

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUANA RODRIGUES DA SILVA E LIMA

**APLICAÇÃO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL
NA ANIMAÇÃO DE COMPONENTES EM UM JOGO 2D VOLTADO
À SAÚDE MENTAL**

BAURU
Novembro/2025

LUANA RODRIGUES DA SILVA E LIMA

**APLICAÇÃO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL
NA ANIMAÇÃO DE COMPONENTES EM UM JOGO 2D VOLTADO
À SAÚDE MENTAL**

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru. Orientador: Profa. Dra. Juliana da Costa Feitosa

BAURU
Novembro/2025

L732a

Lima, Luana Rodrigues da Silva e

Aplicação de ferramentas de Inteligência Artificial na
animação de componentes em um Jogo 2D voltado à Saúde
Mental / Luana Rodrigues da Silva e Lima. -- Bauru, 2025

121 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Ciência da
Computação) - Universidade Estadual Paulista (UNESP),
Faculdade de Ciências, Bauru

Orientadora: Juliana da Costa Feitosa

1. Inteligência artificial. 2. Jogos eletrônicos. 3. Jogos
eletronicos Industria. 4. Computer games Design. I. Título.

Luana Rodrigues da Silva e Lima

APLICAÇÃO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL NA ANIMAÇÃO DE COMPONENTES EM UM JOGO 2D VOLTADO À SAÚDE MENTAL

Trabalho de Conclusão de Curso do Curso de
Bacharelado em Ciência da Computação da Uni-
versidade Estadual Paulista "Júlio de Mesquita
Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Profa. Dra. Juliana da Costa Feitosa

Orientador

Departamento de Computação

Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

**Profa. Dra. Simone das Graças
Domingues Prado**

Departamento de Computação

Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

**Prof. Dr. Kelton Augusto Pontara da
Costa**

Departamento de Computação

Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

Bauru, 13 de Novembro de 2025.

Dedico esse trabalho ao Lupi, meu fiel companheiro.

Agradecimentos

Agradeço à minha mãe, meu pai e minha irmã por me apoiarem, me consolarem e me alegrarem. Obrigada por serem o meu lar e meus anjos. Foi uma jornada extremamente difícil e de muitas descobertas e desafios, e eu não teria conseguido chegar até aqui se não fosse pela paciência, atenção, presença e compreensão de vocês. Obrigada mãe, por estar sempre ao meu lado, por ser meu refúgio, minha melhor amiga, alguém com quem eu sempre posso contar para me dar uma mão. Mesmo quando tento esconder, você sempre percebe quando estou mal e tenta me apoiar, não me deixando esquecer o quanto eu sou importante para alguém e me fazendo me sentir vista. Obrigada pai, por não desistir de mim nos momentos difíceis e por confiar no meu potencial. Mesmo sem entender certas coisas, você sempre faz o melhor de si para me ajudar, e por isso eu sou extremamente grata. Independente do que fosse, você acreditou na minha capacidade, mesmo quando eu mesma não acreditasse. Embora eu sentisse vergonha, ouvir você falar de mim com orgulho me deixava feliz. Obrigada Manu, por ser minha confidente e minha companheira. Em todos os momentos, sei que posso contar contigo, seja para alguma brincadeira, ou até para um consolo, e você sabe que também pode contar comigo. Eu sou grata por você ser minha irmã, por você me aceitar e me entender, por ser a luz da minha vida. Eu tenho muito orgulho de você e da pessoa maravilhosa que você é, a minha maior inspiração.

Agradeço ao Lupi, meu cachorro, por me fazer companhia em todos os momentos em que precisei, mesmo que você me fazia parar com o TCC para me fazer ver o que você queria. Você é o meu suporte e a minha felicidade, só de te ver o meu dia fica um pouquinho melhor. Agradeço aos meus avós e a toda minha família por me darem apoio e me aceitarem da maneira que eu sou, especialmente a minha vovó Naida, que sempre esteve ao meu lado me apoiando em todos os momentos. Eu não sou a melhor em expressar e demonstrar isso, mas eu sou extremamente grata pela presença e companhia de todos vocês, principalmente quando eu não tinha nenhum amigo. Mesmo quando todas as pessoas pareciam assustadoras, eu sempre consegui me sentir segura ao lado de vocês, livre para ser eu mesma. Eu escreveria o nome de todos vocês mas não quero arriscar esquecer ninguém e são muitas pessoas para uma página só. Em especial, agradeço a vovó Alice e ao vovô Tatu por terem feito parte de minha vida e terem criado a mãe incrível que eu tenho hoje. Espero que vocês estejam felizes e descansando no céu, junto com o Max e o Júnior. Agradeço à Juliana, minha psicóloga, e ao doutor Rafael, por me guiarem durante toda essa etapa e me ajudar a entender melhor meu potencial e minhas dificuldades. Vocês me ajudaram a confiar um pouco mais em mim mesma e a aceitar que eu não sou tão horrível quanto eu pensava. Depois de anos passando e me sentindo mal por apenas estar fora de casa, nunca imaginei que conseguiria um dia viver sem essa sensação, e agora isso é uma realidade na maior parte do tempo.

Agradeço à professora Juliana, por sua orientação, ajuda, apoio e receptividade. Obrigada por ser minha orientadora no TCC e co-orientadora na IC. Suas aulas foram incríveis e ver como você estava mais conectada à área de jogos me incentivou a não desistir desse campo. Espero não ter causado muitos problemas pela minha vergonha e pelo texto grande que eu fiz. Agradeço professora Tati por seu jeito e suas brincadeiras. Mesmo que eu não iniciasse uma interação e não mostrasse uma reação, eu amava suas aulas e achava as brincadeiras muito divertidas. Eu sempre me identifiquei muito com você. Agradeço à professora Andréa por seu carinho e apoio. Ao professor Wilson, à professora Roberta e ao professor Higor por me apresentarem e me manterem no mundo da maratona competitiva e da maratona feminina de programação. Ao professor André por sua inclusividade e por permitir que quem tivesse dificuldade não apresentasse. À professora Simone por sua orientação e suas aulas. Ao professor Kelton por ter aceitado a participar da banca. Ao professor Sementille e ao professor Papa por suas aulas. Aos outros professores por me ensinarem e me darem apoio. Ao Kaio, Alex, Hermínio, Garcia e Beto por me acolherem com as seções de RPG. Ao Christian por ter falado comigo e me puxado para o grupo. Ao Luciano pelo seu apoio e ajuda no primeiro ano. Ao Franco pelo seu auxílio e compreensão no primeiro e terceiro ano. À Sofia por me explicar o que eu não sabia sem zombar de mim. Ao Rufino por ter me acolhido e pela sua ajuda. Ao Henrique por não ter me excluído e pela sua compreensão. Ao Leo por me fazer sentir menos errada, por conversar e brincar comigo, por ser atencioso e não me julgar por ser diferente. Ao Fábio por ser o Fábio. Ao Yoshio e o Eduardo por desde a época do CTI me acolherem. Ao Vinícius e Mayara por serem legais e simpáticos comigo. À Rafaela, Dante, Joel, Daniel, Gustavo e Paulo por terem sido meus amigos. À Pati por ser compreensiva, gentil e simpática. A todos em que algum momento foram meus amigos.

Resumo

A saúde mental é extremamente importante para todos, porém frequentemente recebe respostas inadequadas e é representada por meio de estigmas em mídias como jogos digitais. Paralelamente, a aplicação de Inteligência Artificial (IA), especificamente na animação 2D, permanece pouco explorado. O presente trabalho visa abordar esses dois temas a partir do desenvolvimento de um protótipo de jogo sério 2D que explora temas relacionados a saúde mental de forma responsável, e da análise comparativa de ferramentas de IA para auxiliar no processo de criação de animações 2D pixel art para o jogo. A análise revelou que nenhuma ferramenta individual gerou animações perfeitas sem edição manual. Ferramentas generalistas avançadas (Gemini Pro, Vidu) mostraram alta consistência, mas também apresentaram dificuldades com especificidades do 2D e da pixel art, enquanto ferramentas especializadas (Pixel Lab, Pixie.haus) aderiram melhor ao estilo, porém com menor desenvolvimento geral ou flexibilidade. Os resultados mostraram que a IA funciona melhor como ferramenta auxiliar para o animador, sendo crucial a integração de editores robustos para ajustes da animação criada.

Palavras-chave: Inteligência Artificial; Animação 2D; Jogos Sérios; desenvolvimento de jogos.

Abstract

Mental health is extremely important for everyone, yet it often receives inadequate responses and is represented through stigma in media such as digital games. At the same time, the application of Artificial Intelligence (AI) specifically in 2D animation remains underexplored. The present work aims to address these two themes through the development of a 2D serious game prototype that explores themes related to mental health in a responsible manner, and through the comparative analysis of AI tools to assist in the process of creating 2D pixel art animations for the game. The analysis revealed that no single tool generated perfect animations without manual editing. Advanced generalist tools (e.g., Gemini Pro, Vidu) showed high consistency, but also presented difficulties with the specificities of 2D and pixel art, while specialized tools (e.g., Pixel Lab, Pixie.haus) adhered better to the style, however with less general development or flexibility. The work showed that AI works best as an auxiliary tool for the animator, with the integration of robust editors crucial for adjustments to the created animation.

Keywords: Artificial Intelligence; 2D Animation; Serious Games; game development.

Lista de figuras

| | |
|---|----|
| Figura 1 – Vaso de planta desenhado no estilo <i>pixel art</i> | 22 |
| Figura 2 – Exemplos de prompts usados | 25 |
| Figura 3 – Elementos do jogo que foram animados | 25 |
| Figura 4 – Tela do jogo em execução | 29 |
| Figura 5 – Animação de caminhada gerada pelo Rosebud AI | 31 |
| Figura 6 – <i>Sprite sheet</i> gerado pelo SpriteSheetGPT | 32 |
| Figura 7 – Melhor <i>sprite</i> em <i>side view</i> gerado pelo CGDream | 33 |
| Figura 8 – Animação da porta B abrindo em <i>side view</i> gerada pelo Pixie.Haus | 34 |
| Figura 9 – Quadro da animação de caminhada gerada pelo Animated Drawings | 35 |
| Figura 10 – Animação de caminhada gerada no God Mode AI | 36 |
| Figura 11 – Melhor <i>sprite</i> em <i>side view</i> gerado pelo ChatGPT | 38 |
| Figura 12 – Quadro da animação de caminhada gerado pelo Pixel Lab | 38 |
| Figura 13 – Resultados finais editados no Pixel Lab | 39 |
| Figura 14 – Melhor <i>sprite</i> sentado gerado no OpenArt.AI | 40 |
| Figura 15 – Quadro da animação da porta B em <i>side view</i> gerada no Vidu | 41 |
| Figura 16 – Melhores <i>sprites</i> gerados no Gemini Pro | 42 |
| Figura 17 – Animações geradas no Gemini Pro | 42 |
| Figura 18 – Quadros do vídeo da animação 1 de caminhada implementada no jogo | 47 |
| Figura 19 – Quadros do vídeo da animação de pulo implementada no jogo | 48 |
| Figura 20 – Quadros do vídeo da segunda animação de caminhada implementada no jogo | 48 |
| Figura 21 – Quadros do vídeo da animação 3 de caminhada implementada no jogo | 49 |
| Figura 22 – Quadros do vídeo da animação da porta A sem edição implementada no jogo | 49 |
| Figura 23 – Quadros do vídeo da animação da porta A editada implementada no jogo | 50 |
| Figura 24 – Quadros do vídeo das animações da porta B implementada no jogo | 50 |
| Figura 25 – Menu de pausa durante a execução do jogo | 61 |
| Figura 26 – Neblina permanente na borda do cômodo | 63 |
| Figura 27 – Animação gerada pelo Rosebud AI | 67 |
| Figura 28 – Resultados finais | 67 |
| Figura 29 – <i>Sprite sheet</i> gerado pelo SpriteSheetGPT | 68 |
| Figura 30 – Melhores resultados do CGDream utilizando a funcionalidade de imagem | 70 |
| Figura 31 – Comparativo de resultados com e sem Estilo de referência | 71 |
| Figura 32 – Comparativo de resultados do modelo Flux com e sem filtro pixelizado, e do modelo Juggernaut XL com e sem o uso de 3D como palavra negativa | 71 |
| Figura 33 – <i>Frames</i> das animações geradas pelo Pixie.Haus | 73 |
| Figura 34 – Inclusão incorreta da sombra | 73 |
| Figura 35 – <i>Sprites</i> usados para o teste no Animated Drawings | 74 |

| | |
|---|-----|
| Figura 36 – Comparando <i>frames</i> correspondentes | 75 |
| Figura 37 – Braços em formato diferente um do outro | 76 |
| Figura 38 – <i>Frames</i> das animações geradas pelo Animated Drawings 3 | 77 |
| Figura 39 – Comparação do <i>sprite</i> original com quadros das animações geradas no God Mode AI | 78 |
| Figura 40 – Resultados gerados pelo God Mode AI | 79 |
| Figura 41 – <i>Sprite sheet</i> convertido para <i>pixel art</i> | 79 |
| Figura 42 – Artefatos usados como referência para geração de imagens no ChatGPT | 80 |
| Figura 43 – <i>Sprites</i> em <i>side view</i> gerados pelo ChatGPT | 81 |
| Figura 44 – <i>Sprites sheets</i> gerados pelo ChatGPT | 82 |
| Figura 45 – <i>Sprite sheet</i> gerado pelo Gemini Pro | 83 |
| Figura 46 – Artefatos editados no Pixel Lab | 84 |
| Figura 47 – Comparação entre o <i>sprite</i> original e os resultados gerados no Pixel Lab | 85 |
| Figura 48 – Edição no resultado da rotação no Pixel Lab | 85 |
| Figura 49 – <i>Sprite</i> do personagem em <i>side view</i> após ajustes finos no Pixel Lab | 87 |
| Figura 50 – Comparação do <i>sprite</i> original com os <i>frames</i> da animação de base e das geradas no Pixel Lab | 88 |
| Figura 51 – Comparação do <i>sprite</i> de referência com os <i>frames</i> da animação de base e das geradas no Pixel Lab | 89 |
| Figura 52 – Comparação do <i>sprite</i> original com os <i>frames</i> da animação gerada no Pixel Lab | 90 |
| Figura 53 – Processo de edição no Pixel Lab do <i>sprite</i> em <i>side view</i> gerado pelo Gemini Pro | 91 |
| Figura 54 – Processo de edição no Pixel Lab do <i>sprite</i> em <i>back view</i> gerado pelo Gemini Pro | 91 |
| Figura 55 – Processo de edição no Pixel Lab do <i>sprite sheet</i> da porta gerado pelo Vidu | 92 |
| Figura 56 – Processo de edição no Pixel Lab do <i>sprite sheet</i> da porta gerado pelo Gemini Pro | 92 |
| Figura 57 – <i>Frames</i> do vídeo gerado no OpenArt.AI | 93 |
| Figura 58 – Imagem geradas no OpenArt.AI | 94 |
| Figura 59 – Imagens geradas pelo modelo Gemini | 95 |
| Figura 60 – Artefatos usados para referência no vidu | 96 |
| Figura 61 – Comparação do resultado 3D e 2D gerado pelo Vidu | 96 |
| Figura 62 – Quadros dos vídeos em cada um dos níveis de deformação | 98 |
| Figura 63 – <i>Frames</i> dos vídeos gerados no Vidu | 99 |
| Figura 64 – Inconsistências dos vídeos gerados no Vidu | 101 |
| Figura 65 – Inconsistências dos vídeos da porta abrindo gerados no Vidu | 102 |
| Figura 66 – Comparação da geração dos objetos em relação ao quarto no Vidu | 102 |
| Figura 67 – Quadros dos vídeos das portas gerados no Vidu | 104 |
| Figura 68 – Quadro do personagem inclinando para frente na animação gerada no Vidu | 104 |
| Figura 69 – <i>Sprite sheet</i> em <i>pixel</i> sem fundo da animação da porta em <i>side view</i> abrindo | 105 |
| Figura 70 – <i>Sprite 2</i> do personagem Pablo rotacionado 45 graus gerado e editado no Pixel Lab | 106 |
| Figura 71 – <i>Sprites</i> em <i>side view</i> gerados no Gemini Pro | 107 |

| | |
|--|-----|
| Figura 72 – Mais <i>sprites</i> em <i>side view</i> gerados no Gemini Pro | 108 |
| Figura 73 – Processo de edição do melhor <i>sprite</i> em <i>side view</i> no Pixilart | 109 |
| Figura 74 – Processo de edição do melhor <i>sprite</i> em <i>back view</i> no Pixilart | 110 |
| Figura 75 – Quadro da animação de caminhada gerada no Gemini Pro | 113 |
| Figura 76 – Quadros das animações de caminhada geradas no Gemini Pro | 114 |
| Figura 77 – <i>Sprite sheet</i> do ciclo de caminhada sem fundo | 115 |
| Figura 78 – Melhor resultado da animação de pulo gerada no Gemini Pro | 117 |
| Figura 79 – Resultado da animação de abrir porta gerada no Gemini Pro | 118 |
| Figura 80 – Imagem após remoção do fundo | 118 |
| Figura 81 – Melhor resultado da porta A abrindo | 119 |
| Figura 82 – Esquema mostrando animação da porta abrindo em diferentes ângulos | 120 |
| Figura 83 – Melhor resultado da porta B abrindo no Gemini Pro | 120 |

Lista de quadros

| | |
|---|-----|
| Quadro 1 – Testes realizados no RosebudAI | 31 |
| Quadro 2 – Testes realizados no SpriteSheetGPT | 32 |
| Quadro 3 – Testes realizados no CGDream | 33 |
| Quadro 4 – Testes realizados na Pixie.haus | 34 |
| Quadro 5 – Testes realizados no Animated Drawings | 35 |
| Quadro 6 – Testes realizados no God Mode AI | 36 |
| Quadro 7 – Testes realizados no ChatGPT | 37 |
| Quadro 8 – Testes realizados no Pixel Lab | 38 |
| Quadro 9 – Testes realizados no OpenArt.AI | 40 |
| Quadro 10 – Testes realizados no Vidu | 41 |
| Quadro 11 – Testes realizados na Gemini Pro | 43 |
| Quadro 12 – Tabela comparativa das ferramentas de IA utilizadas (Parte 1 de 2) | 43 |
| Quadro 13 – Tabela comparativa das ferramentas de IA utilizadas (Parte 2 de 2) | 44 |
| Quadro 14 – Resumo dos experimentos de geração do <i>sprite sheet</i> no Gemini Pro | 112 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Análise quantitativa de <i>frames</i> com deformação nos vídeos gerados pelo Vidu | 98 |
| Tabela 2 – Análise quantitativa de <i>frames</i> com deformação, comparando testes com referência em <i>front</i> e <i>side view</i> | 99 |

Sumário

| | | |
|------|--|----|
| 1 | INTRODUÇÃO | 17 |
| 2 | REVISÃO BIBLIOGRÁFICA | 19 |
| 2.1 | Jogos Sérios 2D | 19 |
| 2.2 | Inteligência Artificial Aplicada à Animação | 20 |
| 2.3 | Trabalhos relacionados | 20 |
| 3 | METODOLOGIA | 22 |
| 3.1 | Metodologia de desenvolvimento do jogo | 22 |
| 3.2 | Metodologia de Análise das Ferramentas de IA | 23 |
| 4 | DESENVOLVIMENTO DO JOGO | 27 |
| 4.1 | Narrativa e Temas Abordados | 27 |
| 4.2 | Mecânicas de Jogo | 27 |
| 4.3 | Ambiente do Jogo | 28 |
| 4.4 | Arquitetura e Implementação | 29 |
| 5 | ANÁLISE COMPARATIVA DE FERRAMENTAS DE IA PARA ANIMAÇÃO 2D | 30 |
| 5.1 | Visão geral da análise comparativa | 30 |
| 5.2 | Análise do RosebudAI | 30 |
| 5.3 | Análise do SpriteSheetGPT | 32 |
| 5.4 | Análise do CGDream | 32 |
| 5.5 | Análise do Pixie.haus | 33 |
| 5.6 | Análise do Animated Drawings | 34 |
| 5.7 | Análise do God Mode AI | 35 |
| 5.8 | Análise do ChatGPT | 37 |
| 5.9 | Análise do Pixel Lab | 38 |
| 5.10 | Análise do OpenArt.AI | 39 |
| 5.11 | Análise do Vidu | 40 |
| 5.12 | Análise da Ferramenta Gemini Pro | 41 |
| 5.13 | Quadro comparativo e discussão geral | 43 |
| 5.14 | Desenvolvimento de um fluxo de trabalho para animação com IA | 46 |
| 6 | CONCLUSÃO | 51 |
| 6.1 | Conclusão | 51 |
| 6.2 | Trabalhos Futuros | 52 |

| | |
|-------------|----|
| REFERÊNCIAS | 53 |
|-------------|----|

APÊNDICES 57

APÊNDICE A – IMPLEMENTAÇÃO E LÓGICA DE PROGRAMAÇÃO DAS MECÂNICAS 58

| | | |
|------|--|----|
| A.1 | Sistema de Gerenciamento e Configurações do Jogo | 58 |
| A.2 | Sistema de Controle do Personagem e Animação | 58 |
| A.3 | Sistema da Câmera | 59 |
| A.4 | Sistema de Menu | 61 |
| A.5 | Sistema de Interação com Objetos | 61 |
| A.6 | Sistema de Portas e entradas | 62 |
| A.7 | Sistema de neblina | 63 |
| A.8 | Sistema de saúde mental | 64 |
| A.9 | Sistema de mudança de cena | 64 |
| A.10 | Sistema de plataforma | 65 |

APÊNDICE B – ANÁLISE DETALHADA DAS FERRAMENTAS DE IA 66

| | | |
|--------|--|-----|
| B.1 | Análise do RosebudAI | 66 |
| B.2 | Análise do SpriteSheetGPT | 67 |
| B.3 | Análise do CGDream | 68 |
| B.4 | Análise do Pixie.haus | 72 |
| B.5 | Análise do Animated Drawnings | 74 |
| B.6 | Análise do God Mode AI | 77 |
| B.7 | Análise do ChatGPT | 80 |
| B.7.1 | Geração do <i>sprite</i> em <i>side view</i> | 80 |
| B.7.2 | Geração do <i>sprite sheet</i> do personagem andando | 82 |
| B.8 | Análise do Pixel Lab | 82 |
| B.8.1 | Ferramenta de rotação | 84 |
| B.8.2 | Ferramenta de animação para animação | 86 |
| B.8.3 | Uso no pós-processamento | 90 |
| B.9 | Análise do OpenArt.AI | 92 |
| B.10 | Análise do Vidu | 95 |
| B.10.1 | Funcionalidade de referência para vídeo | 96 |
| B.10.2 | Funcionalidade de imagem para vídeo | 103 |
| B.11 | Análise da Ferramenta Gemini Pro | 106 |
| B.11.1 | Geração do <i>sprite</i> em <i>side view</i> | 106 |
| B.11.2 | Geração do <i>sprite</i> em <i>back view</i> | 109 |

| | | |
|--------|--|-----|
| B.11.3 | Geração do <i>sprite sheet</i> do ciclo de caminhada | 111 |
| B.11.4 | Geração da animação de caminhada | 113 |
| B.11.5 | Geração da animação de pulo | 116 |
| B.11.6 | Geração da animação do personagem abrindo a porta | 117 |
| B.11.7 | Geração da animação das portas abrindo | 118 |

1 Introdução

A saúde emocional é um componente crítico do bem-estar geral e ela se manifesta de forma variável em cada pessoa, de maneira muito parecida com a saúde física (MENEFEE; LEDOUX; JOHNSTON, 2022). Ao longo da vida, diversos determinantes contribuem para proteger, prejudicar ou deixar mais vulnerável a saúde mental do ser humano, como fatores psicológicos e biológicos individuais, circunstâncias sociais e ambientes desfavoráveis (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2022b). De acordo com a Organização Mundial da Saúde (OMS), o bem-estar psicológico é extremamente importante para todos. Globalmente, as necessidades de saúde mental são altas, mas as respostas são insuficientes e inadequadas (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2022c).

Esse tema ganhou importância e visibilidade recentemente com a divulgação da Agenda de Desenvolvimento Sustentável 2030 da Organização das Nações Unidas (ONU), pois pela primeira vez as metas incluíram saúde mental de maneira explícita. No Objetivo de Desenvolvimento Sustentável (ODS) 3, a meta 3.4 estabelece a necessidade de "promover a saúde mental e o bem-estar". Esse tema possui grandes impactos na qualidade de vida humana, pois pesquisas mostraram que condições de saúde mental são responsáveis por 13% dos anos vividos com incapacidade e perdidos por morte prematura (HEYMANN; SPRAGUE, 2023). Em 2019, uma a cada oito pessoas viveram com algum transtorno mental, um número que aumentou consideravelmente por causa da pandemia do COVID-19 (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2022a).

Transtornos mentais, em muitos casos, não são facilmente reconhecíveis baseados na aparência e comportamento externo de uma pessoa. Assim, pessoas afetadas por essas condições sofrem com a falta de empatia e de aceitação, sendo alvos de discriminação e psicofobia (preconceito direcionado a pessoas com transtornos mentais) (KASDORF, 2023). O estigma é encontrado no ambiente de trabalho, familiar e escolar, fazendo com que o indivíduo se retraia, seja desencorajado a discutir sobre sua saúde mental, desenvolva mecanismos de enfrentamento prejudiciais e tenha um agravamento dos sintomas. Além disso, existe também a auto estigmatização, que é a internalização de crenças negativas sobre si mesmo, bem como auto discriminação (ROMA, 2024).

A mídia em geral é uma fonte extremamente importante de conhecimento sobre transtornos. Essa representação de assuntos relacionados à saúde mental afeta as atitudes e crenças do público. Com os jogos sendo um dos tipos mais impactantes de mídia, nesse tipo de ambiente é muito comum a representação de transtornos mentais, porém de maneira discriminatória e estigmatizada (KASDORF, 2023). Apesar desse histórico, jogos possuem recursos que permitem diversas maneiras de abordar temas relacionados a saúde mental, como

por exemplo, o uso de mecânicas para criar uma experiência mais imersiva (DUNLAP KELLI ; KOWERT, 2021).

Na indústria de jogos, a IA é utilizada para gerar comportamentos responsivos, adaptativos e inteligentes para personagens não jogáveis (em inglês, NPCs) (SCHIJVEN; KIKKAWA, 2022). De acordo com Jorapur (2024), ela também é implementada para adaptar a história dependendo do comportamento do jogador, para a criação de diferentes conteúdos do jogo como níveis e mundos infinitos, bem como para ajustar a dificuldade de acordo com a performance do player. Além disso, é usada também para fazer animações de personagem (XIAN; SAHAGUN, 2023), capaz de gerar sequências de movimentos 3D contínuos e realistas a partir de uma pose inicial e de uma ação alvo (HUANG, 2024). Embora tenha feito contribuições significativas para os jogos digitais, sua aplicação em animações 2D é pouco explorada. As ferramentas de IA para animação atualmente têm como alvo especificamente vídeos tridimensionais, que não têm um controle efetivo sobre a aleatoriedade (QIU, 2023).

Diante deste cenário, o objetivo principal deste trabalho é investigar a viabilidade e a eficácia das ferramentas de IA disponíveis atualmente para a produção de vídeos 2D. Para essa análise, foi desenvolvido um protótipo de jogo. Um jogo 2D é fundamentalmente composto por animações (de personagens, objetos e interfaces) e, por isso, serve como o ambiente de teste ideal. A necessidade de criar animações funcionais para este projeto permitiu a realização de uma análise comparativa rigorosa, avaliando não apenas a qualidade visual dos resultados gerados, mas também os desafios práticos de sua implementação em um jogo.

Dessa forma, o presente trabalho visa apresentar um jogo sério 2D que utiliza IA para fazer as animações dos personagens, e que aborda temas de saúde mental para trazer uma visão mais empática e conscientizar sobre esse assunto.

Este trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica, abordando os conceitos de desenvolvimento de jogos 2D e como a IA pode ajudar na área da animação; o Capítulo 3 detalha a metodologia utilizada, tanto para o desenvolvimento do jogo quanto para a análise comparativa das ferramentas de IA; o Capítulo 4 apresenta o jogo "O Monstro", descrevendo sua narrativa, mecânicas e implementação; o Capítulo 5 contém a análise detalhada das ferramentas de IA; e o Capítulo 6 apresenta a conclusão do trabalho.

2 Revisão bibliográfica

2.1 Jogos Sérios 2D

O desenvolvimento de jogos 2D envolve a criação do ambiente, dos personagens, das mecânicas e das animações em um espaço bidimensional, cujos componentes são representados usando dois eixos: X e Y (PRAKASH, 2023). De acordo com PRAKASH (2023), apesar de estarmos numa era dominada por gráficos 3D, os jogos 2D mantêm sua popularidade. Isso ocorre pois são mais baratos de serem produzidos, sendo acessíveis para desenvolvedores independentes e equipes pequenas (HALPERN, 2019).

Relacionados a jogos 2D existem diversos termos importantes. *Sprites* são figuras 2D incorporadas em uma cena maior, *tiles* são imagens pequenas que compõem um *tilemap*, enquanto o *tilemap* é composto por *tiles* para formar um cenário (MARCOLINO et al., 2025). O *sprite sheet* é uma imagem com diversos *sprites* separados em grade (TURNER, 2023), que podem ser usados para formar uma animação de forma parecida a um *flipbook*¹ (folioscópio em inglês) (UNITY TECHNOLOGIES, 2019).

Jogos Sérios são jogos cujo propósito primário não é o entretenimento, focando na exploração ativa de problemas sociais (ABT, 1987). Eles são utilizados para diversas finalidades: auxiliar no processo educacional, ajudar pacientes a entender sua condição atual e sua reabilitação, promover a conscientização do público para problemas psicológicos e emocionais, etc (GALVÃO et al., 2024). Um estudo sobre o uso de Jogos Sérios em assuntos de saúde mental aponta que eles podem reduzir sintomas de indivíduos com transtornos como ansiedade, depressão, entre outros (DEWHIRST; LAUGHARNE; SHANKAR, 2022).

Além disso, uma categoria recente de jogos vem ganhando destaque: jogos empáticos. Esses jogos priorizam, por meio de mecânicas, trazer a experiência de como é estar no lugar de outra pessoa. O aspecto interativo que os jogos trazem permite que o jogador participe ativamente do conteúdo mostrado, não sendo apenas observadores passivos, mas sim participantes afetados pelos eventos do ambiente. Dessa forma, é possível fazer com que o usuário tenha interesse em tópicos relacionados à saúde mental (como o luto e transtornos mentais) e ter uma visão mais empática sobre esse tema (GALVÃO et al., 2024).

¹ Livros que formam a ilusão de movimento, após serem rapidamente folheados, por meio de imagens sequenciais que mudam gradualmente

2.2 Inteligência Artificial Aplicada à Animação

A IA é um campo rico e diverso, com aplicabilidade em várias vertentes como automóveis, saúde, entretenimento, educação, segurança, entre outras. Ela foca em aprender com experiências, bem como alterar seu processamento e comportamento baseado em seu aprendizado. Além disso, é considerada a próxima revolução industrial na área de entretenimento, e também é capaz de aumentar a eficiência automatizando numerosas tarefas repetitivas (MEENA; JINGAR; GUPTA, 2020).

A aplicação das técnicas de animação 2D existentes pode ser muito demorada e complexa, dependendo da criação manual de quadros-chaves ou da simulação física. Para evitar o trabalho intensivo e manter o foco na inovação criativa, a animação artística está sendo lentamente transformada pelo uso da IA (LI, 2024b). Porém, esse uso foi muito pouco explorado, fazendo com que possivelmente essa tecnologia ainda tenha um grande potencial para revolucionar o processo complexo de produção de animação 2D (QIU, 2023). Automatizando o processo, o tempo e o esforço são reduzidos, sendo possível produzir animações de alta qualidade mais rapidamente e com um custo menor (XIAN; SAHAGUN, 2023).

2.3 Trabalhos relacionados

No campo específico de jogos que abordam temas de saúde mental, o Jogo Sérioso desenvolvido por Verhalen e Rodrigues (2024) e disponibilizado na plataforma RUFUS² teve como objetivo auxiliar crianças no diálogo sobre morte e luto. Falar sobre a morte ainda é um tabu que pode afetar o desenvolvimento emocional infantil, assim como outros assuntos relacionados à saúde mental. O estudo identificou que o diálogo com crianças sobre o tema costuma ser feito por meio de narrativas. Com base nisso, projetou-se um jogo com uma história definida pelo terapeuta para funcionar como um recurso terapêutico. Essa estratégia permite que o especialista guie a experiência da criança e analise as escolhas feitas por ela durante o jogo. As avaliações conduzidas com profissionais apontaram para a confirmação da hipótese de que o jogo poderia ser um recurso de auxílio eficaz no tratamento.

Um estudo diretamente relacionado ao uso de IA em animação 2D foi realizado por Li (2024b), cuja junção da IA com Modelagem Biomecânica reduziu drasticamente a necessidade de intervenção manual e otimizou o fluxo de trabalho dos animadores. Para a realização da animação 2D, um boneco virtual foi modelado biomecanicamente com um sistema complexo de músculos que imitam o corpo humano. O treinamento neuromuscular foi realizado por uma Rede Neural. Assim, o estudo conclui que a técnica funciona como uma ferramenta de auxílio direto aos artistas, permitindo especialmente que estes capturem a personalidade de um ator, ajudando-os na seleção de movimentos típicos das mãos e dos braços.

² <<https://rufus.icmc.usp.br/login>>. Acesso em: 16 ago. 2025.

Em uma abordagem focada na junção dos temas de animação, jogo 2D e IA, [Qiu \(2023\)](#) apresenta um estudo detalhado sobre a criação e animação de um personagem 2D, para um jogo de plataforma utilizando exclusivamente processos baseado em IA Generativa, e sem nenhuma intervenção manual. Para isso, múltiplas ferramentas são combinadas, iniciando com o ChatGPT para a conceituação, passando pelo Midjourney para o design visual, treinando modelos para gerar imagens mais avançadas das mãos, e utilizando a animação esquelética do Unity. Foi concluído no estudo que a IA demonstra capacidade de substituir métodos tradicionais de animação, apesar do fluxo apresentado possuir certas barreiras, como a necessidade de alto nível de conhecimento técnico especializado.

3 Metodologia

Inicialmente, foi elaborado o Documento de Design de Jogo (GDD, *Game Design Document* em inglês) e a arquitetura de software, que contribuem para a organização da estrutura, roteiro, cenários e personagens do jogo. Isso foi feito de forma paralela com um levantamento bibliográfico mais aprofundado sobre saúde mental, traumas, problemas emocionais, distúrbios mentais e mecanismos de enfrentamento saudáveis.

Depois que os documentos foram elaborados, iniciou-se o desenvolvimento do jogo, desenhando o ambiente de cada cena e programando cada mecânica. Quando o levantamento bibliográfico sobre saúde mental foi concluído, iniciou-se uma pesquisa sobre ferramentas de IA para animação 2D, analisando as técnicas utilizadas por cada uma, as vantagens, as desvantagens e o foco (e.g., realismo, *pixel art*, sem foco). Após a pesquisa, foram escolhidas animações que foram implementadas de forma paralela com o desenvolvimento do jogo. Durante o estudo das ferramentas de IA, foi feita uma análise de como cada ferramenta auxilia no cenário de animações 2D. Ao longo do desenvolvimento, diversos testes foram aplicados para verificar se o comportamento estava de acordo com o esperado.

3.1 Metodologia de desenvolvimento do jogo

O desenvolvimento do jogo "O Monstro" foi realizado de forma paralela à análise das ferramentas de IA para auxílio no processo de animação 2D. No entanto, o fluxo de trabalho para a criação dos artefatos do jogo em si seguiu um processo mais sequencial, no qual uma sub-tarefa era concluída antes de se iniciar a próxima.

A fase inicial consistiu na elaboração de um GDD básico para definir a premissa narrativa, os personagens principais e as mecânicas centrais do jogo. Uma etapa crucial e priorizada foi a criação dos principais *sprites* do jogo e a modelagem dos cenários no Unity. A produção antecipada destes *sprites* foi um pré-requisito para viabilizar a criação das animações a serem analisadas. A arte principal do jogo foi no estilo *pixel art* por ser um estilo simples, rápido e nostálgico. A figura 1 apresenta um vaso de planta nesse estilo.

Figura 1 – Vaso de planta desenhado no estilo *pixel art*



Fonte: Elaborada pela autora.

O projeto foi desenvolvido no Unity¹, usando a linguagem de programação C#. Os *sprites* e *tiles* foram desenhados na ferramenta Pixilart² (para os elementos pixelizados) e no aplicativo ibisPaint X³.

A produção do jogo foi priorizada na implementação progressiva das mecânicas essenciais para criar um protótipo funcional. Os sistemas desenvolvidos nesta fase incluem:

- Programação da mecânica de movimentação e de pulo em 2D, com a integração das animações geradas e dos estados de animação do personagem;
- Implementação de uma câmera que segue o personagem com limites de borda, garantindo que a visão do jogador não ultrapasse os limites do cenário;
- Implementação do menu de pausa com as funcionalidades de continuar e sair do jogo;
- Criação do sistema de interação entre o personagem e os objetos da cena, com o menu do objeto aparecendo e mostrando as opções de interação quando este é selecionado;
- Implementação da mecânica de abrir porta, com a integração das animações geradas e a detecção automática do personagem;
- Criação de um sistema de *fog* (neblina, em inglês) que esconde ou mostra um cômodo, dependendo da presença do personagem e do estado da porta;
- Programação da barra de saúde mental e da lógica associada ao cair em áreas de perigo, mudando o tamanho da barra na interface; e
- Desenvolvimento do sistema de transição entre diferentes cômodos e cenas, salvando os dados do jogador e do ambiente durante a mudança de cena.

3.2 Metodologia de Análise das Ferramentas de IA

Para conduzir a análise comparativa das ferramentas de IA, a metodologia foi estruturada em etapas, partindo de uma seleção ampla de ferramentas até uma avaliação aprofundada das mais promissoras. Inicialmente, foram estabelecidos os seguintes critérios de seleção para a escolha de ferramentas:

- Capacidade de criar vídeos ou imagens que pudessem ser usados para a animação 2D;
- Disponibilidade de um modelo de acesso gratuito, ainda que com limitações de uso;

¹ <<https://unity.com/pt>>. Acesso em: 22 ago. 2025.

² <<https://www.pixilart.com/>>. Acesso em 25 ago. 2025.

³ <<https://ibispaint.com/?lang=pt>>. Acesso em 11 ago. 2025.

- Possibilidade de usar uma imagem pré-existente (do personagem ou objeto) como referência para consistência visual; e
- Acessível para um usuário sem conhecimento aprofundado na ferramenta.

Com base nesses critérios, foram selecionados os seguintes softwares como candidatos para a produção de animação 2D: CGDream ([CGDREAM AI IMAGE GENERATOR, 2025](#)), ChatGPT ([OPENAI, 2025d](#)), OpenArtAI([OPENART.AI, 2025](#)), geminiPro ([GOOGLE, 2025](#)), God Mode AI ([LI, 2024a](#)), PixelLab ([PIXELLAB.AI, 2025](#)), PixieHaus ([URSATJEW, 2025](#)), Rosebud AI ([ROSEBUD AI, 2025](#)), Animated Drawings ([SMITH, 2024](#)), Vidu ([VIDU, 2024](#)), AI Sprite Sheet Maker ([SEGMIND, 2025](#)) e SpriteSheetGPT ([YESCHAT AI, 2024](#)).

O processo de análise foi dividido em duas fases. A primeira fase consistiu em uma análise geral de cada ferramenta, verificando os recursos gratuitos disponíveis, as opções de customização existentes e a capacidade de gerar uma animação 2D útil para o jogo em desenvolvimento, a partir de um *sprite* de referência. Essa triagem inicial permitiu descartar algumas ferramentas que provaram não ser capazes de alcançar o resultado desejado, sobrando apenas as candidatas mais promissoras. Um desafio descoberto nesta etapa foi a limitação no uso do modelo gratuito de muitas plataformas, o que restringiu o número de testes comparativos e reduziu o número de gerações.

Na segunda fase, foi realizado um aprofundamento das ferramentas restantes. Foram conduzidos testes iterativos com diversos prompts (a maioria em inglês para melhores resultados) e imagens de referências nas plataformas que não possuíam um limite para o uso gratuito, ou que o limite era muito alto. Para as plataformas mais restritas, os resultados que chegavam mais perto do desejado eram usados como referência para os outros softwares. As animações satisfatórias foram implementadas no jogo, com o uso de ferramentas auxiliares para converter ou ajustar o formato do arquivo e para pequenas edições na imagem. A figura 2 demonstra alguns dos prompts utilizados.

Foram criadas animações com IA para alguns elementos do jogo: o personagem Pablo (Figura 3a), que realiza as ações de andar, pular, virar de costas, abrir a porta, se deitar e sentar; a personagem Luz (Figura 3b), que realiza a ação de andar; e a porta (Figuras 3c), que executa os movimentos de abrir e fechar. Os resultados adequados foram implementados no jogo.

Ao final do processo, todas as ferramentas analisadas foram sintetizadas em um quadro comparativo geral com os seguintes critérios:

- Foco em 2D;
- Consistência e precisão geral;
- Adequação ao estilo de pixel art;

Figura 2 – Exemplos de prompts usados

This is a closed door. Make an animation of this door opening.
The handle is at the left side and the animation must be 2D

(a) Prompt 1

This is a closed door. Make an animation of this door opening.
The handle is at the right side and the animation must be 2D

(b) Prompt 2

This is a closed door. Make an animation of this door opening.
The handle is at the right side and the animation must be 2D.
The door opens to the left

(c) Prompt 3

Fonte: Elaborada pela autora.

Figura 3 – Elementos do jogo que foram animados



(a) *Sprite* do Pablo



(b) *Sprite* da Luz



(c) *Sprite* da porta A em vista frontal

Fonte: Elaborada pela autora.

- Nível de customização;
- Facilidade de edição e refinamento do material gerado;
- Eficiência; e
- Tipo de conteúdo gerado.

Em seguida, as animações mais satisfatórias foram implementadas e submetidas a uma análise para justificar a construção do fluxo de trabalho final, focando no processo de criação de cada uma delas e a influência do mesmo no resultado final.

4 Desenvolvimento do Jogo

Este capítulo apresenta o conceito da narrativa criada e a aplicação desenvolvida como parte prática deste trabalho: um jogo sério 2D de plataforma e aventura, intitulado como "O Monstro".

4.1 Narrativa e Temas Abordados

"O Monstro" é um jogo 2D de visão lateral que explora, através de metáforas e mecânicas de jogo, temas complexos de saúde mental, como culpa, trauma e o processo de luto. O jogador controla Pablo, um jovem que acorda em um mundo fragmentado e parte em uma jornada para entender o que está acontecendo, encontrando diversas pessoas que têm diferentes graus de ciência sobre o que está acontecendo naquele lugar. Durante essa aventura, Pablo recebe a missão de encontrar e matar um monstro que alguns outros jovens acreditam ser a causa dos eventos peculiares que ocorreram naquele lugar.

O tema do jogo, no entanto, é a desconstrução dessa caça a um culpado externo. O monstro funciona como uma metáfora para a culpa internalizada e o auto-desprezo do protagonista. A progressão narrativa não se dá pela derrota de uma criatura, mas pela redescoberta de memórias fragmentadas que ajudam Pablo a recontextualizar seus sentimentos.

De forma proposital, os eventos específicos que originaram o estado mental do protagonista são mantidos ambíguos e abertos à interpretação. A intenção não é focar em uma única causa, mas sim na luta pela auto aceitação e no sentimento de ser discriminado ou excluído. Ao manter a narrativa abstrata, busca-se permitir que um público maior possa se identificar com a jornada emocional do personagem, aplicando-a às suas próprias vivências.

4.2 Mecânicas de Jogo

A jogabilidade foi projetada para combinar elementos de exploração, plataforma e resolução de quebra-cabeças, com as interações servindo para reforçar os temas narrativos. As mecânicas planejadas no GDD são descritas a seguir:

- Exploração e Plataforma: Navegação pelos cenários através de um sistema de movimentação 2D *side-scrolling* (câmera com visão lateral acompanha o movimento do personagem), com trechos de plataforma que conectam as diferentes áreas do mundo.
- Sistema de Saúde Mental: Em vez de uma barra de vida tradicional, o jogo utiliza um medidor de saúde mental que é afetado por eventos no jogo, como quedas ou encontros

com inimigos.

- Inimigos como Gatilhos: Os inimigos do jogo não são adversários físicos, mas sim representações de gatilhos, traumas e crises de ansiedade.
- Mecanismos de Enfrentamento: O jogador não combate os inimigos de forma tradicional. Em vez disso, aprende e utiliza habilidades ensinadas pela psicóloga, que representam técnicas de enfrentamento, como exercícios de respiração, para fazer com que os inimigos desapareçam temporariamente.
- Sistema de Diálogo e Missões: A progressão da história é guiada por missões e diálogos com os NPCs que habitam o mundo.
- Sistema de missões secundárias: Missões opcionais não necessárias para a conclusão da história que Pablo se sente compelido a aceitar devido à sua natureza altruísta. A conclusão destas tarefas, no entanto, causaria uma pequena diminuição na barra de saúde mental, representando mecanicamente o desgaste emocional do autossacrifício excessivo.

4.3 Ambiente do Jogo

O jogo se passa em um mundo fragmentado, composto pelos cenários seguintes:

- A casa de Pablo: Composta pelo quarto do Pablo, banheiro, cozinha e sala. É o ponto de partida do jogador e o cenário principal da exploração inicial.
- As plataformas: Uma área de transição que conecta a casa a outros locais importantes, como a clínica da psicóloga e a faculdade. Nesta área em que os inimigos aparecem.
- A faculdade: Local onde se encontra a maioria dos outros NPCs que possuem diálogos e missões.
- A clínica da psicóloga: Composta pelo exterior e interior, área cuja psicóloga se encontra.

O design completo do jogo também inclui outros cenários cruciais. A experiência se iniciaria em um tutorial, localizado no topo de um edifício em uma noite chuvosa, cujo jogador aprenderia as mecânicas básicas antes de acordar no quarto de Pablo. O clímax da história ocorreria em uma área final, localizada do outro lado de um grande abismo que o personagem só conseguiria atravessar após completar as missões, levando ao confronto final com o monstro. Esses cenários específicos foram concebidos no GDD, mas sua implementação foi designada como um trabalho futuro do projeto.

A figura 4 apresenta uma captura de tela do jogo em funcionamento, ilustrando o estilo visual e a interface do usuário.

Figura 4 – Tela do jogo em execução



Fonte: Elaborada pela autora.

4.4 Arquitetura e Implementação

A implementação do protótipo foi realizada em C# no motor Unity, seguindo uma arquitetura baseada em componentes. O *software* foi estruturado em sistemas independentes, cada um responsável por uma parte da funcionalidade do jogo. O detalhamento completo da implementação desses sistemas encontra-se no Apêndice A.

O núcleo do jogo é formado pelo sistema de gerenciamento e configurações, responsável por inicializar variáveis globais, e pelo sistema de salvamento para armazenar e restaurar o estado da cena, permitindo a transição entre diferentes locais.

O controle do personagem foi centralizado em um script principal que recebe os eventos de entrada e delega as tarefas para componentes especializados de movimento e animação. A interação com o cenário é gerenciada por um sistema genérico que detecta quando o jogador está próximo a objetos interativos, ativando menus de interação ou mecânicas de portas e portais.

A experiência do jogador é controlada por um sistema de câmera que segue o personagem principal respeitando limites de borda, um sistema de portas e entradas para a transição entre cômodos, um sistema de neblina que gerencia a visibilidade de áreas ainda não exploradas e um sistema de plataforma com colisão unidirecional.

5 Análise comparativa de ferramentas de IA para animação 2D

5.1 Visão geral da análise comparativa

Como já foi mencionado, a aplicação de IA para a criação de animações 2D não foi muito explorada na literatura, tendo um potencial muito grande a ser descoberto. Atualmente, a maioria das ferramentas generativas de vídeo é voltada para ambientes tridimensionais e realistas. Diante desse cenário, esta análise busca investigar a capacidade e o resultado das tecnologias atuais quando aplicadas ao contexto da animação 2D para um jogo.

Durante a análise preliminar, a ferramenta AI Sprite Sheet Maker foi imediatamente descartada do estudo por não ser capaz de gerar resultados. Nas seções seguintes, é apresentada uma análise das demais ferramentas, sendo o objetivo deste capítulo responder a uma série de questões-chave pelos seguintes eixos de investigação:

- Avaliar se ferramentas com foco em realismo podem ser adaptadas para a animação 2D;
- Analisar o nível de desenvolvimento das ferramentas que possuem foco em 2D;
- Determinar o grau de consistência que as IAs mantêm em relação a um design de personagem pré-existente e a um estilo artístico específico;
- Verificar a possibilidade de utilizar ferramentas de geração de imagem para auxiliar na animação, incluindo a criação sequencial de quadros e a geração de novas poses ou vistas do personagem (como a vista lateral a partir da frontal); e
- Investigar a capacidade das ferramentas de gerar uma imagem *pixel perfect*, característico do estilo *pixel art*.

Ao final, busca-se mostrar o papel prático dessas tecnologias no processo de desenvolvimento de um jogo, posicionando-as não como uma possível substituição ao trabalho artístico, mas como ferramentas potenciais para otimizar e facilitar o complexo processo de animação.









O aprofundamento dos experimentos realizados em cada ferramenta, encontram-se no Apêndice B.

5.2 Análise do RosebudAI

A ferramenta RosebudAI foi selecionada por seu foco aparente na criação de *sprite sheets* para jogos. O objetivo dos testes (detalhados no quadro 1) foi gerar um ciclo de

caminhada para o personagem Pablo (Figura 5). No entanto, a ferramenta falhou em todos os testes: a primeira tentativa gerou um protótipo de jogo 3D; uma segunda tentativa produziu uma animação 2D de baixa qualidade que apenas dividia a imagem de referência ao meio; e os testes na seção de *assets* geraram *sprites* inconsistentes que desconsideravam a referência.

Quadro 1 – Testes realizados no RosebudAI

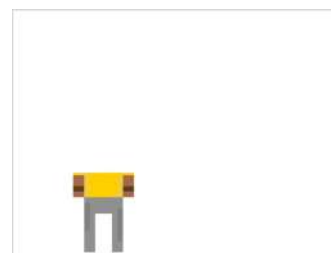
| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 |
|--|---|---|---|---|
| Objetivo | Sprite sheet do ciclo de caminhada | Sprite sheet do ciclo de caminhada | Sprite sheet do ciclo de caminhada | Sprite sheet do ciclo de caminhada |
| Funcionalidade usada | Principal | Principal | Geração de Assets | Geração de Assets |
| Sprite anexado |  |  |  |  |
| Prompt | Simples e direto | Específico ambiente 2D | Simples e direto | Detalhado |
| Contexto do chat | Nenhum | Teste 1 | Nenhum | Nenhum |
| Resultado exemplo |  |  |  |  |
| Semelhança com as imagens de referência | Muito baixa | Extremamente alta | Muito baixa | Extremamente baixa |
| Consistência com o estilo | Extremamente baixa | Extremamente alta | Baixa | Baixa |
| Qualidade do sprite baseada na aparência do personagem | Média | Média | Média | Muito baixa |
| Precisão com o prompt | Extremamente baixa | Muito baixa | Baixa | Média |
| Nível de deformação | Extremamente baixo | Extremamente baixo | Extremamente baixo | Baixo |
| Taxa de repetição | - | Extremamente alta | - | Extremamente alta |
| Taxa de incoerência do movimento | Muito alta | Extremamente alta | Baixa | Baixa |

Fonte: Elaborada pela autora.

Figura 5 – Animação de caminhada gerada pelo Rosebud AI



(a) *Frame 1* (metade superior).



(b) *Frame 2* (metade inferior).



Fonte: Elaborada pela autora.

A análise concluiu que a ferramenta RosebudAI é inadequada para o projeto. Embora demonstre potencial para a prototipagem rápida de jogos simples para usuários sem conhecimento em programação, ela falhou em produzir qualquer animação 2D consistente a partir de uma imagem de referência.

5.3 Análise do SpriteSheetGPT

A ferramenta SpriteSheetGPT foi analisada por seu nome sugerir um foco direto na criação de *sprite sheets*. O objetivo foi gerar um ciclo de caminhada para o personagem Pablo (conforme teste resumido no quadro 2). Nos testes iniciais, a ferramenta falhou em gerar imagens, respondendo apenas com texto. Após diversas interações, a ferramenta produziu uma única imagem (Figura 6) que demonstrou baixa consistência com a referência, baixa precisão com o prompt e apenas reproduziu levemente o estilo *pixel art*. A conclusão é que o SpriteSheetGPT se mostrou extremamente instável para qualquer aplicação prática, destacando a importância da repetibilidade em ferramentas de IA.

Quadro 2 – Testes realizados no SpriteSheetGPT

| Bateria de teste | Teste 1 |
|--|--|
| Objetivo | Sprite sheet do ciclo de caminhada |
| Sprite anexado |  |
| Prompt | Simples |
| Resultado exemplo |  |
| Semelhança com as imagens de referência | Extremamente baixa |
| Consistência com o estilo | Média |
| Qualidade do sprite baseada na aparência do personagem | Muito baixa |
| Precisão com o prompt | Média |
| Nível de deformação | Alto |
| Taxa de repetição | Extremamente alta |
| Taxa de incoerência do movimento | Média |

Fonte: Elaborada pela autora.

Figura 6 – *Sprite sheet* gerado pelo SpriteSheetGPT















Fonte: Elaborada pela autora.

5.4 Análise do CGDream

A ferramenta CGDream foi selecionada por sua capacidade de gerar imagens a partir de uma referência e uma descrição textual. O objetivo foi criar um *sprite* do Pablo em *side view* a partir da imagem em *front view*. A ferramenta oferece dois modelos principais: o Juggernaut XL, ideal para fotorrealismo (XU; COHEN; CLARK, 2025), e

o Flux, conhecido pela alta fidelidade aos *prompts* (GREENBERG, 2025). Os diversos testes realizados (detalhados no quadro 3) exploraram as funcionalidades "Imagem como referência" e "Personagem consistente". A primeira falhou em gerar a pose correta, priorizando a semelhança com a referência em vez da instrução do *prompt*. A segunda ignorou completamente a imagem de referência.

Quadro 3 – Testes realizados no CGDream

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 | Teste 6 |
|--|---|---|---|---|---|---|
| Objetivo | Pablo em side view | Pablo em side view | Pablo em side view | Pablo em side view | Pablo em side view | Pablo em side view |
| Modelo de IA | Flux | Juggernaut XL | Flux | Flux | Juggernaut XL | Juggernaut XL |
| Funcionalidade usada | Imagem de referência | Imagem de referência | Personagem consistente | Personagem consistente | Personagem consistente | Personagem consistente |
| Sprite anexado |  |  |  |  |  |  |
| Estilo de referência | Não | Não | Não | Sim | Sim | Sim |
| Filtro e Palavra negativa | - | - | - | Filtro pixelizado | Palavra "blur" | Palavra "blur" e "3D" |
| Resultado exemplo |  |  |  |  |  |  |
| Semelhança com as imagens de referência | Muito alta | Muito alta | Extremamente baixa | Baixa | Muito baixa | Extremamente baixa |
| Consistência com o estilo | Muito alta | Alta | Baixa | Baixa | Extremamente baixa | Baixa |
| Qualidade do sprite baseada na aparência do personagem | Alta | Média | Extremamente baixa | Alta | Muito alta | Muito baixa |
| Precisão da pose | Muito baixa | Extremamente baixa | Alta | Baixa | Alta | Média |
| Nível de deformação | Extremamente baixo | Baixo | Extremamente baixo | Médio | Muito baixo | Baixo |

Fonte: Elaborada pela autora.

Uma combinação das funcionalidades "Personagem Consistente" e "Estilo de Referência" melhorou a consistência, mas introduziu novos erros, como imagens borradas (modelo Flux) ou fotorrealistas (modelo Juggernaut XL). Foi possível evitar o fotorrealismo utilizando "3D" como palavra negativa, mas isso ainda não gerou uma imagem no estilo pixel art (Figura 7). Concluiu-se que a ferramenta não foi adequada para gerar *sprites* consistentes para o projeto.

Figura 7 – Melhor *sprite* em *side view* gerado pelo CGDream



Fonte: Elaborada pela autora.

5.5 Análise do Pixie.haus

A ferramenta Pixie.haus foi analisada por ser especializada em *pixel art* com um editor de imagens integrado. O objetivo foi animar a porta B em *side view* (conforme resumido no quadro 4). Uma limitação crítica foi a incapacidade de importar *sprites* externos, exigindo que

a porta fosse redesenhada manualmente em seu editor, que se mostrou fraco e pouco eficiente. Os testes de animação, usando o modelo ByteDance Seedance-1-Lite, falharam em gerar o movimento de abertura correto, resultando em animações que deslizavam ou rotacionavam a porta de forma imprecisa (Figura 8).

Quadro 4 – Testes realizados na Pixie.haus

| Bateria de teste | Teste 1 |
|---|--|
| Objetivo | Vídeo da porta em side view abrindo |
| Modelo de IA | ByteDance Seedance-1-Lite |
| Sprite anexado |  |
| Resultado exemplo |  |
| Semelhança com as imagens de referência | Alta |
| Consistência com o estilo | Alta |
| Nível de deformação | Extremamente baixo |
| Incoerência do movimento | Muito alta |

Fonte: Elaborada pela autora.

Figura 8 – Animação da porta B abrindo em side view gerada pelo Pixie.Haus



Fonte: Elaborada pela autora.







O principal diferencial da ferramenta foi sua capacidade de gerar resultados *pixel perfect*. No entanto, devido à imprecisão na interpretação dos *prompts* e à limitação de créditos gratuitos, a ferramenta foi descartada.

5.6 Análise do Animated Drawings

A ferramenta Animated Drawings, que implementa o algoritmo proposto por [Smith et al. \(2023\)](#), foi selecionada por seu foco em animar figuras humanoides 2D. O objetivo foi

animar a personagem Luz, que foi escolhida por não ter membros sobrepostos, um requisito da plataforma. A IA identificava um esqueleto no desenho e aplicava transformações para gerar animações. Os testes (resumidos no quadro 5), avaliaram diferentes *sprites* da personagem.

Quadro 5 – Testes realizados no Animated Drawings

| Bateria de teste | Teste 1 | Teste 2 | Teste 4 |
|---|---|---|---|
| Objetivo | Animação de caminhada | Animação de caminhada | Animação de caminhada |
| Sprite anexado |  |  |  |
| Resultado exemplo |  |  |  |
| Semelhança com as imagens de referência | Extremamente alta | Extremamente alta | Extremamente alta |
| Consistência com o estilo | Alta | Alta | Alta |
| Nível de deformação | Alto | Médio | Baixa |
| Taxa de incoerência do movimento | Extremamente baixa | Extremamente baixa | Extremamente baixa |

Fonte: Elaborada pela autora.

O ponto forte da ferramenta foi garantir completamente a consistência com o *sprite* original, pois ela deforma a imagem em vez de gerar novos quadros. No entanto, essa mesma técnica se mostrou incompatível com a estética *pixel art* (Figura 9), pois quebrava o padrão *pixel perfect* ao criar diagonais e curvas. A ferramenta foi considerada funcionalmente satisfatória, mas inadequada para o estilo visual específico do projeto.

Figura 9 – Quadro da animação de caminhada gerada pelo Animated Drawings



Fonte: Elaborada pela autora.

5.7 Análise do God Mode AI

A God Mode AI destacou-se por seu foco explícito na animação de *sprites* para jogos e por suas atualizações constantes. O objetivo foi gerar a animação de caminhada do Pablo.

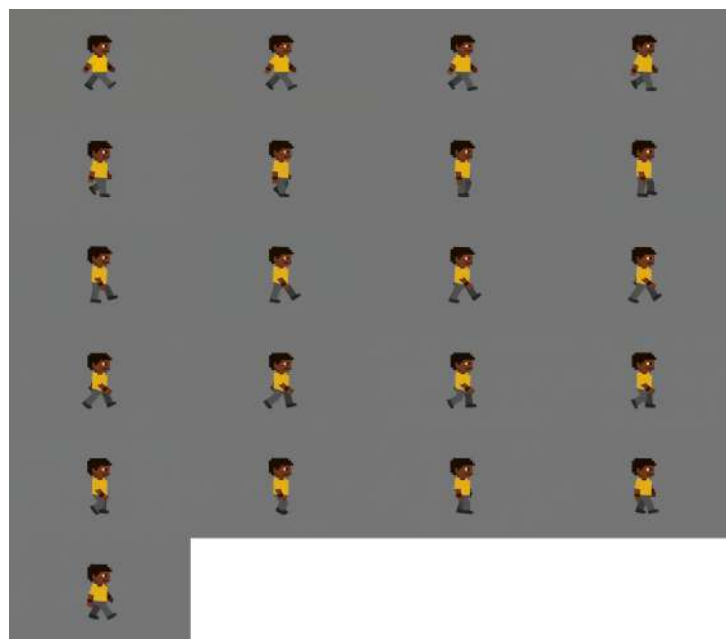
Conforme o quadro 6, o fluxo de trabalho inicial, específico para *pixel art*, produzia resultados *pixel perfect*, mas o estilo gerado era inconsistente e o personagem andava na diagonal. Uma versão posterior da ferramenta, com funcionalidades como auto reposicionamento, gerou um resultado de animação superior (Figura 10), embora não mais *pixel perfect*.

Quadro 6 – Testes realizados no God Mode AI

| Bateria de teste | Teste 1 | Teste 2 |
|---|---|--|
| Objetivo | Animação de caminhada | Animação de caminhada |
| Funcionalidade usada | Geração de sprite em pixel art | Geração de sprite |
| Sprite anexado |  |  |
| Resultado exemplo |  |  |
| Semelhança com as imagens de referência | Alta | Alta |
| Consistência com o estilo | Alta | Muito Alta |
| Nível de deformação | Extremamente baixo | Médio |
| Incoerência do movimento | Média | Extremamente baixa |

Fonte: Elaborada pela autora.

Figura 10 – Animação de caminhada gerada no God Mode AI











Fonte: Elaborada pela autora.

A ferramenta mostrou grande potencial, oferecendo recursos valiosos como re-geração parcial e exportação direta para *sprite sheet*. No entanto, a severa limitação de créditos gratuitos impediu uma análise aprofundada, e os resultados obtidos não possuíam a qualidade necessária para implementação direta.

5.8 Análise do ChatGPT

O ChatGPT, que utilizava na época o modelo GPT-4o, que não possui um único foco. O modelo possui funcionalidades de visão computacional (OPENAI, 2025a) e geração de imagem nativa (OPENAI, 2025b). Os testes (resumidos no quadro 7), tiveram dois objetivos: gerar um *sprite* do Pablo em *side view* e gerar um *sprite sheet* de caminhada.

Quadro 7 – Testes realizados no ChatGPT

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 |
|--|---|---|--|---|
| Objetivo | Pablo em side view | Pablo em side view | Sprite sheet do ciclo de caminhada | Sprite sheet do ciclo de caminhada |
| Sprite anexado |  |  |  |  |
| Contexto do chat | Nenhum | Teste 1 | Nenhum | Nenhum |
| Resultado exemplo |  |  |  |  |
| Semelhança com as imagens de referência | Alta | Muito alta | Média | Muito alta |
| Consistência com o estilo | Extremamente alta | Extremamente alta | Alta | Alta |
| Qualidade do sprite baseada na aparência do personagem | Média | Alta | Baixa | Alta |
| Precisão com o prompt | Extremamente alta | Extremamente alta | Alta | Média |
| Nível de deformação | Baixo | Muito baixo | Baixo | Baixo |
| Taxa de repetição | - | - | Muito alta | Extremamente alta |
| Incoerência do movimento ou pose | Extremamente baixa | Extremamente baixa | Baixa | Baixa |

Fonte: Elaborada pela autora.

No primeiro objetivo, a ferramenta obteve sucesso, gerando um *sprite* (Figura 11) com alta consistência de estilo e características. No segundo objetivo, a ferramenta falhou completamente. Os *sprite sheets* gerados continham majoritariamente quadros repetidos, sem progresso de movimento, além de apresentarem mais deformações. Concluiu-se que o ChatGPT é altamente eficaz para gerar poses estáticas, mas inadequado para criar sequências de animação (*sprite sheets*).

Figura 11 – Melhor *sprite* em *side view* gerado pelo ChatGPT

Fonte: Elaborada pela autora.

5.9 Análise do Pixel Lab

A plataforma Pixel Lab foi uma ferramenta essencial para este trabalho, sendo avaliada tanto por suas capacidades generativas quanto por seu papel como editor de pós-processamento. A análise das funcionalidades (resumida no quadro 8), mostrou que as ferramentas de geração (Rotação e Animação-para-Animação) produziram resultados apenas parciais. A Rotação criou uma base de *side view*, mas com deformações. A Animação-para-Animação iniciou falhou em manter o *sprite* de referência, priorizando a forma da animação base (Figura 12). Apenas quando a animação base e o *sprite* de referência possuíam o mesmo formato, o movimento foi gerado com sucesso (Figura 12).

Quadro 8 – Testes realizados no Pixel Lab

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 |
|---|---|---|--|---|---|
| Objetivo | Pablo em side view | Pablo em side view | Animação de caminhada | Animação de caminhada | Ajuste fino de material de outras ferramentas |
| Funcionalidade usada | Rotação rápida | Rotação | Animação para animação | Animação para animação | Editor |
| Referência anexada |  |  |  |  |  |
| Resultado exemplo |  |  |  |  |  |
| Semelhança com <i>sprite</i> original | Extremamente alta | Extremamente alta | Média | Extremamente alta | Extremamente alta |
| Consistência com o estilo | Extremamente alta | Extremamente alta | Extremamente alta | Extremamente alta | Extremamente alta |
| Qualidade do <i>sprite</i> baseada na aparência do personagem | Baixa | Média | Média | Alta | Muito Alta |
| Precisão da pose e do movimento | Alta | Extremamente alta | Muito alta | Alta | Extremamente alta |
| Nível de deformação | Médio | Muito baixo | Médio | Médio | Extremamente baixo |

Fonte: Elaborada pela autora.

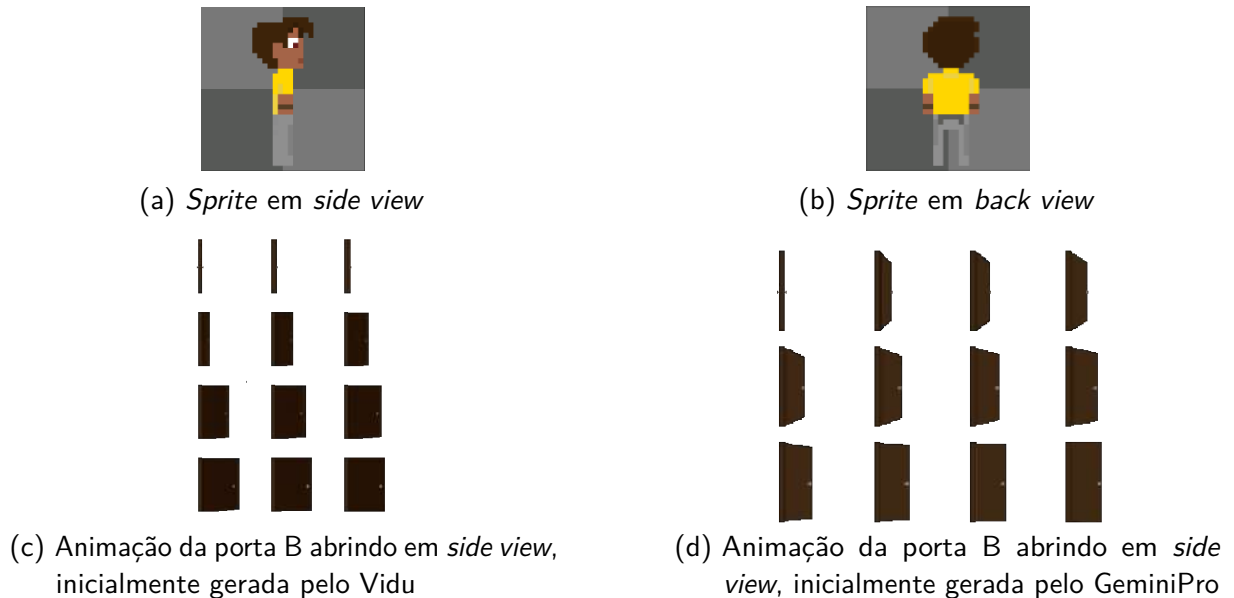
Figura 12 – Quadro da animação de caminhada gerado pelo Pixel Lab



Fonte: Elaborada pela autora.

O diferencial decisivo da ferramenta foi seu robusto editor de *pixel art*. Ele se provou essencial para o fluxo de trabalho híbrido, permitindo o refinamento manual e a correção de artefatos gerados por outras ferramentas. Foi nesta plataforma que alguns dos resultados brutos foram transformados nos *assets* finais de alta qualidade (Figura 13) usados no jogo.

Figura 13 – Resultados finais editados no Pixel Lab



Fonte: Elaborada pela autora.

5.10 Análise do OpenArt.AI

A ferramenta OpenArt.AI foi analisada por sua capacidade de agregar múltiplos modelos de IA para edição de imagem e geração de vídeo. O objetivo foi gerar uma animação ou *sprite* do Pablo sentado (conforme detalhado no quadro 9). O teste de vídeo (modelo Vidu Q1) falhou, gerando uma animação 3D que rotacionava o personagem.

Os testes de edição de imagem via chat (Chat para editar) foram mais promissores. O modelo SeeEdit (provavelmente SeedEdit 3.0 da BytePlus) produziu o melhor resultado em termos de pose (Figura 14), mas falhou em manter o estilo *pixel art*, suavizando as pernas. Os outros modelos, como Flux Kontext e Gemini, falharam ao gerar uma cadeira ou posicionar o personagem incorretamente. A ferramenta mostrou potencial para ideação de poses, mas foi prejudicada pela limitação de créditos, que impediu o ciclo iterativo de refinamento via chat.

Quadro 9 – Testes realizados no OpenArt.AI

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 |
|---|---|---|---|---|---|
| Objetivo | Vídeo do personagem sentado | Sprite do personagem sentado | Sprite do personagem sentado | Sprite do personagem sentado | Sprite do personagem sentado |
| Funcionalidade usada | Elementos para vídeo | Chat para editar | Chat para editar | Chat para editar | Chat para editar |
| Modelo de IA | Vidu Q1 | SeedEdit | Flux Kontext | Gemini | GPT |
| Sprite anexado |  |  |  |  |  |
| Resultado exemplo |  |  |  |  |  |
| Semelhança com as imagens de referência | Muito alta | Extremamente alta | Extremamente alta | Extremamente alta | Muito alta |
| Consistência com o estilo | Muito baixa | Muito alta | Extremamente alta | Alta | Extremamente alta |
| Precisão com o prompt | Muito alta | Extremamente alta | Média | Média | Muito alta |
| Nível de deformação | Extremamente baixo | Extremamente baixo | Extremamente baixo | Extremamente baixo | Extremamente baixo |
| Incoerência do movimento ou pose | Média | Extremamente baixa | Extremamente baixa | Extremamente baixa | Extremamente baixa |

Fonte: Elaborada pela autora.

Figura 14 – Melhor *sprite* sentado gerado no OpenArt.AI

Fonte: Elaborada pela autora.

5.11 Análise do Vidu

A ferramenta Vidu, utilizando o modelo Vidu 2.0, foi analisada por suas funcionalidades de geração de vídeo a partir de referências. Os testes com a funcionalidade "Referência para vídeo" revelaram uma tendência para gerar resultados em 3D. Embora um resultado 2D tenha sido obtido, ele apresentou uma taxa de deformação significativamente maior que as versões 3D. A análise concluiu que a ferramenta possui dificuldade em manter a qualidade ao gerar animações 2D.

A funcionalidade "Imagem para vídeo", que permite definir o primeiro e o último quadro, foi mais bem-sucedida. Embora tenha falhado em movimentos cíclicos (como o pulo), ela gerou uma animação satisfatória da porta abrindo em *side view* (Figura 15). Esse resultado serviu como uma base que, após pós-processamento, foi utilizada no jogo. Os testes estão resumidos no quadro 10.

Figura 15 – Quadro da animação da porta B em *side view* gerada no Vidu

Fonte: Elaborada pela autora, utilizando a ferramenta Vidu.

Quadro 10 – Testes realizados no Vidu

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 |
|---|-----------------------|-----------------------|-------------------------|---------------------------|---------------------------|
| Objetivo | Animação de caminhada | Animação de caminhada | Animação de abrir porta | Animação da porta abrindo | Animação da porta abrindo |
| Funcionalidade usada | Referência para vídeo | Referência para vídeo | Referência para vídeo | Imagem para vídeo | Imagem para vídeo |
| Sprite anexado | | | | | |
| Resultado exemplo | | | | | |
| Semelhança com as imagens de referência | Alta | Extremamente alta | Média | Baixa | Muito alta |
| Consistência com o estilo | Média | Alta | Muito alta | Média | Alta |
| Nível de deformação | Alto | Alto | Alto | Alto | Baixo |
| Incoerência do movimento | Média | Alta | Muito alto | Muito alto | Muito baixa |

Fonte: Elaborada pela autora.

5.12 Análise da Ferramenta Gemini Pro

A ferramenta Gemini Pro, utilizando os modelos Gemini 2.5 Pro, Imagen 4 e Veo 3 (WOODWARD, 2025), foi a ferramenta mais avançada testada. A análise teve múltiplos objetivos, (cujos testes estão resumidos no quadro 11). Na geração de *sprites*, a ferramenta demonstrou um desempenho superior, gerando imagens em *side view* e *back view* (Figura 16) com alta consistência de estilo e características. No entanto, na geração de *sprite sheets*, ela falhou, repetindo quadros e perdendo consistência, mostrando dificuldade em entender sequências de animação.

O maior sucesso da ferramenta foi na geração de vídeo. Ela produziu animações consistentes e de alta precisão para a caminhada (Figura 17a), o pulo (Figura 17b) e as portas (Figuras 17c e 17d). Apesar de ser a melhor ferramenta, ela não possui editor embutido, e seus resultados brutos ainda continham deformações e inconsistências que exigiram pós-processamento.

Figura 16 – Melhores *sprites* gerados no Gemini Pro



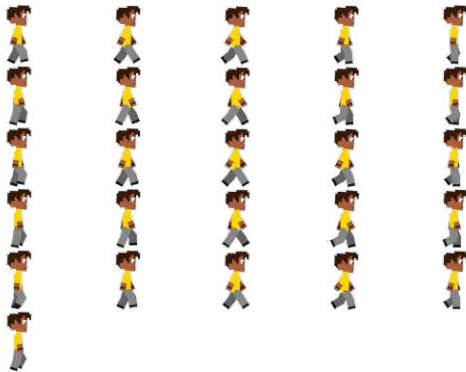
(a) *Sprite em side view*



(b) *Sprite em back view*

Fonte: Elaborada pela autora.

Figura 17 – Animações geradas no Gemini Pro



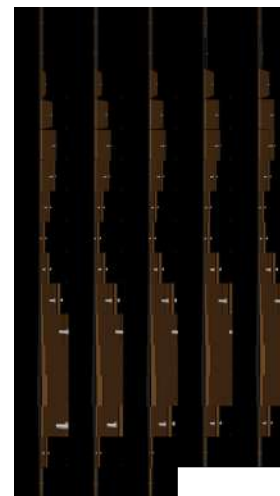
(a) Animação de caminhada



(b) Animação de pulo



(c) Animação da porta A abrindo



(d) Animação da porta B abrindo

Fonte: Elaborada pela autora.

Quadro 11 – Testes realizados na Gemini Pro

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 | Teste 6 |
|---|---|---|---|--|---|---|
| Objetivo | Pablo em side view | Sprite sheet do ciclo de caminhada | Animação de caminhada | Animação de pulo | Animação da porta frontal abrindo | Animação da porta lateral abrindo |
| Sprite anexado |  |  |  |  |  |  |
| Resultado exemplo |  |  |  |  |  |  |
| Semelhança com as imagens de referência | Muito alta | Média | Muito alta | Muito alta | Muito alta | Médio |
| Consistência com o estilo | Extremamente alta | Média | Alta | Alta | Alta | Alta |
| Precisão do movimento ou pose | Extremamente alta | Muito baixa | Extremamente Alta | Extremamente Alta | Extremamente alta | Baixa |
| Nível de deformação | Baixo | Alto | Muito baixo | Baixo | Baixo | Alto |

Fonte: Elaborada pela autora.

5.13 Quadro comparativo e discussão geral

Após a análise individual detalhada de cada ferramenta, esta seção tem como objetivo consolidar os resultados para trazer as respostas dos eixos de investigação definidos na Seção 5.1. Os quadros 12 e 13 apresentam um resumo do desempenho de todas as ferramentas avaliadas, utilizando os critérios simplificados determinados na metodologia.

Quadro 12 – Tabela comparativa das ferramentas de IA utilizadas (Parte 1 de 2)

| Ferramenta | Foco em 2D | Consistência e precisão geral | Adequação ao estilo de pixel art | Nível de customização |
|-------------------|--------------------|-------------------------------|----------------------------------|-----------------------|
| Rosebud AI | Médio | Muito Baixo | Média | Muito Baixo |
| SpriteSheetGPT | Médio | Muito Baixo | Extremamente Baixo | Baixo |
| CGDream | Baixo | Baixo | Muito Baixa | Extremamente Alto |
| Pixie.haus | Extremamente alto | Baixo | Extremamente Alto | Médio |
| Animated Drawings | Alto | Extremamente Alto | Médio | Extremamente Baixo |
| God Mode AI | Extremamente alto | Alto | Alta | Muito Alto |
| ChatGPT | Baixo | Muito Alto | Muito Alto | Baixo |
| Pixel Lab | Extremamente alto | Muito Alto | Extremamente Alta | Extremamente Alto |
| OpenArt.AI | Baixo | Alto | Média | Alto |
| Vidu | Extremamente baixo | Médio | Baixa | Alto |
| Gemini Pro | Baixo | Extremamente Alto | Alto | Médio |

Fonte: Elaborada pela autora.

Observou-se que as ferramentas sem foco específico em 2D apresentaram grande dificuldade em manter o ambiente bidimensional durante a geração de vídeos e em simular o estilo *pixel art* durante a criação de imagens. Entre as sete ferramentas sem especialização, apenas duas (ChatGPT e Gemini Pro) conseguiram se adequar de forma satisfatória ao estilo, por conta de seu nível de desenvolvimento avançado. Com uma maior manipulação do prompt, uso de funcionalidades alternativas e diversas tentativas de teste, foi possível adaptar mais algumas ferramentas (Vidu, OpenArt.AI e CGDream) para gerar um resultado 2D *pixel art*.

Quadro 13 – Tabela comparativa das ferramentas de IA utilizadas (Parte 2 de 2)

| Ferramenta | Facilidade de edição e refinamento do material gerado | Eficiência | Tipo de conteúdo gerado |
|-------------------|---|------------|--|
| Rosebud AI | Extremamente baixa | Baixa | Imagem estática, sprite sheet (conjunto de poses) e animação sem exportação disponível |
| SpriteSheetGPT | Média | Média | Sprite sheet (conjunto de poses) |
| CGDream | Média | Baixa | Imagem estática |
| Pixie.haus | Baixa* | Baixa | GIF |
| Animated Drawings | Baixa | Média | Vídeo |
| God Mode AI | Alta | Média | Vídeo e sprite sheet (animação) |
| ChatGPT | Baixa | Média | Imagem estática e sprite sheet (conjunto de poses) |
| Pixel Lab | Extremamente alta | Alta | Imagem estática e sprite sheet (animação) |
| OpenArt.AI | Alta | Média | Imagem estática e vídeo |
| Vidu | Baixa | Média | Vídeo |
| Gemini Pro | Baixa | Média | Imagem estática, sprite sheet (conjunto de poses) e vídeo |

*O editor da ferramenta não oferece opção de editar a animação gerada

Fonte: Elaborada pela autora.

Essa adaptação ocorre em detrimento da eficiência (exigindo um número maior de iterações para se obter um resultado aceitável) ou se mantém apenas para a geração de imagens e não de animações, como é o foco da análise do momento.

As ferramentas com foco em 2D ou *pixel art* (Pixie.haus, Animated Drawings, Pixel Lab, God Mode AI) demonstraram um estágio de desenvolvimento geral inferior em relação às outras. Além de existir uma gama menor de ferramentas com essa especialidade, limitações como um número de créditos gratuitos mais restritivo, menor flexibilidade nas opções de animação e, em alguns casos, menor precisão na interpretação de prompts foram comuns. Teoriza-se que o mercado mais amplo das ferramentas generalistas impulsiona uma competição que, junto com o maior grau de desenvolvimento e lucros bem estabelecidos, resultam em maior precisão e generosidade nos planos de uso. A grande vantagem das ferramentas especializadas, no entanto, reside em sua alta consistência e adequação ao estilo *pixel art*, gerando resultados com maior coerência e evitando as distorções dos pixels que foram um problema recorrente nos modelos generalistas de geração de vídeo.

A manutenção da consistência com um design pré-existente foi um dos maiores desafios observados. Ferramentas menos desenvolvidas ou não focadas em *pixel art* frequentemente falharam em replicar características, cores e ou o estilo da imagem de referência. Novamente, os modelos generalistas mais avançados (Gemini Pro, ChatGPT) se destacaram, demonstrando uma notável capacidade de manter a consistência do personagem, mesmo sem uma adequação perfeita ao estilo *pixel art*. Ambos superaram inclusive algumas ferramentas especializadas que, embora acertassem no estilo, apresentavam falhas na fidelidade de cores e detalhes. Uma exceção notável foi a ferramenta Animated Drawings, que, por sua metodologia de deformação da imagem original em vez de geração de uma nova, garantiu consistência perfeita, chegando ao mesmo nível das ferramentas mais avançadas nesse quesito específico. Porém, a mesma possui limitações nos movimentos a serem animados e não possui alta adequação ao estilo específico de *pixel art*. Mesmo assim, ela apresenta grande potencial e mostra-se ideal para animações 2D de outros estilos.

A análise confirmou que a geração de imagens é uma etapa fundamental e de grande impacto no fluxo de trabalho. A qualidade das animações geradas por ferramentas de vídeo foi significativamente superior quando a imagem de referência fornecida já estava na perspectiva correta para o movimento desejado. Tentar gerar a nova pose e o movimento simultaneamente sobrecarregou os modelos, resultando em maior inconsistência. Esse fato consolida o papel do Pixel Lab como facilitador do processo de animação mesmo sem gerar diretamente a mesma. Além disso, mostra a importância da funcionalidade de auto reposicionamento presente no God Mode AI, que permite justamente ao usuário escolher a aparência exata do personagem no primeiro *frame* da animação e fazer essa geração antes de criar o vídeo.

A conformidade com o padrão *pixel perfect* foi difícil de ser alcançada sem nenhuma especialização. Ferramentas generalistas muitas vezes geravam um *pixel art* que tinha pixels de tamanhos variados ou retangulares, com apenas as IAs mais avançadas (Gemini Pro e Chat GPT) sendo capazes de simular com exatidão o *pixel perfect* em imagens. Porém, mesmo essas consistentemente falharam em manter esse padrão em animações de vídeo, que introduziam distorções e incoerência com o estilo. Apenas as ferramentas com funcionalidades específicas para *pixel art* (Pixie.haus, Pixel Lab e God Mode AI em seu fluxo de pixelização) foram capazes de gerar ou exportar resultados que respeitavam esse padrão técnico.

O ponto mais revelador da análise foi a importância do pós-processamento. A presença de um editor embutido e robusto, como o do Pixel Lab, mostrou-se um diferencial decisivo, otimizando drasticamente a eficiência e agilizando até o fluxo de trabalho de outras ferramentas. A presença de um editor permite que a IA apenas precise produzir uma base que torne mais rápido o processo de produção de uma animação. Enquanto isso, a ausência do mesmo apenas causa uma perda de eficiência, no qual gerar um resultado de qualidade adequada pode levar mais tempo do que criar uma animação com menos *frames* do zero. Durante os experimentos, foi notável a diferença entre tentar produzir um resultado perfeito apenas utilizando a ferramenta

bruta e sem edição manual, tentar editar um resultado gerado por uma IA sem editor embutido, e editar a animação no mesmo ambiente em que esta foi gerada.

5.14 Desenvolvimento de um fluxo de trabalho para animação com IA

A análise demonstrou que nenhuma ferramenta individual foi capaz de produzir um resultado final satisfatório de forma completamente autônoma. A solução mais eficaz emergiu da criação de um fluxo de trabalho híbrido, que combina os pontos fortes de múltiplas plataformas. Este processo otimizado consiste nas seguintes etapas:

1. Criação de uma referência na pose e perspectiva corretas, se necessário, utilizando uma ferramenta de IA para a geração de imagens;
2. Geração da animação base utilizando uma IA para a geração de vídeos de alta consistência (Gemini Pro ou Vidu), focando em obter um movimento fluido, preciso e fiel às características do personagem;
3. Conversão do vídeo em *sprite sheet* com fundo transparente (se necessário) usando ferramentas auxiliares (ezgif, removebg ou Pixilart);
4. Padronização para *pixel perfect* (se necessário), importando o *sprite sheet* para uma ferramenta de edição de *pixel art* (Pixilart), que ajusta a resolução e garante a uniformidade dos pixels; e
5. Pós-processamento e refinamento manual, importando o *sprite sheet* padronizado para uma ferramenta com um editor robusto de animação (Pixel Lab) para realizar as correções finas, como ajustar detalhes quadro a quadro, corrigir cores e manter a qualidade exigida pelo jogo.

Comparando as animações implementadas¹ no jogo, é possível entender a importância de cada etapa. Assim, quatro animações do personagem, duas animações da porta A e duas animações da porta B foram implementadas no jogo, e os diferentes resultados de cada uma delas destacam a importância da exportação adequada e do ajuste manual. Três dos resultados relacionados ao Pablo representavam o movimento de caminhada, e o que restou mostrava a animação de pulo.

A animação 1 de caminhada foi gerada pelo Gemini Pro (Figura 77, Apêndice B) e passou apenas pelo processo de extração de *sprite sheet* e remoção de fundo, sem nenhuma

¹ <<https://drive.google.com/drive/folders/1KecenFq0EJBUPfAqEwuWYcmsPWbaaSJm?usp=sharing>>. Acesso em: 19 nov. 2025.

outra edição ou conversão. Isso tornou o produto final extremamente semelhante ao produto bruto. Como dito na Seção B.11.4, a animação apresenta o *sprite* antigo do personagem em *side view* com diversas inconsistências, como mudanças nos tons de cores, adição de um sapato preto e perda da coerência com o estilo de *pixel art* (Figura 18). Porém, uma nova falha foi encontrada apenas na implementação, em que durante a conversão para um *sprite sheet* transparente, a ferramenta utilizada não conseguiu remover completamente o fundo, mantendo uma borda branca ao redor do *sprite*. Além disso, a qualidade da animação foi deteriorada pela necessidade de se realizar diversas operações externas no resultado bruto. Isso mostra o quão importante é a ferramenta de IA produzir o resultado no formato adequado, pois embora esse erro não aconteça mais após a transformação do *sprite sheet* em *pixel perfect*, em qualquer outro estilo 2D a borda branca se tornaria um problema difícil de ser consertado sem sacrificar eficiência ou qualidade.

Figura 18 – Quadros do vídeo da animação 1 de caminhada implementada no jogo



(a) Quadro antes da animação 1 de caminhada começar



(b) Quadro da animação 1 de caminhada

Fonte: Elaborada pela autora.

A animação de pulo foi criada novamente pelo Gemini Pro (Figura 78b, Apêndice B), passando pelos mesmos processos que a animação de caminhada. Numa tentativa de corrigir o erro da borda, foi feita uma pequena edição manual pela ferramenta de remoção de fundo, em vez de apenas aceitar o resultado automático. Isso causou outro problema: a erosão das bordas do *sprite*, que resultou em falhas no contorno do personagem e um leve arredondamento das quinas (Figura 19). Essa perda mínima não chamaria tanta atenção em outros estilos de desenho, porém ela compromete a estética *pixel art*. A animação que antes apresentava alta coerência com o estilo *pixel art*, tornou-se um resultado de baixa coerência pela necessidade da remoção de fundo. Isso evidencia o quão difícil de ser consertado é o problema da borda sem a pixelização.

A segunda animação de caminhada também foi gerada pelo Gemini Pro (Figura 80, Apêndice B) e passou pelo processo de extração de *sprite sheet*, conversão para o padrão *pixel perfect* e remoção de fundo, sem nenhuma edição ou ajuste após a pixelização. Isso fez com que o resultado final fosse distinto do resultado bruto, agora completamente coerente com o estilo de *pixel art*. Porém apresentou inconsistências visuais quadro a quadro (geradas pela interpolação incorreta de cores durante o redimensionamento), além de ainda manter pequenas

Figura 19 – Quadros do vídeo da animação de pulo implementada no jogo



(a) Quadro antes da animação de pulo começar



(b) Quadro da animação de pulo

Fonte: Elaborada pela autora.

falhas na fidelidade das características (Figura 20). As novas inconsistências causaram um efeito de *glitch* (erros visuais que parecem distorções digitais), com diversos pixels isolados mudando de cor em cada quadro. Esse resultado demonstra que transformar o vídeo para o padrão *pixel perfect* causa algumas falhas que precisam ser corrigidas, além de não substituir a necessidade de uma edição manual mais atenciosa.

Figura 20 – Quadros do vídeo da segunda animação de caminhada implementada no jogo



(a) Quadro antes da segunda animação de caminhada começar



(b) Quadro da segunda animação de caminhada



(c) Quadro seguinte da segunda animação de caminhada

Fonte: Elaborada pela autora.

A terceira animação de caminhada foi gerada pelo Pixel Lab (Figura 51c, Apêndice B) e não passou por nenhum outro processo ou edição (nem mesmo dentro do editor do Pixel Lab), gerando diretamente o *sprite sheet* transparente. Dessa forma, os mesmos erros descritos anteriormente ainda foram observados na animação (Figura 21), demonstrando a importância da correção manual, independentemente da exportação direta para o Unity e do padrão ser *pixel perfect*.

As animações da abertura da porta A consistem em duas versões diferentes do mesmo vídeo. A figura 22 apresenta diretamente o resultado bruto da IA (Figura 81a, Apêndice B) e a figura 23 mostra a versão final no qual foi realizada extensa edição do mesmo após a pixelização (Figura 81c, Apêndice B). A diferença entre os dois resultados é visível, com a animação bruta possuindo uma qualidade muito menor e demonstrando mudanças na moldura da porta enquanto a versão final é totalmente consistente com a porta ainda mantendo a

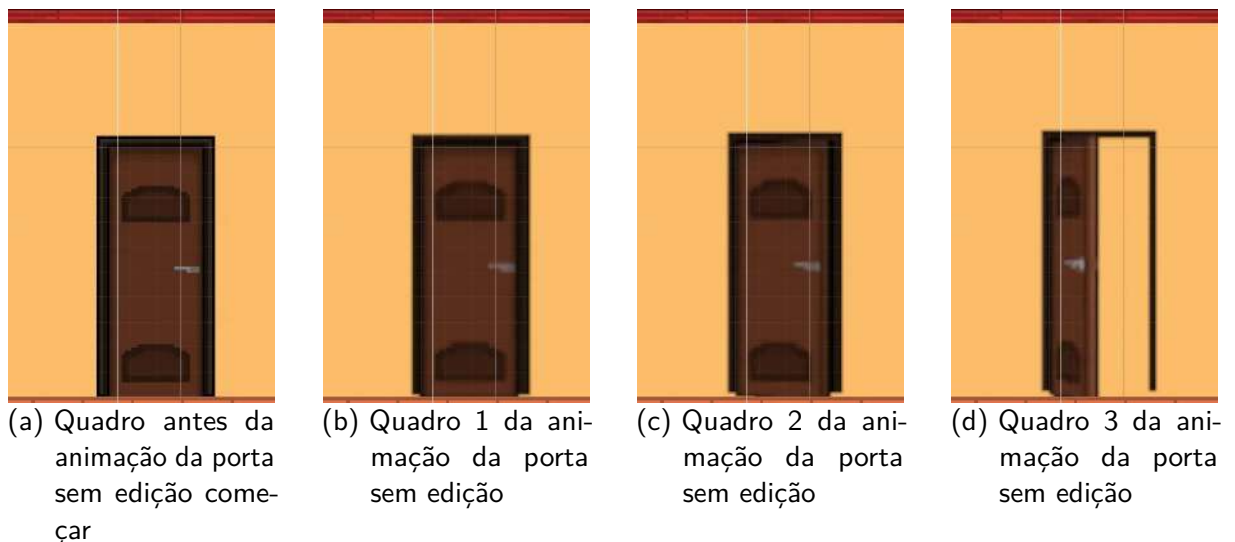
Figura 21 – Quadros do vídeo da animação 3 de caminhada implementada no jogo



Fonte: Elaborada pela autora.

precisão do movimento. A própria implementação da versão final foi mais fácil, pois era apenas necessário usar a mesma escala de tamanho para mantê-la do mesmo tamanho que o *sprite* original da porta, enquanto na versão inicial foi necessário achar esse valor por tentativa e erro.

Figura 22 – Quadros do vídeo da animação da porta A sem edição implementada no jogo



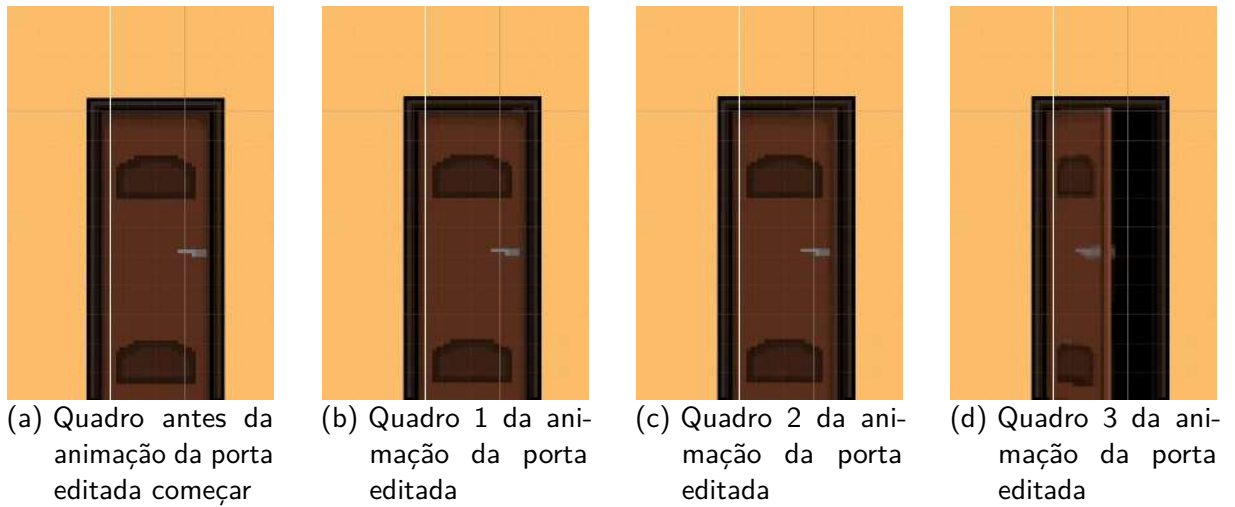
Fonte: Elaborada pela autora.

As animações da porta B em vista lateral foram produzidas utilizando como base os melhores resultados gerados pelas ferramentas Gemini Pro e Vidu. Ambos os vídeos brutos foram submetidos ao mesmo fluxo de trabalho de pós-processamento, que incluiu a conversão para *sprite sheet*, a padronização para *pixel perfect* e um refinamento manual extenso no Pixel Lab.

Como resultado, ambas as abordagens produziram animações finais de alta qualidade e prontas para implementação, conforme pode ser visto na figura 24. As duas versões se mostraram funcionalmente corretas e visualmente consistentes com o estilo do jogo.

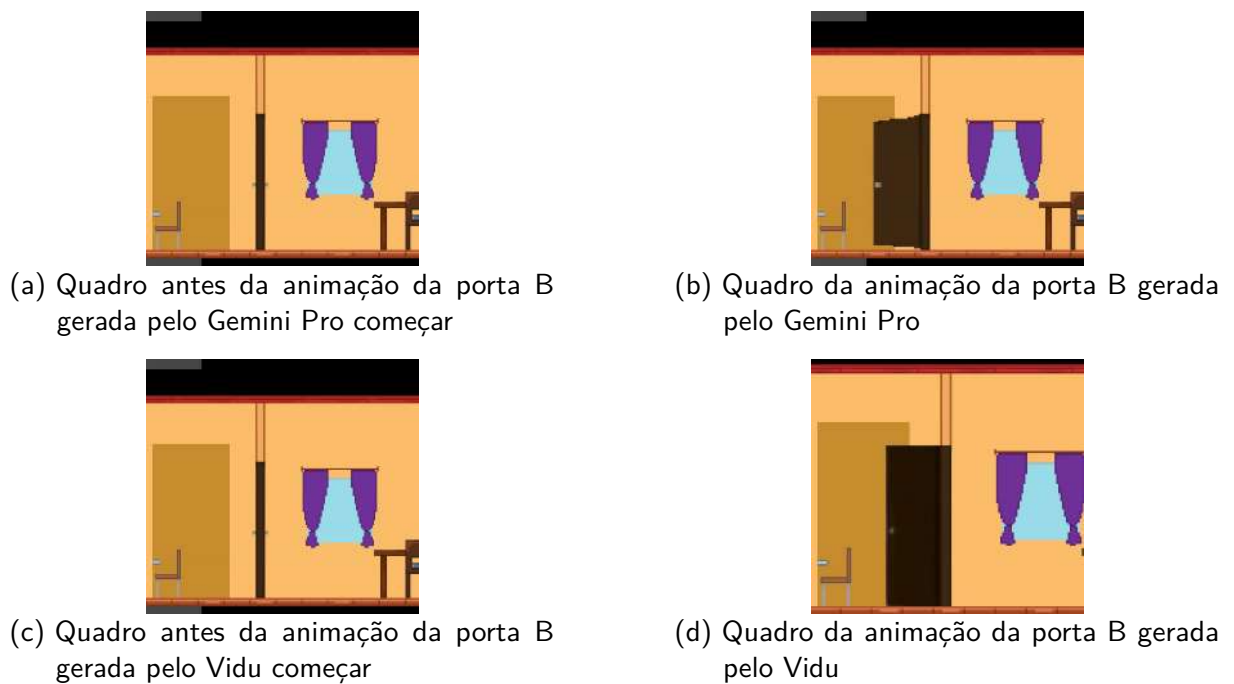
Esta análise dos resultados implementados reforça que a IA não serve como uma substituta do artista, mas sim como uma ferramenta para auxiliá-lo. Ela gera uma base visual

Figura 23 – Quadros do vídeo da animação da porta A editada implementada no jogo



Fonte: Elaborada pela autora.

Figura 24 – Quadros do vídeo das animações da porta B implementada no jogo



Fonte: Elaborada pela autora.

para o movimento, poupando ao artista o trabalho inicial de desenhar cada quadro do zero, porém a correção de falhas e detalhes artísticos específicos permanecem como etapas cruciais do processo criativo humano.

6 Conclusão

6.1 Conclusão

Este trabalho propôs-se a uma dupla jornada: o desenvolvimento de um protótipo de jogo sério que aborda temas de saúde mental e a realização de uma extensa análise comparativa de ferramentas de IA para o processo de animação 2D. Dentre as plataformas analisadas, a Gemini Pro foi a que apresentou o melhor desempenho na geração de animações 2D, possuindo maior precisão e fidelidade à imagem de referência, e adequando-se de forma satisfatória ao ambiente 2D. Contudo, mesmo esse melhor desempenho não eliminou a necessidade de edição manual, o que reforça o argumento de que a IA funciona melhor como uma ferramenta auxiliar.

Os resultados obtidos revelaram que nenhuma plataforma individual se mostrou capaz de produzir, sem erros, animações 2D em *pixel art* que atendessem aos requisitos de consistência e qualidade para aplicação direta em um jogo. A principal descoberta foi a importância de um editor embutido prático, acoplado à ferramenta, que facilite a intervenção manual no ajuste de erros. Também foi demonstrado que um fluxo de trabalho híbrido consegue alcançar resultados melhores em comparação com o uso de uma única IA sem intervenção manual.

A superioridade da qualidade da combinação entre a geração de uma base pela IA e o refinamento manual do artista sugere que o potencial mais significativo da tecnologia reside em sua função como ferramenta auxiliar, e não como substituta. Esta análise argumenta que este não é apenas o estado atual da tecnologia, mas o que deveria ser buscado para sua evolução futura, independentemente do avanço dos modelos. A IA, no contexto da animação, atinge seu maior potencial quando se integra ao ambiente do artista como um instrumento avançado, análogo a outras ferramentas que revolucionaram a arte digital. Assim como o balde de tinta acelera o processo de pintura e um estabilizador aprimora o traço, a IA deveria servir para automatizar as etapas mais trabalhosas e técnicas do processo de animação, como a geração de quadros de transição entre pontos-chave para um movimento mais fluído, ou a criação da base de um personagem em diferentes ângulos.

Nesse sentido, abordagens não-generativas, como a observada na ferramenta *Animated Drawings*, que trabalha sobre um esqueleto a partir de uma arte já criada pelo humano, exemplificam uma filosofia de colaboração mais alinhada a este ideal. Elas não criam do zero, mas sim aceleram uma etapa técnica, deixando o controle criativo e a identidade visual nas mãos do artista sem risco de mudar alguma característica do personagem. Outra possibilidade seria uma abordagem generativa mais especializada para receber um objeto de referência a ser animado, sem gerar inconsistências. O futuro mais promissor para a animação 2D com IA talvez resida em ferramentas que se integrem de forma inteligente ao seu processo, entendendo

e preservando elementos como as camadas de desenho e permitindo uma edição fácil e granular, solidificando o papel da IA como uma ferramenta avançada para o animador.

Em relação às contribuições, este estudo fornece uma análise do estado atual e dos avanços da IA em relação a animação 2D, revelando a eficácia de um fluxo de trabalho cujo resultado oferecido pela ferramenta serve como base para ajustes finos. Entretanto, algumas limitações foram identificadas durante o projeto. A análise das plataformas de IA foi restrita às suas versões gratuitas e aos limites de crédito impostos, o que impediu uma exploração exaustiva de todas as funcionalidades e um processo iterativo mais longo em algumas das ferramentas mais promissoras. Além disso, essa investigação também foi intencionalmente focada em plataformas de alto nível, acessíveis a pessoas sem conhecimento aprofundado em programação. Esta escolha metodológica alinha-se com o objetivo de avaliar a IA como uma ferramenta para o artista.

Devido às restrições de tempo, não foi possível implementar a totalidade das funcionalidades e da narrativa concebida no GDD. O produto final é um protótipo jogável, mas que não representa a experiência completa. A execução do projeto envolveu uma curva de aprendizado, tanto nas especificidades do desenvolvimento 2D na Unity quanto na própria técnica de *pixel art*. Isso demandou um tempo de estudo e pesquisa de referências para a produção dos *assets*. A necessidade de criar toda a arte visual do zero foi uma decisão motivada tanto pela ausência de modelos gratuitos com licenças que permitissem o uso em treinamento de IA quanto pelo alinhamento com a tese do trabalho, de que a IA deve servir como um auxiliar no processo criativo do artista e não como uma substituta completa da criação de arte.

6.2 Trabalhos Futuros

Em trabalhos futuros, é interessante finalizar o desenvolvimento do jogo, incluindo a implementação completa do sistema de diálogos, das mecânicas que faltam e dos cenários não integrados.

Em relação ao uso de IA na animação, seria promissora a concepção e prototipação de um software de animação 2D integrado. Esta ferramenta funcionaria como um aplicativo de animação digital padrão, porém com funcionalidades de IA incorporadas como assistentes para otimizar etapas de produção, como por exemplo, a geração da animação de movimentos cíclicos baseada no esqueleto do personagem. Esta junção permitiria ao animador manter o controle criativo total, usando a IA apenas para acelerar as etapas técnicas mais trabalhosas.

Referências

- ABT, C. *Serious Games*. University Press of America, 1987. ISBN 9780819161482. Disponível em: <<https://books.google.com.br/books?id=axUs9HA-hF8C>>. Acesso em: 12 out. 2025.
- BLACK FOREST LABS. 2025. Disponível em: <<https://bfl.ai/models/flux-kontext>>. Acesso em: 10 set. 2025.
- BYTEPLUS. 2025. Disponível em: <<https://docs.byteplus.com/en/docs/ModelArk/1159178>>. Acesso em: 10 set. 2025.
- CGDREAM AI IMAGE GENERATOR. 2025. Disponível em: <<https://cgdream.ai/gallery>>. Acesso em: 24 ago. 2025.
- DEWHIRST, A.; LAUGHARNE, R.; SHANKAR, R. Therapeutic use of serious games in mental health: scoping review. *BJPsych Open*, v. 8, n. 2, p. e37, 2022.
- DUNLAP KELLI ; KOWERT, R. Mental health in 3d: A dimensional model of mental illness representation in digital games. *Loading*, Canadian Game Studies Association, v. 14, n. 24, p. 122–133, 2021.
- GALVÃO, V. F.; MACIEL, C.; NUNES, E. P. D. S.; RODRIGUES, K. R. D. H. A framework to support the development of empathic games. In: *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2024. (IHC '23). ISBN 9798400717154. Disponível em: <<https://doi.org/10.1145/3638067.3638104>>. Acesso em: 10 ago. 2025.
- GOOGLE. *Gemini*. 2025. Disponível em: <<https://gemini.google.com/app>>. Acesso em: 10 set. 2025.
- GREENBERG, O. Demystifying flux architecture. *arXiv.org*, 2025. Disponível em: <<https://arxiv.org/pdf/2507.09595>>. Acesso em: 25 ago. 2025.
- HAFELL, A. 2025. Disponível em: <https://www.youtube.com/watch?v=g1BCAJqxjog&ab_channel=AIAndy>. Acesso em: 26 ago. 2025.
- HALPERN, J. *Developing 2D Games with Unity: Independent Game Programming with C*. Apress, 2019. ISBN 978-1-4842-3771-7. Disponível em: <<https://doi.org/10.1007/978-1-4842-3772-4>>. Acesso em: 12 out. 2025.
- HEYMANN, J.; SPRAGUE, A. Meeting the un sustainable development goals for mental health: why greater prioritization and adequately tracking progress are critical. *World Psychiatry*, Wiley, v. 22, n. 2, p. 325–326, May 2023. Disponível em: <<https://pmc.ncbi.nlm.nih.gov/articles/PMC10168142/>>. Acesso em: 12 ago. 2025.
- HUANG, Z. Overview of the application of artificial intelligence in computer animation. *Applied and Computational Engineering*, EWA Publishing, v. 40, n. 1, p. 1–6, Feb 2024. Disponível em: <<https://direct.ewa.pub/proceedings/ace/article/view/10215>>. Acesso em: 16 out. 2025.

JORAPUR, A. The evolution of artificial intelligence in gaming. *ART 108: Introduction to Games Studies*, 2024. Disponível em: <<https://scholarworks.sjsu.edu/art108/47/>>. Acesso em: 12 ago. 2025.

KASDORF, R. Representation of mental illness in video games beyond stigmatization. *Frontiers in Human Dynamics*, Frontiers Media SA, v. 5, Sep 2023. Disponível em: <<https://www.frontiersin.org/journals/human-dynamics/articles/10.3389/fhumd.2023.1155821/full>>. Acesso em: 12 out. 2025.

KAVUKCUOGLU, K. *Gemini 2.5: Our most intelligent AI model*. 2025. Disponível em: <<https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking>>. Acesso em: 21 set. 2025.

LI, G. *2D Game Animation Generation: All you need is repeat the same motion 1000 times*. 2024. Disponível em: <<https://github.com/lyogavin/godmodeanimation/>>. Acesso em: 28 ago. 2025.

LI, Z. Research on 2d animation simulation based on artificial intelligence and biomechanical modeling. *EAI Endorsed Transactions on Pervasive Health and Technology*, 05 2024.

MARCOLINO, A. da S.; REIS, H. M.; AVELLAR, G. M. N.; COSTICHE, R. H. S.; OLIVEIRA, T. da S.; BARBOZA, G. S. M. S. Ensino de computação por meio de oficinas de desenvolvimento de jogos 2d: Relatos de experiências de estudantes de licenciatura em computação. *Sbc.org.br*, p. 333–346, Apr 2025. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/34209>>. Acesso em: 12 out. 2025.

MEENA, R.; JINGAR, M.; GUPTA, S. Artificial intelligence: A digital transformation tool in entertainment and media industry. *Our Heritage*, p. 4661, 01 2020.

MENEFEE, D. S.; LEDOUX, T.; JOHNSTON, C. A. The importance of emotional regulation in mental health. *American Journal of Lifestyle Medicine*, v. 16, n. 1, p. 28–31, 2022. PMID: 35185423. Disponível em: <<https://doi.org/10.1177/15598276211049771>>. Acesso em: 12 ago. 2025.

OPENAI. 2025. Disponível em: <<https://platform.openai.com/docs/quickstart>>. Acesso em: 8 set. 2025.

OPENAI. 2025. Disponível em: <<https://platform.openai.com/docs/guides/images-vision?api-mode=responses>>. Acesso em: 8 set. 2025.

OPENAI. 2025. Disponível em: <<https://openai.com/index/introducing-4o-image-generation/>>. Acesso em: 8 set. 2025.

OPENAI. *ChatGPT*. 2025. Disponível em: <<https://chatgpt.com/>>. Acesso em: 27 ago. 2025.

OPENART.AI. 2025. Disponível em: <<https://openart.ai/home>>. Acesso em: 8 set. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. *Mental disorders*. OMS, 2022. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/mental-disorders>>. Acesso em: 12 ago. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. *Mental health*. OMS, 2022. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response>>. Acesso em: 12 ago. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. *World Mental Health Report*. OMS, 2022. Disponível em: <<https://www.who.int/teams/mental-health-and-substance-use/world-mental-health-report>>. Acesso em: 12 ago. 2025.

PICHAJ, S. *Introducing Gemini: our largest and most capable AI model*. 2023. Disponível em: <https://blog.google/technology/ai/google-gemini-ai/?utm_source=chatgpt.com#performance>. Acesso em: 10 set. 2025.

PIXELLAB.AI. 2025. Disponível em: <<https://www.pixellab.ai/editor>>. Acesso em: 27 ago. 2025.

PRAKASH, D. 2d game development using unity engine “minotscape”. *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, v. 07, 08 2023. Disponível em: <<https://doi.org/10.55041/IJSREM25405>>. Acesso em: 12 out. 2025.

QIU, S. Generative ai processes for 2d platformer game character design and animation. *Lecture Notes in Education Psychology and Public Media*, v. 29, p. 146–160, 12 2023. Disponível em: <<https://doi.org/10.54254/2753-7048/29/20231440>>. Acesso em: 12 out. 2025.

REFACTORING.GURU. 2024. Disponível em: <<https://refactoring.guru/pt-br/design-patterns/memento>>. Acesso em: 12 out. 2025.

ROMA, P. F. Psicofobia: lidando com transtornos mentais e o estigma no brasil. *Debates em Psiquiatria*, v. 14, p. 1–5, nov. 2024. Disponível em: <<https://revistardp.org.br/revista/article/view/1315>>. Acesso em: 12 ago. 2025.

ROSEBUD AI. 2025. Disponível em: <<https://lab.rosebud.ai/sprites-animation>>. Acesso em: 24 ago. 2025.

SCHIJVEN, M. P.; KIKKAWA, T. Is there any (artificial) intelligence in gaming? *Simulation & Gaming*, v. 53, n. 4, p. 315–316, 2022. Disponível em: <<https://doi.org/10.1177/10468781221101685>>. Acesso em: 12 ago. 2025.

SEGMIND. *AI Sprite Sheet Maker*. 2025. Disponível em: <<https://www.segmind.com/pixelflows/67ee10c0a86c3ba664125013>>. Acesso em: 24 ago. 2025.

SMITH, H. J.; ZHENG, Q.; LI, Y.; JAIN, S.; HODGINS, J. K. A method for animating children’s drawings of the human figure. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 42, n. 3, jun 2023. ISSN 0730-0301. Disponível em: <<https://doi.org/10.1145/3592788>>. Acesso em: 27 ago. 2025.

SMITH, J. *Animated drawings*. 2024. Disponível em: <<https://sketch.metademolab.com/canvas>>. Acesso em: 24 ago. 2025.

TURNER, I. *Adaptable Object and Animation System for Game Development Adaptable Object and Animation System for Game Development*. [s.n.], 2023. Disponível em: <<https://digitalcommons.wku.edu/cgi/viewcontent.cgi?article=4698&context=theses>>. Acesso em: 12 out. 2025.

UNITY TECHNOLOGIES. 2019. Disponível em: <<https://learn.unity.com/tutorial/introduction-to-sprite-animations>>. Acesso em: 1 set. 2025.

URSATJEW, S. *Pixie.haus*. 2025. Disponível em: <<https://pixie.haus/>>. Acesso em: 26 ago. 2025.

VERHALEN, A. E. C.; RODRIGUES, K. R. D. H. Design of therapeutic digital games that support dialogue with children about death. In: *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2024. (IHC '23). ISBN 9798400717154. Disponível em: <<https://doi.org/10.1145/3638067.3638112>>. Acesso em: 12 out. 2025.

VIDU. Vidu AI, 2024. Disponível em: <<https://www.vidu.com/create/character2video>>. Acesso em: 26 ago. 2025.

VIDU. *First & Last Frame*. 2025. Disponível em: <<https://shengshu.feishu.cn/wiki/NwCEwsdltiPjRakIBFPcvjpanIT?fromScene=spaceOverview>>. Acesso em: 20 set. 2025.

VIDU. *Vidu Prompt User Guide*. 2025. Disponível em: <<https://shengshu.feishu.cn/docx/MtKHd8OhNokOtaxzxQHcdUDPnKh>>. Acesso em: 18 set. 2025.

WOODWARD, J. *Gemini gets more personal, proactive and powerful*. 2025. Disponível em: <<https://blog.google/products/gemini/gemini-app-updates-io-2025/#gemini-live>>. Acesso em: 21 set. 2025.

XIAN, D.; SAHAGUN, J. An automated generation from video to 3d character animation using artificial intelligence and pose estimate. *Artificial Intelligence Advances*. <https://doi.org/10.5121/csit>, 2023. Disponível em: <<https://csitcp.org/paper/13/137csit03.pdf>>. Acesso em: 12 ago. 2025.

XU, T.; COHEN, M. C.; CLARK, J. Generative ai models for images with copyright-free training. Jun 2025. Disponível em: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5331280>. Acesso em: 25 ago. 2025.

YESCHAT AI. *SpriteSheetGPT - AI-Powered Sprite Animation*. 2024. Disponível em: <<https://www.yeschat.ai/gpts-ZxWyYju4-SpriteSheetGPT>>. Acesso em: 24 ago. 2025.

Apêndices

APÊNDICE A – Implementação e Lógica de Programação das Mecânicas

A.1 Sistema de Gerenciamento e Configurações do Jogo

Para o controle de variáveis globais e configurações do jogo, foram implementadas duas classes. O *script VarGlobal*, anexado ao *GameManager*, é responsável por inicializar as variáveis estáticas e garantir que objetos essenciais não sejam destruídos durante a transição de cenas utilizando a funcionalidade *DontDestroyOnLoad* (não destrói ao carregar, em inglês). Ele também gerencia a destruição de objetos duplicados que possam surgir ao recarregar uma cena.

A classe estática *GameSettings* atua como um ponto de acesso central para as configurações e referências principais do jogo (como o personagem, a câmera e o cômodo atual), agilizando a comunicação entre os diferentes *scripts*. É esta classe que gerencia a lógica de troca de cômodos, atualizando o *currentPlace* (local atual, em inglês) e configurando a câmera, a interface, as referências de chão e paredes.

A.2 Sistema de Controle do Personagem e Animação

O controle do personagem foi implementado utilizando o novo sistema de entradas (Input System, em inglês) do Unity, que permite o mapeamento de ações associadas a um conjunto de teclas para diferentes dispositivos de entrada. Foram criadas e implementadas para o projeto as ações:

- *Move* (em inglês, mover), ligada às teclas A, D e as setas para a esquerda e para a direita;
- *Jump* (em inglês, pular), conectada ao botão de espaço;
- *Interact* (em inglês, interagir), relacionada ao W e à seta para cima;
- *Back* (em inglês, voltar), ligada ao S e à seta para baixo; e
- *Escape* (em inglês, escapar), ligada à tecla esc.

O componente *Player Input* (entrada do jogador, em inglês) é responsável por acionar os eventos determinados quando uma ação específica é realizada, enviando informações de contexto sobre a ação, como por exemplo o valor e o estado.

O *script* MC (personagem principal, em inglês *main character*) controla o personagem principal. Esta classe recebe os eventos do Player Input e, com base no estado atual do jogo, delega as tarefas para os componentes especializados. Ademais, ela possui funções relacionadas ao controle e ao estado do personagem principal. Por exemplo, ao receber o evento de pulo, ela verifica se o personagem está no chão e, em caso afirmativo, comanda os componentes de movimento e de animação. Além disso, checa-se de forma contínua se o personagem voltou a entrar em contato com o chão, e quando isso ocorre, o fim da animação é encarregado ao *script* específico.

A classe *AnimatorAnimation* (animação do animador em inglês) é responsável por interagir com o componente Animator (controlador das animações) de um objeto. Esse *script* aciona gatilhos e muda os valores das variáveis para iniciar animações, além de detectar quando uma animação acaba.

A classe *Movement2D* (em inglês, movimento 2D) é a única classe com permissão para modificar diretamente os componentes *rigid body* (corpo rígido, em inglês) e *transform* (transformação, em inglês) de um *game object*, garantindo que toda a lógica de movimentação esteja centralizada. Quando uma de suas funções é chamada, realiza no objeto ao qual está acoplada o movimento requisitado.

A.3 Sistema da Câmera

Para a câmera que segue o personagem, foi implementada a classe *MovedBy* (movido por, em inglês), que é descendente de *Movement2D*. Ela move o objeto ao qual está acoplada de acordo com o deslocamento do objeto fonte (neste caso, o personagem principal), respeitando os limites de borda estabelecidos para o cômodo atual (exceto em casos onde a câmera já estava fora da borda, situações nas quais ela apenas segue o movimento da fonte até voltar para dentro do limite).

A borda é uma linha imaginária que traça a distância mínima à qual a câmera deve estar de uma barreira (horizontal ou vertical). No jogo "O Monstro" só são consideradas as barreiras do cômodo atual em que o personagem está. Código 1, é apresentada a função correspondente ao cálculo da distância que a câmera deve percorrer, dependendo da fonte estar fora da borda ou não.

Em resumo, existem tipos específicos de casos em que a distância é calculada de forma diferente:

- Se o personagem principal estava dentro da borda e continuou dentro dela, a câmera se move a mesma distância que o personagem;
- Se o personagem principal estava fora da borda e se movimentou para dentro, a câmera move a distância que existe entre ela e o personagem;

Código 1 – Código comentado da função que calcula o movimento que a câmera deve realizar

```

1 //CALCULA MOVIMENTO DO OBJETO -----
2
3 float CheckMove(Collider2D[] barrier, float dist, float minDist)
4 {
5     float newDist, distBarrier;
6     if ((distBarrier = MoveAllowed(barrier, minDist)) == 0f)
7         //0 = nao tem barreira ou fonte esta mais distante que
8         //distancia minima de todas as barreiras ou camera tambem
9         //esta alem da distancia minima
10    {
11        if (outBound == true) //se moveu saindo de alem da
12            //distancia minima
13        {
14            outBound = false; //nao esta alem da distancia
15            //minima
16            return GetX(source) - GetX(transform); //objeto move
17            //distancia que estava da borda
18        }
19        return dist; //move mesmo tanto que fonte
20    }
21
22    //calcular movimento
23    int sign = (dist > 0) ? 1 : -1; //se move para direita,
24    //sentido positivo, se move para esquerda negativo
25    newDist = sign * dist + distBarrier - minDist; //( quanto
26    //fonte se moveu + distancia fonte da parede ) - distancia
27    //minima
28    newDist *= sign; //deixar no mesmo sentido
29    //dist pode ser negativo
30    //new dist pode ser sinal oposto
31    return (newDist * dist > 0) ? newDist : 0;
32 }

```

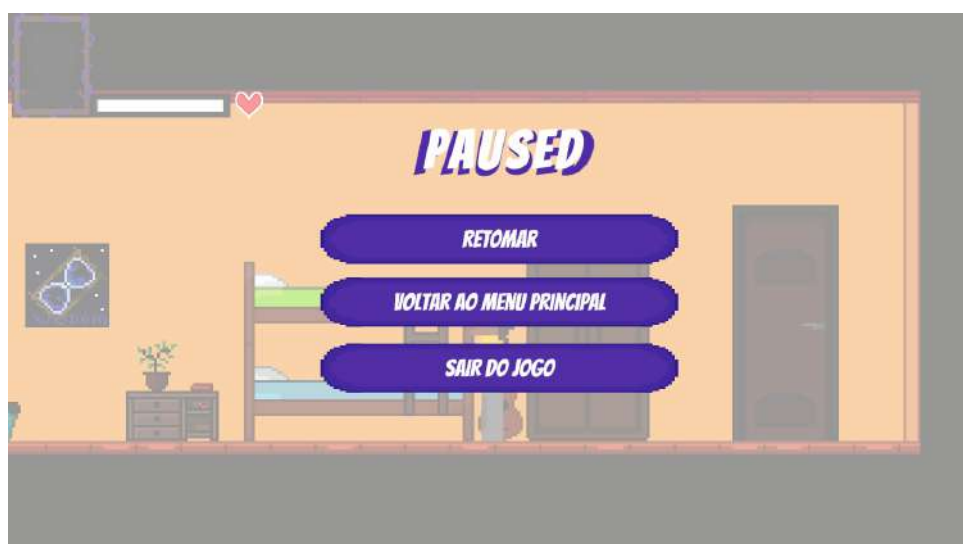
- Se o personagem principal estava dentro da borda e se movimentou para fora, a câmera se move a distância que existe entre ela e a borda¹;
- Se o personagem principal e a câmera estavam fora da borda e o personagem continuou fora dela, a câmera se move a mesma distância que o personagem; e
- Se apenas o personagem principal estava fora da borda e se manteve fora dela, a câmera não se move.

¹ Calculada pela soma da distância personagem-câmera com a distância personagem-barreira, subtraída pela distância mínima.

A.4 Sistema de Menu

Foi criado um sistema de menus genéricos e escaláveis. Foi implementada a classe Menu, que funciona de maneira simples: ao entrar no menu o jogo é pausado, qualquer outro menu previamente aberto é salvo na variável `prevMenu` e este menu é colocado como menu ativo; e ao sair do menu, o jogo é despausado caso não exista nada em `prevMenu`, `prevMenu` é colocado como menu ativo, e este menu é desligado. Foram implementados o menu de pausa e o menu de interação, que não podem ser abertos ao mesmo tempo, fazendo com que a funcionalidade do menu anterior ainda não tenha sido testada. A Figura 25 mostra a tela do menu de pausa aberta.

Figura 25 – Menu de pausa durante a execução do jogo



Fonte: Elaborada pela autora.

A.5 Sistema de Interação com Objetos

Para permitir que o jogador interaja com o cenário, foi criado um sistema de interação genérico e expansível. A classe `Object` é a base para qualquer tipo de objeto interativo do jogo, conectada a um menu de interação e possuindo uma função `Interact` (interagir em inglês) que é chamada quando ocorre uma interação com o objeto. O menu de interação foi implementado por meio da classe `InteractMenu` (menu de interagir em inglês), que é descendente de `Menu` e controla as opções de interação. Todas as instâncias dessa classe apontam para o mesmo game object como menu de opções, porém, dependendo do objeto ao qual está associada, ela modifica os botões que aparecem.

O *script Interaction* (interação em inglês) é anexado a todos os objetos interativos para detectar quando o ponteiro do personagem principal está apontando para esse objeto através do colisor (em inglês, *collider*). Basicamente, quando o ponteiro entra na área do colisor do

objeto, este é colocado como alvo (target, em inglês) de interação para o jogador; e quando o ponteiro sai dessa área, o objeto é removido do target. O Código 2 apresenta o código desse processo.

Código 2 – Código comentado da classe que detecta se um objeto é alvo de interação

```

1 public class Interaction : MonoBehaviour
2 {
3
4     //----- DETECTAR QUANDO PLAYER ESTA NA FRENTE DO OBJETO
5     -----
6
7     //COLOCA ESSE OBJETO COMO ALVO INTERACT SE PLAYER ESTIVER NA
8     FRENTE
9
10    void OnTriggerEnter2D(Collider2D other)           //se collider
11        com rigidbody entra nesse collider
12    {
13        if (other.gameObject != player)           //se nao for
14            collider principal do player (tem que ser collider de
15            ponteiro)
16        {
17            scriptPlayer.Target = gameObject;     //target
18            player setado como esse objeto
19        }
20    }
21
22    void OnTriggerExit2D(Collider2D other)           //se collider
23        com rigidbody sai desse collider
24    {
25        if (other.gameObject != player)           //se nao for
26            collider principal do player (tem que ser collider de
27            ponteiro)
28        {
29            scriptPlayer.Target = null;           //tira esse
30            objeto de target player
31        }
32    }
33 }

```

A.6 Sistema de Portas e entradas

O sistema de portas e entradas foi implementado utilizando as classes *Door* (porta, em inglês) e *SideDoor* (porta lateral, em inglês) para a animação de abrir e fechar a porta, e a classe *Portal* para transportar o jogador para o outro cômodo, se necessário. Este sistema

gerencia os seguintes tipos de objetos: porta frontal; porta lateral; entrada frontal; e entrada lateral.

A classe *Door* anexada em portais frontais é descendente de *Object* e é responsável pela lógica de abrir e fechar a porta, acionando a transição de local, se aplicável. Enquanto *SideDoor* herda de *Door*, ela é responsável pela lógica de abrir e fechar a porta automaticamente pela proximidade do player, e é anexada em portas laterais.

O *script* *Portal* possui um destino e a posição do personagem no novo lugar, controlando a lógica de transição (na mesma cena ou em uma diferente). Em caso do destino ser em outra cena, as informações são guardadas na classe estática *TransitionData* (dados de transição em inglês) e a mudança de cena é iniciada. *Portal* pode ser ativado pelo contato com o personagem quando anexado em entradas ou algumas portas laterais.

Outra classe relevante de ser mencionada é a *SetPlace* (definir lugar, em inglês), que aciona o aviso de mudança de cômodo quando o jogador encosta no objeto ao qual está anexado. Isso é necessário pois algumas portas laterais não precisam da classe portal, porém ainda necessitam acionar a troca de cômodo.

A.7 Sistema de neblina

Para não revelar o ambiente inteiro para o jogador antes da hora e manter um foco maior no cômodo que o personagem principal está, foi implementado um sistema de visibilidade que utiliza uma *fog*, composta por *tilemaps* que cobrem o cômodo com *tiles* pretos. Existe uma neblina permanente (Figura 26) e uma neblina temporária que é controlada pela classe *FogControl* (controle de neblina, em inglês). Esse *script* está anexado às portas laterais que levam a um cômodo na mesma cena, sendo capaz de controlar a neblina do destino (apenas um).

Figura 26 – Neblina permanente na borda do cômodo



Fonte: Elaborada pela autora.

Mesmo que em teoria o destino da porta lateral depende de onde o personagem está, de maneira não convencional foram criados dois objetos para a porta lateral, cada um tendo

um destino diferente e apenas um deles podendo estar ativado durante a execução. Isso foi feito por diversos motivos, tais como a hierarquia da cena mantinha todos os objetos de jogo relacionados a um cômodo como pertencentes ao grupo dele, não sendo possível definir onde a porta lateral ficaria se ela pertencesse a dois diferentes.

Para lidar com esses objetos gêmeos, implementou-se a classe *Twin* (gêmeo, em inglês), que quando chamada pelo *FogControl* desativa o objeto ao qual está anexada, ativa o objeto gêmeo e aciona o mesmo a ativar a neblina.

A.8 Sistema de saúde mental

Numericamente, o valor atual e máximo da saúde mental é armazenado na classe MC já mencionada. Porém o jogador só obterá a informação visual da barra de vida, que utiliza o *rect transform* (transformação retangular), configurada para se esticar horizontalmente para preencher um contêiner (fundo da barra). Vale ressaltar que o controle da sua largura é feito por meio do ajuste de seus deslocamentos (offsets) laterais em relação às bordas do contêiner. Para simular a perda de vida, o valor do deslocamento da borda direita (right offset) é aumentado, o que faz a barra diminuir de tamanho.

A classe *Life* (vida, em inglês) é responsável pelo controle dessa barra de vida, definindo o tamanho dela considerando sua vida atual e sua vida máxima através de uma regra de três simples com o comprimento máximo da barra e o cálculo do deslocamento necessário para o novo tamanho. O *script Damage* (dano, em inglês) é anexado a qualquer entidade ou objeto capaz de causar dano ao personagem principal. Ele controla a lógica de causar dano, acionando a perda de vida e enviando a quantidade de dano ao atingir as condições para causar esse prejuízo.

A.9 Sistema de mudança de cena

A classe *SceneTransitionManager* (controlador da transição de cena, em inglês) é responsável por inicializar o sistema de salvamento, chamar o carregamento de dados, criar e acionar um objeto portal com os dados em *TransitionData*, iniciar o salvamento de dados, e começar a mudança de cena.

A classe estática *SaveSystem* (salvar sistema) é responsável por salvar e carregar os dados das cenas, que são do tipo *SceneMemento* (recordação de cena, em inglês). O *SceneMemento* consegue armazenar o estado da cena em seus *snapshots* (foto instantânea, em inglês). Não é possível acessar os atributos desse memento, porém a classe *SceneOriginator* (originador da cena, em inglês) consegue restaurar e criar o *snapshot* do *SceneMemento*. Esse design é conhecido como memento, que permite salvar o estado de um objeto sem expor a sua implementação (REFACTORING.GURU, 2024).

A.10 Sistema de plataforma

Foi implementado um sistema de colisão unidirecional nas plataformas, cuja colisão ocorre apenas quando o jogador cai do seu topo. Isso foi feito para o teto de uma plataforma não travar o jogador quando ele tenta pular para outra e para a altura do pulo não precisar ser irreal. O *script PlatformCollision* controla o colisor do objeto ao qual está anexado, desativando-o se a altura do pé do jogador for menor que a do topo da plataforma e ativando-o caso contrário.

APÊNDICE B – Análise detalhada das Ferramentas de IA

B.1 Análise do RosebudAI

A ferramenta RosebudAI foi selecionada por demonstrar foco na criação de *sprite sheets*, especificamente para jogos. Na sua página inicial, afirmações como "Use IA para criar *sprites* para seu jogo" apontavam para a capacidade da plataforma em criar animações para um personagem. No entanto, a análise revelou uma ferramenta com múltiplas funcionalidades que, em todos os testes, falhou em produzir um *sprite sheet* 2D consistente a partir de uma imagem de referência.

Os testes foram realizados em junho de 2025, focando no ambiente principal da plataforma. O objetivo era produzir o *sprite sheet* ou animação do *walking cycle* (ciclo de caminhada, em inglês) do personagem utilizando o *sprite* de Pablo em *front view*. Na primeira tentativa, em vez de um *sprite sheet*, a ferramenta apresentou um protótipo de jogo 3D, incluindo um *script* de 600 linhas de código. O personagem gerado manteve vagamente as cores da referência, porém com um estilo cúbico inadequado, em uma aparente tentativa de emular o estilo *pixel art* em um ambiente tridimensional.

Em uma tentativa subsequente, com um prompt ajustado para especificar um cenário 2D e manter a consistência do personagem, a ferramenta produziu uma animação de baixa qualidade¹, na qual metade da imagem de referência é apenas deslocada horizontalmente, alternando entre a parte inferior ou superior visível na tela, como é demonstrado na Figura 27. A análise dos quadros revelou que a IA interpretou a imagem de referência como se fosse um *sprite sheet* completo de dois quadros, dividindo-a ao meio e alternando entre as metades superior e inferior.

Diante desse resultado, os prompts foram ajustados para especificar a criação de um *sprite sheet* do personagem em várias posições diferentes. Após algumas interações sem sucesso no ambiente principal, a análise foi direcionada para uma seção separada dedicada à geração de *assets*. O primeiro teste nesta seção resultou na geração de um *sprite* único que desconsiderou completamente a imagem de referência, criando um personagem novo em um estilo distinto.

Além disso, foi gerado apenas um *sprite* em vez do *sprite sheet* do personagem andando. A tentativa de refinar os prompts, utilizando a IA principal para descobrir como referenciar a imagem corretamente, também levou a resultados insatisfatórios. Conforme demonstrado na

¹ <<https://drive.google.com/file/d/1yPtpKDM2CYCaxSqFJr3NFbVswduzkFTY/view?usp=sharing>>

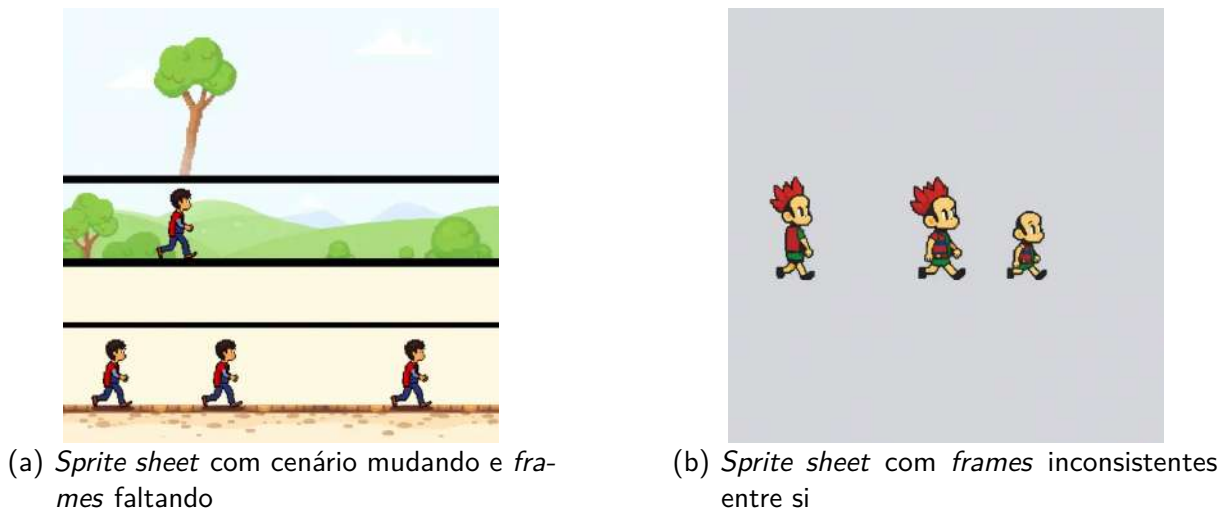
Figura 27 – Animação gerada pelo Rosebud AI



Fonte: Elaborada pela autora.

Figura 28, os *sprite sheets* gerados apresentaram falhas graves, como a mudança do cenário e inconsistência entre os quadros, além de ainda desconsiderar a referência.

Figura 28 – Resultados finais



Fonte: Elaborada pela autora.

Considerando que nenhuma das abordagens produziu um resultado satisfatório, a ferramenta foi considerada inadequada para o projeto. Embora não seja viável para a criação de animações 2D personalizadas, a plataforma demonstra potencial para a prototipagem rápida de jogos simples para usuários que não possuem conhecimento em programação.

B.2 Análise do SpriteSheetGPT

A ferramenta SpriteSheetGPT, encontrada no site YesChat.AI, à primeira vista parecia promissora por seu aparente foco na criação de *sprite sheets*, como seu nome sugere. O objetivo principal do teste era produzir o *sprite sheet* do *walking cycle* do personagem utilizando o *sprite* de Pablo em *front view*.

Os primeiros testes realizados com esta ferramenta não foram satisfatórios, pois os resultados foram apenas textuais. Após muita insistência, a ferramenta gerou uma imagem. O artefato resultante (Figura 29) demonstrou baixa consistência com a imagem de referência, tendo cores apenas vagamente semelhantes. Além disso, apresentou baixa precisão com o prompt, sem representar o personagem andando, e reproduziu apenas parcialmente o estilo de *pixel art*. O único ponto positivo é que foi formado um *sprite sheet* de um personagem em 2D, conforme esperado.

Figura 29 – *Sprite sheet* gerado pelo SpriteSheetGPT



Fonte: Elaborada pela autora.

Ao tentar replicar a geração com os mesmos prompt e imagem de referência, novamente a ferramenta gerou resultados textuais. Portanto, mesmo após conseguir produzir uma imagem, a conclusão da análise é que a ferramenta se mostra extremamente instável para qualquer aplicação prática no momento. A incapacidade de reproduzir um resultado preciso em relação ao prompt e consistente com a imagem de referência, além da ocasional falta de resultado visual, são falhas críticas.

Apesar do descarte da ferramenta, seu comportamento destaca um aspecto fundamental dos modelos de IA generativa: a variabilidade de resultados para um mesmo *input*. Essa aleatoriedade (controlada pela semente de geração) demonstra a importância de que o usuário realize múltiplas iterações da mesma instrução para alcançar um resultado satisfatório. Em modelos com uso gratuito limitado, essa necessidade de re-geração torna o sucesso dependente da sorte, uma vez que o resultado desejado pode não ser alcançado dentro do limite de tentativas disponíveis.

B.3 Análise do CGDream

A ferramenta CGDream foi selecionada por sua capacidade de gerar imagens a partir de uma referência e uma descrição textual, com o objetivo de criar uma imagem do personagem Pablo em *side view* a partir de sua arte em *front view*. A lógica por trás disso é que essa imagem possa ser usada também de referência para a animação. A plataforma se destaca pela vasta gama de opções de customização, porém apresenta uma interface visualmente poluída, o que pode dificultar a localização de suas funcionalidades. A ferramenta permite o envio de uma referência e oferece múltiplos modos de uso para essa imagem, tais como:

- Estilo de referência: manter o estilo;
- Estrutura de referência: pegar uma estrutura, como por exemplo construções;

- Imagem como referência: usar uma figura como referência para a geração;
- 3D para imagem: transformar um modelo 3D em uma imagem; e
- Personagem consistente: reconhecer um personagem e usar como guia para a geração.

A plataforma também possui outras opções de customização, como filtros específicos para direcionar a geração, uma variável chamada *prompt guidance* (orientação do prompt) e dois modelos de IA: o Juggernaut XL, ideal para fotorrealismo de acordo com [Xu, Cohen e Clark \(2025\)](#); e o Flux, conhecido por sua alta fidelidade aos prompts ([GREENBERG, 2025](#)) e o mais recomendado para uso ([HAFELL, 2025](#)). A análise foi focada em duas funcionalidades principais: "Imagem como referência" e "Personagem consistente".

Os testes direcionados à funcionalidade "Imagem como referência" apresentaram resultados interessantes, comparando o desempenho dos modelos Flux e Juggernaut XL.

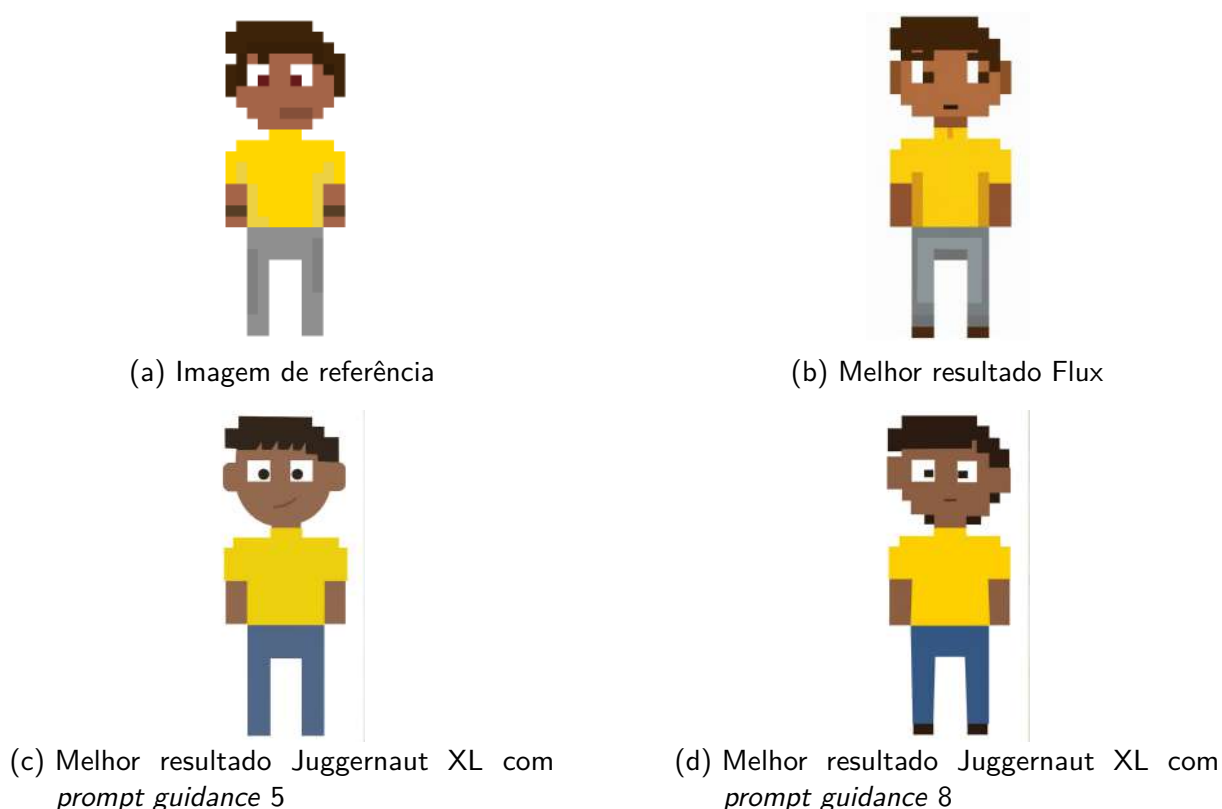
Na primeira interação, foi selecionado o modelo Flux no modo Dev, com o prompt "*boy facing east*" (menino voltado para o leste, em português). Os resultados apresentados demonstraram uma semelhança média com a referência e mantiveram o estilo *pixel art 2D*, porém o personagem foi gerado em *front view*, e em uma das imagens ele apenas movia os olhos para a esquerda. Analisando esses dados, a ferramenta parece ter interpretado o prompt como se o personagem devesse estar olhando para a esquerda com apenas os olhos, sem o corpo estar virado.

Na segunda interação, foi selecionado o modelo Juggernaut XL no modo *Quality*, mantendo o exato mesmo prompt para fins de comparação com o resultado anterior. As imagens geradas mantiveram o ambiente 2D e apresentaram semelhanças médias com a imagem de referência, porém o estilo de *pixel art* não foi incorporado de maneira satisfatória e também ignorou a instrução textual. Por esse motivo, outro teste foi realizado aumentando o valor de *prompt guidance*, o que melhorou a fidelidade do estilo, porém não corrigiu a pose.

A Figura 30 compara o melhor resultado obtido em cada um dos testes. Embora o modelo Flux tenha sido superior na manutenção do estilo, nenhum dos resultados foi consistente o suficiente para uso no jogo, e o objetivo principal (gerar o personagem em *side view*) não foi alcançado. Adicionalmente, nenhuma das imagens geradas atingiu um padrão *pixel perfect*.

Em testes posteriores voltados para o modelo Flux, com prompts ajustados para especificar melhor a pose, a ferramenta não apresentou resultados melhores, continuando a ignorar a instrução textual, apesar de gerar um personagem consistente. Analisando os resultados, é possível perceber que a funcionalidade "Imagem como referência" desenha a nova figura mantendo a semelhança com a figura enviada, mesmo que tenha que ignorar o prompt para isso. Devido a nenhum dos resultados ter sido satisfatório, essa funcionalidade foi descartada.

Figura 30 – Melhores resultados do CGDream utilizando a funcionalidade de imagem



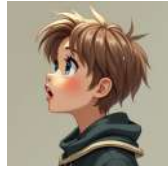
Fonte: Elaborada pela autora.

Os testes direcionados à funcionalidade Personagem Consistente apresentaram resultados insatisfatórios. Seguindo a mesma lógica das interações com a imagem de referência, primeiro foi selecionado o modelo Flux Dev e depois o Juggernaut XL, ambos com o mesmo prompt: "*boy facing east*". A ferramenta ignorou completamente a imagem de referência, gerando personagens, estilos e cenários totalmente novos, apenas cumprindo de maneira parcial a instrução da pose. Mais algumas tentativas foram feitas, ajustando o prompt e usando a funcionalidade de filtro e de palavras negativas (exclusivo do modelo Juggernaut XL, que permite especificar o que é para ser evitado na geração), o que gerou a pose precisa, porém manteve os problemas de consistência (Figuras 31a e 31c).

Uma nova estratégia foi implementada, combinando a funcionalidade Personagem Consistente com a de Estilo de Referência, utilizando a mesma imagem em ambas e visando explorar as outras funcionalidades do site. Além disso, a funcionalidade de filtro também foi usada para especificar a imagem em *pixel art* e o prompt foi ajustado para descrever o personagem, visto que os resultados anteriores não mantiveram nenhuma característica da referência. Essa abordagem se mostrou mais eficaz em manter as características do personagem original, como pode ser observado na Figura 31.

Apesar da melhora na consistência, novos problemas surgiram. O modelo Flux passou a gerar imagens borradas e com a pose de costas. Esse problema ocorre independentemente do

Figura 31 – Comparativo de resultados com e sem Estilo de referência



(a) Imagem gerada com o modelo Flux apenas utilizando personagem de referência



(b) Imagem gerada com o modelo Flux usando personagem e estilo de referência



(c) Imagem gerada com o modelo Juggernaut XL apenas utilizando personagem de referência



(d) Imagem gerada com o modelo Juggernaut XL usando personagem e estilo de referência

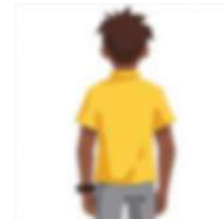
Fonte: Elaborada pela autora.

prompt ou filtro usado, como é demonstrado nas Figuras 32a e 32b. O modelo Juggernaut XL, por sua vez, gerava imagens no estilo realista quando "3D" não era explicitamente utilizada como palavra negativa, indicando que a referência de estilo era usada apenas para capturar as características do personagem e não sua estética *pixel art* (como pode ser observado nas Figuras 32c e 32d).

Figura 32 – Comparativo de resultados do modelo Flux com e sem filtro pixelizado, e do modelo Juggernaut XL com e sem o uso de 3D como palavra negativa



(a) Imagem gerada sem utilizar filtro pixelado



(b) Imagem gerada usando filtro pixelado



(c) Imagem gerada com apenas "blur" como palavra negativa



(d) Imagem gerada com "3D" e "blur" como palavras negativas

Fonte: Elaborada pela autora.

Após uma extensa bateria de testes, conclui-se que a ferramenta CGDream, apesar de seu potencial e complexidade, não é adequada para a tarefa de gerar *sprites* consistentes para

um jogo já em desenvolvimento. Ambas as funcionalidades testadas falharam em um ou mais critérios essenciais: manter o estilo *pixel art*, reproduzir fielmente o design do personagem ou seguir a instrução de pose do prompt. Portanto, a ferramenta foi descartada.

B.4 Análise do Pixie.haus

A análise da Pixie.haus focou em seu potencial como uma ferramenta especializada na geração de imagens e animações em *pixel art*. Uma de suas funcionalidades é o editor de imagens diretamente integrado na plataforma, o que, em tese, possibilitaria a correção rápida de pequenos erros e facilitaria o processo de ajuste fino dos resultados. O objetivo principal do experimento é produzir a animação da porta B do cenário abrindo em uma perspectiva lateral.

Na geração de imagem, existem algumas opções de customização, como a resolução, paleta de cores, remoção do fundo e seleção do modelo de IA, sendo FLUX1.schnell e Luma Photon Flash os mais recomendados. Enquanto isso, na geração de vídeo é possível apenas adicionar uma imagem de referência e escolher o modelo de IA.

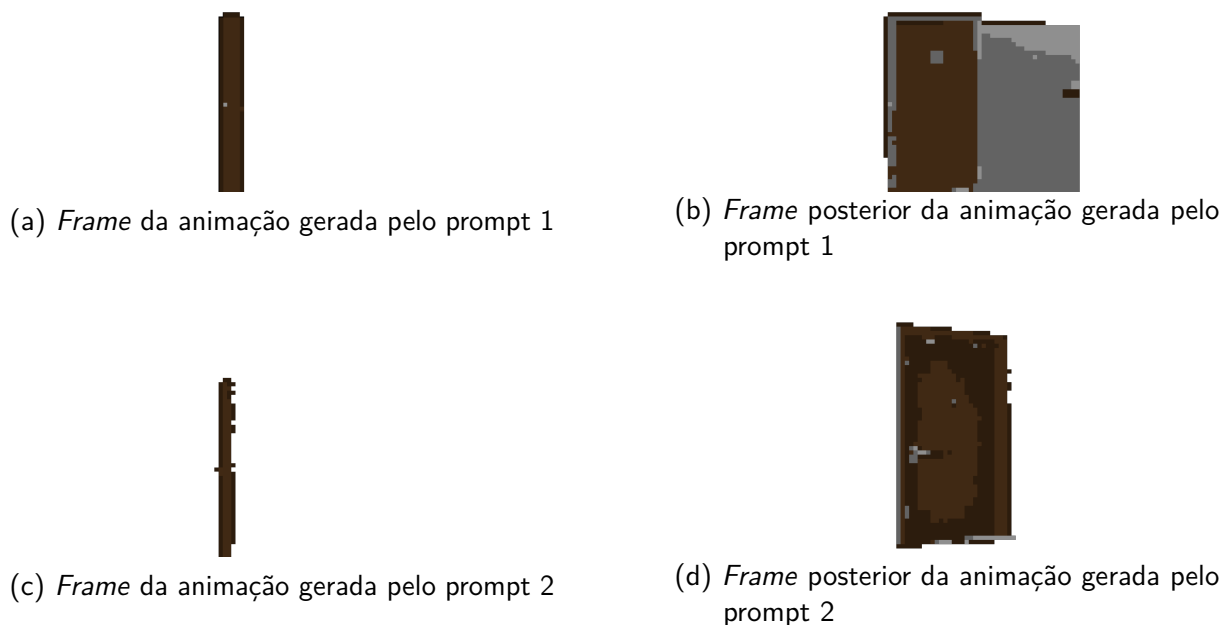
Durante a preparação, foi encontrada uma limitação significativa da ferramenta: a funcionalidade de imagem de referência só aceita artes criadas na plataforma, sem opções para importar *sprites* externos. Isso exigiu que a porta fosse recriada manualmente. O editor, contudo, mostrou-se extremamente fraco, com ausência de ferramentas essenciais como seleção de área, exibição de coordenadas e precisão no preenchimento, tornando o processo pouco eficiente. Além disso, as funcionalidades existentes nesse editor só podiam ser utilizadas através de atalhos, descritos em uma interface separada.

Foram disponibilizados créditos suficientes para a geração de duas animações, utilizando o modelo de texto e imagem para vídeo ByteDance Seedance-1-Lite. Em cada tentativa, foi usado um prompt diferente: o primeiro focando em especificar a ação de maneira direta e com precisão, e o segundo com uma descrição mais técnica sobre como os *frames* deveriam se comportar.

Ambas as tentativas produziram resultados insatisfatórios². O principal problema foi a falha da ferramenta em gerar o movimento de abertura solicitado. No primeiro teste, a IA criou uma animação onde a porta se materializa deslizando horizontalmente antes de abrir incorretamente em *front view*. No segundo, além da porta se materializar da mesma forma que ocorreu no resultado anterior, foi feita uma rotação completa da porta sobre seu eixo central. A IA aparenta ter interpretado a referência inicial como sendo apenas uma parte incompleta da porta. A Figura 33 apresenta *frames* de ambas as animações geradas.

Adicionalmente, foi observada uma falha no pós-processamento de remoção de fundo (além da animação final, a ferramenta também disponibilizou automaticamente o estado do

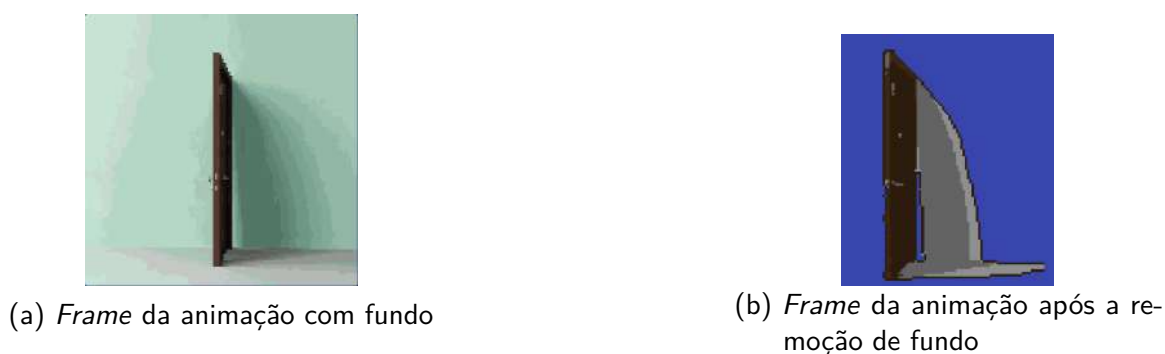
² <<https://drive.google.com/drive/folders/1rnU-l261vEqKgXC7RwA23u1XItUGiDwM?usp=sharing>>

Figura 33 – *Frames* das animações geradas pelo Pixie.Haus

Fonte: Elaborada pela autora.

vídeo antes do fundo ser removido), que manteve a sombra da porta e a deformação do fundo como parte do *sprite*, o que pode ser verificado na Figura 34. Embora os resultados mantivessem o estilo correto e um padrão *pixel perfect*, a imprecisão na interpretação das instruções textuais foi o problema mais crítico.

Figura 34 – Inclusão incorreta da sombra



Fonte: Elaborada pela autora.

Como a análise da Pixie.haus foi restringida pelo seu limitado modelo de créditos gratuitos, não foi possível testar outros modelos e prompts. Com base nos poucos testes realizados, a ferramenta foi descartada da fase final da análise, pois as animações geradas mostraram-se imprecisas e inadequadas para o objetivo do projeto. Apesar disso, a plataforma apresenta um conceito promissor, mas com falhas na execução. Embora a integração de um

editor de *pixel art* tenha como objetivo a praticidade, sua implementação atual não é adequada para edições complexas e não permite editar os gifs finais gerados.

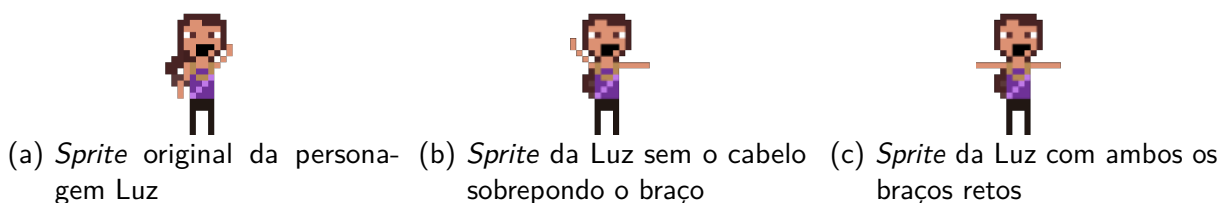
No entanto, o principal diferencial da ferramenta é sua capacidade de gerar resultados *pixel perfect*. Este padrão é alcançado através de uma série de etapas de pós-processamento, que incluem o redimensionamento pelo método do *nearest neighbor* (vizinho mais próximo, em inglês) para manter as bordas mais nítidas, o ajuste da paleta de cores para garantir consistência e a remoção de fundo. Essa característica assegura que os *sprites* gerados possam ser facilmente manipulados em *softwares* de edição externos de *pixel art* sem sofrer distorção. Isso confere à Pixie.haus um grande potencial que, no momento, não pôde ser plenamente explorado.

B.5 Análise do Animated Drawings

A ferramenta Animated Drawings, encontrada na plataforma AI Demos da Meta³, implementa o algoritmo para animação de desenhos proposto por Smith et al. (2023). A tecnologia, conforme descrita no artigo, visa animar automaticamente desenhos infantis de figuras humanoides, contendo diversas opções específicas de animações com movimentos e poses definidas. A IA é completamente gratuita, porém seu termo não permite o uso comercial das animações geradas. Seu foco em desenhos amadores facilita o reconhecimento de personagens com traços mais simples e abstratos.

Para os testes, o objetivo era produzir uma animação do personagem andando a partir do *sprite* dele. Como o algoritmo exige que os membros do personagem não estejam sobrepostos, foi selecionado o *sprite* de Luz para submissão. Além da pose da Figura 35a, duas novas poses foram criadas para a personagem (Figura 35).

Figura 35 – *Sprites* usados para o teste no Animated Drawings



Fonte: Elaborada pela autora.

A geração de qualquer animação requer que o personagem seja primeiramente analisado e reconhecido pela plataforma. Este processo consiste em quatro etapas principais:

- Enviar o desenho, com o personagem a ser animado;

³ <<https://aidemos.meta.com/>>

- Encontrar o personagem, cujo usuário ajusta a caixa de seleção de maneira que o personagem caiba dentro dela sem sobrar muito espaço;
- Destacar o personagem, cujo objetivo é segmentar o personagem do restante da imagem; e
- Marcar as articulações do personagem, fase em que se deve posicionar cada bolinha (que representam uma articulação entre: olho, orelha, centro da cabeça, ombro, cotovelo, pulso, quadril, joelho e calcanhar) no lugar correto.

Durante cada um dos processos posteriores ao envio da imagem, o algoritmo realiza uma detecção automática, permitindo que o usuário realize ajustes, se necessário. Após o reconhecimento, existe uma série de animações específicas que podem ser aplicadas ao personagem sem necessidade de prompts textuais.

Na interação inicial, a ferramenta mostrou-se capaz de reconhecer o personagem sem necessidade de ajustes muito grandes. No entanto, a animação formada⁴ apresentou algumas distorções visíveis, principalmente no rosto e no cabelo, que se alongavam e entortavam de acordo com o movimento dos braços. Apesar disso, a natureza do algoritmo, que deforma a imagem original em vez de gerar novos quadros do zero, garantiu total consistência com o estilo e design do *sprite*. O resultado foi avaliado como parcialmente satisfatório: não era um sucesso completo, mas também não era um fracasso.

A análise do resultado indicou que a ferramenta teve dificuldades em separar corretamente as áreas do braço e da cabeça, provavelmente devido à sobreposição do cabelo no desenho original. Para testar essa hipótese, um novo *sprite* foi desenhado (Figura 35b). Essa abordagem revelou uma melhora significativa, corrigindo a distorção da cabeça, conforme pode ser vista na comparação de resultados na Figura 36

Figura 36 – Comparando *frames* correspondentes



(a) *Frame* da animação do *sprite* original



(b) *Frame* da animação do *sprite* modificado

Fonte: Elaborada pela autora.

⁴ <<https://drive.google.com/file/d/16zrzJ4IZsnEMS90w3vYGKTGQjKC5adqx/view?usp=sharing>>

Outras animações foram selecionadas para verificar o desempenho da ferramenta em diferentes poses. Os resultados⁵ se mantiveram consistentes durante todos os testes. Apesar do sucesso na correção da distorção, um problema fundamental foi observado: as transformações que o *sprite* sofre fazem com que a animação perca o estilo *pixel perfect*. Este efeito, já presente no resultado anterior, é uma consequência direta da abordagem da ferramenta, que deforma a imagem em vez de recriá-la quadro a quadro. Não se trata de um erro de implementação, mas de uma incompatibilidade inerente entre a técnica de deformação e a estética específica da *pixel art*, não ocorrendo em outros estilos 2D.

Nos casos testados, essa falha foi mais evidente pelo fato de um dos braços estar na diagonal e o outro reto. Linhas diagonais em *pixel art* são significativamente diferentes de linhas retas, o que torna as manipulações da ferramenta visualmente incongruentes, como é possível notar na Figura 37.

Figura 37 – Braços em formato diferente um do outro



Fonte: Elaborada pela autora.

Diante desse contexto, um terceiro *sprite* é desenhado (Figura 35c) para que os braços tenham o mesmo formato durante as animações. Nesse teste, a ferramenta foi capaz de identificar o personagem com muito mais precisão, não sendo necessário fazer nenhuma correção manual na etapa de Destacar personagem. As animações resultantes⁶ ainda mantiveram a quebra no *pixel perfect*, contudo de forma menos acentuada quanto nos casos anteriores. A análise dos resultados também revelou uma nova limitação: a ferramenta considera o cabelo como parte do torso, não compreendendo a diferença de profundidade entre os dois elementos, causando uma sobreposição incorreta do cabelo em relação ao braço. Capturas de tela das animações criadas podem ser observadas na Figura 38.

A abordagem da ferramenta, que consiste em identificar um esqueleto no desenho para então aplicar transformações em cada parte do corpo, apresenta benefícios claros. A consistência do personagem é garantida, pois não há geração de novos quadros. A eficiência também é alta, pois um único esqueleto permite a aplicação de múltiplas animações. Além disso, a interface interativa, que permite a correção da IA em etapas, torna a plataforma prática e evita a repetição de prompts comum em outras ferramentas.

⁵ <<https://drive.google.com/drive/folders/1dFLISEUVXPb-lwEFv5RscT-sCtU-tKsc?usp=sharing>>

⁶ <<https://drive.google.com/drive/folders/1GkIV-37RprR4LGQN-1lyjayqs-gPcpDI?usp=sharing>>

Figura 38 – *Frames* das animações geradas pelo Animated Drawings 3



Fonte: Elaborada pela autora.

Contudo, a abordagem também possui suas limitações. O número de animações disponíveis é restrito e não customizável. A segmentação de partes do corpo é simplificada, o que pode levar a erros de interpretação, como o ocorrido com o cabelo. Por fim, a ferramenta não é capaz de gerar mudanças de ângulo complexas, limitando-se a espelhar a imagem. Após todos esses testes, a ferramenta é considerada funcionalmente satisfatória para criar animações de personagens 2D, embora seja inadequada para o estilo visual do projeto.

B.6 Análise do God Mode AI

Diferente das outras plataformas, a God Mode AI destacou-se por seu processo de evolução constante. Durante o período de testes, foram observadas atualizações periódicas na interface e na aplicação, o que resultou em experiências de uso distintas. Além disso, a ferramenta possui foco explícito na animação de *sprites* para jogos.

Durante os testes, o objetivo principal foi gerar uma animação do personagem Pablo andando. Na primeira interação com a ferramenta, havia um setor e procedimento específico para *pixel art*. O processo consiste em três etapas:

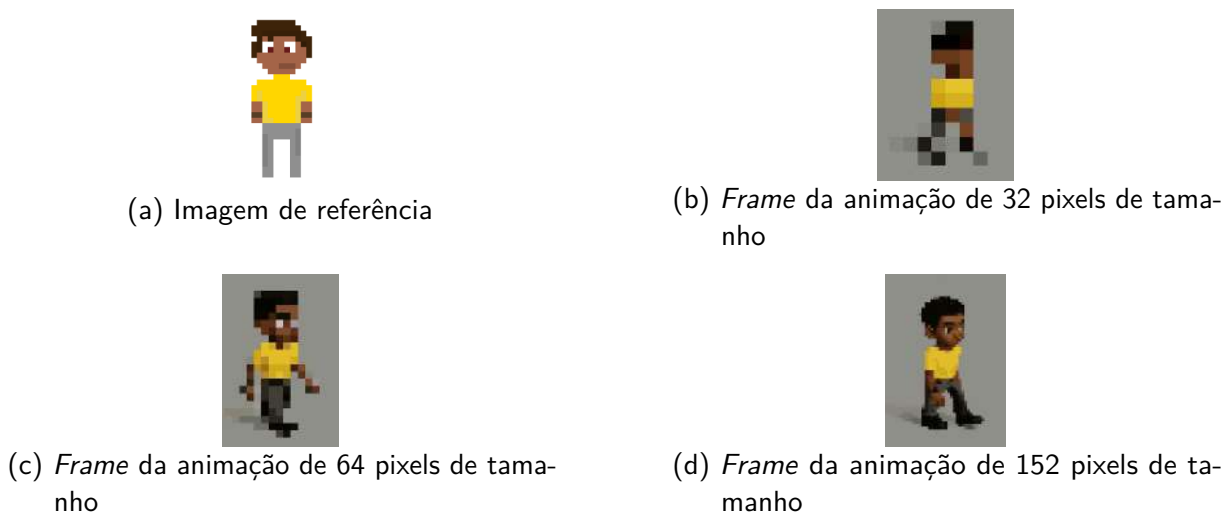
- O *sprite* original é convertido para uma versão de alta resolução;
- A animação era gerada nesse formato de alta qualidade, com alternativas de direção, movimento e auto reposicionamento (que ajusta o ângulo e a pose da imagem para corresponder ao primeiro quadro da animação), além da possibilidade de usar prompts; e
- A animação resultante era convertida de volta para o estilo *pixel art*, com opções de personalização de tamanho do pixel e paleta de cores.

Os resultados obtidos ⁷ foram superiores ao de outras ferramentas analisadas na época, conseguindo ser *pixel perfect* e reconhecendo em geral as características do personagem. Contudo, observou-se uma clara diferença visual entre o *sprite* original e a animação, pois o

⁷ <<https://drive.google.com/drive/folders/1BKe0jiX1ohblO0YUeFw540fDjaotn8tD?usp=sharing>>

estilo de *pixel art* específico não foi replicado, de forma que o personagem ficava ou muito detalhado, ou muito simples (Figura 39). Além disso, a animação formada exibia o personagem andando na diagonal, inadequada para o cenário *side view* do jogo 2D, característica que persistiu mesmo após a pixelização.

Figura 39 – Comparação do *sprite* original com quadros das animações geradas no God Mode AI



Fonte: Elaborada pela autora.

A análise foi limitada pelo sistema de créditos da plataforma, que disponibilizou apenas três créditos gratuitos (um por etapa), número insuficiente para uma exploração mais aprofundada.

Em uma segunda fase de testes, a interface da ferramenta havia sido modificada, recomendando o uso da funcionalidade principal antes de recorrer ao módulo específico de *pixel art*, com um crédito adicional disponibilizado. É importante comentar que a ferramenta principal correspondia à segunda etapa da seção de *pixel art*. A interface também passou a especificar que múltiplas tentativas de reposicionamento não gastam crédito, o que permitiu uma interação mais completa até chegar a um resultado aceitável (Figura 40a).

O resultado ⁸ gerado no ambiente principal foi visivelmente superior ao da primeira análise, embora ainda não representasse um personagem exatamente igual ao da referência. Diferente da animação anterior, esta não era *pixel perfect*, apresentando deformações, como pode ser visto na Figura 40b.

A principal inovação observada foi a funcionalidade de re-geração parcial, que permite ao usuário selecionar quadros específicos de uma animação para serem gerados novamente. Este recurso possui grande potencial para o ajuste fino de animações, mas, por custar um crédito, não pôde ser testado.

⁸ <<https://drive.google.com/file/d/1uleHmuF0SmPfPp72ER-hFL9hV-NA0qok/view?usp=sharing>>

Figura 40 – Resultados gerados pelo God Mode AI



(a) Resultado do auto reposicionamento



(b) Quadro da animação com a mão deformada no God Mode AI

Fonte: Elaborada pela autora.

O God Mode AI demonstrou ser a ferramenta com o desenvolvimento mais ativo e com o maior potencial entre as analisadas até o momento. Funcionalidades como auto reposicionamento e re-geração parcial são diferenciais importantes para uma edição mais rápida da animação formada. Embora ofereça um número limitado de animações, a plataforma permite treinar um novo tipo de movimento no modelo de IA pelo custo de 10 créditos. Outra funcionalidade importante é a exportação da animação como um *sprite sheet*, o que facilita seu processo de edição e importação para o Unity.

Apesar do potencial, devido à limitação de créditos, os resultados dos testes não atingiram a consistência e a qualidade necessárias para a aplicação direta no jogo. No entanto, o *sprite sheet* gerado nesta etapa serviu como referência para a análise da ferramenta Pixel Lab (detalhada na Seção B.8.2), tendo que ser convertido para o padrão *pixel perfect* através do Pixilart (plataforma online para desenhar no estilo *pixel art*, abrindo qualquer imagem e permitindo a redimensão do conteúdo da mesma em pixels). Na Figura 41 pode ser verificado o resultado dessa conversão.

Figura 41 – *Sprite sheet* convertido para *pixel art*

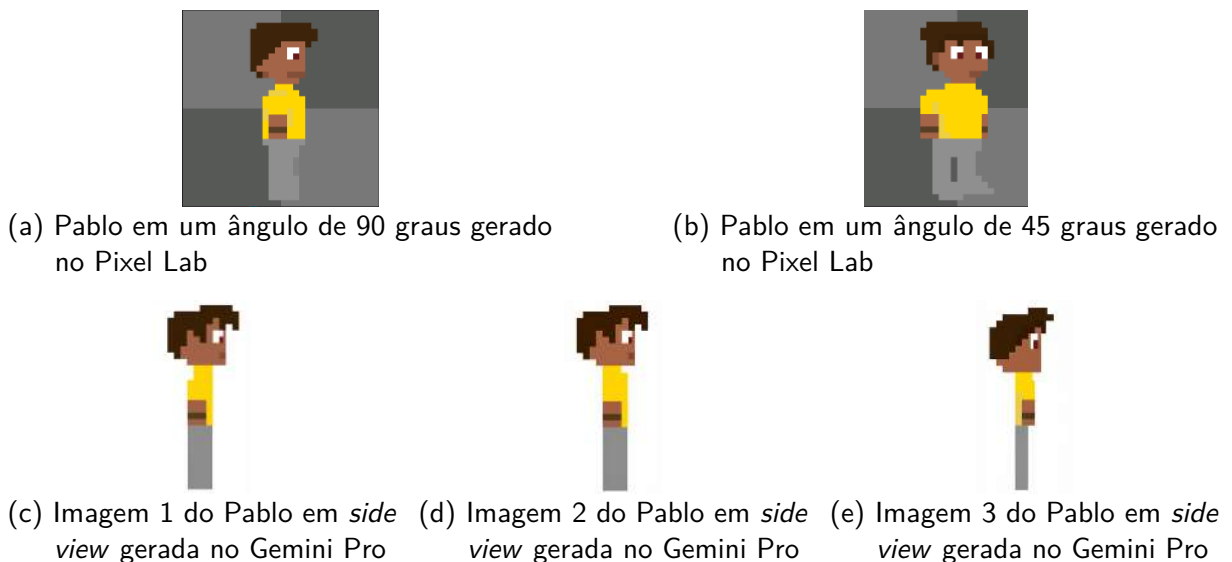
Fonte: Elaborada pela autora.

B.7 Análise do ChatGPT

A ferramenta ChatGPT foi selecionada por ser uma interface simples de conversação que usa modelos de IA capazes de gerar imagens e anexar arquivos. O modelo utilizado pelo ChatGPT no período de testes foi o GPT-4o, que não possui como único foco a geração de figuras, também tendo funcionalidades para geração de texto, processamento de linguagem natural, visão computacional (análise visual), entre outros (OPENAI, 2025a). Ele também possui um modelo de geração de imagem nativo, gpt-image-1, capaz de entender textos e figuras, aproveitando seu amplo conhecimento do mundo (OPENAI, 2025b).

Durante a análise, o objetivo principal foi gerar a imagem do personagem Pablo de lado e o *sprite sheet* dele andando, usando diferentes imagens como referência (Figura 42).

Figura 42 – Artefatos usados como referência para geração de imagens no ChatGPT



Fonte: Elaborada pela autora.

B.7.1 Geração do *sprite* em *side view*

Durante os testes iniciais, foi anexado o *sprite* em *front view do Pablo*, em sua versão de resolução 16x32, com cada pixel da arte realmente valendo um pixel real. Na primeira tentativa, o prompt utilizado instruiu a ferramenta a redesenhar o personagem olhando para o lado direito da tela, o que foi interpretado como o personagem apenas com os olhos virados nessa direção em vez do corpo inteiro. Levando esse fator em conta, ainda na mesma interação, o prompt foi reformulado, especificando melhor qual era a posição desejada. Ambos os resultados gerados mantiveram o estilo de *pixel art*, a consistência das características e tamanho dos pixels, com cores extremamente similares. Apesar disso, o bracelete que o personagem usava não foi gerado, além de ter sido adicionado um sapato cinza escuro, como pode ser visto na Figura 43a. Outro detalhe notado foi que o olho ficou ao máximo à direita, sem nenhum pixel do rosto depois

dele, priorizando aparentar ao máximo ser *pixel perfect* em vez de formar uma imagem mais precisa.

Figura 43 – *Sprites* em *side view* gerados pelo ChatGPT



Fonte: Elaborada pela autora.

Posteriormente, foi utilizada a mesma imagem de antes, porém em sua versão aumentada, com um prompt textual instruindo a ferramenta a rotacionar o personagem em 90 graus. Além disso, no teste seguinte, também foram anexadas as Figuras 42a e 42b. Os resultados gerados apresentaram uma melhora na consistência em relação às características, mantendo o bracelete. Porém, foi mais aparente a falta do padrão *pixel perfect*, além das imagens formadas apresentarem mais detalhes e tons diferentes, o que resultou em divergências extras com o design original e diferentes formatos de corpo, não tendo uma melhora significativa em comparação com o *sprite* em *side view* de referência. O melhor resultado (Figura 43b) foi gerado antes do anexo dos novos ângulos do personagem para referência, o que indica que as figuras criadas no Pixel Lab diminuíram a performance da geração no ChatGPT.

Na tentativa subsequente, foi iniciado um novo chat para apontar em detalhes o que cada imagem de referência representava, anexando, além da imagem em *front view*, os melhores *sprites* gerados em *side view* (Figuras 42c a 42e e Figura 43b). Também foi incluída para contexto a Figura 42b, com o intuito de mostrar um meio termo de resultado satisfatório entre a *front view* e a *side view*. Os resultados apresentaram um erro de *cropping* (recorte, em inglês), cuja parte superior e inferior das imagens ficavam além da tela, sendo cortadas para fora, uma limitação já conhecida da ferramenta (OPENAI, 2025c), conforme observado na Figura 43c. Foram testados alguns prompts direcionando a IA a editar a imagem e corrigir a falha, porém essa abordagem não foi efetiva.

Apesar do ChatGPT não apresentar melhor performance em comparação com as próximas a serem analisadas, é possível notar sua grande capacidade em manter a consistência e o estilo do personagem, sendo extremamente fácil e acessível de usar. Os erros encontrados não tiveram nenhuma relação com a imagem ser em 2D, e não houve grande dificuldade em gerar uma *pixel art*, apesar de não ser *pixel perfect*. O melhor *sprite* gerado nessa fase (Figura 43b) vai ser usado como referência durante a análise da Ferramenta Gemini Pro (detalhada na Seção B.11).

B.7.2 Geração do *sprite sheet* do personagem andando

Para a geração do *sprite sheet* foram utilizadas duas estratégias: a primeira utilizando apenas o *sprite* de *front view* como referência e a segunda utilizando mais imagens para a referência. Os resultados gerados pelo primeiro método foram insatisfatórios. Apesar de, comumente, haver uma consistência entre a referência e a geração, algumas características eram deixadas de lado, e o estilo de *pixel art* era ou mais complexo, ou mais simples. O principal problema encontrado foi o fato de o *sprite sheet* apresentar apenas um ou dois *sprites* substancialmente distintos, com o resto deles sendo basicamente repetidos, como pode ser visto na Figura 44a.

Figura 44 – *Sprites sheets* gerados pelo ChatGPT



(a) *Sprite sheet* com basicamente a mesma etapa do movimento de andar



(b) *Sprite sheet* usando a imagem em *side view* de referência

Fonte: Elaborada pela autora.

Durante a segunda etapa de testes, além de sempre anexar a imagem em *front view*, no primeiro caso utilizou-se também as imagens geradas pelo Pixel Lab como referência (Figuras 42a e 42b), enquanto no segundo foi usada como referência a melhor geração do *sprite* em *side view* no ChatGPT (Figura 43b).

Analisando os resultados, foi notada uma piora na qualidade do produto, com mais deformações nas características, como um terceiro braço ou duas pupilas, e erros na reprodução da *pixel art*. Além disso, o *sprite sheet* ainda é principalmente composto pelo mesmo *frame*, onde as únicas mudanças não afetam o movimento desejado e apenas alteram a aparência do personagem. Esses detalhes podem ser vistos na Figura 44b

Em conclusão, o ChatGPT não mostrou-se adequado para tentar criar um *sprite sheet*, apresentando mais erros em relação à consistência ao tentar criar esse artefato do que ao criar apenas um *sprite* específico.

B.8 Análise do Pixel Lab

A plataforma Pixel Lab foi selecionada por seu foco em *pixel art*, com diversas funcionalidades para o auxílio na criação e refinamento da animação, como: ferramenta de rotação; animação para animação; e edição de imagem.

A ferramenta também possui a capacidade de gerar os movimentos de um personagem por meio do sistema de animação baseado no esqueleto, porém esse recurso não pôde ser

testado por ser pago.

Além disso, existe um editor embutido especificamente para *pixel art* no ambiente da geração, sendo possível fazer a edição de pixels específicos e tendo outras funcionalidades básicas para a criação e edição de *sprites*. Esse editor possui uma separação de quadros da animação, podendo receber diretamente o *sprite sheet* e separar cada uma das imagens em seu respectivo *frame*, além de também conseguir exportar esse *sprite sheet* em diversos formatos, com números variados de linhas e colunas. Apesar de não formar nenhum vídeo diretamente, a ferramenta consegue tocar a animação, considerando todos os quadros ou apenas aqueles pertencentes a uma *tag* (etiqueta em inglês). Uma *tag* pode ser criada clicando com o botão direito do mouse nos números dos *frames* e selecionando a opção *new tag* (nova etiqueta, em inglês).

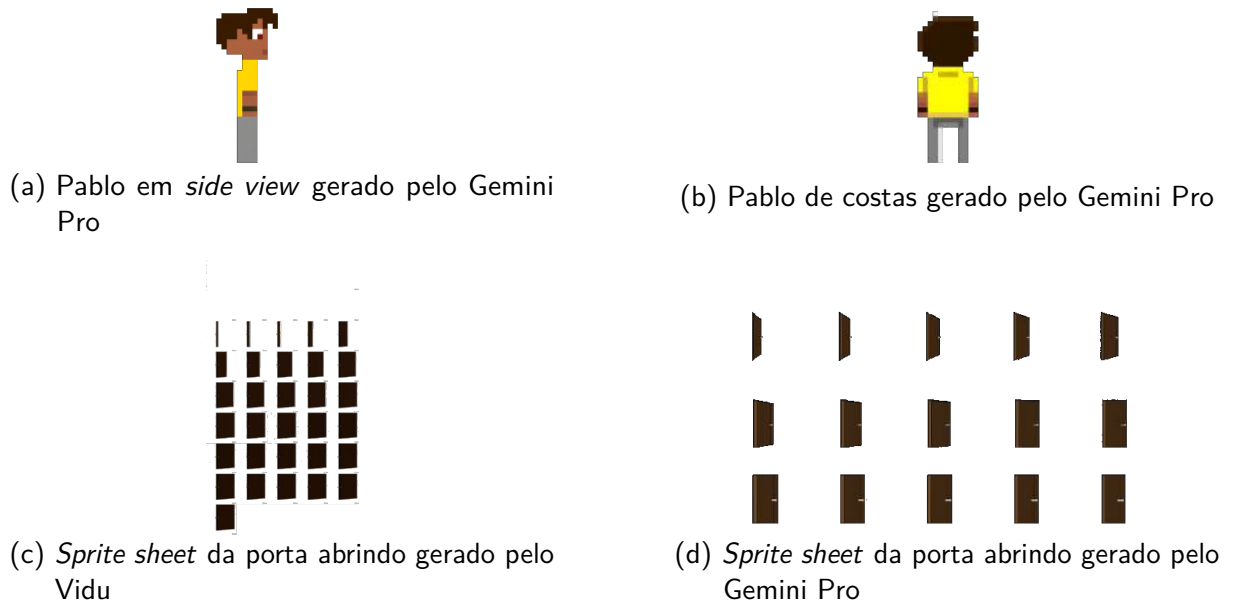
Foi realizada uma série de testes dentro da ferramenta, grande parte deles visando auxiliar na pré-produção de uma animação (criando imagens de referência), no ajuste fino e edições finais do resultado gerado por outra ferramenta. Foram usadas como referência as Figuras 3a, 3b, 41 e 45. A Figura 46 demonstra os artefatos pós processados dentro do Pixel Lab.

Figura 45 – *Sprite sheet* gerado pelo Gemini Pro



Fonte: Elaborada pela autora.

Figura 46 – Artefatos editados no Pixel Lab



Fonte: Elaborada pela autora.

B.8.1 Ferramenta de rotação

A ferramenta de rotação possui dois formatos: o componente chamado *quick rotate* (rotação rápida, em inglês), no qual, após ser selecionado pelo usuário, gera a nova imagem de acordo com o quanto foi arrastado horizontalmente (para definir o ângulo da rotação) e verticalmente (para o ângulo da inclinação), desde o momento do clique do mouse até a sua soltura; e a seção *rotate* (rotação, em inglês), que abre uma tela com várias configurações para gerar o personagem rotacionado.

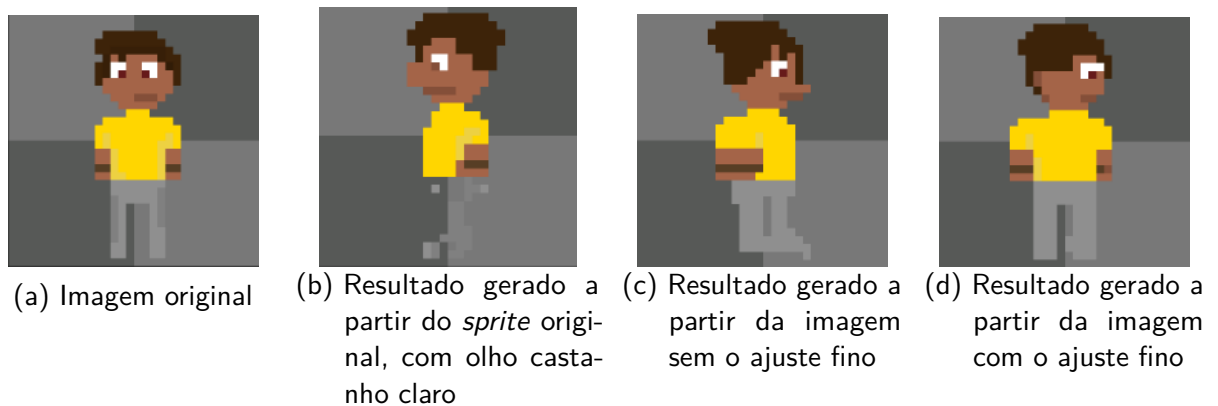
De acordo com a documentação, a ferramenta é melhor em fazer rotações pequenas, de forma que o resultado de uma rotação pode ser usado como a imagem inicial a ser rotacionada. Nesse método, porém, os erros são acumulados a cada rotação. Enquanto isso, há também a possibilidade de apenas fazer a rotação maior para evitar o acúmulo de erros, apesar de ser mais complicado para a IA. Os testes dessa funcionalidade específica visavam criar a imagem do Pablo em *side view*, a partir do *sprite* mostrado na Figura 3a.

Durante as primeiras tentativas, os resultados não geraram nenhuma rotação, apenas fazendo deformações no personagem. Porém, foi descoberto em testes posteriores que, para a geração de um bom resultado, o *sprite* deve estar centralizado no meio da tela. Dessa forma, os testes prosseguiram posicionando a imagem corretamente, gerando rotações consistentes com o personagem e o estilo, porém com algumas deformações. Analisando os resultados, foi possível perceber que a ferramenta apresenta uma dificuldade na região onde ficaria o nariz, gerando essa parte com mais imperfeições. Além disso, foi notado que existe um processo para conseguir usar somente as cores da imagem de referência:

- A imagem é gerada sem restrição nas cores;
- A imagem é recriada com tons de cores similares aos da paleta do *sprite* original, não possuindo restrição no número de tonalidades; e
- O fundo é removido e cada uma das cores é igualada à mais semelhante da paleta original.

Esse processo faz com que nem todas as características do personagem possuam a cor correta, além de muitas vezes causar o desaparecimento da calça por reconhecê-la como parte do fundo, já que as cores são semelhantes. Esses detalhes são mostrados na Figura 47a e 47b.

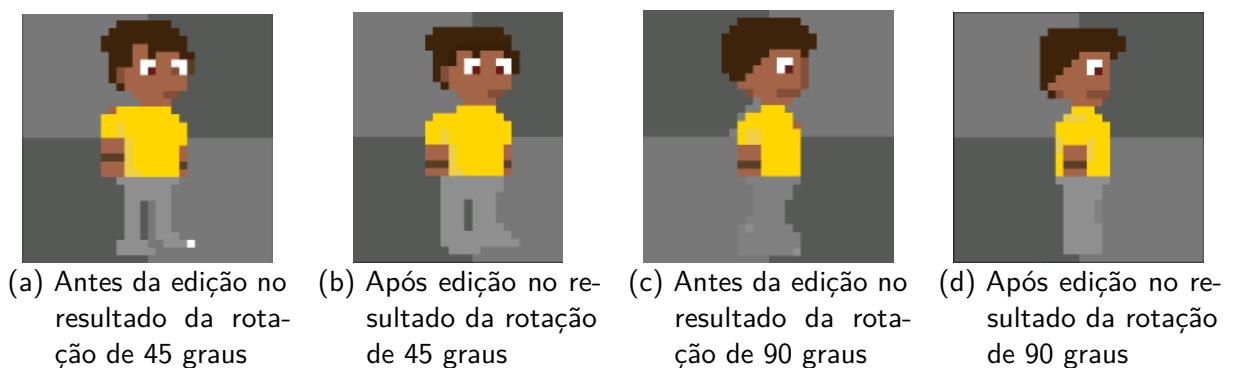
Figura 47 – Comparação entre o *sprite* original e os resultados gerados no Pixel Lab



Fonte: Elaborada pela autora.

Nos testes seguintes, foram realizados ajustes finos em algumas das imagens geradas de 45°, com o objetivo de gerar novamente o personagem em 90°, em uma tentativa de contornar o problema dos erros cumulativos. Como a ferramenta possui um editor integrado, é extremamente fácil e eficiente corrigir erros, diferente do que aconteceu em outras ferramentas. As Figuras 48a e 48b apresentam uma das edições feitas.

Figura 48 – Edição no resultado da rotação no Pixel Lab



Fonte: Elaborada pela autora.

Após esse ajuste, mais testes foram feitos, porém os resultados gerados não mostraram nenhuma melhora significativa, aparentando possuir mais deformações e imprecisões do que anteriormente. Isso pode ser melhor notado nas Figuras 47.

Considerando a falha na abordagem anterior, uma nova estratégia foi montada considerando uma das opções de customização que a ferramenta *rotate* oferece: *init image* (imagem de inicialização, em inglês). Essa funcionalidade permite ao usuário indicar uma imagem para guiar a IA em como deve ficar o resultado final, tendo uma variável associada chamada *init image strength* (força da imagem de inicialização, em inglês) para indicar o quanto essa inicialização deve ser utilizada.

Na primeira instância, o melhor resultado gerado do personagem em *side view* na ferramenta é usado como imagem de inicialização. Os resultados desta abordagem apresentaram uma melhor performance comparada aos *sprites* gerados sem nenhuma imagem de inicialização, porém demonstraram erros na região da calça, o que também pode ser visto na imagem de inicialização, com menos intensidade. Devido a esse fator, foi feita uma edição na *init image*, que pode ser vista nas Figuras 48c e 48d.

Analisando as imagens geradas ainda não foi encontrado nenhum resultado satisfatório, com todos os *sprites* gerados sem as pernas. Apesar da ferramenta quase ter gerado resultados satisfatórios, nenhum dos *sprites* alcança os padrões de qualidade para sua aplicação no jogo. Porém, as Figuras 48b e 48d foram usadas como referência para fornecer um contexto maior do personagem nas ferramentas ChatGPT (detalhada na Seção B.7) e GeminiPro (detalhada na Seção B.11).

A análise da funcionalidade de rotação indica que, embora a ferramenta não elimine a necessidade de intervenção manual, ela otimiza o processo de criação. Os resultados, embora exijam edições e ajustes finos para atingir a qualidade desejada, fornecem uma base com alta consistência e fidelidade que auxilia na produção dos *sprites* do personagem.

B.8.2 Ferramenta de animação para animação

A ferramenta de *animation to animation* (animação para animação, em inglês) utiliza o *sprite* de uma pessoa e uma animação qualquer como referência para animar o movimento desse personagem. A funcionalidade pode gerar até 15 *frames* e também usa uma descrição do personagem a ser animado e da ação a ser feita, com opções de customização do contorno e *shading* (sombreamento, em inglês) que o resultado final deve ter. A funcionalidade de limitação de cores considera a paleta do primeiro *frame* da animação de base, e não do *sprite* de referência. Existem também duas variáveis que influenciam a geração: *AI freedom* (liberdade da IA, em inglês), que determina o quanto a IA deve imitar a animação de referência; e *guidance weight* (peso da orientação, em inglês), que indica o quanto a descrição influencia no resultado.

O objetivo dos primeiros testes dessa funcionalidade foi gerar uma animação do

personagem Pablo andando, utilizando como base o *sprite sheet* gerado pela ferramenta God Mode AI (Figura 41). Antes de ser usada como referência, a imagem gerada pela ferramenta Gemini Pro (detalhada na Seção B.11) do personagem Pablo em *side view* (Figura 46a) passou por ajustes finos, como pode ser visto na Figura 49a. É importante notar que este *sprite* representa uma versão intermediária do resultado final da edição. Análises posteriores levaram a um refinamento adicional, cujo processo é detalhado na Seção B.8.3, resultando na versão final utilizada no jogo.

Figura 49 – *Sprite* do personagem em *side view* após ajustes finos no Pixel Lab



(a) *Sprite* do personagem em *side view* após ajuste fino inicial



(b) *Sprite* do personagem em *side view* após ajuste fino final

Fonte: Elaborada pela autora.

Durante o teste inicial, a imagem corrigida e a animação citada anteriormente foram colocadas como referência, enquanto era utilizado um prompt simples que descrevia as roupas usadas pelo personagem. O resultado gerado⁹ foi insatisfatório, apresentando uma aparência diferente do personagem de referência, além de não possuir o mesmo estilo de *pixel art*, como pode ser visto na Figura 50b.

Com base nisso, em testes posteriores foi adicionada a descrição da aparência do personagem no prompt. Durante cada uma das interações, algumas palavras do prompt e as configurações de sombreamento são mudadas, visando obter um resultado mais consistente com o *sprite* de referência.

A geração de uma animação possui o mesmo processo da geração de imagem, visto que um vídeo é formado por várias figuras (quadros). Durante um dos testes, ocorreu algo inesperado: nenhuma animação foi gerada. Investigando mais a fundo, foi descoberto que, na última etapa do processamento, onde o fundo deveria ser removido, o *sprite* inteiro foi deletado. Esse erro não se repetiu em mais nenhum outro teste.

Analisando os resultados, nenhum deles manteve o formato da silhueta original, os tons de cores ficaram levemente distintos do personagem a ser animado, e possuíam falhas de pixels que tornavam as características do rosto menos reconhecíveis. Foi possível notar que o tamanho e a forma do *sprite* da animação de referência influenciaram as imagens geradas, o que pode ser observado mais atentamente na Figura 50. Também foi possível observar que isso acontece

⁹ <https://drive.google.com/file/d/1TYecwF1D5EqbKaJIOJle_iZbkVOuWsux/view?usp=sharing>

por causa do funcionamento da funcionalidade animação para animação, que cria o esqueleto da animação de referência, e usa a movimentação que o esqueleto apresentou para criar uma nova animação. O esqueleto da animação de base mantém o formato e o tamanho da mesma, e o resultado final é gerado por cima desse esqueleto, utilizando as características deste.

Figura 50 – Comparação do *sprite* original com os *frames* da animação de base e das geradas no Pixel Lab



Fonte: Elaborada pela autora.

A análise mostra que usar a ferramenta com uma animação que não é consistente com o *sprite* em tamanho e formato não gera resultados satisfatórios que possam ser aplicados em um jogo. Considerando esse fator, posteriormente foi realizada uma segunda etapa de testes, utilizando para a geração da nova animação, o *sprite sheet* do vídeo do personagem andando gerado pela ferramenta Gemini Pro (Figura 45), que possui características extremamente próximas ao *sprite* original. Além disso, como imagem de referência será usada a versão final do ajuste fino do *sprite* do personagem Pablo em *side view* gerado pelo Gemini Pro (Figura 49b).

Os resultados¹⁰ gerados obtiveram uma alta consistência, porém a qualidade deles não foi maior do que a da animação de referência, com tons de cores diferentes do *sprite* original e erros nos pixels. Todavia possuindo o formato mais preciso da mão, como pode ser visto na Figura 51.

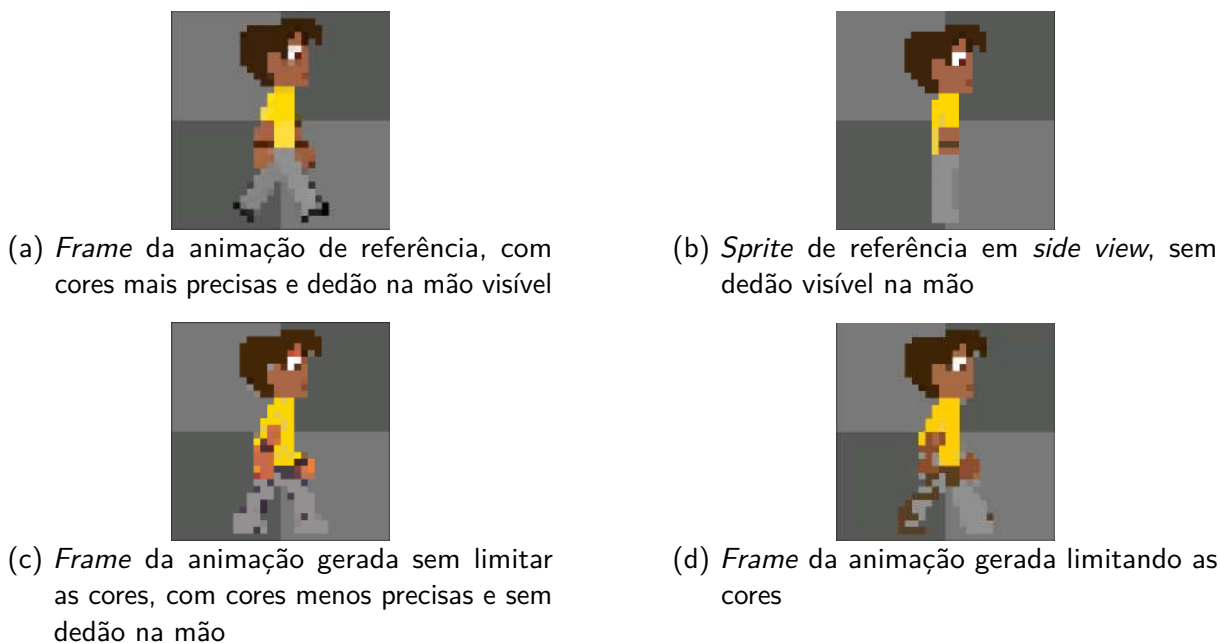
Nos testes posteriores, foi utilizada a opção de limitar as cores da animação gerada de acordo com a paleta do primeiro *frame*, que foi editada para apresentar o *sprite* original antes da animação. O objetivo era testar a capacidade da ferramenta sem buscar apenas produzir uma animação melhor pela edição manual. Como a animação mostrada parcialmente na Figura 51c possui um formato mais preciso (apesar de ter tons de cores diferentes), ela foi utilizada como referência para a animação.

Os resultados¹¹ obtiveram uma queda na qualidade, com as cores sendo utilizadas de maneira imprecisa e apresentando ainda mais erros nos pixels, como pode ser visto na Figura 51d. Investigando mais a fundo, é descoberto que esse declínio provavelmente foi causado pelo fato de que o *sprite* original não possui tons suficientes para trazer um efeito de profundidade

¹⁰ <<https://drive.google.com/drive/folders/1xmE-wpvT9xLguyX2izbWv2Xn0glh84IZ?usp=sharing>>

¹¹ <https://drive.google.com/drive/folders/1YHiklOXkW1FhYC_fBS00c21wt8fMNIq?usp=drive_link>

Figura 51 – Comparação do *sprite* de referência com os *frames* da animação de base e das geradas no Pixel Lab



Fonte: Elaborada pela autora.

na região das pernas e dos braços, fazendo a IA aplicar cores totalmente diferentes em regiões que precisam ter essa diferença.

Para explorar os limites da ferramenta, foi conduzido um último teste. O objetivo era verificar se essa funcionalidade também poderia ser utilizada para gerar uma animação de rotação completa, uma tarefa normalmente designada à ferramenta de rotação.

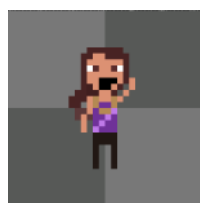
Foi criada manualmente uma animação de pseudo-rotação utilizando os *sprites* já existentes do personagem Pablo (visão frontal, lateral, de costas e lateral espelhada), cada um ocupando um quadro para simular um giro de 360 graus. Esta animação serviu como referência de movimento. Como *sprite* de referência, foi utilizada a personagem Luz, que possui um corpo com proporções e tamanho significativamente diferentes.

O resultado¹² gerado foi completamente insatisfatório, produzindo uma animação completamente deformada que tentava aplicar as proporções maiores do personagem Pablo ao corpo menor da personagem Luz. Este experimento serviu como confirmação definitiva da teoria levantada nos testes anteriores: a ferramenta prioriza a estrutura geométrica e as proporções do esqueleto da animação de referência, em vez de seguir o formato do *sprite* de referência. A inconsistência dimensional entre os dois *inputs* leva inevitavelmente a um resultado de baixa precisão, como pode ser visto na Figura 52.

A análise revela que a ferramenta é capaz de gerar animações consistentes se a animação de referência é parecida com o *sprite*, porém o resultado formado ainda exige ajustes finos,

¹² <<https://drive.google.com/file/d/1NEfmTRU46067ueFgWvvhpd1MQBiyhP1H/view?usp=sharing>>

Figura 52 – Comparação do *sprite* original com os *frames* da animação gerada no Pixel Lab



(a) *Sprite* original



(b) *Frame* 1 da animação gerada



(c) *Frame* 2 da animação gerada

Fonte: Elaborada pela autora.

possuindo falhas de pixels e tons de cores não idênticos aos desejados, o que faz com que sua performance na geração não alcance a mesma qualidade comparada às ferramentas descritas nas próximas seções. Apesar disso, assim como a funcionalidade de rotação, a criação de animação baseada em outra animação apresenta grande potencial de otimização do processo em cenário com uma animação base de alta qualidade, ou com diversos personagens de tamanho e forma parecidos.

B.8.3 Uso no pós-processamento

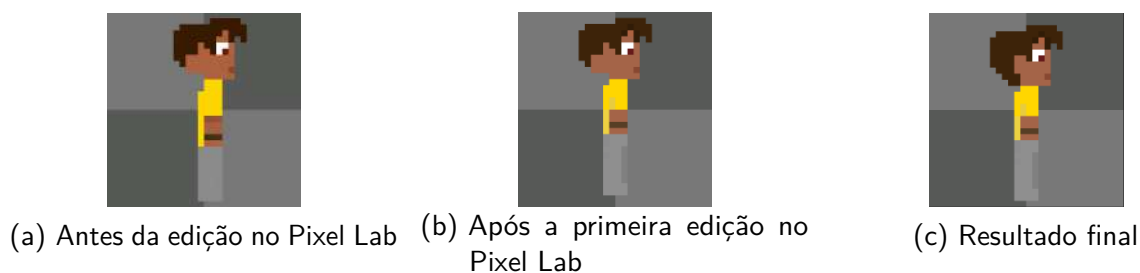
Um dos principais usos do PixelLab.AI neste projeto foi como ferramenta de pós-processamento, aplicando ajustes finos manuais nos resultados gerados por outras ferramentas de forma que o resultado final já podia ser diretamente aplicado no jogo. A forma de exportação, o editor embutido e o ambiente em *pixel art* foram funcionalidades que trouxeram eficiência para essa etapa de correções, que é justamente o que falta na maioria das outras ferramentas.

Como comentado em seções anteriores, o *sprite* do personagem Pablo em *side view* gerado pela ferramenta Gemini Pro (Figura 46a) passou por uma correção, onde os tons de cores e o tamanho foram ajustados, comparando o *sprite* lado a lado com a imagem do mesmo personagem de frente. Esse processo é extremamente importante, pois durante o jogo, o *sprite* de um personagem deve ser realmente consistente com os outros *sprites* daquele personagem, uma vez que haverá transições entre cada uma dessas figuras. Se a consistência não se mantém, em vez de apenas demonstrar um movimento ou mudança de posição, causa um estranhamento no jogador. Posteriormente, uma nova edição foi realizada para ajustar detalhes como a falta de cabelo na nuca e o formato da orelha. Esse processo é demonstrado na Figura 53.

Também foram realizados ajustes finos na Figura 46b, que mostra o *sprite* do personagem Pablo de costas gerado pelo Gemini Pro (detalhado na Seção B.11). O processo foi parecido com o do personagem em *side view*, usando a visão de frente para corrigir as cores e o tamanho da imagem. Além disso, pixels do que antes era o fundo branco foram apagados, mantendo a imagem com um fundo transparente. Esse processo pode ser verificado na Figura 54.

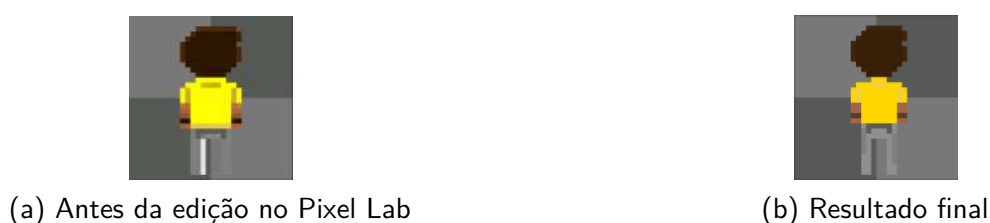
Os ajustes finos feitos no *sprite sheet* da porta abrindo em *side view* (Figura 46c) gerada pelo Vidu (detalhado na Seção B.10) foram parcialmente parecidos com os anteriores,

Figura 53 – Processo de edição no Pixel Lab do *sprite* em *side view* gerado pelo Gemini Pro



Fonte: Elaborada pela autora.

Figura 54 – Processo de edição no Pixel Lab do *sprite* em *back view* gerado pelo Gemini Pro



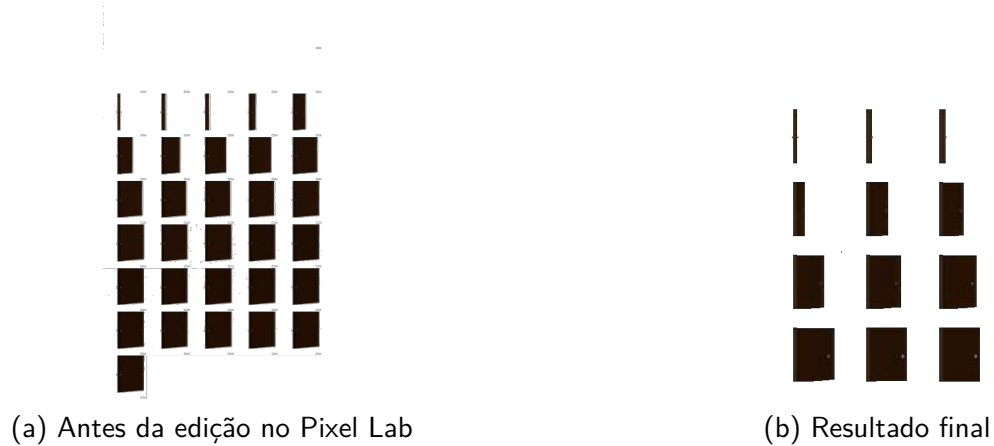
Fonte: Elaborada pela autora.

usando o *sprite* da porta fechada para corrigir a altura, além de retirar qualquer pixel branco. Além disso, a moldura da porta (presente no *sprite*) foi fixada na mesma posição em todos os quadros, pois a moldura deve permanecer igual sem se mover, mesmo quando a porta é aberta. Após isso, para cada um dos quadros, foi apagada a maçaneta ainda ligada à moldura, pintada novamente aquela parte da porta e desenhada uma nova maçaneta no canto correto. Após 12 quadros (dos 31 totais), foi observada que a porta praticamente não ficava mais aberta, com a maior parte dos outros *frames* igual ou muito similares. Dessa forma, os *frames* restantes foram apagados para não ocorrer perda de tempo em mudanças insignificantes.

A parte mais complexa do processo foi refazer a maçaneta, pois era necessário entender o formato correto que ela ficaria, dependendo do ângulo, e como representar essa forma através dos tons de cores. Apesar disso, foi uma tarefa muito mais simples e rápida do que fazer a porta inteira do zero múltiplas vezes, com cada *frame* tendo apenas pequenas modificações. O resultado desse processo final de edição pode ser verificado na Figura 55.

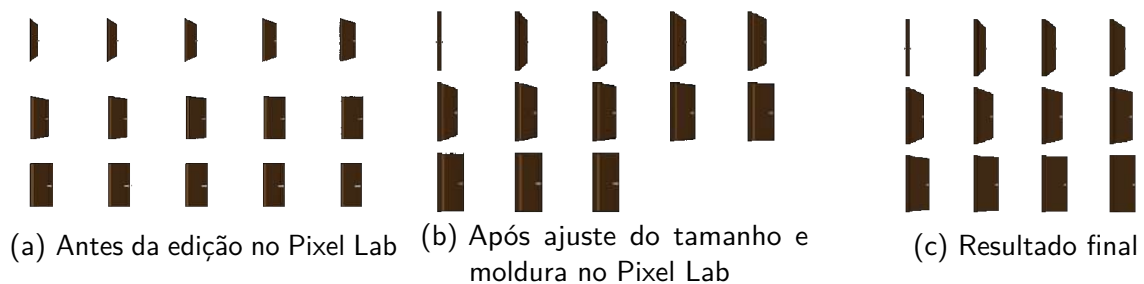
As edições realizadas no *sprite sheet* da porta abrindo em *side view* (Figura 46d) gerada pelo GeminiPro (detalhado na Seção B.11) foram parcialmente similares às anteriores, utilizando o *sprite* da porta fechada para corrigir a altura, além de fixar a moldura na mesma posição em todos os quadros. Após isso, para cada um dos quadros, foram ajustadas as cores do *sprite*. O resultado desse processo final de edição pode ser verificado na Figura 56.

Figura 55 – Processo de edição no Pixel Lab do *sprite sheet* da porta gerado pelo Vidu



Fonte: Elaborada pela autora.

Figura 56 – Processo de edição no Pixel Lab do *sprite sheet* da porta gerado pelo Gemini Pro



Fonte: Elaborada pela autora.

B.9 Análise do OpenArt.AI

A ferramenta OpenArt AI foi selecionada para análise pela sua capacidade de edição de imagens por chat e geração de vídeos, agregando múltiplos modelos de IA para isso. A plataforma possui outros módulos interessantes além dos que foram testados, como o módulo de personagem, que não pôde ser investigado pela limitação de créditos. Como base, todo usuário ao se cadastrar recebe 40 créditos grátis, sendo possível aumentar esse valor seguindo os canais da ferramenta em diferentes redes sociais.

O objetivo durante os testes foi gerar uma animação ou *sprite* do personagem Pablo sentado em *front view*, utilizando o *sprite* dele em pé como referência. O objeto cujo personagem se senta não deveria aparecer na imagem ou vídeo formado, pois a ideia era usar a mesma animação de sentar para qualquer objeto.

Primeiro, foi averiguado o módulo de vídeo, que fornecia diversas funcionalidades diferentes para a geração da animação, porém só foi possível testar apenas uma delas. As funções que pareciam mais adequadas com o objetivo foram: Imagem para Vídeo e Elementos para Vídeo. Explorando melhor a interface e instruções das duas, foi escolhida a ferramenta de

Elementos para Vídeo, porque a outra não permitia o uso de um prompt textual, criando um vídeo apenas com a imagem anexada sem nenhuma instrução.

A ferramenta permitia anexar até sete elementos ao vídeo, todavia apenas o *sprite* do personagem foi anexado para que nenhum outro objeto aparecesse em cena. Havia dois modelos de IA disponíveis para uso, Vidu Q1 e Kling 1.6. Analisando as opções, o Vidu Q1 foi considerado o mais adequado, pois o outro era mais especializado para fotorrealismo. Foi utilizado um prompt descrevendo como deveria ser a animação de sentar em detalhes.

O resultado ¹³ gerado foi insatisfatório, fazendo uma animação 3D cujo personagem é rotacionado e mexe a cabeça para cima antes de sentar em *side view*. O movimento de sentar também não é muito preciso, pois a movimentação para no meio por alguns momentos, para depois o personagem se agachar completamente, alguns segundos, depois movendo as pernas para frente e para baixo do nível inicial do chão. Apesar disso, a ferramenta foi capaz de manter as características físicas consistentes e permitiu a exportação em gif. A Figura 57 apresenta *frames* da animação gerada.

Figura 57 – *Frames* do vídeo gerado no OpenArt.AI



(a) Primeiro *frame*



(b) Último *frame*

Fonte: Elaborada pela autora.

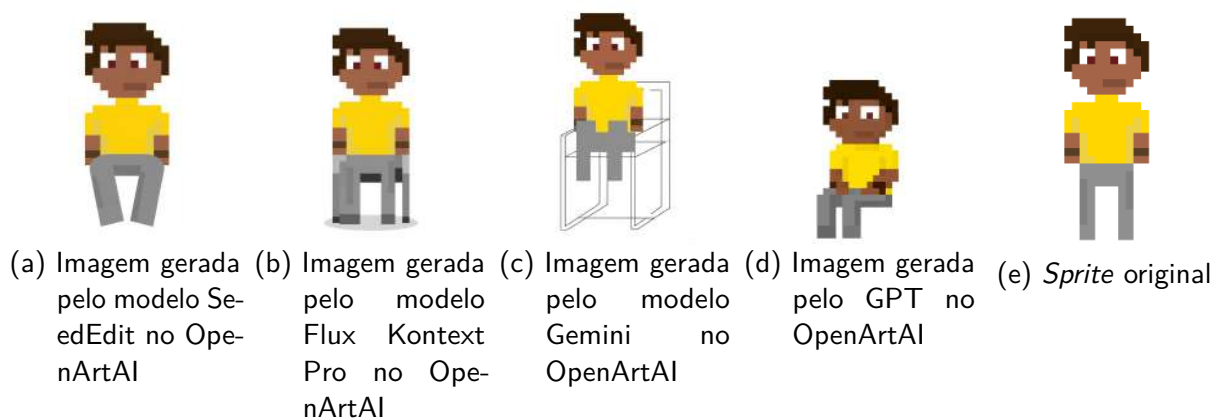
Devido à geração do vídeo não ser bem-sucedida, os próximos testes focaram apenas na geração do *sprite* do personagem sentado. Explorando o módulo de imagem, havia diversas funcionalidades, porém apenas duas delas se mostraram acessíveis e adequadas para a análise: criar imagem e chat para editar. Foi escolhida a funcionalidade de chat para editar, pois em tese a consistência do personagem seria garantida. Essa ferramenta apresentava várias opções de modelo de IA com diferentes níveis de geração. Uma interação com cada modelo foi realizada, visando editar o personagem em pé para a posição de sentado. Foi utilizado o mesmo prompt para os testes com diferentes modelos.

O modelo SeeEdit (provavelmente uma referência ao SeedEdit 3.0 da BytePlus) é um modelo de edição de imagens através de instruções de texto, se destacando em modificar áreas-chave com precisão, mantendo outras informações detalhadas com alta consistência (BY-TEPLUS, 2025). O resultado gerado foi satisfatório, sem mudar nenhuma região desnecessária do personagem, mantendo a característica e proporção da perna, e mantendo o ambiente em 2D, porém perdendo a característica do estilo de *pixel art* na área específica modificada. Essa

¹³ <<https://drive.google.com/file/d/1tCKdj0FJEHVq5y0V6qRCAkA7e2ZNrcz/view?usp=sharing>>

incongruência não chamou tanta atenção pois ainda manteve o estilo simples, fazendo com que as curvas suaves geradas nas pernas não destoassem do design geral (Figura 58a).

Figura 58 – Imagem geradas no OpenArt.AI



Fonte: Elaborada pela autora.

O Flux Kontext é uma família de modelos que utilizam o método *flow matching* (correspondência de fluxo, em inglês) para geração e edição de imagens. Diferente do modelo Flux, ele é capaz de entender imagens existentes e modificá-las de acordo com instruções de texto simples (BLACK FOREST LABS, 2025). O resultado gerado foi quase satisfatório, mantendo o personagem e o estilo consistente, porém gerando a cadeira, adicionando um sapato cinza escuro e com uma perna mais grossa que a outra (Figura 58b).

O Gemini é um modelo de IA capaz de entender, operar e combinar diversos tipos de dados, como texto e imagem (PICHAI, 2023). Porém, na plataforma OpenArt.AI não foi encontrada nenhuma informação específica sobre a versão do modelo. O resultado gerado não foi satisfatório, pois, apesar de manter a consistência, foi desenhada uma cadeira em outro estilo e ângulo. Além disso, o personagem não foi bem encaixado no assento e os joelhos ficaram dobrados para cima (Figura 58c). Testando o mesmo prompt diretamente no Gemini Pro, os resultados não foram melhores e continuou havendo problemas de precisão, como pode ser visto na Figura 59.

O resultado foi insatisfatório, alterando levemente os tons de cores e o tamanho do personagem, além de mudar o ângulo em 45 graus. Apesar disso, a imagem manteve o estilo de *pixel art* e as características em geral do personagem, não mostrando a cadeira. As Figuras 58d e 58e demonstram o personagem original ao lado do *sprite* gerado.

Apesar da animação gerada não ter sido satisfatória, o OpenArt.AI demonstrou grande potencial na edição de imagem, servindo como uma excelente ferramenta para a iteração de edições. Nenhum modelo individual foi capaz de gerar um resultado sem erros, porém deve-se notar que a análise foi limitada a uma única iteração por modelo devido à restrição de créditos. Funcionalidades de edição via chat são iterativas, projetadas para que o usuário aponte as

Figura 59 – Imagens geradas pelo modelo Gemini



Fonte: Elaborada pela autora.

falhas de uma geração e a IA as corrija em tentativas subsequentes. A impossibilidade de realizar este ciclo de refinamento pode ter impactado o desempenho final de cada modelo.

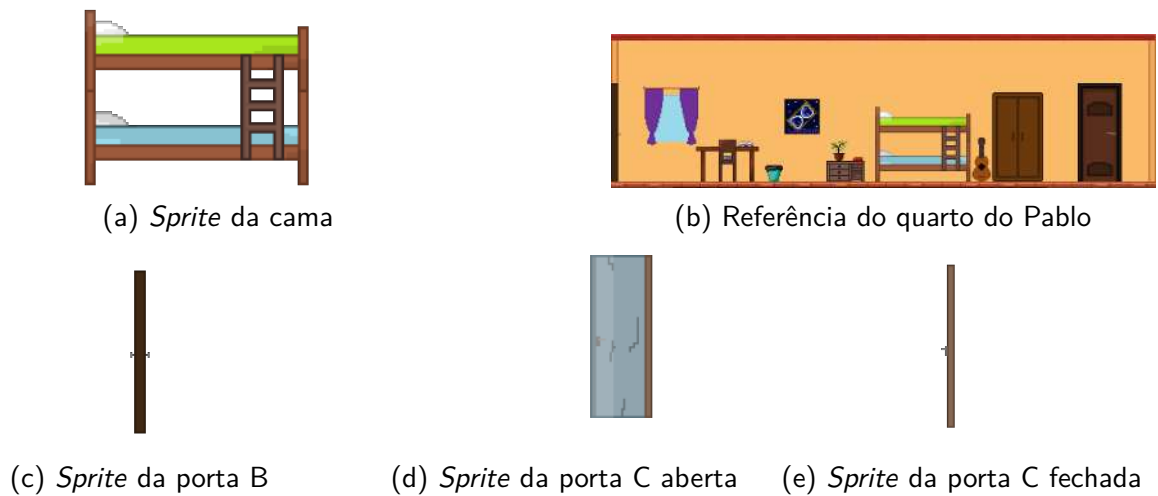
Apesar da animação gerada não ter sido satisfatória, o OpenArt.AI demonstrou grande potencial na edição de imagem, servindo como uma excelente ferramenta para a iteração e ideação de poses. Nenhum modelo individual foi capaz de gerar um resultado final sem erros, contudo, é fundamental ressaltar que a análise foi limitada a uma única iteração por modelo devido à restrição de créditos. Funcionalidades de edição via chat são inerentemente iterativas, projetadas para que o usuário aponte as falhas de uma geração e a IA as corrija em tentativas subsequentes. A impossibilidade de realizar este ciclo de refinamento pode ter impactado o desempenho final de cada modelo.

B.10 Análise do Vidu

A ferramenta Vidu foi escolhida por sua capacidade de gerar vídeos a partir de imagens. A funcionalidade mais chamativa foi a geração de vídeos por meio de referências, que prometia manter a consistência dos personagens, lugares e objetos. O site também oferecia a geração de vídeo através de uma imagem, permitindo definir o *frame* inicial e final. No OpenArt.AI (descrito na Seção B.9), era utilizado um dos modelos que pertence a essa ferramenta, o Vidu Q1, que é o modelo mais avançado para a geração de vídeos, não disponível na plataforma para uso gratuito. Dessa forma, o modelo Vidu 2.0 foi o utilizado durante os testes.

Durante a análise, o objetivo mudou de acordo com as necessidades do projeto, e alguns testes mais ambiciosos foram feitos. No total, foram criadas animações do Pablo andando, pulando, abrindo a porta, se deitando/levantando da cama, e de diferentes portas abrindo. Foram usadas de referência para o personagem as Figuras 3a, 43b, 53c e 54b. Para a cama a Figura 60a, para o quarto a Figura 60b e para as portas as Figuras 3c, 60c, 60d e 60e.

Figura 60 – Artefatos usados para referência no vidu



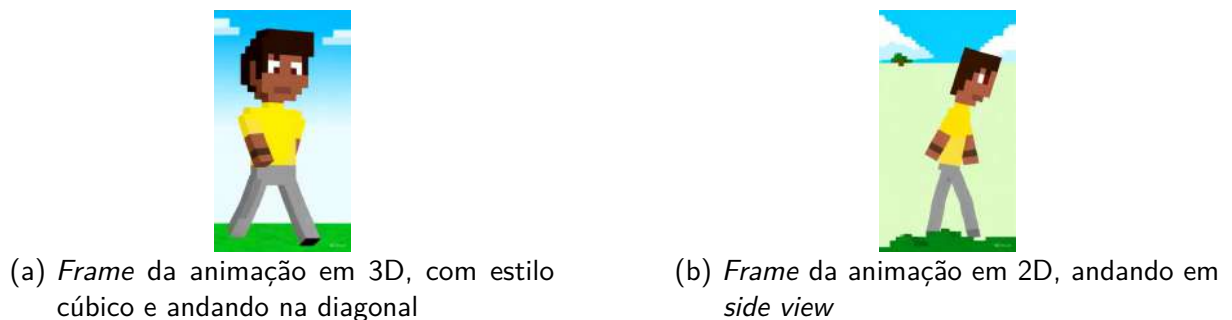
Fonte: Elaborada pela autora.

B.10.1 Funcionalidade de referência para vídeo

Os primeiros testes realizados, usando o *sprite* em *front view* (Figura 3a) para criação da animação de caminhada em *side view*, deixaram evidente uma grande dificuldade da ferramenta em manter o ambiente 2D. O personagem foi transformado para o 3D em um estilo cúbico, numa tentativa de replicar o estilo *pixel art* tridimensionalmente e mantendo as características físicas consistentes com a referência. Além disso, os resultados¹⁴ adicionavam uma paisagem ao fundo e mostravam o personagem andando na diagonal, uma direção não presente no jogo desenvolvido.

Porém, em um dos resultados, os erros de dimensão e direção foram corrigidos, formando uma animação 2D e realmente em *side view*. A Figura 61 apresenta a diferença entre os resultados, comparando um quadro do vídeo em 3D com um do em 2D.

Figura 61 – Comparação do resultado 3D e 2D gerado pelo Vidu



Fonte: Elaborada pela autora.

¹⁴ <https://drive.google.com/drive/folders/10WGbLbvQspGPJIN8q57Up7GsKqN250aA?usp=drive_link>

Apesar disso, a nova imagem formada para o personagem em *side view* possuía a cabeça muito quadrada e, mesmo que o design ainda fosse em *pixel art*, a animação deformava esse estilo, da mesma forma que aconteceu com a ferramenta Animated Drawings (descrita na Seção B.5). Outro detalhe observado nesse vídeo específico foi que, durante a animação de andar, o personagem nem sempre dobra a perna ou, quando dobra, é muito pouco, formando uma movimentação que causa estranheza ao olhar. A postura inclinada para frente, o movimento brusco do braço e a quantidade variável de movimentação do mesmo contribuem para esse estranhamento.

Para entender a causa de apenas um dos testes produzir o vídeo em 2D, analisou-se a metodologia de cada um deles. A principal diferença encontrada foi a estrutura do prompt textual. Como a plataforma mostra no tutorial, nos vídeos em 3D, a *tag* da imagem de referência foi usada como sujeito da frase. Enquanto no vídeo em 2D, a instrução foi mais imperativa e descritiva, sem considerar a *tag* como uma palavra a ser usada e apenas como forma de marcar que a imagem foi anexada. Baseado nisso, levanta-se a hipótese de que, ao não tratar a imagem como um sujeito imutável, a IA teve maior liberdade para reinterpretar o personagem e criar um novo *sprite* 2D em *side view*, em vez de apenas rotacionar a imagem de referência, o que mantinha características do *sprite* em *front view* e adicionava profundidade para parecer parcialmente de lado.

Comparando tudo, foi notado que o vídeo em 2D aparentava possuir mais *frames* com partes borradas em relação às animações em 3D. Para confirmar essa teoria, os vídeos foram transformados em *sprite sheet*, com a ferramenta ezgif¹⁵ (transforma vídeo em gif, e gif em *sprite sheet*), e foi feita uma análise quadro a quadro, verificando quais *frames* apresentavam uma grande deformação e nenhuma deformação.

A Tabela 1 comprova que o vídeo gerado em 2D, apresentou uma taxa de *frames* com deformação grave (partes muito borradas) quase duas vezes maior que a taxa somada dos vídeos em 3D. Consequentemente, o número de *frames* sem deformação foi quase metade em comparação com as versões 3D. A Figura 62 apresenta exemplos visuais que definem cada categoria de deformação utilizada na análise de cada um dos vídeos.

O aumento de deformações no resultado 2D possui duas causas prováveis: ou a estrutura do prompt menos restritiva (sem usar a *tag* como sujeito) resultou em maior instabilidade; ou a ferramenta Vidu possui uma dificuldade em manter a qualidade quadro a quadro ao gerar animações 2D, sendo mais otimizada para a manipulação de referências em 3D.

Para tentar corrigir os erros encontrados nos vídeos, uma segunda etapa de testes foi realizada. Desta vez, foi utilizada como imagem de referência o melhor *sprite* em *side view* disponível até o momento (Figura 43b). A expectativa era que, ao fornecer uma imagem já na perspectiva correta, a IA teria menos dificuldade em gerar uma animação 2D consistente, sem

¹⁵ <https://ezgif.com/>

Tabela 1 – Análise quantitativa de *frames* com deformação nos vídeos gerados pelo Vidu

| Nível de deformação | Vídeo 1 (3D) | Vídeo 2 (3D) | Vídeo 3 (2D) |
|--|--------------|--------------|----------------------------|
| Deformação grave (%) (<i>Frames</i>) | 7,32% 3 | 12,2% 5 | 36,59% 15 |
| Deformação leve (%) (<i>Frames</i>) | 48,78% 20 | 41,46% 17 | 39,02% 16 |
| Sem deformação (%) (<i>Frames</i>) | 43,9% 18 | 46,34% 19 | 24,39% 10 |
| Total (%) (<i>Frames</i>) | 100% 41 | 100% 41 | 100% 41 |

Fonte: Elaborada pela autora.

Figura 62 – Quadros dos vídeos em cada um dos níveis de deformação



(a) Quadro classificado como sem deformação



(b) Quadro com deformação leve (perna e bracelete levemente borrados)



(c) Quadro com deformação grave (rosto e corpo borrados, sem braços)

Fonte: Elaborada pela autora.

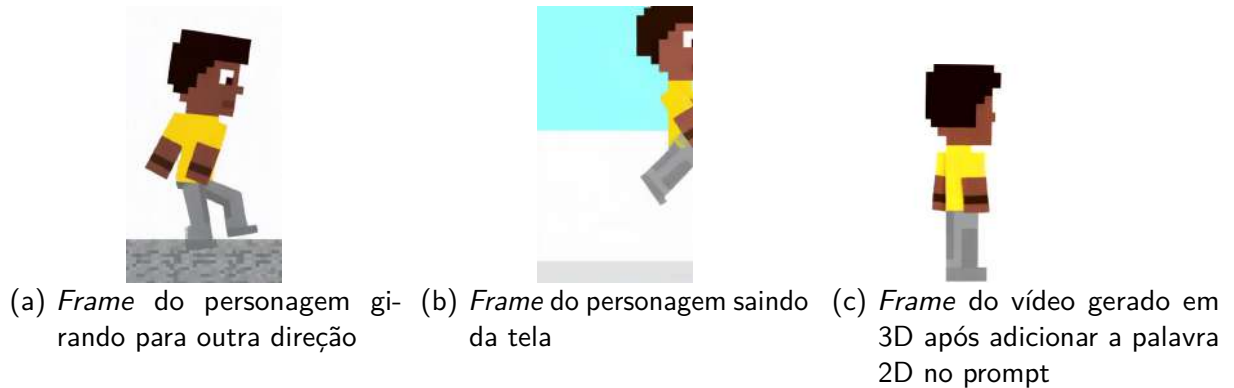
causar erro na direção e dimensão, além de apresentar um *sprite* mais adequado.

Os resultados¹⁶ confirmaram parcialmente a expectativa, eliminando em maior parte o problema da geração em 3D. Entretanto, eles introduziram novos erros, tais como a movimentação do personagem ficou ainda mais imprecisa, com a IA adicionando novas ações como pular e girar (Figura 63a), e fazendo o personagem sair da tela (Figura 63b). Ao modificar a proporção da geração de vídeo, com o objetivo do personagem ter mais espaço para se movimentar sem sair do enquadramento, o *sprite* passou a deslizar horizontalmente de maneira imprevisível. Ademais, em um dos casos, de forma contraintuitiva, adicionar a palavra 2D ao prompt resultou em uma animação 2D que se tornava 3D ao final (Figura 63c).

Nessa segunda fase de testes, foi possível gerar um resultado em 2D utilizando a *tag* da imagem como sujeito do prompt. Mesmo assim, a animação ainda apresentou a inconsistência no movimento, o que indica que a estrutura anterior de prompt não foi a causa dessa instabilidade.

Numa tentativa de corrigir um dos erros que se manteve em vários dos resultados nessa ferramenta (a presença de uma paisagem ao fundo que dificulta o processo de extrair

¹⁶ <<https://drive.google.com/drive/folders/1oUOF8-v87bZVZkrv7cTRFy0qcgdaMPZL?usp=sharing>>

Figura 63 – *Frames* dos vídeos gerados no Vidu

Fonte: Elaborada pela autora.

apenas o *sprite*), foi solicitado um vídeo com fundo transparente (utilizando a *tag* da imagem como sujeito). O resultado gerado foi uma animação 3D, que manteve o movimento de andar na direção incorreta. Para aprofundar a análise e confirmar a hipótese sobre a dificuldade da ferramenta com o 2D, a mesma análise quantitativa de deformação nos quadros foi replicada nos novos vídeos.

A Tabela 2 apresenta os dados de ambas as fases de testes, permitindo uma comparação direta e reforçando a hipótese de que as deformações são decorrentes da dificuldade da ferramenta com a geração de uma animação 2D, não tendo relação direta com a maneira em que o prompt é estruturado. Primeiramente, observa-se que todos os vídeos gerados em 2D, independentemente da imagem de referência e da estrutura de prompt, apresentam uma taxa de deformação grave e um número de *frames* sem deformação significativamente piores do que qualquer um dos vídeos gerados em 3D. Embora o uso de uma referência em *side view* tenha melhorado a estabilidade em alguns casos (Vídeo 1a na Tabela 2), a qualidade geral permaneceu baixa e outros erros sempre estavam presentes.

Tabela 2 – Análise quantitativa de *frames* com deformação, comparando testes com referência em *front* e *side view*.

| Nível de deformação | <i>Front View</i> como referência | | <i>Side View</i> como referência | | | | |
|----------------------|-----------------------------------|--------------|----------------------------------|---------------|-----------------|--------------|--------------|
| | Vídeo 2 (3D) | Vídeo 3 (2D) | Vídeo 1a (2D) | Vídeo 1b (2D) | Vídeo 2 (2D/3D) | Vídeo 3 (2D) | Vídeo 4 (3D) |
| Deformação grave (%) | 12,2% | 36,6% | 17,1% | 22,0% | 26,8% | 48,8% | 12,2% |
| (<i>Frames</i>) | 5 | 15 | 7 | 9 | 11 | 20 | 5 |
| Deformação leve (%) | 41,5% | 39,0% | 56,1% | 63,4% | 56,1% | 34,1% | 48,8% |
| (<i>Frames</i>) | 17 | 16 | 23 | 26 | 23 | 14 | 20 |
| Sem deformação (%) | 46,3% | 24,4% | 26,8% | 14,6% | 17,1% | 17,1% | 39,0% |
| (<i>Frames</i>) | 19 | 10 | 11 | 6 | 7 | 7 | 16 |

Fonte: Elaborada pela autora.

Outro fator observado foi que no vídeo 2 da segunda fase de testes, dos onze *frames* com deformação grave, apenas dois deles eram em 3D. Enquanto isso, dos sete *frames* sem deformação, cinco deles eram em 3D. Isso mostra que até mesmo durante o mesmo vídeo, a qualidade melhorou no momento em que a animação se tornou 3D.

Crucialmente, como já foi comentado, nesta segunda etapa de testes foi possível gerar um resultado em 2D (nomeado como Vídeo 3 na Tabela 2) utilizando a *tag* da imagem como sujeito. O fato de a animação ainda assim apresentar a maior taxa de deformação de todos os testes (48,8%), invalida a hipótese de que a estrutura do prompt era a causa principal da falha. A evidência agora aponta de forma mais conclusiva para a segunda hipótese: a ferramenta Vidu possui uma dificuldade em manter a consistência e a qualidade ao gerar animações em 2D.

Após as análises anteriores revelarem as dificuldades do Vidu na geração de animações 2D, o foco dos testes foi redefinido, explorando a capacidade da ferramenta em usar mais de uma imagem para a geração do vídeo. Tendo em vista que uma animação de caminhada satisfatória já havia sido obtida com a ferramenta Gemini Pro (detalhada na Seção B.11), o objetivo dos próximos testes no Vidu passou a ser a geração de animações com interações entre personagens, objetos e lugares.

Além disso, uma exploração mais aprofundada da funcionalidade de referência para vídeo revelou uma forma distinta de utilização da mesma: em vez de anexar uma imagem para ser considerada como referência, a plataforma permite criar uma referência nomeada. Essa opção permite associar um nome a um conjunto de imagens de diferentes ângulos e a uma descrição, com a hipótese de que, ao fornecer à IA uma compreensão mais completa do personagem, a consistência da animação gerada poderia ser melhorada.

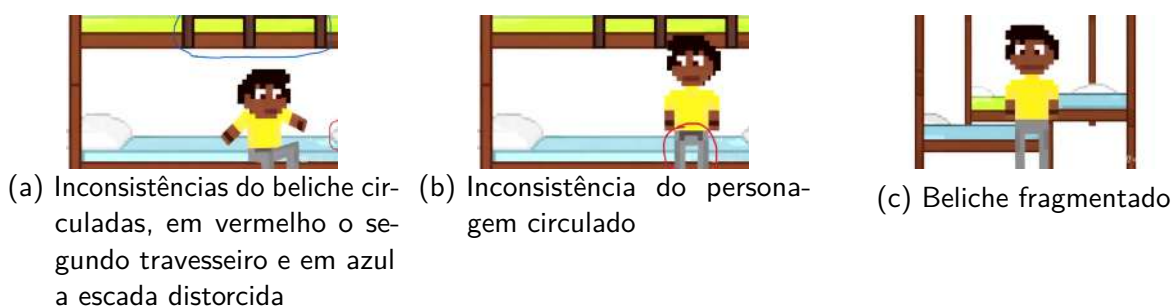
Primeiro, foram criadas as referências do personagem Pablo (Figuras 3a, 53c e 54b), do beliche (Figura 60a), da porta marrom (Figuras 3c e 60c), e do quarto do Pablo (Figura 60b). Essa criação possuía duas etapas: anexar imagens que representam a referência e escolhendo um nome para ela; e revisar o estilo e a descrição da referência, geradas automaticamente.

Inicialmente, nessa nova etapa de testes, o objetivo era criar uma animação do personagem deitado se levantando da cama. Para isso, o nome das referências substituiu o que seria a descrição do personagem e do objeto. O resultado¹⁷ parecia promissor, mantendo boa parte das características do personagem e da cama, mantendo um cenário 2D e o estilo correto. Porém, ao analisar melhor o vídeo, é possível notar imprecisões no design do personagem e do beliche. O beliche gerado sofre um grande nível de zoom, cortando parte do *sprite*, além de que a cama debaixo possui dois travesseiros (um de cada lado), e a escada foi distorcida e movida para a lateral da cama de cima (Figura 64a). O personagem possuía características de ângulos diferentes em momentos errados, como a sua calça que ficou igual à do *sprite* em *back view* mesmo quando o personagem estava virado para frente ou para o lado (Figura 64b). Além disso, a IA fez o personagem inicialmente sentado na cama, em vez de deitado, e apresentou *frames* borrados.

Para tentar criar a animação com o personagem realmente se levantando após estar deitado, foi feito um novo prompt especificando que o Pablo devia se deitar e, após isso, se

¹⁷ <<https://drive.google.com/file/d/1LG15atEW7Eba102-ADYkqW0z4RIYG67O/view?usp=sharing>>

Figura 64 – Inconsistências dos vídeos gerados no Vidu



Fonte: Elaborada pela autora.

levantar. O resultado¹⁸ foi bem pior que o esperado, e apesar do personagem ter ficado mais consistente, a cama ficou muito mais incongruente, além da ação solicitada não ser gerada. O beliche foi dividido em partes e remontado como se fossem duas camas (Figura 64c).

Numa tentativa de gerar uma animação com o beliche mais consistente, foi também usado o quarto para ser o cenário da animação. A hipótese era que a falta de um fundo definido fizesse com que a IA modificasse a cama para ficar menos vazia o ambiente. O resultado¹⁹ gerou o personagem deitado e se levantando, porém as incongruências do beliche foram mantidas. Além disso, o quarto também não era idêntico à referência (apesar das características serem as mesmas, a forma com que elas eram desenhadas não era igual), e o personagem se teleportava de um lado para o outro na cama antes de sair dela.

Em uma última tentativa utilizando apenas o personagem e a cama, o resultado²⁰ apresentou uma melhora significativa. Embora ainda apresentasse inconsistências, como uma leve mudança de perspectiva da cama e o personagem deslizando antes de iniciar o movimento, a estrutura do beliche foi gerada com maior precisão, e a ação principal de levantar-se foi executada.

Apesar de ter sido a melhor animação produzida pela ferramenta até o momento, o resultado ainda exigiu um processo de edição manual extenso para ser considerado satisfatório, incluindo a potencial remoção do beliche para ajustar a animação do personagem diretamente no cenário do jogo. Considerando que o esforço de modificação seria desproporcional à importância desta animação para o projeto, optou-se por não editar ou implementar o resultado final.

Na etapa de testes seguinte, o objetivo foi produzir uma animação do personagem abrindo a porta. Por causa dos erros de consistência do personagem nos vídeos anteriores, foi adicionada na descrição do Pablo uma parte especificando o ângulo de cada imagem. Os resultados²¹ apresentaram ou deformações na porta (Figura 65a) ou imprecisões na animação da abertura dela (Figura 65b). Apesar do movimento em específico do personagem ser adequado,

¹⁸ <https://drive.google.com/file/d/1PHQvJ2kcFuv2NWMYS08FiMOpw8_T9ygG/view?usp=sharing>

¹⁹ <https://drive.google.com/file/d/1PLI_ThD7HTMA2iA1j4OQ3exOZDhML5ZJ/view?usp=sharing>

²⁰ <<https://drive.google.com/file/d/1caNcLrZhX69wZk3ShfJYDh1qzkkP57Zo/view?usp=sharing>>

²¹ <https://drive.google.com/drive/folders/1aWPXy7SAJmMUJIEvNTxy2__ntK6ZjZy6?usp=drive_link>

é possível encontrar quadros em que ele possui incongruências (Figura 65c) e que a alta taxa de *frames* deformados ainda continua.

Figura 65 – Inconsistências dos vídeos da porta abrindo gerados no Vidu



Fonte: Elaborada pela autora.

Comparando as baterias de testes, notou-se que foi gerada a mesma área parcial do cenário nos testes em que o quarto foi utilizado, mesmo que a referência tivesse mais partes. Além disso, o beliche foi desenhado no mesmo lugar em que estava localizado no quarto, enquanto a porta foi posicionada numa região completamente distinta.

Com a análise comparativa foi criada uma hipótese sobre o comportamento da IA: ela parece processar apenas a área central da imagem de referência do cenário. Como o beliche estava posicionada perto do centro da imagem do quarto, sua localização no vídeo foi correta. A porta, no entanto, que estava na borda da imagem de referência, foi cortada de sua posição original e gerada incorretamente no centro da cena.

Figura 66 – Comparação da geração dos objetos em relação ao quarto no Vidu



Fonte: Elaborada pela autora.

Em busca de tentar aprimorar os resultados, foi encontrada uma técnica mais complexa de criar prompts para a ferramenta do Vidu. Para gerar vídeos mais consistentes e unificados, foi necessário: destacar o estilo artístico, usar termos precisos e simples, descrever o sujeito, a cena e o ambiente, e utilizar palavras-chave atmosféricas (VIDU, 2025b). Dessa forma, um novo prompt foi idealizado, gerando um vídeo²² melhor que os anteriores. Os *sprites* não possuem inconsistência notável em relação à referência e a porta abre de maneira precisa, porém o personagem só segura a maçaneta após a porta abrir, além de que ela começou aberta e depois

²² <https://drive.google.com/file/d/1o_BNadSvUQ5w4TaJZ5DBUaw0V66rc-na/view?usp=sharing>

repentinamente tornou-se fechada. A animação da porta abrindo foi satisfatória, mesmo que o movimento do personagem para abri-la não tenha sido. Apesar disso, outra ferramenta foi capaz de gerar uma animação da mesma qualidade para a porta e com menos necessidade de edições porque o personagem não estava cobrindo parte do objeto.

Ainda usando essa nova estratégia de prompt, no próximo teste o objetivo passou a ser produzir uma animação do personagem pulando. Para isso foi utilizada apenas a imagem em *side view* em vez da referência Pablo, com o objetivo de evitar erros em que o personagem vire o corpo em outro ângulo no meio da animação. Os vídeos²³ criados não alcançaram as expectativas, renderizando um fundo branco e preto e fazendo com que o personagem parecesse estar caindo ou apenas pulando de um pé só. Além disso, essa animação específica demonstra uma grande taxa de quadros borrados, ainda maior do que o esperado.

Também foi criada uma animação para a porta abrindo, só que dessa vez em *side view* e sem a presença do personagem. Os resultados²⁴ não foram satisfatórios, pois a ferramenta não animou a abertura de forma correta, gerando vídeos que rotacionavam a perspectiva da porta ou a duplicavam.

Em geral, essa funcionalidade da ferramenta apresentou grande potencial, porém possui uma grande queda de qualidade para gerar animações especificamente 2D, além de apresentar alguns erros de consistência e precisão. Esse método consegue fornecer uma base para a animação ser modificada manualmente, corrigindo diversos detalhes.

B.10.2 Funcionalidade de imagem para vídeo

A funcionalidade de imagem para vídeo permite que o usuário defina o primeiro e, opcionalmente, o último *frame* do vídeo a ser gerado, além da clássica descrição do prompt, podendo especificar ações para diferentes quadros.

Nos testes iniciais, o objetivo era criar uma animação das portas fechando ou abrindo (que ainda não havia sido feita na época). Nessa funcionalidade, não foi possível mais usar as referências criadas anteriormente, sendo possível apenas anexar uma imagem simples para os *frames* inicial e final. Utilizando as recomendações descobertas anteriormente, um prompt foi confeccionado para ser usado em todas as tentativas, que foi apenas levemente modificado de acordo com as necessidades.

Para a porta cinza, foi colocado o *sprite* dela aberta (Figura 60d) para o primeiro quadro, e a figura dela fechada (Figura 60e) para o último quadro. O resultado gerado²⁵ foi um sucesso parcial. A câmera se mexe de acordo com o movimento da porta, que foi distorcida a ponto de perder os detalhes das rachaduras e expandida na horizontal (Figura 67a) antes de se fechar. Apesar disso, encontrou-se uma oportunidade de usar a animação como base para

²³ <https://drive.google.com/drive/folders/1Lpi7zzY0BvaPkgjSv7xvc5aVubXCEVor?usp=drive_link>

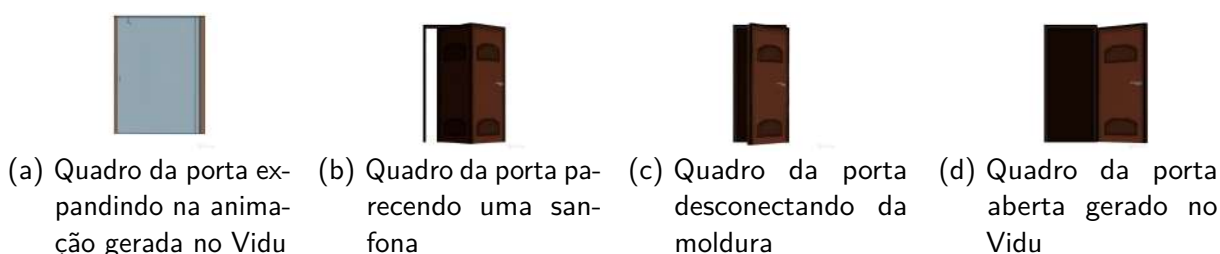
²⁴ <https://drive.google.com/drive/folders/1oplBZG15V_LfexCUw-X973PXb3gNfzIP?usp=drive_link>

²⁵ <<https://drive.google.com/file/d/1MExWoA7CkPSTmd0MBeR3Z3-u9n4KmTfW/view?usp=sharing>>

alguns ajustes, como adicionar as rachaduras novamente e manter a câmera fixa no mesmo ponto. Isso ocorreu pois o movimento de fechar e o estilo pixel art foram mantidos, permitindo que a distorção inicial fosse cortada no pós-processamento.

Para a porta marrom, apenas foi anexado ao quadro inicial o *sprite* dela aberta. Os resultados²⁶ não foram satisfatórios, pois no primeiro vídeo, a porta abre pelo lado direito (oposto ao correto) como uma sanfona (Figura 67b). Por causa disso, o prompt foi ajustado para especificar que a porta deveria abrir para o lado esquerdo. Mesmo assim, o resultado não foi preciso, pois a porta continuou abrindo do lado errado, além de desconectar-se da moldura no começo da animação (Figura 67c). Uma melhora em relação à animação anterior foi que o *frame* final gerou a porta aberta quase sem nenhuma inconsistência (Figura 67d).

Figura 67 – Quadros dos vídeos das portas gerados no Vidu



Fonte: Elaborada pela autora.

No teste seguinte, o objetivo foi gerar uma animação para o pulo do personagem, visto que foi encontrada uma animação adequada para a porta marrom. O resultado²⁷ foi extremamente insatisfatório, sem o movimento do pulo sequer ser gerado. No vídeo, o personagem dobra as pernas sem realmente se agachar e abre os braços, se inclinando para frente enquanto o fundo fica metade preto e os quadros ficam borrados (Figura 68).

Figura 68 – Quadro do personagem inclinando para frente na animação gerada no Vidu



Fonte: Elaborada pela autora.

A hipótese elaborada para a animação ter sido completamente imprecisa e com mudanças extras foi que, sem o *frame* final, a IA tem mais dificuldade em fazer movimentos cuja cena não acaba diferente do quadro inicial. O pulo não é um movimento linear: o personagem chega a um ponto mais alto e depois cai até a mesma altura em que se estava antes. Porém, a

²⁶ <<https://drive.google.com/drive/folders/1EOloislzuRKM0gus7U78opbw4Q6lfjr-?usp=sharing>>

²⁷ <https://drive.google.com/file/d/1QcPnZp21dFyTtQ_J57pV7zzT_W0MR8Js/view?usp=sharing>

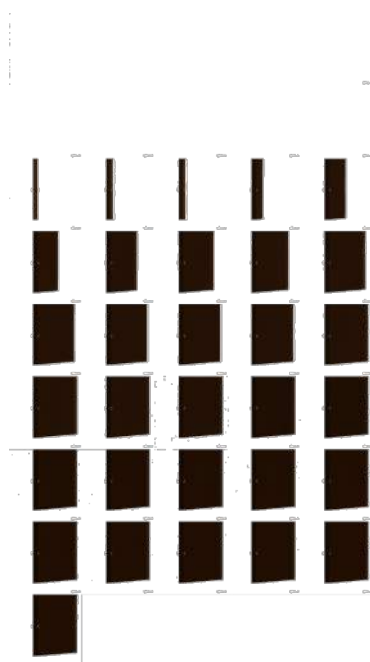
funcionalidade parece ter sido feita para mostrar uma progressão na transformação, ou seja, chegar de um estado A para B sem nenhum terceiro ponto no caminho. