



Departamento de Design - Instituto de Artes da UnB

Isabela Leal do Rosario

Ralison Marçal Resende

Trabalho de Conclusão de Curso

Melius: Game Design de um jogo Metroidvania

Brasília - DF

Agosto de 2024



Departamento de Design - Instituto de Artes da UnB

Isabela Leal do Rosario

Ralison Marçal Resende

**Melius: Game Design de um jogo
Metroidvania.**

Relatório apresentado ao Departamento de Design da Universidade de Brasília - UnB como requisito para obtenção do diploma de Bacharel em Design, sob a orientação do Professor Tiago Barros Pontes e Silva.

Brasília - DF

Agosto de 2024

Dedicatória

A todos os estudantes que aspiram se tornar profissionais de Game Design, áreas associadas ao desenvolvimento de jogos ou que simplesmente tem paixão por jogos digitais e desejam aprender mais sobre o seu desenvolvimento. Esperamos que o processo descrito neste relatório possa auxiliar e direcionar em como se dá o processo de Game Design, e que as soluções encontradas por nós possam ser aplicadas e expandidas. Desejo-lhes muita sorte e sucesso no projeto que estiverem desenvolvendo.

Agradecimentos

Gostaríamos de agradecer a todos os nossos familiares e amigos que nos acompanharam e apoiaram durante todo o processo de criação do jogo e deste relatório.

Agradecemos imensamente aos membros do grupo que participaram diretamente na produção do Melius durante a disciplina de Design de Jogos: Raimundo Marques, Luísa Rocha, Gabriel Fontenelle, Moisés Felipe e Maurício Ribeiro. Em especial aos membros que permaneceram para as etapas de revisão e expansão do jogo. Agradecemos também ao nosso amigo, Lucas Melo por oferecer a Oficina de Animação Rebellious, repleta de conhecimentos sobre animação para jogos, e também pela oportunidade de participar no desenvolvimento do seu jogo e ter uma experiência prévia ao desenvolvimento do Melius.

Aos professores Gabriel Lyra, Felix Alves, Daniel Fernandes e Renata Rinaldi por permitirem o desenvolvimento da Zine sobre o universo do jogo como entrega final da disciplina de Análise da Linguagem e Produção de Quadrinhos, e por nos guiar durante a produção. Agradecemos também à colega Mônica por colaborar no desenvolvimento e ilustração desse quadrinho.

Ao Tiago Barros, por nos ensinar sobre o desenvolvimento do jogo na disciplina de Design de Jogos e por nos oferecer a oportunidade de criar o TCC com este tema. Por nos acompanhar durante toda essa jornada dedicando muitas horas para nos guiar e esclarecer nossas dúvidas, sempre com muita atenção e cuidado, e sempre checando como estávamos nos sentindo e se estávamos satisfeitos com o projeto.

Agradecemos também ao nosso amigo, Alexandre de Carvalho que muito nos ajudou na produção deste relatório, através de longas conversas e sugestões.

Resumo

O presente relatório descreve de forma detalhada como se deu o processo de Design, mais especificamente Game Design, para a criação do jogo eletrônico Melius. Ao longo do relatório, descrevemos cada uma das etapas envolvidas no processo de desenvolvimento, detalhando como ocorreram e sua importância para a criação do jogo, entre elas, a concepção e definição do jogo, a pesquisa de referências, a definição das regras e mecânicas, a criação de mapas, cenários, personagens e animações. A ideia do projeto surgiu no contexto da disciplina Design de Jogos, ofertada pelo Departamento de Design na Universidade de Brasília. Tínhamos como objetivo a criação de um jogo do gênero Metroidvania que fosse divertido e proporcionasse uma experiência completa para o jogador, e, ao mesmo tempo, experimentar e aprender sobre o desenvolvimento de jogos e as áreas do Design envolvidas nesse processo. Ao final da disciplina de Design de Jogos, alcançamos um resultado satisfatório: um Metroidvania com início, meio e fim, que oferece uma experiência divertida ao jogador. No entanto, identificamos, através de uma série de testes, diversas áreas que poderiam ser aprimoradas para tornar a experiência de *gameplay* mais agradável ao jogador. Com isso, iniciamos um ciclo de revisão, buscando melhorar esses aspectos e expandir o escopo do projeto, chegando assim à uma segunda versão do jogo, que, embora ainda não esteja completa, já possibilita avaliar algumas melhorias no *gameplay*.

Palavras chave: game design, jogo eletrônico, metroidvania, pixel art, jogo 2D, trabalho em equipe.

Abstract

This article describes in detail how the design process, more specifically game design processes took place for the creation of the videogame Melius. We describe in depth, throughout the article, each of the phases involved in the development process, detailing how they unfolded and their importance for the creation of the game, including the conception and definition of the game, research of references, definition of rules and mechanics, creation of maps, scenarios, characters and animations. The idea for the project came about in the context of the Design de Jogos course, offered by the Design Department at the Universidade de Brasília. Our goal was to create a game in the Metroidvania genre that would be fun and provide a complete experience for the player, while at the same time experimenting and learning about game development and the Design subjects involved in this process. At the end of the Design de Jogos course, we achieved a satisfactory result: a Metroidvania with a beginning, middle and end, which offers a fun experience for the player. However, through a series of tests, we identified a number of areas that could be improved to make the gameplay experience more enjoyable for the player. So we began a revision cycle, seeking to improve these aspects and expand the scope of the project, thus arriving at a second version of the game, which although not yet complete, already makes it possible to evaluate some gameplay improvements.

Keywords: game design, videogame, metroidvania, pixel art, 2D game, teamwork.

Glossário

Aseprite: É um software de desenho e animação de *pixel art*, voltado principalmente para a criação de assets para jogos.

Assets: São os elementos que constituem um jogo, como arte, ilustrações, animações, efeitos sonoros, música, *scripts*, texto etc.

Boss: Ou *bosses* é uma palavra que vem do inglês e é usada para se referir a um inimigo particularmente poderoso que o jogador deve enfrentar e derrotar, geralmente ao final de uma fase, para poder continuar a avançar no jogo.

Brainstorming: É uma técnica que, por meio do compartilhamento espontâneo de ideias, busca encontrar a solução para um problema ou gerar possibilidades criativas.

Combo: Ou “sequência de golpes”, é uma técnica muito utilizada em jogos que possuem combate. Ela consiste em realizar e conectar uma série de movimentos específicos de forma rápida e precisa, resultando em uma combinação de golpes que causam dano significativo ao oponente ou inimigo.

Companion: Se refere a um personagem no jogo na qual o jogador tem controle limitado sobre, muitas vezes indiretamente ou diretamente. Esse personagem acompanha o jogador durante o *gameplay*.

Concept art: É um processo essencial na produção de filmes, animações, jogos, quadrinhos e em diversas outras indústrias do entretenimento. Sendo uma etapa da pré-produção em que ideias, descrições e roteiros são expressados e materializados visualmente em diversas alternativas visuais ou *concepts*, que depois são refinados até que seja desenvolvida uma aparência coesa para os personagens, cenários, vestimentas, etc.

Design de personagem: O nome dado ao processo no qual todas as características de um personagem, como história, personalidade, características físicas, etc. São sintetizados e definidos visualmente em uma aparência única que consiga transmitir quem ele é.

Feedback: É uma palavra que vem do inglês, e se refere ao ato de fornecer sugestões, informações, opiniões e avaliações sobre o desempenho, trabalho ou comportamento de algo ou alguém, de forma a permitir melhorias.

Figma: É uma ferramenta de edição em vetor e prototipagem, que permite o trabalho e acesso simultâneo de vários usuários, geralmente usada para a criação de projetos de *design*.

Flippaclip: É um aplicativo gratuito de animação para celulares e tablets, voltado para iniciantes.

Frame: Ou quadro, é uma única imagem da sequência de imagens que compõem a animação.

Gameplay: Trata-se da experiência do jogador ao jogar e interagir com todas as regras e mecânicas do jogo.

GDD: Abreviação para *game design document*, ou documento de design de jogo, é um documento escrito contendo em detalhes todas as informações e decisões de Design relevantes para a produção do jogo como: mecânica do jogo, a narrativa, os personagens, o design de níveis, os gráficos e outros aspectos importantes.

Google Docs: É um processador de texto on-line que permite criar e formatar documentos e colaborar com outras pessoas

Hit: É a nomenclatura utilizada para sinalizar a colisão de duas *hitboxes*.

Hitbox: Ou “caixa de acerto”, é a indicação da área de contato de objetos e *assets* do jogo.

HUD: Ou *heads up display*, é um conjunto de elementos gráficos projetados na tela para transmitir informações ao jogador durante a execução do jogo.

Insight: É uma palavra de origem inglesa que descreve o ato de ser ter uma compreensão súbita sobre algo, de forma a gerar novas perspectivas, ideias e soluções.

Jogo 2D: É um tipo de jogo eletrônico no qual a movimentação tanto do jogador quanto da câmera é limitada a apenas dois eixos, geralmente são compostos de *assets* bidimensionais.

Jogo 3D: É um tipo de jogo eletrônico que utiliza gráficos tridimensionais para criar ambientes, personagens e objetos, dando origem a um mundo com três dimensões no qual, geralmente, tanto o jogador, quanto a câmera são capazes de se mexer em três eixos.

Jogo Plataforma: Também conhecido como *jump 'n' run*, é um gênero de jogos eletrônicos de ação em que o objetivo principal é mover o personagem do jogador entre pontos em um ambiente. Os jogos de plataforma são caracterizados por níveis que consistem em terrenos

irregulares e plataformas suspensas de altura variável que exigem pular e escalar para atravessar.

Level design: É nome dado à uma etapa do desenvolvimento de jogos, no qual se criam fases, mapas, missões e objetivos para o jogador. É o momento onde é criado o ambiente onde se passará o jogo, com todos seus obstáculos, desafios, objetivos e formas do jogador passar por ele. Também é o nome dado ao resultado final de toda essa tomada de decisões.

Lineart: É o nome dado para todas as linhas (internas ou externas) utilizadas para dar forma ao desenho.

Low fantasy: É um subgênero de fantasia um pouco mais preso à realidade, onde o mundo e os personagens seguem uma linha mais realista e os elementos mágicos aparecem de forma mais moderada.

Mecânica: Se refere ao conjunto de definições e sistemas que ditam como funciona o jogo e como se dá a interação do jogador com esse jogo.

Metroidvania: É um gênero de jogo 2D de plataforma e exploração que contém um grande mapa interconectado, onde as áreas são liberadas de forma não linear conforme o jogador adquire novos movimentos e habilidades (exemplo, um pulo duplo que permite alcançar uma plataforma mais alta). O nome vem da junção de Metroid e Castlevania, dois pioneiros no gênero.

Missão: Em jogos, é um termo utilizado para tarefas dadas ou objetivos a se cumprir pelo jogador no jogo.

Mood: É uma palavra de origem inglesa, que traduzida ao português significa “humor” ou “sentimento”. Nesse caso é utilizado com uma conotação artística, onde a obra evoca no observado um sentimento específico a partir das cores e tom da peça.

Moodboard: É um arranjo de referências visuais, auditivas e textuais que auxiliam na criação de um projeto, permitindo a visualização de um estilo particular ou um conceito, e servido de base para a definição do estilo do projeto.

NPC: É uma sigla para “*non-playable character*” ou em português “personagem não jogável” e se refere a todos os personagens presentes no jogo na qual o jogador não tem controle direto ou indireto sobre.

Parallax: É a diferença de posição do objeto em relação a visão. Em um jogo 2d isso é feito de maneira com que camadas mais próximas passem mais rápido pela câmera que camadas mais distantes, criando uma sensação de tridimensionalidade em um mundo 2d.

Photoshop: É um *software* gráfico para edição de imagens, criação de arte digital, design gráfico e animações.

Pixel art: É um estilo de arte digital, no qual as imagens são criadas usando quadrados chamados de *pixels*, de forma que os mesmos fiquem claramente visíveis. É um estilo que remete a estética originária dos computadores de 8 e 16 *bits*, que tinham uma resolução (número de *pixels* que compõem a tela) limitada, e que, portanto, só podiam gerar imagens onde os mesmos ficavam visíveis.

Playtest: Refere-se ao processo de se testar um jogo em desenvolvimento para que se identifiquem erros de funcionamento, problemas de desempenho e avaliar a jogabilidade.

Puzzle: Ou quebra-cabeças são desafios que não possuem uma solução imediata, exigindo que o jogador pare e reflita em uma forma de resolvê-lo. Eles normalmente são utilizados como mecânicas para adicionar obstáculos (conflitos), os quais tornam o jogo mais interessante e demandam uma ação calculada do jogador.

Roguelike: É um subgênero de jogo RPG, caracterizado pela geração de nível aleatória ou procedural durante a partida, mapa geralmente baseado em ladrilho e morte permanente.

Shortcuts: Ou atalhos, são caminhos que conectam mapas previamente distantes e o jogador pode liberar ou encontrar para se mover de forma mais rápida e eficiente pelo mapa.

Sidescroller: Também conhecido como jogo de rolagem lateral, é um jogo eletrônico no qual a jogabilidade é vista do ângulo da câmera de visão lateral e, à medida que o personagem do jogador se move para a esquerda ou para a direita, a tela rola com ele.

Sprite: É uma imagem bidimensional que representa um personagem, objeto, projétil etc. E que será propriamente implementada no jogo.

Spritesheets: São arquivos que guardam, dentro deles, uma sequência de sprites, os quais, ao serem colocados em uma *timeline* como *keyframes*, geram uma animação de determinado sprite.

SteamPunk: É um gênero de ficção científica fortemente inspirado no séc. XIX, especialmente na tecnologia e estética das máquinas movidas à vapor.

Worldbuilding: É uma expressão do inglês, podendo ser traduzida literalmente para “construção de mundo”, e é nome dado ao processo de criação de um mundo fictício coerente e detalhado em narrativas.

Zine: São revistas independentes e não profissionais com fins lucrativos ou não. O termo zine veio de *fanzine*, aglutinação de *fan magazine* ou “revista de fãs”. Tornando-se popular como um meio de divulgação de trabalhos artísticos, literários, musicais ou de qualquer outro tipo de cultura.

Lista de figuras

Figura 1. Tennis for Two de Willy Higinbotham em um Osciloscópio DuMont Lab Type 304-A. 1958. Disponível em: https://pt.wikipedia.org/wiki/Tennis_for_Two Acesso em: 23 de junho de 2024.....	22
Figura 2. Delineamento do projeto em formato de fluxograma de desenvolvimento.....	27
Figura 3. Raposa de autoria da Isabela.	31
Figura 4. Kingdom: New Lands (2016).....	32
Figura 5. Momodora: Reverie Under the Moonlight (2016)	32
Figura 6. Hollow Knight (2017).....	34
Figura 7. Ori and the Blind Forest (2015).....	35
Figura 8. Tutorial de <i>pixel art</i> por Pedro Medeiros. Disponível em: https://saint11.org/blog/pixel-art-tutorials/ Acesso em: 11 de Setembro de 2024.	38
Figura 9. <i>Moodboard</i> de gêneros e estilos.	39
Figura 10. <i>Moodboard</i> de crianças medievais.....	40
Figura 11. <i>Moodboard</i> de inimigos.....	41
Figura 12. <i>Moodboard</i> de alquimistas.	42
Figura 13. <i>Moodboard</i> de raposas espirituais.	43
Figura 14. <i>Moodboard</i> de ataques e magias.....	44
Figura 15. Spritesheet do jogo Momodora (2016).	45
Figura 16. <i>Moodboard</i> de mecânicas de ataque.	46
Figura 17. <i>Moodboard</i> de <i>level design</i> , mapa, movimentação e mecânicas gerais.	47
Figura 18. <i>Moodboard</i> de telas iniciais de jogos.	48
Figura 19. <i>Moodboard</i> de <i>HUDs</i> de jogos.	48
Figura 20. Fluxo narrativo inicial.	53
Figura 21. “Melius: A Queda de um Reino” 14/06/2023	56
Figura 22. Fluxo narrativo atual.	59
Figura 23. Blocos de colisão	63
Figura 24. Exemplificação da <i>hitbox</i> no jogo.....	64
Figura 25. Blocos de dano.	65
Figura 26. Mecânica de movimentação.....	66
Figura 27. Exemplificação da mecânica de salto.	67
Figura 28. Mecânica de ataque físico.	69
Figura 29. Mecânica da habilidade especial Defesa Mágica.	70
Figura 30. Mecânica da habilidade especial Ataque Mágico.	71
Figura 31. Mecânica da habilidade especial Ataque Carregado.....	72
Figura 32. Revisão da mecânica do ataque físico.....	76

Figura 33. Mecânica da habilidade especial Mordida da Raposa.....	78
Figura 34. Revisão da mecânica da habilidade especial Ataque Carregado.	79
Figura 35. Mecânica da habilidade especial Dash Tático.....	80
Figura 36. Dash Tático para evitar ataques.	81
Figura 37. Mecânica da habilidade especial Empurrão.....	81
Figura 38. Mecânica da habilidade especial Dive.	82
Figura 39. da habilidade especial Erupção.	83
Figura 40. Mecânica de cálculo de caminho do <i>companion</i>	84
Figura 41. Mecânica de distância do <i>companion</i> do jogador.....	84
Figura 42. Mecânica de combate do boss: Investida.....	86
Figura 43. Mecânica de combate do boss: Arremesso de Pote.	87
Figura 44. Mecânica de combate do boss: Arremesso de Veneno.....	88
Figura 45. <i>Concepts</i> do protagonista.	91
Figura 46. Protagonista de Momodora (2016).....	92
Figura 47. Desenvolvimento do <i>asset</i> do protagonista.....	93
Figura 48. <i>Asset</i> final do protagonista.	93
Figura 49. <i>Concepts</i> das magias do protagonista.	94
Figura 50. Desenho final das magias do protagonista.....	94
Figura 51. <i>Concepts</i> da raposa.....	95
Figura 52. Desenvolvimento do <i>asset</i> da raposa.	96
Figura 53. <i>Concepts</i> do alquimista.....	97
Figura 54. Desenvolvimento do alquimista.....	97
Figura 55. <i>Concepts</i> do alquimista boss.	98
Figura 56. Desenvolvimento do alquimista na forma de monstro (<i>boss</i>).	98
Figura 57. <i>Concepts</i> do inimigo rastejante.	99
Figura 58. Desenvolvimento do <i>asset</i> do inimigo rastejante.....	100
Figura 59. Desenvolvimento do <i>asset</i> dos inimigos humanoides.	101
Figura 60. Exemplo de <i>spritesheet</i> de corrida da Momodora (2016).	101
Figura 61. Primeira versão da animação de corrida do protagonista.....	102
Figura 62. Rascunho feito no FlipaClip e QR code para um vídeo contendo as animações.	102
Figura 63. Processo de transferência das animações para o Aseprite.....	103
Figura 64. Animação de teste contendo apenas a silhueta do protagonista.	103
Figura 65. Organização das animações a partir do sistema de <i>loops</i> e <i>tags</i>	104
Figura 66. Detalhe da organização com o sistema de <i>loops</i> e <i>tags</i>	104
Figura 67. <i>Spritesheet</i> da animação de correr do protagonista.	105
Figura 68. <i>Spritesheet</i> da animação de ataque do protagonista.	105

Figura 69. <i>Spritesheet</i> da animação de <i>idle</i> da raposa.	106
Figura 70. <i>Spritesheet</i> da animação de fala do alquimista.	106
Figura 71. <i>Spritesheet</i> da animação de <i>idle</i> da mãe do protagonista.	106
Figura 72. <i>Spritesheet</i> da animação de andar do rastejador.	107
Figura 73. <i>Spritesheet</i> da animação de correr do humanoide.	107
Figura 74. <i>Spritesheet</i> da animação de dano do <i>boss</i>	107
Figura 75. <i>Spritesheet</i> da animação de ataque do <i>boss</i>	108
Figura 76. <i>Spritesheet</i> da animação de investida do <i>boss</i>	108
Figura 77. <i>Spritesheet</i> da animação de correr da raposa.	109
Figura 78. <i>Spritesheet</i> da animação de <i>dash</i> da raposa.	109
Figura 79. Primeira alternativa de layout para o mapa.	111
Figura 80. Segunda alternativa de layout para o mapa.	112
Figura 81. <i>Level design</i> blocagem das salas.	113
Figura 82. <i>Level design</i> inclusão de plataformas.	114
Figura 83. Posicionamento de plataformas.	115
Figura 84. Portões para bloquear a passagem do jogador.	116
Figura 85. <i>Level design</i> detalhamento da cidade subterrânea.	117
Figura 86. Super Mario Bros mundo 1-1 (1985) Disponível em: https://codegolf.stackexchange.com/questions/40008/pixel-art-episode-2-display-the-map-1-1-of-super-mario-bros Acesso em: 21 de Agosto de 2024.	118
Figura 87. Rascunho e planejamento do <i>level</i> da floresta.	119
Figura 88. <i>Level design</i> Floresta.	120
Figura 89. <i>Level design</i> : cidade subterrânea.	121
Figura 90. <i>Level design</i> da primeira versão do jogo.	122
Figura 91. Grid no <i>level design</i>	123
Figura 92. Padronização de cores no <i>level design</i>	124
Figura 93. <i>Concept</i> da Floresta cores e combinações 1.	125
Figura 94. <i>Concept</i> da Floresta cores e combinações 2.	126
Figura 95. <i>Concept</i> da Floresta cores de outono.	126
Figura 96. Rascunho em papel da casa do protagonista: exterior.	127
Figura 97. Rascunho em papel da casa do protagonista: interior.	127
Figura 98. Rascunho em papel da Entrada da caverna.	128
Figura 99. Rascunho em papel da Casa do alquimista exterior.	128
Figura 100. Rascunho em papel da Casa do alquimista interior.	129
Figura 101. <i>Concept</i> Cidade Subterrânea: cores frias.	129
Figura 102. <i>Concept</i> árvore sagrada.	130
Figura 103. <i>Concept</i> fonte da raposa.	131

Figura 104. Comparação das paletas de cores.....	132
Figura 105. <i>Concept</i> expressão visual da doença.....	132
Figura 106. <i>Tileset</i> e construção de mapa Super Mario Bros. Disponível em: https://codegolf.stackexchange.com/questions/40008/pixel-art-episode-2-display-the-map-1-1-of-super-mario-bros Acesso em: 21 de Agosto de 2024.	134
Figura 107. Primeira versão do <i>tilesets</i>	135
Figura 108. Segunda versão do <i>tilesets</i>	135
Figura 109. Primeira versão dos <i>tileset</i> de informação.....	136
Figura 110. Exemplo de cenários exportados do TileD.	137
Figura 111. Testes de plataformas.....	138
Figura 112. Nomenclatura das camadas <i>background faraway</i>	139
Figura 113. Nomenclatura das camadas <i>background</i>	139
Figura 114. Nomenclatura das camadas <i>environment</i>	140
Figura 115. Nomenclatura das camadas <i>characters</i> e <i>foreground</i>	140
Figura 116. Comparação <i>level design</i> com as camadas.....	142
Figura 117. Cenário dividido e organizado no sistema de camadas.....	143
Figura 118. Resultado após a combinação e alinhamento das camadas.....	144
Figura 119. Posicionamento de animações nos cenários.....	145
Figura 120. Assets de árvores e plataformas.....	146
Figura 121. <i>Assets</i> de vegetação e rochas.....	147
Figura 122. <i>Assets</i> para o <i>background</i>	147
Figura 123. <i>Asset</i> casa do protagonista exterior.....	148
Figura 124. <i>Asset</i> mansão do Alquimista.....	148
Figura 125. <i>Assets</i> de árvores infectadas.....	149
Figura 126. <i>Assets</i> de elementos da doença 1.....	149
Figura 127. <i>Assets</i> de elementos da doença 2.....	150
Figura 128. <i>Assets</i> de pilares e plataformas.....	151
Figura 129. <i>Assets</i> de paredes, jaulas e decorações.....	151
Figura 130. <i>Assets</i> para o <i>checkpoint</i> , mecanismo, barreiras e outros.....	152
Figura 131. <i>Asset</i> igreja destruído.....	153
Figura 132. <i>Asset</i> portão de saída da Cidade Subterrânea.....	153
Figura 133. <i>Asset</i> da porta mágica inacessível.....	154
Figura 134. <i>Asset</i> da árvore milenar.....	154
Figura 135. <i>Asset</i> do caixão de vidro.....	155
Figura 136. <i>Asset</i> da fonte da grande raposa.....	155
Figura 137. <i>Spritesheet</i> do mecanismo sendo ativado.....	156
Figura 138. <i>Spritesheet</i> da porta de saída da cidade subterrânea abrindo.....	156

Figura 139. Exemplo de <i>spritesheet</i> das árvores.	156
Figura 140. Exemplo de <i>spritesheet</i> para as cachoeiras.	156
Figura 141. <i>Spritesheet</i> do <i>checkpoint</i>	157
Figura 142. <i>Spritesheet</i> dos itens da poção.	157
Figura 143. Cenário finalizado Floresta na superfície.	158
Figura 144. Cenário finalizado Cidade Subterrânea.	158
Figura 145. Cenário finalizado casa do protagonista exterior.	159
Figura 146. Cenário finalizado casa do protagonista interior.	159
Figura 147. Cenário finalizado entrada e saída da Cidade Subterrânea.	160
Figura 148. Cenário finalizado mecanismos e <i>checkpoints</i>	161
Figura 149. Cenário finalizado locais de coleta dos itens da poção.	162
Figura 150. Cenário finalizado mansão do Alquimista.	163
Figura 151. Cenário finalizado mansão do Alquimista infectada.	163
Figura 152. Expansão do <i>layout</i> do mapa.	164
Figura 153. Detalhamento da concepção de novas áreas.	165
Figura 154. Concepção Ala Médica.	166
Figura 155. <i>Concepts</i> novos checkpoints.	167
Figura 156. <i>Concepts</i> trono real.	167
Figura 157. Adaptação do level ao novo <i>concept</i>	168
Figura 158. Expansão do novo <i>level design</i>	169
Figura 159. Posicionamento dos mecanismos no novo <i>level design</i>	170
Figura 160. Novas barreiras posicionadas no novo <i>level design</i>	170
Figura 161. Novo <i>level design</i> completo.	172
Figura 162. Posicionamentos das habilidades especiais no novo <i>level design</i>	173
Figura 163. Posicionamentos dos elevadores no novo <i>level design</i>	173
Figura 164. Áreas da concepção presentes no novo <i>level design</i>	174
Figura 165. Novo <i>tileset</i> de informação.	175
Figura 166. Um <i>level</i> construído na Unity	176
Figura 167. Novos <i>assets</i> de árvores.	177
Figura 168. Novos <i>assets</i> arbustos, flores, ponte etc.	178
Figura 169. Novos <i>assets</i> pilares, plataformas, correntes etc.	179
Figura 170. <i>Assets</i> portas e jaulas.	179
Figura 171. <i>Assets</i> elevadores.	180
Figura 172. Novo <i>asset</i> checkpoint.	180
Figura 173. Fonte dos 4 Heróis.	181
Figura 174. Novo cenário casa do garoto.	182
Figura 175. Novos cenários da floresta inicial.	182

Figura 176. Cenários: entrada da cidade subterrânea.	183
Figura 177. Capela dos 4 Heróis e arredores.....	183
Figura 178. Ala médica: Segundo elevador.	184
Figura 179. Ala médica.	184
Figura 180. Ala médica: Sala do <i>boss</i>	185
Figura 181. Cenários: Novos <i>checkpoints</i>	185
Figura 182. Cenário: Entrada para o Palácio Real.	186
Figura 183. Teste inicial da tela de <i>gameplay</i>	189
Figura 184. Teste do HUD.	190
Figura 185. Teste do HUD contendo nossos personagens.	191
Figura 186. Primeiro <i>concept</i> da barra de vida do protagonista.	192
Figura 187. Testes de HUD para barra de vida, de magia e itens do protagonista.....	192
Figura 188. Testes de HUD para barra de vida e magia do protagonista.	193
Figura 189. Versão final do HUD para somente a barra de vida do protagonista.	193
Figura 190. Versão final do HUD para a barra de vida e de magia do protagonista.	193
Figura 191. HUD finalizado no jogo.	194
Figura 192. Fluxo de telas.....	195
Figura 193. Protótipo inicial das telas.	196
Figura 194. Testes de tipografia.	197
Figura 195. Tipografia escolhida.	197
Figura 196. Exemplo de caixa de texto do Owlboy (2016).	198
Figura 197. Teste caixa de diálogo.	199
Figura 198. Caixa de diálogo implementada no jogo.....	199
Figura 199. Logo do Melius.....	200
Figura 200. Menu inicial.	201
Figura 201. Tela de créditos.	202
Figura 202. Tela de opções.	202
Figura 203. Primeira versão: <i>gameplay</i> na superfície.	205
Figura 204. Primeira versão: <i>gameplay</i> no subterrâneo.....	205
Figura 205. Primeira versão: <i>gameplay</i> na superfície infectada e QR code para vídeo.....	206
Figura 206. Segunda versão: <i>gameplay</i> floresta subterrânea e floresta inicial.	207
Figura 207 Segunda versão: <i>gameplay</i> cidade subterrâneas.....	207
Figura 208. Segunda versão: <i>gameplay</i> luta com o <i>boss</i> alquimista.	208

Sumário

1. Introdução	21
2. Delineamento	26
3. Definição do jogo e High Concept	30
4. Pesquisa de referências de estilo do jogo	37
5. História: Narrativa e Enredo do Jogo	50
5.1 Processo de desenvolvimento do Enredo	50
5.1.1 Enredo Selecionado:	51
5.2 Processo de desenvolvimento da Narrativa	52
5.2.1 Narrativa Selecionada:	53
5.3 Modificações propostas no Primeiro Ciclo de Revisão	54
5.3.1 Acontecimentos anteriores ao jogo	57
5.3.2 Enredo Mais Atual:	58
5.3.3 Narrativa Mais Atual:	59
5.4 Considerações sobre a Narrativa	60
6. Regras e Mecânicas	61
6.1 Regras e Mecânicas Gerais e de Mapas	62
6.2 Regras e Mecânicas do Personagem Jogável	65
6.2.1 Movimentação	66
6.2.2 Regras Gerais	67
6.2.3 Combate	68
6.3 Regras e Mecânicas de <i>Companion</i> e NPCs	72
6.3.1 Companions	72
6.3.2 NPCs	73
6.4 Regras e Mecânicas de Inimigos e <i>Bosses</i>	73
6.4.1 Inimigos	73
6.4.2 Bosses	74

6.5	Modificações propostas as Regras e Mecânicas no primeiro ciclo de revisão.	74
6.5.1	Revisões gerais e de mapa.....	74
6.5.2	Revisões ao personagem jogável	75
6.5.3	Revisões e novas habilidades especiais	77
6.5.4	Revisões ao <i>companion</i>	83
6.5.5	Revisão Mecânica de inimigos e <i>bosses</i>	85
6.6	Considerações sobre as regras e mecânicas	88
7.	Personagens	90
7.1	Design de Personagem.....	90
7.1.1	Protagonista.....	90
7.1.2	Raposa	94
7.1.3	Alquimista	96
7.1.4	Inimigos	98
7.2	Animação dos Personagens	101
7.2.1	Protagonista.....	102
7.2.1	Animação (NPCs, Inimigos e <i>Boss</i>)	106
7.3	Revisão e Refinamento da Animação	108
7.4	Considerações sobre os Personagens	109
8.	Mapa e Cenários	110
8.1	<i>Level Design</i> do Mapa	110
8.1.1	Produção	111
8.1.2	Alternativa Seleccionada	119
8.2	Concepção dos Cenários	124
8.2.1	Floresta.....	125
8.2.2	Cidade subterrânea	129
8.2.3	Desenvolvimento da expressão visual da Doença.	131
8.3	Uso de Tiles para a construção do Mapa.....	133
8.4	Uso de Camadas para a construção dos Cenários.....	138

8.4.1 Nomenclatura e função das camadas	139
8.4.2 Exportar mapas finalizados	141
8.5 <i>Assets</i> e Finalização do Cenário	145
8.5.1 Criação de <i>Assets</i> do cenário	146
8.5.1 Animação de <i>Assets</i>	156
8.5.2 Detalhamento e Finalização dos Cenários	157
8.6 Modificações propostas aos mapas e cenários	164
8.6.1 Desenvolvimento de Concepts adicionais	164
8.6.2 Modificações propostas ao Level Design	168
8.6.3 Mudanças do sistema de tiles	174
8.6.4 Modificações propostas para os Cenários e novos <i>assets</i>	176
8.7 Considerações sobre Mapas e Cenários	186
9. Telas e Heads-Up Display	188
9.1 Elementos do HUD	189
9.2 Telas e Menus	194
9.3 Considerações sobre o HUD e as Telas	202
10. O Jogo Melius	204
11. Conclusão	209
Referências Bibliográficas	213
Apêndice A: Lista de <i>spritesheets</i> dos personagens	214
Apêndice B: Lista de <i>spritesheet</i> de <i>assets</i> dos cenários	221
Apêndice C: Cenários finalizados	227

1. Introdução

Os jogos eletrônicos, ou *videogames*, são uma forma de entretenimento interativo na qual um ou mais jogadores usam um dispositivo de entrada, como um teclado ou controle, para realizar uma série de ações que são refletidas na tela. O pesquisador dinamarquês Jesper Juul lista seis aspectos principais para entender o que torna uma atividade um jogo.

Para Juul (2003) todo jogo é estruturado por um conjunto de regras fixas que definem o que é possível e o que não é dentro do jogo. Um jogo tem um resultado variável e quantificável. Isso significa que os jogadores podem obter diferentes resultados baseados em suas ações e decisões dentro do jogo, e esses resultados podem ser quantificados, como, por exemplo, por meio de sistema de pontuações que define o vencedor ao final do jogo. A valorização do resultado de um jogo pelos jogadores implica em o resultado ter significado ou importância para os participantes, ou mesmo em se obter resultados melhores do que outros jogadores. Assim, o jogo também exige a interação e esforço dos jogadores para transformar esses resultados. Este esforço pode envolver habilidades, estratégias, ou trabalho árduo para alcançar o objetivo do jogo. Juul observa também que os jogadores frequentemente se sentem emocionalmente envolvidos com o resultado do jogo. Esse apego nem sempre está ligado ao esforço que o jogador coloca no jogo, por exemplo, o jogador pode se sentir feliz por ganhar um jogo de pura sorte. Esse apego pode aumentar o engajamento e a motivação dos jogadores, tornando o resultado do jogo pessoalmente significativo. Finalmente, Juul aponta que as consequências de um jogo podem ser negociáveis. Isso significa que, em muitos casos, as consequências podem ser ajustadas ou adaptadas com base nas escolhas dos jogadores ou nas regras acordadas, permitindo certa flexibilidade e adaptação dentro do jogo (Juul, 2003).

É importante ressaltar que, segundo o próprio Juul, estas não são as únicas características de um jogo, mas sim de uma descrição clássica com bordas lúcidas, e jogos estão evoluindo para além destas características. Já os autores Katie Salen e Eric Zimmerman (2012, pp.96) em seu livro definem um *videogame* como: “sistema no qual jogadores se envolvem em um conflito artificial, definido por regras, que resultam em um resultado quantificável”.

Assim como outros tipos de jogos, como os de cartas ou tabuleiro, os *videogames* possuem regras e condições de vitória e derrota que proporcionam uma experiência divertida para o jogador. O primeiro *videogame*, chamado Tennis Programing, foi criado em 1958 por

Willy Higinbotham e era simplesmente um software que simulava uma partida de tênis usando um osciloscópio (Figura 1). Com o passar do tempo, mais jogos foram lançados e essa forma de entretenimento se tornou cada vez mais popular. Em 1972, o Odyssey, o primeiro console, foi lançado pela empresa Magnavox, e no mesmo ano, a Atari lançou Pong, que se tornou um grande sucesso (Donovan, 2010; Rabin, 2005).



Figura 1. Tennis for Two de Willy Higinbotham em um Osciloscópio DuMont Lab Type 304-A. 1958. Disponível em: https://pt.wikipedia.org/wiki/Tennis_for_Two Acesso em: 23 de junho de 2024.

Até então, os *videogames* eram simples e bastante semelhantes entre si, mas em 1987 a Nintendo lançou o Family Computer, ou Famicom, que veio com jogos do Game Designer Shigeru Miyamoto, como Donkey Kong e Mario, que foram grandes sucessos de vendas e revolucionaram a indústria dos jogos 2D e o *game design* (Donovan, 2010; Rabin, 2005; Kent, 2001).

Devido às limitações da tecnologia, por muito tempo todos os jogos eram em 2D, ou seja, jogos nos quais a movimentação do jogador e da câmera era limitada a apenas dois eixos, geralmente sendo compostos por imagens bidimensionais. Porém, em 1992, o primeiro jogo na qual o jogador é capaz de se movimentar por ambientes tridimensionais, Wolfenstein 3D, foi lançado, desenvolvido pela Id Software e publicado pela Apogee Software. Com o

passar dos anos os jogos 3D foram se tornando muito mais comuns, o que lentamente fez com que jogos 2D deixassem de ser o padrão e se transformassem em um gênero dentro da indústria, assim como os gêneros de ação, aventura, plataforma¹ etc. (Donovan, 2010; Rabin, 2005; Kent, 2001).

Dentro do gênero 2D surgiria o subgênero Metroidvania, cujo nome veio dos clássicos Super Metroid, (Nintendo, 1994), e Castlevania: Symphony of the Night (Konami, 1997). Na época de seus lançamentos, esses jogos eram classificados como apenas jogos de ação e aventura mas, posteriormente, definiram os elementos fundamentais do *gameplay*² de um Metroidvania, como: mapas extensos e interconectados, geralmente de plataformas, com foco na exploração não linear e a forte presença de *backtracking*, que são momentos nos quais o jogador precisa voltar a áreas já visitadas, agora com novidades ou caminhos liberados para alcançar novos objetivos, avançar na narrativa ou resolver *puzzles*, que são desafios nos quais o jogador precisa raciocinar e usar seus conhecimentos das mecânicas³ do jogo para cumprir um objetivo.

Para criar um jogo, é necessário conhecimento em várias áreas. O Designer é responsável por desenvolver e conceber os elementos de um jogo. Isso inclui a criação de mecânicas de jogabilidade, narrativa, personagens e ambientes, garantindo uma experiência envolvente para os jogadores. Esse profissional é conhecido como Game Designer.

No curso de Design da Universidade de Brasília, a disciplina de Design de Jogos (DJ) permite que os alunos desenvolvam um produto, especificamente um jogo eletrônico. Esta disciplina é ofertada em conjunto com disciplinas de Computação e da Música, isso é feito com o objetivo de formar grupos interdisciplinares, oferecendo aos estudantes a oportunidade de trabalhar com colegas de diferentes áreas do conhecimento e aprender na prática como criar um jogo, influenciando o tipo de jogo, seu tema e todos os seus elementos. Porém, devido ao curto prazo somado a complexidade processo de desenvolvimento, é difícil que um grupo tenha um

¹ **Plataforma**, também conhecido como *jump 'n' run*, é um gênero de jogos eletrônicos de ação em que o objetivo principal é mover o personagem do jogador entre pontos em um ambiente. Os jogos de plataforma são caracterizados por níveis que consistem em terrenos irregulares e plataformas suspensas de altura variável que exigem pular e escalar para atravessar.

² **Gameplay** trata-se da experiência do jogador ao jogar e interagir com todas as regras e mecânicas do jogo.

³ **Mecânica** se refere ao conjunto de definições e sistemas que definem como funciona o jogo e como se dá a interação do jogador com esse jogo.

jogo completo ao final da disciplina. Muitas vezes, o que é desenvolvido consiste em uma versão de demonstração, que engloba as principais ideias do grupo.

Aos estudantes do Design, também há a oportunidade de experimentar com aspectos não tão abordados no curso, como *design* de personagens e cenários. O gênero Metroidvania, por sua vez, oferece oportunidades adicionais de aprendizagem, pois requer um planejamento extra em *level design*⁴ e mecânicas para incorporar elementos de *backtracking* e *puzzles*. Além disso, essa complexidade de reutilização dos cenários precisa de um significado atrelado ao enredo e narrativa, por isso demanda sua elaboração mais profunda, inclusive com a definição de elementos que se integrem com o *level design* e cenários.

Nosso principal objetivo é criar um jogo com inspirações no gênero Metroidvania, divertido e completo, isto é, um jogo no qual o jogador tenha uma experiência com início, meio e fim, ao mesmo tempo em que possamos experimentar livremente e aprender sobre o desenvolvimento de jogos durante todo processo. Para tanto, temos que aprender como se dá o processo de criação de um jogo desde a etapa mais inicial até a entrega de um produto finalizado, o qual qualquer pessoa pode experimentar e aproveitar a experiência de jogar. Assim, também temos que definir as mecânicas, desenvolver o enredo, planejar o estilo artístico de forma que todos os membros possam contribuir para criar uma arte homogênea, além de fazer o *level design*, criar os cenários, personagens, telas, entre outros, em coordenação com uma equipe multidisciplinar, muito semelhante a uma equipe real de desenvolvimento de jogos.

O desenvolvimento do nosso jogo começou bem antes da disciplina de Trabalho de Conclusão de Curso (TCC). A ideia surgiu no contexto da disciplina de Design de Jogos. Nosso grupo foi composto por quatro designers, dois programadores e um músico, que trabalharam juntos até o final do semestre para iniciar a concepção do jogo descrito neste relatório. Após o fim da disciplina, o nosso jogo passou por ciclos de revisão durante mais três semestres, a fim de continuarmos aprendendo sobre o desenvolvimento de jogos, testarmos novas ferramentas, e expandirmos o projeto criado na disciplina. Alguns membros deixaram o grupo e outros se ausentaram por alguns semestres, como será explicado na próxima seção.

⁴ *Level design* é nome dado à uma etapa do desenvolvimento de jogos, no qual se criam fases, mapas, missões e objetivos para o jogador. É o momento onde é criado o ambiente onde se passará o jogo, com todos seus obstáculos, desafios, objetivos e formas do jogador passar por ele. Também é o nome dado ao resultado final de toda essa tomada de decisões.

O desenvolvimento de um jogo eletrônico envolve diversos campos do conhecimento, necessitando de um trabalho complexo e multidisciplinar para se atingir o resultado desejado. No campo do Design, utilizamos diversos conhecimentos sobre desenvolvimento e organização projetual, com o objetivo de garantir um processo produtivo e harmonioso. Além disso, também aplicamos processos de desenvolvimento visual do Design, adaptados para auxiliar na criação do jogo.

Assim, o presente relatório descreve um trabalho prático desenvolvido ao longo de várias disciplinas e, em parte, iniciado antes do contexto do TCC. Por esse motivo, ele não apresenta um capítulo reflexivo e teórico a partir de revisão de literatura. O foco da argumentação é voltado para a descrição da *práxis* projetual do jogo. Optamos também por utilizar os termos recorrentes do campo dos jogos eletrônicos ao longo do texto, mesmo que muitas vezes sejam expressos em inglês. Essa decisão foi tomada devido à ampla disseminação e utilização dos conceitos pelo mercado, desenvolvedores, comunicadores e jogadores.

O relato possui uma breve descrição do delineamento do projeto, que então é aprofundado, com a descrição de como foram feitas as etapas de *high concept*, de pesquisa, desenvolvimento do enredo e narrativa, concepção de personagens e cenários, criação de *assets*⁵ e telas. Por fim, apresentamos o produto final e a conclusão do relatório.

⁵ *Assets* são os elementos que constituem um jogo, como arte, ilustrações, animações, efeitos sonoros, música, scripts, texto etc.

2. Delineamento

Neste capítulo, vamos abordar o modelo de produção que utilizamos para o desenvolvimento do jogo. No entanto, é importante ressaltar que o desenvolvimento de jogos requer pessoas com expertise em diferentes áreas do conhecimento, e que todas elas são igualmente importantes para se atingir o resultado desejado para o projeto. Neste sentido, para este relatório, detalhamos principalmente como se deu o processo de Design, mais especificamente no processo de Game Design, que se trata de um campo dentro de Design que engloba diversas áreas como, por exemplo, planejamento de projeto, interface de usuário, animação e produção de artes, entre outros.

O desenvolvimento do jogo pode ser dividido em três principais etapas: a pré-produção, produção e pós-produção. Esse processo não ocorre de maneira linear, diversas etapas acontecem de forma simultânea. Então, descreveremos na presente seção de maneira macro todas essas etapas e o modo com que elas se relacionam entre si, podendo haver momentos em que uma fase ocorre simultaneamente à outra, como pode ser visto na Figura 2.

O nível de detalhamento no esquema visual está heterogêneo, descrevendo mais a fundo as etapas de Design, visto que este é o foco deste relatório. No entanto, é importante ressaltar que todos os membros do grupo também participaram do início ao fim do processo de produção do jogo e de tomada de decisões, tanto com suas ideias e opiniões, quanto com o próprio trabalho produzido em suas respectivas áreas influenciando no jogo e no processo de Game Design.

A pré-produção é a primeira etapa, na qual a equipe decide as principais características de como será o jogo, o *high concept*. Todos os membros do grupo contribuem, trazendo ideias, opiniões, exemplos de outros jogos e debatendo-as durante as reuniões. Além disso, é a etapa na qual ocorrem pesquisas, coleta e organização de referências que auxiliam nessa tomada de decisões e guiam o desenvolvimento do jogo.

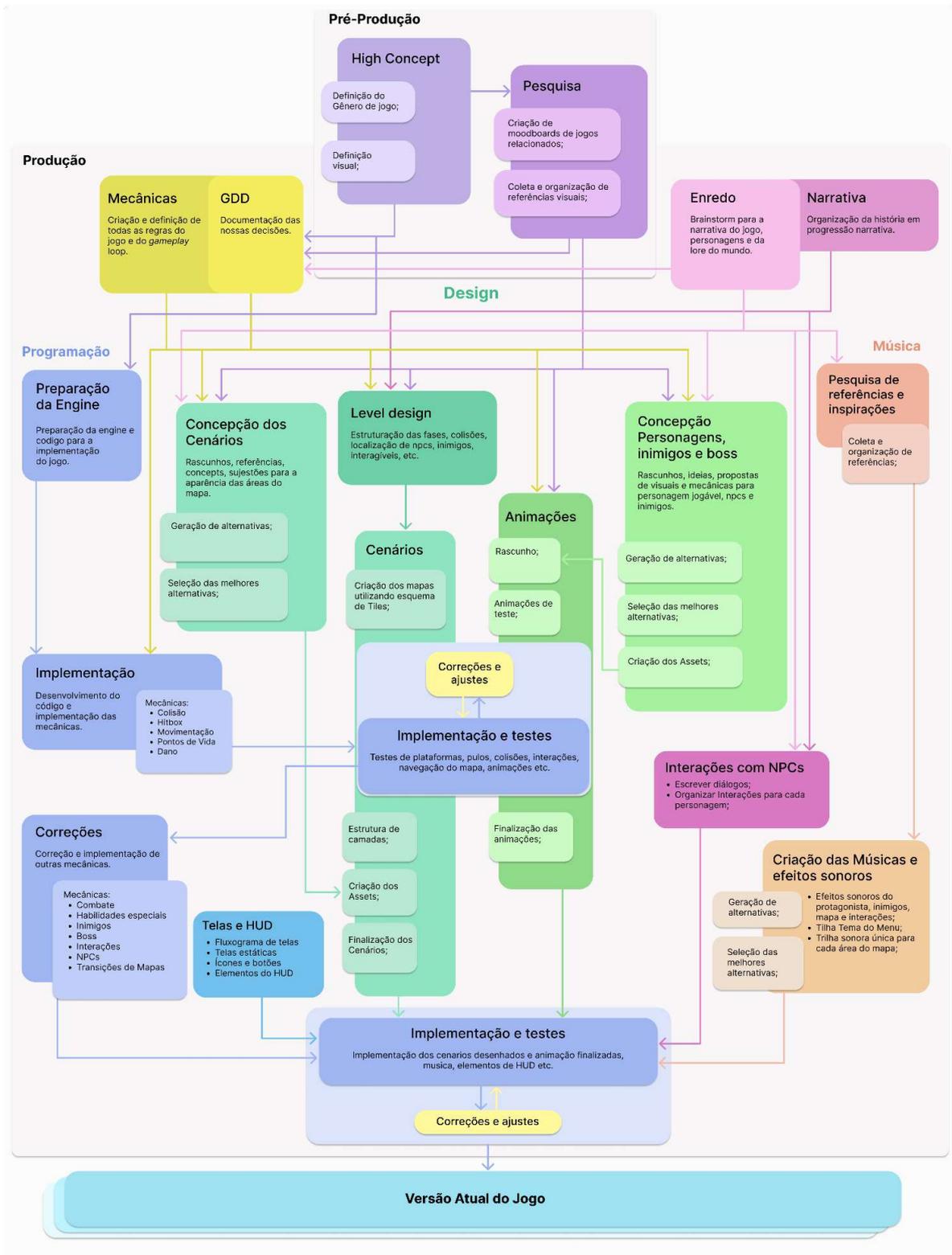


Figura 2. Delineamento do projeto em formato de fluxograma de desenvolvimento.

A produção é a segunda etapa. É nela que se inicia a criação e junção dos elementos que compõem o jogo, como o desenvolvimento da arte de cenários e personagens, animações, a narrativa e enredo, mecânicas, *level design*, elementos do *heads-up display* (HUD⁶) e telas, o desenvolvimento da *engine*, a criação da música etc. Essa etapa se inicia antes mesmo da anterior ser totalmente finalizada, ocorrendo de forma simultânea por um tempo. Por exemplo, ao mesmo tempo que a equipe desenvolvia e refinava a proposta do jogo, os membros já contribuía para a concepção de alguns desses elementos, como a narrativa e o *level design*.

Para o nosso jogo adotamos um sistema no qual a etapa de produção se repete; no qual em mais de uma ocasião, finalizamos uma versão atual jogável e então voltamos uma etapa anterior, como a de pesquisa, repetindo o fluxo em quase toda sua totalidade. Estes são os ciclos de revisão do jogo, que consistem na realização de testes, mudanças, expansão, refinamentos e polimentos do jogo, que costumam a ser realizados, caso seja constatado que essas mudanças podem trazer melhorias significativas para a experiência de *gameplay*.

A pós-produção é a última etapa, tratando da divulgação e lançamento do jogo. Ela não está presente na imagem, pois, até o momento da escrita deste relatório, ela ainda não foi realizada. Nessa fase, são criadas artes de divulgação e realizado o *marketing* para que o jogo chegue ao maior número de pessoas possível. Essa etapa costuma se iniciar antes do fim da produção, para maximizar as vendas assim que o jogo é finalizado e lançado.

Ao longo do relatório, descrevemos mais a fundo cada uma das fases envolvidas dentro de cada uma das etapas de desenvolvimento, detalhando como ocorreram e sua importância para a criação do nosso jogo.

A produção do jogo se estendeu por vários semestres e disciplinas, mas teve início durante a disciplina de Design de Jogos (DJ). Nela há uma breve introdução ao mundo e desenvolvimento dos jogos no início das aulas, mas rapidamente as turmas são divididas em grupos que, unidos por uma ideia em comum, têm como objetivo desenvolver um jogo até o fim da disciplina. O grupo de desenvolvimento do nosso jogo se formou nesse contexto, tendo como participantes quatro designers: Isabela Leal, Ralison Marçal, Raimundo Marques e Luísa Rocha; dois programadores: Gabriel Fontenelle e Moisés Felipe; e um músico: Maurício Ribeiro.

⁶ **HUD** é um conjunto de elementos gráficos projetados na tela para transmitir informações ao jogador durante a execução do jogo.

No fim da disciplina de DJ, entregamos a primeira versão do jogo, que foi apresentada junto de todos os jogos produzidos pelos alunos naquele semestre para uma plateia composta de outros alunos, convidados e professores. Após todas as apresentações, ocorreram uma série de premiações, sendo três relacionadas a áreas do desenvolvimento (arte/design, programação/implementação e música) e uma que engloba todas as áreas e avalia o produto final (melhor jogo). Essas premiações são um meio de incentivar os alunos da disciplina a produzirem o melhor jogo que conseguirem naquele período, nosso jogo ficou em primeiro lugar em três das quatro premiações: melhor arte, melhor implementação e melhor jogo.

Com o término da disciplina, o jogo foi levado para uma amostra de jogos universitários na Campus Party Brasília de 2023 (CPBSB5), graças a uma parceria com a Associação dos Desenvolvedores de Jogos Eletrônicos do Distrito Federal (ABRING). Lá, pudemos mostrar o protótipo produzido em DJ para várias pessoas, que tiveram a chance de jogar e compartilhar suas opiniões. Nesse momento, pudemos notar diversos problemas e equívocos que poderiam ser repensados e melhorados no jogo.

Após a mostra de jogos na Campus Party, com exceção de dois membros, o restante do grupo expressou o desejo de continuar a desenvolver o jogo e até mesmo expandi-lo, dando início ao primeiro ciclo de revisão. Nos dois semestres seguintes, nosso programador iniciou o processo de preparação de uma nova *engine* para receber o nosso jogo, enquanto os designers ficaram responsáveis por organizar e revisar os materiais já produzidos, e em seguida recomeçar os trabalhos de pré-produção e produção.

Complementarmente, durante o processo de expansão da história, vimos a oportunidade, em acordo com os professores da disciplina de Estudos Avançados em Design: Ateliê de Produção de Quadrinhos, de desenvolver uma Zine⁷ com a temática do nosso jogo, com o apoio da Mônica Beatriz como membro do grupo na disciplina.

No semestre 02/2023, o grupo voltou à matéria de DJ e continuou desenvolvendo o jogo. Nesse período, focamos na etapa de produção, revisando e expandindo mecânicas, narrativa, *level design*, criação e animação de personagens e inimigos, correção de animações, expansão do cenário etc. Ao final da disciplina, apresentamos a versão mais atual do jogo.

⁷ **Zine** são revistas independentes e não profissionais com fins lucrativos ou não. O termo zine veio de *fanzine*, aglutinação de *fan magazine* ou “revista de fãs”. Tornando-se popular como um meio de divulgação de trabalhos artísticos, literários, musicais ou de qualquer outro tipo de cultura.

3. Definição do jogo e High Concept

Neste capítulo, falaremos sobre o *high concept* do projeto, uma etapa essencial para a produção de um jogo, por nos permitir consolidar uma breve e sucinta estrutura conceitual, bem como convergir e materializar as ideias que tomam forma durante o processo criativo. Durante sua elaboração, delimitamos um norte para o desenvolvimento do projeto, definindo seus principais elementos e características. A partir desse norte, estabelecemos o eixo sobre o qual tomamos todas as decisões relativas ao projeto. Essa fase é imprescindível, pois nela definimos as principais características do jogo, como, por exemplo, o gênero e o estilo visual a serem seguidos.

A elaboração do *High Concept* partiu do acordo coletivo entre os membros da equipe, a princípio, feito através de reuniões nas quais as ideias dos integrantes do grupo foram dispostas em sessões de *brainstorming*⁸. O objetivo era formar uma ideia inicial de como seria o jogo com base em uma visão compartilhada, considerando também as possibilidades e limitações relativas ao tempo e aos recursos que tínhamos à disposição. Nesse momento, cada membro do grupo trouxe ideias e proposições, submetendo-os a discussões para determinar o que seria mantido e o que seria descartado.

De imediato, concluímos que criaríamos um jogo no formato 2D, devido às limitações impostas pela circunstância de produção inicial (contexto acadêmico), que dificultariam a criação de um projeto em outro formato. Essa escolha também foi motivada pelo fato de que, durante as discussões iniciais a respeito do projeto, uma integrante do grupo de *design* trouxe uma *pixel art*⁹ autoral de uma pequena raposa (Figura 3). O grupo imediatamente se sentiu inclinado a transformá-la em uma personagem relevante no jogo. Sua presença tornou-se um elemento fundamental não apenas para o desenvolvimento do estilo artístico, mas também para a elaboração do enredo e para a disposição das mecânicas do jogo.

⁸ **Brainstorming** é uma técnica que, por meio do compartilhamento espontâneo de ideias, busca encontrar a solução para um problema ou gerar possibilidades criativas.

⁹ **Pixel art** é um estilo de arte digital, no qual as imagens são criadas usando quadrados chamados de pixels, de forma que os mesmos fiquem claramente visíveis. É um estilo que remete a estética originária dos computadores de 8 e 16 *bits*, que tinham uma resolução (número de *pixels* que compõem a tela) limitada, e que portanto só podiam gerar imagens onde os mesmos ficavam visíveis.

Após definirmos o estilo visual, buscamos referências que poderíamos usar para definir as ideias iniciais do mundo que queríamos criar.



Figura 3. Raposa de autoria da Isabela.

A arte da raposa também se tornou um dos motivacionais para a escolha do *pixel art* como estilo artístico do jogo. A partir dessa escolha começamos a busca por referências e inspirações artísticas que se encaixam nesse estilo, que é mais simplificado, sem a presença de qualquer *lineart*¹⁰ ou contorno. Após um breve período de pesquisa, foram apresentadas e discutidas pelos membros do grupo algumas possíveis inspirações. Como resultado, decidimos por utilizar o jogo Kingdom: New Lands (Noio, 2016) como a principal inspiração visual para a criação dos cenários, pois apresenta cenários deslumbrantes, mas ao mesmo tempo com um desenho mais simples e sem a presença de contorno, como pode ser visto na Figura 4. A segunda grande inspiração que serviu como referência para a criação das personagens foi o jogo Momodora: Reverie Under the Moonlight (Bombservice Studio, 2016), caracterizado pela simplicidade de estilo, e assim como o Kingdom: New Lands produzido sem *lineart* ou contorno, possui personagens carismáticos e cativantes (Figura 5), e ao mesmo tempo possui uma animação fluida e funcional.

¹⁰ *Lineart* é o nome dado para todas as linhas (internas ou externas) utilizadas para dar forma ao desenho.



Figura 4. Kingdom: New Lands (2016)



Figura 5. Momodora: Reverie Under the Moonlight (2016)

Após discussões entre os integrantes do grupo, optamos por construir uma estética *low fantasy*¹¹ situada em um mundo Medieval/SteamPunk¹², à exemplo do título Arcane (Fortiche, 2021). No entanto, diante de limites impostos pelos prazos de entrega do produto, decidimos seguir um caminho mais simples, mantendo poucos elementos do SteamPunk e dando mais ênfase no estilo medieval, focando em criar uma ambientação completamente devastada, obscura, pobre, caótica e com poucos sinais de civilização.

Optamos por um título simples e evocativo, escolhendo *Melius* que vem do latim para “melhorar”, não só representando nossa constante aspiração pelo desenvolvimento, mas também entrelaçando com a história e narrativa que nesse momento já estávamos começando a desenvolver e delimitar os principais elementos a serem abordados. No que diz respeito a história, delineamos o enredo em torno da jornada de uma criança, que moraria sozinha com sua mãe. No início do jogo, com o adoecimento da mãe, o protagonista seria forçado a sair de casa e explorar esse mundo perigoso sozinho. Nessa jornada ele encontraria a pequena raposa, um ser mágico que salva o protagonista de uma morte quase certa, e a partir disso os dois desenvolvem uma grande amizade, passando a trabalhar juntos para sobreviver a todas as adversidades que encontrarem e ir progressivamente desbloqueando novas habilidades mágicas da pequena raposa ao longo da exploração. A trilha sonora, concebida como parte integral da atmosfera, foi planejada para acentuar a estética Medieval/SteamPunk e ao mesmo tempo trazer um clima melancólico e perigoso para intensificar a experiência emocional dos jogadores.

Ainda nesta fase, junto aos elementos centrais do projeto, pensamos em algumas ideias de mecânicas e *gameplay* que futuramente foram refinadas. Nesse momento, determinamos também qual seria o gênero do jogo: *Metroidvania*. Essa escolha foi motivada pelo desejo da equipe em enfatizar a exploração e a não-linearidade, com base nos gostos e preferências da equipe quanto ao estilo de jogo a ser seguido.

No que diz respeito às mecânicas e ao *gameplay* do jogo, utilizamos como referência o título *Hollow Knight* (Team Cherry, 2017) (Figura 6). Este jogo se destaca como um exemplo de simplicidade nas mecânicas de movimentação, combate e habilidades especiais, acompanhada da eficácia em proporcionar uma experiência de *gameplay* fluida para a exploração, intuitiva nos controles e imersiva para o jogador, aspecto crucial de inspiração para

¹¹ *Low fantasy* é um subgênero de fantasia um pouco mais preso à realidade, onde o mundo e os personagens seguem uma linha mais realista e os elementos mágicos aparecem de forma mais moderada.

¹² **SteamPunk** é um gênero de ficção científica fortemente inspirado no séc. XIX, especialmente na tecnologia e estética das máquinas movidas à vapor.

o desenvolvimento do nosso próprio projeto. Além das mecânicas, buscamos, em Hollow Knight, e também de Ori and the Blind Forest (Moon Studios, 2015) (Figura 7), inspiração para guiar o Design do cenário e dos níveis do nosso jogo. A complexidade dos ambientes em Hollow Knight e a atmosfera envolvente de Ori foram elementos que consideramos valiosos para incorporar em nossa abordagem de *level design*, criando ambientes desafiadores que contam uma história visualmente cativante.



Figura 6. Hollow Knight (2017)



Figura 7. Ori and the Blind Forest (2015)

Essas escolhas não apenas refletem o reconhecimento da qualidade de jogos existentes, mas também destacam nossa habilidade de adaptar e reinterpretar esses elementos para conceber algo novo e cativante em nosso próprio contexto de desenvolvimento. Ao consolidarmos as definições finais para o jogo, buscamos uma síntese das escolhas feitas durante o processo de concepção e planejamento. Optamos por focar em plataformas acessíveis, como PC, para o lançamento, mantendo a possibilidade de expansão para outras plataformas posteriormente.

Portanto, a partir desta etapa, foram definidas as seguintes características:

- Nome do jogo: Melius;
- Gêneros: Metroidvania voltado para exploração e combate;
- Câmera: Visão lateral 2D;
- Estilo visual: *pixel art*;
- Estética: Medieval/ SteamPunk.

Em suma, essas definições finais refletiram não apenas nossas inspirações e preferências como equipe, mas também uma compreensão pragmática da viabilidade e dos prazos estipulados. Com Melius, iniciamos a fase de desenvolvimento com uma visão clara e

ambiciosa do que aspiramos criar. A partir disso, fomos para a próxima etapa do processo, com o início de uma pesquisa mais aprofundada de referências tanto estéticas quanto de *gameplay*.

4. Pesquisa de referências de estilo do jogo

Neste capítulo, abordaremos as múltiplas etapas envolvidas na pesquisa para a concretização do projeto. Este processo engloba a coleta e a organização de uma gama de referências e fontes de inspiração organizados em formato de *moodboard*¹³, abrangendo não apenas elementos gerais como referências de gênero, visuais e auditivas, mas também elementos específicos como estruturas de telas, HUD, mecânicas de jogo entre outros.

O desenvolvimento desta fase é considerado relevante, uma vez que a partir dela é possível delinear os locais e fontes das quais nos baseamos para reunir, discutir e ampliar a biblioteca de referências selecionadas que auxiliarão no desenvolvimento visual do jogo. O pesquisador Jesse Schell define a estética do jogo como tudo aquilo em que o jogador vê, ouve, cheira e sente. E a relação mais direta entre o jogo e o jogador, e a maneira como é desenvolvida pode passar ao o jogador sensações e emoções, o imergindo no jogo (Schell, 2014)

Além disso, ela nos permite elucidar o processo de organização e categorização adotado, que viabilizou à equipe de produção uma biblioteca ampla, organizada, categorizada, de fácil acesso e navegação, de forma fluida e eficiente por meio de *moodboards*, definindo as nuances e diretrizes estabelecidas para o projeto.

Durante esta etapa inicial, um dos primeiros passos foi a elaboração da pesquisa sobre estilos e tamanhos de *pixel art* (Figura 8). O aprofundamento nesse aspecto tornou a coleta de referências mais precisa e a categorização de imagens mais sucinta. Como já discutido na etapa de *high concept*, desde o início tínhamos claro que esse seria o estilo visual adotado para o nosso jogo.

¹³ **Moodboard** é um arranjo de referências visuais, auditivas e textuais que auxiliam na criação de um projeto, permitindo a visualização de um estilo particular ou um conceito, e servindo de base para a definição do estilo do projeto.



Figura 8. Tutorial de *pixel art* por Pedro Medeiros. Disponível em: <https://saint11.org/blog/pixel-art-tutorials/> Acesso em: 11 de Setembro de 2024.

A coleta de referências se deu a partir de mecanismos de pesquisa como o Google, bancos de imagens e em redes sociais voltadas para o compartilhamento de imagens como o Pinterest. A busca foi feita a partir dos nomes de jogos já conhecidos previamente pelo grupo ou a partir do uso de palavras chaves, descrevendo estilos visuais como, por exemplo, “personagem medieval”, “raposa espiritual”, a partir de gêneros de jogos escolhidos como inspiração ou até pesquisas mais genéricas como “jogos 2D”, “jogos *pixel art*”, “*spritesheets*”.

Em seguida, elaboramos um *moodboard* (Figura 9) abrangente, explorando gêneros e estilos de jogos semelhantes aos que buscávamos alcançar como resultado para nosso projeto, como Momodora (2016) e Blasphemous (The Game Kitchen, 2019). Também foram incluídos jogos com nível artístico que gostaríamos de alcançar como Kingdom (2015), Dead Cells (Motion Twin, 2018) e Celeste (Maddy Makes Games, 2018). Foram consideradas até referências artísticas mais indiretas, nas quais o estilo artístico acaba não se encaixando completamente com o nosso objetivo, mas que expressam alguma dimensão do que nós tentamos atingir, como o *mood*¹⁴ de obras como Arcane (2021) e Sword & Sworcery (Superbrothers e Capybara Games, 2011). Utilizamos a ferramenta *Figma*¹⁵ para criação e organização de todos os *moodboards* que se alinhassem com a visão que tínhamos para nosso jogo.

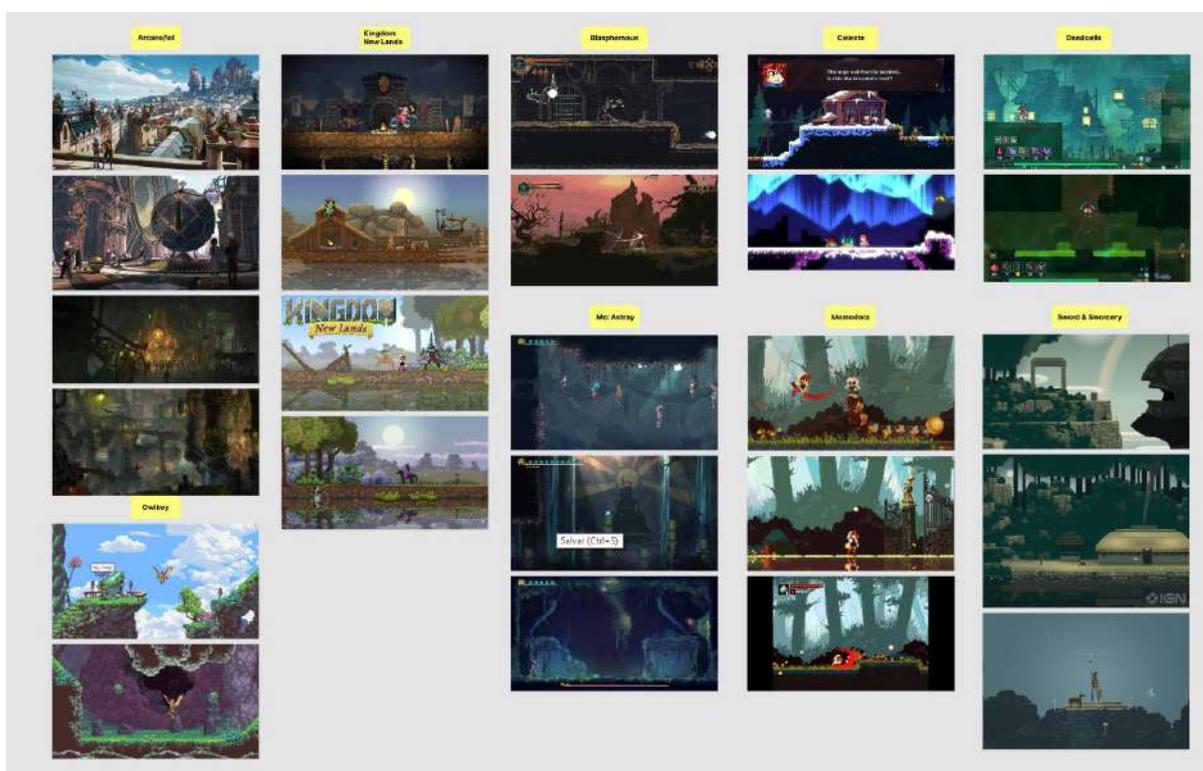


Figura 9. *Moodboard* de gêneros e estilos.

¹⁴ **Mood** é uma palavra de origem inglesa, que traduzida ao português significa “humor” ou “sentimento”. Nesse caso é utilizado com uma conotação artística, onde a obra evoca no observado um sentimento específico a partir das cores e tom da peça.

¹⁵ **Figma** é uma ferramenta de edição em vetor e prototipagem, que permite o trabalho e acesso simultâneo de vários usuários, geralmente usada para a criação de projetos de *design*.

Além disso, ampliamos nossa pesquisa para a criação dos personagens, elaborando vários *mini-moodboards* que abrangiam diferentes aspectos, como *sprites*¹⁶, vestimentas e caracterização de crianças medievais (Figura 10), inimigos (Figura 11), alquimistas (Figura 12) e raposas espirituais (Figura 13).

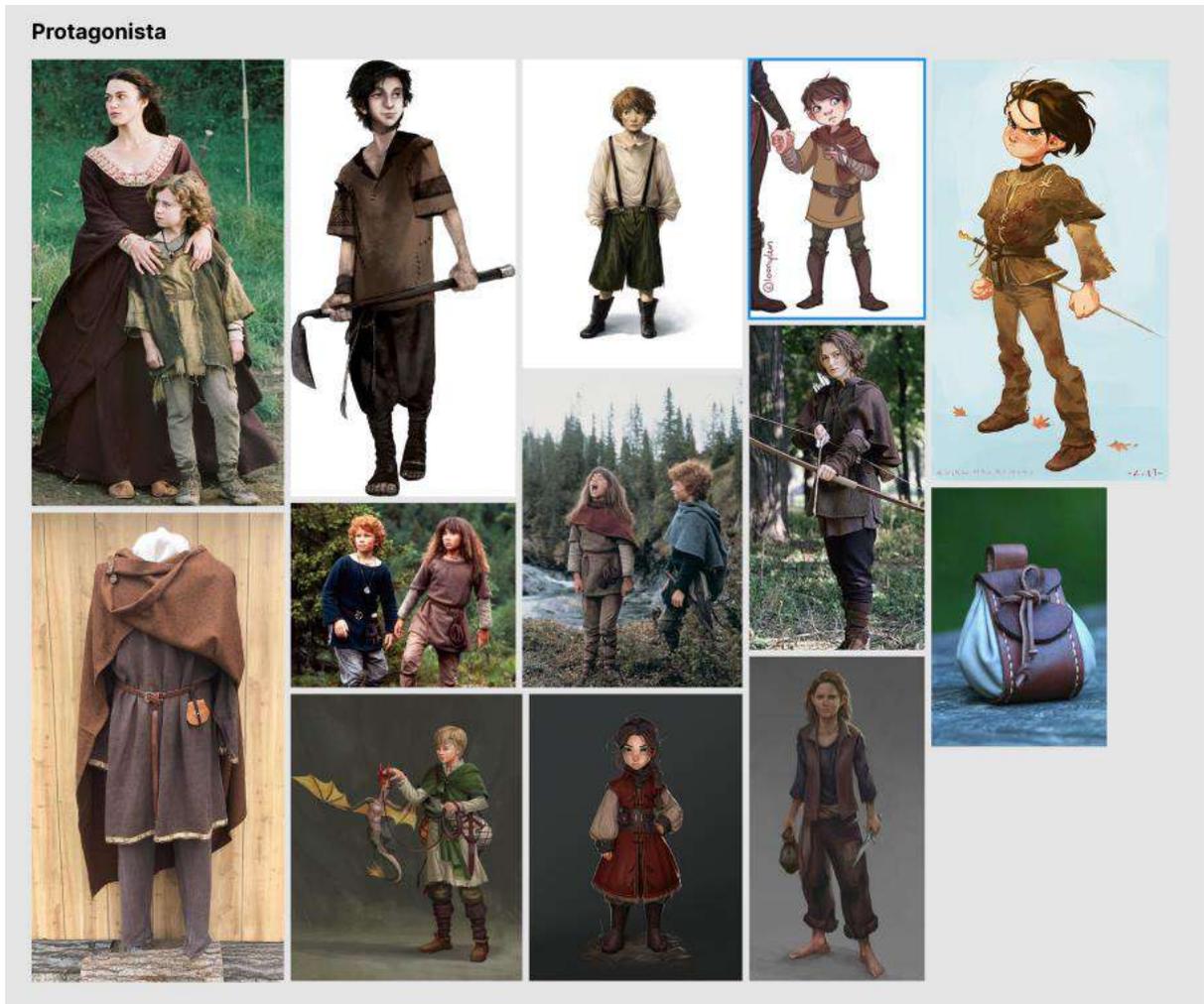


Figura 10. *Moodboard* de crianças medievais.

¹⁶ *Sprite* é a arte de um personagem ou um objeto dentro dos jogos 2D.

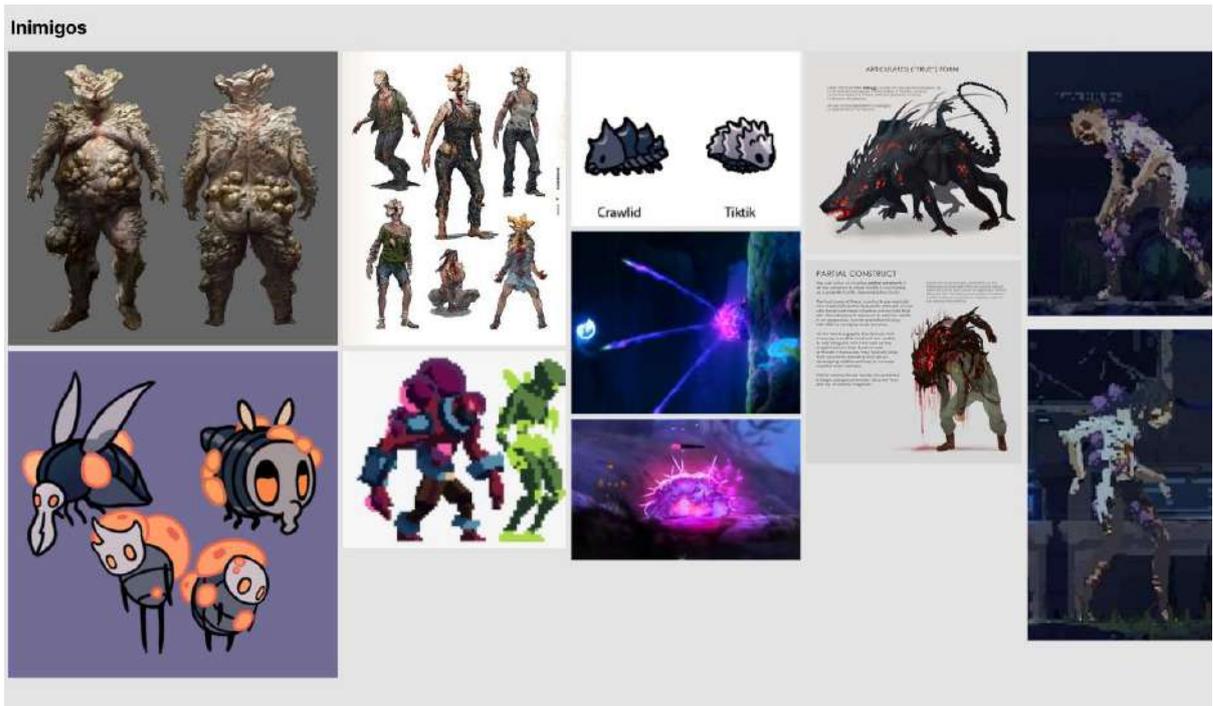


Figura 11. *Moodboard* de inimigos.

Alquimista

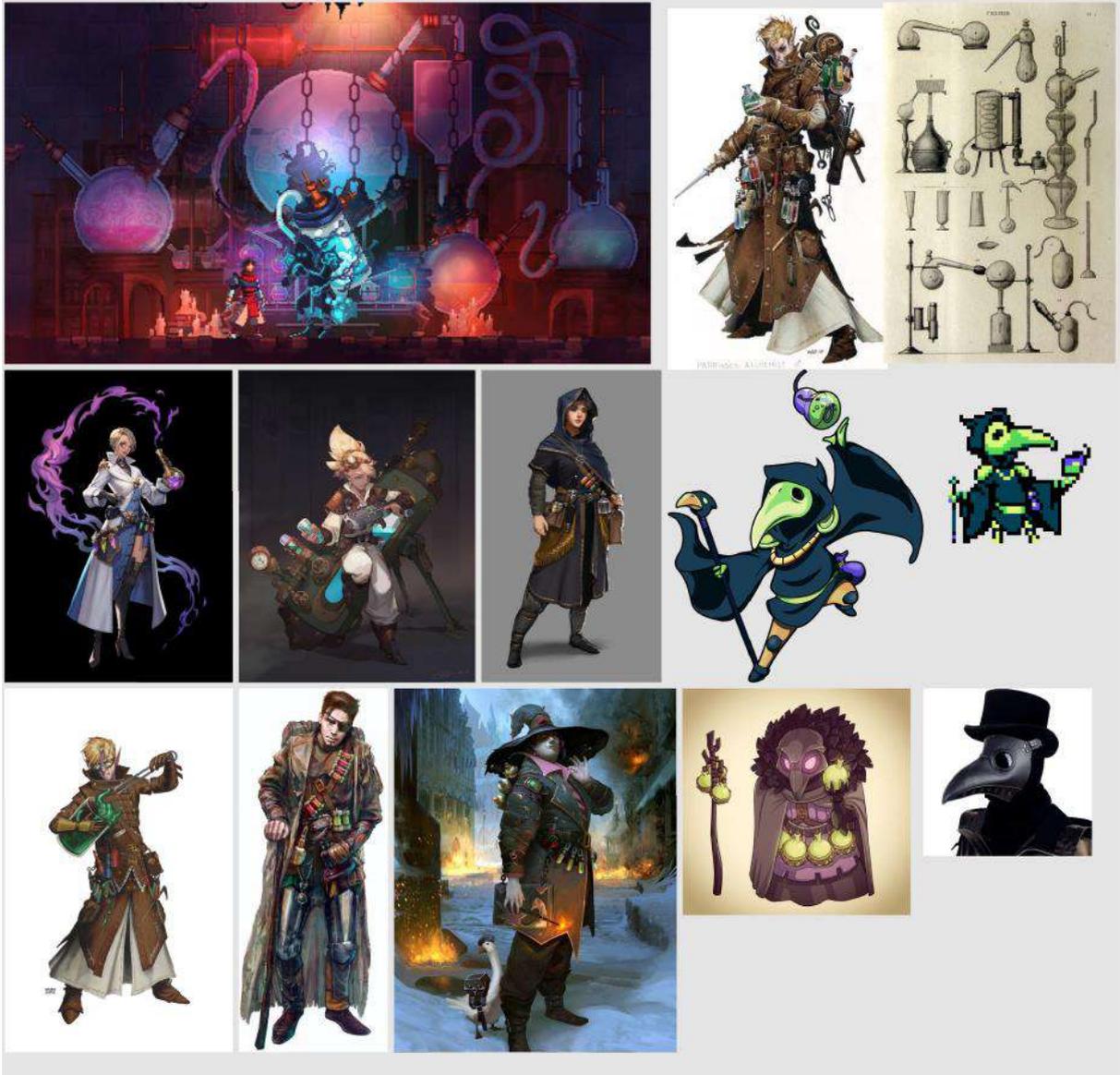


Figura 12. Moodboard de alquimistas.

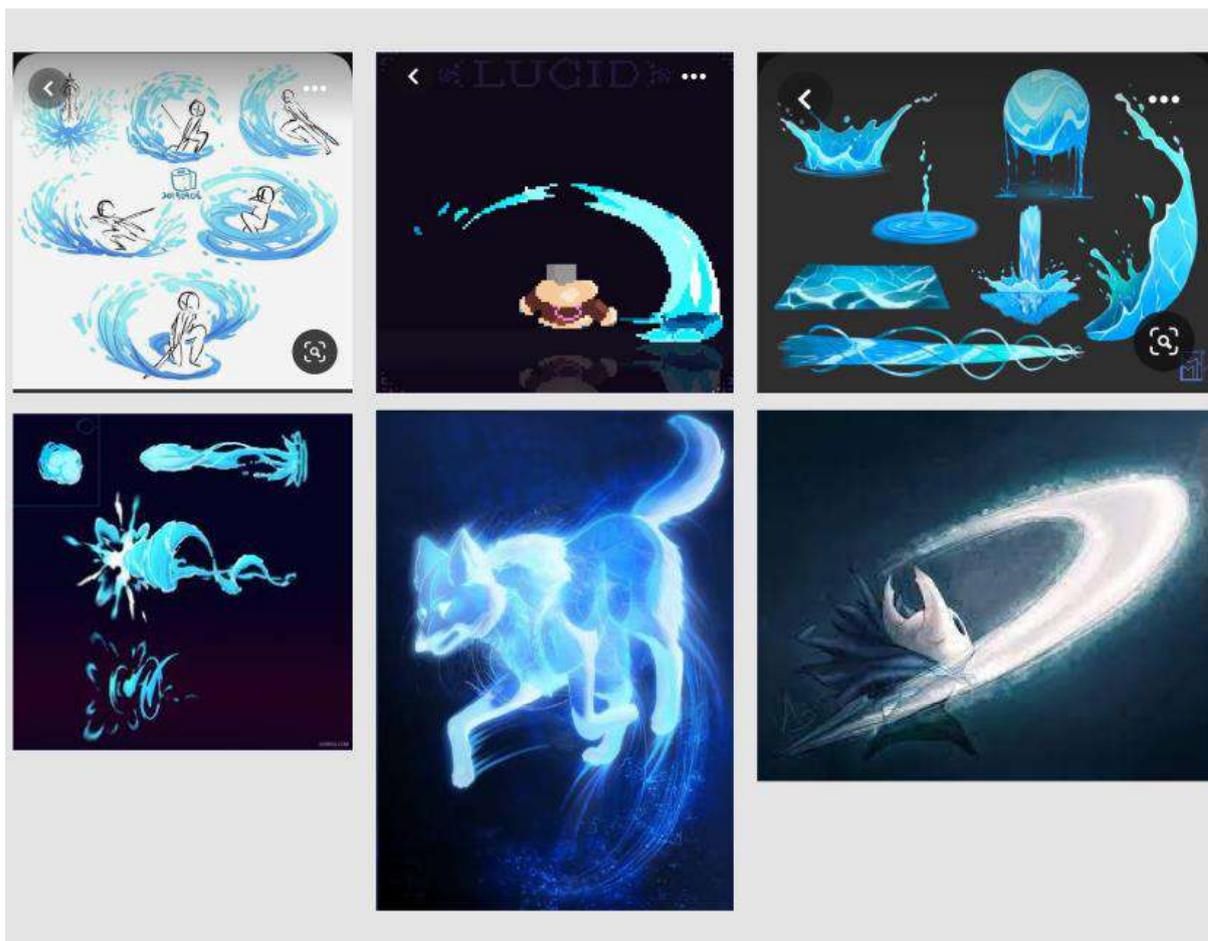


Figura 14. *Moodboard* de ataques e magias.

Além disso, utilizamos o *spritesheet*¹⁷ do jogo Momodora (2016) como principal referência e inspiração para a criação das animações do nosso protagonista (Figura 15).

¹⁷ *Spritesheets* são arquivos que guardam, dentro deles, uma sequência de *sprites*, os quais, ao serem colocados em uma *timeline* como *keyframes*, geram uma animação de determinado *sprite*.

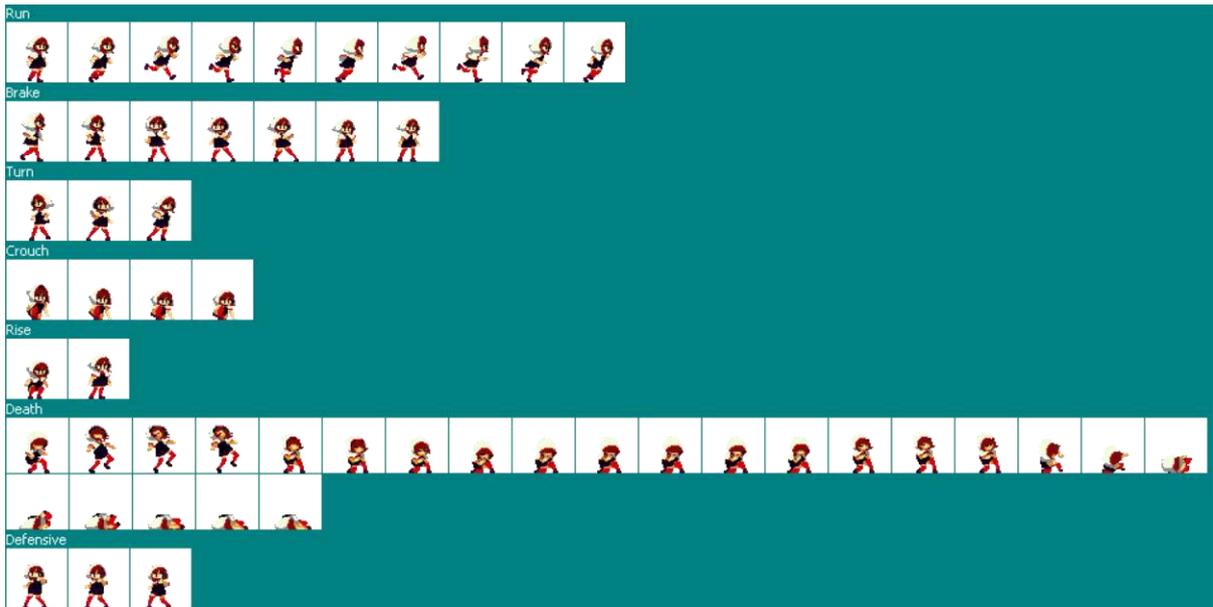


Figura 15. Spritesheet do jogo Momodora (2016).

Além dos aspectos visuais, exploramos as mecânicas de combate e movimentação (Figura 16). Também recolhemos e debatemos outros exemplos de jogos, dessa vez dentro dos gêneros Metroidvania e plataforma para servirem de referência para *level design*, mapa, movimentação e mecânicas gerais do jogo (Figura 17). Esses jogos costumam a conter um mapa grande e interconectado, que lentamente vai se tornando acessível ao jogador conforme ele ganha novas habilidades e explorando o cenário, muitas vezes descobrindo coisas novas ao voltar para ambientes já explorados. Nosso objetivo era enriquecer a experiência de jogo e tornar a navegação pelos mapas mais cativante no resultado final.

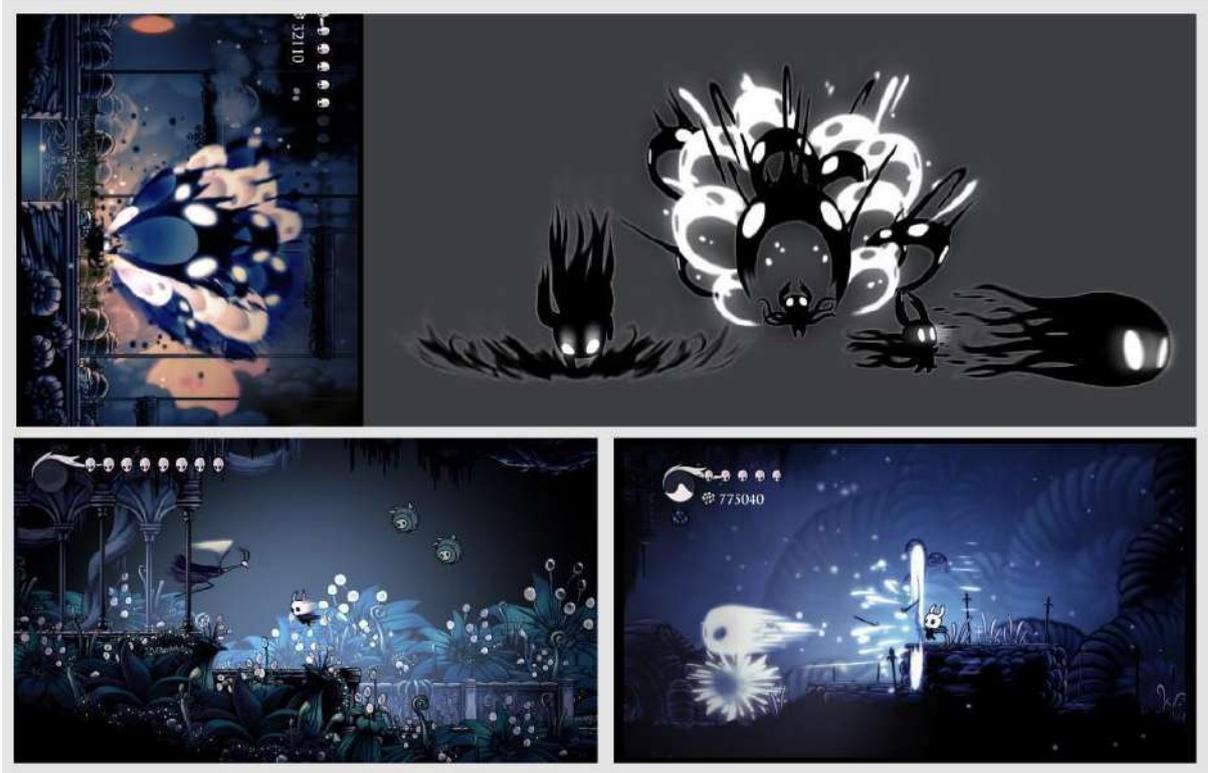


Figura 16. *Moodboard* de mecânicas de ataque.

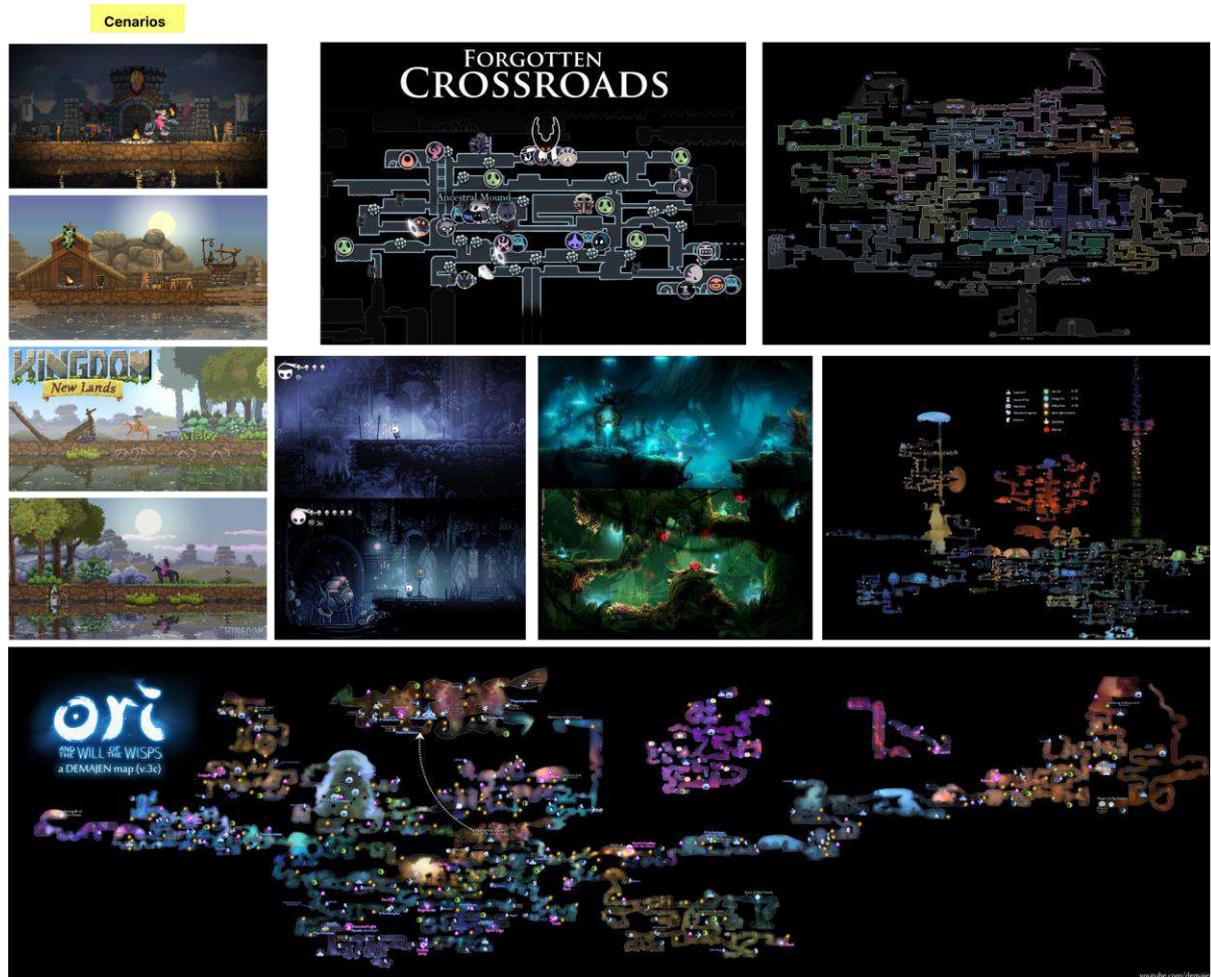


Figura 17. Moodboard de level design, mapa, movimentação e mecânicas gerais.

Em todas as etapas anteriores, foram debatidas cada uma das referências coletadas, e sobre como cada uma poderia servir de inspiração para definir a identidade do nosso jogo. Selecionamos e organizamos as principais referências de gênero e estilos visuais, destacando aquelas que consideramos mais relevantes. Além do já mencionado Metroidvania e plataforma, exploramos outros gêneros como *sidescroller*¹⁸ e *roguelike*¹⁹, reunindo exemplos de jogos representativos para cada um deles. Essa abordagem nos proporcionou uma base sólida para a construção e desenvolvimento criativo do nosso projeto.

Adicionalmente, analisamos e organizamos referências para telas e interfaces (Figura 18) que nos agradaram, coletando inspirações de HUDs (Figura 19) para compreender

¹⁸ *Sidescroller*, também conhecido como jogo de rolagem lateral, é um jogo eletrônico no qual a jogabilidade é vista do ângulo da câmera de visão lateral e, à medida que o personagem do jogador se move para a esquerda ou para a direita, a tela rola com ele.

¹⁹ *Roguelike* é um subgênero de jogo RPG, caracterizado pela geração de nível aleatória ou procedural durante a partida, mapa geralmente baseado em ladrilho e morte permanente.

como organizar os elementos na tela de forma que o jogador consiga visualizá-los de forma clara e rápida. Identificamos quais elementos devem permanecer visíveis o tempo todo e quais podem ser escondidos totalmente ou apresentados em um segundo menu. Esse processo contribuiu para a criação de uma interface intuitiva e amigável para o jogador.

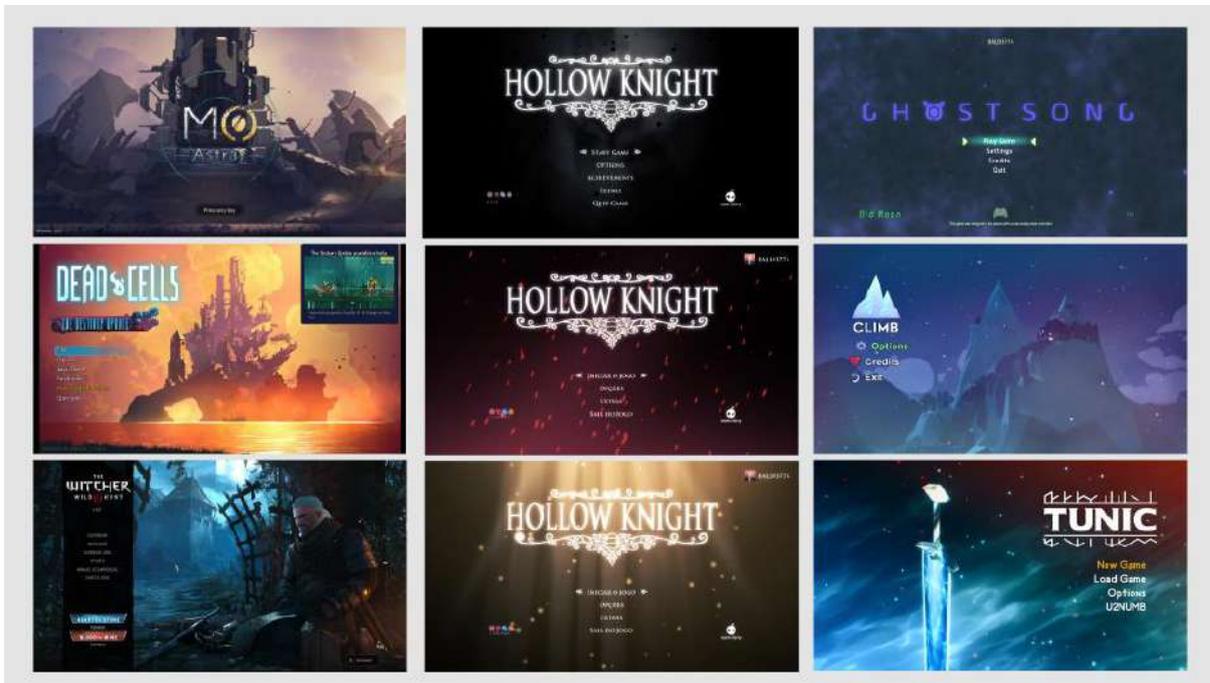


Figura 18. *Moodboard* de telas iniciais de jogos.

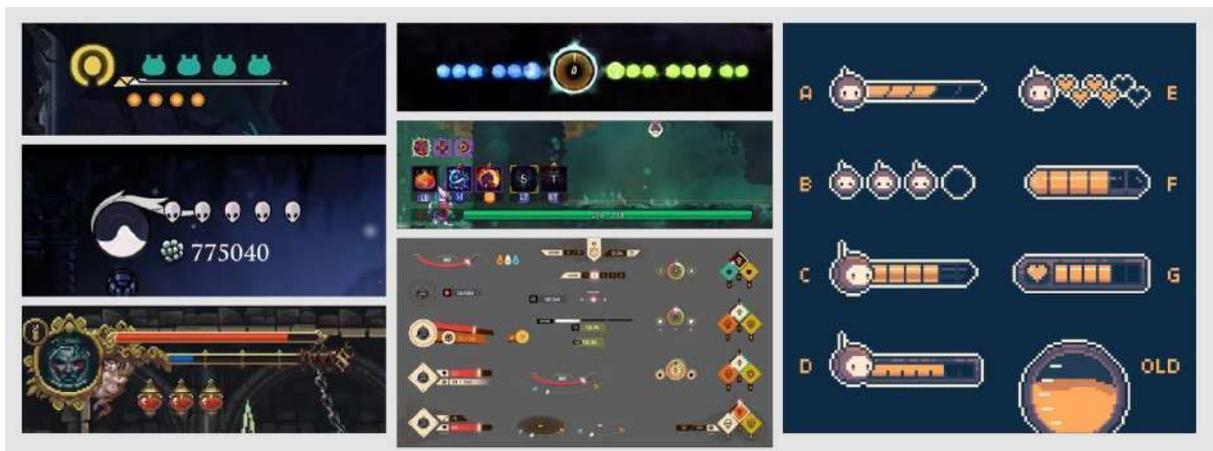


Figura 19. *Moodboard* de HUDs de jogos.

Em suma, mediante um processo que buscou precisão quanto ao humor pretendido, estabelecemos uma extensa e variada coleção de referências visuais, auditivas, textuais e mecânicas. Essa coleção foi organizada em *moodboards* disposta a todos os membros da

equipe, viabilizando uma plataforma centralizada e de fácil acesso. Essa acessibilidade foi fundamental para garantir que todos os colaboradores pudessem se inspirar e consultar os mesmos recursos ao contribuir para o desenvolvimento do jogo. Essa compilação não apenas serviu como uma base para o nosso trabalho criativo, mas também promoveu uma compreensão compartilhada dos elementos visuais, sonoros e narrativos que pretendemos integrar ao projeto.

A colaboração e a coesão da equipe, dessa forma, foram fortalecidas pela troca contínua de ideias e inspirações provenientes dessas referências comuns, permitindo o alinhamento de nossas visões, de forma que pudéssemos trabalhar de maneira mais eficiente para concretizar nossa visão coletiva. Ao centralizar e organizar nossas referências dessa maneira, maximizamos nosso potencial criativo e garantimos que todos os aspectos do jogo fossem desenvolvidos de forma harmoniosa.

5. História: Narrativa e Enredo do Jogo

Neste capítulo, abordaremos o processo de desenvolvimento e conclusão da versão final da história do jogo *Melius*. Schell define a história em jogos como o relato experienciado enquanto o jogador avança na sequência de eventos do jogo. Ele também complementa sobre a importância que a definição de outros elementos do jogo como mecânicas, estética e cenários tem sobre a história, podendo auxiliar na contação da história ou na imersão do jogador (Schell, 2014).

Para o nosso projeto, decidimos separar a história em dois pontos, que denominamos de enredo e a narrativa. O enredo consiste em todos os eventos, acontecimentos e conflitos que envolvem o universo criado, podendo ou não conter a presença de personagens, movidos por suas motivações, e de um contexto, que envolve o desenrolar dos acontecimentos. Um jogo pode existir sem uma trama definida, tendo como foco apenas uma série de desafios sequenciais para o jogador superar. Quando presente, a história serve para contextualizar os eventos do jogo, bem como explicar as motivações dos personagens e a lógica do universo criado.

O enredo do jogo pode afetar diretamente a narrativa. A **narrativa** consiste na descrição da sequência de eventos pelos quais o jogador passará, ou seja, é a progressão de eventos que o jogador experimenta durante o jogo. Portanto, consideramos a narrativa como a perspectiva do jogador sobre o significado na progressão dos eventos.

5.1 Processo de desenvolvimento do Enredo

Logo nas primeiras reuniões que o grupo organizou para discussão do projeto foi decidido criar uma história e que ela teria alta relevância e influência na progressão narrativa do jogo. Por esse motivo, em um primeiro momento, nos reunimos já para decidir seus contornos iniciais da história, e partir de uma primeira sessão de *brainstorming* com todo o grupo, discutimos as ideias dispostas por cada um dos membros, e as organizamos em um único documento no *google docs*²⁰. Assim, foi permitido que cada membro do grupo tivesse acesso

²⁰ *Google docs* é um processador de texto on-line que permite criar e formatar documentos e colaborar com outras pessoas

ao conjunto de ideias. Aos poucos, mediante a adoção, exclusão, refinamento, discussões e adaptação de ideias, moldamos o esboço geral da história do jogo.

Alguns elementos foram decididos de imediato, como, por exemplo, a presença de uma raposa, que poderia tanto ser um personagem principal quanto um *NPC*²¹ ou *companion*²². A ideia de um protagonista quadrúpede, no entanto, após uma breve pesquisa e conversa com o grupo, apresentou uma série de desafios artísticos e técnicos para o desenvolvimento e implementação em um jogo eletrônico, motivo pelo qual descartamos a ideia de tê-la como personagem principal. Além disso, percebemos que grande parte das referências por nós selecionadas contavam com protagonistas humanos ou humanoides. Porém, ainda sem querer descartar a ideia de ter a raposa no jogo, decidimos torná-la um *companion*, que poderia interagir e acompanhar o personagem principal durante grande parte do jogo.

Partindo desses dois elementos centrais, pincelamos as primeiras linhas da história: o jogo contaria a aventura de um garoto órfão e uma raposa, que, se ajudariam, buscando sobreviver em um mundo cheio de perigos, animais agressivos, monstros, bárbaros e adultos armados. Mais adiante, chegamos à conclusão de que o protagonista possuiria, de fato, uma mãe, não sendo mais órfão. A mãe seria o gatilho da história, estando doente, forçaria o protagonista da história a partir em busca de uma cura para ela visto, também servindo de elemento motivacional para o jogador.

5.1.1 Enredo Selecionado

Após diversas discussões e refinamento, a versão do enredo construída pelo grupo ficou da seguinte forma:

“O protagonista é apenas um garoto que mora sozinho com sua mãe em um bosque afastado da cidade. Sua mãe acaba adoecendo em um certo momento, ficando muito debilitada, o que leva o garoto a sair de casa para procurar ajuda. Antes de sair, no entanto, sua mãe o chama e lhe entrega um amuleto muito importante que estava em seu pescoço, dizendo ainda que ele irá protegê-lo. Com isso, o menino sai em busca de ajuda, indo atrás de um alquimista famoso do qual havia ouvido falar. O alquimista, após ouvir os apelos do garoto, diz saber como

²¹ *NPC* é uma sigla para “*non playable character*” ou em português “personagem não jogável” e se refere a todos os personagens presentes no jogo na qual o jogador não tem controle direto ou indireto sobre.

²² *Companion* se refere a um personagem no jogo na qual o jogador tem controle limitado sobre, muitas vezes indiretamente ou diretamente. Esse personagem acompanha o jogador durante o *gameplay*.

criar uma cura para aquela doença, mas que precisaria de 4 ingredientes, que o protagonista teria que trazer para ele. Com isso, o alquimista guia o garoto à entrada de uma antiga cidade subterrânea em ruínas, explicando que os ingredientes só poderiam ser encontrados lá.

Ao chegar no local indicado pelo alquimista, o protagonista acaba caindo num buraco que o leva para os níveis mais fundos dessas ruínas. Lá ele acaba sendo encurralado por um monstro, e a raposa surge protegendo-o e entrando em seu amuleto logo em seguida.

Ambos exploram as ruínas juntos, a raposa emprestando seus poderes para que o garoto pudesse se proteger, e após enfrentar vários perigos, eles conseguem todos os ingredientes. De volta à superfície, eles vão em busca do alquimista, mas ele acaba se revelando o vilão, que queria usar os ingredientes para os seus propósitos, se transformando também num monstro, que o garoto também precisa enfrentar. Com a derrota do alquimista, o garoto vasculha seus estudos, e com ajuda da raposa, consegue completar um elixir de cura, que ele leva para casa salvando sua mãe.

O alquimista aqui seria o grande vilão, que acabou causando a destruição da cidade subterrânea após seus experimentos com pessoas darem errado, e gerarem uma doença terrível que transformou a maioria dos moradores em monstros. Ele conseguiu escapar e recomeçou a sua pesquisa na cidade próxima ao bosque onde morava o protagonista, o que leva mais pessoas a ficarem doentes, incluindo a mãe do garoto. Porém, muitas das coisas que ele precisava para prosseguir com seus experimentos acabaram ficando na cidade subterrânea, e por isso, ele enviou diversas pessoas para lá, prometendo grandes pagamentos, ou ajuda medicinal.

A raposa seria uma entidade espiritual ligada à água e a vida, que era venerada pelos moradores da cidade subterrânea e trazia prosperidade à ela. Após a destruição da cidade e o alastramento da doença, ela ficou esquecida e enfraquecida.”

5.2 Processo de desenvolvimento da Narrativa

O processo de desenvolvimento da narrativa ocorreu como parte do desenvolvimento da história do jogo. Por isso, além de ter sido diretamente afetado pelo delineamento do enredo, ele também o influenciou. Para que a história pudesse se fazer presente, a experiência do jogador teria que caminhar junto com a progressão de acontecimentos do enredo, ditada pela narrativa, que deveria ser distribuída de maneira efetiva e engajante

dentro do jogo, sem qualquer tipo de empecilho ou barreira à história ou ao *gameplay*. Além do mais, decidimos, ainda durante as discussões previamente realizadas, que a narrativa precisaria ser bem definida, mas que a sequência de eventos não seria totalmente rígida. Essa é uma característica própria do gênero que decidimos como o principal do Melius, isto é, Metroidvania, baseado na ideia de progressão, exploração e de história não linear.

5.2.1 Narrativa Selecionada:

A ordem de eventos narrativos final decidida pelo grupo é a seguinte (Figura 20):

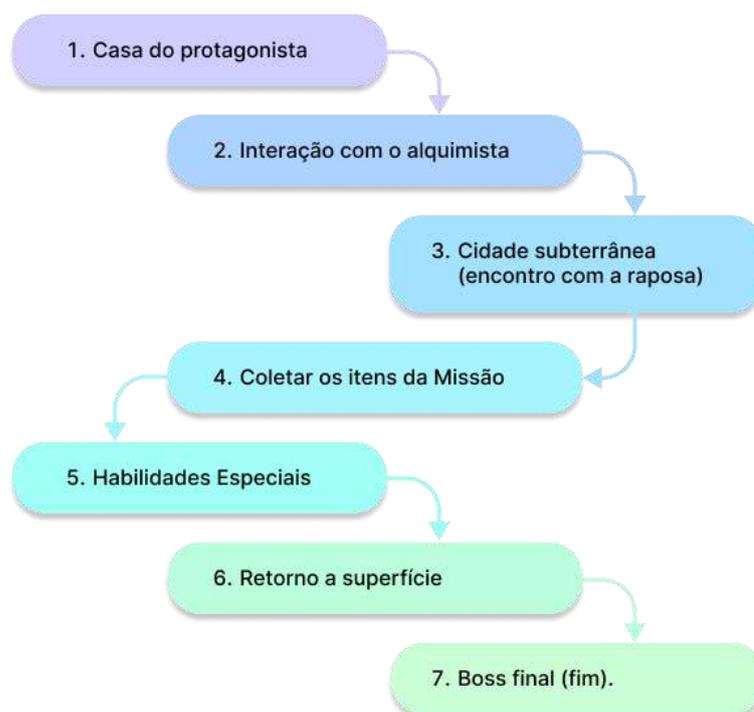


Figura 20. Fluxo narrativo inicial.

1. Personagem principal, motivado pela necessidade de salvar sua mãe doente decide partir de sua casa em direção Leste, onde ele ouviu falar que um alquimista muito habilidoso vive;
2. Chegando lá o alquimista o manda voltar para Oeste e entrar na caverna que leva até a cidade subterrânea, onde ele teria que encontrar 4 itens importantes para criar uma poção para salvar a sua mãe. (após interagir com o alquimista a entrada para a cidade subterrânea abre);

3. Ao Chegar na cidade subterrânea o jogador libera a Raposa como sua *companion*. (após adquirir a Raposa como seu *companion* o jogador ganha habilidade únicas);
4. O jogador pode ter acesso a mais habilidade únicas durante a exploração da cidade subterrânea;
5. Na cidade subterrânea o jogador precisa coletar os 4 itens e, para conseguir sair da cidade, ele precisa ativar 3 mecanismos para abrir a porta de saída. (O jogador é livre para decidir a ordem de coleta dos itens e ativação dos mecanismos);
6. Coletando todos os itens e voltando ao alquimista ele descobre que o alquimista nunca teve a intenção de ajudá-lo ou salvar sua mãe,
7. Então o alquimista se torna o vilão principal e o *Boss*²³ final, o jogador precisa derrotá-lo em uma batalha.

5.3 Modificações propostas no Primeiro Ciclo de Revisão

Após o final da matéria de Design de Jogos, optamos por continuar o desenvolvimento do jogo. Por essa razão, acabamos expandindo e refinando a história e adicionando um processo de *worldbuilding*²⁴. Com isso, introduzimos novos eventos, que, apesar de estarem situados temporalmente antes do início do jogo, possuem relação direta ou indireta com os eventos presenciados pelos jogadores. Além disso, demos maior atenção ao desenvolvimento e aprofundamento de alguns personagens, detalhando suas motivações e características.

O Alquimista, por exemplo, deixou de ser o vilão principal, tornando-se apenas um antagonista inicial, afetado por uma série de eventos que antecedem o jogo. Também demos uma atenção maior à relação do protagonista e sua mãe, que passaram a ter uma conexão direta com a cidade subterrânea. Decidimos que sua mãe seria uma peça central na história que antecede o enredo do jogo. A doença, que, de início, espalhava-se por todo o universo criado por nós, tornou-se uma espécie de entidade, motivada pelo propósito de destruição, originada na cidade subterrânea. A cidade, por sua vez, existiria para proteger o universo deste “mal”. Por

²³ **Boss** ou *bosses* é uma palavra do inglês usada para se referir a um inimigo particularmente poderoso que o jogador deve enfrentar e derrotar, geralmente ao final de uma fase, para poder continuar a avançar no jogo.

²⁴ **Worldbuilding** é uma expressão do inglês, podendo ser traduzida literalmente para “construção de mundo”, e é nome dado ao processo de criação de um mundo fictício coerente e detalhado em narrativas.

fim, decidimos que a raposa não seria o único espírito animal/elemental. Ela passou a fazer parte de outras 3 entidades que viviam na cidade subterrânea, em harmonia com seu povo.

Assim como mencionado anteriormente no capítulo de Delineamento, tivemos a oportunidade de trabalhar na expansão da história durante a disciplina de Estudos Avançados em Design: Ateliê de Produção de Quadrinhos, desenvolvendo uma Zine intitulada “Melius a Queda de um Reino” (Figura 21). Nela, contamos e ilustramos parte da história que criamos para universo do nosso jogo como trabalho final para a disciplina, processo que nos permitiu enriquecer mais o universo criado.



Figura 21. “Melius: A Queda de um Reino” 14/06/2023

5.3.1 Acontecimentos anteriores ao jogo

“Anos atrás, o reino subterrâneo de Aiyrah vivia em paz. A cidade prosperou sob o comando da rainha Eliyah, que tinha a benção dos deuses. Ela não só governa a cidade, mas mantém um pacto que sua família fez há dezenas de gerações atrás. Esse pacto se deu quando a Doença surgiu nesse mundo milhares de anos atrás, se espalhando rapidamente levando a humanidade a quase completa extinção. Tendo uma origem mística, a Doença, além de ser altamente infecciosa e mortal, ela transformava uma parte do infectados em corpos ambulantes, criando algo que se assemelhava a um exército de criaturas quase imortais, não permitindo ela ser contida completamente por meios convencionais. O povo de Aiyrah recorreu então aos Deuses por ajuda. Quatro deles decidiram ajudar, firmando então um contrato com a rainha daquele povo, dando a eles o seu poder em troca dela e seu povo tomar para si a responsabilidade de manter a doença selada para todo o sempre. Isto ocorreu por meio de todas as gerações. Era sempre a rainha quem herdava a capacidade de se comunicar com os Deuses, por esse motivo Aiyrah é um reino matriarcal.

A rainha designava os quatro guardiões da cidade, cada qual portava um amuleto com proteção de uma entidade mágica que lhe concede poderes. São eles: a Guardiã da raposa da água (mãe do protagonista), a Guardiã da coruja do ar, o Guardião do cervo da terra e o Guardião do lagarto espinhoso do fogo.

Os guardiões são portadores dos poderes dos deuses, escolhidos à dedo pela rainha, e tem como principal papel auxiliar na contenção da Doença. Eles também exercem papéis de proteção da cidade, proteção da rainha e comandantes do exército de Aiyrah. Além de auxiliar em outras questões quando necessário.

No entanto, vários impérios na superfície cobiçavam a magia e poderes do povo subterrâneo, e tentavam de várias formas negociar com Aiyah para ter posse de alguma fração desse poder, sem sucesso. Mas após várias tentativas falhas, esses Impérios começaram a recorrer a meios mais brutais para obter tais poderes, até que em que através de uma falsa visita diplomática, três dos impérios formaram uma aliança para enviar representantes para forjar um acordo diretamente com a rainha.

Durante as negociações um dos três diplomatas atacou a rainha e, infelizmente, os guardiões não foram capazes de reagir ao ataque surpresa e a Rainha Eliyah Loraethus faleceu no ano de 8XX, ela foi sepultada nas profundezas da cidade junto com suas antecessoras, próximo ao local do pacto com os deuses como manda a tradição. Após a perda de sua rainha,

a conexão do reino subterrâneo de Aiyrah e seus guardiões aos Deuses foi enfraquecida, por a rainha não possuía nenhuma sucessora.

A aliança começou a se mover imediatamente, levando seu exército em direção ao reino de Aiyrah, que se fechou completamente após o ataque e mesmo sem a sua rainha, se manteve forte e resistiu aos ataques, usando suas tropas e o poder dos guardiões em batalha. Assim, os guardiões tiveram que tomar a frente e proteger o reino dos invasores e a Doença. Mas em meio a guerra, enquanto os guardiões lutavam para proteger os portões da cidade, eles esqueceram do seu propósito inicial, que era vigiar e conter a doença devastadora que vinha sendo contida pelos povos do subterrâneo por gerações. Nesse momento de descuido a doença escapou das profundezas da cidade subterrânea e se alastrou rapidamente pela cidade, passando pelo campo de batalha e finalmente chegando à superfície, devastando tudo em seu caminho e matando a grande maioria, deixando poucos sobreviventes e muitos infectados para trás.

No meio do caos, a guardiã da raposa de água decidiu fugir para salvar seu filho recém-nascido. Ela encontrou uma casa abandonada, que logo se tornou o seu novo lar, e foi onde eles viveram durante anos juntos. A fim de manter seu filho seguro, ela o presenteou com o amuleto de proteção da raposa. A guardiã havia contraído a doença, mantendo em segredo do seu filho, mas então chegou um ponto em que era impossível esconder que estava doente.”

5.3.2 Enredo Mais Atual:

“A partir daí, determinado a encontrar uma cura para a mãe doente, o protagonista sai daquele ambiente seguro que vivia em busca de ajuda; indo atrás de um alquimista famoso do qual havia ouvido falar. O alquimista, após ouvir os apelos do garoto, diz saber como criar uma cura para aquela doença, mas que precisaria de quatro ingredientes, que o protagonista teria que trazer para ele. Com isso, o alquimista guia o garoto à entrada de uma antiga cidade subterrânea em ruínas, explicando que os ingredientes só poderiam ser encontrados lá.

Ao chegar no local indicado pelo alquimista, o protagonista acaba caindo num buraco que o leva para os níveis mais fundos dessas ruínas. Lá ele acaba sendo encurralado por um monstro, e a raposa surge protegendo-o e entrando em seu amuleto logo em seguida.

Ambos exploram as ruínas juntos, a raposa emprestando seus poderes para que o garoto pudesse se proteger, e após enfrentar vários perigos, eles conseguem todos os ingredientes. No

meio do caminho, descobrirá mais sobre a história da sua mãe, daquele povo e daquele reino, além de desvendar os segredos por trás da doença enquanto enfrenta inimigos, sobrevive perigos e enfrenta o que um dia foram guardiões da cidade e companheiros de sua mãe com a ajuda da raposa protetora.”

5.3.3 Narrativa Mais Atual:

A ordem de eventos narrativos final decidida pelo grupo são os seguintes em ordem linear (figura 22):

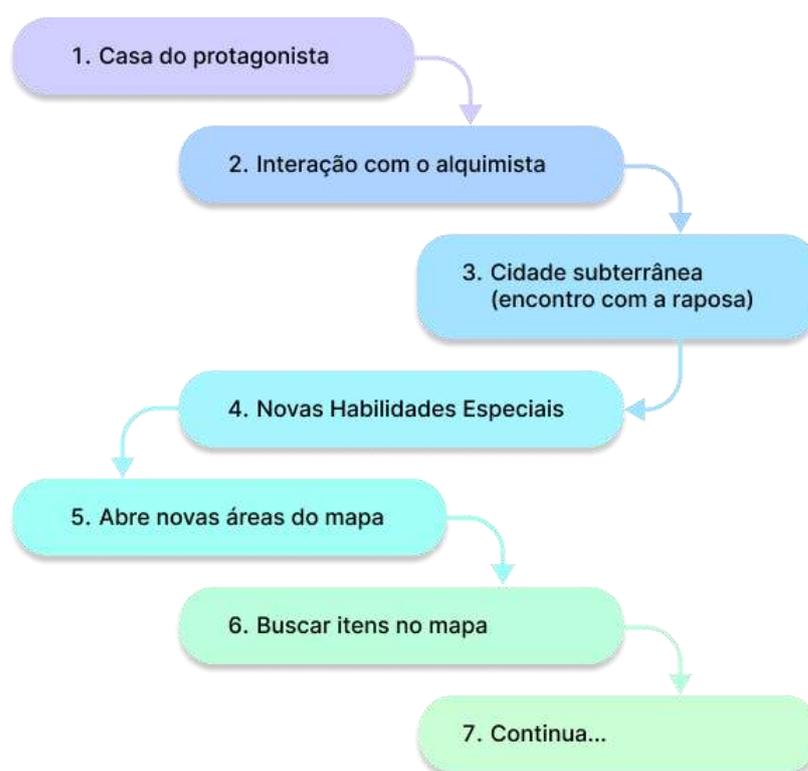


Figura 22. Fluxo narrativo atual.

1. Personagem principal, motivado pela necessidade de salvar sua mãe doente decide partir de sua casa em direção Leste, onde ele ouviu que um alquimista muito habilidoso vive;
2. Chegando lá o alquimista o manda voltar para Oeste e entrar na caverna que leva até a cidade subterrânea onde ele teria que encontrar os itens necessários para criar uma poção para salvar a sua mãe. (após interagir com o alquimista a entrada para a cidade subterrânea abre);

3. Ao Chegar na cidade subterrânea o jogador libera a Raposa como sua *companion*. (após adquirir a Raposa como seu *companion* o jogador ganha habilidade únicas);
4. O jogador passa ter acesso a mais habilidade únicas (necessárias e opcionais) durante a exploração da cidade subterrânea;
5. Para conseguir acesso a outras áreas do reino o jogador precisa ativar 3 mecanismos para liberar a passagem para essas outras áreas. (O jogador é livre para decidir a ordem de ativação dos mecanismos e exploração das áreas);
6. Os itens necessários para criar a poção de cura estão espalhados em todos os mapas (O jogador é livre para decidir a ordem de coleta dos itens);

5.4 Considerações sobre a Narrativa

Assim, pudemos desenvolver toda a história do jogo, que, por sua vez, influenciou também a narrativa e o desenvolvimento visual dos personagens e cenários. A narrativa, como dissemos, foi definida em conjunto com a história, tendo em vista que ambas estão intrinsecamente associadas ao jogo. Sua definição foi imprescindível pois ela influencia diretamente o *level design*, sempre pensado de acordo com a progressão narrativa do jogo, temas que abordaremos com mais detalhes posteriormente.

6. Regras e Mecânicas

Neste capítulo, falaremos sobre as regras e mecânicas do jogo. As mecânicas consistem no campo de possibilidade de ações dispostas ao jogador dentro do universo do jogo, como, por exemplo, a resposta do personagem ao acionamento de um botão específico do controlador. Já as regras são a delimitação de uma série de regulações que condicionam as ações do jogador durante sua experiência, como a impossibilidade de atravessar paredes ou sair da área de jogo definida. Schell (2014) define as regras e mecânicas como o que determina como os jogadores interagem com o jogo, o objetivo do jogo e como jogadores podem ou não alcançá-lo, assim como o que acontece quando eles tentam.

É importante o desenvolvimento desta etapa, pois é nela que estabelecemos uma série de características de como o jogo vai se comportar, e de como o jogador vai interagir com ele. Sem uma definição detalhada e quantificada, a implementação do jogo pode ficar comprometida, divergindo muito da experiência de jogo planejada para o jogador. Por exemplo, a mecânica de salto define e quantifica altura, distância e outras características para que seja implementada pela programação, que influencia como a animação de pulo deve ser feita e em como as plataformas serão posicionadas no *level design*.

Começamos com um *brainstorming*, momento em que cada membro do grupo sugeriu ideias a respeito das mecânicas e regras com base em suas próprias experiências e referências. Todas as ideias foram colocadas em discussão, e, em um primeiro momento, priorizamos avaliar o quão interessante ou divertida cada ideia proposta parecia, e como cada uma se encaixaria com a temática e gênero de jogo que estabelecemos como base para nosso jogo. Após a primeira rodada de filtragem de ideias, partimos para uma discussão um pouco mais prática, em que consideramos três fatores principais: o nível de implementação das mecânicas e regras propostas (considerando o processo de programação), a viabilidade da criação e produção dos *assets* (considerando o nível de habilidade e experiência dos artistas) e o tempo disponível para o seu desenvolvimento.

Após a definição de um norte para essa etapa do projeto, começamos a implementar as mecânicas e regras estabelecidas, assim como demos início à produção dos *assets*. De início, elaboramos esboços iniciais que permitiram-nos realizar uma série de testes. É importante ressaltar que, nesse momento, a etapa de definição de regras e mecânicas começou a ocorrer

em conjunto com as etapas posteriores, então veremos mais detalhes sobre as etapas de testes nos capítulos que tratam dos personagens e no capítulo que trata dos mapas e cenários. Junto com os testes, mantivemos constantes sessões de debates, nas quais pudemos avaliar a necessidade de realizar mudanças e ajustes de acordo com a experiência de jogo obtida.

6.1 Regras e Mecânicas Gerais e de Mapas

Regras e mecânicas gerais são aquelas que constituem a base para o jogo e se aplicam tanto ao personagem jogável quanto aos inimigos e NPCs. Em alguns casos específicos, se aplicam de maneira diferente ou não se aplicam. Elas também funcionam como base para a construção das mecânicas e regras específicas e únicas que se aplicam a eles.

Começaremos pela **Colisão**: áreas/objetos/blocos sólidos nos quais um *asset* com colisão não pode atravessar outro. Essencial para criação dos mapas onde serão utilizados para criar chão, plataformas, paredes, e toda a área navegável pelo jogador (Figura 23), ela também será utilizada para circular os limites de cada mapa, para evitar que o jogador saia da área intendida de jogo. Além disso, o personagem jogável, inimigos e NPCs também possuem colisão, para que assim eles possam se movimentar sobre o mapa.

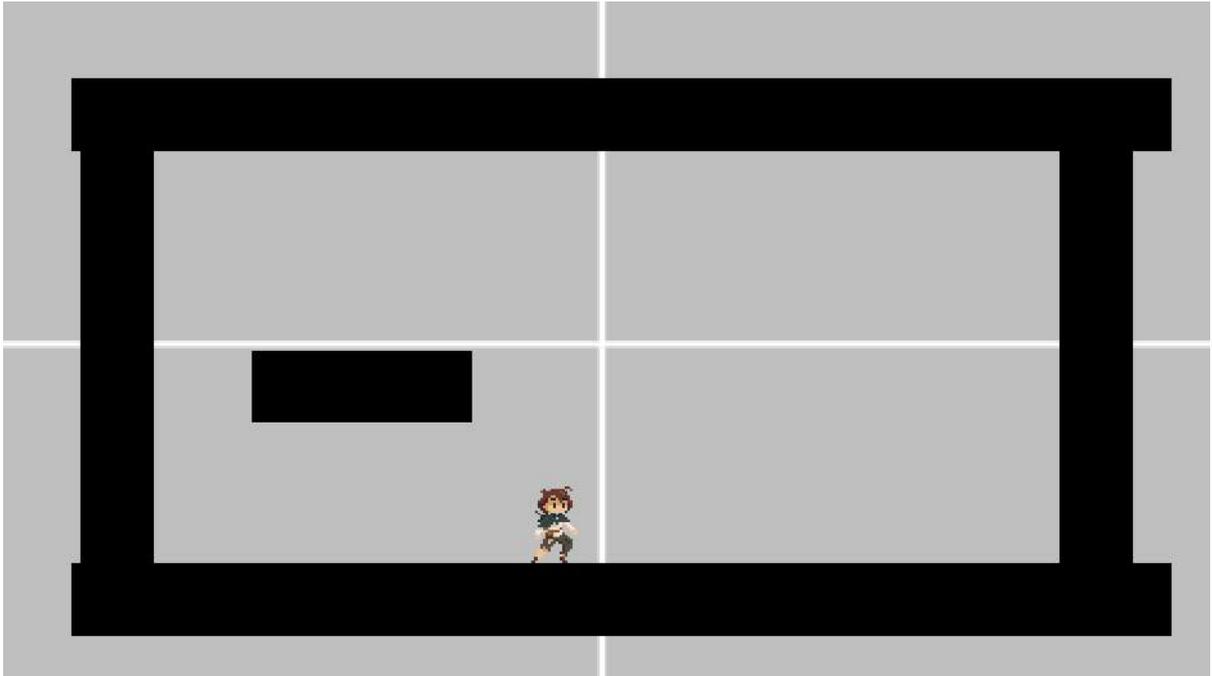


Figura 23. Blocos de colisão

Para que a detecção da colisão seja feita, é necessário que *assets* possuam uma *hitbox*. *Hitbox*, ou “caixa de acerto”, em uma tradução direta, é a indicação da área de contato de objetos e *assets* do jogo, mecânica essencial utilizada, como dito anteriormente, para a detecção da colisão entre o personagem jogável e os blocos de colisão do mapa, como pode ser visto na imagem demarcado em vermelho (Figura 24). *Hitbox* é também utilizada para a detecção de dano e como base para quase todas as outras regras e mecânicas do jogo.

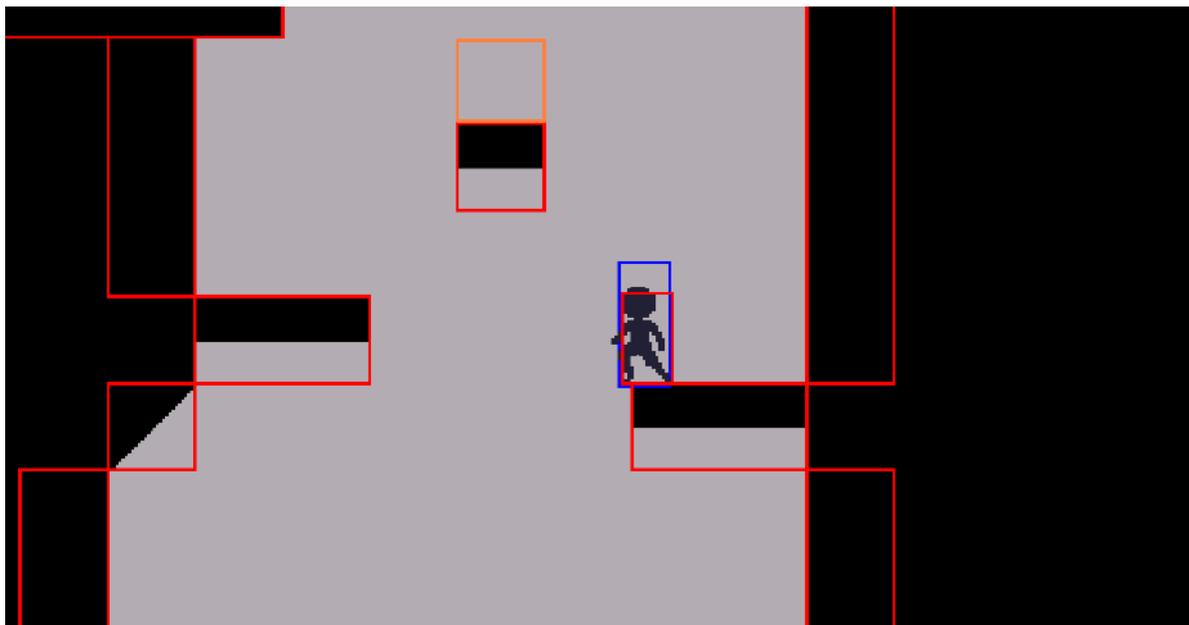


Figura 24. Exemplificação da *hitbox* no jogo.

Por se tratar de um jogo 2D, a **movimentação** é restrita apenas ao eixo X e Z, o personagem jogável. O personagem jogável, inimigos e *bosses* possuem a capacidade de se movimentar no eixo X para a esquerda e para a direita, e somente o personagem jogável pode se movimentar no eixo Z a partir de uma habilidade de pulo.

O personagem jogável e todos os inimigos possuem um número de **pontos de vida**, a quantidade que cada um possui varia e foi decidida individualmente. Os pontos podem ser subtraídos ao sofrer **dano**, ao sofrer um ataque em combate ou, ao encostar em um bloco de dano, é subtraída uma quantidade específica do total de pontos de vida (a quantidade deduzida é definida pelo infligidor do dano). Se esses pontos de vida chegarem a zero, ele **morre**, uma animação de morte é ativada e, no caso de inimigos e *bosses*, sua inteligência artificial e *hitbox* são desativados.

O **combate** no jogo consiste em o jogador ao pressionar os botões determinados para que o personagem controlado por ele execute uma ação, podendo ser uma série de ataques ou habilidades especiais disponíveis ao personagem jogável para se defender e sair vitorioso no embate contra inimigos e *bosses*. Em algumas situações, o combate é opcional, permitindo, se o jogador desejar, evitá-lo completamente.

Ainda para os mapas, haverá locais denominados blocos de dano. **Blocos de dano** são áreas fixas no mapa que afetam o personagem jogável, inimigos e *bosses*, e, ao ser registrado

um *Hit*²⁵ entre eles e um bloco de dano, uma quantidade fixa de pontos de vida é deduzida dos pontos de vida que ele tem. A duração do ataque é de 1 *hit* e a quantidade de pontos de vida deduzidos é definida por quem recebeu o dano, sendo elas 1 ponto de vida (por *hit*) ao personagem jogável, todos os pontos de vida restantes (por *hit*) a inimigos e 1 ponto de vida (por *hit*) a *bosses*. Esses blocos serão utilizados no *level design* (Figura 25) para criar obstáculos para o jogador enquanto ele navega pelo mapa. Isso torna a travessia pelo mapa mais interessante, fazendo o jogador calcular os seus pulos e até analisar o mapa para usar ao seu favor, criando armadilhas e atraindo os inimigos para esses blocos.

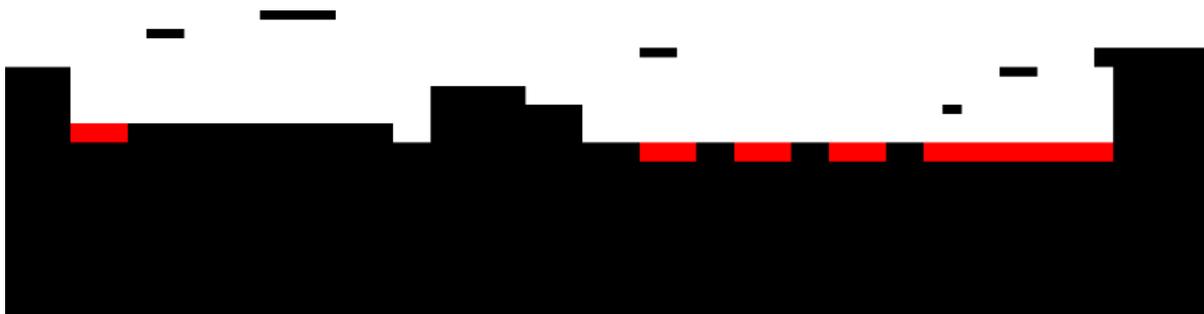


Figura 25. Blocos de dano.

6.2 Regras e Mecânicas do Personagem Jogável

Agora, falaremos sobre as regras e mecânicas do personagem jogável, abordando em detalhes e justificando cada uma delas, além de comentar especificações das mecânicas gerais que talvez se apliquem de maneira diferente ao personagem jogável, um exemplo sendo os blocos de dano mostrados anteriormente, que ao registrar um *hit* repelem o personagem jogável, o que não acontece com os inimigos, que morrem instantaneamente ao entrar em contato com tais blocos.

²⁵ *Hit* é a nomenclatura utilizada para sinalizar a colisão de duas *hitboxes*.

6.2.1 Movimentação

A movimentação se dá de maneira que o personagem seja controlado a partir do comando do jogador, podendo, o personagem, se movimentar livremente em uma velocidade fixa para a direita e para a esquerda, a partir da utilização das setas do teclado (Figura 26).

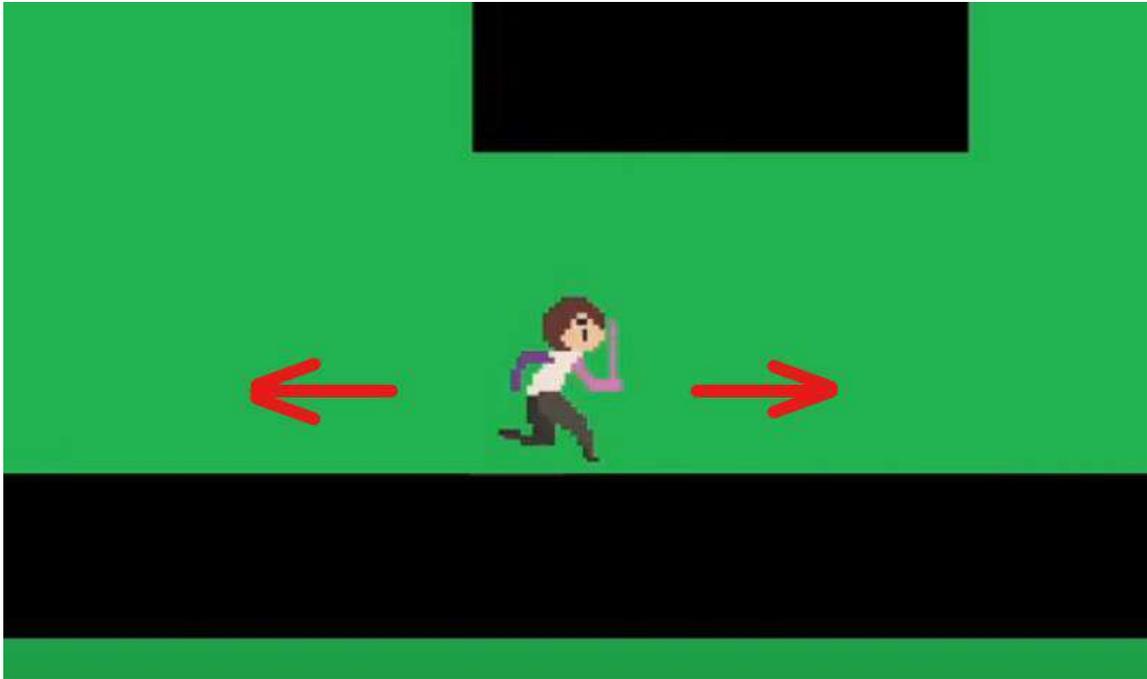


Figura 26. Mecânica de movimentação.

A movimentação no eixo Z se dá a partir da habilidade de **salto** ativada pressionando a tecla “S” no teclado. O salto possui uma altura máxima de até 4 *tiles* e o jogador mantém o controle da direção do personagem durante o salto. Além disso, o jogador consegue controlar a altura do salto e ajustá-lo de acordo com o tempo em que o botão de salto é mantido pressionado (Figura 27).

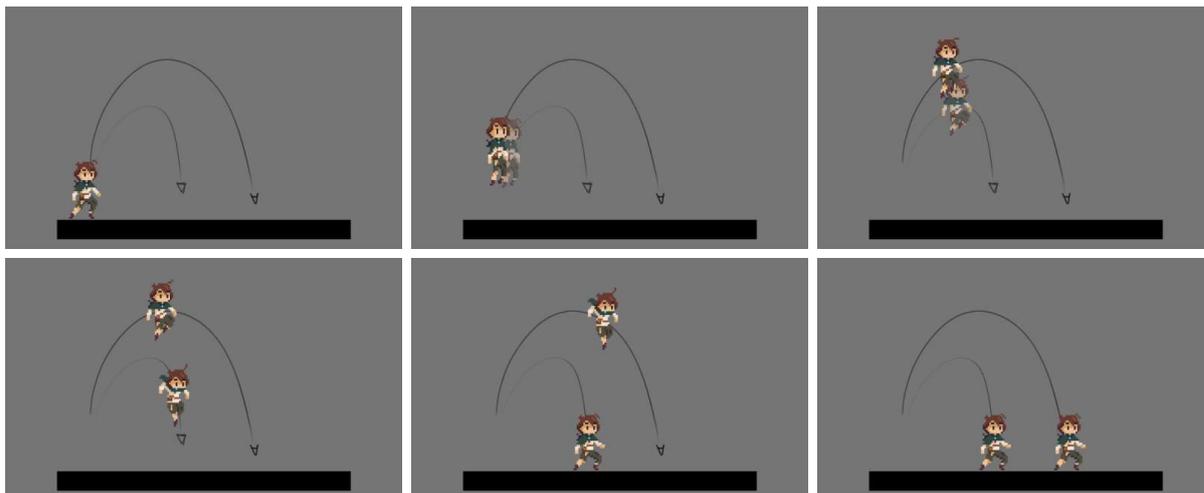


Figura 27. Exemplificação da mecânica de salto.

6.2.2 Regras Gerais

O personagem jogável possui uma quantidade máxima de 6 **pontos de vida**. Os pontos podem ser subtraídos ao encostar em inimigos, *bosses* e blocos de dano ou receber um ataque de inimigos e *bosses*. Ao tomar **dano**, o personagem jogável é **arremessado** 3 *tiles* na direção contrária de onde o dano veio e fica **invulnerável** por 1 segundo mas o jogador perder o controle do personagem por 0.5 segundos. A invulnerabilidade permite que o jogador tenha um momento para se reposicionar em uma posição mais favorável. A circunstância em que o jogador é arremessado ao tomar dano torna principalmente as situações de combate mais interessantes, fazendo com que o jogador avalie bem os seus arredores para evitar ser encurralado ou jogado em algum bloco de dano.

Se os pontos de vida do jogador atingirem um valor igual ou menor que zero, o personagem jogável **morre**, se isso acontecer a tela escurece e o jogador retorna para o último *checkpoint*, perdendo o progresso que ele fez depois do *checkpoint*. **Checkpoints** são locais espalhados pelo mapa nos quais o jogo salva o progresso feito pelo jogador até aquele momento. Além disso, caso os pontos de vida do jogador não estejam na quantidade máxima, o *checkpoint* restaura os pontos perdidos, sendo a única maneira (além de morrer) de recuperar os pontos de vida do personagem.

Além dos pontos de vida, posteriormente o jogador adquire **pontos de magia**, que são necessários para a execução de habilidades especiais adquiridas durante a narrativa e exploração do jogo. A quantidade máxima de pontos de magia inicial é de 100 pontos, e, após serem gastos,

eles recarregam automaticamente, sendo 10 pontos por segundo. Além disso, o jogador possui a tecla “A” para interagir com NPCs, coletar itens e interagir com o cenário em momentos específicos.

6.2.3 Combate

Durante o jogo o jogador vai encontrar momento que ele precisará entrar em combate, para isso ele possui, desde o início do jogo, um *combo*²⁶ com 2 ataques físicos e, ao longo do jogo, ele poderá desbloquear as 3 habilidades especiais, sendo elas Ataque mágico, Defesa mágica e Ataque carregado.

6.2.3.1 Ataques físicos

Os Ataques físicos são movimentos de combate simples e de curto alcance que o jogador possui acesso desde o início do jogo pressionando a tecla “D”. Por isso, eles não possuem restrição de uso. São dois ataques, um seguido por outro, que podem ser utilizados em formato de *combo* ou não. Como mostrado na Figura 28.

²⁶ **Combo** ou “sequência de golpes”, é uma técnica muito utilizada em jogos que possuem combate. Ela consiste em realizar e conectar uma série de movimentos específicos de forma rápida e precisa, resultando em uma combinação de golpes que causam dano significativo ao oponente ou inimigo.

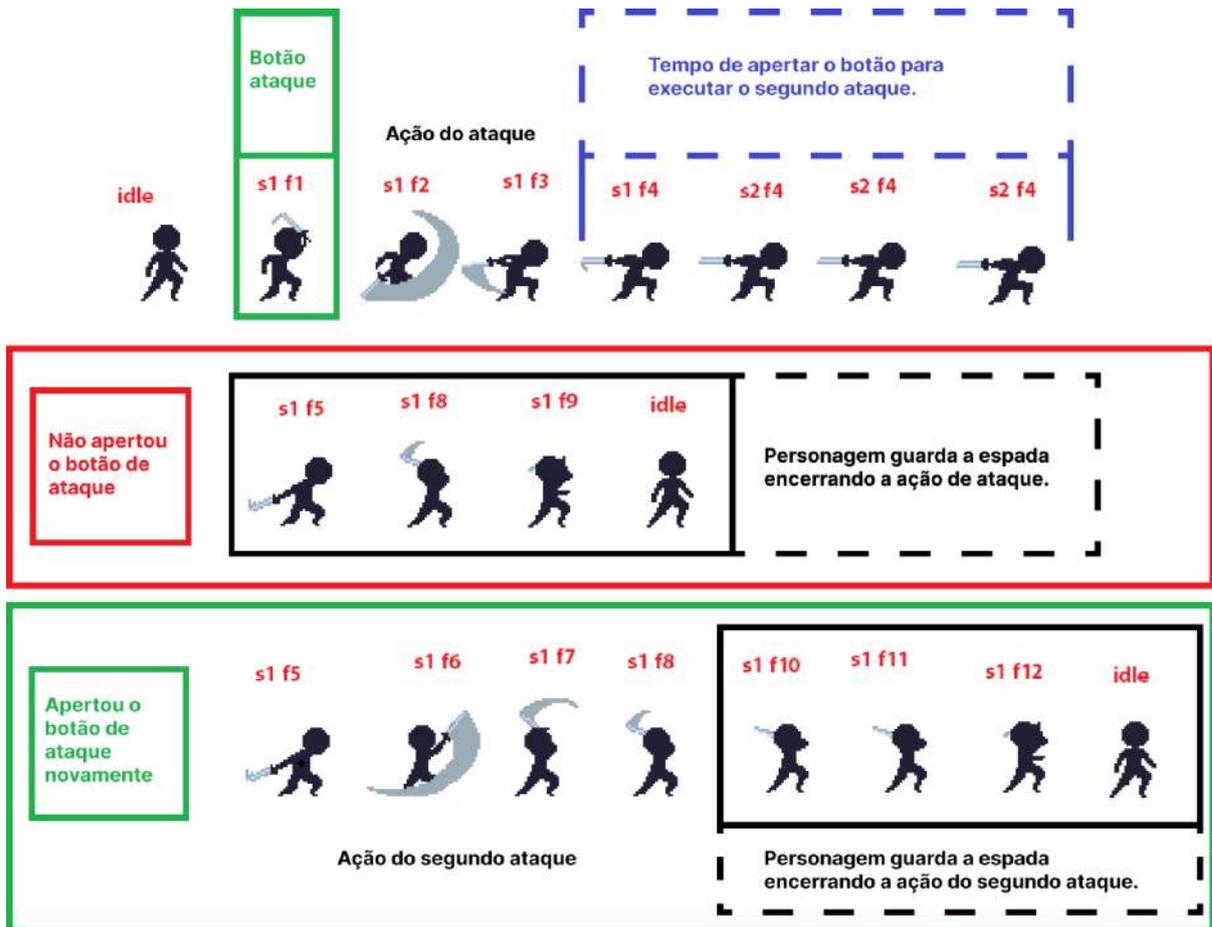


Figura 28. Mecânica de ataque físico.

O jogador aperta o botão de ataque e o personagem executa o ataque imediatamente, mas após a finalização do primeiro ataque o jogador tem uma janela de quatro *frames*²⁷ para apertar o botão de ataque novamente e finalizar o combo, caso o jogador não aperte o botão de ataque novamente, o *combo* é cancelado e ele retoma controle do personagem.

O dano do ataque físico é de 1 ponto de vida por *hit*, fazendo com que o combo completo possa infringir um dano de até 2 pontos de vida, ao executar o primeiro ataque, o personagem jogável se desloca 16 pixels na direção em que ele está virado para um alcance maior, porém, durante o restante da ação de ataque, o personagem jogável fica imóvel. Fica a critério do jogador executar somente o primeiro ataque ou pressionar o botão de ataque novamente, executando o segundo ataque e completando o combo.

²⁷ *Frame* ou quadro é uma única imagem da sequência de imagens que compõem uma animação.

6.2.3.2 Habilidade especiais

Habilidades especiais são movimentos que o jogador pode adquirir como parte essencial do jogo ou recompensa por exploração do mapa. Além da utilização em combate, essas habilidades podem ter mecânicas atreladas de exploração, solução de *puzzle*²⁸, abertura de passagens e *shortcuts*²⁹, etc. Essas técnicas serão descritas junto com a habilidade especial, quando for o caso. Todas as habilidades especiais gastam pontos de magia para serem ativadas.

A **Defesa mágica** é uma habilidade adquirida que, ao pressionar e segurar o botão de ativação, cria um escudo mágico em volta do personagem (Figura 29). O escudo absorve 1 *hit* de dano de qualquer ataque físico ou mágico. Ele se mantém ativado enquanto o jogador mantiver o botão de ativação pressionado ou até o jogador receber dano. A habilidade tem um custo de 30 pontos de magia para ser ativada.

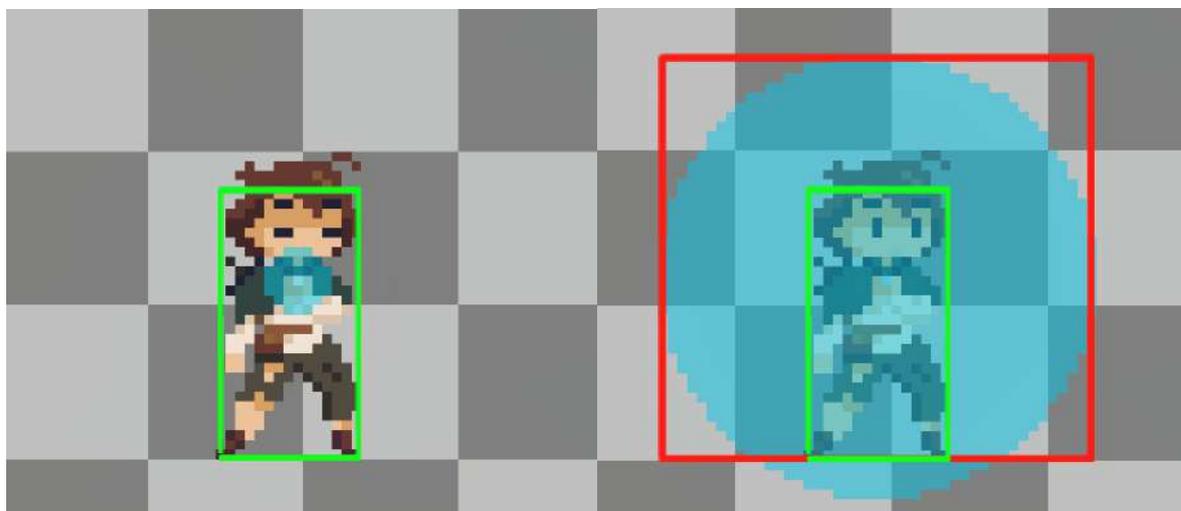


Figura 29. Mecânica da habilidade especial Defesa Mágica.

O **Ataque mágico** é uma habilidade adquirida que, ao pressionar o botão de ativação, invoca o espírito da raposa, que executa uma mordida, infringindo um dano de 4 pontos de vida por *hit* em inimigos próximos e em uma área extensa à frente do personagem (Figura 30). O custo de ativação da habilidade é de 80 pontos de magia. Além do uso em combate, a habilidade

²⁸ *Puzzle* ou quebra-cabeças são desafios que não possuem uma solução imediata, exigindo que o jogador pare e reflita em uma forma de resolvê-lo. Eles normalmente são utilizados como mecânicas para adicionar obstáculos (conflitos), os quais tornam o jogo mais interessante e demandam uma ação calculada do jogador.

²⁹ *Shortcuts* ou atalhos, são caminhos que conectam mapas previamente distantes e o jogador pode liberar ou encontrar para se mover de forma mais rápida e eficiente pelo mapa.

também é necessária para destruir barreiras mágicas espalhadas pelo mapa, liberando o acesso a áreas previamente inacessíveis.

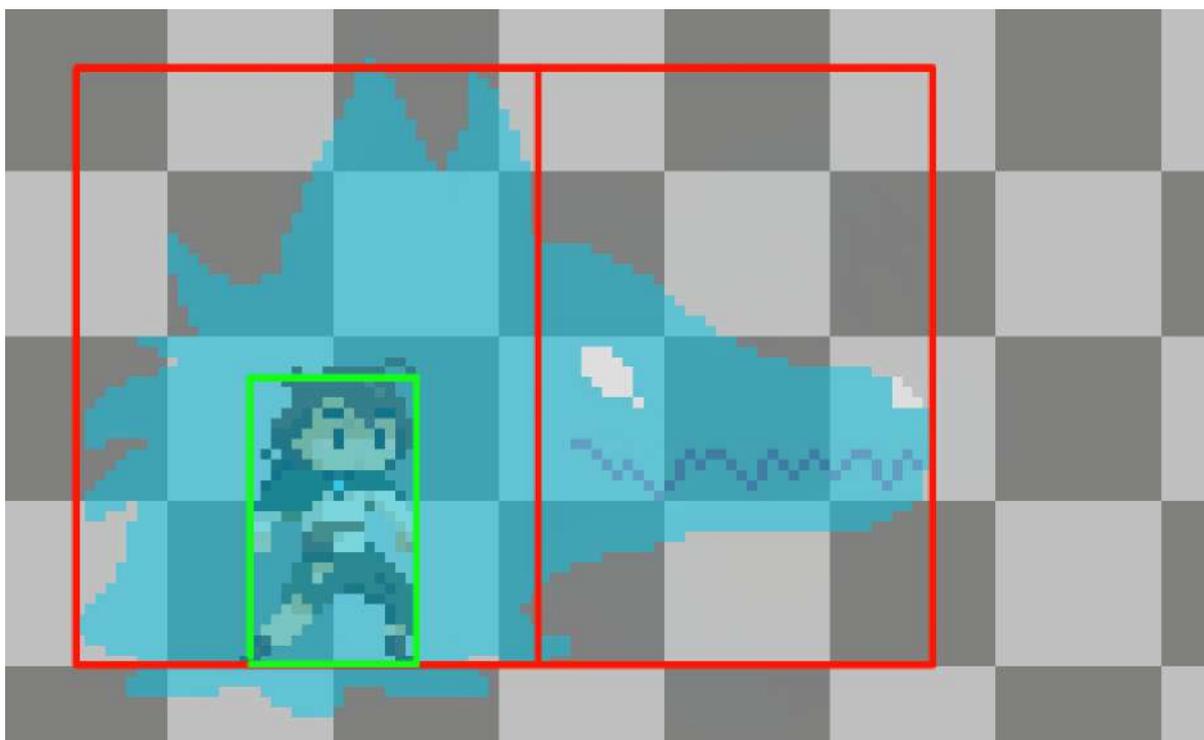


Figura 30. Mecânica da habilidade especial Ataque Mágico.

O **Ataque carregado** também é uma habilidade adquirida. Ao pressionar e segurar o botão de ataque físico por 3 segundos, um ataque mais forte é carregado, ao soltar o botão o ataque, é lançado com um dano de 3 pontos de vida por *hit* em inimigos em uma área considerável a frente do personagem jogável (Figura 31). Ele consome 50 pontos de magia por uso e também pode ser utilizado para empurrar determinados objetos no mapa, liberando passagens e áreas previamente inacessíveis.



Figura 31. Mecânica da habilidade especial Ataque Carregado.

6.3 Regras e Mecânicas de *Companion* e NPCs

Neste capítulo, vamos abordar todo o conjunto de regras e mecânicas relacionadas aos *companions* e NPCs presentes no Melius. *Companion* se refere a um personagem no jogo sobre qual o jogador tem controle limitado. Esse personagem geralmente acompanha o jogador durante o *gameplay*, podendo ajudar no combate e no desenvolvimento da história.

NPC é uma sigla para “*non-playable character*” ou, em português, “personagem não jogável”, e se refere a todos os personagens presentes no jogo sobre os quais o jogador não tem controle direto ou indireto. NPCs podem ou não ser interativos, geralmente utilizados para preencher locais que na história seriam populados. Podem também possuir diálogos e ser utilizados como maneira de exposição de história para o jogador, passar informações e até missões³⁰ para o jogador.

6.3.1 *Companions*

A *companion* presente no Melius é a **Raposa**: Definida por nós como entidade espiritual ligada à água e a vida. Atua em conjunto com o personagem principal, ao interagir com ela pela primeira vez, é liberado ao jogador a barra de mana e a possibilidade de utilizar habilidades especiais, assim, a raposa funciona como catalisador para elas. Visualmente, ela aparece apenas em momentos de utilização de tais habilidades.

³⁰ **Missão** ou *quests*, em jogos, é um termo utilizado para tarefas dadas ou objetivos a se cumprir pelo jogador no jogo.

6.3.2 NPCs

O Melius possui dois NPCs, a **Mãe do Protagonista**, personagem presente apenas no primeiro mapa ao lado do protagonista. Por estar doente na história, ela funciona como motivação inicial para o início do jogo. É um npc intangível que apenas possui diálogo como mecânica: ao apertar o botão de interação, aparecem balões de fala acima dos personagens mostrando em texto o que eles estão dizendo.

O segundo NPC é o **Alquimista**. Ele o primeiro objetivo que o jogador precisa alcançar, é um NPC intangível, possui balões de fala e está presente em uma área fixa no mapa. A interação com ele ativará o próximo objetivo para o jogador: coletar 4 ingredientes para a poção de cura na cidade subterrânea, além de liberar a entrada para ela.

6.4 Regras e Mecânicas de Inimigos e Bosses

O jogo possui um total de 3 inimigos únicos sendo eles **Rastejador**, **Infectado** e **Corredor**, além de um **Boss** final.

6.4.1 Inimigos

Rastejador é inimigo com uma inteligência artificial simples que não reage de maneira hostil à presença do jogador e que possui um trajeto pré-definido, sendo capaz de se movimentar apenas no eixo X para a esquerda e direita em uma velocidade fixa. Possui 2 pontos de vida e infringe 1 ponto de dano ao encostar no personagem jogável. Ele funciona mais como um obstáculo ambulante do que uma ameaça direta ao jogador.

O **Infectado** humanoide possui uma inteligência artificial mais complexa, mais hostil a partir de certa distância, é capaz de identificar, perseguir e atacar o jogador. Pode se movimentar livremente no eixo X para a esquerda e direita em uma velocidade fixa. Possui 3 pontos de vida e seu ataque infringe 1 ponto de dano ao personagem jogável. Ele é uma ameaça direta ao jogador, e só para de persegui-lo ao morrer ou perder o jogador de vista. O inimigo **Corredor** possui as mesmas características, porém, sua velocidade de movimentação é variável, sendo capaz de lançar uma disparada em direção ao personagem jogável para executar um ataque.

6.4.2 Bosses

Alquimista transformado é o único *boss* e também o *boss* final, possuindo uma inteligência artificial mais complexa e extremamente hostil ao jogador. Ele é capaz de identificar/localizar o jogador em todo o mapa que ele está presente, além de o perseguir e atacar de maneira mais persistente. Pode se movimentar livremente no eixo X para a esquerda e direita em uma velocidade fixa.

Possui 40 pontos de vida, 2 ataques físicos que, assim como os do jogador, podem formar um combo ou não, infringindo 1 ponto de dano por ataque, um ataque carregado que, também, assim como o do jogador, após carregar por alguns segundo tem um alcance maior e infringe 2 pontos de dano, possuindo um *cooldown*, ou tempo de espera para que ele possa usá-lo novamente, de 10 segundos.

6.5 Modificações propostas as Regras e Mecânicas no primeiro ciclo de revisão

As revisões em regras e mecânicas começaram por revisar e discutir todas as previamente decididas e implementadas, trouxemos de volta todos os *feedbacks*³¹ coletados nos períodos de teste com o público, além de que, cada membro da equipe, após poder testar o jogo mais extensivamente, trouxe seus próprios comentários e *feedbacks* para as discussões. Passamos por cada uma das questões, validando-as e discutindo as possíveis correções, também usamos a oportunidade para *brainstorming* de novas ideias para expandir o jogo, seguindo um processo muito semelhante ao adotado anteriormente.

6.5.1 Revisões gerais e de mapa

Para as regras e mecânicas gerais e de mapas, mantivemos boa parte das previamente estabelecidas, fazendo alteração em duas mecânicas e adicionando apenas uma. A mecânica alterada foi a dos *checkpoints*, que agora funcionarão como salas seguras. Todas as mecânicas

³¹ *Feedback* é uma palavra que vem do inglês, e se refere ao ato de fornecer sugestões, informações, opiniões e avaliações sobre o desempenho, trabalho ou comportamento de algo ou alguém, de forma a permitir melhorias.

associadas a eles permanecerão, mas agora cada *checkpoint* será uma fase isolada do resto do mapa, sem a presença de inimigos. Esses espaços servirão para contemplação, permitindo que o jogador recupere vida e interaja com a raposa, ressaltando o vínculo que se desenvolverá entre eles ao longo do jogo.

Outra mecânica alterada foi a de **empurrar objetos** previamente estabelecidos, previamente ela estava totalmente atrelada a habilidade especial Ataque carregado, porém essa mecânica foi dividida. O jogador pode empurrar pequenos objetos com o personagem jogável desde o início do jogo, sem precisar de uma habilidade especial para isso. A habilidade de empurrar objetos médios e grandes foi atrelada a uma nova habilidade especial.

Outra mecânica adicionada é a dos **elevadores**. Eles funcionam essencialmente como um teletransporte, levando o jogador de um ponto a outro do mapa instantaneamente, oferecendo uma alternativa para navegação e exploração, que irão ser posicionados em pontos de interesse do jogador funcionando como atalhos. Após serem ativados, os elevadores estarão disponíveis para uso livre pelo jogador.

6.5.2 Revisões ao personagem jogável

A primeira mudança feita ao personagem jogável é que ao levar **dano** de inimigos ou *bosses*, além de ser arremessado para trás, ele fica **invulnerável** por 2 segundos, já ao levar dano de **blocos de dano**, ele retorna ao início da sala onde ele está naquele momento. Percebemos que a janela de invulnerabilidade anterior era muito curta, não sendo suficiente para o jogador se recuperar ao sofrer dano, já a mudança ao receber dano de blocos de dano se mostrou necessária pois em diversas regiões do mapa onde estes blocos foram posicionados se tornaram *death traps*, ou armadilhas de morte, na qual ao cair em uma o jogador tomava diversos *hits* de dano e muitas vezes morria antes de conseguir se recuperar.

Para o combate base, percebemos que ao deixar o personagem imóvel ao executar a ação de **ataque físico** se tornou uma mecânica não intencionalmente punitiva ao jogador, que possui uma janela de tempo muito curta para executar o ataque sem ser punido, isso resultou em o jogador sofrer dano constante ao executar um ataque antecipado ou atrasado. Então para flexibilizar o combate e dar mais controle ao jogador, removemos a mecânica na qual o jogador executa o ataque o personagem jogável avança alguns *pixels*, e passamos a deixar com que o

jogador possa mover o personagem livremente enquanto a ação de ataque físico é executada, não podendo apenas mudar a direção do ataque no meio da ação.

Outra importante mecânica ligada ao *companion* que foi modificada está relacionada ao combate do jogador. Os ataques do personagem jogável causam dano aos inimigos apenas quando a raposa está no estado “junto ao jogador”, ou seja, acompanhando o jogador. Quando a raposa está em estado “separado do jogador”, seus ataques tem alcance menor e apenas empurram os inimigos para longe, sem causar dano. O *companion* só entra no estado “separado do jogador” ao executar ações específicas, como a habilidade especial ataque carregado, que mantém a raposa nesse estado por dois segundos. Após o término do estado “separado do jogador”, a raposa se teleporta imediatamente para o *tile* à esquerda do personagem jogável. Isso também ocorre se a raposa ficar presa ou se afastar muito do jogador.

Portanto, o ataque físico só causaria **dano** em inimigos quando o personagem jogável estivesse junto com o *companion*, sem ela os ataques tem alcance de 14 *pixels* menor e apenas empurram inimigos pequenos e médios para longe, sem causar dano. Isto seria representado visualmente com o rastro da espada sendo diferente quando o jogador está com o *companion* e estado “junto ao jogador” ou “separado do jogador” (Figura 32).



Figura 32. Revisão da mecânica do ataque físico.

6.5.3 Revisões e novas habilidades especiais

As habilidades especiais foram aprimoradas durante os testes realizados após a conclusão da primeira versão do jogo. Observamos diversos pontos onde elas poderiam ser melhoradas e aproveitamos a oportunidade para discutir a introdução de possíveis novas habilidades no jogo.

O **Ataque Mágico** manteve grande parte de suas características, mas passou a ser denominado "**Mordida da Raposa**". A mudança no nome foi realizada para que ele faça referência ao ataque, ao invés de utilizar um nome genérico como o anterior. Mecanicamente, a habilidade sofreu algumas alterações: a primeira é que agora o jogador pode utilizá-la no ar, durante um salto, e não somente no chão, como era feito anteriormente, a segunda é que a *hitbox* foi movido diretamente para frente do personagem jogável, e não o cobrindo como anteriormente (Figura 33), o custo de magia para a ativação passou a ser de 40 pontos, e por fim introduzimos uma nova mecânica que ao utilizar a Mordida da Raposa em combate faz com que o *companion* fique em estado “separado do jogador” por dois segundos.

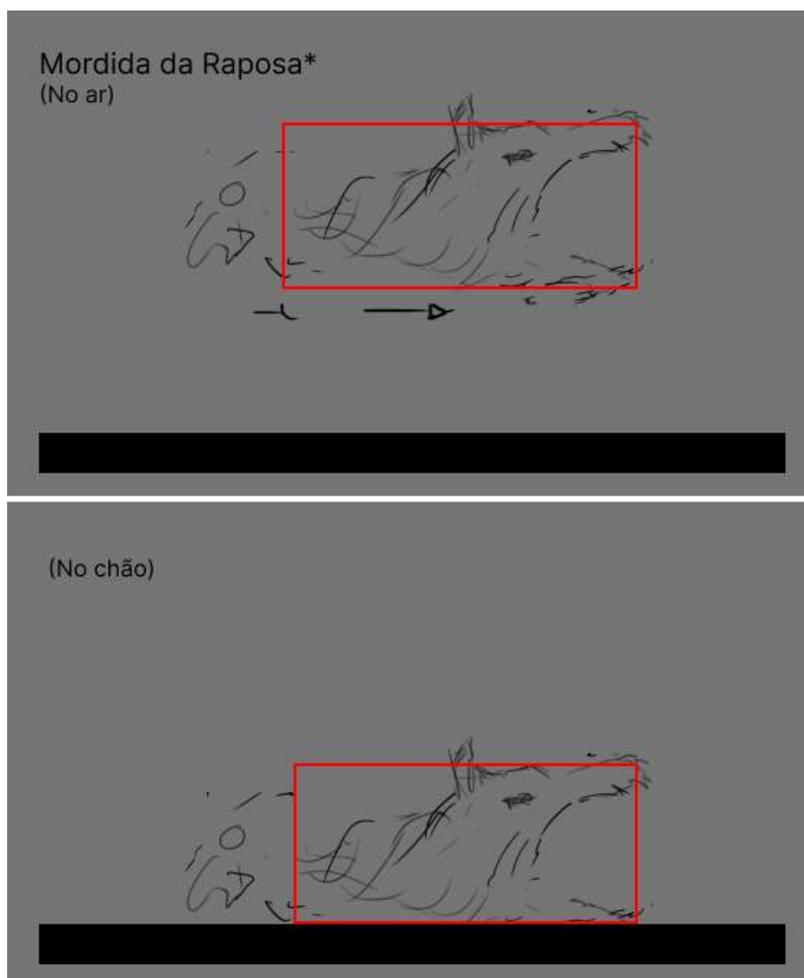


Figura 33. Mecânica da habilidade especial Mordida da Raposa.

O **Ataque Carregado** foi identificado como uma habilidade muito difícil de ser utilizada e pouco viável para combate, devido ao seu tempo de carregamento, que deixava o personagem jogável imóvel e vulnerável, além de seu curto alcance. Portanto, a habilidade foi revisada e modificada quase que completamente.

Agora, o jogador pode se movimentar lentamente e alterar a direção do Ataque Carregado enquanto ele está sendo carregado. Ao soltar o botão de carregamento, o ataque é lançado para frente, com um alcance significativamente maior de até 14 *tiles* à frente e a capacidade de atingir múltiplos inimigos (Figura 34). Ao utilizar o Ataque Carregado em combate, o companheiro fica em "estado separado" do jogador por quatro segundos.

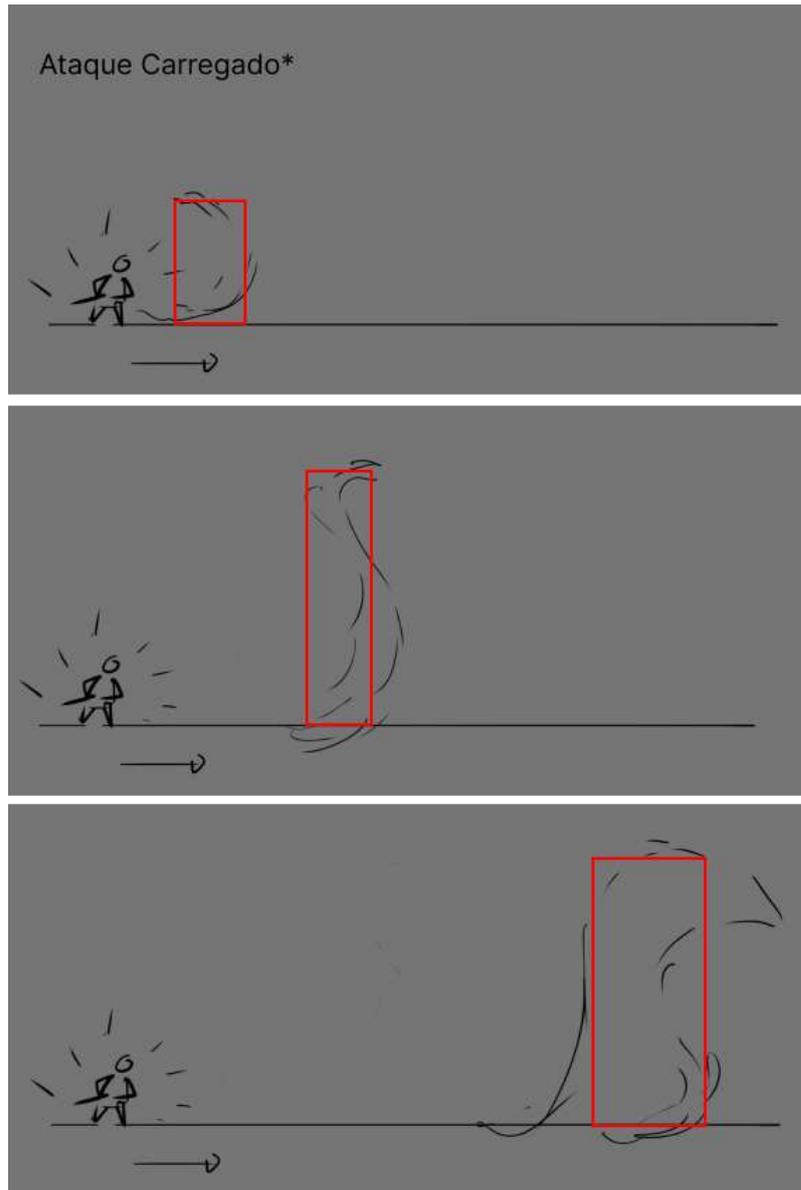


Figura 34. Revisão da mecânica da habilidade especial Ataque Carregado.

Além disso, o Ataque Carregado deixou de ser utilizado para mover caixas ou determinados objetos, passando a ser usado para liberar caminhos obstruídos por rochas ou destruir paredes rachadas.

Para novas mecânicas desenvolvidas pela equipe, temos um total de quatro novas habilidades especiais, sendo elas o Dash Tático, Empurrão, Dive e Erupção. O **Dash Tático** é uma habilidade especial de movimentação que permite ao jogador, tanto no chão quanto no meio de um salto, realizar um salto curto para frente em formato de arco. Essa habilidade pode ser utilizada para alcançar regiões inacessíveis apenas com um salto (Figura 35) ou para desviar

de ataques lançados em sua direção e passar por inimigos, já que torna o personagem jogável invulnerável durante sua execução da ação (Figura 36). A habilidade não faz com que o *companion* entre em estado “separado do jogador” e não gasta pontos de magia.

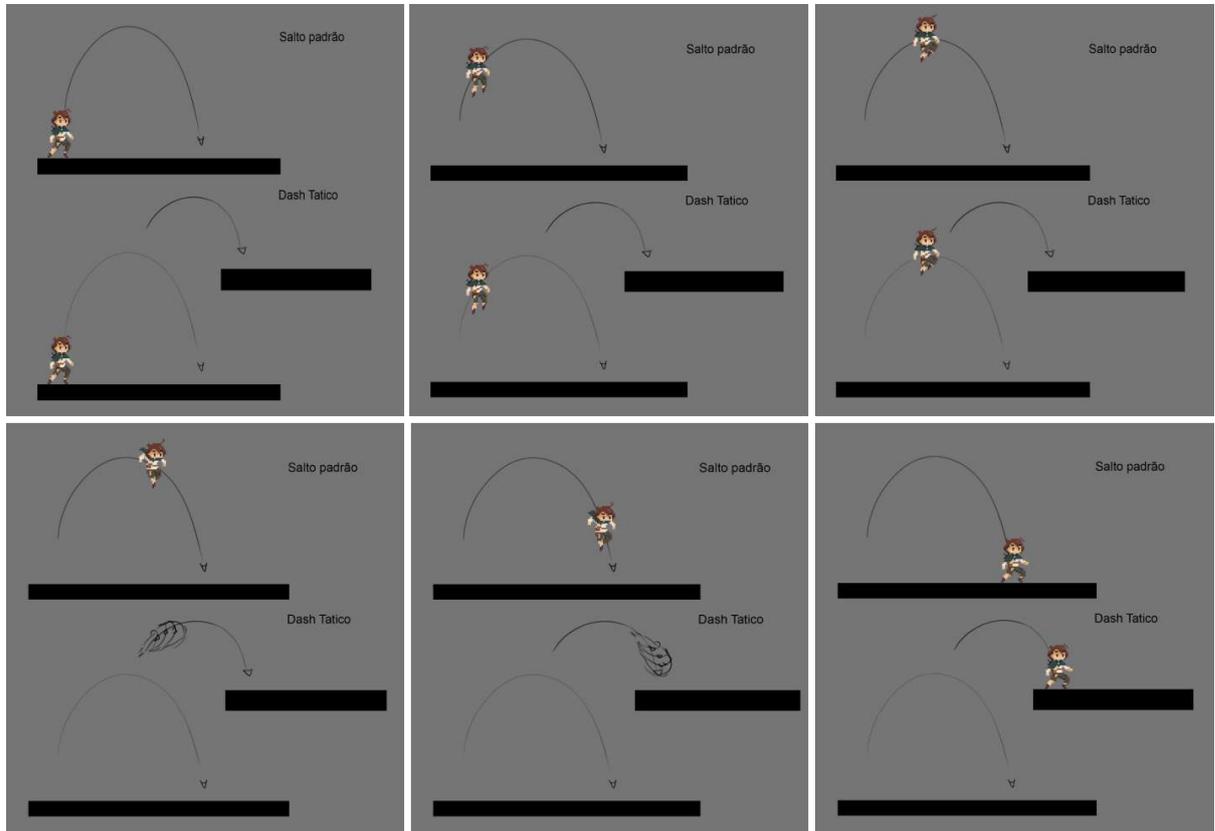


Figura 35. Mecânica da habilidade especial Dash Tático.

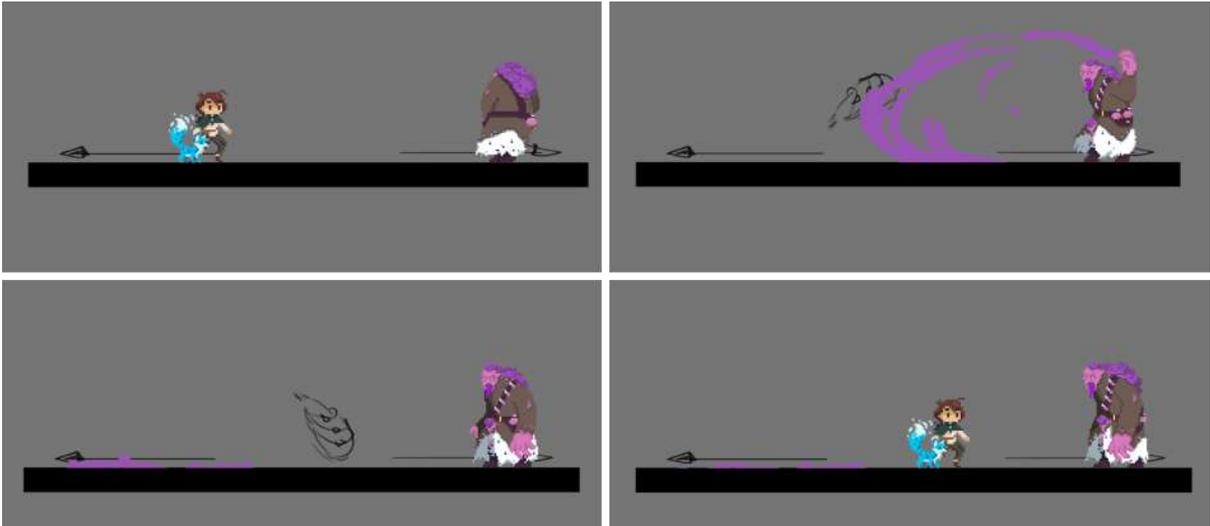


Figura 36. Dash Tático para evitar ataques.

O **Empurrão** é uma nova habilidade especial na qual, ao apertar o botão de ação, o personagem jogável pisa no chão e uma pequena onda se forma à sua frente, empurrando inimigos pequenos e grandes e causando um dano de 1 ponto de vida. Ativar a habilidade custa 30 pontos de magia, só pode ser usado no chão e não faz com que o companheiro entre em "estado separado" do jogador. Além disso, pode ser utilizado para empurrar grandes objetos e resolver *puzzles* específicos e liberar novas passagens e áreas do mapa (Figura 37).

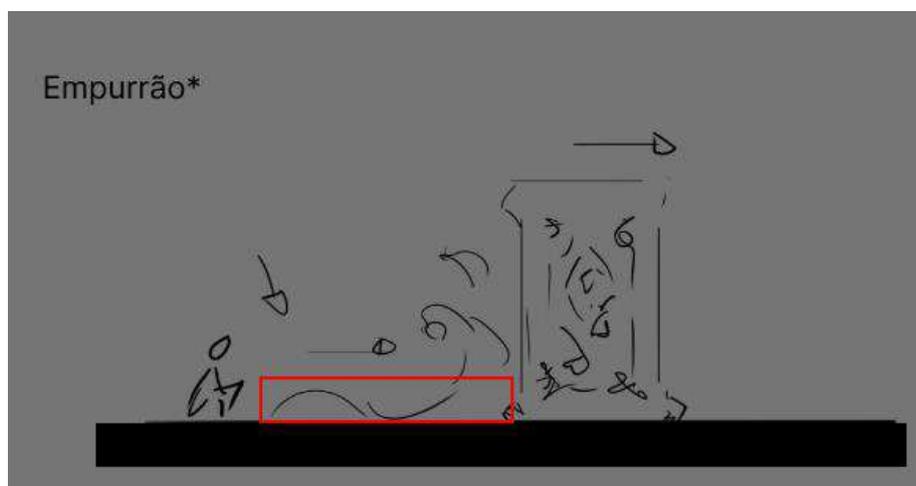


Figura 37. Mecânica da habilidade especial Empurrão.

O **Dive** é uma habilidade em que, ao acionar o botão de ação, o *companion* é lançado para cima em um movimento de arco até atingir um bloco de colisão ou a altura máxima determinada. Ao alcançar esse estado, ela executa uma queda em alta velocidade, causando dano no caminho durante a queda e novamente ao tocar o chão. Durante a queda, o dano causado é de 1 ponto de vida e ao tocar o chão o dano causado seria de de 2 pontos de vida, a habilidade custa 60 pontos de magia para ser ativada (Figura 38). Ao executar essa ação, o companheiro fica em "estado separado" do jogador por quatro segundos.

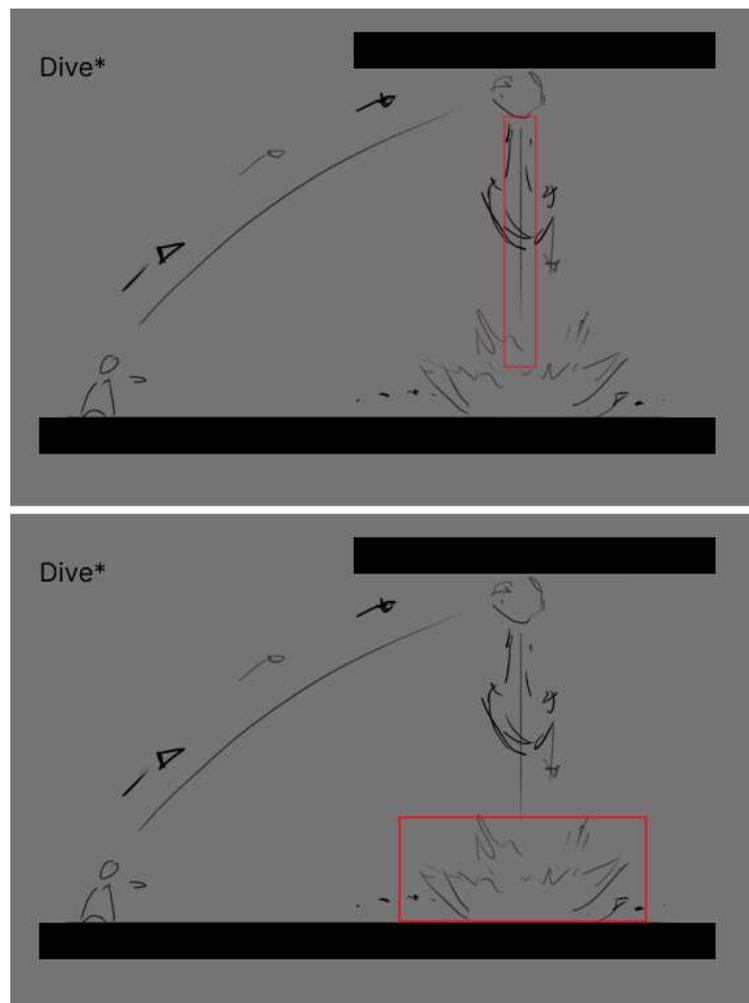


Figura 38. Mecânica da habilidade especial Dive.

A **Erupção** é uma habilidade de alto dano em área. Ao apertar o botão de ação, uma enorme cabeça de raposa emerge do chão ao redor do personagem jogável, causando dano em área a todos os inimigos ou *bosses* presentes na região marcada em vermelho. Cada *hitbox* causaria dano de 1 ponto de vida (Figura 39). Executar essa ação faz com que o companheiro fique em "estado separado" do jogador por quatro segundos.

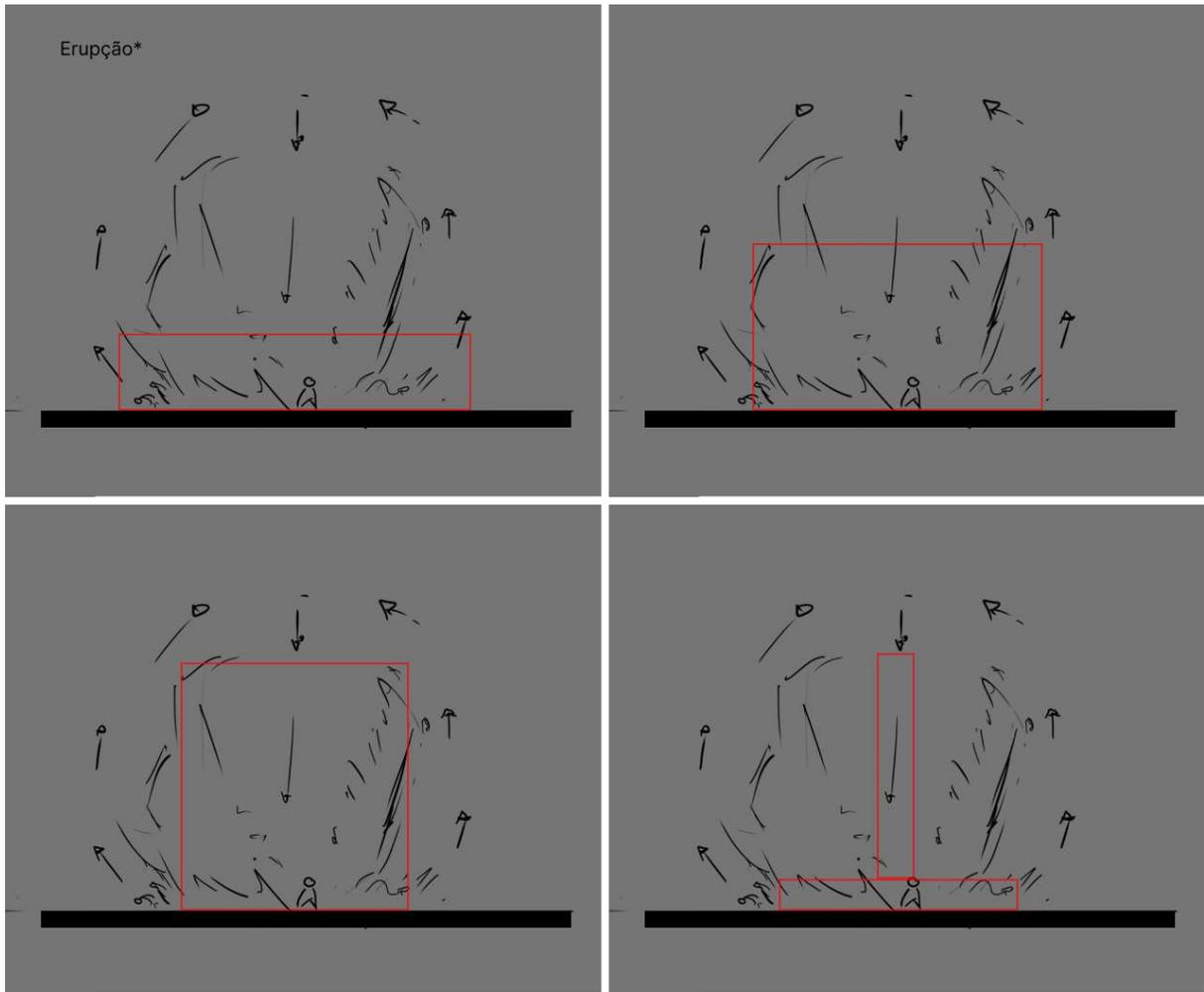


Figura 39. da habilidade especial Erupção.

6.5.4 Revisões ao *companion*

Com a conclusão da disciplina de Design de Jogos e da primeira versão do jogo, um dos principais objetivos da equipe para os ciclos de revisão é de transformar a raposa em um *companion* funcional. O objetivo é que ela seja capaz de acompanhar o jogador ao seu lado durante todo o decorrer do jogo, em vez de ser um elemento mais narrativo do que mecânico, como ficou na primeira versão do jogo.

Para alcançar isso, desenvolvemos um *companion* que pudesse andar, correr e pular de forma independente do jogador, para ser capaz de navegar pelo mapa do jogo. Além disso, a inteligência artificial dela é capaz de reconhecer a posição do personagem jogável e calcular todos os caminhos possíveis dentro da sala, identificando o caminho mais curto até esse ponto (Figura 40). Dessa forma, o *companion* será capaz de acompanhar o jogador enquanto ele explora o mapa.



Figura 40. Mecânica de cálculo de caminho do *companion*.

Também foram realizados ajustes para tornar a movimentação da raposa mais natural. Por exemplo, a mecânica que determina o quão perto ou longe a raposa deve estar do personagem jogável é maleável entre 2 à 5 *tiles*, ou seja, permite que a raposa esteja mais próxima em alguns momentos e mais distante em outros (Figura 41). Esse ajuste também se aplica à distância a partir da qual a raposa começa a seguir o personagem jogável depois que ele fica parado por alguns instantes.

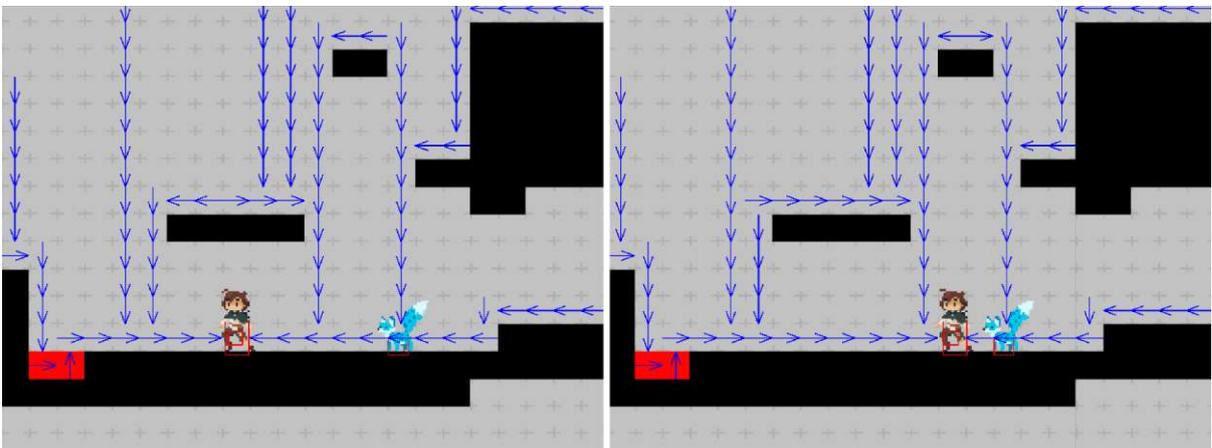


Figura 41. Mecânica de distância do *companion* do jogador

Além disso, a raposa é capaz de pular de qualquer altura e não possui pontos de vida, o que significa que ela não morre ao se aproximar de inimigos, *bosses* ou blocos de dano. A decisão de torná-la invulnerável visa permitir que ela se mova livremente sem que o jogador precise se preocupar em mantê-la viva, especialmente porque o jogador não controla seus movimentos.

6.5.5 Revisão Mecânica de inimigos e *bosses*

Os inimigos normais do jogo não tiveram mudanças nas suas mecânicas, com exceção do *boss* alquimista, que passou por algumas alterações. Antes, ele era o chefe final do jogo, mas agora foi movido para a Cidade Subterrânea. Nesta nova posição, ele bloqueia uma passagem importante que o jogador precisa para obter um item-chave necessário para avançar no jogo.

As mecânicas básicas de combate do *boss* alquimista foram preservadas, apenas sua velocidade de movimentação foi ajustada para torná-lo mais rápido na arena. Esse aumento de velocidade faz com que ele fique mais próximo do jogador alterando a dinâmica do combate, tornando-o mais desafiador.

Adicionalmente, foram introduzidos três novos ataques ao repertório do *boss* alquimista. O primeiro ataque é a **investida** (Figura 42).

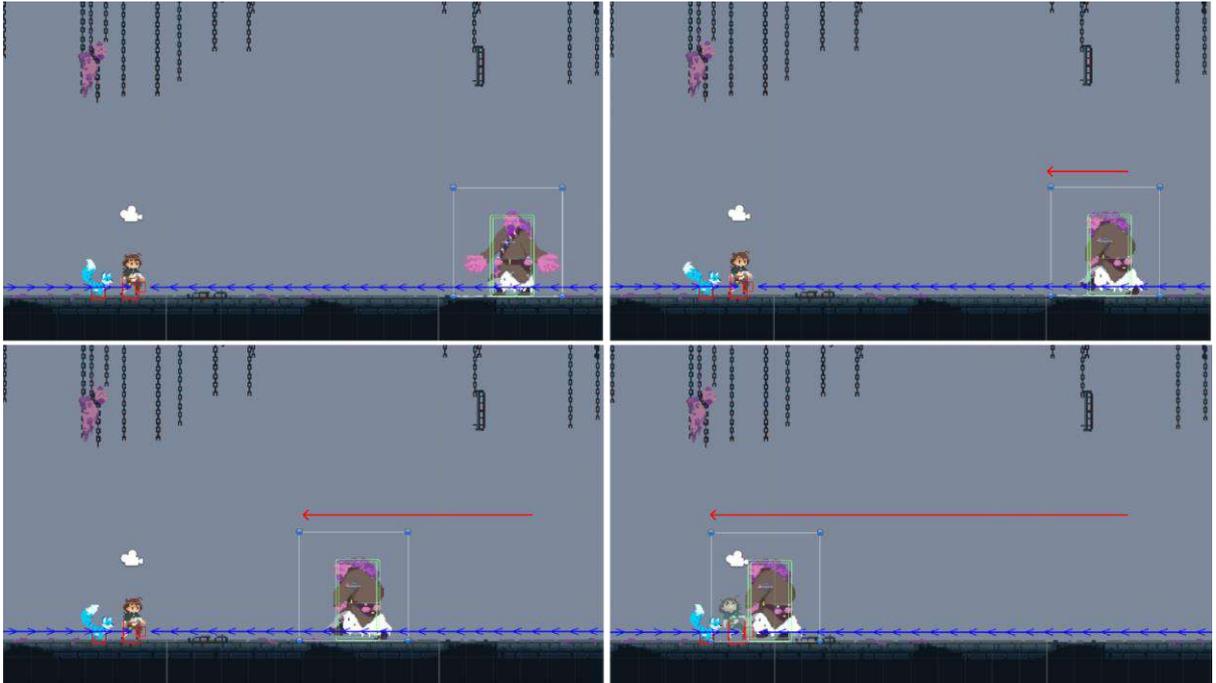


Figura 42. Mecânica de combate do boss: Investida.

Neste ataque o *boss* avança em alta velocidade em direção ao jogador, causando dano de dois pontos de vida se acertar o jogador, o alquimista só para a investida se atingir o jogador ou o alcance máximo do ataque, que é de 20 *tiles* de distância a partir do ponto que ele iniciou o ataque. O segundo é o **arremesso de pote** (Figura 43).

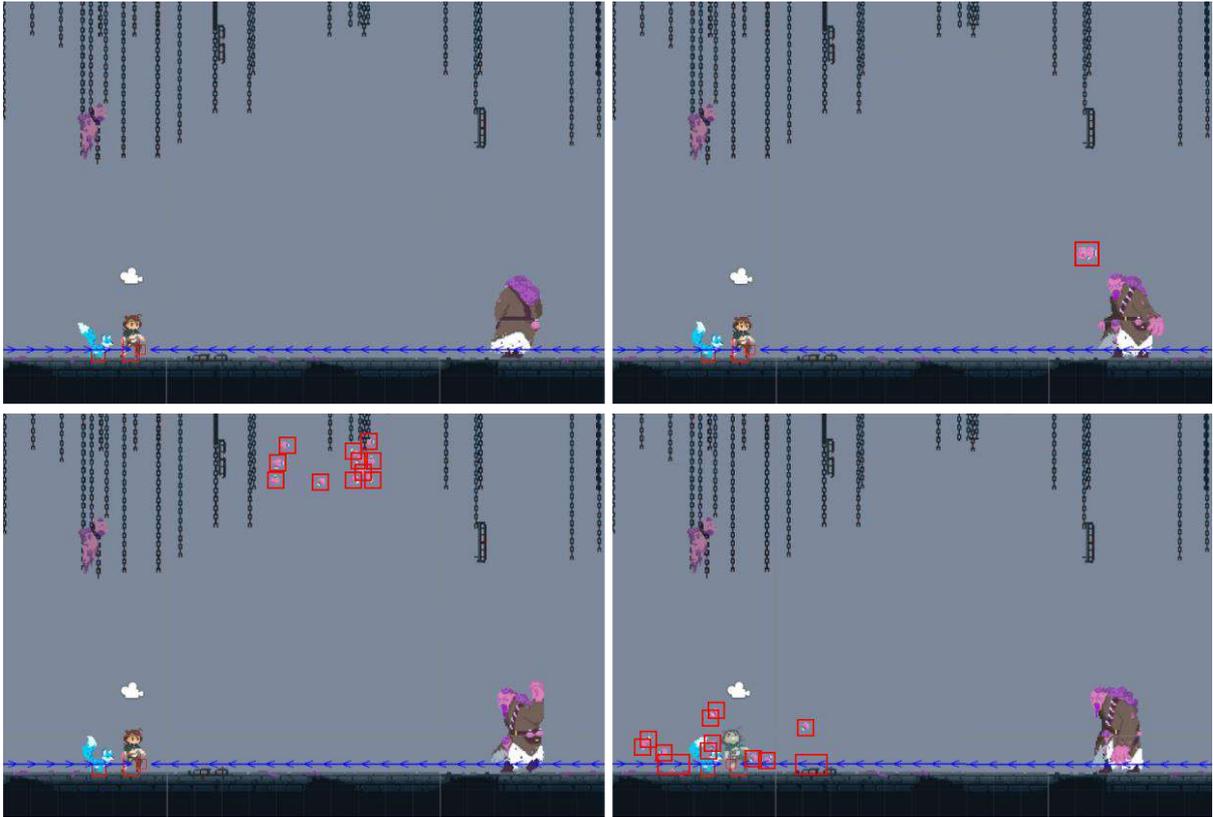


Figura 43. Mecânica de combate do boss: Arremesso de Pote.

Esse ataque consiste no arremesso de múltiplos potes de vidro contendo veneno em direção ao jogador. A quantidade de potes é randomizada e cada um deles possui uma *hitbox* própria que expande ao atingir algo, simbolizando que o vidro se quebrou e espalhou veneno pelo chão. O jogador recebe dois pontos de dano ao ser atingido diretamente por um vidro ou caso se encontre próximo da região onde eles quebraram. E por fim o terceiro novo ataque, o **arremesso de veneno** (Figura 44).



Figura 44. Mecânica de combate do *boss*: Arremesso de Veneno

Este também é um ataque de arremesso, porém, diferente do anterior, ele lança veneno direto na direção do jogador. O ataque também deixa uma poça de veneno no chão que fica presente naquela posição por 30 segundos. É um ataque em área com um alcance fixo de 13 *tiles* causa dois pontos de dano ao jogador se atingido pelo ataque ou se encostar no veneno deixado no chão.

6.6 Considerações sobre as regras e mecânicas

Como dito no início deste capítulo, as regras e mecânicas definem como o jogo irá se comportar com a interação do jogador, por isso buscamos por isso buscamos abranger todos os elementos do jogo nesta etapa. A definição desses elementos não se deu de forma instantânea, mas a partir de sessões de brainstorming e debates, nas quais fomos registrando todos os pontos discutidos. Importante ressaltar que não atingimos o melhor resultado na definição e quantificação logo na primeira tentativa, ao invés disso nós definimos as mecânicas e atribuímos os valores que a partir da implementação e testes elas foram sendo ajustados até atingirmos um resultado satisfatório.

Nem todo o processo ocorreu de forma totalmente tranquila, houveram momentos em que os membros possuíam ideias diferentes de como poderíamos seguir com as mecânicas. Um exemplo disso foi a própria mecânica de combate do protagonista envolvendo a raposa, que em certo momento gerou uma certa divisão na equipe, com todos tendo ideias diferentes de como

isso poderia ser feito. Para que chegássemos a um consenso foram necessários diversos debates e buscas por novas referências de outros jogos, para que aos poucos fossemos unindo as diferentes ideias e estabelecendo uma nova mecânica de combate que deixasse todos satisfeitos.

Embora em parte tenha ocorrido simultaneamente com fases posteriores, essa abordagem facilitou o restante do desenvolvimento. Especificamente no que diz respeito à implementação e na criação de elementos visuais, essa metodologia se revelou eficaz e produtiva.

7. Personagens

Neste capítulo, iremos explorar o processo de criação e desenvolvimento dos personagens, isto é, a análise de suas histórias individuais, o processo de *concepts art*³², a produção dos *assets* e animações para o jogo. Esta fase foi fundamental para a definição mais precisa das identidades dos personagens, incluindo suas personalidades, motivações, características únicas, histórias pessoais, mecânicas de jogo e contribuições para a narrativa. Após essa etapa, conseguimos condensar esses aspectos em um desenho mais coeso e adequado ao contexto do jogo.

7.1 Design de Personagem

Assim que determinamos a trama e o caminho a ser percorrido pelo jogador, iniciamos o processo de aprofundamento dos personagens. Isso incluiu mais reuniões e discussões, nas quais refinamos a história e as particularidades de cada personagem, começando pelo protagonista. Além disso, organizamos os *designers* do grupo em duas equipes: uma focada nos cenários e outra dedicada aos personagens. Com as características dos personagens mais claras, a equipe responsável pelos personagens trabalhou com base nas decisões coletivas para criar uma identidade visual que se encaixasse no jogo, sintetizando todas as características de cada personagem, como, sua história, personalidade, características físicas, etc em uma aparência única que conseguisse transmitir quem ele é, processo que chamamos de Design de Personagens.

7.1.1 Protagonista

Após um debate em grupo, optamos por definir o protagonista como um jovem pré-adolescente de origem humilde, cujas habilidades em luta e caça foram ensinadas por sua mãe. No início do jogo, ele estaria equipado apenas com uma espada e seria capaz de executar ações

³² *Concept art* é um processo essencial na produção de filmes, animações, jogos, quadrinhos e em diversas outras indústrias do entretenimento. Sendo uma etapa da pré-produção em que ideias, descrições e roteiros são expressados e materializados visualmente em diversas alternativas visuais ou *concepts*, que depois são refinados até que seja desenvolvida uma aparência coesa para os personagens, cenários, vestimentas, etc.

básicas como correr, pular e golpear. Conforme avançasse na sua jornada, ao encontrar a raposa e desbloquear seus poderes, ele descobriria novas habilidades. Queríamos transmitir a imagem de um pequeno garoto corajoso enfrentando desafios muito maiores que ele, contando com a ajuda da raposa para alcançar a vitória.

Com essa descrição estabelecida, passamos para a fase de criação de *concepts*. A equipe encarregada pelo *design* de personagens gerou diversas alternativas para a aparência do protagonista (Figura 45), explorando diferentes estilos e detalhes. Utilizamos métodos tradicionais, como lápis e papel, para criar esboços detalhados, além de experimentar *pixel art* no software Aseprite³³, buscando inspiração na protagonista de Momodora (Figura 46). Durante esse processo, focamos principalmente em desenvolver ideias relacionadas a vestimentas, poses e características físicas.

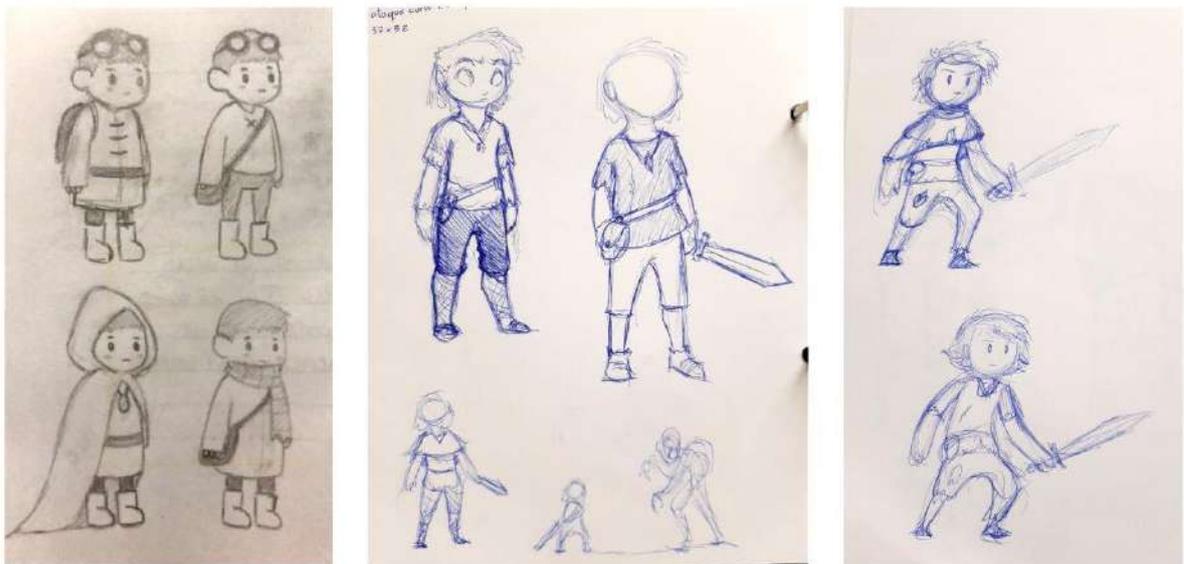


Figura 45. *Concepts* do protagonista.

³³ O **Aseprite** é um *software* de desenho e animação de *pixel art*, voltado principalmente para a criação de *assets* para jogos.



Figura 46. Protagonista de Momodora (2016).

Após essa etapa, com a pose e o estilo mais estabelecidos, transferimos toda a atividade para o ambiente digital, focado na criação do *asset* final do personagem em *pixel art*. Começamos o processo gerando a silhueta do personagem, o que nos permitiu verificar a clareza das proporções e da pose antes de avançar para a fase de detalhamento. Passo a passo, adicionamos todas as características planejadas nos esboços anteriores, experimentando diferentes abordagens para representá-las com *pixels*.

Ao longo desse processo, as cores foram selecionadas e ajustadas de forma orgânica, com inspiração nas referências coletadas anteriormente. Optamos por tons de cores mais quentes e naturais, inserindo um toque de azul no colar do protagonista para destacá-lo e estabelecer uma conexão visual com a raposa. A escolha das vestimentas foi resultado de testes nos quais combinamos vários estilos encontrados em nossas referências. Decidimos, também, por uma postura mais dinâmica e pronta para o combate, refletindo a determinação e a coragem do jovem protagonista, sempre preparado para enfrentar qualquer desafio em sua jornada, especialmente se isso significasse poder novamente reunir-se com sua mãe. Aspectos essenciais da caracterização incluíam ressaltar a origem modesta do personagem. (Figura 47)



Figura 47. Desenvolvimento do *asset* do protagonista.

Com isso chegamos ao *asset* final (Figura 48):



Figura 48. *Asset* final do protagonista.

Para as magias que seriam usadas pelo protagonista, definimos que seria crucial evidenciar a ligação destas com a raposa, visto que não haveria nenhum diálogo no jogo

explicitando essa informação. Para isso, durante o processo de concepção, buscamos trazer diversos aspectos da raposa, desde sua paleta de cores, até a sua própria imagem.

Com isso, criamos duas magias inicialmente, uma voltada para defesa, e outra para ataque. Para defesa, optamos por criar um escudo de água que conteria a imagem da raposa, como se ela rodeasse o jogador a fim de protegê-lo. Já para o ataque, pensamos numa cabeça de raposa, que surge a partir da água envolvendo o protagonista e atacando todos que estiverem à sua frente. Ambas passaram por um processo similar, sendo primeiro esboçadas no papel (Figura 49) e depois trabalhadas em *pixel art* e refinadas já na etapa de animação (Figura 50).

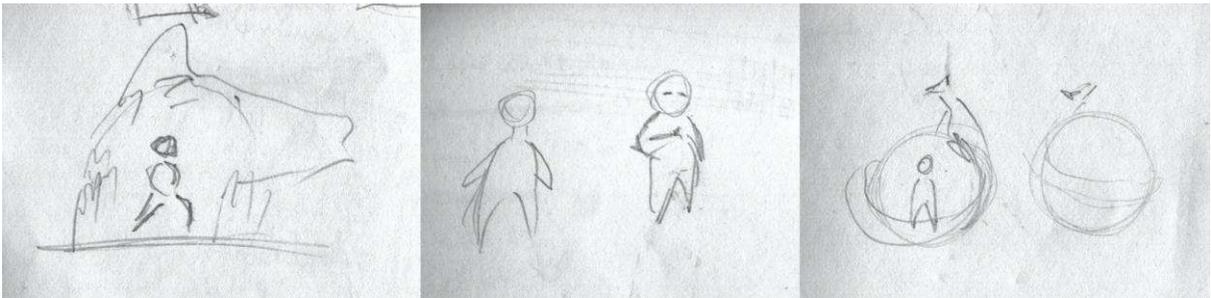


Figura 49. *Concepts* das magias do protagonista.

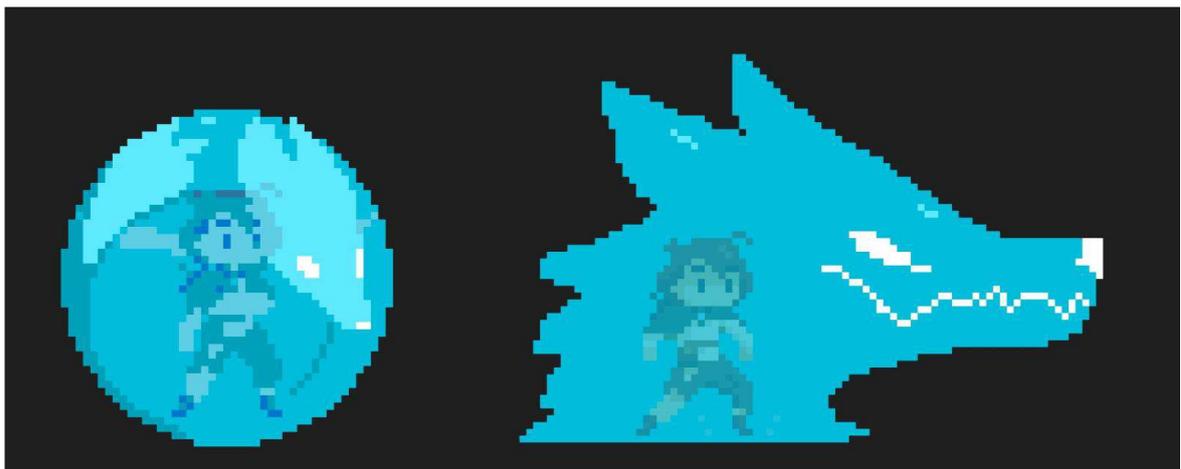


Figura 50. Desenho final das magias do protagonista.

7.1.2 Raposa

Todos os personagens, incluindo o protagonista, foram submetidos a um processo similar de desenvolvimento, que envolveu a elaboração detalhada de suas histórias, a criação de conceitos visuais e a produção dos recursos gráficos necessários. A raposa, representando

um espírito ancestral da água, ocupava um papel central na trama, sendo venerada pelos antigos habitantes da cidade subterrânea por sua capacidade de trazer vida e prosperidade. Apesar da queda dessa civilização, a raposa permaneceu, encontrando o protagonista em uma situação de perigo e tornando-se sua aliada, emprestando-lhe seus poderes.

O estilo da raposa visava transmitir uma essência mágica e benevolente, destacando sua natureza especial e não convencional. Para concretizar essa ideia, iniciamos a fase de *concepts*, começando com esboços tradicionais em papel (Figura 51) e, em seguida, migrando para o Aseprite. Nesse processo, exploramos diferentes representações visuais, buscando capturar sua magia e espiritualidade. A ênfase recaiu sobre sua pequenez, aura mágica e aspecto benevolente, reforçando sua importância na narrativa e sua ligação especial com o protagonista.



Figura 51. *Concepts* da raposa.

Optamos por uma seleção de cores centrada em tons de azul e branco, frequentemente associados à magia em jogos, para criar um contraste vívido com os tons de rosa e roxo que simbolizam a doença, o principal elemento antagonista do nosso jogo, cuja a escolha da cor será melhor explicada no capítulo de Mapas e Cenários. Além disso, decidimos

que a raposa manteria sua cauda erguida quando parada, e que brilharia e liberaria partículas ao usar suas habilidades mágicas, uma característica única que a distinguiria das raposas comuns. A pelagem da raposa foi adornada com várias manchas azuladas, sugerindo elementos mágicos e partículas de forma simbólica (Figura 52).



Figura 52. Desenvolvimento do *asset* da raposa.

7.1.3 Alquimista

O alquimista foi concebido como uma figura de renome e conhecimento vasto, porém também como alguém moralmente questionável e manipulador, sem escrúpulos ao usar outros para atingir seus propósitos. Originário da cidade subterrânea, no enredo ele foi responsável por desencadear a doença por meio de experimentos descontrolados, levando à ruína do local. Escapando ileso, estabeleceu-se na superfície próximo às ruínas, onde manipulava indivíduos para trazer itens em troca de recompensas e poções milagrosas. Queríamos que ele fosse alto e imponente em comparação com o protagonista, ao mesmo tempo que exalasse um ar de sabedoria.

No clímax do jogo, ele se revela como o *boss* final, contaminado pela doença e desejoso dos ingredientes que o protagonista coleta para si próprio, transformando-se em uma criatura monstruosa após um breve diálogo com o jogador. O processo de concepção do alquimista seguiu uma abordagem semelhante à dos outros personagens, embora, na fase de *concepts* da sua versão humana, tenhamos optado por trabalhar diretamente com arte em *pixel*. Selecionamos cores com maior contraste e exploramos várias poses e proporções (Figura 53). Após escolhermos um modelo, refinamos as proporções e acrescentamos mais detalhes e cores, substituindo, eventualmente, as cores anteriores, para melhorar a representação de sua essência

(Figura 54). Um aspecto importante foi trazer pequenos elementos com a coloração rosa, como forma de referenciar de forma sutil, a ligação do alquimista à doença.



Figura 53. *Concepts* do alquimista.



Figura 54. Desenvolvimento do alquimista.

A versão *boss* foi concebida de forma quase simultânea. Após esboçar os primeiros *concepts* do alquimista, realizamos um rascunho em papel, avaliando como ele poderia ser alterado para se tornar mais monstruoso, além de evidenciar a infecção (Figura 55). Esse rascunho foi transformado em uma versão em *pixel art*, que, aos poucos, foi ganhando mais cores e detalhes conforme a proposta original do alquimista também ia sendo melhor desenvolvida (Figura 56).



Figura 55. *Concepts do alquimista boss.*



Figura 56. *Desenvolvimento do alquimista na forma de monstro (boss).*

7.1.4 Inimigos

Os inimigos seriam seres afetados pela doença criada pelo alquimista, mas que não sucumbiram a ela, transformando-se em criaturas monstruosas, inspiradas nos inimigos de jogos (Figura 11), como, *The Last of Us* (Naughty Dog, 2013), *Dead Cells* e *MO: Astray* (Archpray Inc., 2019). Inicialmente, concebemos apenas dois tipos de inimigos que seriam ameaças na jornada do jogador. Um deles, o rastejador, mover-se-ia de um lado para o outro, não sendo hostil, mas causando dano ao encostar no jogador. O outro, o infectado, mais semelhante a um humano, demonstraria uma inteligência básica capaz de identificar, perseguir e atacar o jogador. Mais tarde, uma evolução deste último foi introduzida, o corredor, uma variante capaz de perseguir o jogador mais ferozmente. A seguir, iremos detalhar o desenvolvimento deles:

7.1.4.1 Rastejador

A inspiração para o inimigo rastejante veio de um pequeno animal infectado, posteriormente identificado como um tatu, buscando inspiração na fauna brasileira. O processo de concepção começou com esboços, explorando uma versão monstruosa do tatu, repleta de espinhos e formas circulares que lembravam fungos ou bolsas de infecção, alertando os jogadores a manterem distância (Figura 57). Em seguida o transferimos para *pixel art*, inicialmente em preto e branco para destacar formas e proporções. A seleção de cores foi refinada por meio de esboços adicionais feitos no meio digital, antes de ser aplicada na arte final em *pixel* (Figura 58).



Figura 57. *Concepts* do inimigo rastejante.



Figura 58. Desenvolvimento do *asset* do inimigo rastejante.

7.1.4.2 Infectado e Corredor

Para desenvolver o inimigo humanoide, exploramos a ideia de uma criatura com uma aparência distorcida e infectada, exibindo uma alteração na coloração da pele, com diversos fungos crescendo de forma proeminente sobre ela devido à infecção. Dado que o terror não era o foco principal, optamos por uma abordagem mais caricatural, conferindo aos infectados uma aparência mais monstruosa em vez de retratá-los como indivíduos doentes e enfermos. A criação dessas criaturas ocorreu diretamente na forma de *pixel art*, seguindo o mesmo método utilizado para o alquimista, com a elaboração de múltiplas alternativas que foram posteriormente analisadas e aprimoradas para garantir o melhor resultado possível (Figura 59).



Figura 59. Desenvolvimento do *asset* dos inimigos humanoides.

7.2 Animação dos Personagens

Com os *assets* prontos, avançamos para a fase de animação, na qual nos dedicamos a dar vida aos personagens. A animação consiste em uma sequência de imagens que retratam um movimento. Ao serem reproduzidas rapidamente, essas imagens, também chamadas de quadros ou frames, proporcionam a ilusão de movimento ao observador. Nos jogos 2D, essa alternância entre imagens é essencial para transmitir a sensação de que os personagens estão se locomovendo e de que o mundo é dinâmico.

Para se armazenar essas animações, são utilizados os *spritesheets* (Figura 60), arquivos de imagem compostos de diversos *sprites*³⁴ do mesmo personagem ou objeto, cada um representando um *frame* da animação. Isso permite que tudo fique organizado de forma mais eficiente e melhora o desempenho do jogo, já que todos os *frames* da animação ficam armazenados num único arquivo.



Figura 60. Exemplo de *spritesheet* de corrida da Momodora (2016).

Iniciamos com um planejamento abrangente, no qual identificamos e catalogamos todas as animações necessárias para cada personagem, ao mesmo tempo em que buscamos

³⁴ *Sprite* é uma imagem bidimensional que representa um personagem, objeto, projétil, etc, e que será propriamente implementada no jogo.

referências e tutoriais para cada uma delas. Nesse processo, a *spritesheet* da Momodora (2016) desempenhou um papel importante, auxiliando-nos inicialmente no planejamento e posteriormente permitindo-nos estudar os movimentos dos personagens e compreender como cada animação estava estruturada: quantos *frames* cada uma continha, como estavam divididas, entre outros detalhes.

7.2.1 Protagonista

O primeiro personagem que abordamos foi o protagonista. Começamos criando uma animação dele correndo diretamente em *pixel art* (Figura 61). Embora a animação estivesse funcional, percebemos que havia uma certa complexidade associada à animação e ao trabalho com *pixel art*, com os quais a equipe ainda não estava totalmente familiarizada.



Figura 61. Primeira versão da animação de corrida do protagonista.

Com isso, passamos a produção das animações para fora do *pixel art*, com a criação de rascunhos no aplicativo Flipaclip³⁵ (Figura 62), opção escolhida para que pudéssemos testar os movimentos antes de nos preocuparmos com os *pixels*.

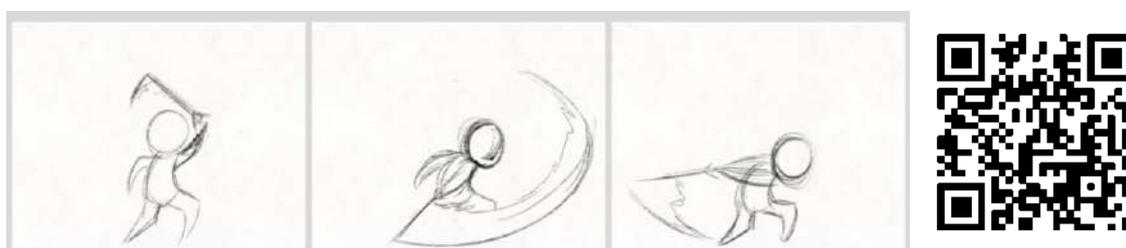


Figura 62. Rascunho feito no FlipaClip e QR code para um vídeo contendo as animações.

Para manter a uniformidade das animações, optamos por desenhar o personagem em uma pose idêntica e com proporções iguais às do *asset*. A partir dessa base, desenvolvemos todas as outras animações, garantindo não só a consistência das proporções do personagem,

³⁵ O **Flipaclip** é um aplicativo gratuito de animação para celulares e tablets, voltado para iniciantes.

mas também transições suaves entre as animações e o estado de *idle*, animação padrão para quando o personagem está parado, apenas respirando. Entre as animações criadas para o protagonista, incluíram-se corrida, ataque, ataque no ar, salto, utilização de escudo mágico, lançamento de ataque mágico, interação olhando para cima, interação olhando para baixo e a animação de morte.

Em seguida, transferimos os esboços das animações para o Aseprite. Usando uma funcionalidade do software que permite trazer outras animações como uma “camada de referência” (Figura 63), visível por trás da versão em *pixel art*, conseguimos agilizar significativamente o desenvolvimento das animações. Inicialmente fizemos animações de teste contendo apenas a silhueta do personagem (Figura 64), de maneira que pudessem ser introduzidas no jogo, testadas e corrigidas com o mínimo de retrabalho.



Figura 63. Processo de transferência das animações para o Aseprite.



Figura 64. Animação de teste contendo apenas a silhueta do protagonista.

No Aseprite, optamos também por criar um arquivo para cada personagem, contendo todas as animações associadas a ele. Organizamos essas animações utilizando o sistema de *loops* e *tags* integrado ao software. Esse sistema permite segmentar as diferentes seções do arquivo em animações distintas, que podem ser posteriormente exportadas como *spritesheets* separadas, mesmo estando contidas em um único arquivo.

Além disso, o sistema possibilita a reprodução contínua de uma única animação em *loop*, ou seja, percorrer todos os *frames* dessa animação e retornar ao primeiro, repetindo esse ciclo até que a animação seja interrompida. Essa funcionalidade facilitou significativamente o processo de animação, permitindo-nos verificar facilmente se tudo estava fluindo corretamente. Dessa forma, organizamos todas as animações de acordo com os movimentos do personagem: andar, correr, pular, atacar, entre outros (Figura 65) e (Figura 66), e separamos também os *frames* que serviriam como transição entre cada movimento.



Figura 65. Organização das animações a partir do sistema de *loops* e *tags*.



Figura 66. Detalhe da organização com o sistema de *loops* e *tags*.

Na imagem acima (Figura 66), por exemplo, nota-se que a raposa possui um conjunto de frames intitulado “walk completo”, que engloba dois *frames* de *idle* (44 e 55), um *frame* de transição entre o *idle* e o movimento (45: *start* 54: *stop*), e a animação propriamente dita de caminhada (46 ao 53: *walk*). Essa organização não apenas permite verificar se a

animação de caminhada está funcionando corretamente, mas também visualizar a transição entre o caminhar e o ficar parado.

As animações em silhueta foram então entregues aos programadores na forma de *spritesheets*, para que pudessem ser implementadas no jogo e testadas, permitindo assim, que eles confirmassem se elas estavam operando adequadamente e oferecendo um retorno de possíveis melhorias. Consequentemente, realizamos ajustes nas animações que precisavam de correção, enquanto aquelas que estavam funcionando foram refinadas na etapa de detalhamento, durante a qual adicionamos todos os detalhes e cores do personagem, além de incluir animações extras, como movimentação de cabelo, capa, entre outros.

Posteriormente, as animações finalizadas foram novamente enviadas aos programadores para implementação no jogo. Todo o processo foi aplicado às animações do protagonista, desde as mais simples, como *idle*, interação e corrida (Figura 67), até as mais complexas, como atacar (Figura 68) e usar habilidades especiais. No caso da raposa, foram criadas algumas poucas animações, como *idle* (Figura 69), uso de magia, sentar, levantar e andar, que seriam utilizadas brevemente no início do jogo, quando ela encontrasse o protagonista pela primeira vez, visto que passaria a maior parte do tempo escondida no colar do protagonista. Ao todo, foram geradas 34 *spritesheets* para o protagonista e 7 para a raposa, contabilizando cerca de 100 e 34 frames únicos para cada personagem, respectivamente.



Figura 67. *Spritesheet* da animação de correr do protagonista.



Figura 68. *Spritesheet* da animação de ataque do protagonista.



Figura 69. *Spritesheet* da animação de *idle* da raposa.

7.2.1 Animação (NPCs, Inimigos e Boss)

Os demais NPCs e inimigos em geral passaram por um processo simplificado em comparação com o protagonista. Foram desenvolvidos diretamente no Aseprite, começando com esboços básicos ou, em alguns casos, já incluindo algumas animações detalhadas para testes, pulando algumas das etapas anteriores. Isso se deve à sua presença menos frequente no jogo, à necessidade de menos animações e à movimentação mais simples, exceto por situações específicas em que encontramos dificuldades durante o desenvolvimento ou a necessidade de um planejamento mais detalhado. Nessas ocasiões, algumas etapas anteriores foram reintroduzidas ao processo. Para o alquimista, ainda sem sua forma humana como NPC, foram criadas apenas duas animações: uma de *idle* e outra de fala (Figura 70). Da mesma forma, para a mãe do protagonista, que aparece apenas no início do jogo, foi desenvolvida apenas uma animação de *idle* enquanto está deitada em uma cama (Figura 71).



Figura 70. *Spritesheet* da animação de fala do alquimista.



Figura 71. *Spritesheet* da animação de *idle* da mãe do protagonista.

Para os inimigos, nos baseamos fortemente em referências coletadas de outros jogos. Para o rastejador (Figura 72), um inimigo com uma mecânica simples, foram necessárias poucas animações, dentre elas, uma animação de *idle*, andar, virar, levar dano e morrer. Já para

os infectados humanoides (Figura 73), mecanicamente mais complexos, foram necessárias algumas animações extras, como a reação ao visualizar o jogador, correr e atacar.



Figura 72. *Spritesheet* da animação de andar do rastejador.



Figura 73. *Spritesheet* da animação de correr do humanoide.

A versão *boss* do Alquimista (Figura 74), por ser um inimigo muito mais complexo, demandou o desenvolvimento de uma quantidade maior de animações de movimentação e ataque, cuja execução também foi mais complexa, uma vez que o personagem possui muitos detalhes. Sua roupa, por exemplo, se move a cada movimento. Além disso, ele possui ataques diferentes, que vão desde bater no jogador (Figura 75), investir sobre ele (Figura 76) e atirar poções.

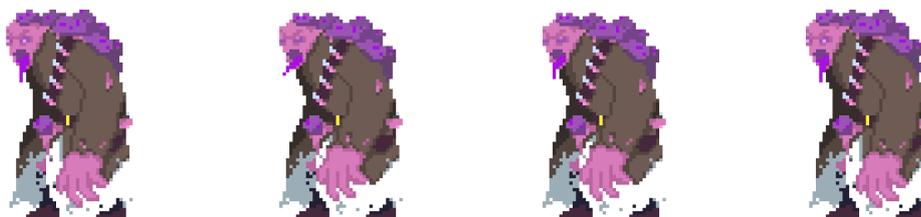


Figura 74. *Spritesheet* da animação de dano do *boss*.



Figura 75. *Spritesheet* da animação de ataque do *boss*.



Figura 76. *Spritesheet* da animação de investida do *boss*.

Todas as animações de NPCs, inimigos e *boss* passaram por etapas de teste, e, quando necessário, foram corrigidas conforme o *feedback* da equipe, até serem finalizadas e implementadas no jogo.

7.3 Revisão e Refinamento da Animação

Após a fase de testes, todos os membros da equipe experimentaram o jogo, e assim como foi explicado no delineamento, ele também foi apresentado na CPBSB5, onde pudemos ver várias pessoas jogarem e identificar diversas áreas que poderiam ser aprimoradas nas animações dos personagens. Algumas das animações do protagonista, por exemplo, estavam demasiadamente lentas, comprometendo a fluidez do movimento, enquanto certas animações dos inimigos eram rápidas demais, dificultando a percepção dos ataques que precisavam ser evitados.

Além disso, surgiu o desejo da equipe de expandir o jogo, adicionando novas mecânicas, personagens, inimigos e uma narrativa mais abrangente. Isso demandou a criação de novos personagens e animações, bem como a revisão das animações existentes. A primeira personagem a receber essa atenção foi a raposa, que passou a desempenhar um papel mais significativo na história e na jogabilidade. Originalmente concebida para ter uma presença mínima, permanecendo a maior parte do tempo dentro do colar do protagonista, decidimos, inspirados em jogos com *companions*, como Planet of Lana (Wishfully, 2023), que ela deveria ser mais visível, seguindo o jogador durante o jogo e sendo crucial para superar obstáculos.

Para destacar o vínculo entre ela e o protagonista, assim como introduzir pequenos momentos entre os dois, decidimos dedicar mais atenção às suas animações, tratando-as com o mesmo cuidado dado ao protagonista. Assim, utilizamos o FlipaClip para criar esboços de todas as animações da raposa, refinando algumas das antigas que haviam sido feitas às pressas. Como ela agora acompanharia o jogador em vez de permanecer escondida no colar, adicionamos diversas novas animações que não haviam sido previamente consideradas, como corrida (Figura 77), *dash* (Figura 78), pulo, entre outras.



Figura 77. *Spritesheet* da animação de correr da raposa.



Figura 78. *Spritesheet* da animação de *dash* da raposa.

7.4 Considerações sobre os Personagens

Com isso, foi possível se desenvolver uma aparência única para cada personagem, que iriam não só contribuir para a experiência estética do jogador, como também seriam capazes de comunicar diversas informações de forma visual. Um exemplo, sendo demonstrar quais personagens seriam inimigos a partir do uso de tons de rosa e roxo. Também, foi possível criar diversas animações que, além de dar vida aos personagens, contribuem para o funcionamento das mecânicas, dando mais peso a elas e deixando claro o seu funcionamento. Por exemplo, permitindo que o jogador perceba facilmente se o seu ataque acertou ou não um ataque contra o inimigo, por meio da animação de dano.

8. Mapa e Cenários

Neste capítulo, vamos abordar todas as etapas para a criação dos mapas e cenários do Melius. O mapa consiste em toda a área designada pelos desenvolvedores para ser navegável e explorável pelo jogador. Cenários, por sua vez, classificamos como a parte mais visual e artística do desenvolvimento. Partindo das decisões já tomadas nas etapas anteriores, vamos elaborar as etapas de *level design*, *concept art*, utilização de tiles e camadas para construção dos mapas, criação dos *assets* e a finalização dos mapas.

8.1 Level Design do Mapa

O *level design* consiste na criação de todo o mapa explorável juntamente com definição dos caminhos que o jogador deverá tomar, acompanhada da definição de todos os pontos de interesse e interação, como plataformas, *checkpoints*, barreiras, habilidades especiais, itens coletáveis, posicionamento NPCs etc. As principais referências para o *level design* do Melius foram jogos como Hollow Knight, Ori and the Blind Forest, Super Mario Bros (Nintendo, 1987), Metroid (Nintendo, 1989) e Metroid Dread (MercurySteam, 2021). Com exceção de Super Mario Bros, todos esses jogos se encaixam no gênero Metroidvania, que, conforme vimos, foi o escolhido para o nosso jogo.

Dentro do gênero Metroidvania, as principais características que tentamos nos inspirar são a ênfase na liberdade e não-linearidade da exploração e na criação de um mapa grande e conectado. Ao mesmo tempo em que dá ao jogador a possibilidade de percorrer múltiplos caminhos até atingir seu objetivo, essa forma de *level design* permite a criação de formas criativas de guiar o jogador para o caminho determinado pela narrativa do jogo. Outra característica deste gênero que influenciou a construção do *level design* foi o *backtracking*, isto é, momentos do jogo nos quais o jogador precisa voltar por uma fase ou área que ele já visitou anteriormente, porém, com alguma novidade, ou com a liberação de um caminho previamente inacessível, para atingir seu próximo objetivo, enfatizando a sensação de exploração e familiarização do jogador com o mapa.

8.1.1 Produção

Inicialmente, utilizando a plataforma Figma, o grupo começou a explorar esquemas de *layout* simples para o mapa. Levando em conta a narrativa já estabelecida, definimos a estrutura básica do mapa, identificando as grandes áreas e delineando o percurso que o jogador seguiria ao longo do jogo, bem como os principais pontos de interesse. Nesta fase, elaboramos duas opções (Figura 79) e (Figura 80).

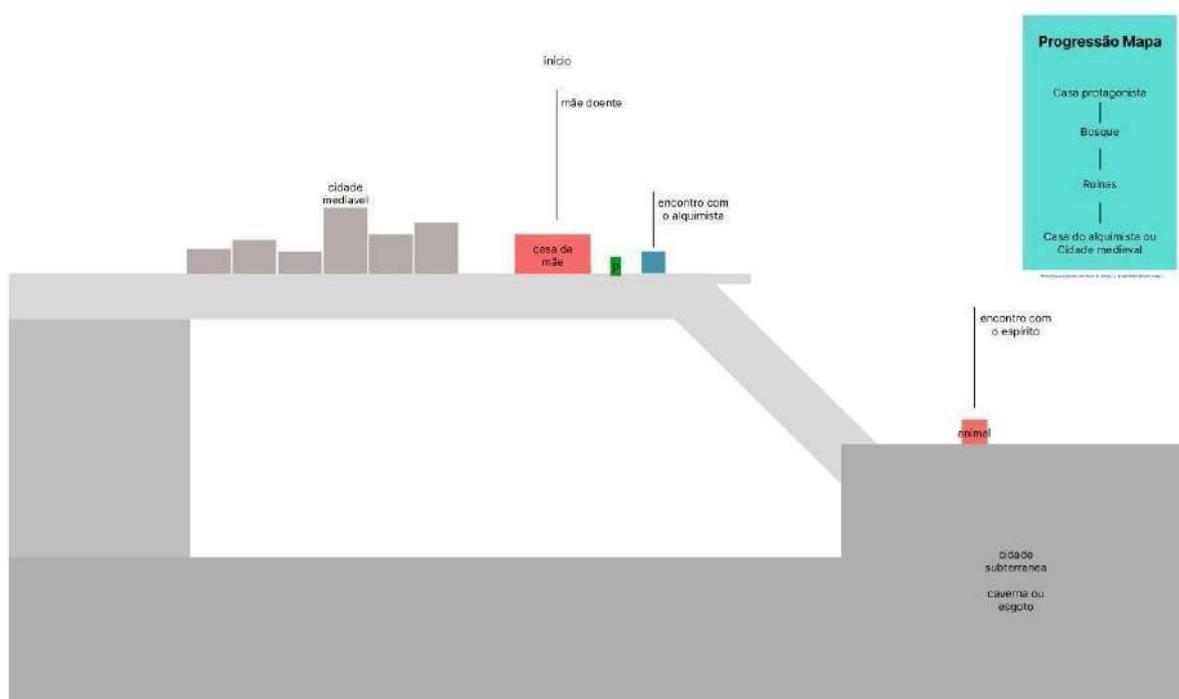


Figura 79. Primeira alternativa de *layout* para o mapa.

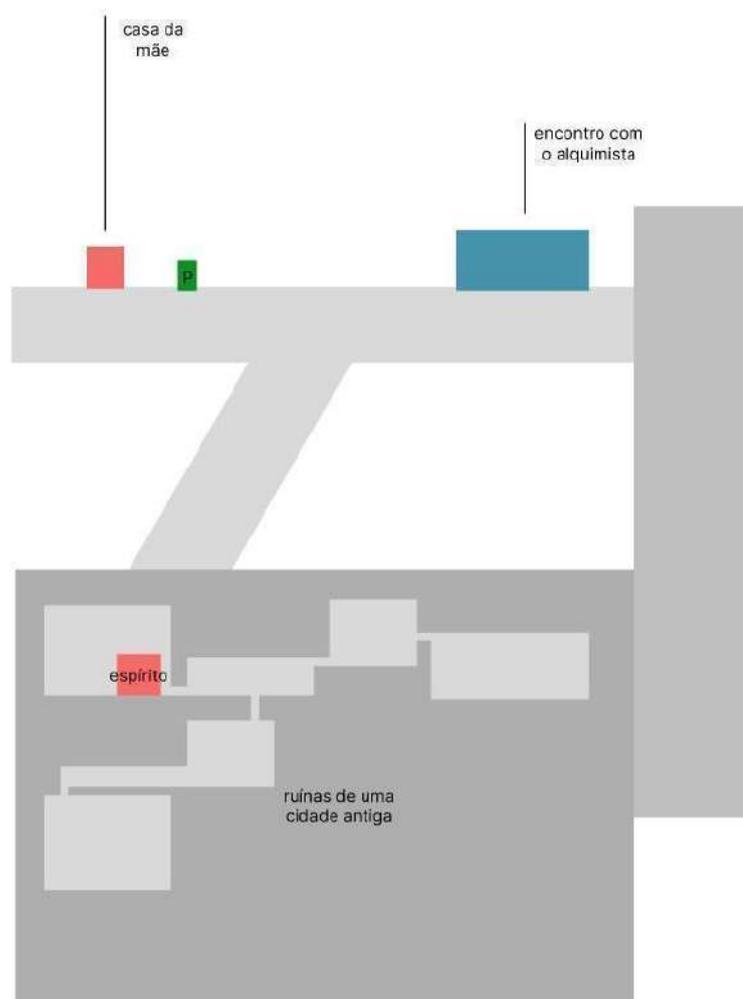


Figura 80. Segunda alternativa de *layout* para o mapa.

Ambas as opções são compatíveis com a narrativa proposta, embora ainda carecessem de detalhamento em termos de *level design*. Essa simplicidade dos *layouts* possibilitou uma maior flexibilidade posteriormente na elaboração e desenvolvimento principalmente do *level design*. A progressão planejada compreendia: Casa do protagonista > bosque/floresta > ruínas/cidade subterrânea > casa do alquimista ou cidade medieval.

Neste momento, decidimos alocar dois membros do grupo como principais responsáveis pelo restante do planejamento, além do desenvolvimento do *level design*, *concepts* e *assets* de todo o mapa e cenários. Enquanto isso, um outro membro ficou responsável pela implementação dos cenários criados, e o restante da equipe forneceu *feedbacks* e auxiliou em momentos pontuais sempre que se mostrou necessário.

Selecionamos a segunda alternativa e, mantendo grandes retângulos como base, avançamos no detalhamento do *level design* (Figura 81).

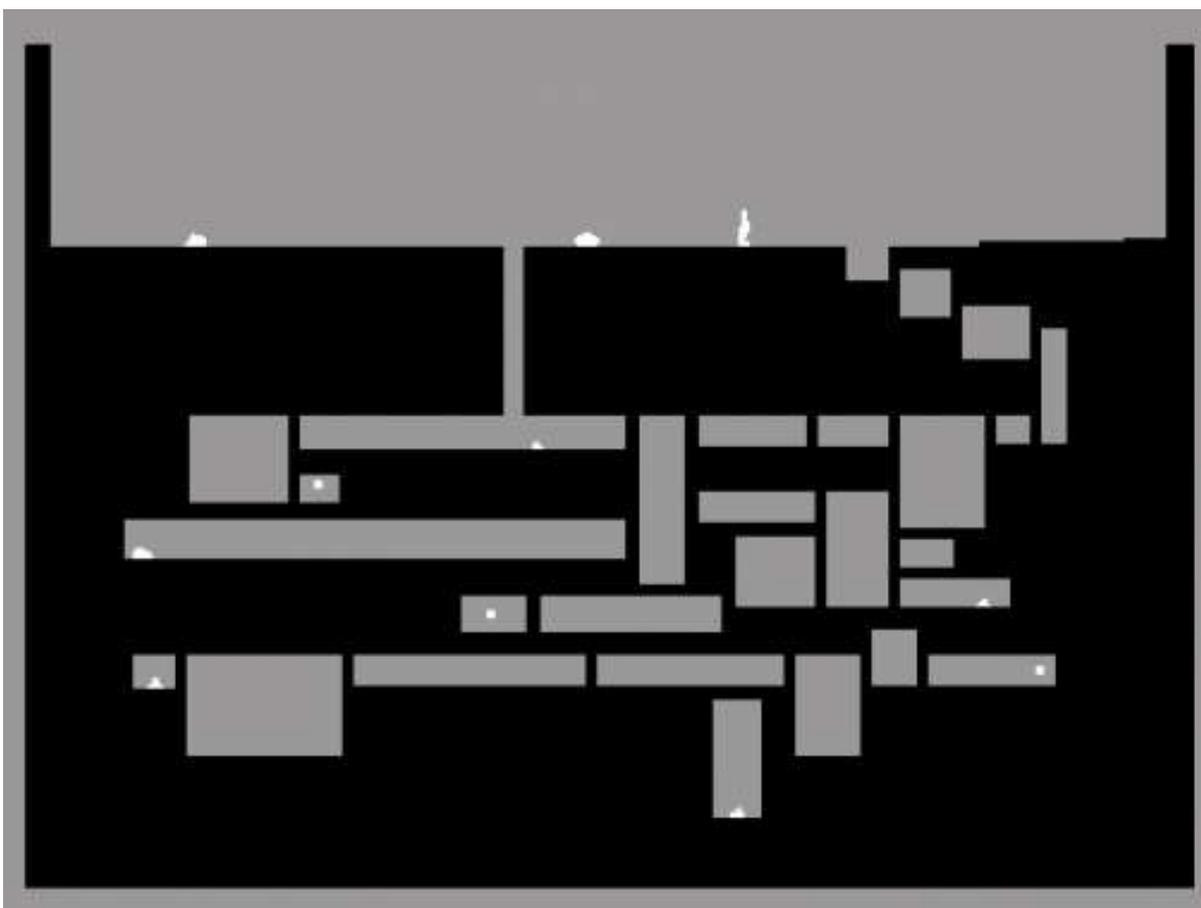


Figura 81. *Level design* blocagem das salas.

Iniciamos a definição mais precisa das salas, ainda sem estabelecer conexões diretas entre elas, mas ainda assim considerando onde essas conexões poderiam ser criadas. Em seguida, começamos a elaborar uma progressão mais detalhada, identificando salas específicas e potenciais pontos de interesse, indicados em branco.

Em seguida, procedemos com a adição de plataformas, sem planejar sua altura ou espaçamento. Estabelecemos a conexão entre as salas e refinamos os detalhes para conferir uma aparência mais natural ao mapa, tornando a navegação mais intuitiva. Nesta etapa, concentramo-nos na área da cidade subterrânea, identificando os quatro principais pontos de interesse, que incluíam os itens necessários para a poção, além de três pontos secundários representados pelos mecanismos para abrir a porta (Figura 82).



Figura 82. *Level design* inclusão de plataformas.

Consultaremos o restante da equipe e receberemos *feedback* em como melhorar os elementos de plataforma, e aprimorar o detalhamento do mapa do Melius para incorporar o estilo Metroidvania. Revisamos a conexão entre as salas, introduzindo diferentes caminhos que levam o jogador ao mesmo objetivo. Isso lhe permitiria ter a liberdade de escolher seu percurso ou criar um ciclo, retornando a áreas previamente exploradas após descobrir novas regiões.

As plataformas foram posicionadas levando em consideração o tamanho do personagem e a mecânica de salto. Nosso objetivo foi garantir que o jogador tivesse espaço para se movimentar com liberdade, mantendo as plataformas acessíveis sem grandes desafios (Figura 83).



Figura 83. Posicionamento de plataformas.

Decidimos incluir barreiras físicas (Figura 84) para bloquear o avanço do jogador até que ele alcance objetivos específicos: o primeiro portão (à esquerda) impede a entrada no subterrâneo até que o jogador interaja com o alquimista. O segundo portão (à direita) impede que o jogador deixe o terreno subterrâneo antes que ele colete os quatro itens necessários para criar a poção e interaja com os três mecanismos que liberam o portão.

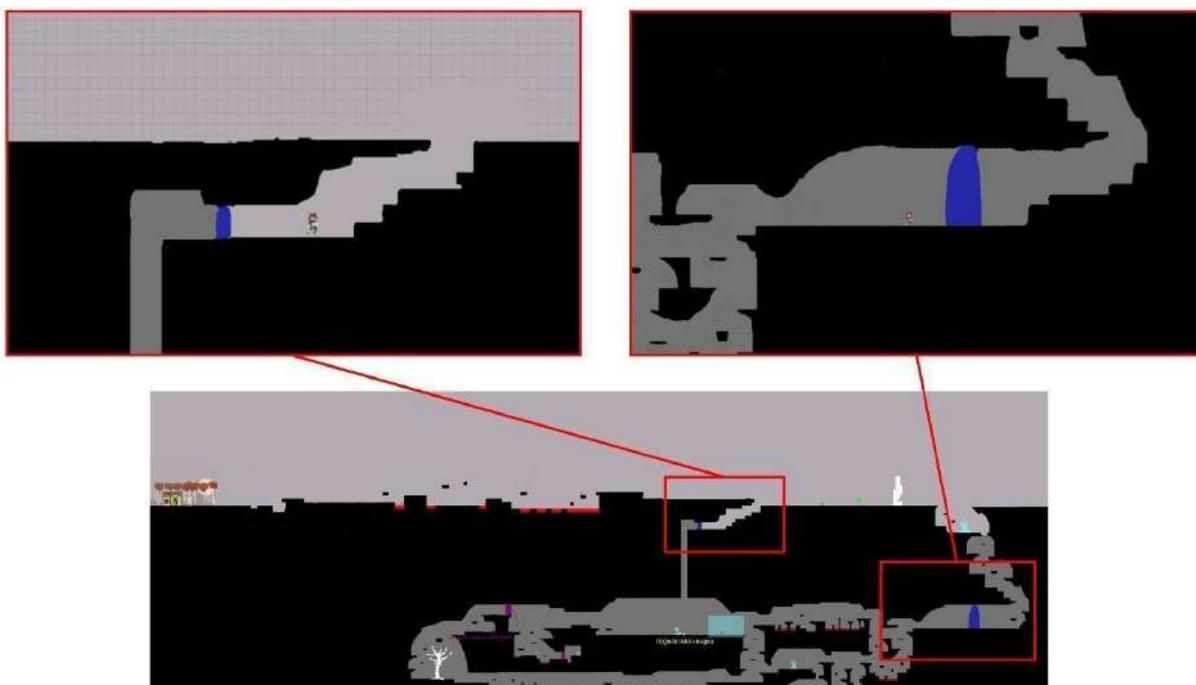


Figura 84. Portões para bloquear a passagem do jogador.

Além disso, foram posicionados os locais de *checkpoint* (sinalizados em azul claro) e marcamos em vermelho o posicionamento dos blocos de dano para servir como obstáculos e desafios durante a exploração do mapa (Figura 85).



Figura 85. *Level design* detalhamento da cidade subterrânea.

Para criar o *level design* da superfície, adotamos medidas especiais, pois essa área funcionaria também como um tutorial para ajudar o jogador a se familiarizar com as mecânicas e controles do jogo. Nesse estágio, conduzimos uma pesquisa, estudando de perto jogos conhecidos pela qualidade do seu *level design* e tutoriais eficazes. Um dos jogos que se destacou nesse aspecto foi o Super Mario Bros (1985), especialmente o mundo 1-1 (Figura 86), criado por Shigeru Miyamoto e Takashi Tezuka, que se tornou uma das nossas principais referências.



Figura 86. Super Mario Bros mundo 1-1 (1985) Disponível em: <https://codegolf.stackexchange.com/questions/40008/pixel-art-episode-2-display-the-map-1-1-of-super-mario-bros> Acesso em: 21 de Agosto de 2024.

Esse nível é um exemplo notável de como o *level design* orienta gradualmente o jogador, ensinando as mecânicas fundamentais sem depender de textos na tela. Segundo Miyamoto e Tezuka em entrevista ao canal do Youtube: Eurogamer (2015), a intenção por trás da criação dessa fase foi permitir que o jogador aprenda e se acostume com os controles e mecânicas conforme são introduzidos ao longo do percurso, mas seja livre para experimentar o jogo e explorar as possibilidades (Miyamoto, 2015; Tezuka, 2015).

A série Metroid, tanto o clássico de 1986 quanto o mais recente Metroid Dread, foram uma grande fonte de inspiração. Observamos como esses jogos organizam o *level design*, orientando sutilmente o jogador em direção aos objetivos, enquanto ainda permitem uma exploração livre e dinâmica. Para começar, criamos alguns esboços no papel (Figura 87), explorando várias ideias para o *level design* da superfície.

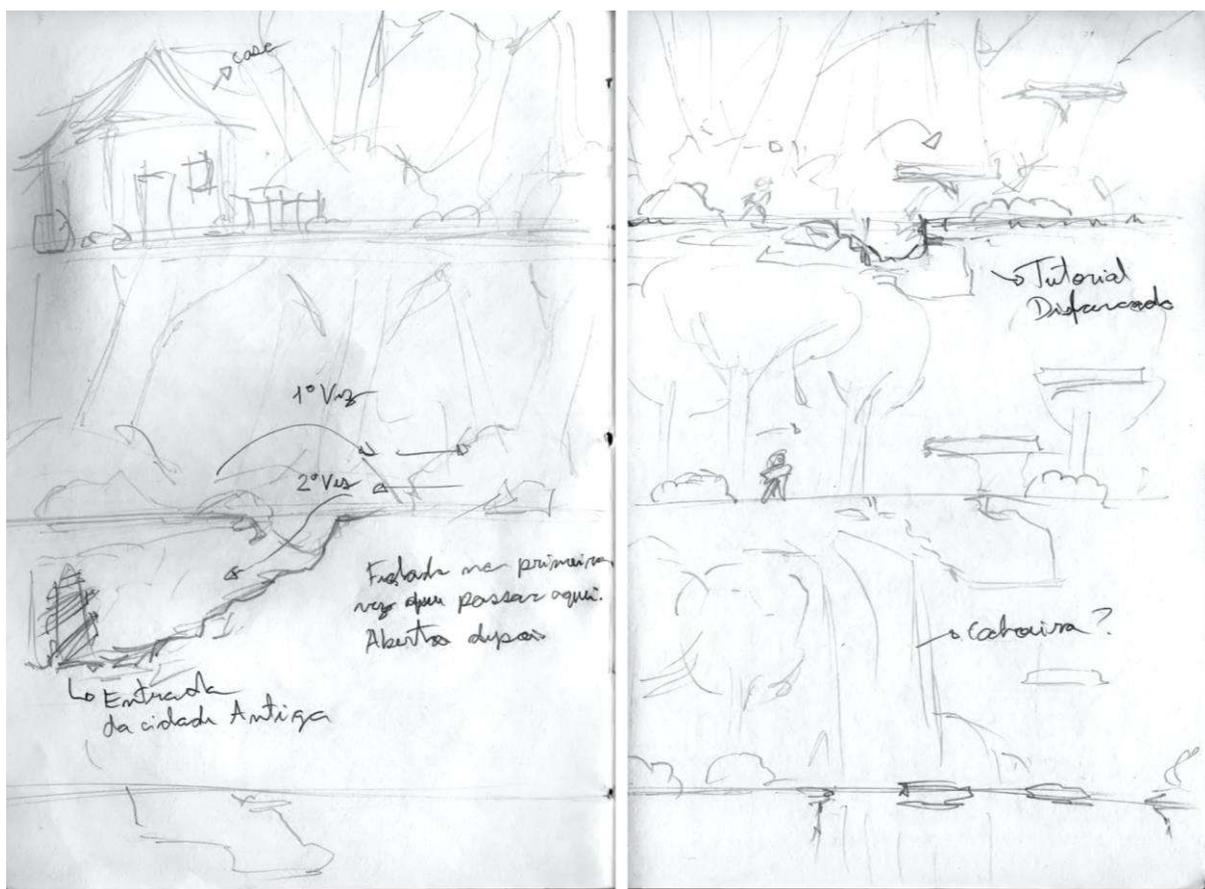


Figura 87. Rascunho e planejamento do *level* da floresta.

Esses esboços incluíam propostas para a disposição das plataformas e outros elementos de interesse. Além disso, destacamos um ponto importante: a entrada da cidade subterrânea. Desde o início, planejamos que essa entrada estaria fechada. O jogador passaria por ela uma vez, seguindo seu caminho. Somente após interagir com o alquimista e receber sua primeira missão é que a entrada seria aberta.

8.1.2 Alternativa Selecionada

A alternativa escolhida para o level design da superfície ficou da seguinte maneira (Figura 88). Neste ponto, já temos, também, um mapa que descreve a progressão do jogador e inclui todas as mecânicas desenvolvidas.

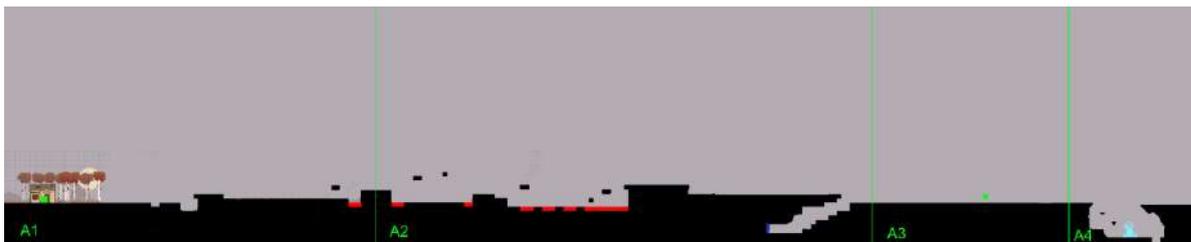


Figura 88. *Level design* Floresta.

O jogador começaria o jogo na casa da mãe, na área A1 (Figura 88) e, ao sair, passaria pela floresta ou bosque, que também serviria como o tutorial do jogo, onde encontraria os primeiros saltos, que ainda não apresentando perigos, permitiriam que ele se familiarizasse com a mecânica de pular conforme descrito anteriormente. Mais adiante, ele encontraria saltos mais desafiadores, que conteriam obstáculos como blocos de dano, além de diferentes caminhos e maneiras de superá-los, aumentando a dificuldade de forma gradual. Em seguida, o jogador encontraria o primeiro inimigo simples, permitindo que ele teste a mecânica de combate. Logo após, ele passaria por uma caverna que seria a entrada para a cidade subterrânea, na fase A2 (Figura 88). Contudo, a entrada estaria fechada, obrigando-o a seguir para a direita até encontrar o alquimista, área A3 (Figura 88).

Ao interagir com o alquimista, a passagem para a cidade subterrânea seria aberta, e o jogador receberia a missão de coletar quatro itens dentro da cidade. Para a cidade subterrânea o *level design* ficou da seguinte maneira (Figura 89).

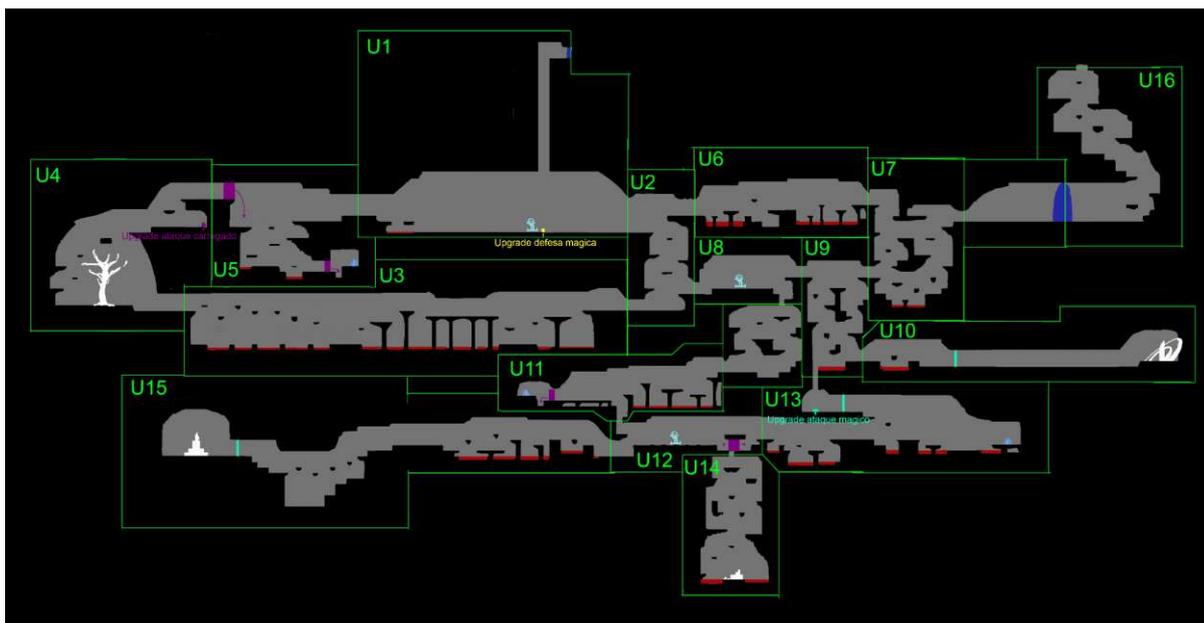


Figura 89. *Level design*: cidade subterrânea.

Ao adentrar na caverna, o jogador pula em um buraco e cai na cidade subterrânea. Nesse momento, ele pode escolher entre quatro caminhos diferentes, sem uma ordem específica. Primeiro, ele pode se dirigir ao portão de saída da cidade marcado em azul escuro e localizado entre a fase U7 e U16, onde será informado de que precisa coletar quatro itens e interagir com três mecanismos para liberar o portão.

Outra opção é seguir em direção à árvore marcada em branco na fase U4, onde ele encontrará o primeiro item da poção e, em seguida, obterá a primeira habilidade especial de ataque carregado (roxo), permitindo ao jogador então, acessar todas as passagens bloqueadas em roxo. Ele também pode optar por buscar a habilidade especial de ataque mágico em ciano na fase U13, que permite acessar todas as passagens bloqueadas em ciano.

Por fim, o jogador pode seguir para baixo, onde passará por dois *checkpoints* e ativará o primeiro totem também na fase U13, necessário para abrir a porta de saída da cidade. O jogador está livre para explorar o mapa e coletar os itens na ordem que desejar, cada um dos caminhos iniciais passa informação ou abre novos caminhos para o jogador progredir no jogo.

Depois de coletar todos os itens, o jogador poderá abrir os portões da cidade e voltar para a floresta, alcançando o alquimista por um caminho anteriormente inacessível. Com isso, o *level design* está completamente desenvolvido (Figura 90).



Figura 90. *Level design* da primeira versão do jogo.

Durante o desenvolvimento do *level design*, toda a imagem foi desenvolvida em tamanho real levando em consideração o tamanho dos *tiles*, para isso foi utilizada a ferramenta de *grid* do Photoshop³⁶, configurada para apresentar um *grid* de 28x28 *pixels* (Figura 91).



Figura 91. Grid no *level design*.

Assim pudemos incluir o personagem jogável e o tamanho definido da tela, que estão destacados em azul ciano com transparência, para fins de comparação e testes. Isso ajudou a manter a proporção do mapa e na definição do tamanho das fases, para que, por exemplo, elas não ficassem muito pequenas e apertadas ou muito grandes ao ponto que boa parte da fase nem seria vista durante o jogo.

É importante destacar a padronização de cores que implementamos no desenvolvimento do *design* do nível (Figura 92).

³⁶ **Photoshop** é um *software* gráfico para edição de imagens, criação de arte digital, *design* gráfico e animações.



Figura 92. Padronização de cores no *level design*.

Cada cor foi atribuída a um significado específico, representando objetos, mecânicas, e outros elementos. Essa padronização facilitou o entendimento e a comunicação entre os membros da equipe.

8.2 Concepção dos Cenários

A criação dos cenários teve início ao considerar as referências da pesquisa e o enredo desenvolvido para o jogo. Primeiramente, foram realizados testes para definir a direção artística, a paleta de cores e o clima de cada uma das áreas principais. Ao analisar os jogos usados como inspiração, notamos uma característica comum: os cenários têm o poder de contar

a história do jogo visualmente, incentivando os jogadores a refletirem sobre eventos passados e a formular teorias sobre o enredo. Essa abordagem visa engajar os jogadores de maneira mais ativa na história, em vez de apenas serem espectadores. Então, um aspecto crucial foi levar em conta a história previamente estabelecida como guia na hora de experimentar com *concepts*. Dessa forma, a equipe começou a planejar os principais pontos de interesse de cada mapa, como a casa do protagonista, e criou conceitos específicos para essas áreas únicas.

8.2.1 Floresta

O primeiro cenário principal decidido foi o da floresta isolada na superfície, onde o jogo se inicia. Primeiro, foram feitos alguns *concepts* livres para experimentar paleta de cores e combinações (Figura 93) e (Figura 94).

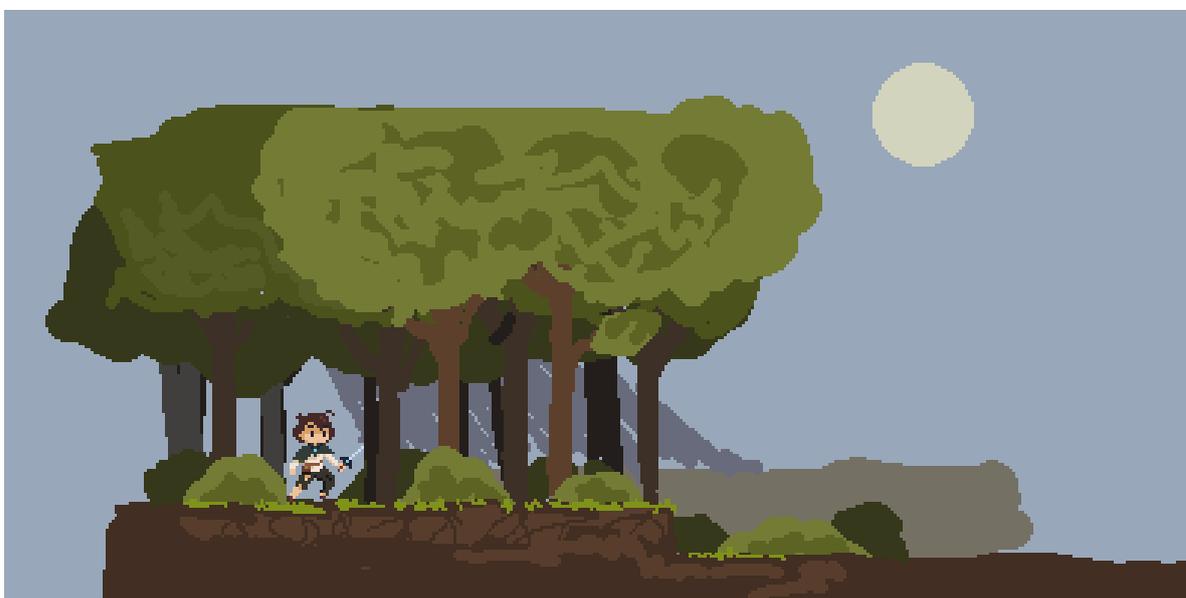


Figura 93. *Concept* da Floresta cores e combinações 1.

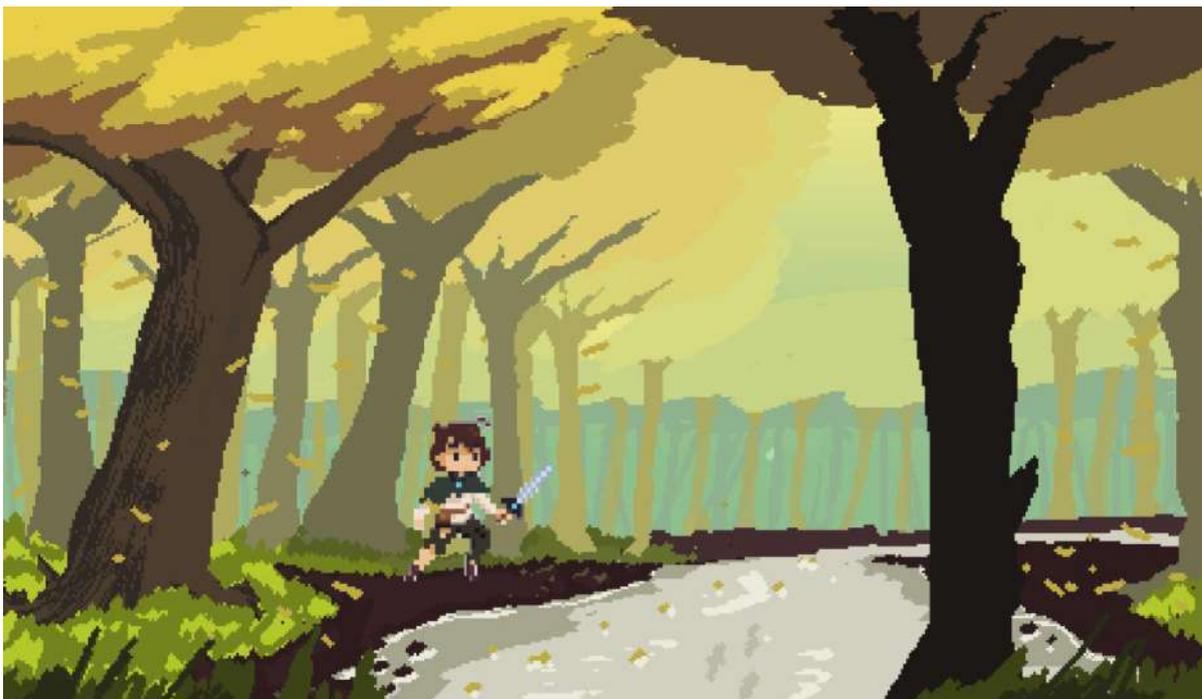


Figura 94. *Concept* da Floresta cores e combinações 2.

Em discussão com a equipe, chegamos à conclusão de que uma floresta com árvores verdes não combinaria com o tom que gostaríamos de passar naquele momento. Mas também não queríamos fazer da área inicial uma floresta completamente destruída e soturna. Por isso, optamos por uma paleta quente, mais amarela com cores de outono para a floresta, com o objetivo de passar uma sensação mais melancólica ao jogador (Figura 95).

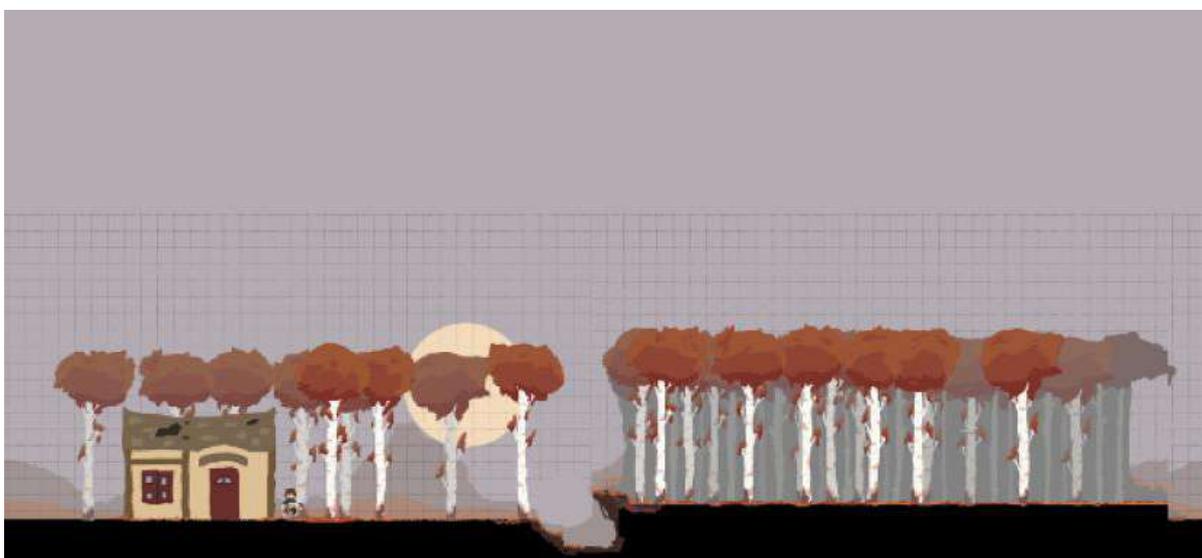


Figura 95. *Concept* da Floresta cores de outono.

Também desenvolvemos *concepts* de áreas importantes presentes na floresta, como a casa onde o personagem principal vive com a mãe dele, pensada a partir da perspectiva exterior (Figura 96) e interior (Figura 97).



Figura 96. Rascunho em papel da casa do protagonista: exterior.

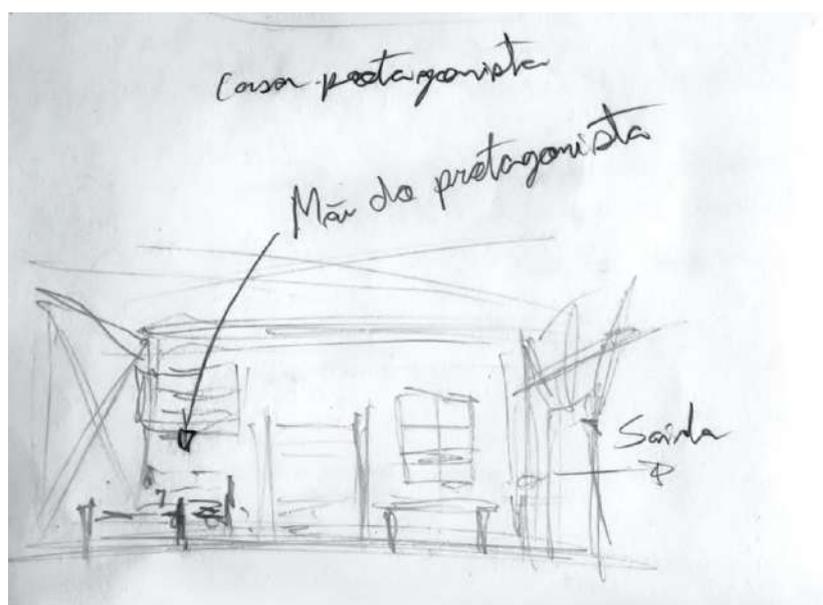


Figura 97. Rascunho em papel da casa do protagonista: interior.

Nesse local, também estaria localizada a entrada de uma caverna que levaria ao subterrâneo (Figura 98) e à casa (ou castelo) do alquimista, que também foi pensada tanto a partir da perspectiva exterior (Figura 99) quanto interior (Figura 100).



Figura 98. Rascunho em papel da Entrada da caverna.

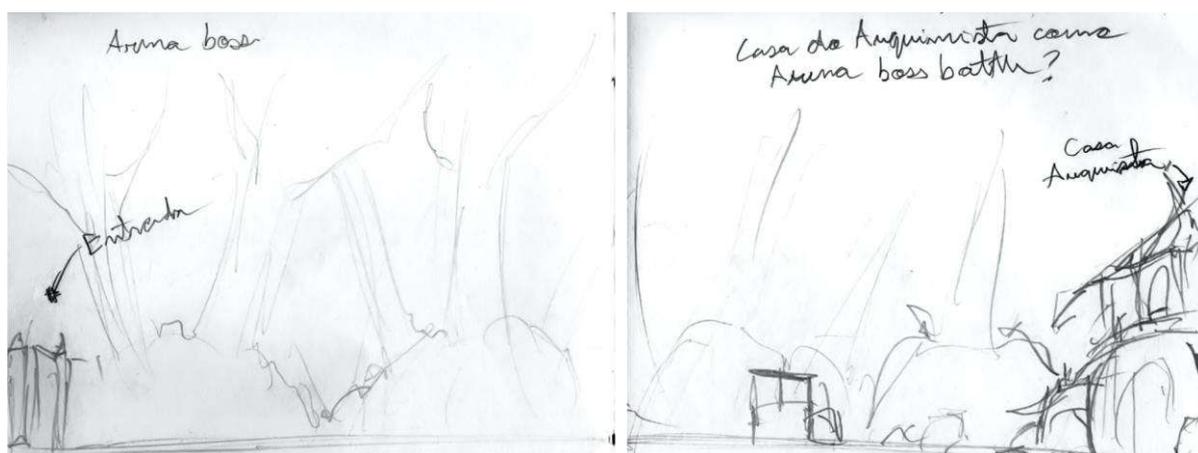


Figura 99. Rascunho em papel da Casa do alquimista exterior.

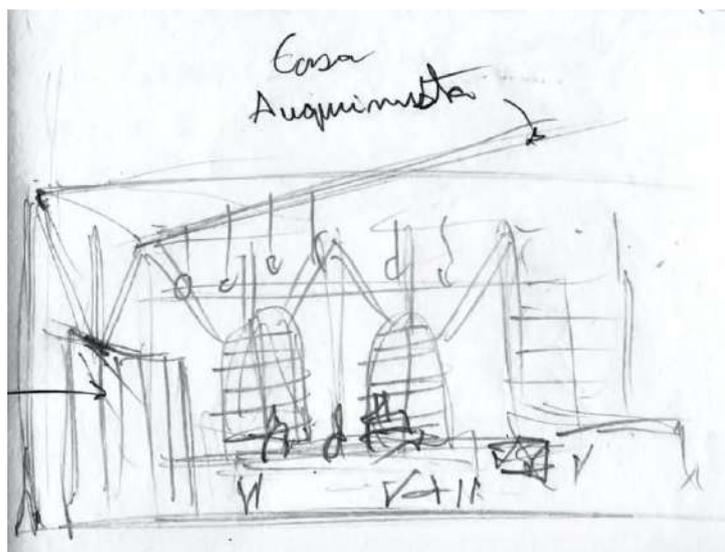


Figura 100. Rascunho em papel da Casa do alquimista interior.

8.2.2 Cidade subterrânea

O segundo e maior mapa é uma cidade subterrânea, completamente isolada e devastada por uma guerra e, posteriormente, também por uma doença, conforme descrito no enredo (Figura 101).

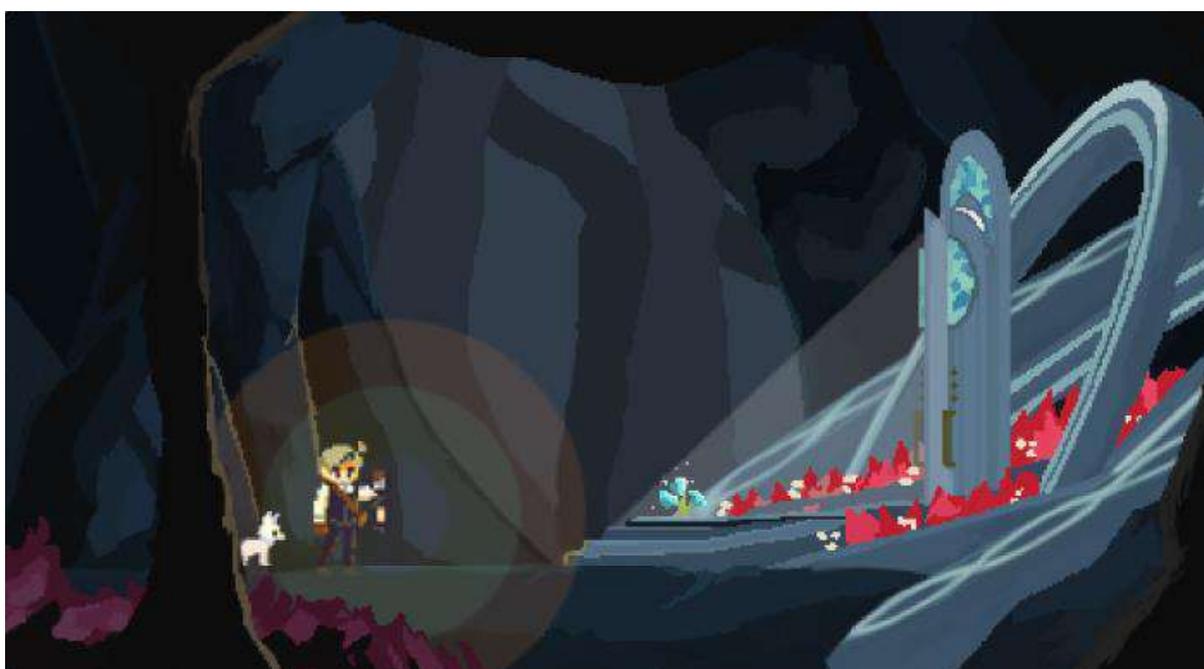


Figura 101. *Concept* Cidade Subterrânea: cores frias.

Optamos por uma paleta de cores mais fria, com tons de azul e cinza, para criar uma atmosfera solitária, melancólica e perigosa. Além disso, trabalhamos para que objetos importantes tivessem cores distintas que se destacam facilmente, como locais mágicos em azul ciano. O monumento foi desenvolvido bem cedo na concepção. Ele foi feito para ser uma porta mágica inicialmente inacessível e um dos locais de coleta de itens da poção

Desenvolvemos *concepts* para outras áreas importantes presentes no subterrâneo. Um deles é “a grande árvore” na fase U4, desenvolvida para ser um dos locais onde o personagem coleta um dos itens para a poção (Figura 102). Outro também pensando brevemente nesta etapa, foi a estátua da raposa que posteriormente seria colocada na fase U15 (Figura 103).

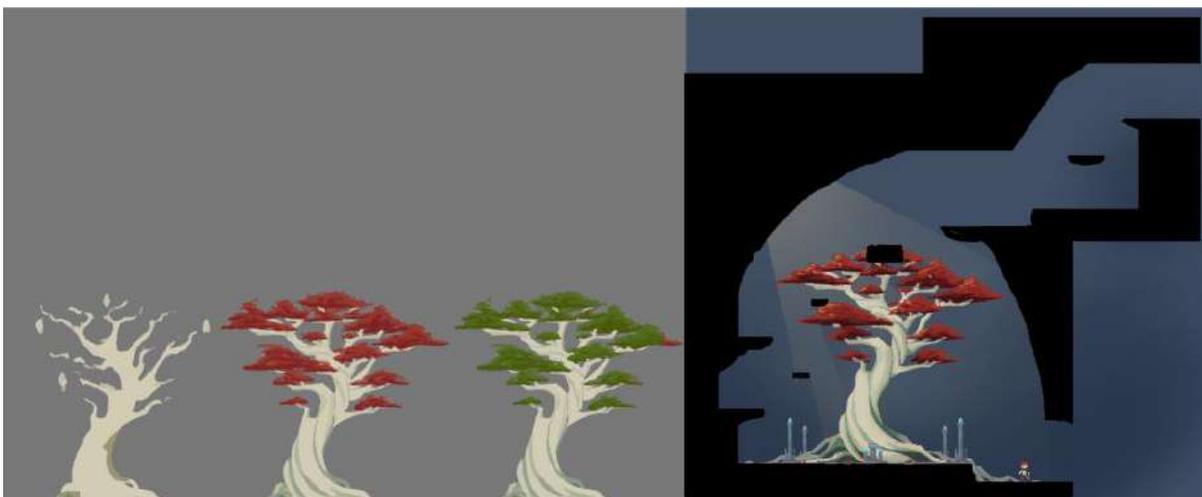


Figura 102. *Concept* árvore sagrada.

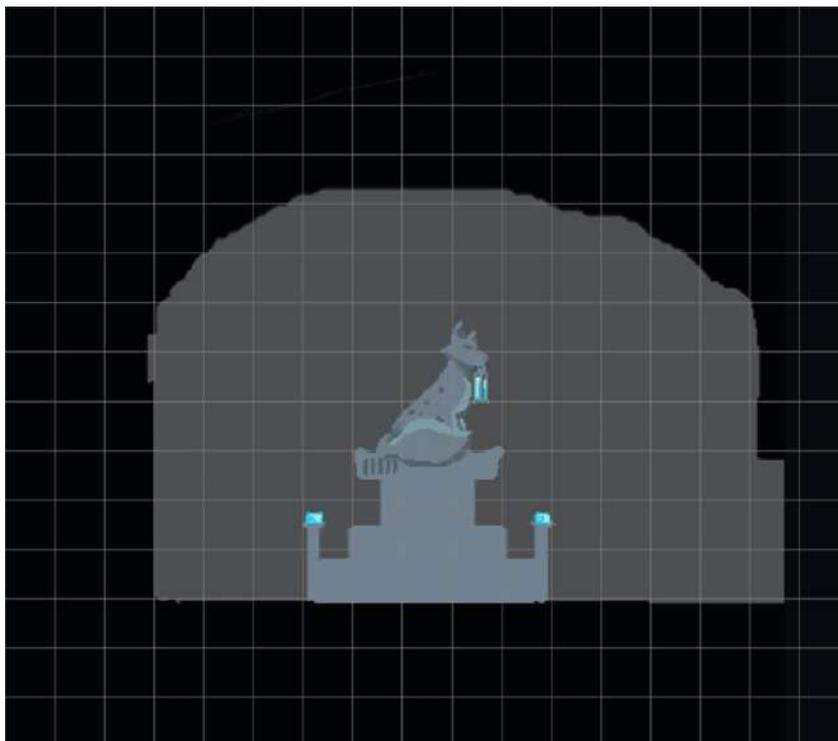


Figura 103. *Concept* fonte da raposa.

8.2.3 Desenvolvimento da expressão visual da Doença.

A Doença é a principal força antagonista do jogo, que se alastra pela superfície e o subterrâneo, infectados seres vivos e os transformando em monstros, que o jogador tem que enfrentar. Ela também é o que leva o jovem protagonista a sair em sua jornada, ao infectar e debilitar a mãe dele. Durante o processo de concepção de características para representar essa infecção nos inimigos e cenário, percebemos, a partir de observação das referências coletadas (Figura 9) e (Figura 11), que a cor roxa é frequentemente utilizada para representar perigo ou toxicidade. Também notamos, que o roxo é uma cor que se distancia bem e mantém uma boa relação de harmonia com o azul utilizado no subterrâneo e o laranja escolhido para a floresta (Figura 104) se destacando bem nos cenários. Além de ter um bom contraste com o azul que seria usado na raposa mágica.



Figura 104. Comparação das paletas de cores.

Então optamos pela utilização da cor roxa para desenvolver as experimentações de como iríamos representar a doença nos cenários (Figura 105). Apesar de o resultado ter ficado diferente da concepção inicial desenvolvida, todos os *concepts* serviriam como base e guia para a construção dos cenários.

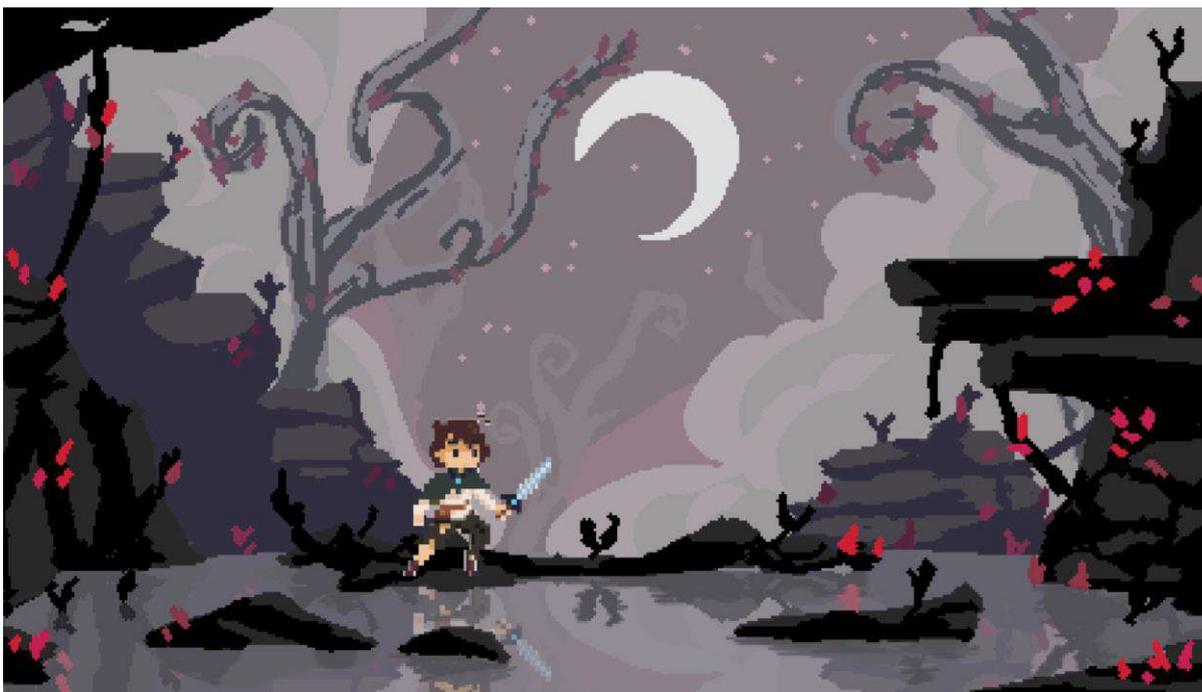


Figura 105. *Concept* expressão visual da doença.

Aqui, trouxemos uma aparência desolada para o cenário afetado pela doença, com muitos tons escuros e uma aparência distorcida para a vegetação. Posteriormente, durante a produção dos *assets* e finalização dos cenários, decidimos usar fungos e cogumelos, inspirados nas referências coletadas anteriormente (Figura 9), para trazer a imagem de que a doença não só estaria infectando pessoas e animais, como até mesmo o próprio cenário, se espalhando pelo solo, pelas construções e infectando até mesmo a vegetação.

8.3 Uso de Tiles para a construção do Mapa

Neste tópico, abordaremos como a equipe utilizou *tiles* utilizando a ferramenta TileD para a construção dos mapas de Melius. TileD é um editor de *level design* para jogos 2D, permitindo importar um *tileset* e construir todo o *level* de maneira estrutural e visual. Posteriormente, exporta-se um arquivo que pode ser utilizado pela equipe de programação para implementar um nível com a mesma configuração espacial dentro do jogo.

Um *tileset* é um arquivo de imagem que agrupa todos, ou uma parte, dos *tiles* utilizados no jogo. Os *tiles*, por sua vez, são imagens que representam os diversos elementos que compõem um cenário, como o chão onde o jogador pisa, as paredes, os objetos, as construções, as texturas, e muito mais. Esses elementos são muitas vezes divididos em várias partes, cada uma sendo um único *tile*, que no jogo são organizados e repetidos de forma a criar um desenho plausível para o cenário. Assim, com alguns desenhos, o projetista é capaz de criar diversas fases diferentes para o jogo, sem que seja necessário ilustrar tudo do zero. Então com apenas 32 *tiles*, por exemplo, repetindo e reorganizando eles, você consegue criar o chão de todos os cenários de um jogo, como pode ser visto na Figura 106 retirada do jogo Super Mario Bros.



Figura 106. *Tileset* e construção de mapa Super Mario Bros. Disponível em: <https://codegolf.stackexchange.com/questions/40008/pixel-art-episode-2-display-the-map-1-1-of-super-mario-bros> Acesso em: 21 de Agosto de 2024.

O TileD permite posicionar todos os *tiles* a partir de um *tileset*, e ainda, atribuir propriedades a eles, como, por exemplo, fazer com que determinado *tile* tenha colisão, ou que funcione como bloco de dano. Todos os *tiles* precisam ter o mesmo tamanho ou serem múltiplos do valor original, para que a imagem possa ser segmentada em programas como TileD para criação dos mapas e para que possa ser implementado de maneira correta no jogo.

O uso dessa ferramenta facilita a conexão entre arte e programação, permitindo que os *designers* criem os níveis de maneira mais livre e, em seguida, forneçam aos programadores o arquivo com todos os *tiles* posicionados e pronto para ser implementado. Começamos definindo as dimensões dos *tiles*, optando por um tamanho de 28x28 *pixels*. Esse tamanho é menor que o protagonista, mas ainda oferece espaço suficiente para criar artes detalhadas no estilo *pixel art*. Os primeiros *tilesets* desenvolvidos foram os de colisão e os blocos de dano para o cenário subterrâneo. Utilizamos o Photoshop com a ferramenta de *grid* presente no programa, e com base nas referências, começamos a desenhar, ainda sem muitos detalhes, os *tiles* que achamos que seriam mais essenciais, para rapidamente colocá-los a teste na construção dos mapas. À medida em que a necessidade de novos *tiles* surgia, nós desenhávamos novos *tiles* no mesmo *tileset* (Figura 107).

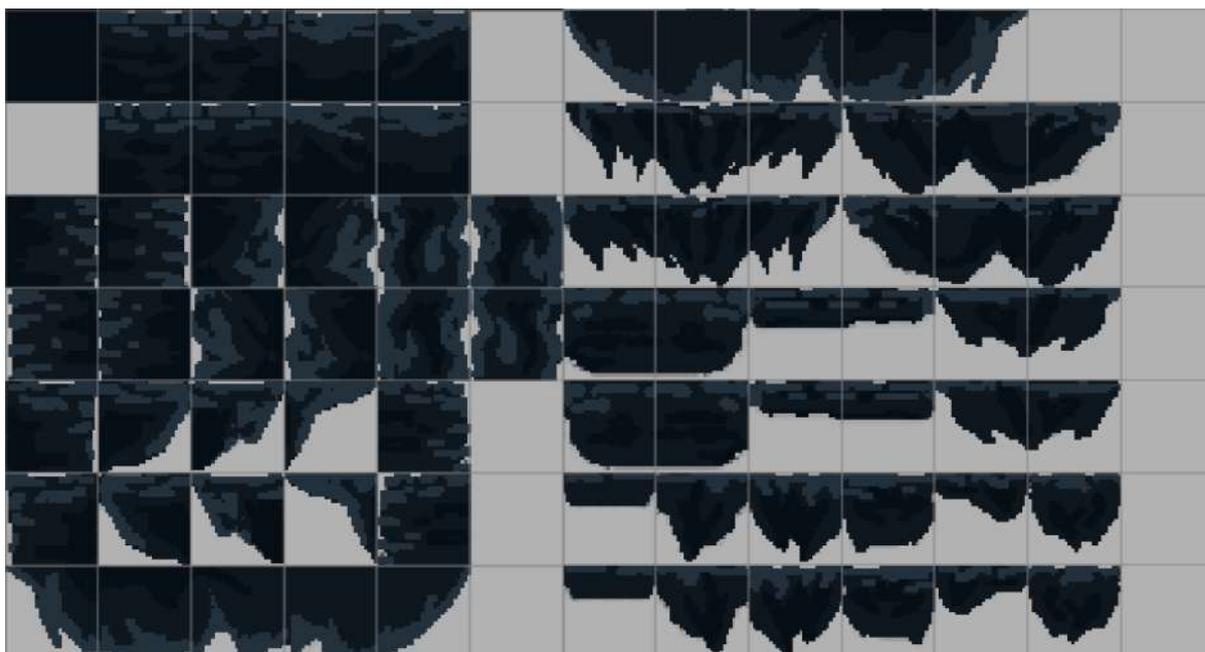


Figura 107. Primeira versão do *tilesets*.

Conforme a quantidade de *tiles* foi se mostrando suficiente para construir o mapa, nós voltamos ao *tileset* para reorganizar os *tiles* de maneira mais sucinta e adicionar detalhes ao desenho (Figura 108).

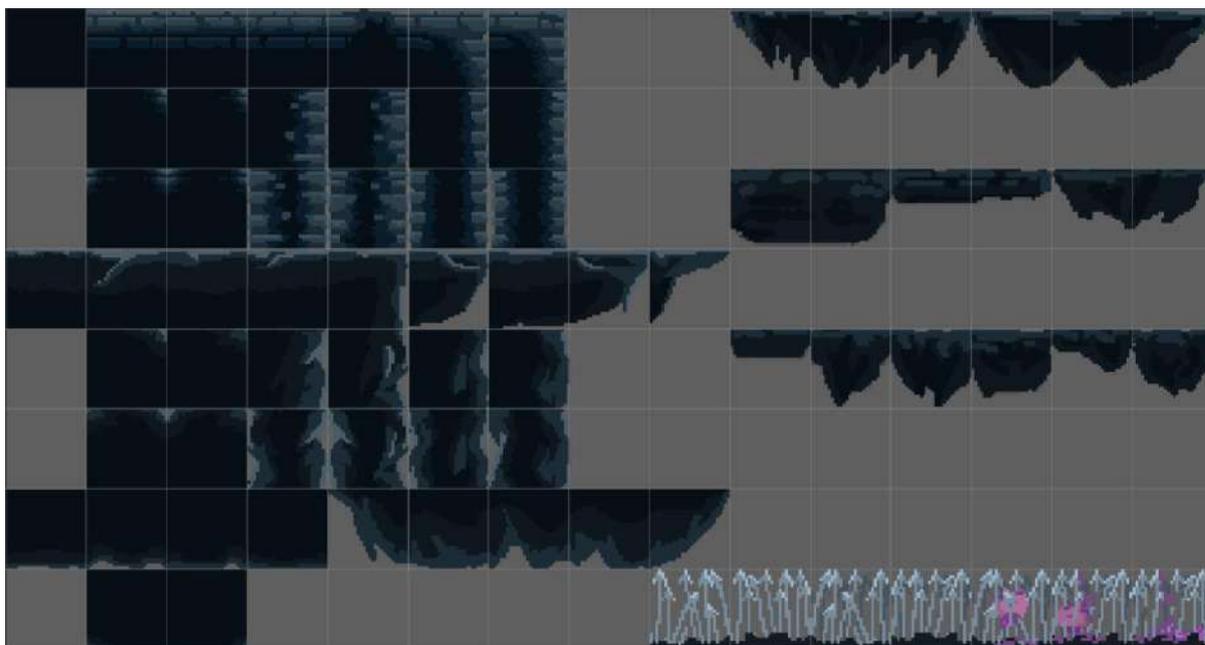


Figura 108. Segunda versão do *tilesets*.

Apesar dos benefícios, a criação de mapas utilizando de *tilesets* previamente desenhados, para a construção de todos os elementos visuais se tornou uma barreira, pois acabou por limitar os artistas que desejavam criar cenários únicos e detalhados para cada uma das áreas.

Assim, em discussões a equipe decidiu usar o *tiles* apenas para transmitir informações aos programadores e a arte seria feita separadamente. Os *tilesets* já criados foram mantidos para definir o *layout* da fase, áreas de colisão e áreas de dano. Além disso, criamos um *tileset* voltado apenas para orientação e informação, com cada *tile* indicando locais e elementos de interação específicos (Figura 109).

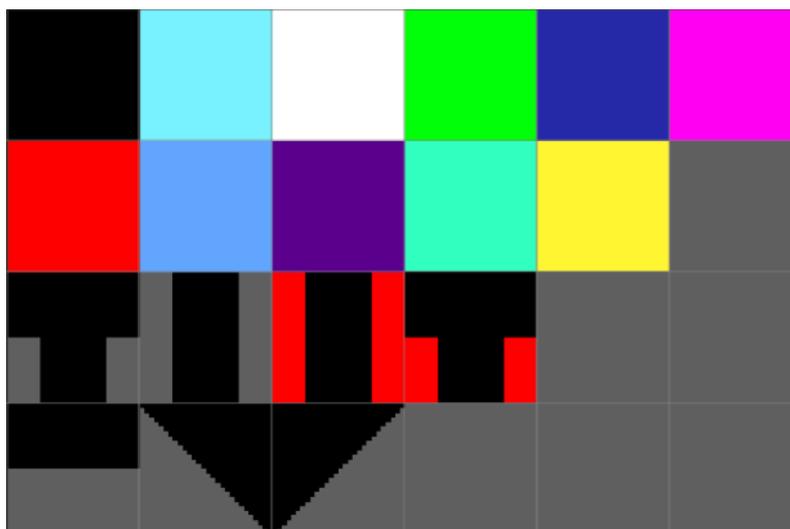


Figura 109. Primeira versão dos *tileset* de informação.

Os programadores puderam associar as funções aos *tiles* de informação, já que estes estavam corretamente posicionados no mapa pelos *designers*. A coloração dos *tiles* foi feita com base na lista de representações já criada no *level design*, como foi visto na Figura 109 apresentada anteriormente, facilitando a comunicação entre a equipe e acelerando o desenvolvimento do jogo.

Essa mudança permitiu que os *designers* entregassem o *level design* mais rapidamente para que os programadores pudessem iniciar a implementação. Além disso, ganharam mais tempo para desenhar as fases sem a limitação dos *tiles*, precisando apenas seguir o posicionamento e o *layout* já definido no TileD. O resultado dos cenários após essa etapa foi o seguinte: (Figura 110) todas as fases foram feitas seguindo essa padronização.

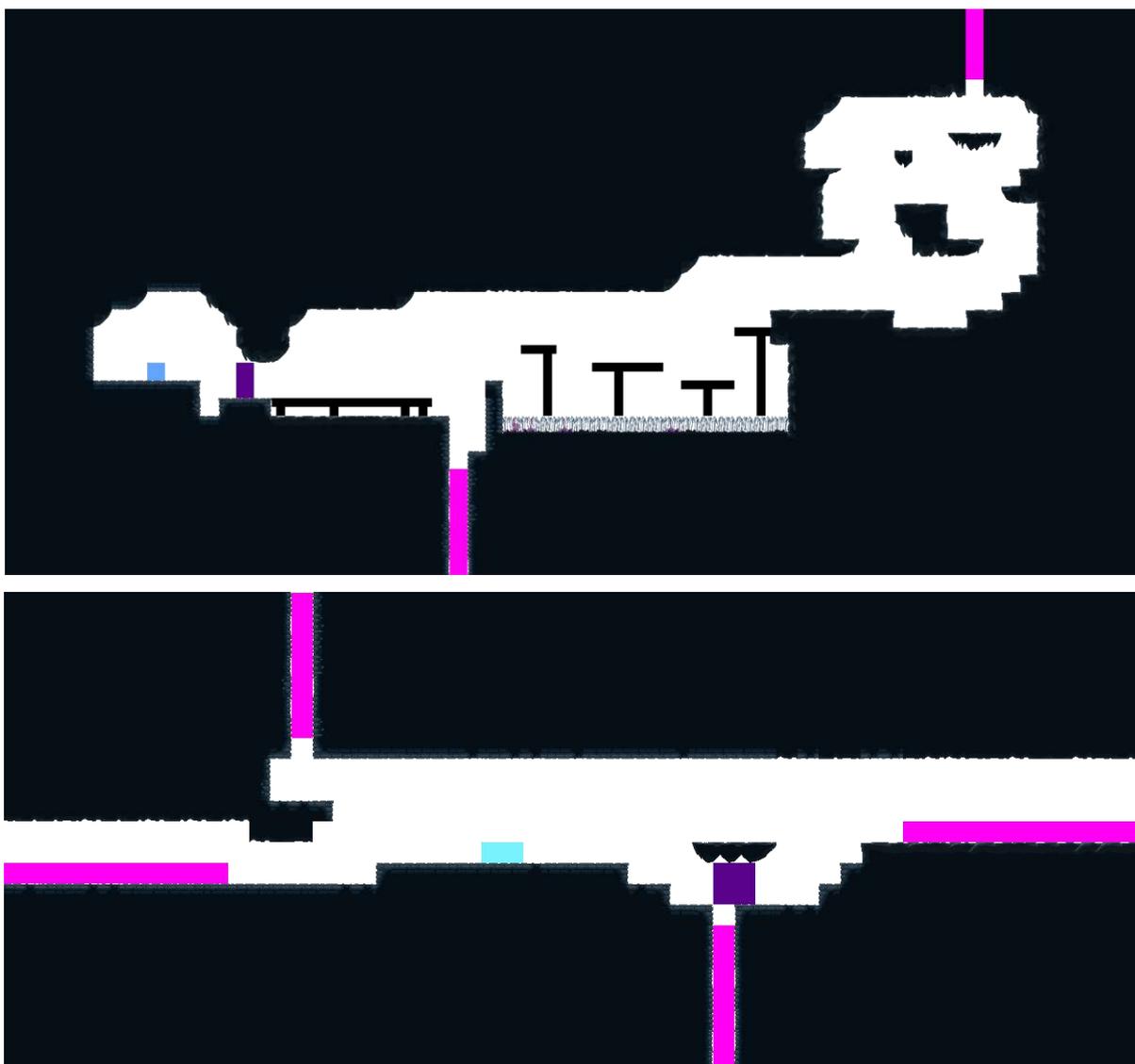


Figura 110. Exemplo de cenários exportados do TileD.

Com o *level design* das fases construído no TileD, foi possível implementar as fases e testar as plataformas, verificando se elas estavam devidamente posicionadas considerando a altura de pulo do personagem e a necessidade de movimentação. A partir de testes e *feedbacks* da própria equipe, ajustes necessários foram realizados (Figura 111).

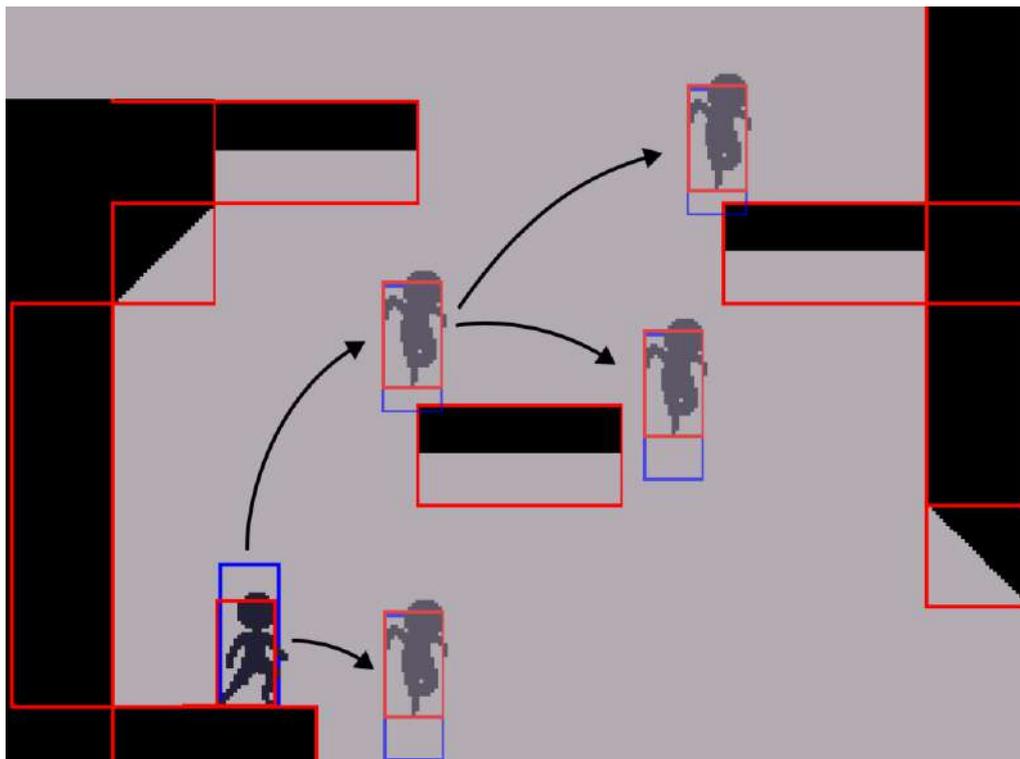


Figura 111. Testes de plataformas.

8.4 Uso de Camadas para a construção dos Cenários

Como mencionado anteriormente, devido às limitações dos *tilesets*, a equipe de arte e de programação debateram e decidiram não seguir com a criação da parte visual dos cenários no TileD. Em vez disso, optou-se pela produção de ilustrações inteiras para os cenários. Para implementar essa nova abordagem, criamos um sistema padronizado para a divisão das ilustrações dos mapas em uma série de camadas, que serviriam para posicionar e separar elementos visuais do jogo, e evitar que eles se sobreponham de maneira indesejada durante o *gameplay*, sendo pré-estabelecida a ordem de sobreposição.

Essa divisão foi feita antecipadamente, para que os programadores pudessem implementar o código e ajustar os valores, enquanto os artistas já teriam conhecimento dos limites na produção dos cenários, organizando-os de modo a se encaixar dentro do limite de camadas. Essa organização também permitiu a aplicação do efeito de *parallax*³⁷ com cada

³⁷ *Parallax* é a diferença de posição do objeto em relação a visão. Em um jogo 2D isso é feito de maneira com que camadas mais próximas passem mais rápido pela câmera que camadas mais distantes, criando uma sensação de tridimensionalidade em um mundo 2D.

camada podendo ter um valor modificador único, criando um efeito de profundidade mais detalhado, enriquecendo visualmente os cenários.

8.4.1 Nomenclatura e função das camadas

Foi decidido por estabelecer um total de 19 camadas, começando a partir da camada //0 até a camada //18. Das 19 camadas, 14 foram destinadas somente a elementos do cenário começando a partir da camada //0 (Figura 112).

_Background_FarAway, //0 Fundo (no parallax)

Figura 112. Nomenclatura das camadas *background faraway*.

Ela é mais distante, em todos os cenários ela é uma cor sólida e o seu propósito é preencher o fundo completamente e evitar partes vazadas no cenário. O grupo de camadas //1 até //7 fazem parte do fundo (Figura 113).

_Background_Far, //1 (parallax)
 _Background, //2 (parallax)
 _Background_Close, //3 (parallax)
 _Background_VeryClose, //4 (parallax)
 _Scenery_Far, //5 (parallax)
 _Scenery, //6 (parallax)
 _Scenery_Close, //7 Elementos do fundo (parallax)

Figura 113. Nomenclatura das camadas *background*.

Todas elas passam atrás do jogador, podendo ser utilizadas para imagens estáticas ou animações de cenário. Quanto menor o número, mais distante e menor a velocidade com que a camada se movimenta em relação à velocidade da câmera, criando, assim, o efeito de *parallax*. O valor exato com que cada camada se movimenta foi decidido individualmente para cada fase.

O grupo //8 até //10 (Figura 114), é o mais próximo do jogador e cada uma tinha um propósito específico.

```

_Environment_Far, //8 (no parallax)
_Environment, //9 : Camada dos tiles (no parallax)
_Environment_Close, //10 (no parallax)

```

Figura 114. Nomenclatura das camadas *environment*.

A camada //8 foi destinada a todo o cenário próximo, construções e elementos que passam atrás do jogador, mas não possuem *parallax* e se movem na mesma velocidade da câmera. A camada //9 é a camada dedicada a ajustes ou adição de detalhes no tiles de colisão e/ou dano. A camada //10 foi destinada a objetos interativos do cenário com ou sem animação, como barreiras mágicas, blocos que podem ser arrastados etc.

É importante ressaltar que as imagens que ocupariam esse grupo de camadas obrigatoriamente precisavam ter o mesmo tamanho em *pixels* dos layouts criados no TileD. Com exceção das animações, cuja elaboração será explicada mais adiante.

As camadas //11 até a //15 foram reservadas para o posicionamento do personagem jogável, inimigos e NPCs. Já das camadas //16 até a //18 (Figura 115), temos elementos que passariam a frente do personagem jogável.

```

_Characters, //11 até //15 (no parallax)
_ForeGround, //16 (no parallax)
_ForeGround_Close, //17 (parallax)
_ForeGround_VeryClose, //18 (parallax)

```

Figura 115. Nomenclatura das camadas *characters* e *foreground*.

A camada //16 ainda se movimentava na mesma velocidade da câmera, ou seja, não apresentaria *parallax*. As demais camadas se movimentam mais rápido que a câmera, criando, assim, o efeito de *parallax*.

8.4.2 Exportar mapas finalizados

Um importante detalhe para a exportação dos mapas finalizados é que eles precisavam estar perfeitamente alinhados com o *layout* criado no TileD para que a estratégia de posicionar os *tiles* de informação funcionasse. Como pode ser visto na Figura 116.

A área de colisão em preto, os blocos de dano em vermelho e os blocos marcados na cor ciano, que representa uma barreira que só pode ser aberta utilizando o ataque mágico, se alinham com o mapa desenhado. Além de outras informações importantes como o bloco marcado em branco sinalizando a posição de um dos itens da poção e os blocos em rosa marcando a área que ao entrar em contato com o personagem jogável, ativa a transição de mapa.

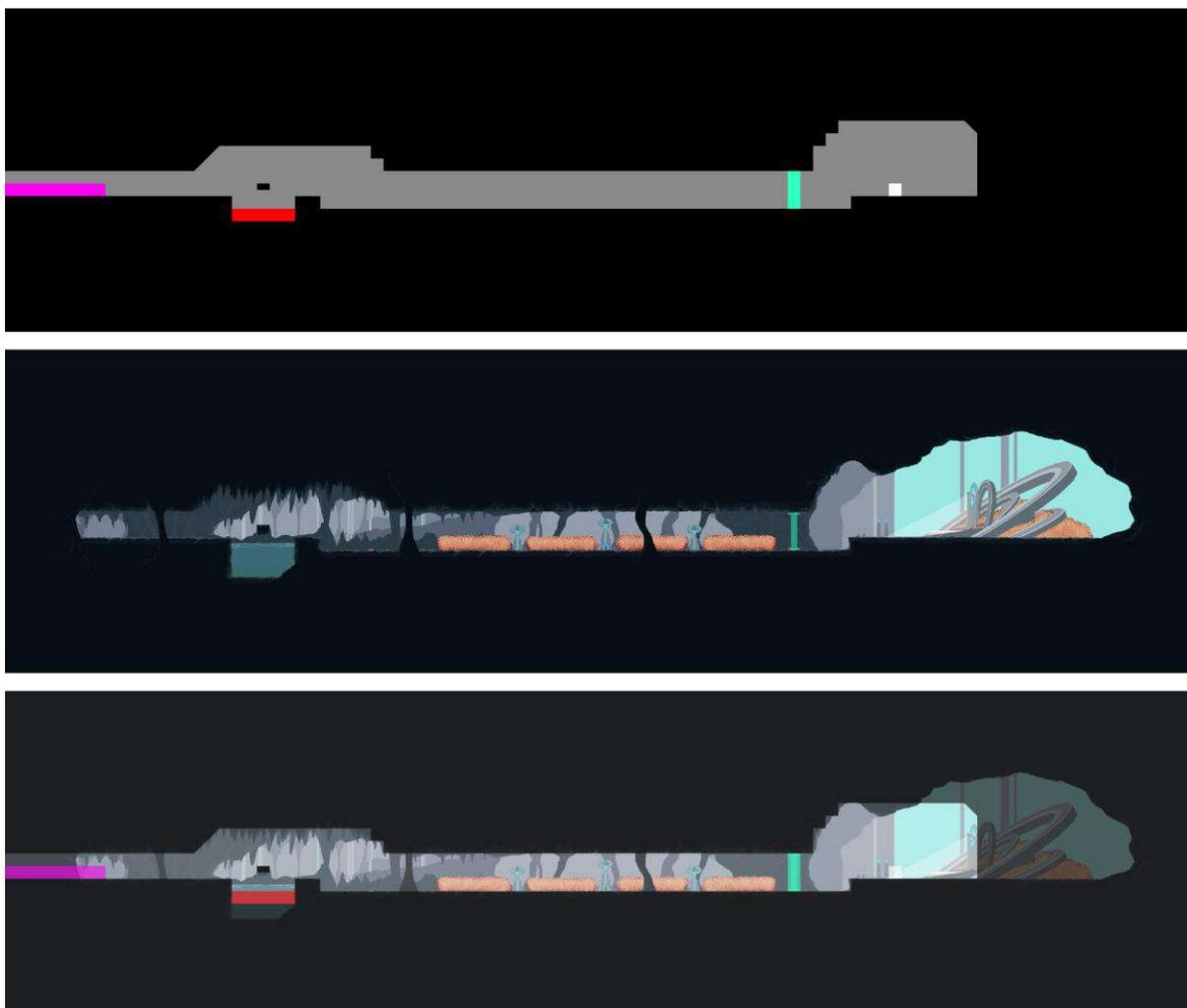


Figura 116. Comparação *level design* com as camadas.

A numeração das camadas foi também adotada na nomenclatura dos arquivos, facilitando a identificação e implementação pela equipe de programação. Apenas as camadas //0 e //9 são obrigatórias para todos os cenários, enquanto o uso de quantas e quais camadas fica a critério do artista. O número de camadas foi deliberadamente pensado para ser mais do que suficiente.

Além disso, caso necessário, mais de uma imagem pode ser colocada na mesma camada, desde que as áreas desenhadas das imagens não se sobreponham. Permitimos isso para casos no qual o artista não consiga concentrar a arte em uma única imagem, mas deseja que ambas tenham o mesmo valor de *parallax*. Abaixo está um exemplo de um cenário pronto (Figura 117) e como ficou a separação na estrutura de camadas definida por nós (Figura 118).

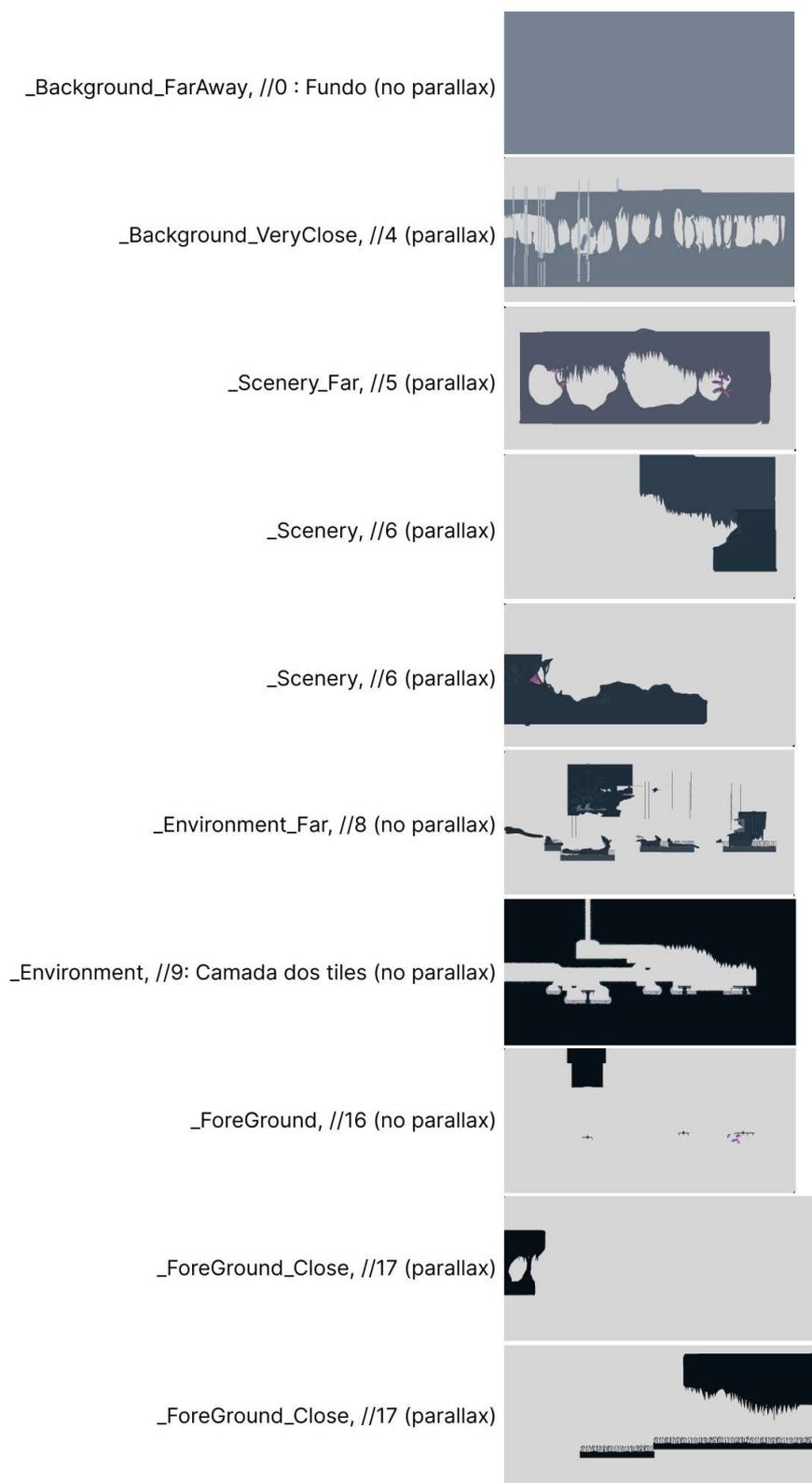


Figura 117. Cenário dividido e organizado no sistema de camadas.

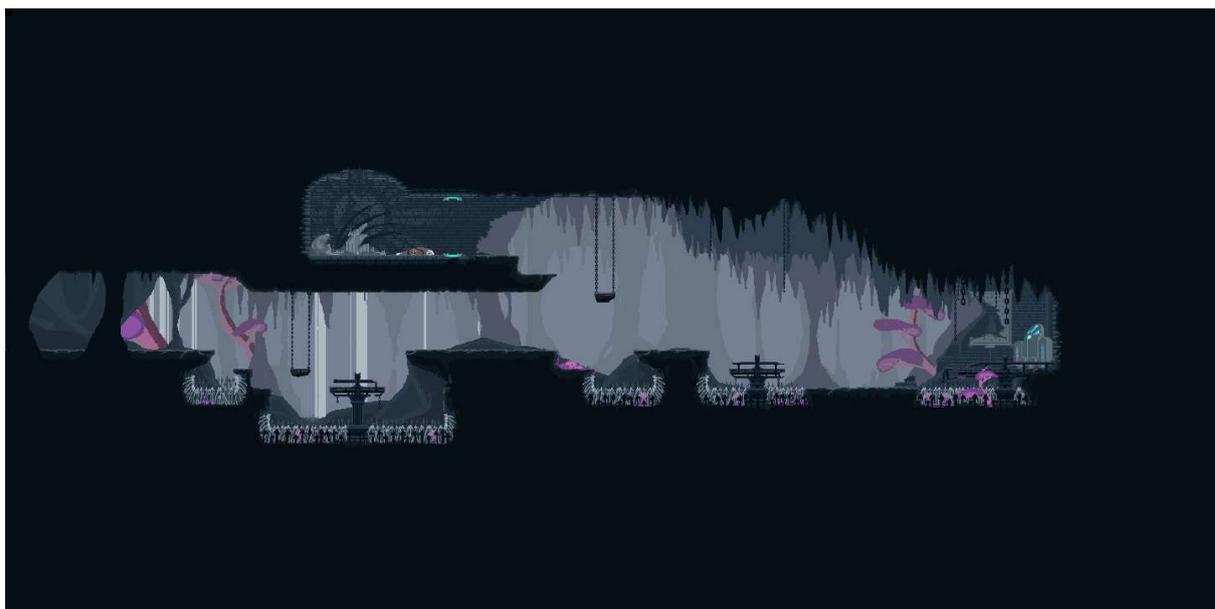


Figura 118. Resultado após a combinação e alinhamento das camadas.

Para as animações presentes no cenário, foram utilizadas *spritesheets*, posicionadas por coordenadas em *pixels* fornecidas pelos artistas. Essa coordenada é marcada a partir da localização do primeiro *pixel* no canto superior direito da *spritesheet* em relação ao mapa (Figura 119).

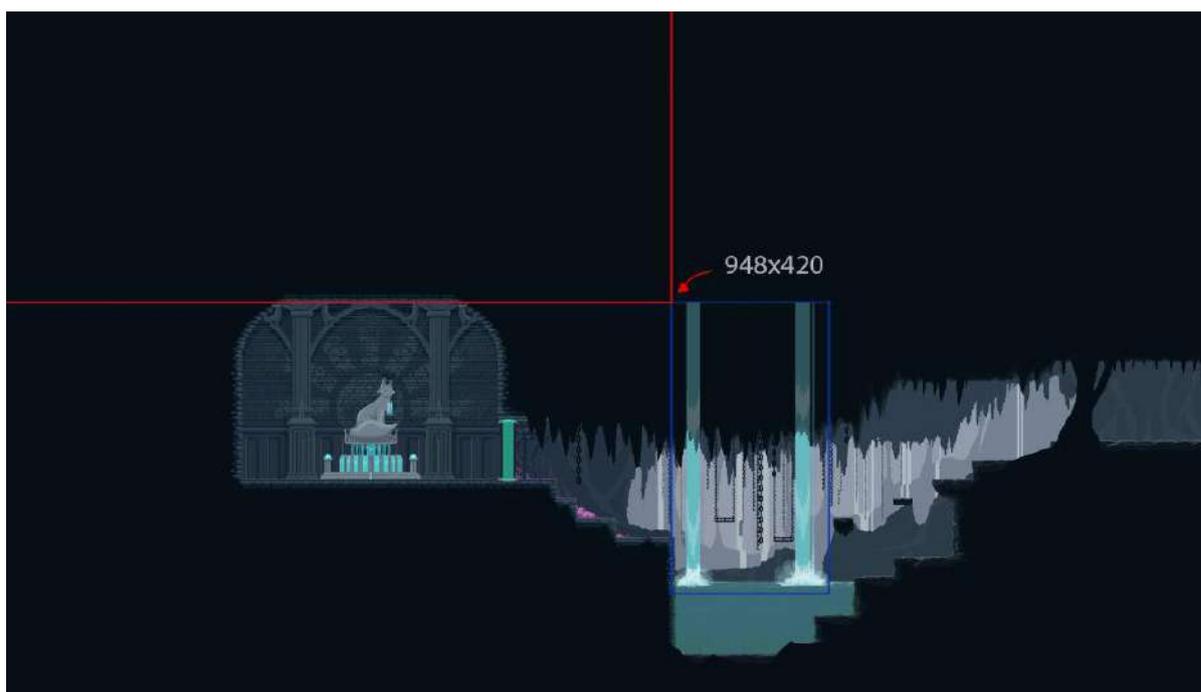


Figura 119. Posicionamento de animações nos cenários.

Todo esse sistema de regras e definições feito pelo time permitiu um trabalho mais independente e coordenado de ambas as partes. Essa decisão permitiu que o jogo tivesse um cenário mais extenso e detalhado.

8.5 Assets e Finalização do Cenário

Com todas as predefinições estabelecidas, os artistas de cenário tiveram liberdade para trabalhar em cada cenário da maneira que achassem melhor para atingir o resultado desejado. Podendo adicionar animações, detalhes únicos e personalizar cada área para alcançar o melhor resultado visual possível. Os *assets* foram produzidos de maneira intercalada com o Photoshop, que, com algumas configurações, permitiu criar a *pixel art* sem grandes dificuldades, com o Aseprite, que, também foi onde todos os *assets* foram animados. Os cenários foram finalizados e divididos em camadas primariamente no Photoshop.

8.5.1 Criação de Assets do cenário

Antes de partir para a finalização dos cenários, a equipe de artes focou na criação de alguns *assets* que poderiam ser repetidos ao longo de diversos cenários. Uma lógica semelhante à dos *tilesets*, mas sem precisar se limitar ao tamanho do *tile* ou ter que dividir o *asset* em vários *tiles* menores como, por exemplo, para o mapa na floresta foram feitas diversas variações de árvores, plataformas (Figura 120), arbustos, flores e pedras (Figura 121), além de elementos menos detalhados para preencher o fundo (Figura 122).

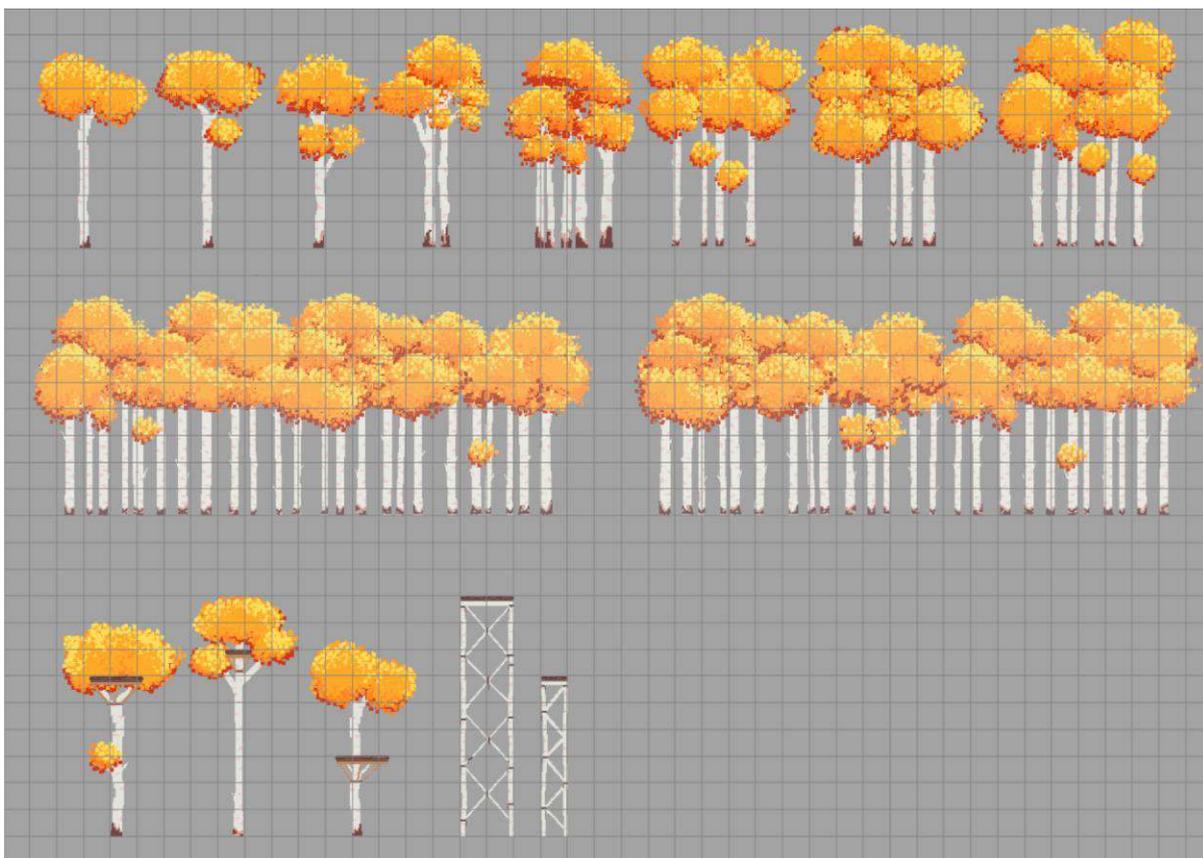


Figura 120. Assets de árvores e plataformas.

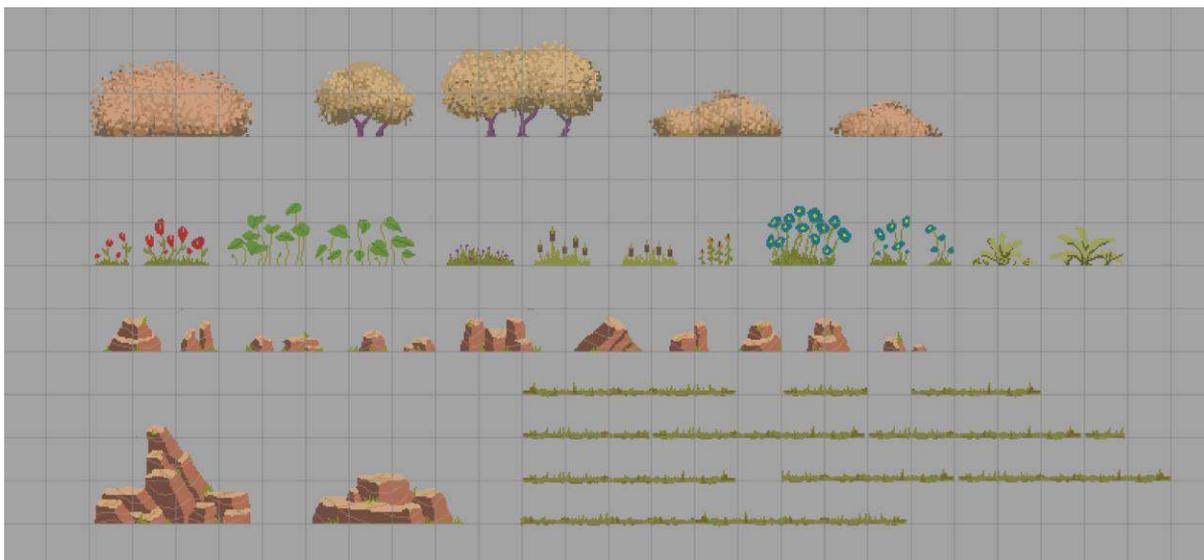


Figura 121. *Assets* de vegetação e rochas.

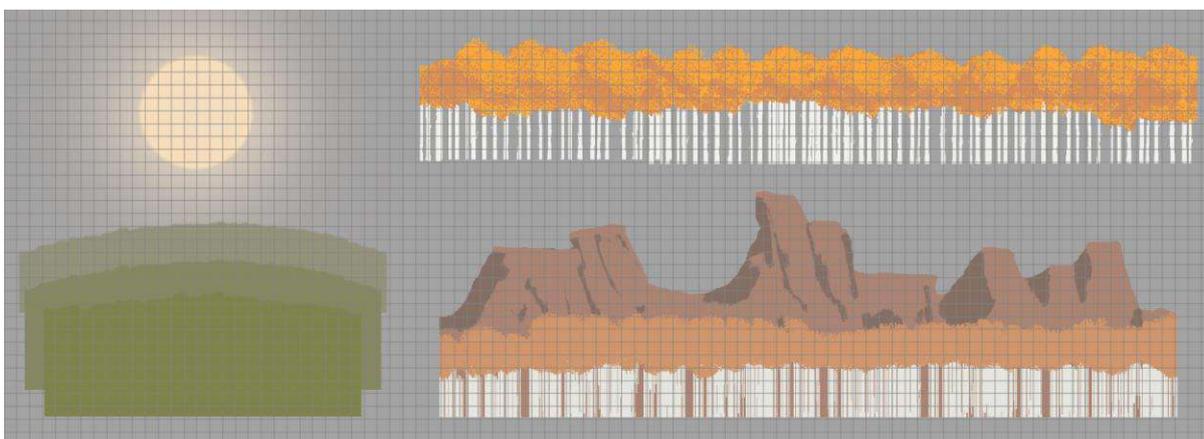


Figura 122. *Assets* para o *background*.

Também para a floresta foram criados alguns *assets* únicos, que aparecem apenas em cenários específicos, como a casa do protagonista, que também contou com outros feitos para colocar em volta da casa para criar uma imersão maior, parecer que a casa realmente estava habitada (Figura 123). Para a casa do alquimista também foram desenvolvidos *assets* únicos tanto para a casa quanto para os portões, que posteriormente seriam fechados (Figura 124).



Figura 123. Asset casa do protagonista exterior.



Figura 124. Asset mansão do Alquimista.

Para representar a doença espalhando pela floresta e pelo subterrâneo, desenhamos alguns *assets* para serem utilizados na criação dos cenários como árvores e vegetação infectada (Figura 125), cogumelos de diversos tamanhos, também utilizamos o inimigo infectado para criar *assets* de inimigos mortos para os espalharmos pelo mapa (Figura 126) e (Figura 127).

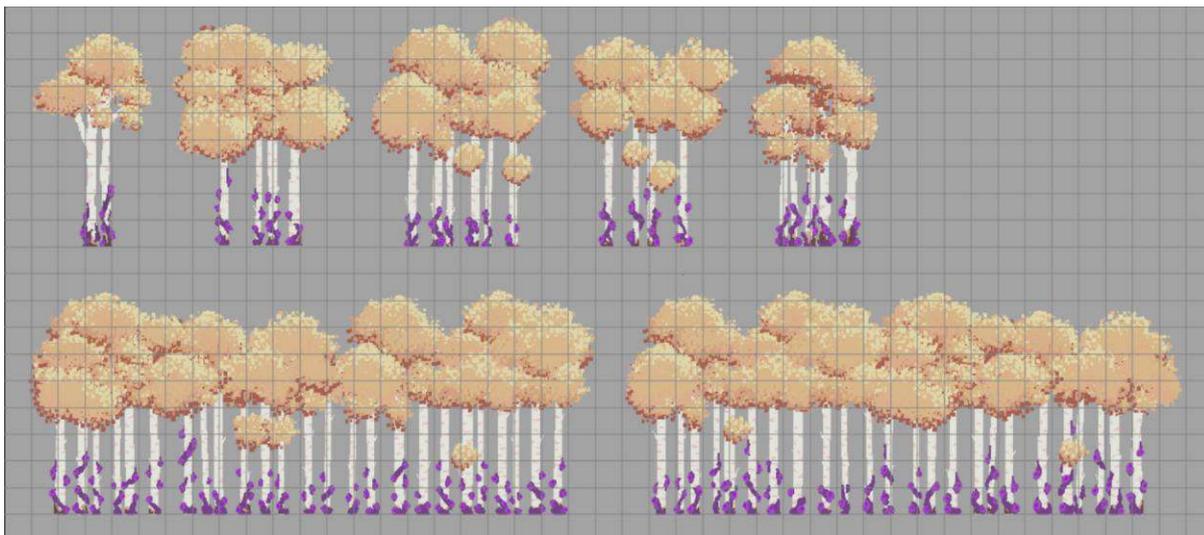


Figura 125. *Assets* de árvores infectadas.

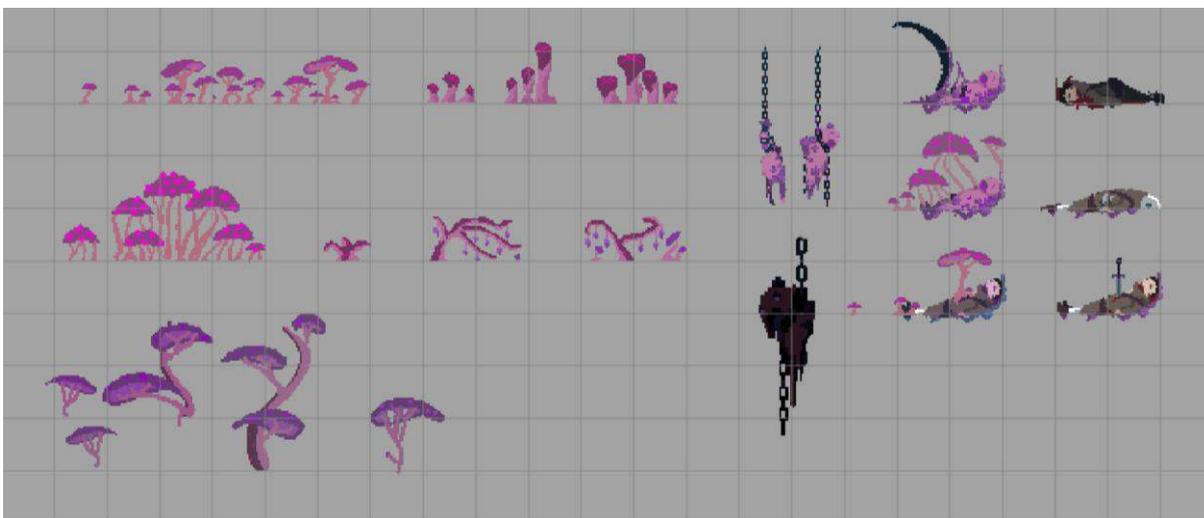


Figura 126. *Assets* de elementos da doença 1.

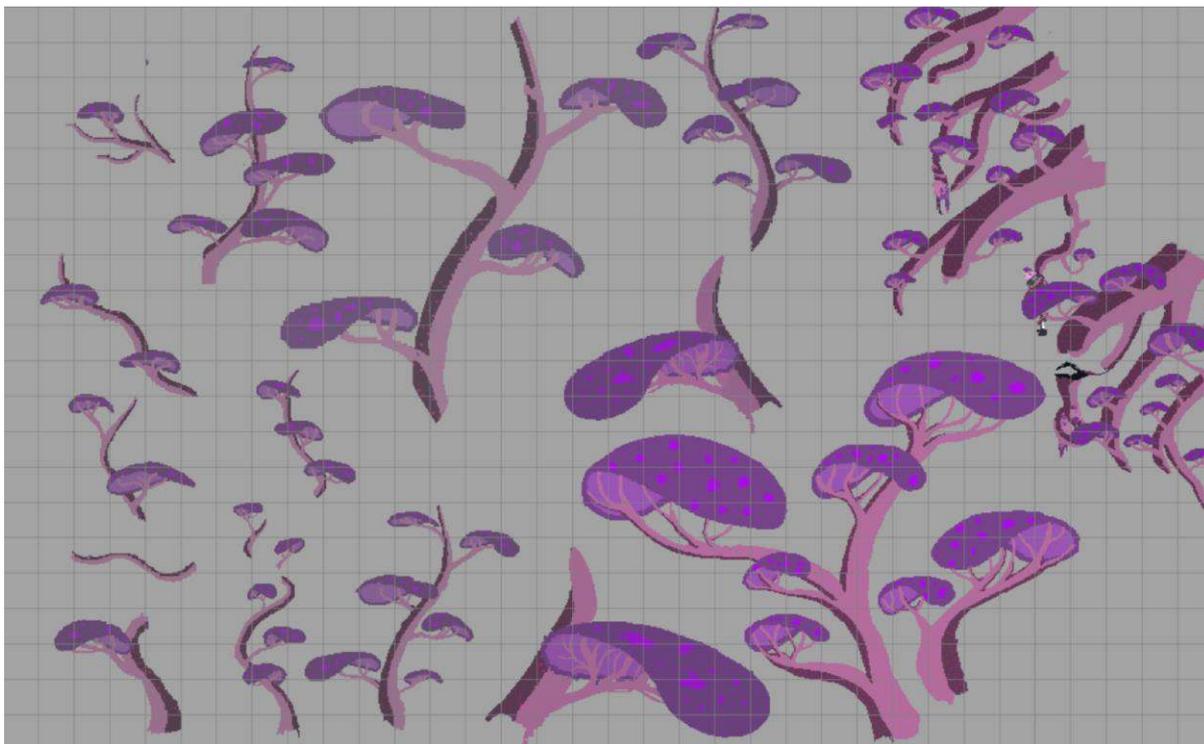


Figura 127. *Assets* de elementos da doença 2.

Para o subterrâneo, foram criados diversos *assets* para representação da cidade destruída, foi desenhando vários pilares, plataformas, correntes (Figura 128), paredes, jaulas e prisões (Figura 129).

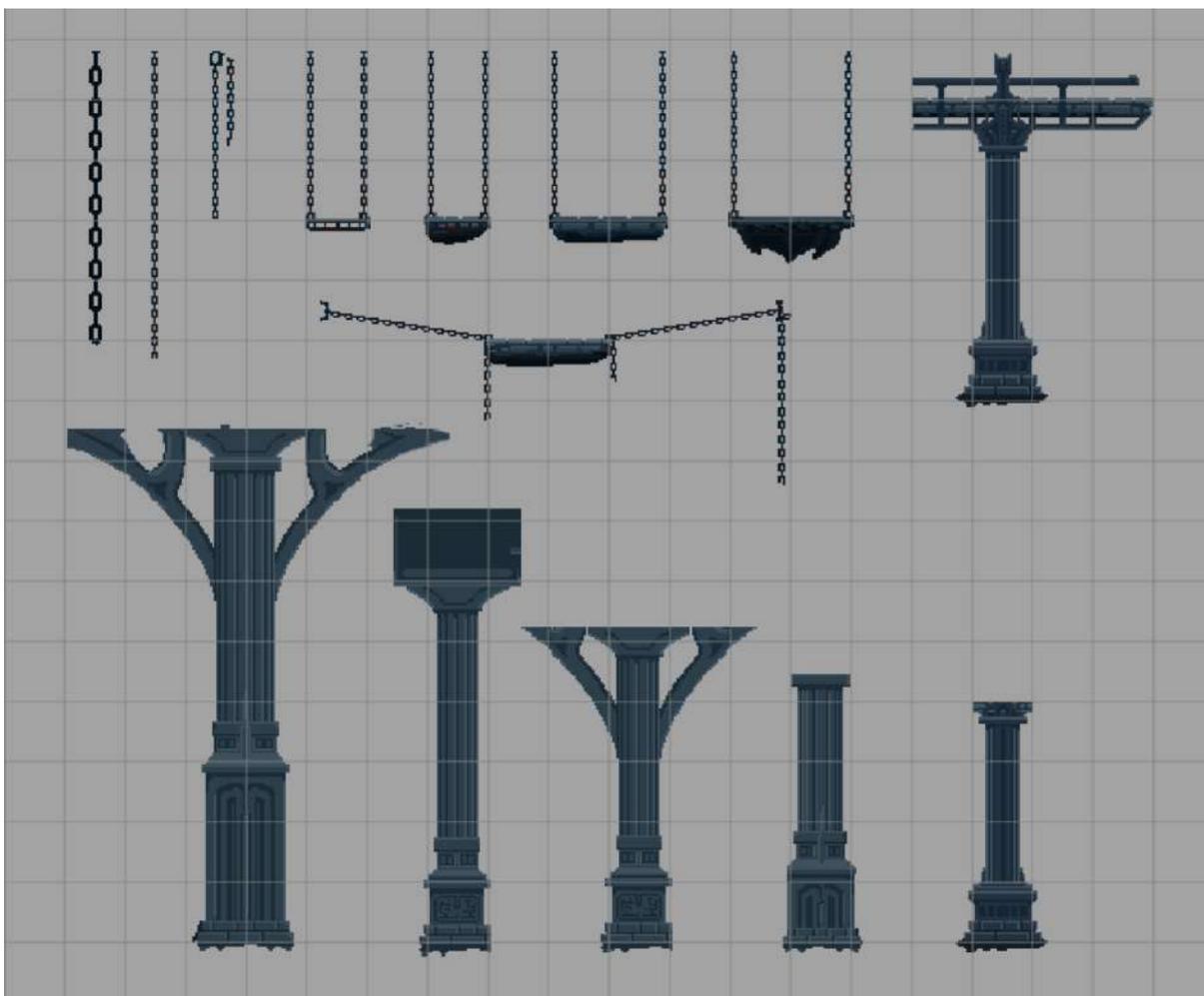


Figura 128. *Assets* de pilares e plataformas.

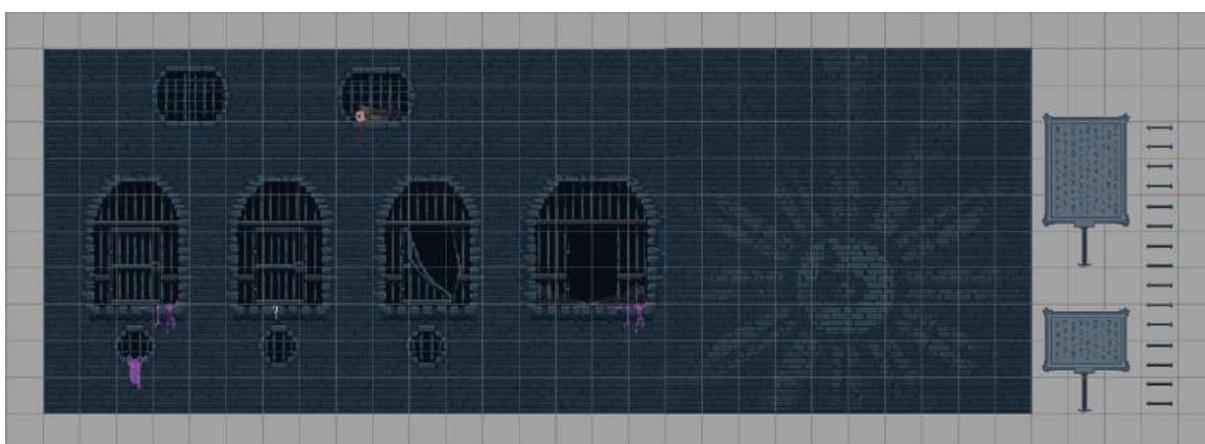


Figura 129. *Assets* de paredes, jaulas e decorações.

E também foram criados *assets* únicos, mas que também poderiam se repetir, como, por exemplo, o *checkpoint*, o mecanismo que abre a porta, às pedras que podem ser empurradas com o ataque carregado e as barreiras que podem ser abertas com o ataque mágico (Figura 130).



Figura 130. *Assets* para o *checkpoint*, mecanismo, barreiras e outros.

Para a cidade subterrânea também foram desenvolvidos uma série de *assets* mais complexos e únicos com o objetivo de destacar algumas áreas específicas do mapa como, por exemplo, o local na qual o jogador chegaria primeiro na cidade (Figura 131), o portão de saída da cidade (Figura 132), e todos os quatro locais onde o jogador encontraria os itens da poção, sendo eles, uma porta mágica inacessível (Figura 133), uma árvore milenar (Figura 134), um caixão de vidro com uma figura misteriosa dentro (Figura 135) e uma fonte com uma grande raposa segurando em sua boca um candelabro azul (Figura 136).

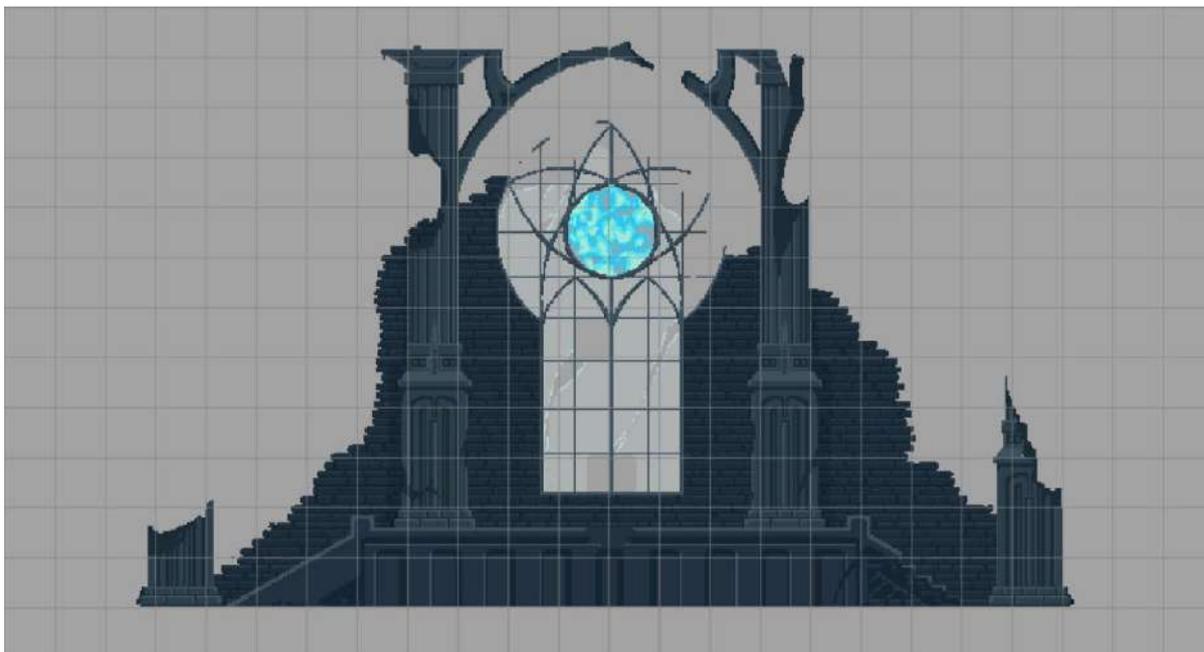


Figura 131. *Asset igreja destruída.*

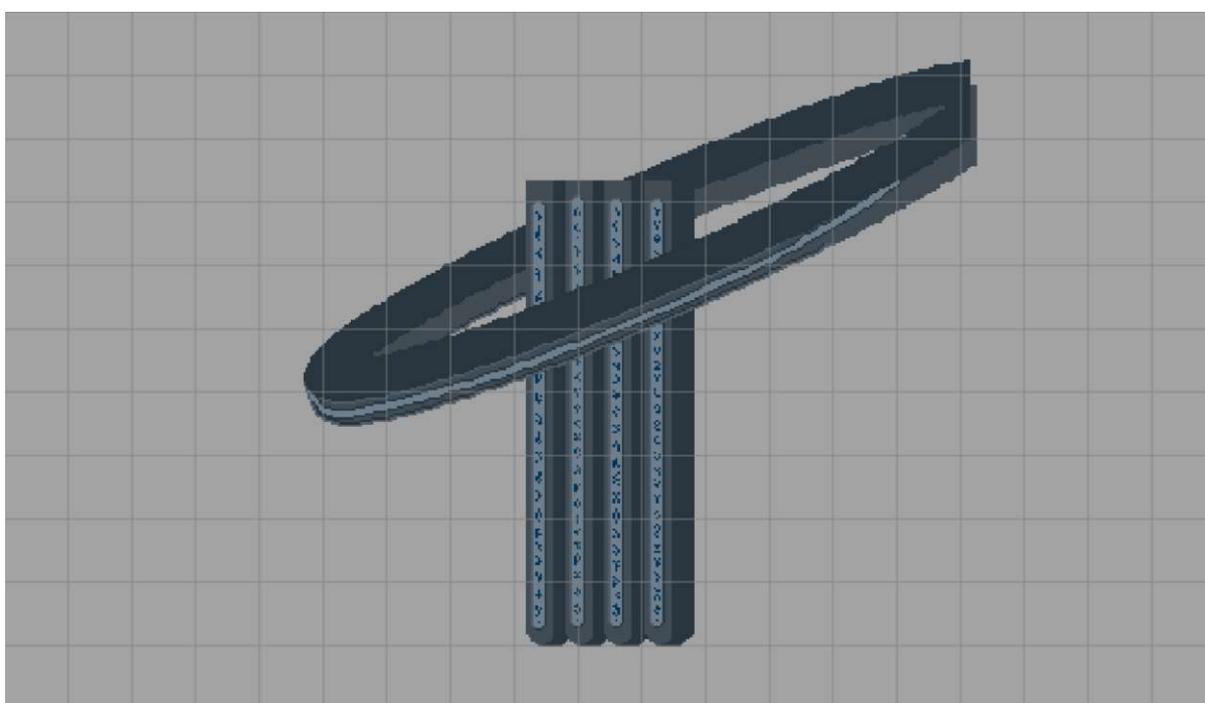


Figura 132. *Asset portão de saída da Cidade Subterrânea.*

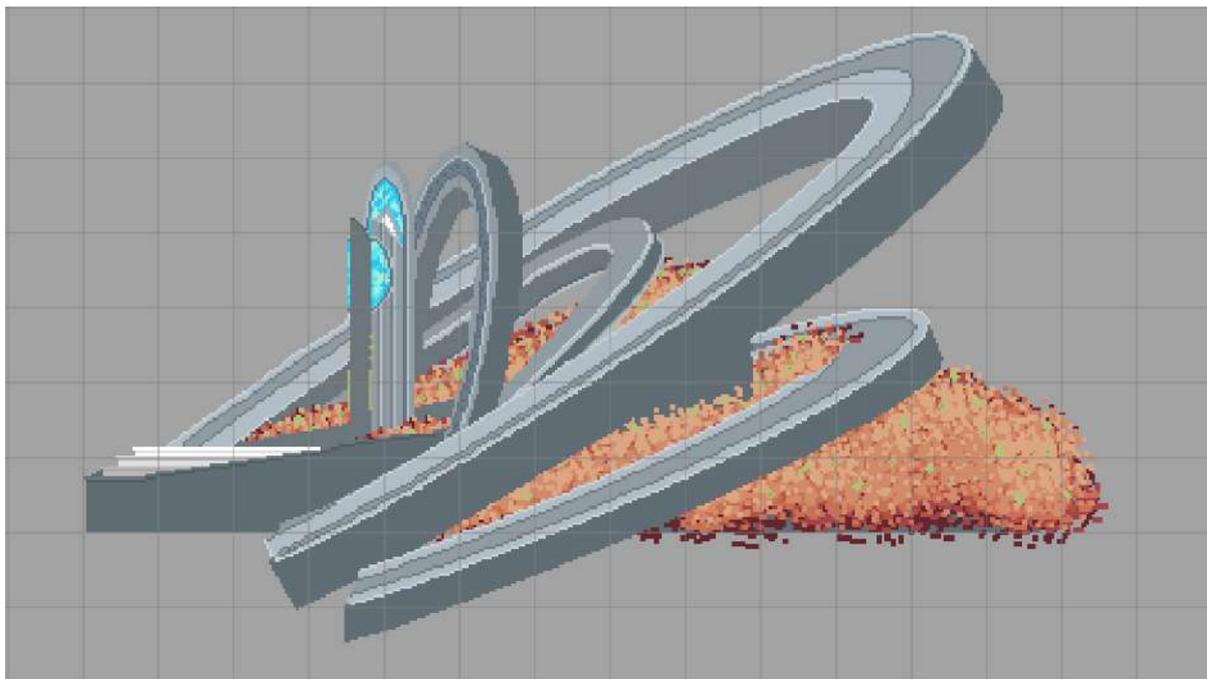


Figura 133. *Asset da porta mágica inacessível.*



Figura 134. *Asset da árvore milenar.*

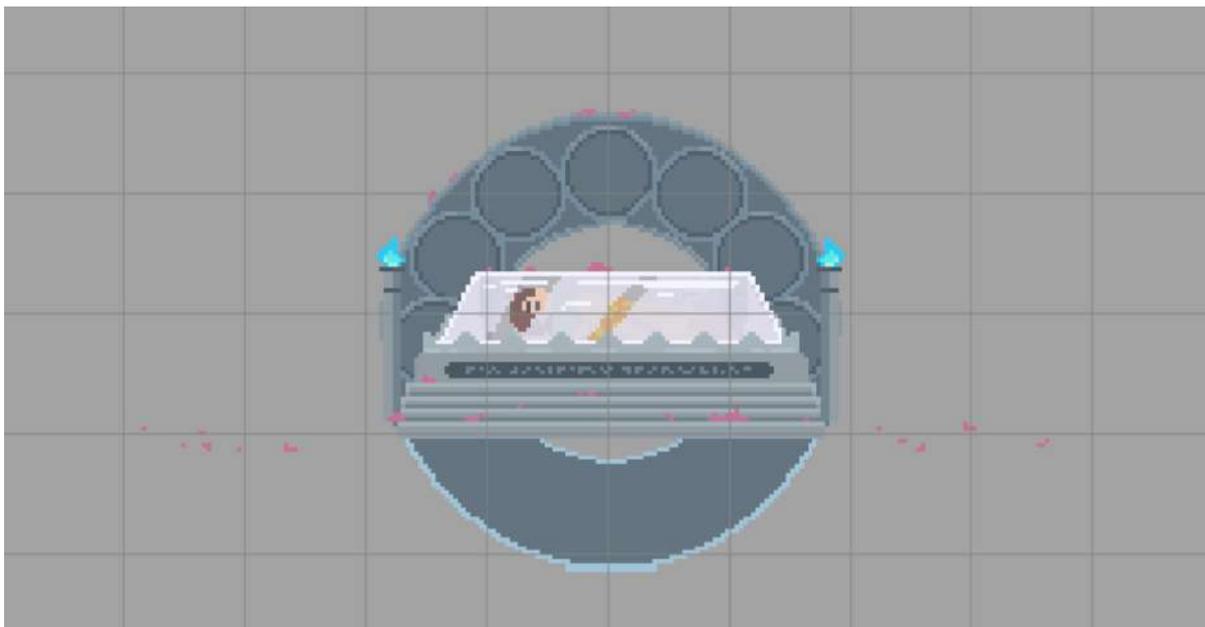


Figura 135. *Asset* do caixão de vidro.

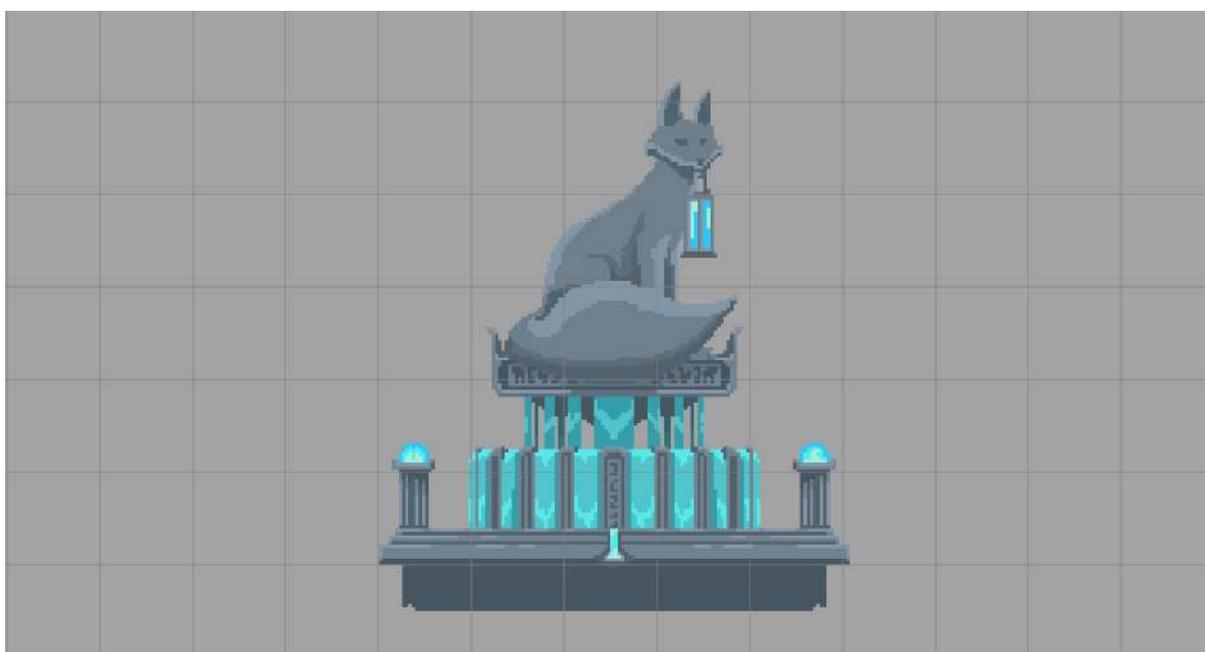


Figura 136. *Asset* da fonte da grande raposa.

Importante ressaltar que essa etapa foi realizada em conjunto com o detalhamento e finalização dos cenários. Na medida que os cenários foram detalhados, a criação e o compartilhamento dos *assets* foi sendo feita, o que progressivamente acelerou o processo de finalização dos cenários, à medida que a biblioteca de *assets* foi se expandindo.

8.5.1 Animação de Assets

Além disso, alguns objetos do cenário foram animados. Algumas para indicar a interação com o jogador, como o mecanismo presente no mapa do subterrâneo que abre a porta de saída (Figura 137) a animação é ativada quando o jogador interage com o objeto, ou o portão de saída da cidade que ativa somente se o jogador tiver cumprido todos os objetivos necessários (Figura 138).



Figura 137. *Spritesheet* do mecanismo sendo ativado.



Figura 138. *Spritesheet* da porta de saída da cidade subterrânea abrindo.

Outras animações foram feitas para enriquecer visualmente os cenários, como as folhas de árvores (Figura 139), cachoeiras (Figura 140), o *checkpoint* (Figura 141) entre outras.



Figura 139. Exemplo de *spritesheet* das árvores.



Figura 140. Exemplo de *spritesheet* para as cachoeiras.

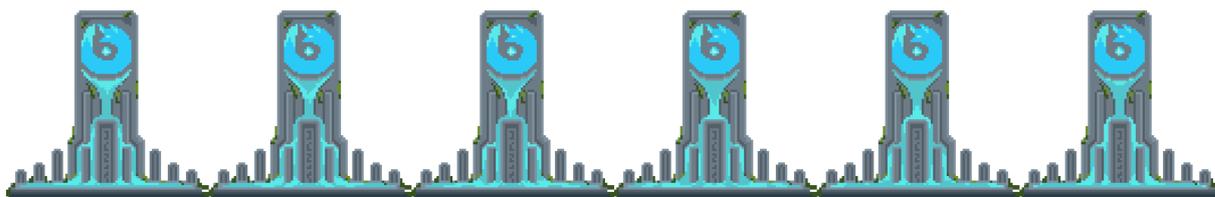


Figura 141. *Spritesheet* do *checkpoint*.

Também foi criada uma animação para destacar os objetos coletáveis pelo jogador, como os itens para a poção ou as habilidades especiais adquiridas no mapa (Figura 142).

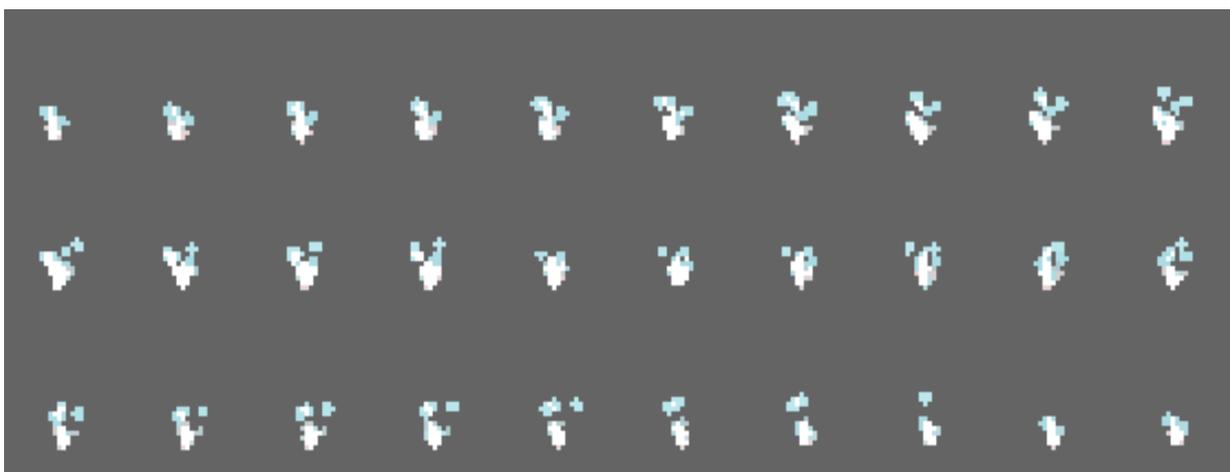


Figura 142. *Spritesheet* dos itens da poção.

8.5.2 Detalhamento e Finalização dos Cenários

Nessa etapa, os cenários foram trabalhados em todos os detalhes com base nos *layouts* já criados. Os artistas tiveram total liberdade para criar novos *assets* ou utilizar, combinar e modificar os existentes, conforme necessário, para tornar cada cenário interessante. Cada fase foi individualmente detalhada para que cada uma seja única, ao mesmo tempo em que se tomou cuidado para garantir coesão com o restante dos cenários (Figura 143) (Figura 144)



Figura 143. Cenário finalizado Floresta na superfície.



Figura 144. Cenário finalizado Cidade Subterrânea.

Os *assets* únicos foram posicionando nas suas respectivas fases e foram mesclados com o restante do cenário utilizando *assets* genéricos ou até desenhando novos para destacar aquela área. Um exemplo é a casa do protagonista, que em seu exterior, em volta dos *assets* únicos, foram combinadas diversas plantas e pedras para ajudar a mesclá-los ao ambiente (Figura 145). A casa do protagonista ainda teve o interior completamente desenhado (Figura 146).

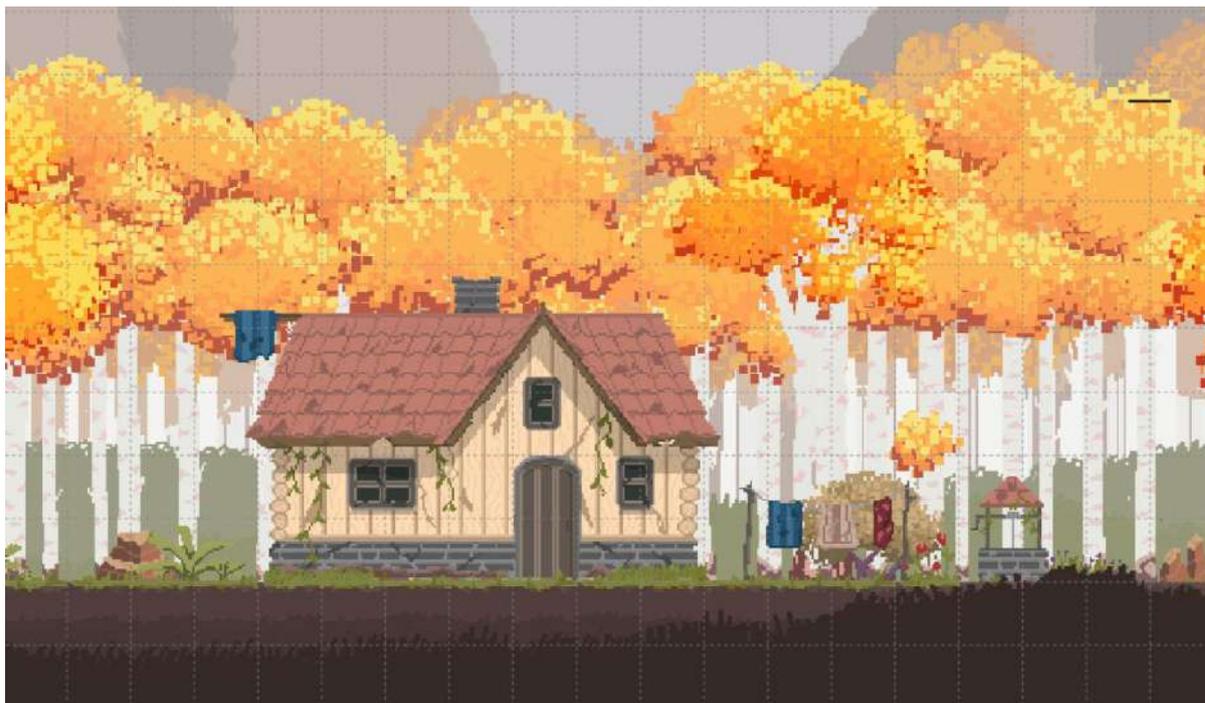


Figura 145. Cenário finalizado casa do protagonista exterior.

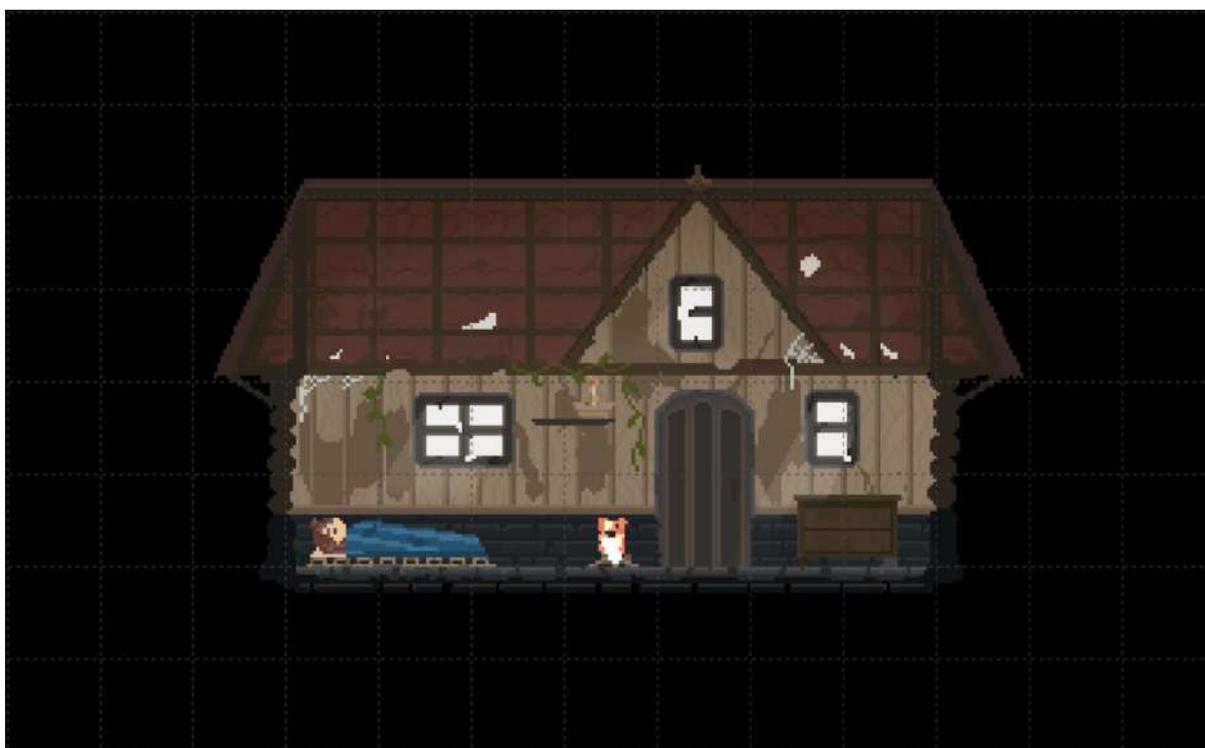


Figura 146. Cenário finalizado casa do protagonista interior.

Outros *assets* únicos também foram incluídos nos cenários, como, por exemplo, a entrada e o portão de saída da cidade subterrânea (Figura 147), *checkpoints*, mecanismo da

porta (Figura 148), além de todos os locais onde o jogador encontrará itens para a poção (Figura 149).



Figura 147. Cenário finalizado entrada e saída da Cidade Subterrânea.

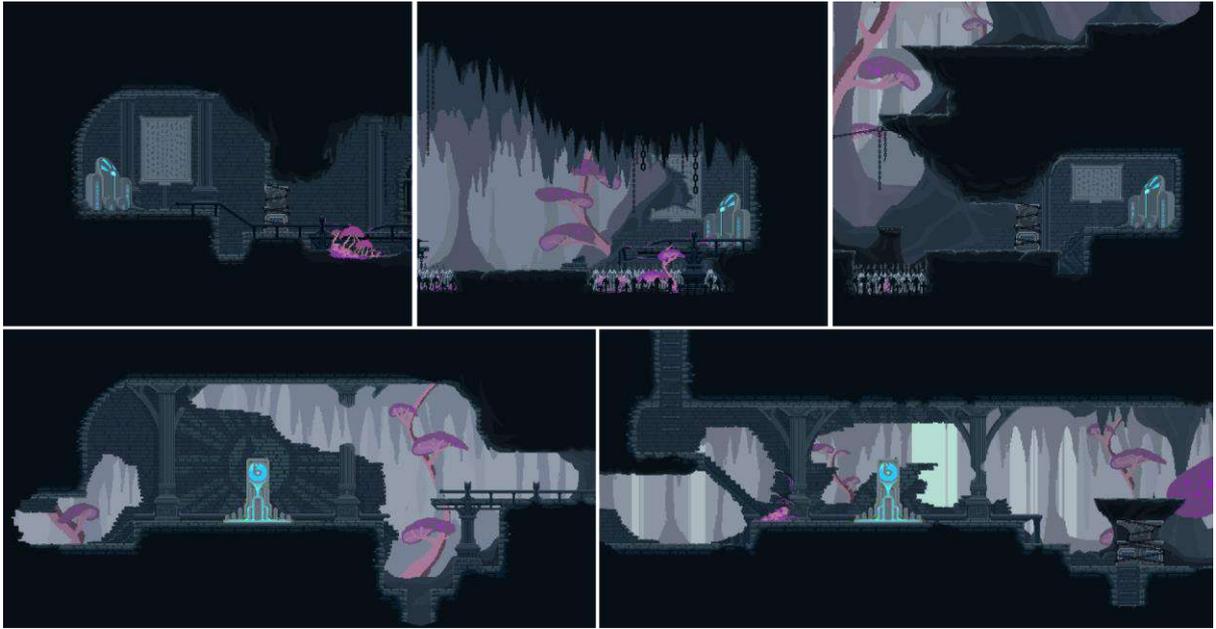


Figura 148. Cenário finalizado mecanismos e *checkpoints*.

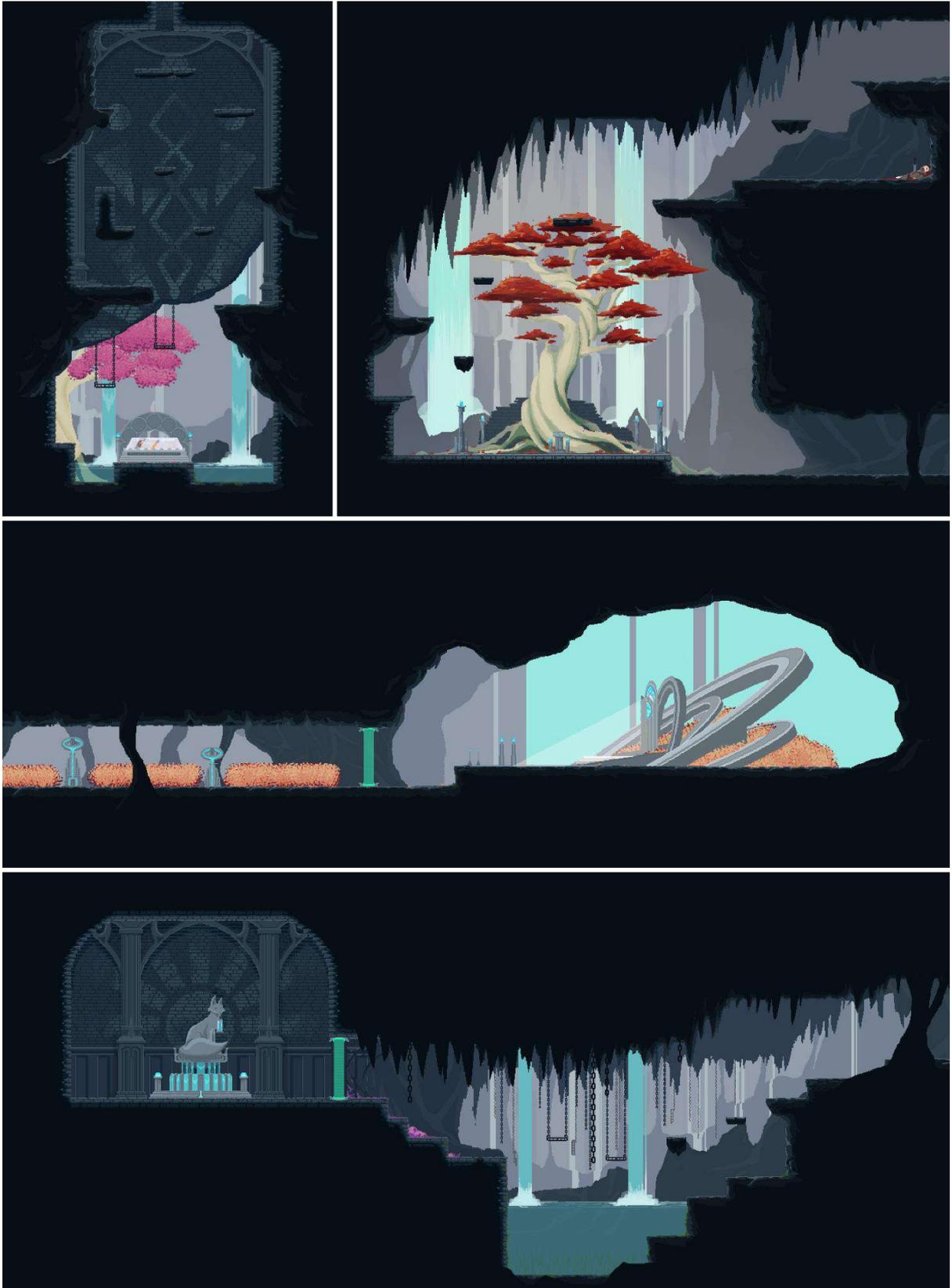


Figura 149. Cenário finalizado locais de coleta dos itens da poção.

Um ponto interessante a destacar é que o cenário da casa do alquimista possui duas variações. A primeira variação (Figura 150) é apresentada quando o jogador passa por essa área no início do jogo e o alquimista dá a missão de entrar no subterrâneo e coletar os itens para a poção. A segunda variação (Figura 151) acontece quando você retorna à área após ter coletado todos os itens e encontra o alquimista novamente.



Figura 150. Cenário finalizado mansão do Alquimista.



Figura 151. Cenário finalizado mansão do Alquimista infectada.

A troca de variações deste cenário foi feita para incorporar mais elementos do enredo nos cenários. Isso demonstra que, quando o jogador coleta todos os itens e abre a saída da cidade subterrânea, a doença também escapa e se espalha pela superfície.

8.6 Modificações propostas aos mapas e cenários

Após a implementação, testes e coleta de *feedback*, identificamos no *level design*, concepção e finalização dos *assets* diversas falhas na execução e questões que não foram consideradas anteriormente, além de aspectos que não atingiram o resultado desejado. E com a migração de *engine* e a necessidade de transferir, e em alguns aspectos, reconstruir as fases, surgiu o desejo de revisar e retrabalhar esses problemas previamente encontrados. Além disso, ao abrirmos novamente discussões sobre outras áreas do desenvolvimento do jogo e começarmos a realizar modificações, como nas mecânicas e narrativas, tornou-se evidente a necessidade de retornar aos mapas e cenários para ajustes.

8.6.1 Desenvolvimento de *Concepts* adicionais

Dessa vez começamos a expansão do mapa em conjunto com a concepção das novas áreas. A equipe de mapas iniciou a criação do *layout* mais rico em detalhes, utilizamos o que já havíamos desenhado para expandi-lo e ir criando a partir dele novas áreas e novas salas de forma simples, ampliando, modificando e desenhando de maneira livre e lúdica a aparência daquela região (Figura 152).



Figura 152. Expansão do *layout* do mapa.

O desenvolvimento desse *concept* se deu de maneira simples e lúdica, já que o estilo artístico do jogo já está decidido, o foco dessa vez foi tornar cada uma das áreas mais distintas visualmente, cada uma possuindo um *mood* próprio que se relaciona com o enredo e conversa com o universo criado do jogo. Posteriormente, expandimos ainda mais a imagem, adicionando regiões completamente novas. Além disso, com base nos novos elementos desenvolvidos para

a história do jogo, cada uma das áreas do mapa foi nomeada, detalhada sua função no universo fictício do jogo e um pequeno enredo atribuído a ela, além de ter sido coletado imagens que representam o *mood* serviram como inspiração para criação delas (Figura 153).

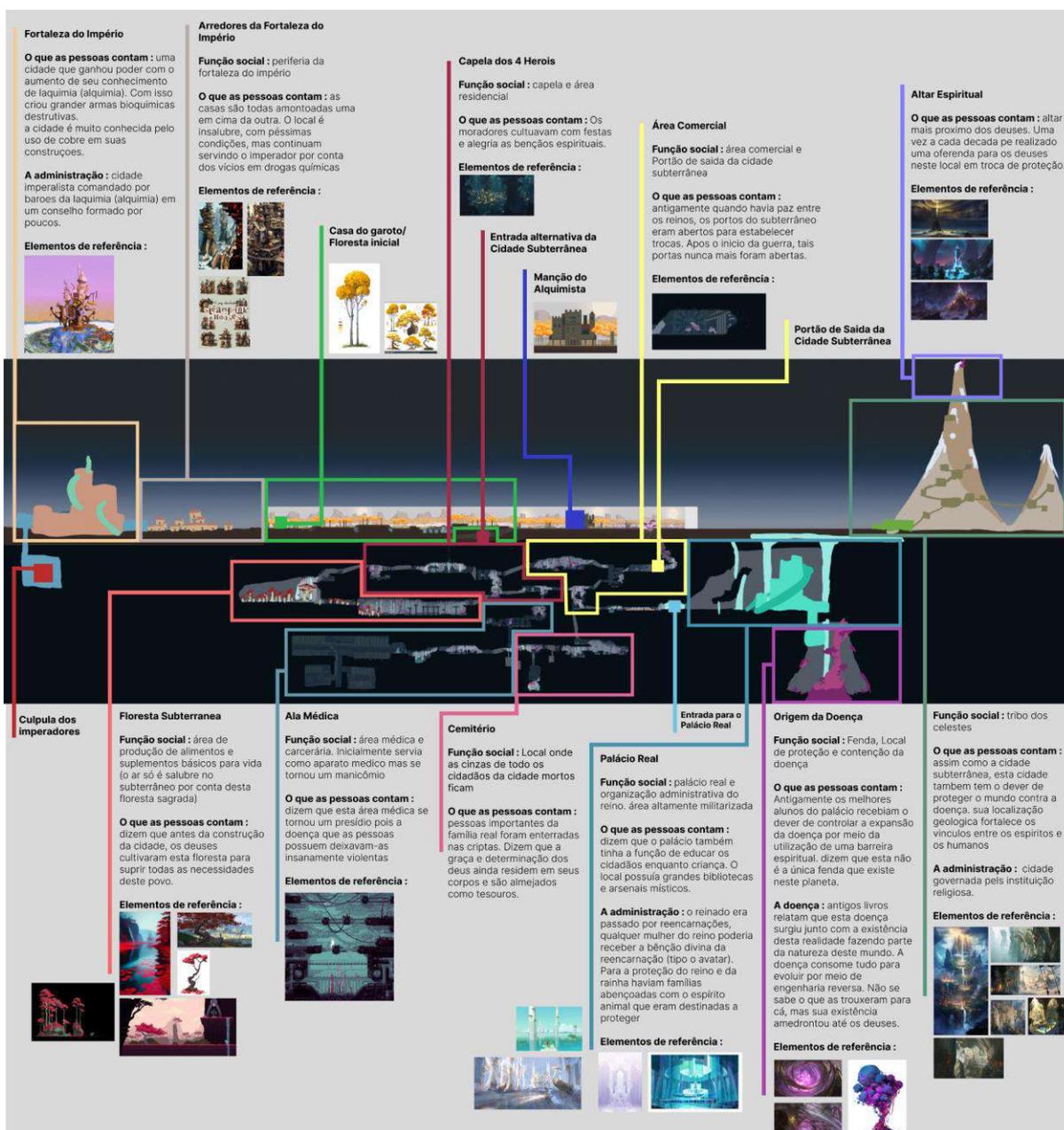


Figura 153. Detalhamento da concepção de novas áreas.

Criamos um nome específico para identificar cada uma das áreas novas e já existentes, escrevemos a função imaginada de cada área no universo do jogo e oferecemos uma breve descrição lúdica, como, por exemplo, a Ala Médica (Figura 154). Esse desenvolvimento serviu tanto para o retrabalho do level design quanto para a criação de novos *assets*.



Figura 154. Concepção Ala Médica.

Também criamos alguns *concepts* pontuais para complementar os anteriores. Por exemplo, com a alteração na mecânica dos *checkpoints*, desenvolvemos várias versões da sala para serem utilizadas conforme a localização do checkpoint no mapa (Figura 155).

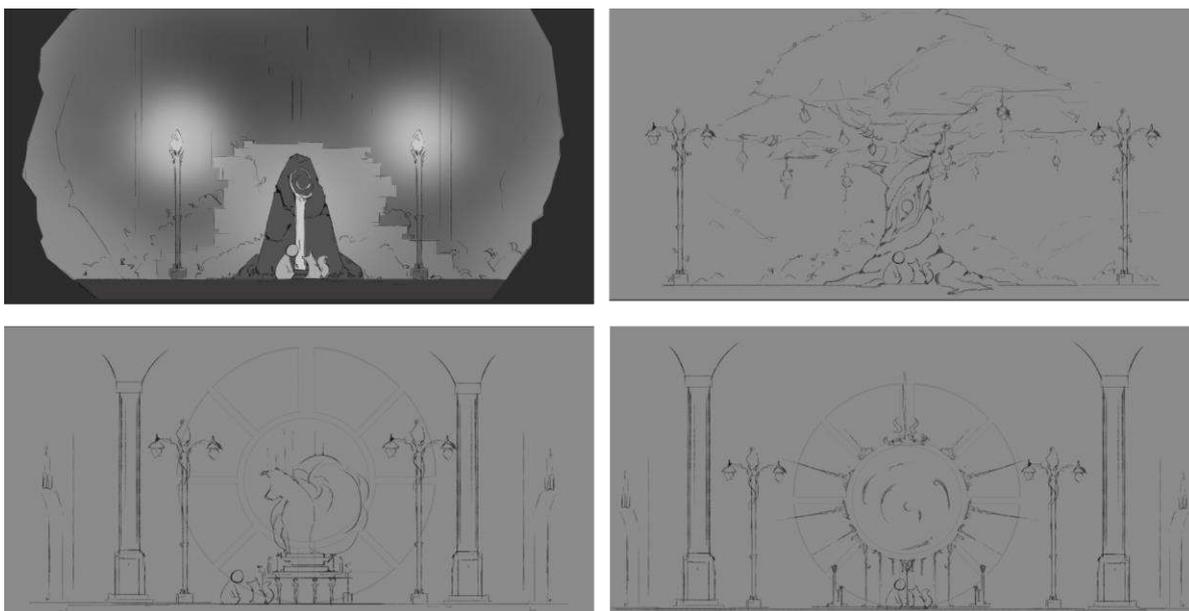


Figura 155. *Concepts* novos checkpoints.

Partindo da esquerda para a direita, a primeira alternativa seria utilizada nas áreas iniciais em volta da Capela dos 4 Heróis, Área Comercial e Ala Médica, a segunda alternativa para ser utilizada na região da Floresta Subterrânea, a terceira nos arredores do Altar Espiritual e por fim a quarta para ser utilizada no Palácio Real.

Trabalhamos também em *concepts* para o castelo real (Figura 156), que é um elemento importante na Zine e também uma localização presente no jogo.



Figura 156. *Concepts* trono real.

8.6.2 Modificações propostas ao *Level Design*

Após a implementação e os testes, identificamos equívocos e falhas presentes em diversos pontos do *level design* implementado na primeira versão do jogo. Por exemplo, não consideramos adequadamente o posicionamento dos inimigos no mapa, o que tornou alguns combates excessivamente punitivos e quase impossíveis de superar sem sofrer dano. Além disso, certas áreas se mostraram muito "apertadas", dificultando a navegação e movimentação do jogador pelo mapa. Dificuldades, essas, que não foram intencionalmente implementadas.

Ao abriremos novamente as discussões sobre o mapa do jogo e ao decidirmos expandi-lo. A equipe de mapas iniciou a criação do *level design* adaptando o ao *layout* desenvolvido e discutido na concepção do mapa (figura 157).

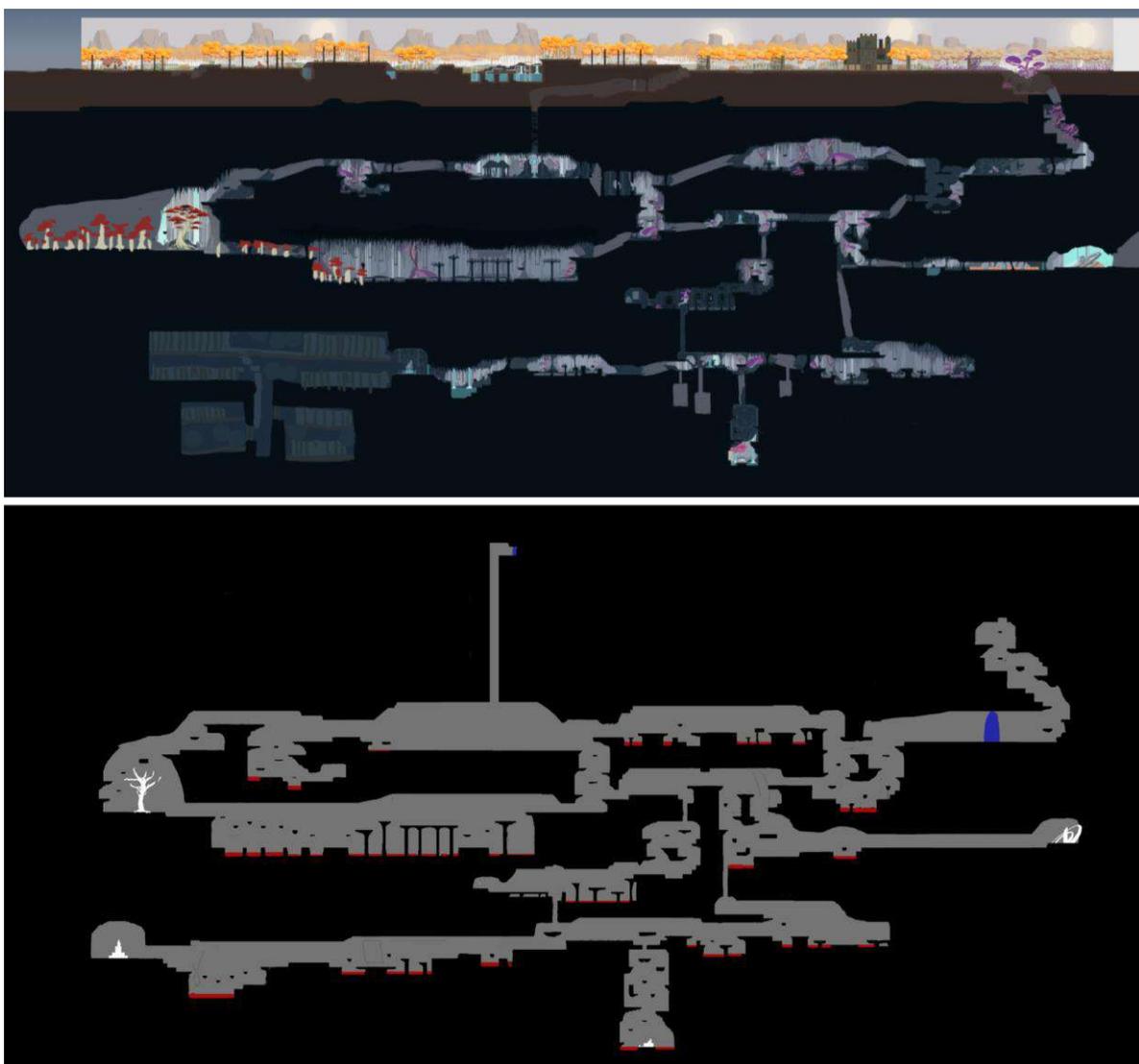


Figura 157. Adaptação do level ao novo *concept*.

Com base na expansão do mapa que realizamos, continuamos o planejamento do novo *level design*, o desenvolvendo, expandindo e adequando-o ao *concept* desenvolvido (Figura 158).



Figura 158. Expansão do novo *level design*.

Expandimos ainda mais a ideia original modificando rotas e caminhos, adicionando novas salas, modificando a posição dos pontos de interesse (marcados em branco), os mecanismos de portões (Figura 159), posicionamos as barreiras que só podem ser abertas com alavancas ou mecanismos (marcados em azul escuro), posicionamos novamente os *checkpoints* de maneira que possam ser separados das outras fases (marcados em ciano) e por fim colocamos também o novo tipo de barreira que só poderia ser liberada com a habilidade especial **ataque carregado** (Figura 160).



Figura 159. Posicionamento dos mecanismos no novo *level design*.



Figura 160. Novas barreiras posicionadas no novo *level design*.

Adicionamos a área da Floresta inicial e finalizamos o *level design* estabelecendo a separação de cada sala e marcando todos os pontos de interesse essenciais. Também mantivemos a utilização do esquema de cores para marcação, o expandindo para fazer novas marcações necessárias (Figura 161).

Nesta versão marcamos a posição de todas as habilidades especiais presentes nestas regiões do mapa, sendo elas Ataque Carregado, Mordida da Raposa, Dash Tático e Defesa Mágica (Figura 162), e os seus locais de interação. Além de fazer o posicionamento de inimigos, dos pontos de elevadores (Figura 163) e posicionar os objetos que o jogador pode empurrar.



Figura 161. Novo *level design* completo.



Figura 162. Posicionamentos das habilidades especiais novo *level design*.



Figura 163. Posicionamentos dos elevadores no novo *level design*.

Com isso chegamos nesta versão que é mais atual do *level design*, com todas as marcações importantes recolocadas em suas respectivas cores como mostra a legenda de cores no canto esquerdo da imagem. Também marcamos todas as áreas da concepção que decidimos incluir no mapa, para servir de referência na criação de *assets* e finalização de cada uma (Figura 164).

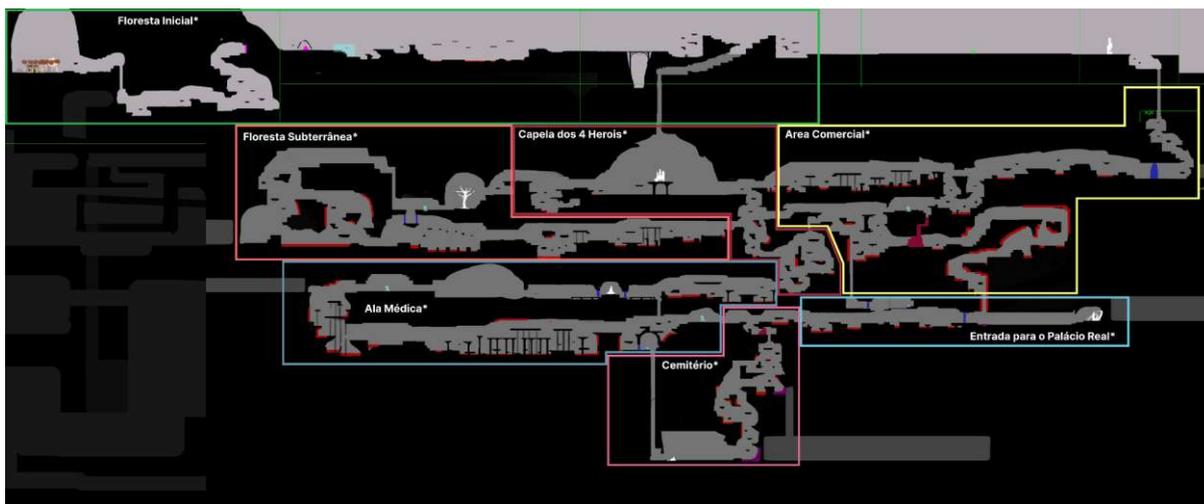


Figura 164. Áreas da concepção presentes no novo *level design*.

Essa versão do *level design* contempla apenas as regiões centrais da Floresta Inicial, Entrada para a Cidade Subterrânea, Capela dos Quatro Heróis, Área Comercial, Floresta Subterrânea, Ala Médica, Cemitério e Entrada para o Palácio Real.

A inclusão de novas áreas seguiu a direção definida para o enredo e a narrativa, expandindo o mapa e estabelecendo a posição espacial de cada grande área, além de elementos visuais e narrativos. Isso foi feito para que ao desenvolver o *level design* fosse levado em consideração o enredo por trás de cada região, e para que um pouco desse enredo seja passado na construção do *level design*.

8.6.3 Mudanças do sistema de *tiles*

Com a mudança de *engine*, a utilização da ferramenta externa TileD para construção do *level design* utilizando *tiles* se tornou desnecessária, pois a nova *engine*, a Unity, permite que designers construam e testem as fases diretamente nela, cortando o tempo e o processo de transferência de uma ferramenta externa para a *engine*.

Aproveitando a mudança e o fato que todas as fases precisam ser refeitas, decidimos por desenvolvemos um novo *tileset*, reduzindo o tamanho dos *tiles* de 28x28 para 14x14 *pixels*. o tamanho anterior, ao longo do desenvolvimento, se mostrou muito grande em alguns momentos para a construção do mapa, e, em alguns aspectos limitador, então optamos por reduzir pela metade. Além disso, a nova *engine* permitiu a inclusão de *tiles* de diferentes tamanhos no

mesmo *tileset*, sendo o tamanho máximo 14x14, tamanhos menores como meio tiles e tiles de canto poderiam ser utilizados sem problemas com o sistema de colisão do jogo (Figura 165).

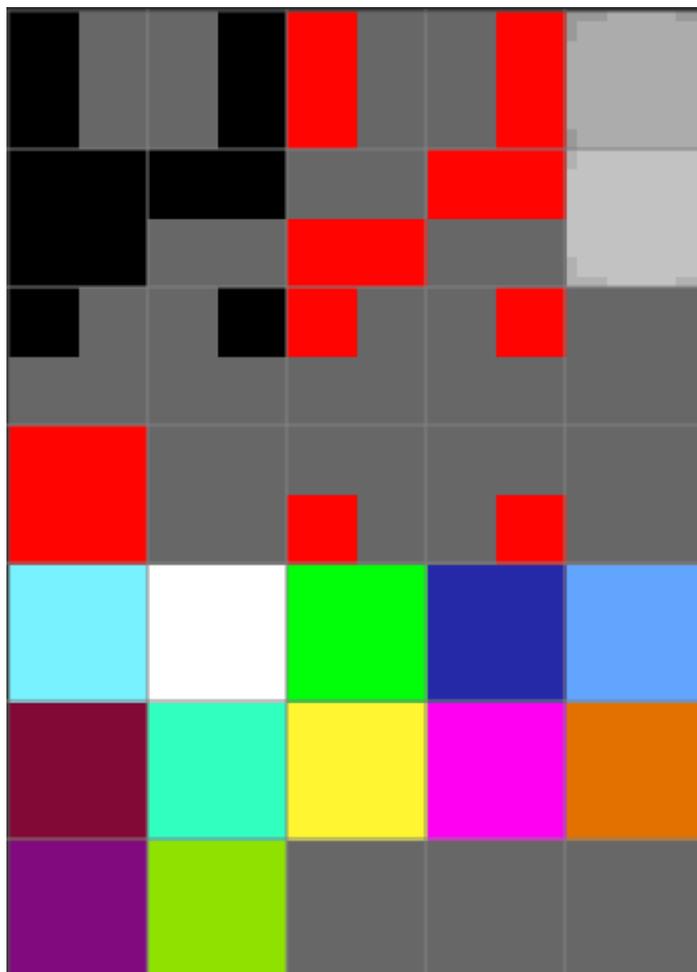


Figura 165. Novo *tileset* de informação.

Também abandonamos por completo o uso *tiles* para construção visual dos cenários e passamos a fazer a utilização deles apenas para criar o *layout* da colisão, posicionar os blocos de dano e informar o posicionamento de itens e locais de interação (Figura 163). A coloração dos *tiles* novamente seguiu o esquema de marcação de cores desenvolvido no novo *level design* visto anteriormente na Figura 166.



Figura 167. Um *level* construído na Unity

Com essas mudanças, conseguimos construir rapidamente o *level design* do jogo dentro da nova *engine* e implementá-lo para testes internos. Isso nos permitiu coletar *feedback* e corrigir diversos elementos antes de avançar para a criação dos *assets* e cenários finais.

8.6.4 Modificações propostas para os Cenários e novos *assets*

Na fase de construção do *level design* na Unity, a equipe decidiu preservar o sistema de padronização de camadas que havia sido adotado na versão anterior do jogo, o que permitiu uma transição direta para a criação de novos *assets* e a finalização dos novos cenários.

No processo de desenvolvimento dos *assets*, utilizamos todos aqueles que já haviam sido previamente criados e demos liberdade aos artistas para que desenhassem novos *assets* para incorporar elementos narrativos e mecânicos adicionais, além de enriquecer os cenários com mais detalhes. Por exemplo, para a Floresta Inicial e as áreas da Floresta Subterrânea, foram criados *assets* que complementam os já existentes, incluindo uma variedade de estilos de árvores (Figura 167), arbustos, flores, uma ponte e outros objetos (Figura 168), que contribuíram para uma ambientação mais rica e detalhada.

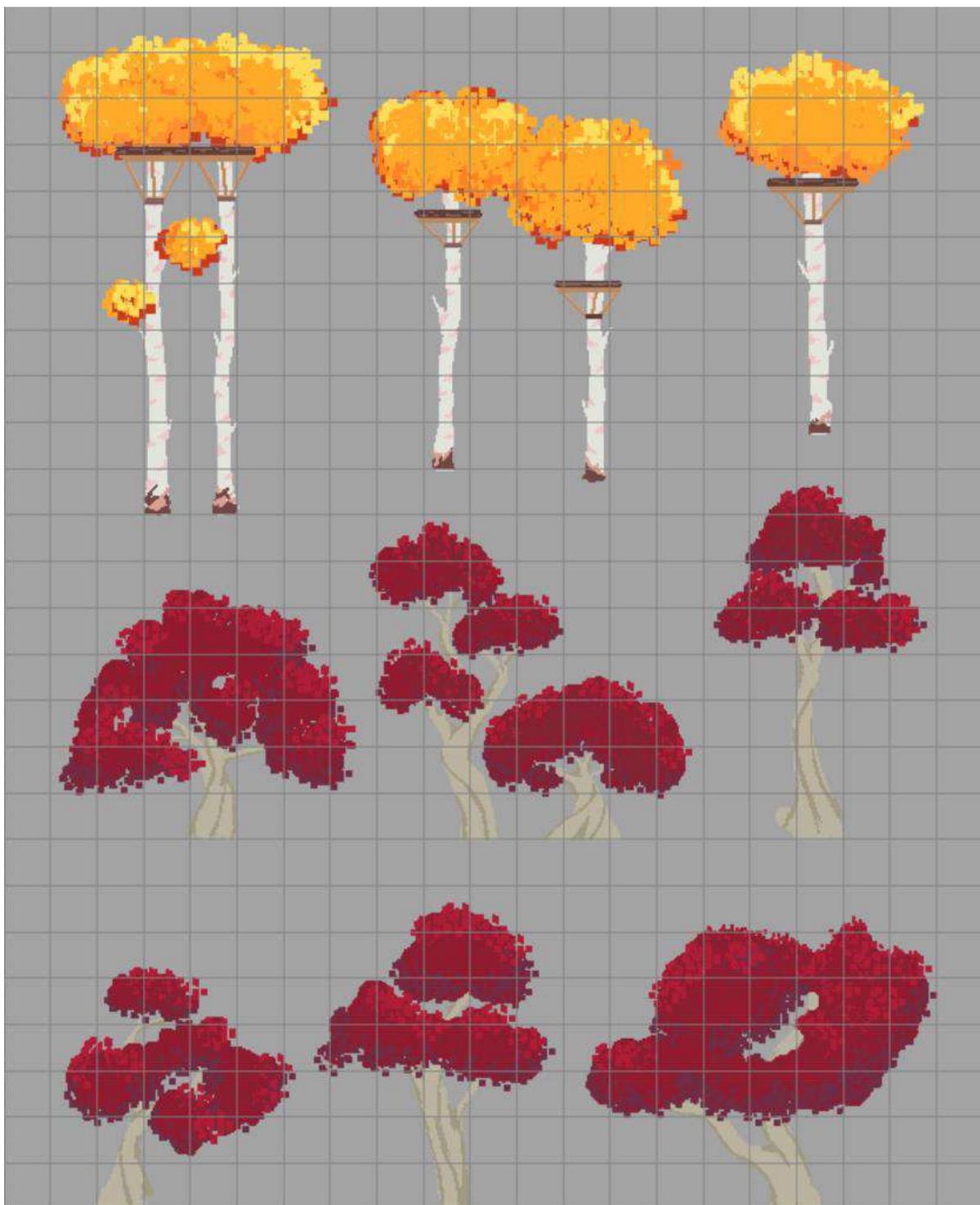


Figura 168. Novos *assets* de árvores.



Figura 169. Novos *assets* arbustos, flores, ponte etc.

Adotamos uma lógica semelhante para a Cidade Subterrânea, onde desenvolvemos novos *assets* para enriquecer os cenários e torná-los visualmente mais interessantes, como pilares, plataformas, correntes (Figura 169), portas, jaulas e prisões (Figura 170). Além disso, foram criados *assets* para atender às novas necessidades de mecânicas e level design, como elevadores (Figura 171), checkpoints (Figura 172), bem como um *asset* exclusivo: a fonte localizada na Capela dos 4 Heróis (Figura 173).

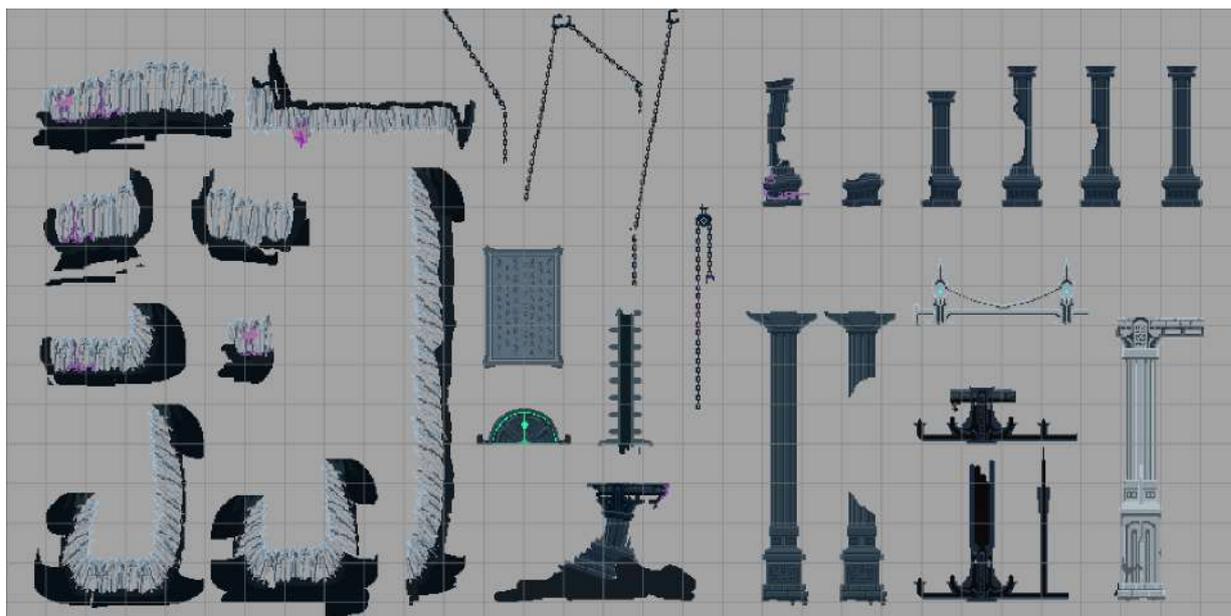


Figura 170. Novos *assets* pilares, plataformas, correntes etc.

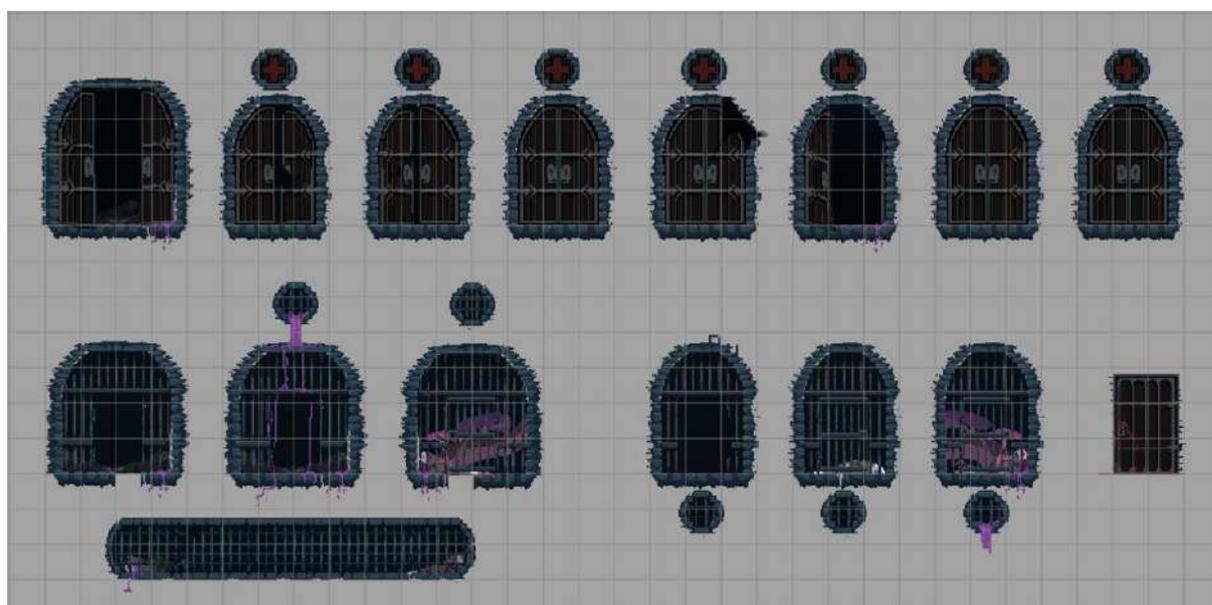


Figura 171. *Assets* portas e jaulas.

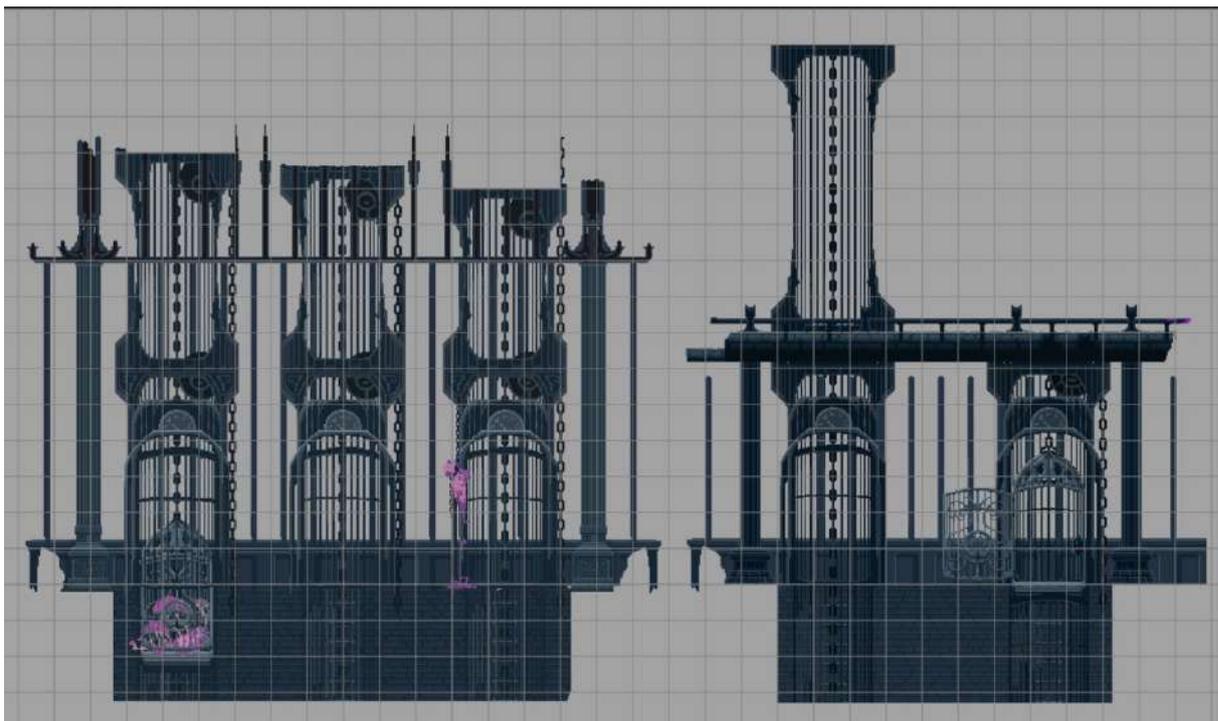


Figura 172. *Assets* elevadores.

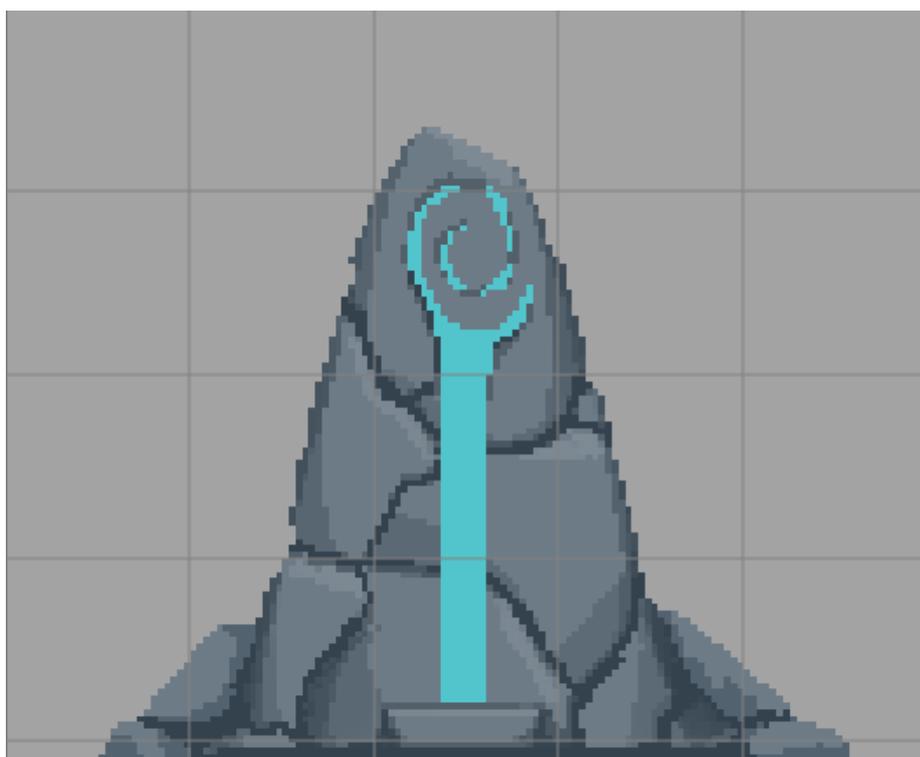


Figura 173. Novo *asset* checkpoint.

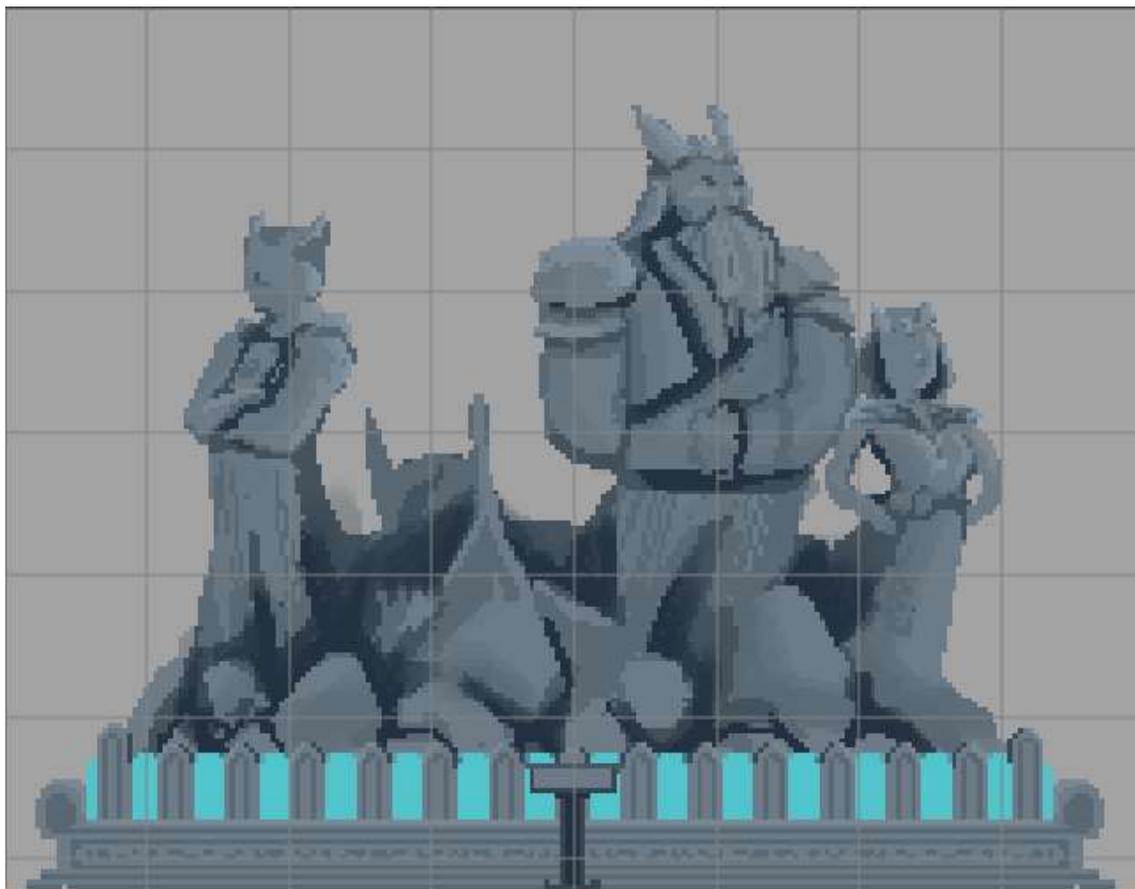


Figura 174. Fonte dos 4 Heróis.

À medida que avançávamos na criação dos novos *assets*, iniciamos a finalização dos cenários. Baseados no layout estabelecido no *level design*, cada cenário foi projetado e detalhado individualmente, garantindo que interagisse visualmente com a área em que estava situado, ao mesmo tempo em que preservava sua identidade única. Como, por exemplo, o início do jogo e onde a Casa do Garoto (Figura 174) está situada, que recebeu um visual exclusivo e distinto da Floresta Inicial, que também passou por ajustes significativos e recebeu novos detalhes (Figura 175).



Figura 175. Novo cenário casa do garoto.



Figura 176. Novos cenários da floresta inicial.

Entrada para a Cidade Subterrânea (Figura 176), foi transformada para refletir as características da cidade subterrânea. A área da Capela dos 4 Heróis (Figura 177), foi finalizada com a adição da fonte e da primeira estação de elevadores, as fases ao redor também foram retrabalhadas.

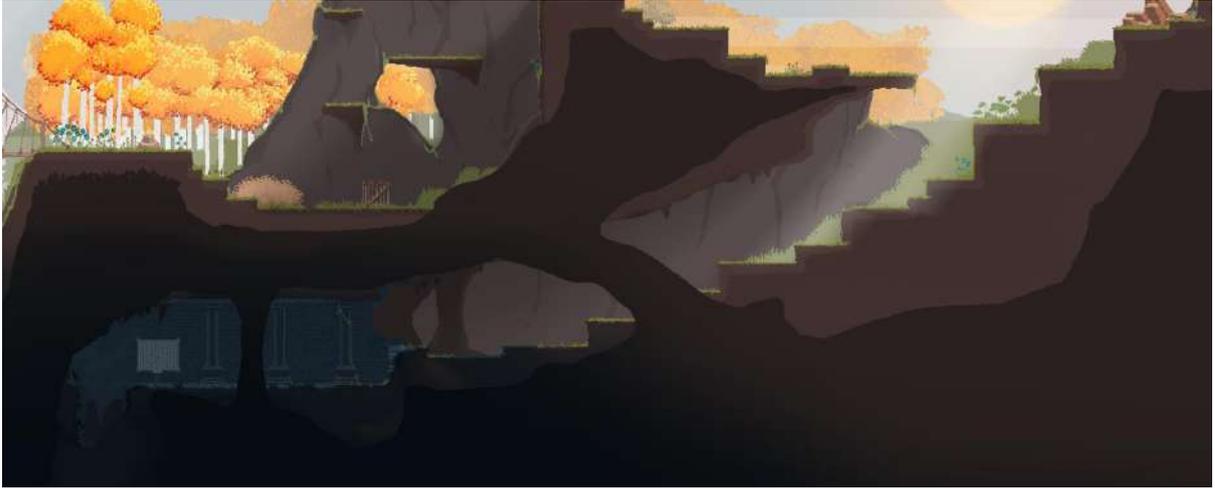


Figura 177. Cenários: entrada da cidade subterrânea.

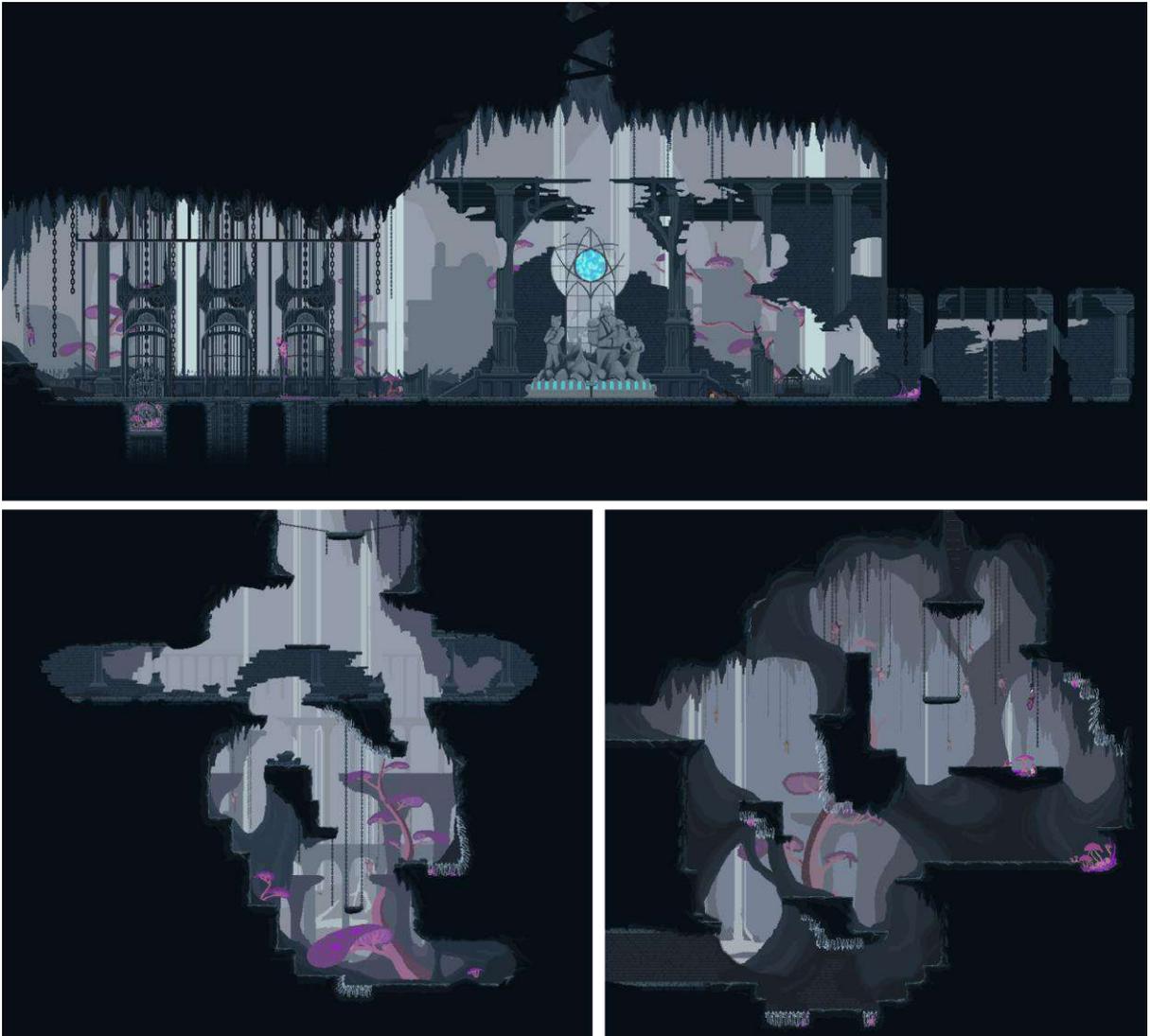


Figura 178. Capela dos 4 Heróis e arredores.

Na Ala Médica, foram adicionados o segundo elevador (Figura 178), às áreas em volta foram modificadas e refeitas para combinar com a região (Figura 179) e uma arena projetada para a primeira batalha do jogador contra um *boss* foi construída (Figura 180). Além do reposicionamento de *assets* e pontos de interesse definidos no *level design*.



Figura 179. Ala médica: Segundo elevador.



Figura 180. Ala médica.



Figura 181. Ala médica: Sala do *boss*.

Além disso, foram desenvolvidas diversas salas de checkpoint espalhadas pela Cidade Subterrânea (Figura 181). Cada sala foi projetada com características ligeiramente distintas para se integrar de maneira mais eficaz à sua localização no mapa, proporcionando uma sensação de coesão e fluidez na experiência do jogador.

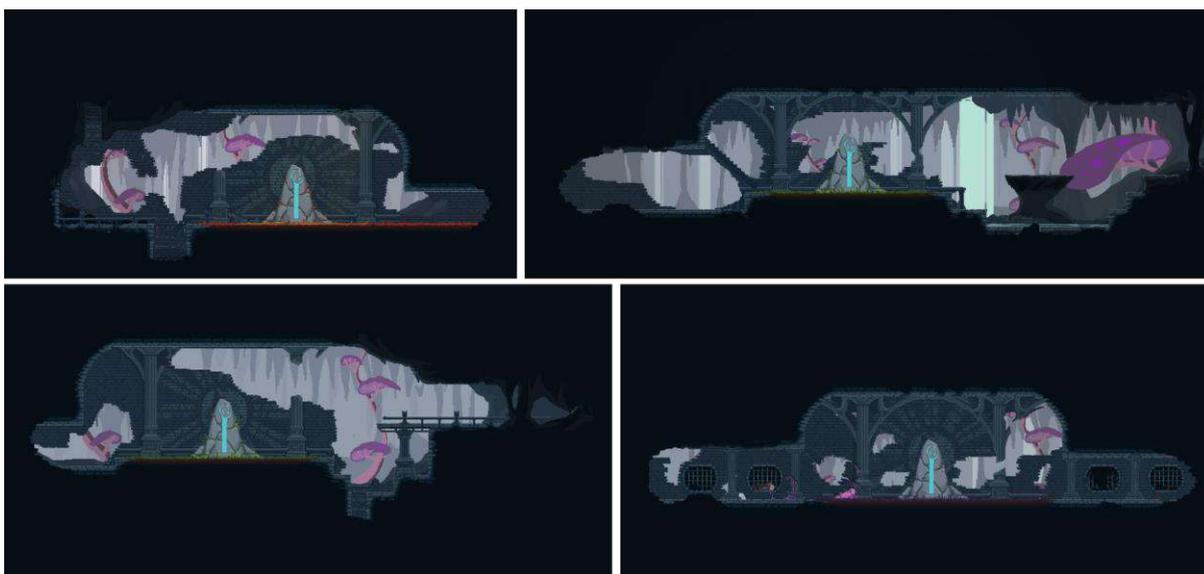


Figura 182. Cenários: Novos *checkpoints*.

E por fim, a Entrada para o Palácio Real foi elaborada (Figura 182). A detalhamos com elementos visuais que antecipam o estilo e a estética futuros do Palácio Real. Esses elementos não apenas servem para criar uma transição visualmente coerente para o palácio, mas também oferecem uma visão antecipada de como a área se desenvolverá.

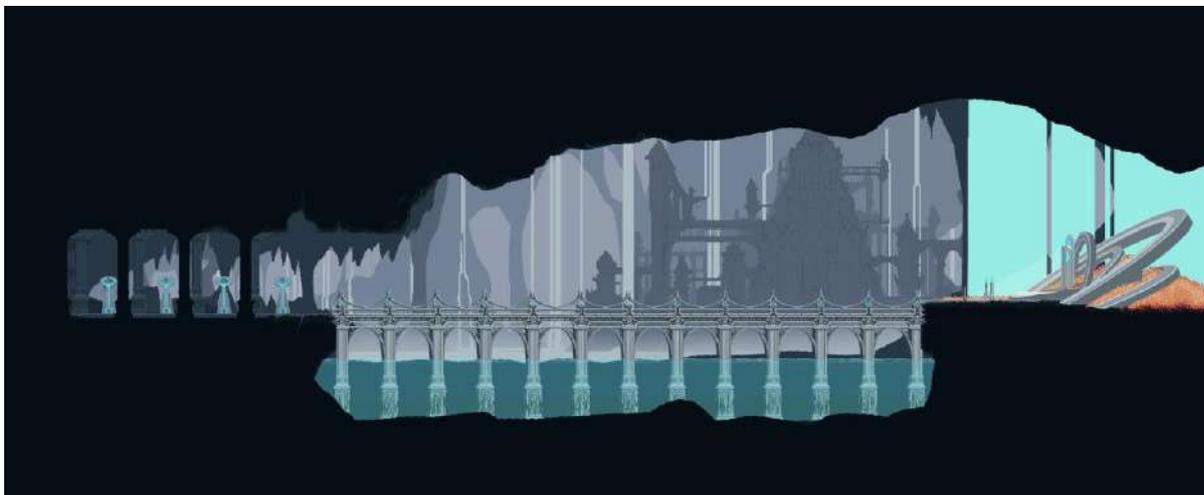


Figura 183. Cenário: Entrada para o Palácio Real.

8.7 Considerações sobre Mapas e Cenários

Os mapas e os cenários foram um dos grandes desafios para o desenvolvimento do nosso jogo, por serem um dos elementos que definem o gênero Metroidvania, por esse motivo decidimos por planejar o *level design* com mais cuidado, posicionando os principais pontos de interesse em um *layout* simplificado, para então lentamente começar a adicionar as salas e definir os até chegar aos detalhes como posicionamento de plataformas e inimigos. Esse processo de partir de uma versão simplificada e ir lentamente adicionando complexidade facilitou muito a criação de um mapa extenso mantendo a coesão e conexão entre as salas.

A criação do cenário em *tiles* começou antes mesmo da finalização do *level design*, onde nós começamos a testar com a ferramenta e criar os primeiros *tilesets*, logo nesse momento percebemos as barreiras que esse método oferece, barreiras essas que foram discutidas nas reuniões com o grupo, que eventualmente nós trouxe a decisão de deixar de usar *tiles* para a criação visual dos cenários, e ao invés disso usar uma imagem inteira para o cenário, que então foi desenvolvida para o sistema de camadas.

Com os mapas sendo implementados logo cedo graças a utilização dos *tiles* apenas para informação, como o posicionamento de plataformas ou ponto de interação, e o esquema de camadas padronizado, a implementação da arte dos cenários se tornou muito mais simplificada, o que deu aos artistas mais tempo para trabalhar individualmente em cada uma das artes e animações para o cenário, refletindo diretamente na qualidade dos cenários.

Assim, a equipe conseguiu entregar um mapa extenso, começando com o foco inicial no desenvolvimento do level design e avançando rapidamente para a criação do *level* em *tiles* para ser implementado as colisões e outras mecânicas do mapa. Enquanto os artistas trabalharam de maneira mais flexível, em paralelo com a equipe de programação, sem com que uma precisasse parar esperando entregas da outra parte.

9. Telas e Heads-Up Display

Neste capítulo, discutiremos o processo de desenvolvimento do Heads-Up Display (HUD) e das telas do jogo, incluindo o menu inicial, os créditos e outros elementos gráficos como botões e logotipos. A criação dessas telas é relevante, pois são os primeiros elementos visuais que o jogador tem contato ao abrir o jogo. Elas oferecem uma primeira impressão do que o jogador pode esperar da experiência de jogo, além de fornecerem informações e acesso a uma série de configurações importantes. O HUD, por sua vez, é essencial para a experiência do jogador durante o *gameplay*. Trata-se de um conjunto de elementos gráficos projetados na tela, que são responsáveis por comunicar informações do estado de jogo de maneira rápida e em tempo real, como, por exemplo, a quantidade de vida do jogador ou a disponibilidade de um determinado ataque no momento.

A criação das telas começou no Figma. Utilizando elementos de outros jogos, como o cenário de Kingdom (2015) e os *sprites* dos protagonistas de Momodora e Blasphemous, montamos um protótipo inicial de uma tela de *gameplay* (Figura 183). Nosso objetivo era ter uma noção inicial do tamanho real da tela e dos personagens inseridos nela, o que nos ajudou a definir o tamanho do cenário, do protagonista, de possíveis inimigos etc. Com isso, a equipe pôde determinar um tamanho de tela em *pixels* adequado para começar os trabalhos.



Figura 184. Teste inicial da tela de *gameplay*.

Essa fase foi conduzida no início do projeto. Além de ser essencial para o desenvolvimento do HUD e das telas pela equipe de Design, consistiu também em um importante registro para a comunicação com a equipe de programação, que precisava criar e configurar a câmera. Os *designers* precisavam ter uma noção clara do que seria exibido na tela para desenvolver personagens e cenários que se ajustassem ao dimensionamento da tela proposto.

9.1 Elementos do HUD

Com a definição do tamanho da tela de *gameplay*, ainda utilizando os *sprites* dos personagens, mas sem o cenário do Kingdom (2015), criamos uma segunda versão preliminar da tela de *gameplay*, desta vez focando mais no HUD (Figura 184). Nesse estágio, planejamos a disposição dos elementos essenciais que o jogador precisa acessar durante o jogo. Posicionamos a barra de vida e de pontos de magia do protagonista no canto superior esquerdo, a barra de vida de um *boss* no canto superior direito, e a indicação de itens coletados abaixo da vida do protagonista. À medida que os *sprites* dos personagens foram sendo finalizados, os incorporamos a essa versão para visualizar melhor o arranjo dos elementos.



Figura 185. Teste do HUD.

Continuamos testando e refinando o tamanho e o posicionamento dos elementos do HUD no Figma, criando uma versão que agora incluía apenas personagens desenvolvidos por nós (Figura 185). Adicionamos blocos com tamanho aproximado de 28×28 pixels, correspondente ao tamanho dos *tiles* do nosso jogo, para testar visualmente a relação de proporção e tamanho entre os elementos do jogo e do HUD.



Figura 186. Teste do HUD contendo nossos personagens.

Nosso objetivo era encontrar um equilíbrio, garantindo que os elementos do HUD fossem visíveis sem ocupar espaço excessivo na tela ou bloquear a visão do jogador. Decidimos manter um HUD simples, com as informações básicas da primeira versão, como a vida e magia do protagonista, além da vida do alquimista *boss* durante a batalha final. Também consideramos deixar os espaços de itens, que indicam o progresso na coleta da poção, sempre visíveis no HUD abaixo da barra de vida do protagonista.

