que era inocente, porém não foi o suficiente, e o Xerife sentenciou sua morte por enforcamento.

Com o caso dado como encerrado, alguns meses se passaram até que Kane Ropes começou a receber cartas de outras cidades pedindo por ajuda, dizendo que supostos demônios estão atacando, e dessa forma o Xerife parte para desvendar este mistério.

### **Conclusão:**

Conforme o Xerife Kane Ropes derrota os demônios e limpa as cidades, ele descobre que quem está por trás disso é um espírito maligno chamado de O Enforcado, que diz estar coletando as almas dos verdadeiros pecadores e deseja se vingar do xerife, o desafiando para um duelo.

Após o Xerife derrotar O Enforcado, o espírito diz que na verdade Rayetayah e todo este tempo era inocente, pois o banqueiro havia se suicidado. O Xerife também descobre que as vezes que pensou ter morrido, ele havia sido revivido pelo espírito para que pudessem se encontrar.

Com vergonha, Kane Ropes reflete sobre seus atos e decide se aposentar, resultando na libertação do espírito do jovem.