Com os elementos do HUD decididos, avançamos para o desenvolvimento dos *assets* no Aseprite. Para trazer um diferencial ao nosso jogo, realizamos vários testes incorporando elementos dos personagens nas barras de vida e magia. Inicialmente, usamos a espada do personagem e para representar sua vida, mudamos a representação visual de barra para pontos, ilustrados por uma série de círculos em cima da espada, representando os pontos de vida (Figura 186). Esta mudança se deu por uma decisão na mecânica do jogo, em que a vida do personagem jogável seria medida por pontos. Posteriormente, adicionamos a cauda da raposa enrolada à espada, destacando o vínculo entre ela e o protagonista (Figura 187).



Figura 187. Primeiro *concept* da barra de vida do protagonista.



Figura 188. Testes de HUD para barra de vida, de magia e itens do protagonista.

Desenvolvemos várias alternativas, testando diferentes formatos e colorações para a cauda e para os pontos de vida (Figura 188). Nesse estágio, decidimos remover os ícones dos ingredientes coletados do HUD, assim como a barra azul que representava os pontos de magia até então. Agora, os pontos de magia do protagonista seriam simbolizados pela cauda da raposa, cujo comprimento variaria de acordo com a quantidade de magia disponível do protagonista. Assim, a cauda aumentaria de tamanho conforme o jogador ganhasse pontos de magia e diminuiria quando ele a utilizasse para algum poder. A cauda também passaria a surgir de um pequeno cristal azul presente na espada do protagonista. Os pontos de vida foram alterados para vermelho, para proporcionar mais contraste e, assim, ficarem mais evidentes para o jogador. Também criamos uma versão aumentada dessas barras, considerando a possibilidade de o jogador adquirir mais vida e magia no futuro.

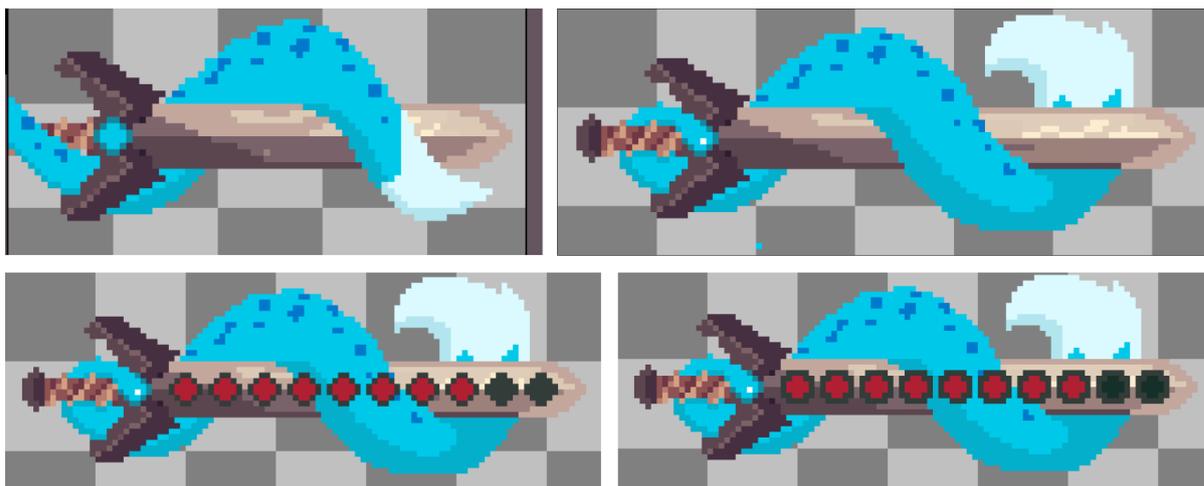


Figura 189. Testes de HUD para barra de vida e magia do protagonista.

Por fim, decidimos criar duas versões do HUD: uma com a cauda da raposa e outra sem. Inicialmente, o jogador veria apenas a espada com os pontos de vida (Figura 189). No entanto, após encontrar a raposa, a cauda dela emergiria do cristal azul da espada, representando a magia do personagem (Figura 190). Isso destacaria a chegada da raposa e tornaria mais clara a conexão entre a raposa e as novas habilidades mágicas adquiridas pelo jogador.



Figura 190. Versão final do HUD para somente a barra de vida do protagonista.

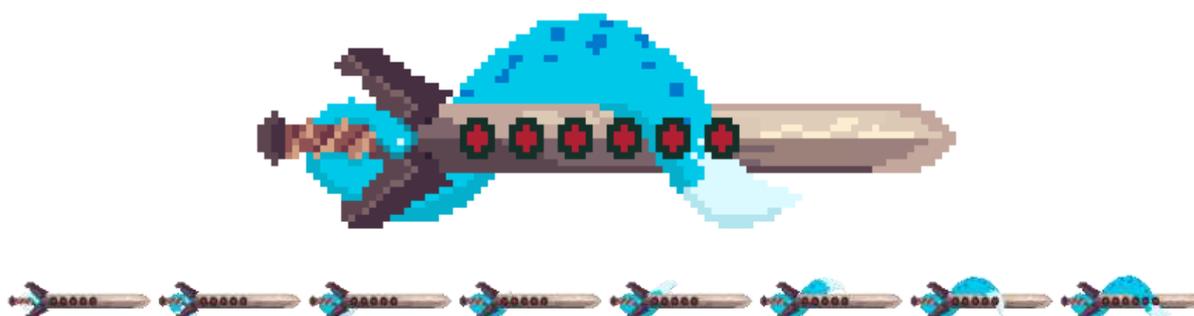


Figura 191. Versão final do HUD para a barra de vida e de magia do protagonista.

Para a vida do alquimista *boss*, adotamos um método semelhante ao da barra de vida do protagonista, mas de forma mais simplificada. Concebemos uma barra temática inspirada em um frasco de laboratório, em que o líquido de cor vinho recuará conforme o *boss* sofresse danos. A barra de vida do protagonista seria posicionada no canto superior esquerdo da tela, enquanto a do *boss* ficaria no lado direito (Figura 191).



Figura 192. HUD finalizado no jogo.

9.2 Telas e Menus

Para elaborar as telas, iniciamos com o planejamento do fluxo de navegação (Figura 192), determinando quais telas seriam necessárias e como cada uma estaria conectada às outras. A partir desse ponto, identificamos as telas essenciais para o jogo e aquelas que teriam uma prioridade menor, sendo desenvolvidas apenas se houvesse tempo disponível. Entre as telas principais escolhidas estão o menu inicial, os créditos, as opções e a tela de *gameplay*, incluindo o HUD.

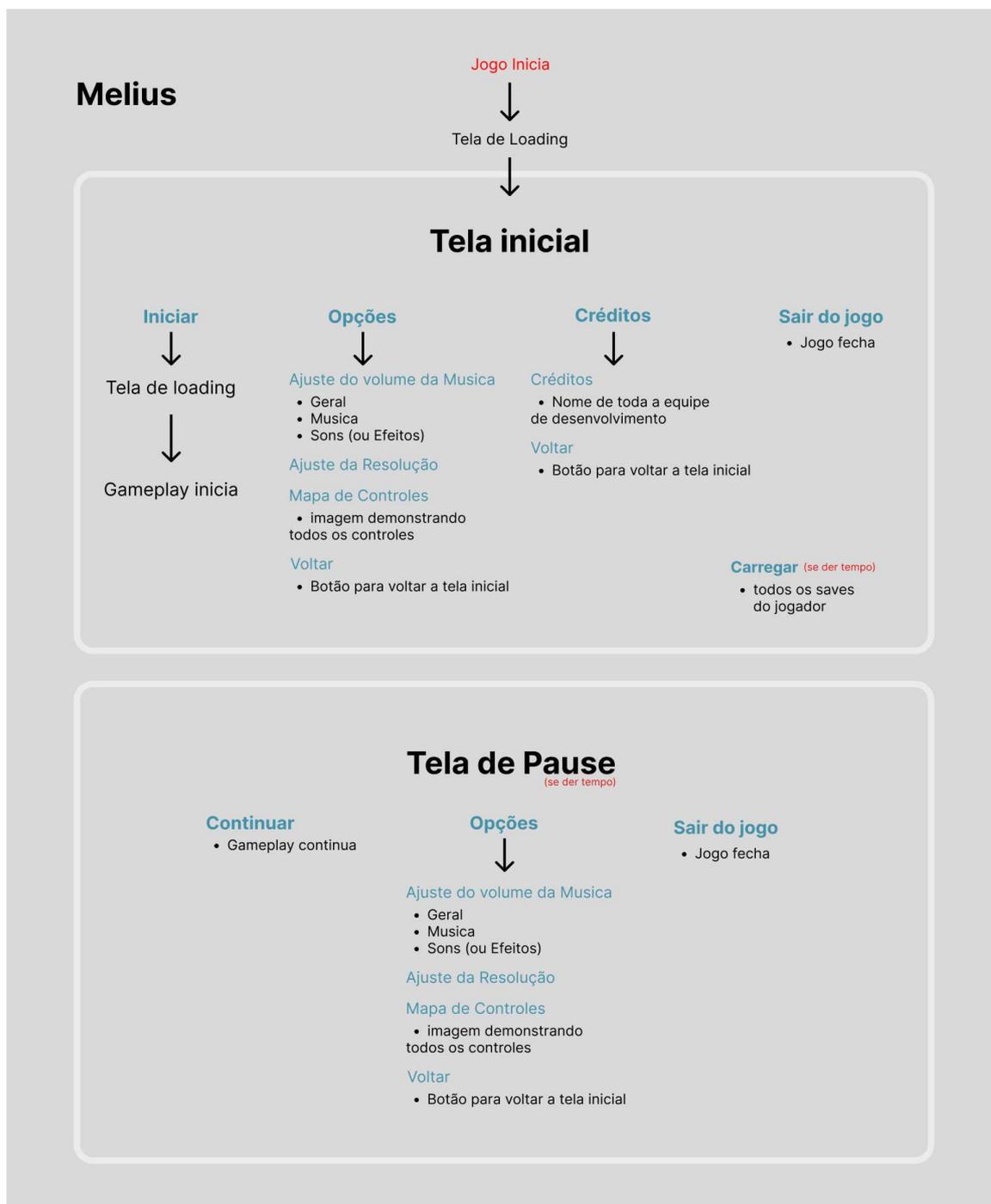


Figura 193. Fluxo de telas.

Com as telas definidas, fizemos uma série de protótipos no Figma de como elas poderiam ser, pensando, principalmente, na proporção, tamanho e posicionamento de elementos como botões, logo etc. (Figura 193).

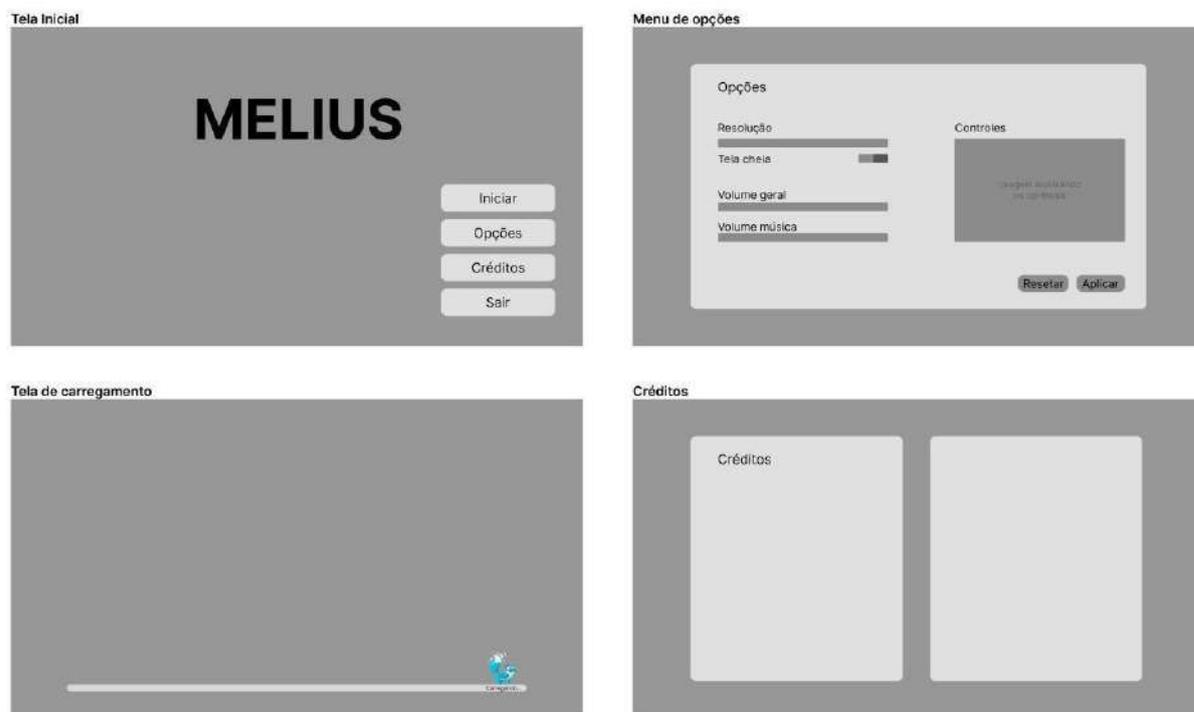


Figura 194. Protótipo inicial das telas.

Dessa forma, embarcamos na criação dos elementos visuais das telas, iniciando com a procura por uma tipografia em *pixel art* que harmonizasse com o estilo do jogo e que seria empregada nos menus, botões, e caixas de texto, entre outros. Após uma série de experimentações com várias tipografias distintas (Figura 194), a equipe optou pela Jupiter Crash BRK (Figura 195) para as caixas de texto e menus. Esta fonte “pixelada” mantém proporções mais verticais e mostrou-se bastante legível.

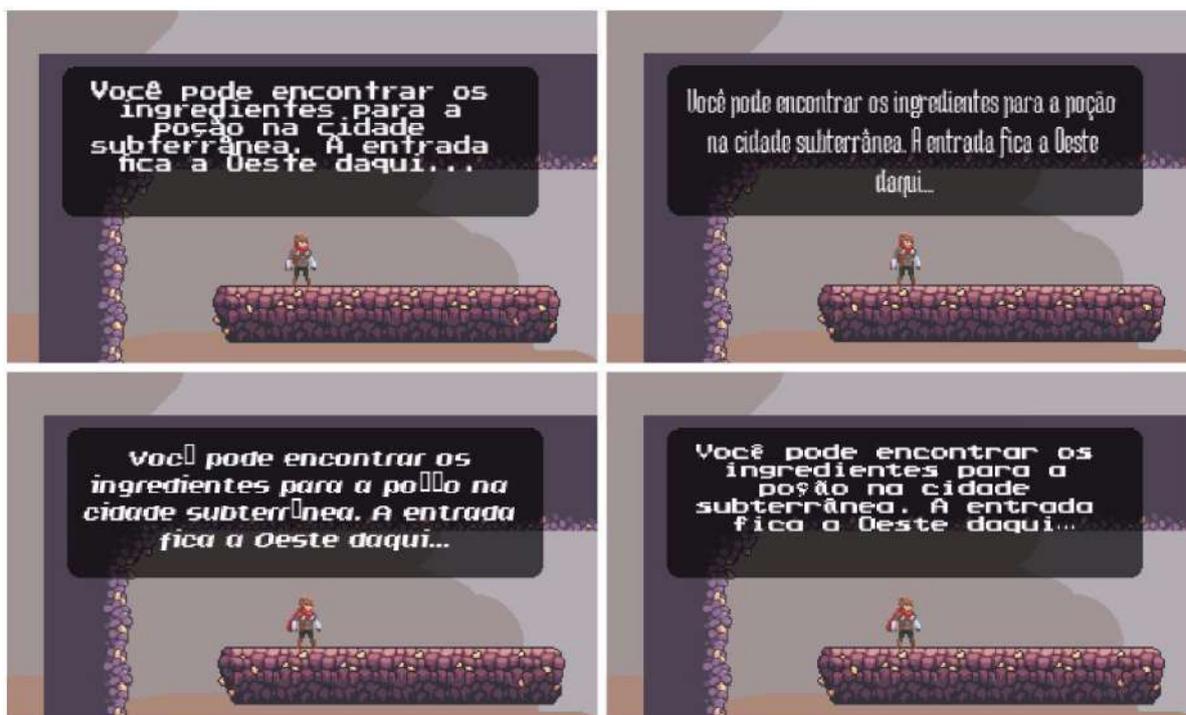


Figura 195. Testes de tipografia.

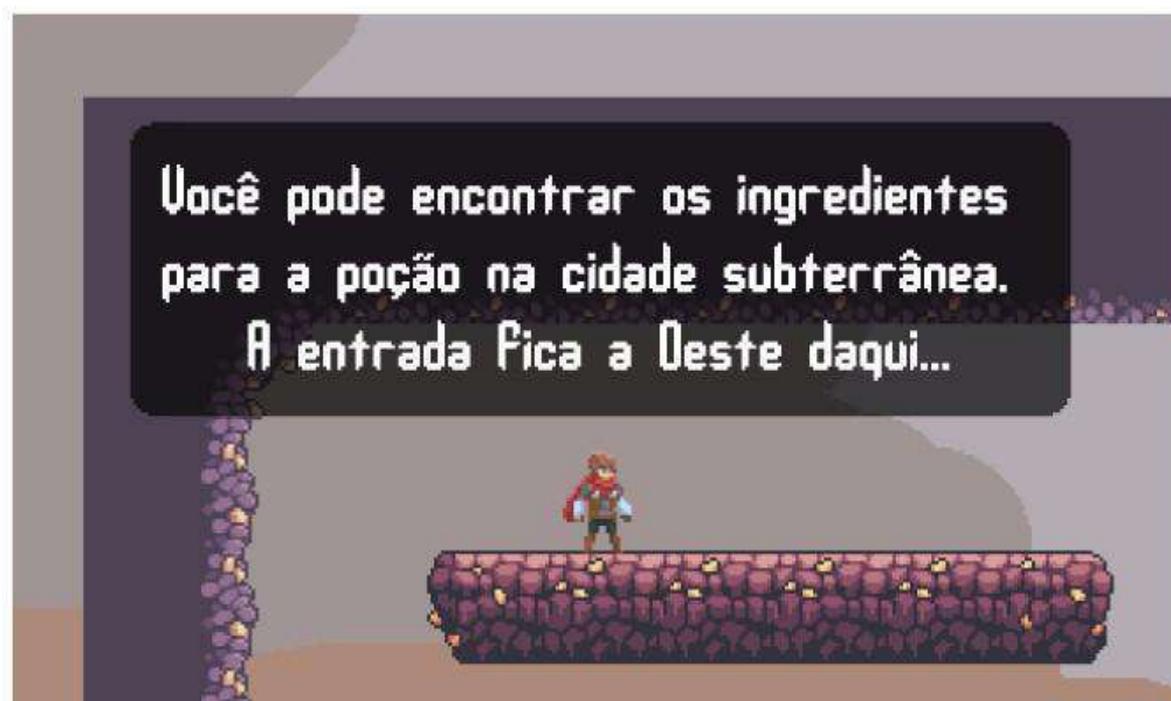


Figura 196. Tipografia escolhida.

Com a tipografia definida, demos início ao processo criativo com a elaboração de caixas de texto para os diálogos, as quais serviriam como base para a construção dos menus.

Inspiramo-nos no jogo Owlboy, desenvolvido pela D-Pad Studio (2016), que apresenta caixas de texto posicionadas tanto acima quanto abaixo dos personagens, além de um rabicho, uma indicação comum em balões de fala de quadrinhos para identificar o falante (Figura 196). Iniciamos o processo com a criação de um protótipo simples no Figma (Figura 197), no qual testamos principalmente as proporções da caixa de texto em relação aos outros elementos da tela.

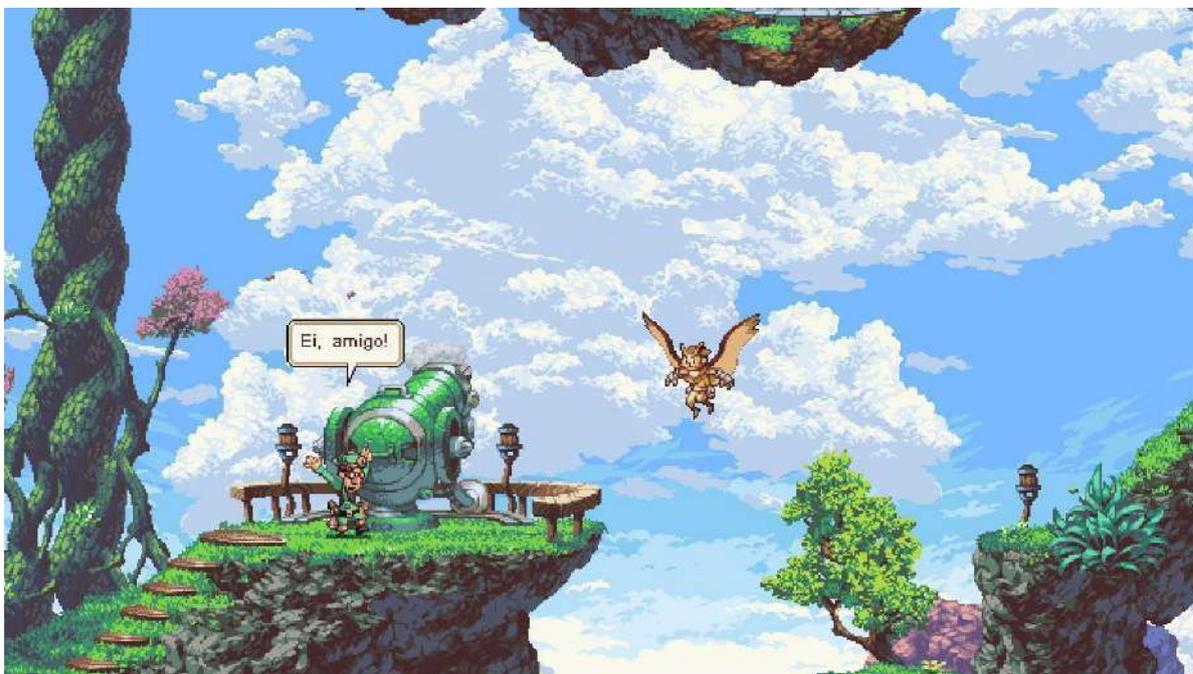


Figura 197. Exemplo de caixa de texto do Owlboy (2016).

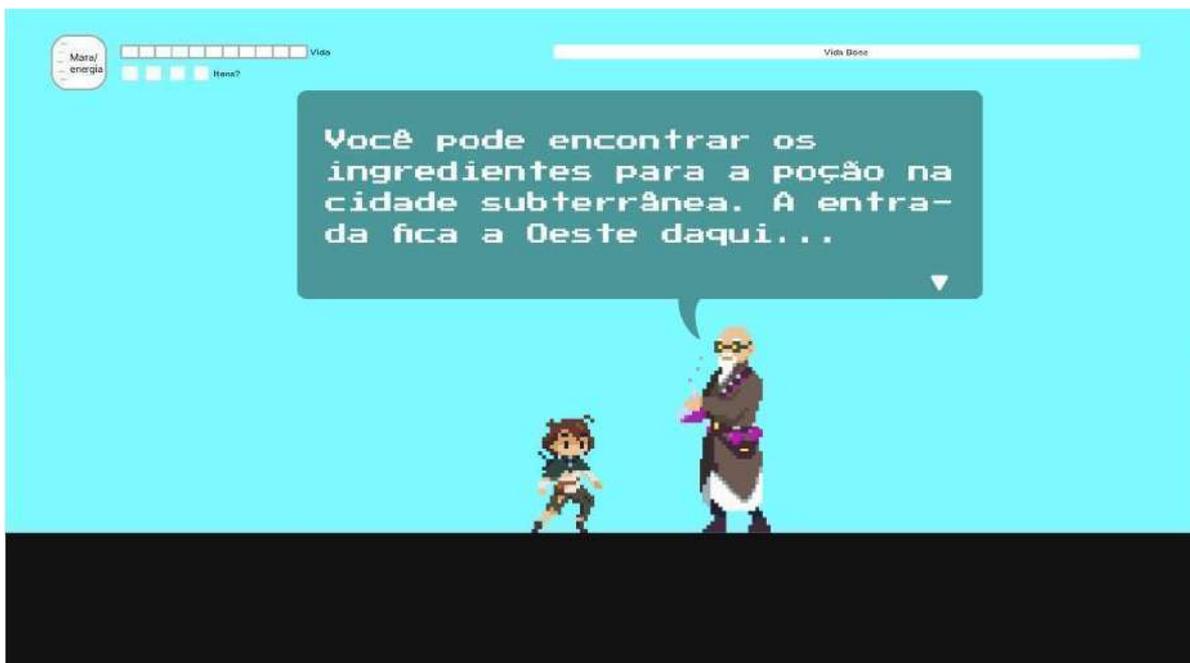


Figura 198. Teste caixa de diálogo.

Em seguida, passamos para o Aseprite, onde criamos um desenho simples de moldura, usando poucas cores, principalmente tons de castanho (Figura 198). Durante o jogo, o texto seria sobreposto a essa caixa, exibindo os diálogos dos personagens. Devido a limitações no momento, não foi viável implementar o rabicho. Portanto, no jogo, tivemos que incluir o nome do personagem que está falando junto ao texto.



Figura 199. Caixa de diálogo implementada no jogo.

Adiante, dedicamo-nos à criação da logo/título do jogo. A tipografia da logo foi concebida por meio da estilização das formas das letras em *pixel art*, utilizando fontes de outros jogos como referência. Na tentativa de transmitir o estilo do jogo para o nome, exploramos diversos elementos do jogo, como árvores, cogumelos e detalhes da caverna. Contudo, a sobrecarga de informações resultou em uma logo poluída, levando-nos a reduzir o número de elementos. Como resultado, removemos a maioria dos elementos e nos concentramos em destacar apenas os detalhes do castelo do alquimista e dos cogumelos da cidade subterrânea (Figura 199).



Figura 200. Logo do Melius.

Em seguida, nos concentramos na criação dos menus e botões. Utilizando a caixa de texto como referência, optamos por um desenho simples, com cores semelhantes e a tipografia escolhida. Os menus seguiram uma estrutura de moldura semelhante à das caixas de texto, porém em proporções maiores. Com a logo e outros elementos finalizados, passamos para a criação das telas de menu, créditos e opções.

O menu inicial (Figura 200) foi desenvolvido com base no *moodboard* elaborado na fase de pesquisa, sendo influenciado por jogos como *Hollow Knight* e *Ghost Song*, que apresentam menus com todos os elementos centralizados, a logo no topo e os botões abaixo. Para o fundo, utilizamos uma imagem do cenário da fase 01-01, o bosque onde a casa do protagonista está localizada, reduzindo sua proporção em 4 *pixels* para 1 e ajustando detalhes como o posicionamento do sol no canto superior direito da tela, ao lado da logo, em uma posição diferente da original da fase.



Figura 201. Menu inicial.

Na tela de créditos (Figura 202), criamos um *layout* simples, contendo a logo da UnB, os nomes e funções dos membros do grupo, e um botão para retornar ao menu inicial. Para a tela de opções (Figura 203), selecionamos configurações comuns de outros jogos, como o volume da música e a resolução do jogo, e também desenvolvemos uma tela simples, incluindo barras e botões para ajustar essas configurações. No entanto, essa tela não foi implementada no protótipo criado na disciplina de DJ.

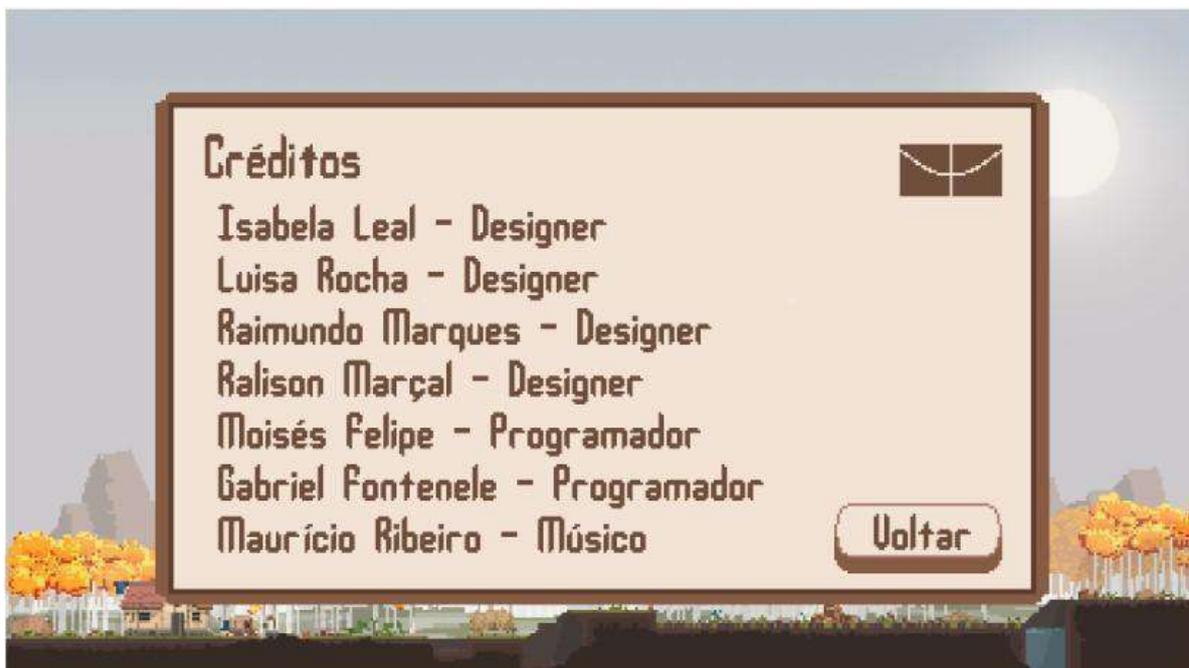


Figura 202. Tela de créditos.

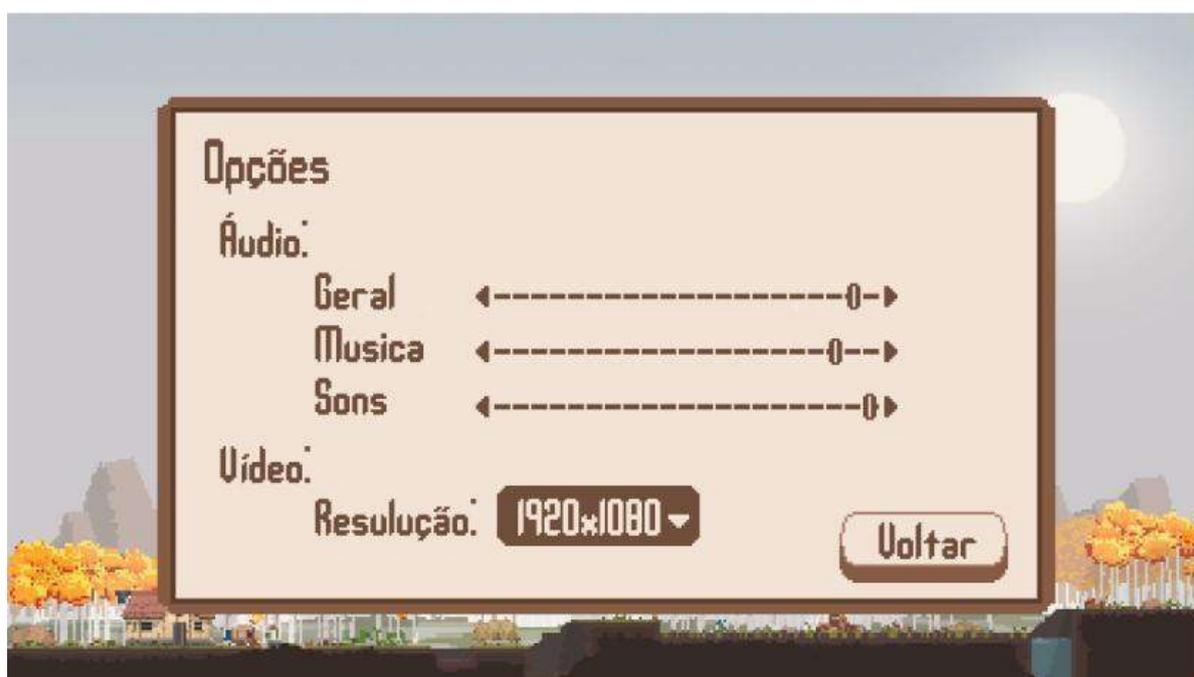


Figura 203. Tela de opções.

9.3 Considerações sobre o HUD e as Telas

Dessa forma, conseguimos criar todas as telas que fornecem as informações e configurações essenciais para o jogador, estabelecendo uma identidade visual que se estende

por todas as telas iniciais e elementos durante o jogo, como os balões de texto. Além disso, desenvolvemos um HUD que não apenas transmite informações importantes para o jogador durante o jogo, mas também possui uma conexão visual com a narrativa do jogo. O HUD, ainda, desempenha um papel fundamental durante o *gameplay*, transmitindo para o jogador uma série de metadados importantes que afetam o personagem jogável, mas que normalmente não são perceptíveis para o jogador. Sendo assim, ele funciona como uma camada de sentidos que auxilia o jogador a experienciar o mundo do jogo.

10. O Jogo Melius

Neste capítulo, apresentaremos as diferentes versões e protótipos interativos do Melius, descrevendo o processo de revisão e as modificações feitas nas mecânicas e animações com base em testes e *feedbacks* obtidos. Estes testes foram realizados de maneira exploratória a partir de observações globais participativas e entrevistas não estruturadas com os jogadores em situação de jogo. As informações foram coletadas livremente enquanto *insights*³⁸ de projeto, não ocorrendo uma sistematização do desempenho dos jogadores ou mesmo dos resultados qualitativos de suas percepções. Assim, a intenção foi coletar problemas mais básicos do jogo, de modo que as etapas mais detalhadas sobre o *playtest*³⁹ são previstas para versões posteriores e refinadas do jogo.

Ao término da disciplina de Design de Jogos, completamos a primeira versão jogável do protótipo de Melius (Figura 203) (Figura 204) (Figura 205). Esta fase permitiu uma análise de aspectos como jogabilidade, funcionamento das mecânicas e navegação pelo mapa, resultando na identificação de erros e inconsistências na versão inicial.

³⁸ *Insight* é uma palavra de origem inglesa que descreve o ato de ser ter uma compreensão súbita sobre algo, de forma a gerar novas perspectivas, ideias e soluções.

³⁹ *Playtest* refere-se ao processo de se testar um jogo em desenvolvimento para que se identifiquem erros de funcionamento, problemas de desempenho e avaliar a jogabilidade e experiência de *gameplay*.



Figura 204. Primeira versão: *gameplay* na superfície.

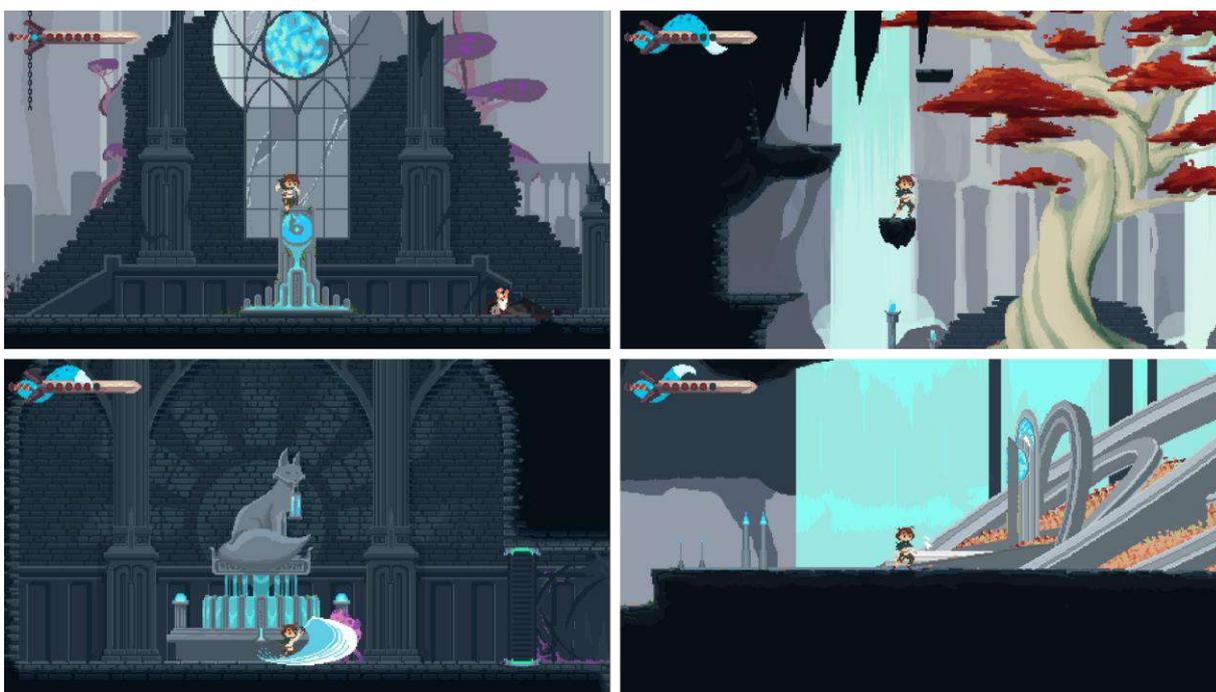


Figura 205. Primeira versão: *gameplay* no subterrâneo.



Figura 206. Primeira versão: *gameplay* na superfície infectada e QR code para vídeo.

Após a conclusão da disciplina, tivemos a oportunidade de apresentar o Melius na Campus Party (CPBSB5). Durante o evento, o jogo foi testado por vários participantes, incluindo jogadores e desenvolvedores. O *feedback* obtido revelou dificuldades na orientação pelo mapa e na identificação dos objetivos a serem cumpridos para avançar. Além disso, o personagem demonstrou rigidez e lentidão durante os ataques, e o *boss* alquimista foi considerado excessivamente difícil devido às suas animações de ataque rápidas. Também foram identificados problemas no posicionamento dos inimigos e a ausência da raposa, a personagem central. O posicionamento inadequado das plataformas também dificultou a navegação.

Essas observações levaram ao início do primeiro ciclo de revisão do jogo, com a transição para a *engine* Unity. Esta mudança foi motivada pelas funcionalidades da Unity, que prometiam facilitar o trabalho dos programadores. Durante esse processo, foram feitas alterações nas mecânicas e animações, com foco na inclusão da raposa.

Embora esta segunda versão do jogo ainda não esteja completa e não contenha todos os elementos discutidos anteriormente, como as habilidades especiais, NPCs, algumas áreas novas do mapa e elementos do HUD, já é possível avaliar algumas melhorias no *gameplay* (Figura 206). O ataque do jogador foi ajustado para ser mais rápido e a inclusão da raposa introduziu uma nova dinâmica, permitindo que o jogador explore o jogo acompanhado.

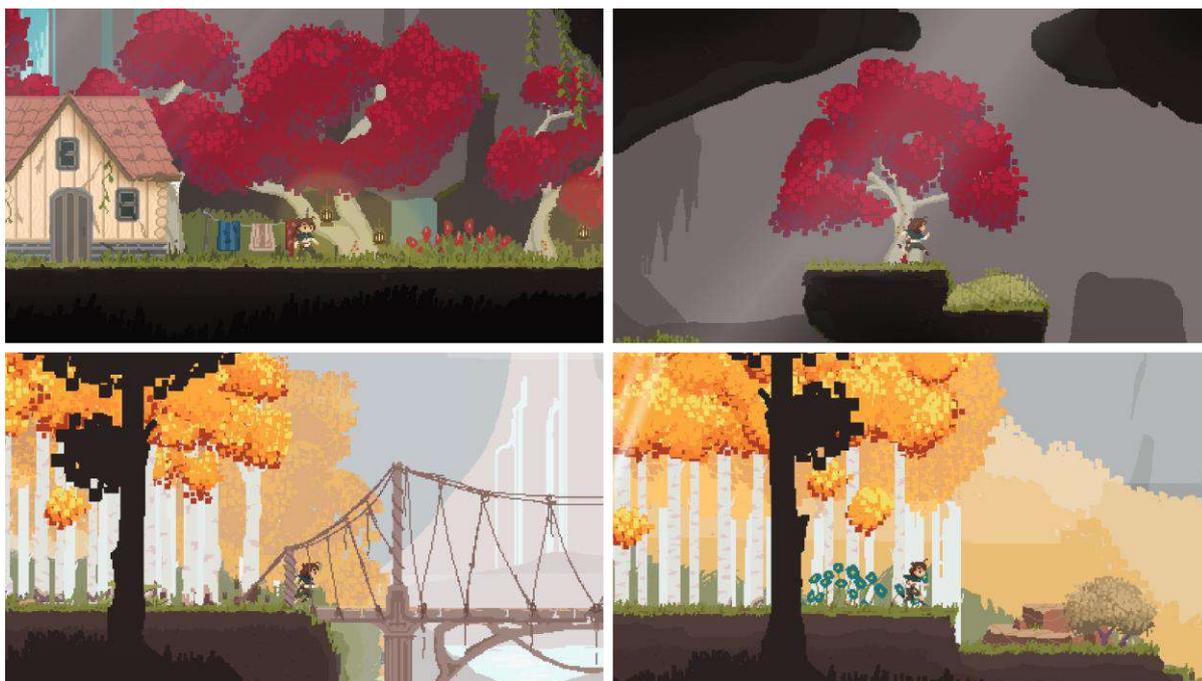


Figura 207.Segunda versão: *gameplay* floresta subterrânea e floresta inicial.

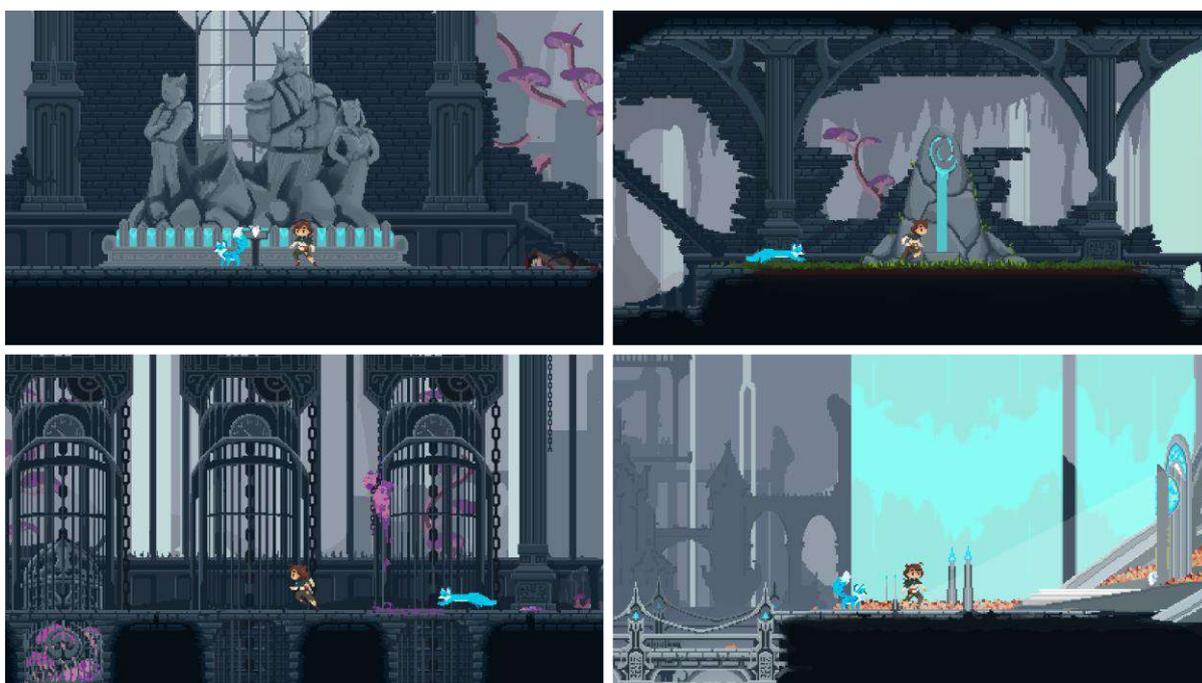


Figura 208 Segunda versão: *gameplay* cidade subterrânea.



Figura 209. Segunda versão: *gameplay* luta com o *boss* alquimista.

Além disso, o mapa foi modificado para oferecer maior variedade aos ambientes, facilitando a localização e proporcionando uma experiência de jogo mais diversificada. O posicionamento das plataformas e dos inimigos foi revisado para melhorar a navegação. A narrativa, agora mais detalhada, foi ajustada para incluir elementos no mapa que ajudam a revelar a trama ao jogador.

Assim, o *Melius* está se desenvolvendo para se tornar um jogo mais completo, com uma narrativa aprimorada, mecânicas ajustadas e uma estética mais refinada, suportado por uma tecnologia que amplia as possibilidades de desenvolvimento.

11. Conclusão

No início deste trabalho, o principal objetivo era criar um jogo *Metroidvania* que fosse completo e divertido, proporcionando ao jogador uma experiência com início, meio e fim. Para isso, aprendemos o processo de criação de um jogo, começando pela fase criativa e lúdica de concepção. Todos os membros do grupo contribuíram para esta etapa, trazendo ideias, referências e desejos para o que eles gostariam que o jogo se tornasse, garantimos que todas elas foram consideradas e validadas pela equipe, sendo elas incluídas ou não no projeto final. Consideramos importante para manter uma relação harmônica e o engajamento da equipe, que todos os membros do grupo sentissem que o projeto lhes pertence. Então, mesmo quando um membro apresentasse uma ideia que se distanciasse muito do jogo que estávamos planejando ou que fosse muito além do escopo do projeto, nós a colocamos em discussão, debatendo as possibilidades de implementação e a sua coerência com o projeto, para ao final decidir se ela seria incluída ou não. Assim, mesmo que sua ideia fosse rejeitada, o membro sente que ela foi validada pela equipe e não apenas descartada imediatamente.

Em seguida, passamos para a organização e execução das etapas de planejamento e produção, abrangendo a definição de elementos do jogo, como narrativa e mecânicas, até a criação de cenários e animações para implementação. Essa etapa, a mais longa do projeto e devido a limitação de tempo, teve seu início antes da anterior ser completamente fechada, exigiu um planejamento e coordenação eficaz para na distribuição de tarefas garantir que todos os membros contribuíssem em uma área de seu interesse e de forma organizada, permitindo que todas as áreas do desenvolvimento fluírem sem grandes obstáculos.

Também estudamos e planejamos o estilo artístico escolhido, o *pixel art*. Um dos desafios que sabíamos que tínhamos que enfrentar era garantir que o estilo artístico do jogo se mantivesse consistente, apesar de ter diversos artistas trabalhando simultaneamente na criação de *assets*. Para enfrentar esse problema, começamos por desenvolver a etapa de pesquisa de maneira detalhada e estruturada, em grupo, montamos diversos *moodboards* que serviram de referência para diferentes aspectos do nosso jogo, cada um foi nomeado e categorizado, para que todos os membros tomassem partida deles para criação das artes. Outra medida adotada foi que, apesar da divisão de tarefas para cada membro do grupo, as reuniões e trabalhos em grupo se mantiveram frequentes, para que os membros pudessem fazer consultas entre si, pedir ajuda caso necessário e comparar e compartilhar os *assets* desenvolvidos até então.

Além dos aspectos gerais do desenvolvimento, abordamos características específicas do gênero *Metroidvania*, características estas que tiveram peso mais significativo nas etapas de criação da história, mecânicas e mapas. Foi necessário um planejamento mais cuidadoso e etapas extras de pesquisa, recorremos ao estudo de jogos clássicos e recentes e também de pesquisadores e Game Designers mais experientes no desenvolvimento de jogos e no gênero *Metroidvania*. Essas questões, muitas vezes não abordadas na graduação, foram fundamentais para a primeira versão do jogo.

Outro desafio para o projeto *Melius* foi a organização projetual. Como dito anteriormente, partimos de uma ideia e percorremos um caminho até termos um jogo interativo. Durante este processo nós precisamos organizar e distribuir as etapas para que elas pudessem ser desenvolvidas de maneira simultânea. Para isso, com base no processo de Game Design e no nosso conhecimento sobre o gênero *Metroidvania*, procuramos mapear tudo que precisaria ser desenvolvido, identificando também quais etapas dependeriam de outras para serem iniciadas, colocando como prioridade as etapas que possuíam um maior número de dependências dela.

Um aspecto muito importante do processo foi a implementação e testagem antecipada de elementos, como as mecânicas, as animações, ainda em silhueta, e o *level design* construído ainda no TileD. Isso permitiu avaliarmos como o *gameplay* estava funcionando logo cedo no desenvolvimento. Assim, nesse momento, pudemos testar e reconsiderar mecânicas de jogo, ajustar as animações, tanto na movimentação quanto na contagem de frames, e refinar o *level design*, corrigindo o posicionamento de plataformas e ajustando altura das fases, dentre outras coisas.

Ao final da disciplina de Design de Jogos, alcançamos um resultado satisfatório: um jogo *Metroidvania* completo. No entanto, após testes livres realizados por nós mesmos e outros jogadores, identificamos elementos mal planejados ou que poderiam ser aprimorados ou expandidos para tornar a experiência de jogo mais agradável e interessante. O que nos levou a continuar os trabalhos no jogo, dando início ao primeiro ciclo de revisão.

Com os ciclos de revisão, buscamos melhorar esses aspectos e expandir o escopo do projeto para tornar a experiência mais completa e envolvente. No entanto, o aumento significativo do escopo trouxe novas dificuldades. A principal delas foi a falta de tempo, sem mais a disciplina de DJ para garantir encontros semanais da equipe, a realização de reuniões regulares se tornou um desafio. O aumento do escopo, combinado com a saída de alguns

membros, a diminuição do número de encontros e a dificuldade de reunir todos com regularidade, resultou em um desenvolvimento mais lento e desorganizado e a implementação de novos elementos no jogo mais desafiador do que o esperado. Buscamos alternativas para remediar a situação, como manter contato frequente com a equipe, mantendo todos os membros do grupo atualizados sobre o andamento do projeto, ou até a participação especial na disciplina de DJ novamente com o projeto. Isso fez com que o projeto voltasse a se mover, mas este ainda segue como a maior dificuldade que enfrentamos no momento.

Ainda assim, o aprendizado sobre a gestão do processo de Game Design, assim como o domínio técnico dos recursos necessários para produção dos *assets* e organização dos entregáveis para comunicação da equipe permitiram o refinamento e evolução da experiência pretendida de maneira satisfatória. Portanto, entendemos que o objetivo do presente projeto foi atingido.

Para o futuro do Melius, o plano é continuar o trabalho no jogo, aplicando e testando as decisões tomadas no último ciclo de revisão. O objetivo permanece criar um jogo completo e envolvente, com uma narrativa que tenha início, meio e fim, e que cativa o jogador com o universo que estamos desenvolvendo. Para tanto, será necessário sistematizar melhor as etapas de *playtest*, organizando o processo de revisão do jogo a partir do levantamento e análise de situações de *gameplay*.

Para aqueles que desejam criar um jogo, oferecemos as seguintes recomendações:

- Realizar reuniões regulares para manter o grupo unido e alinhado quanto ao andamento do desenvolvimento do jogo. Não é preciso que todos os membros da equipe estejam presentes em todas as reuniões, mas é importante que os membros não presentes sejam informados do que foi discutido, alterado ou decidido sobre o projeto na reunião;
- Anotar as decisões tomadas nas reuniões, deixando-as registradas para que o projeto esteja sempre avançando. Assim, evita-se que a equipe acabe voltando atrás, por esquecimento ou memórias destoantes sobre as decisões tomadas, e acabe por ter que discutir novamente sobre questões previamente estabelecidas, ou que as decisões acabem sendo implementadas de maneira diferente pelos membros da equipe, por dependerem apenas da memória para lembrar dessas decisões. É importante também, deixar todas essas anotações facilmente acessíveis para todo o grupo.
- Garantir que todas as opiniões sejam consideradas e organizar as ideias e referências de todos os membros do grupo, dando oportunidade para que todos participem e opinem

nas decisões, e sintam-se envolvidos com o projeto. Isso ajuda a manter cada um dos membros engajado, já que suas sugestões importam e influenciam no desenvolvimento do jogo;

- Definir claramente as responsabilidades e tarefas para cada membro, de forma a manter o processo organizado e garantir que todos saibam como contribuir e que todos os membros saibam no que cada um está contribuindo. Dessa forma a produção fica mais organizada e flui de forma mais eficiente;
- Manter o escopo do projeto realista em relação ao tempo disponível, tratando ideias adicionais como extras, para que o projeto não fique grande demais para ser finalizado e ainda acabe diminuindo a confiança do grupo em conseguir finalizá-lo.

Referências Bibliográficas

DONOVAN, Tristan. *Replay: the History of Video Games*. Edição Kindle. Lewes: Yellow Ant, 2010.

JUUL, Jesper. Games telling stories? *Game Studies, the international journal of computer game research*. V. 1, n. 1. Disponível em: <http://www.gamestudies.org/0101/juul-gts/> Acesso em: 16 de Julho de 2024.

JUUL, Jesper. The Game, the Player, the World: Looking for a Heart of Gameness. *Level Up: Digital Games Research Conference Proceedings*. V. 1, n. 1. Disponível em: <https://www.jesperjuul.net/text/gameplayerworld> Acesso em: 16 de Julho de 2024.

KENT, Steven. *The Ultimate History of Video Games: From Pong to Pokémon and Beyond*. Edição Kindle. Nova Iorque: Three Rivers Press, 2001.

RABIN, Steve. *Introduction to Game Development*. Charles River Media, 2005.

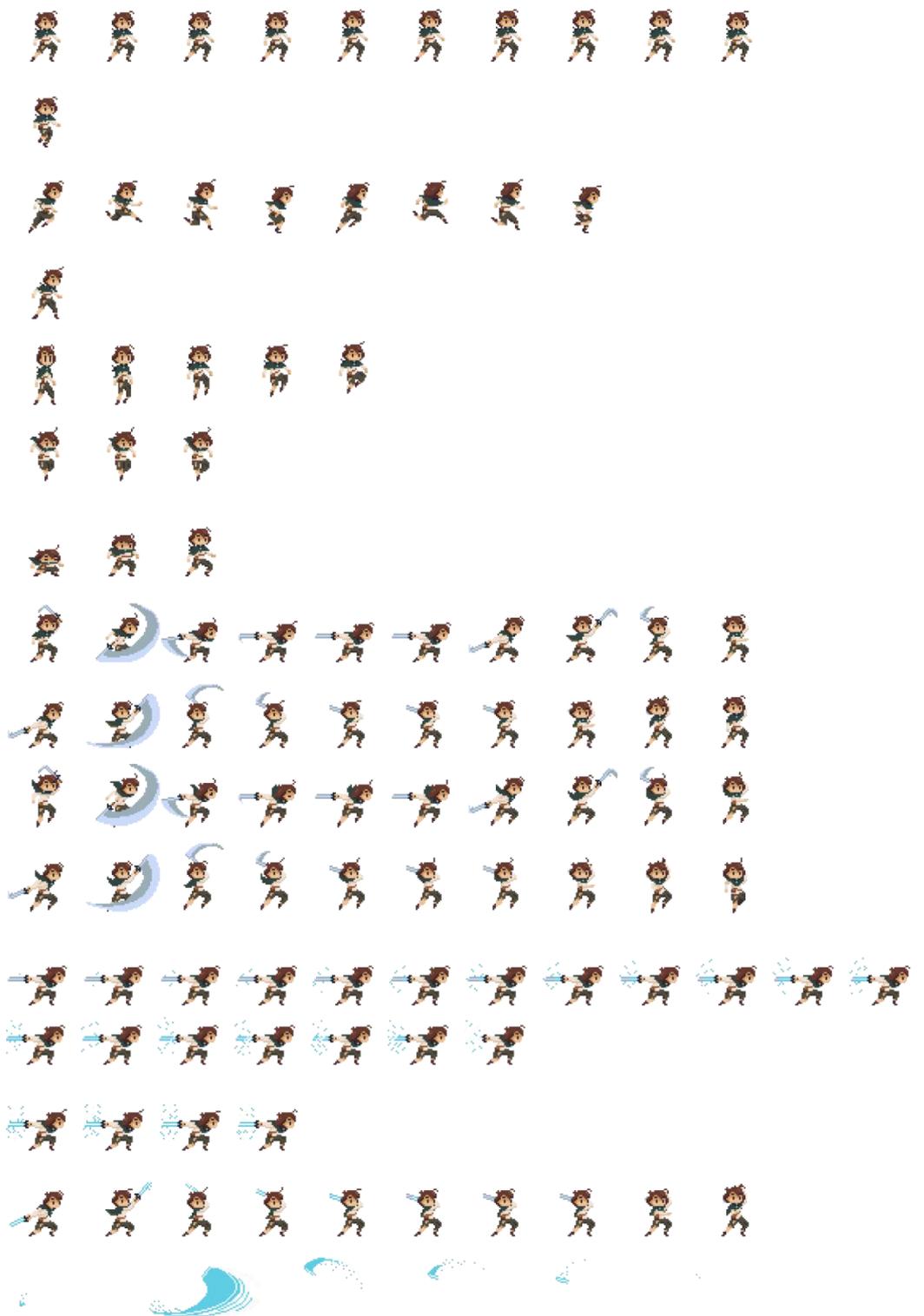
SALEN, K.; ZIMMERMAN, E. *Rules of Play: Game Design Fundamentals*. Cambridge: MIT Press, 2004.

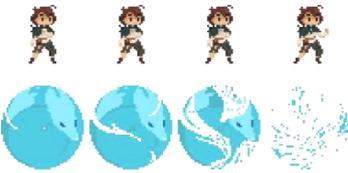
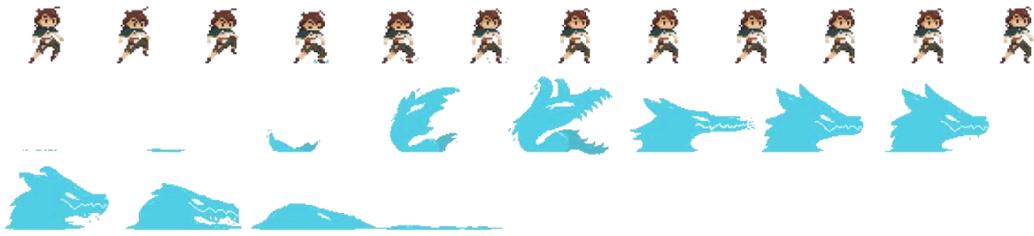
SCHELL, Jesse. *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann, 2014.

MIYAMOTO, S; TEZUKA, T. *Miyamoto on World 1- 1: How Nintendo made Mario's most Iconic level*. [Youtube]. Reino Unido: Eurogamer, 2015. Disponível em: <https://youtu.be/zRGRJRUWafY?si=VHExi-7p7Wyfa8Y1> Acesso em: 18 de agosto de 2024.

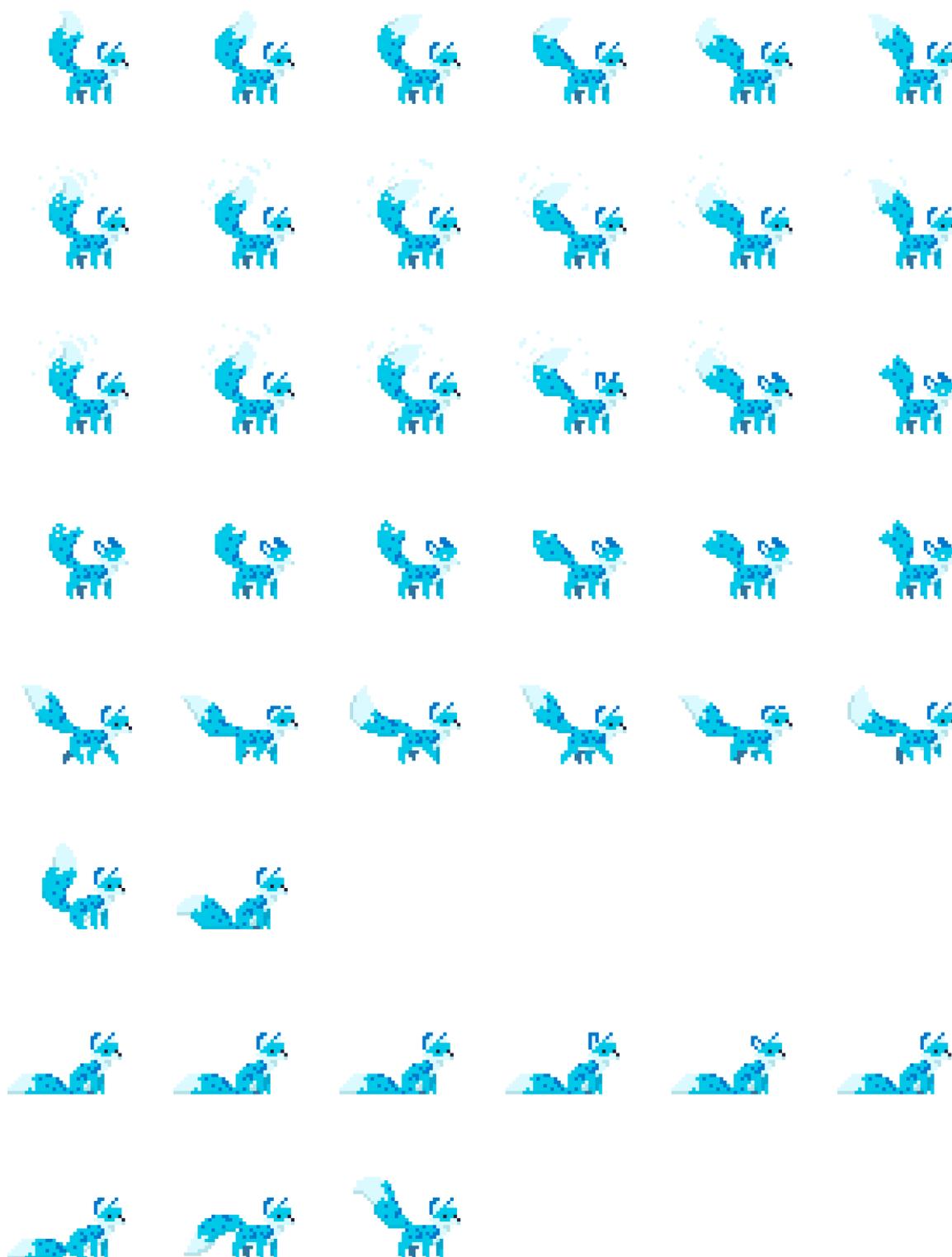
Apêndice A: Lista de *spritesheets* dos personagens

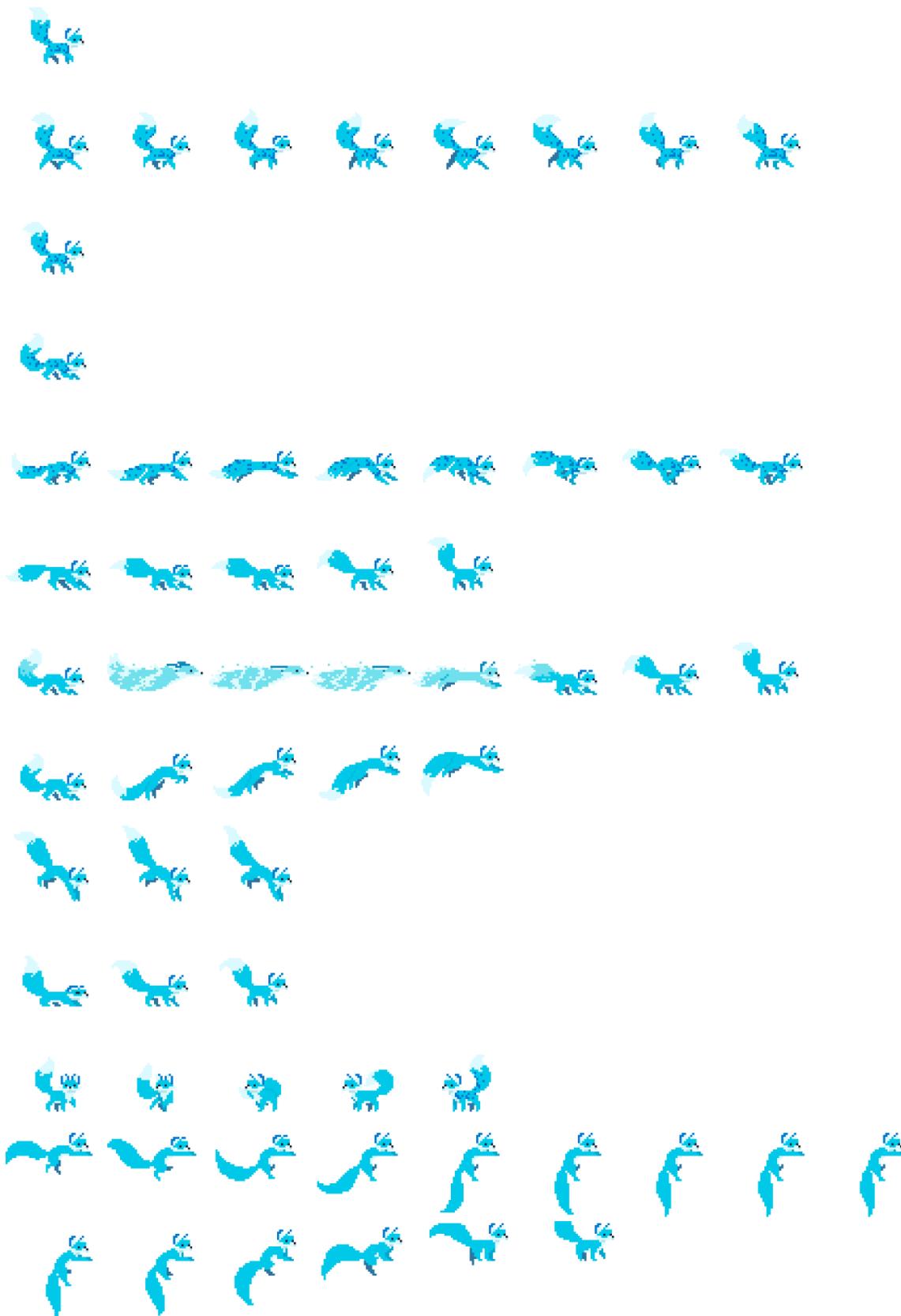
Spritesheets do Protagonista





Spritesheets do Protagonista com alterações do ciclo de revisão

Spritesheets da Raposa

Spritesheets da Raposa criadas no ciclo de revisão

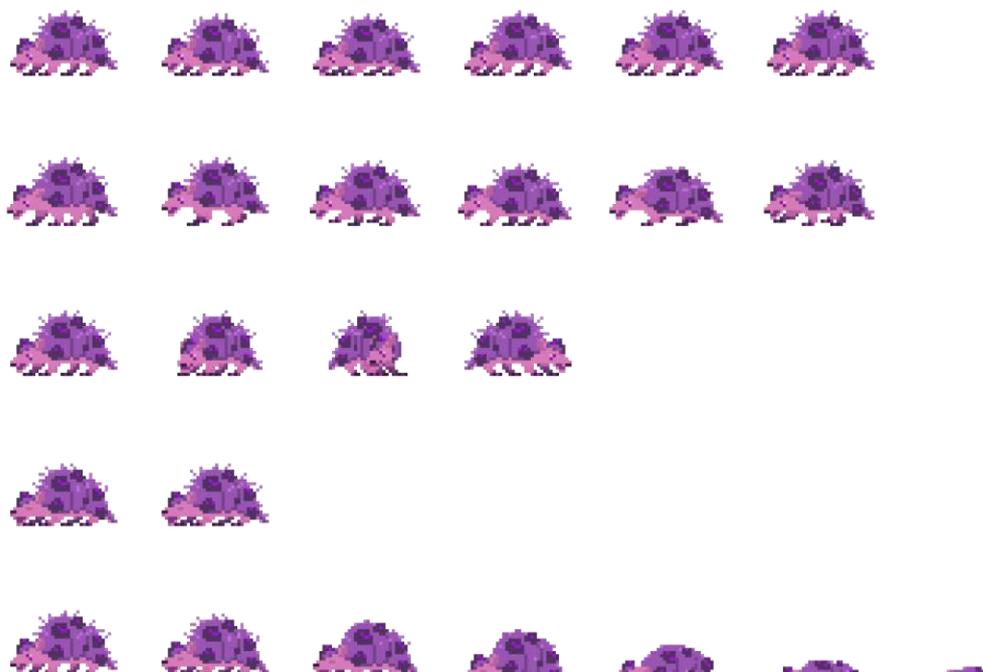
Spritesheets da Mãe do Protagonista



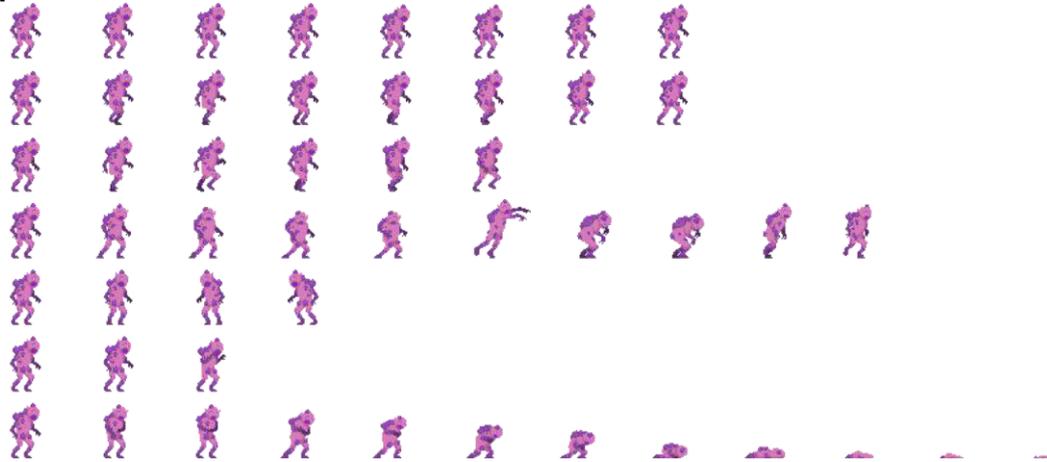
Spritesheets do Alquimista



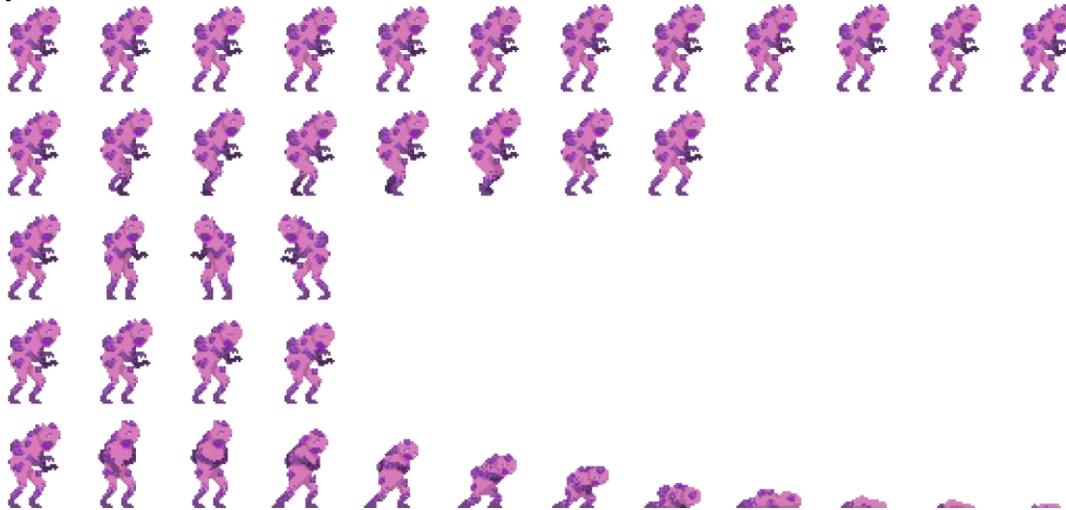
Spritesheets do Rastejante



Spritesheets do Infectado



Spritesheets do Corredor

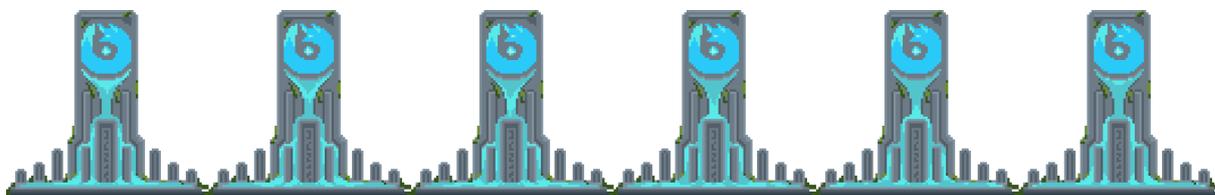


Spritesheets do Boss



Apêndice B: Lista de *spritesheet* de assets dos cenários

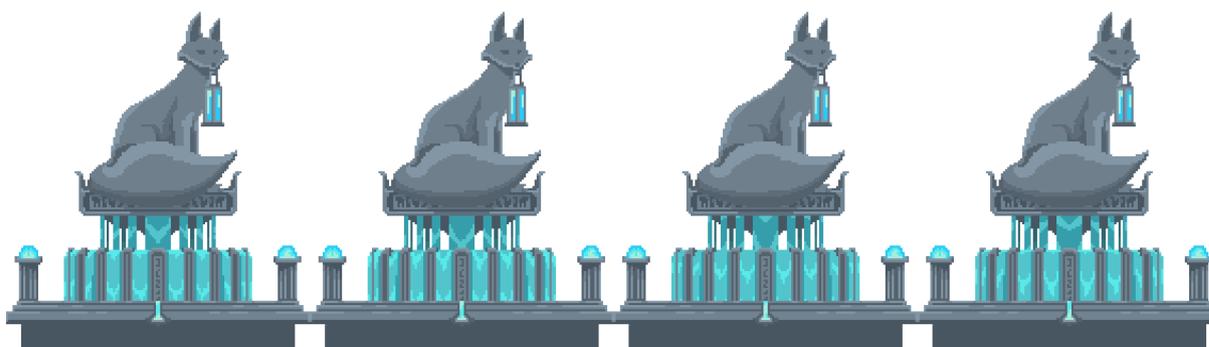
Spritesheet primeira versão do checkpoint



Spritesheet segunda versão do checkpoint



Spritesheet fonte da raposa

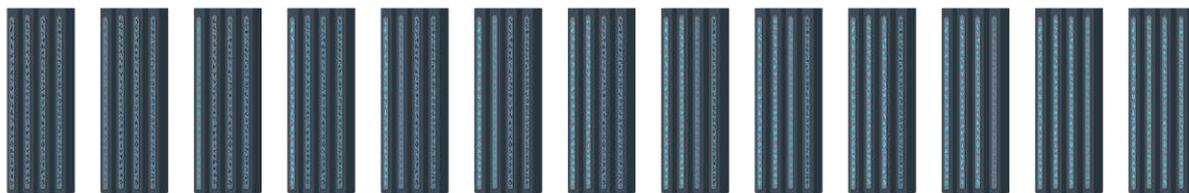
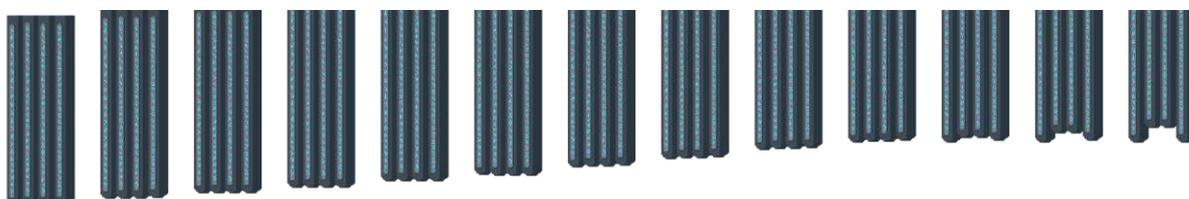
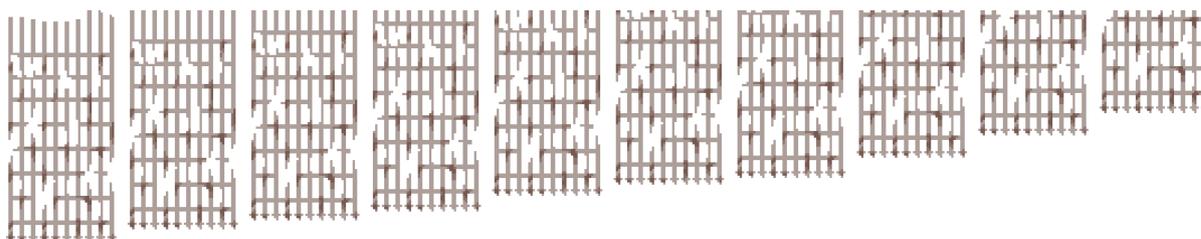
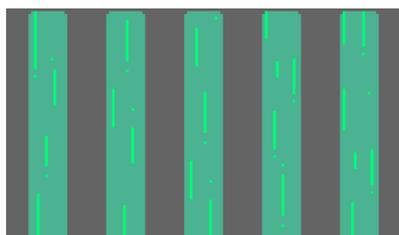


Spritesheet fonte dos quatro heróis



Spritesheet mecanismo



Spritesheet fonte genérica**Spritesheet portão de saída da cidade subterrânea ativando****Spritesheet portão de saída da cidade subterrânea abrindo****Spritesheet portão da entrada alternativa da cidade subterrânea abrindo****Spritesheet barreira mágica****Spritesheet fogo**

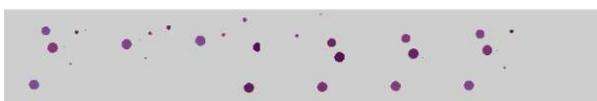
Spritesheet item chave



Spritesheet nuvem de poeira



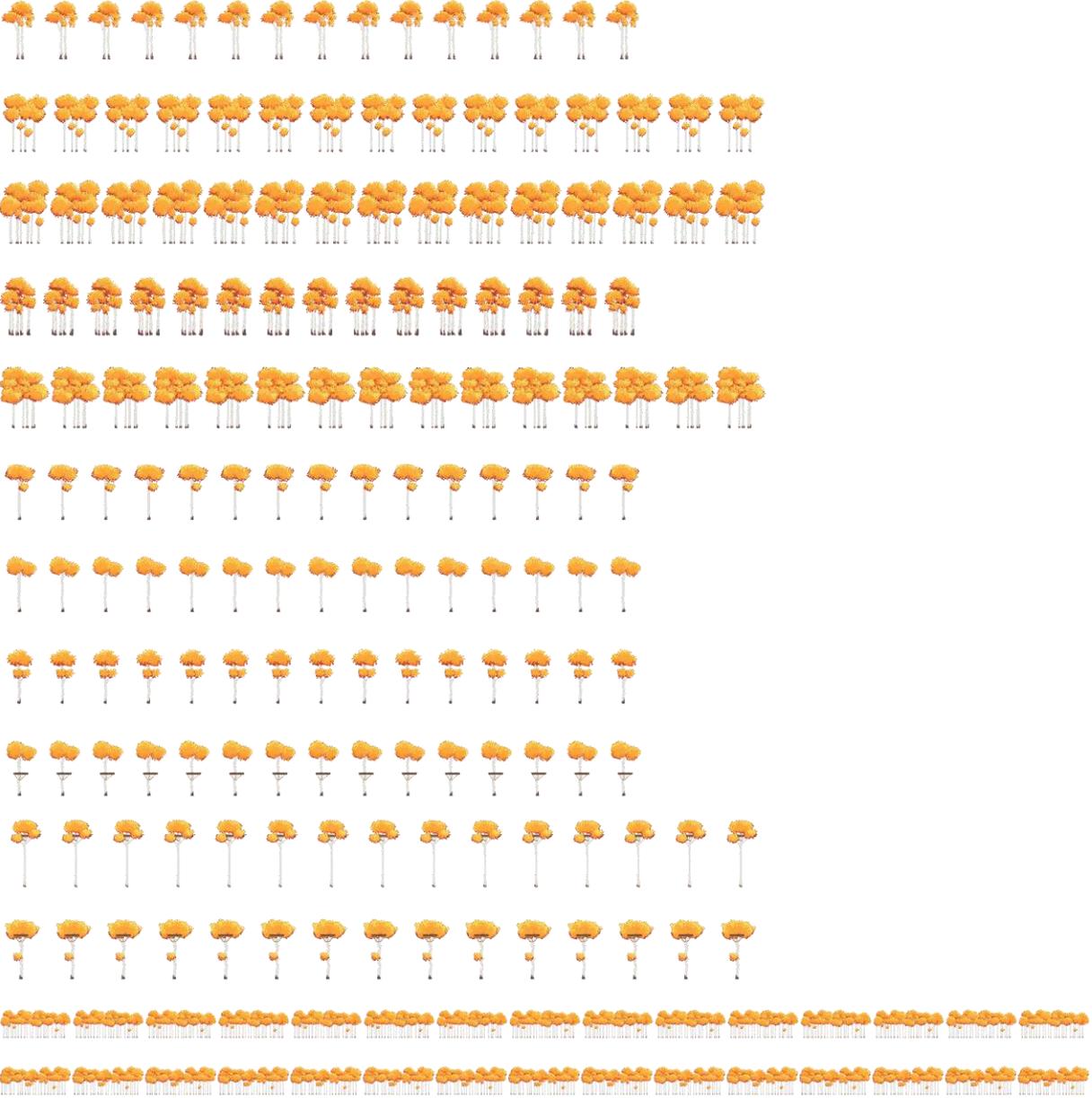
Spritesheets partículas da infecção



Spritesheets cachoeiras



Spritesheets árvores da superfície



Spritesheets árvores da superfície infectadas



Spritesheets árvores da floresta subterrânea



Apêndice C: Cenários completos

Cenário interior da casa do protagonista



Cenário floresta (primeira versão) 01



Cenário floresta (primeira versão) 02



Cenário floresta (primeira versão) 03-a



Cenário floresta (primeira versão) 03-b

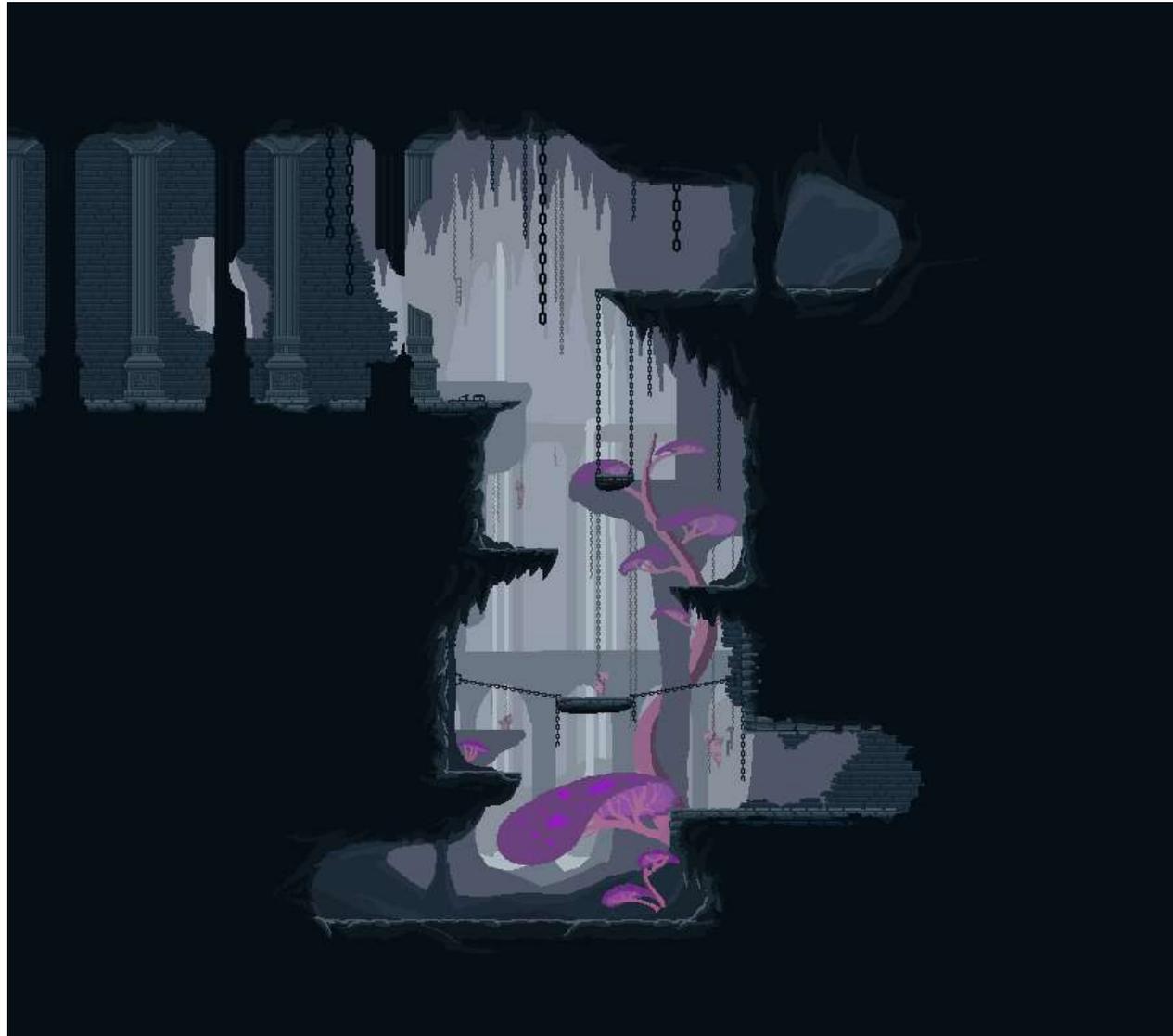


Cenário floresta (primeira versão) 04

Cenário cidade subterrânea (primeira versão) 01



Cenário cidade subterrânea (primeira versão) 02



Cenário cidade subterrânea (primeira versão) 03



Cenário cidade subterrânea (primeira versão) 04

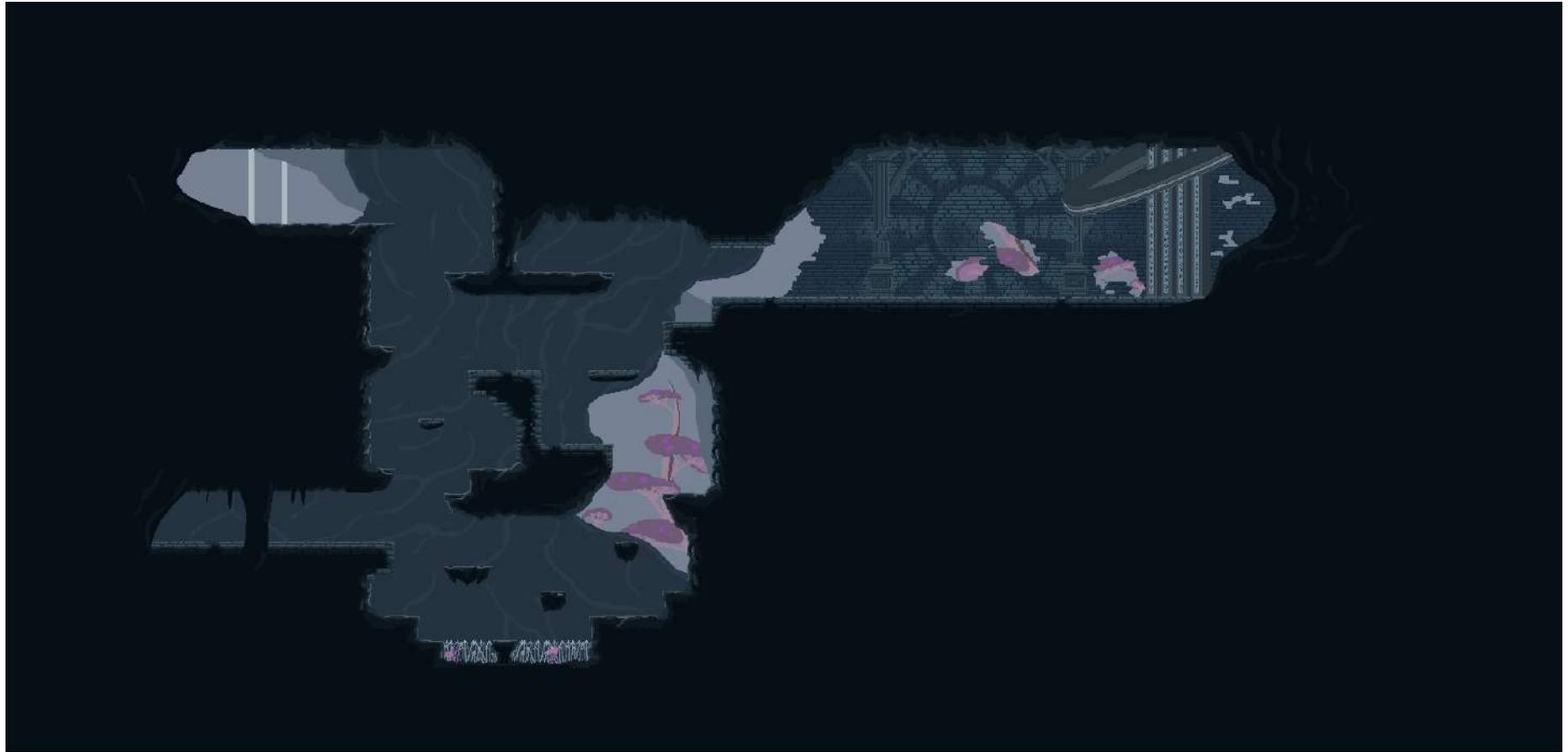


Cenário cidade subterrânea (primeira versão) 05

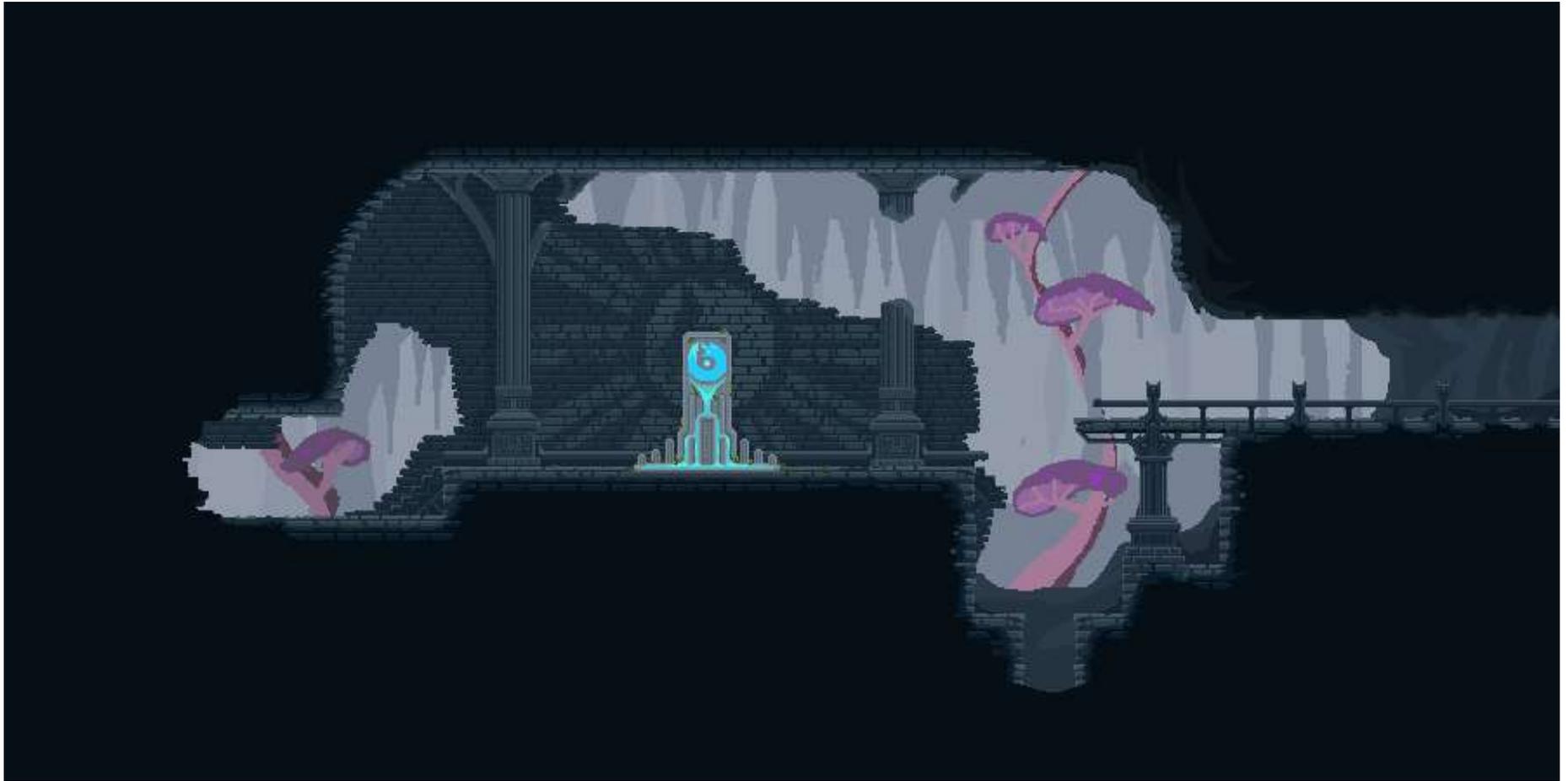


Cenário cidade subterrânea (primeira versão) 06

Cenário cidade subterrânea (primeira versão) 07



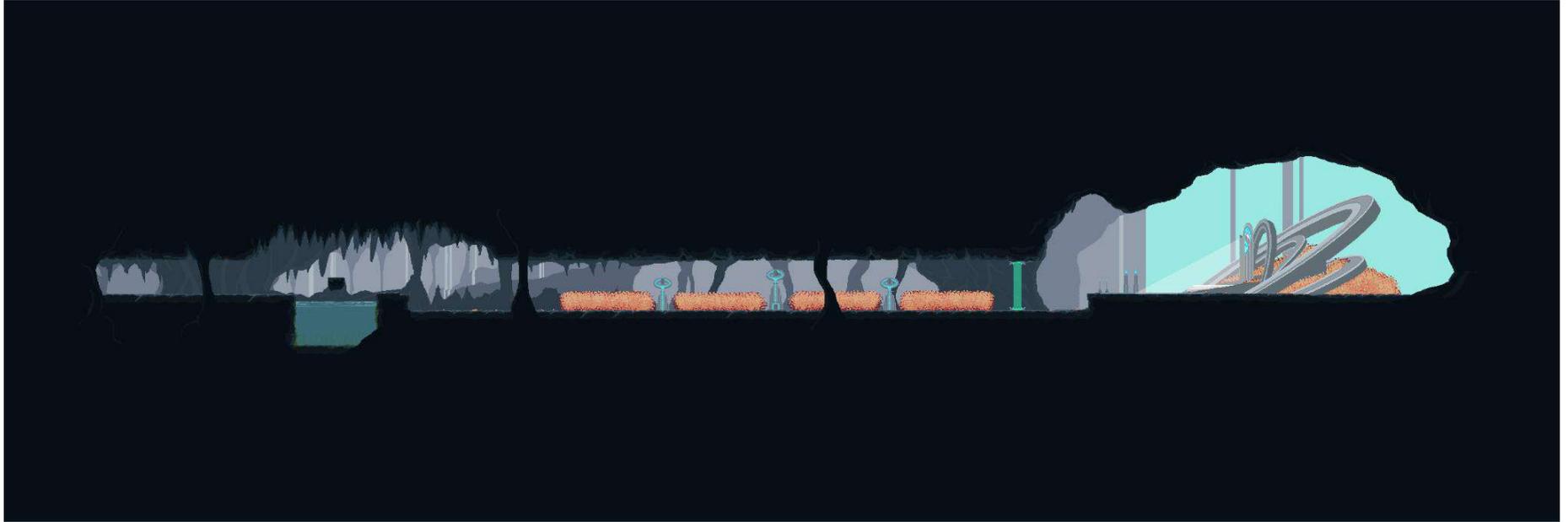
Cenário cidade subterrânea (primeira versão) 08



Cenário cidade subterrânea (primeira versão) 09

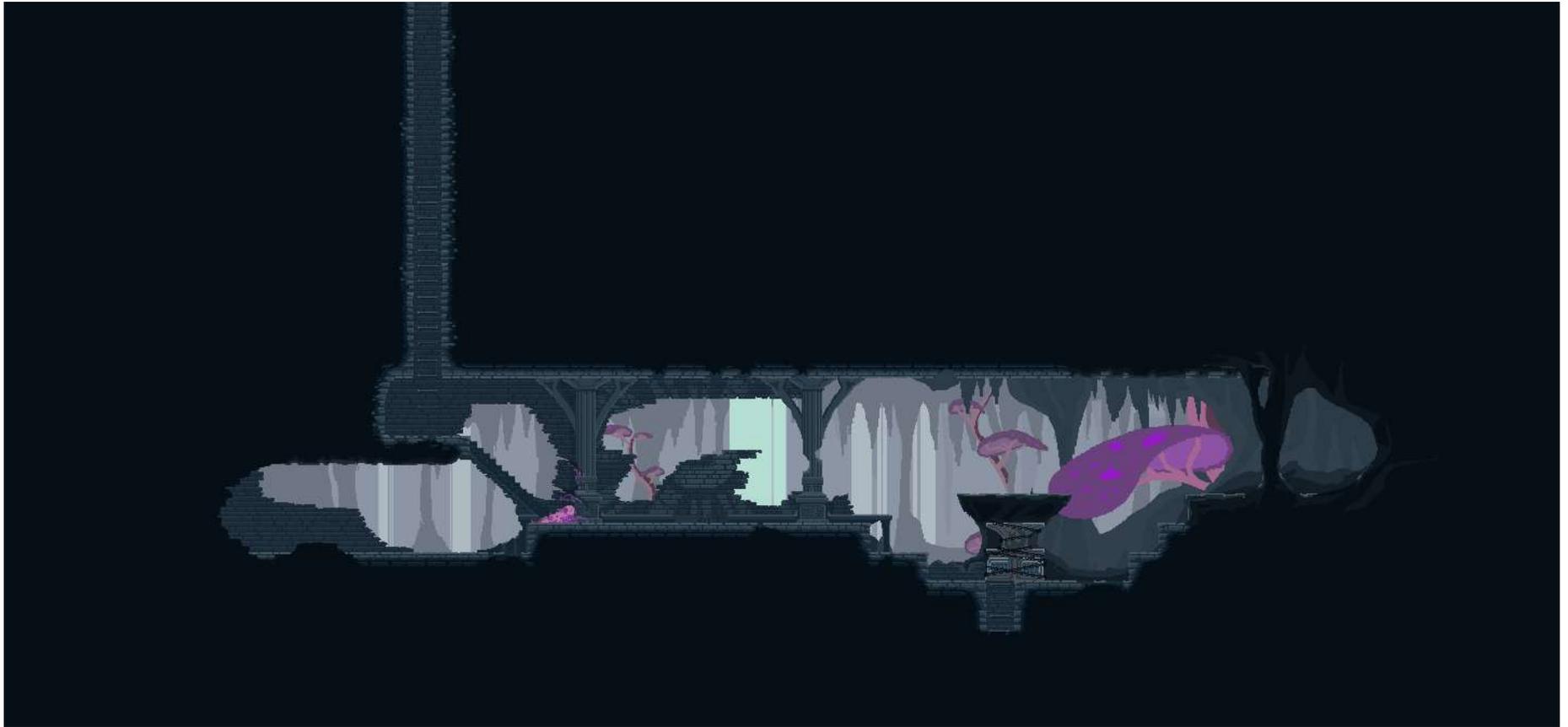


Cenário cidade subterrânea (primeira versão) 10



Cenário cidade subterrânea (primeira versão) 11

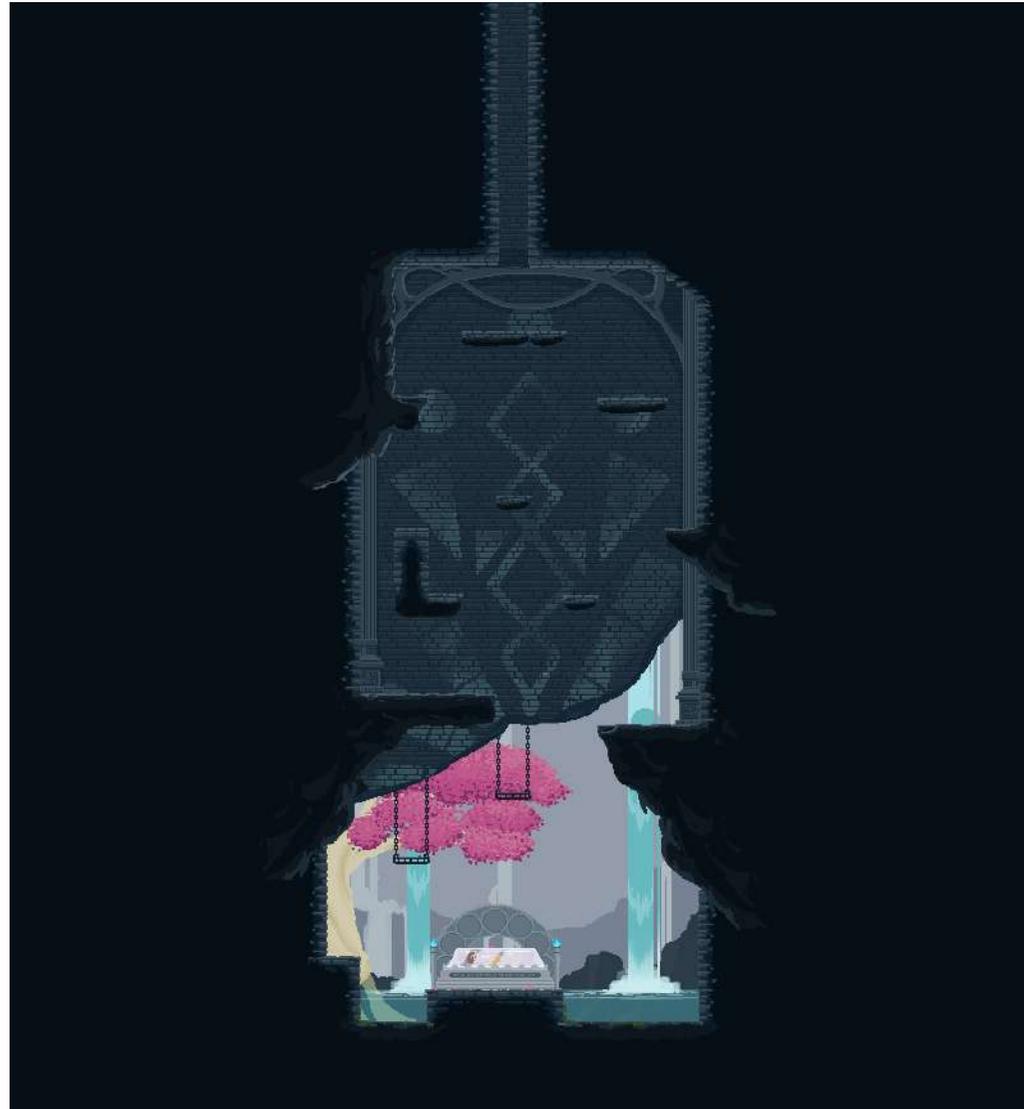
Cenário cidade subterrânea (primeira versão) 12



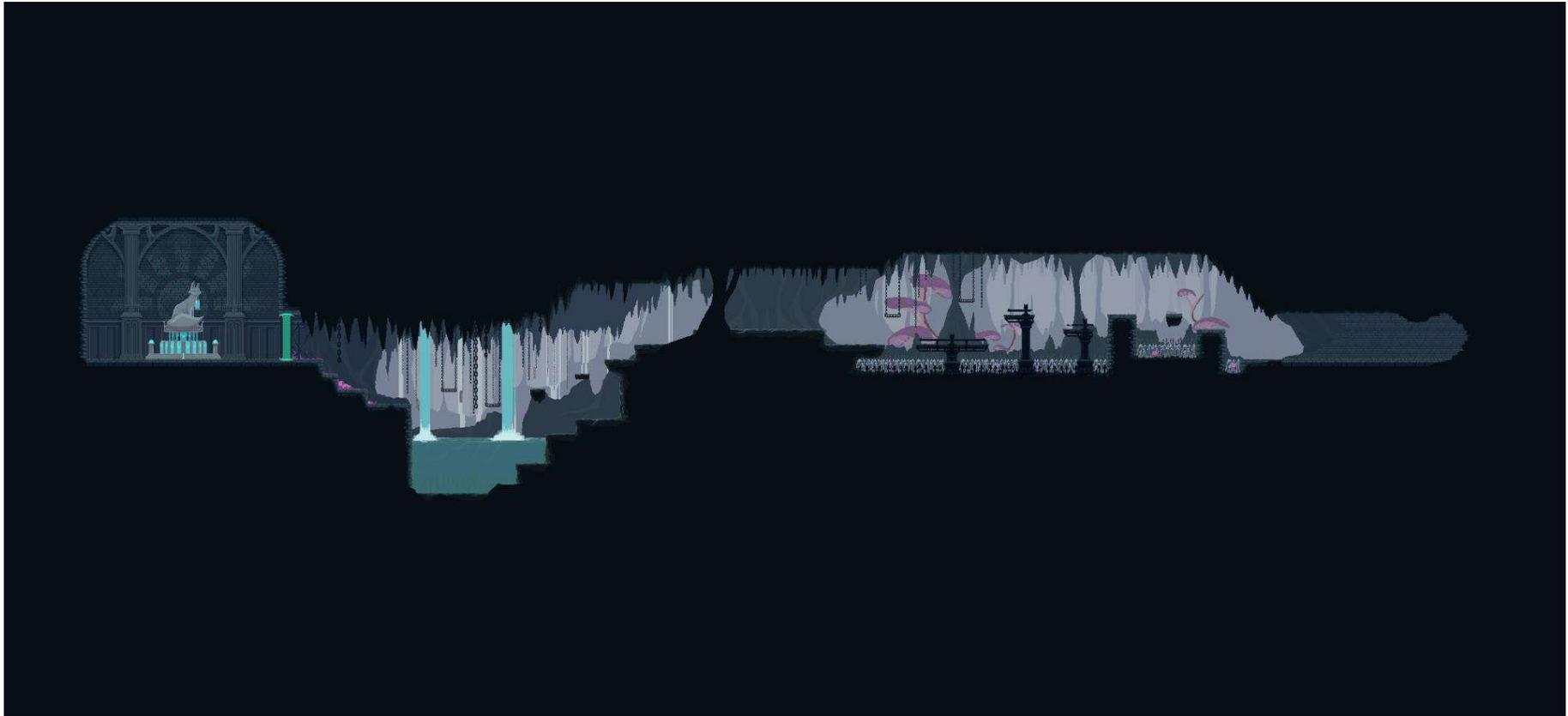
Cenário cidade subterrânea (primeira versão) 13



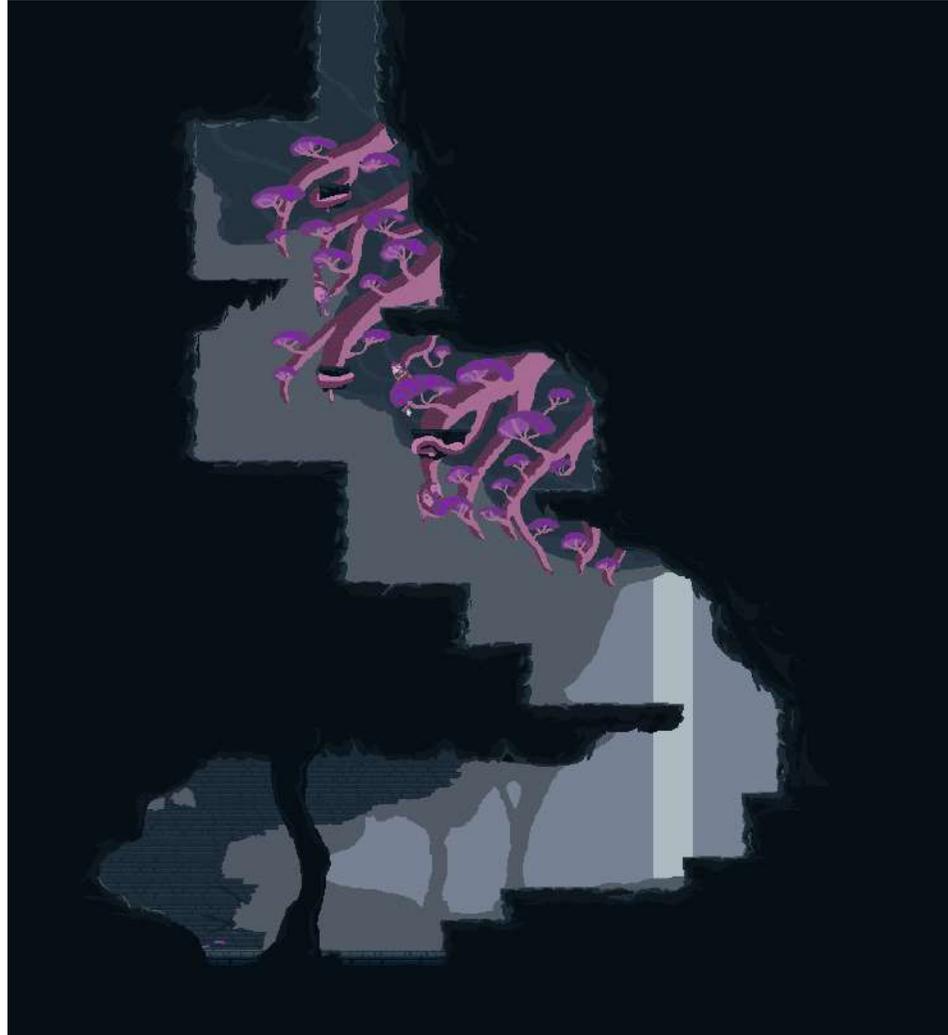
Cenário cidade subterrânea (primeira versão) 14



Cenário cidade subterrânea (primeira versão) 15



Cenário cidade subterrânea (primeira versão) 16

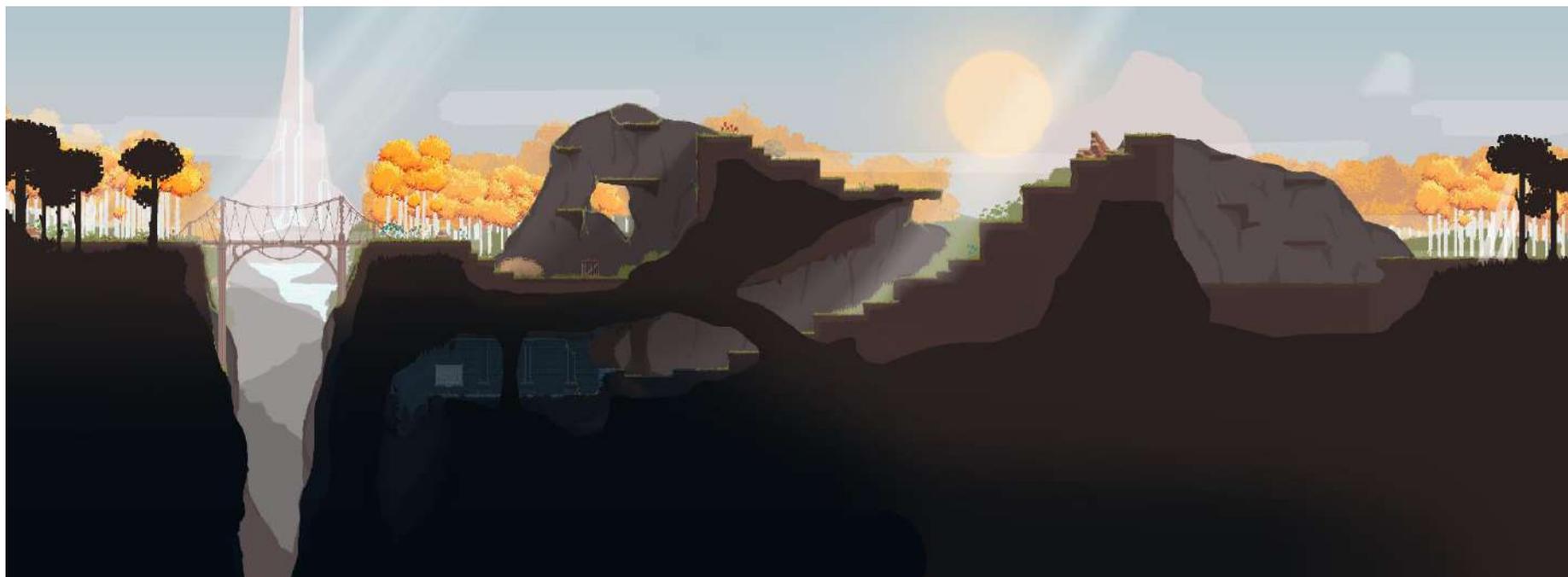


Cenário floresta subterrânea 01

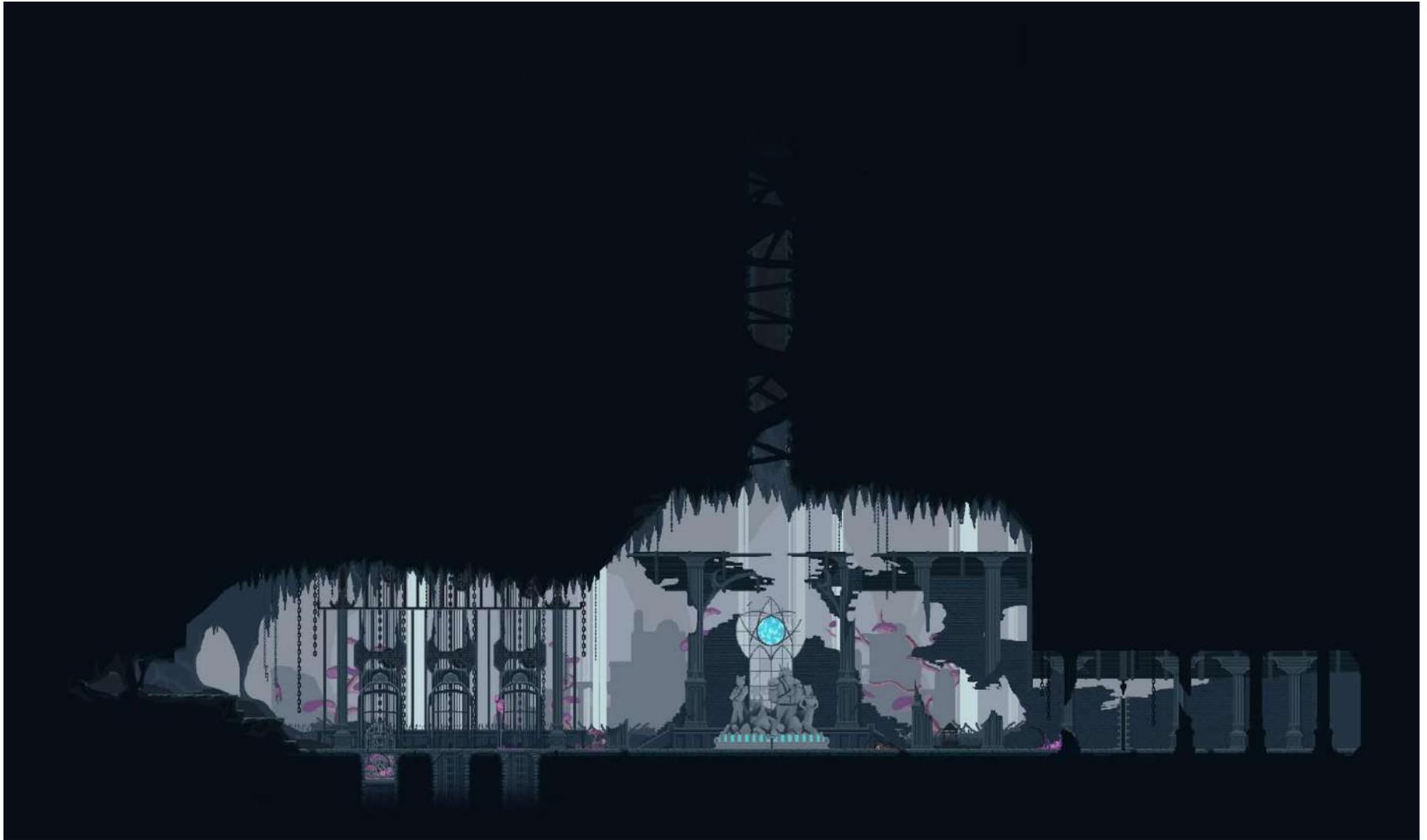


Cenário floresta (segunda versão) 02

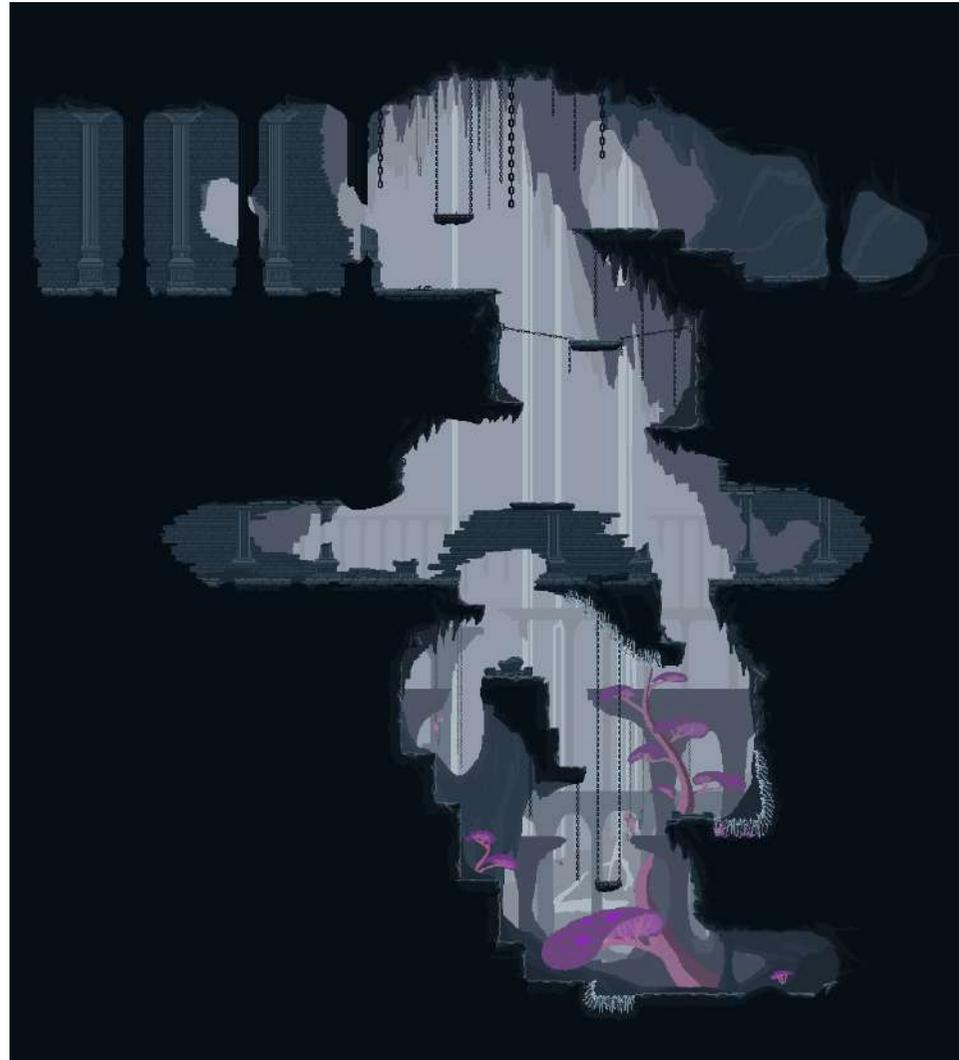


Cenário floresta (segunda versão) 03

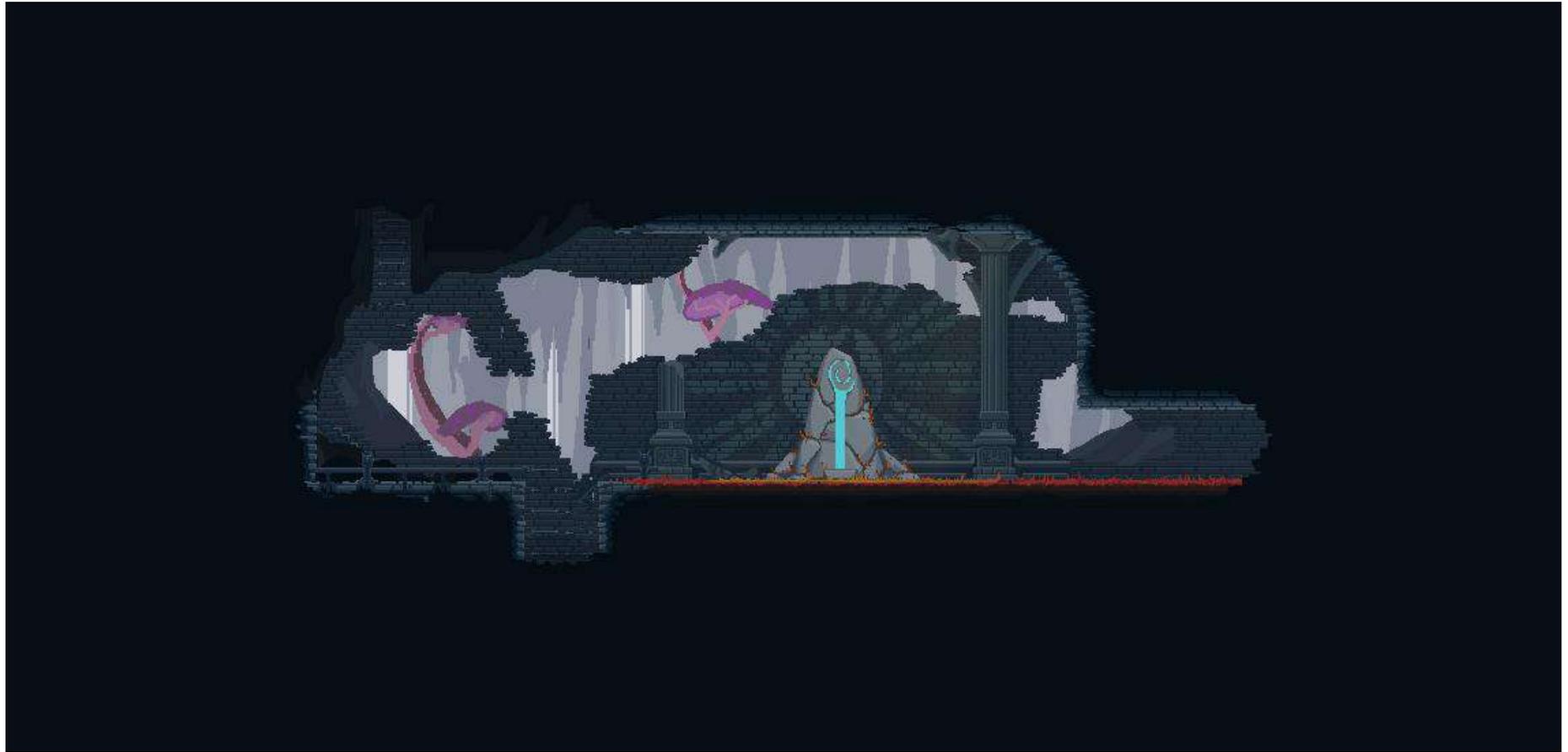
Cenário cidade subterrânea (segunda versão) 01

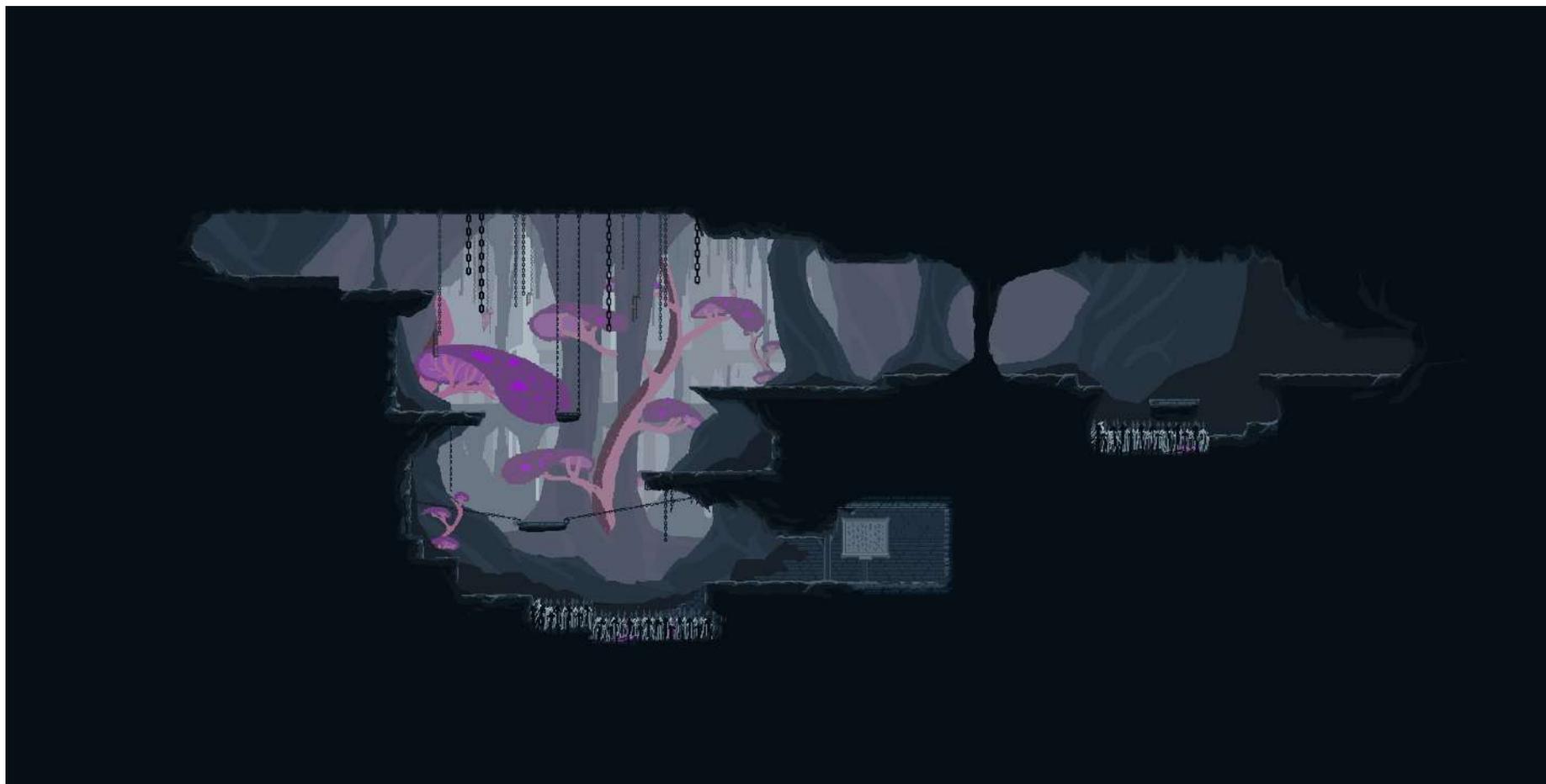


Cenário cidade subterrânea (segunda versão) 02



Cenário cidade subterrânea (segunda versão) 06-1

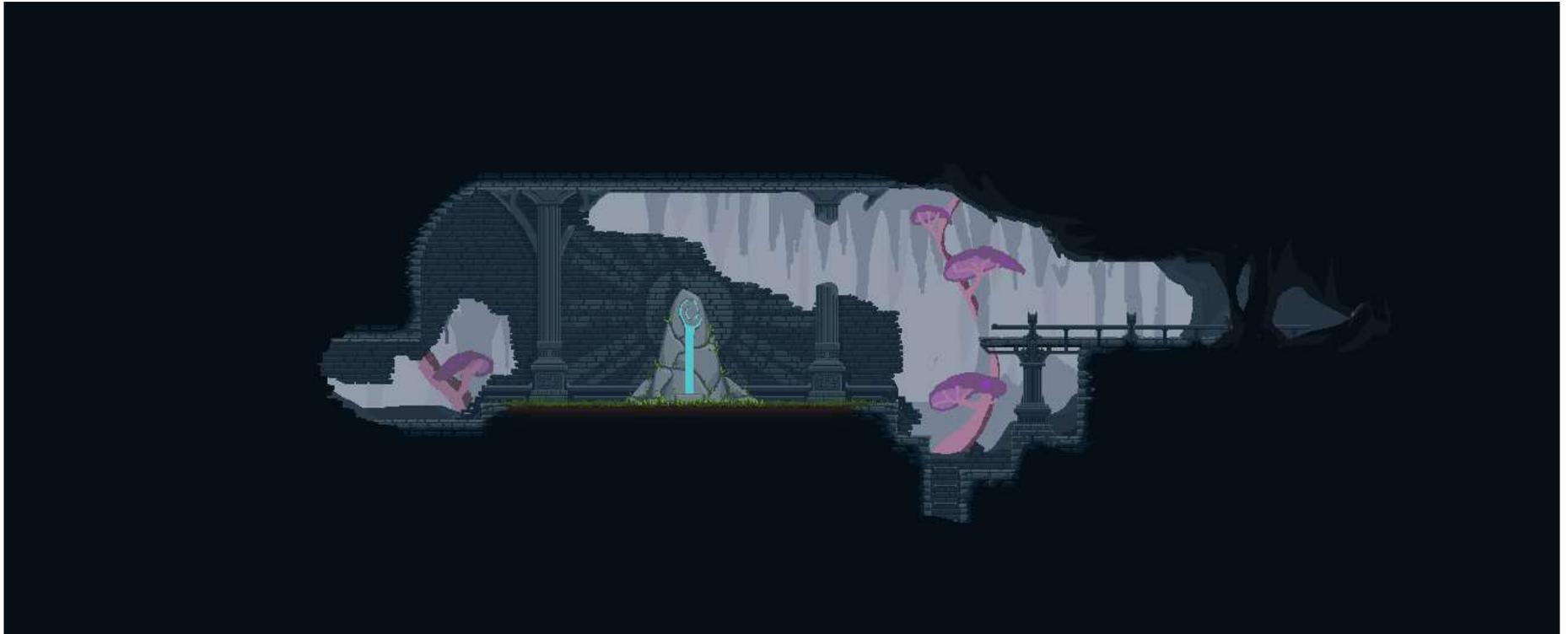


Cenário cidade subterrânea (segunda versão) 07

Cenário cidade subterrânea (segunda versão) 08



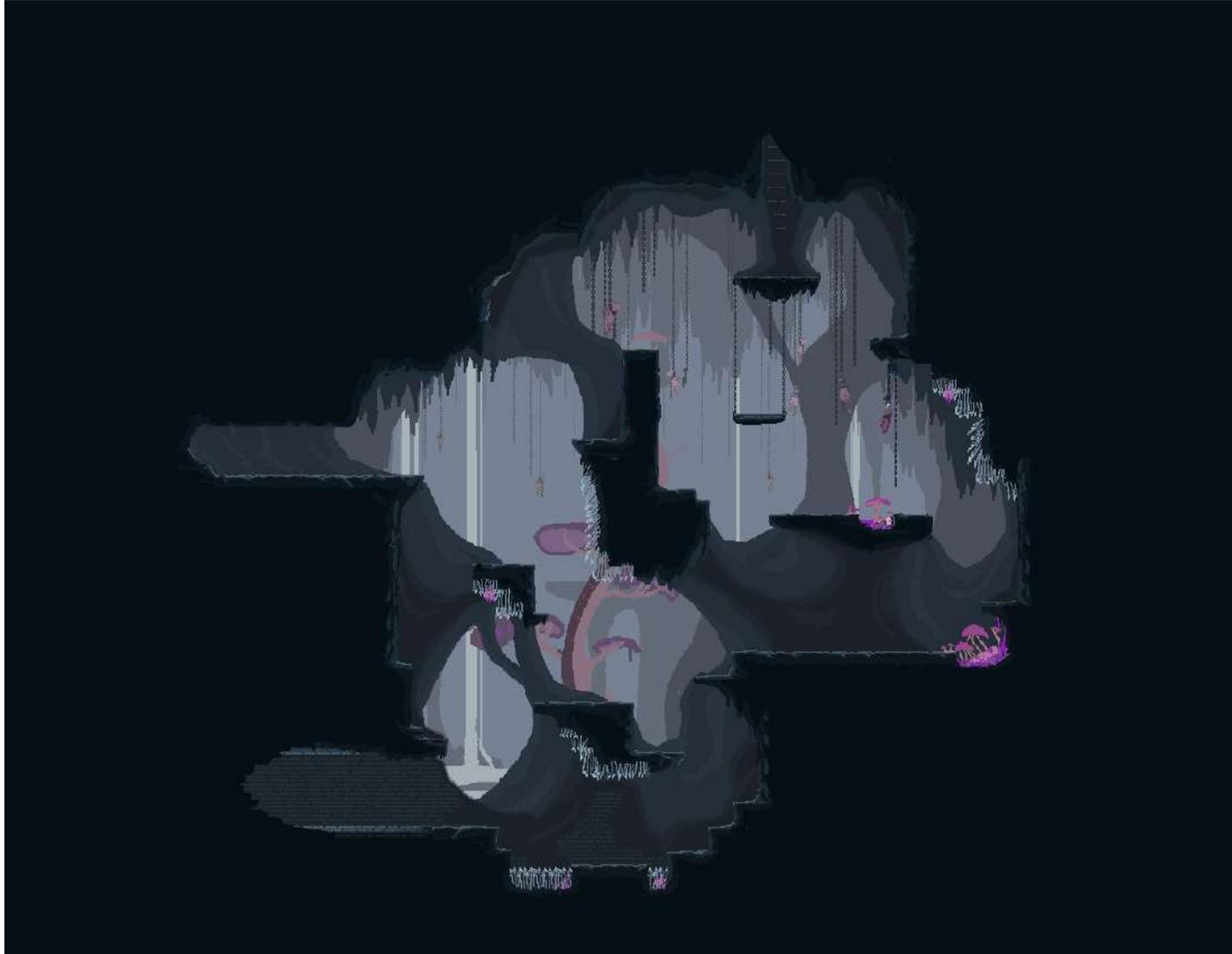
Cenário cidade subterrânea (segunda versão) 10



Cenário cidade subterrânea (segunda versão) 11



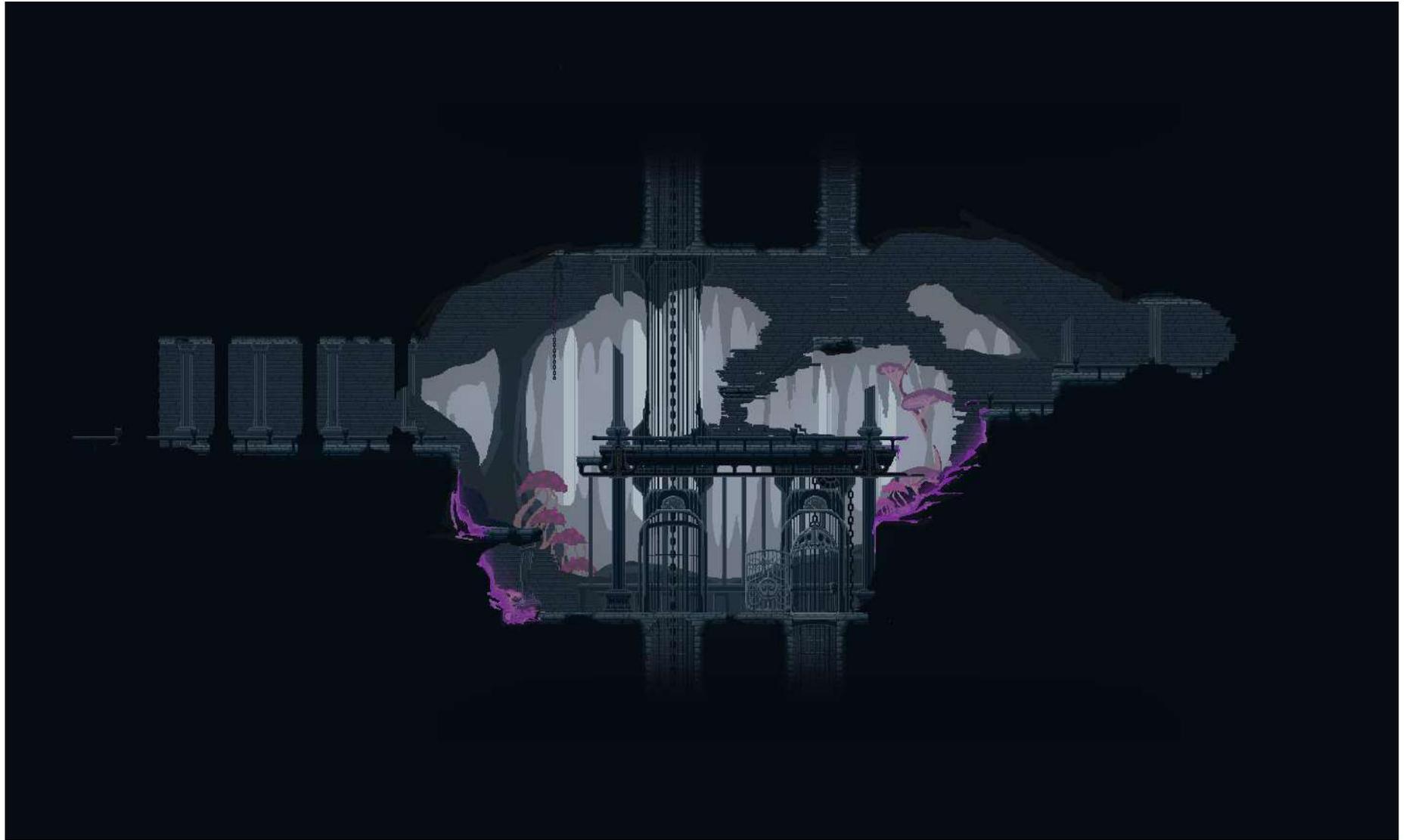
Cenário cidade subterrânea (segunda versão) 12



Cenário cidade subterrânea (segunda versão) 15



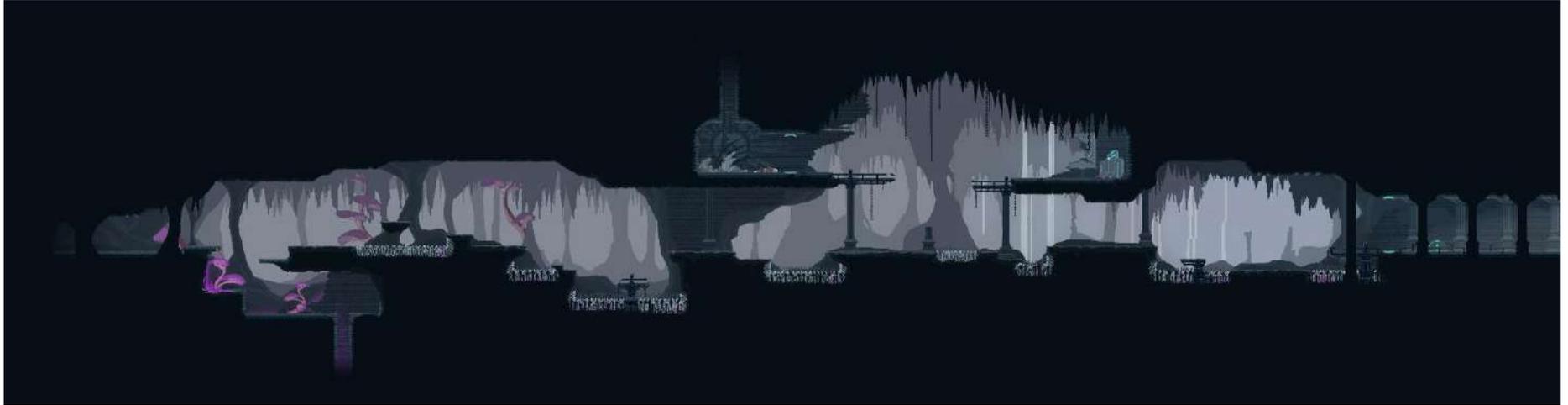
Cenário cidade subterrânea (segunda versão) 16



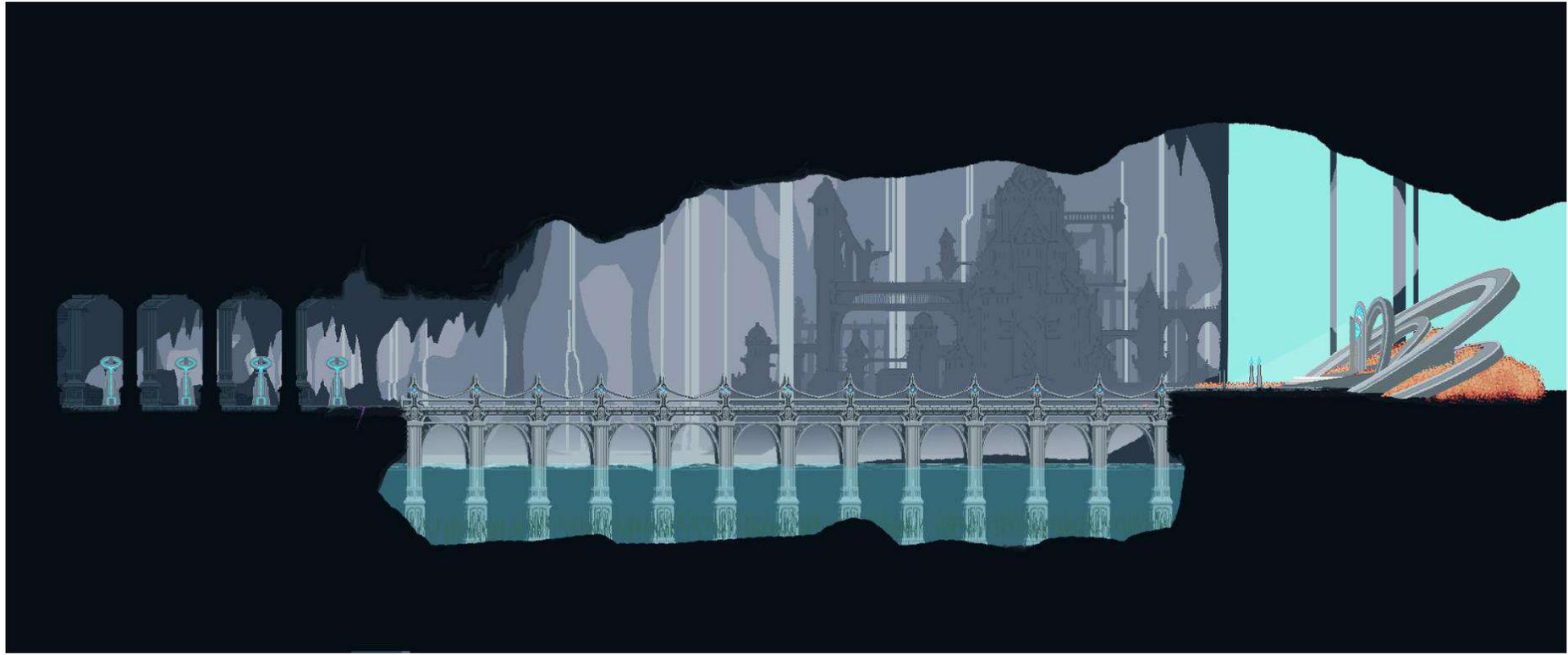
Cenário cidade subterrânea (segunda versão) 17



Cenário cidade subterrânea (segunda versão) 18



Cenário cidade subterrânea (segunda versão) 21



Cenário cidade subterrânea (segunda versão) 22



Cenário cidade subterrânea (segunda versão) 24



Cenário cidade subterrânea (segunda versão) 25

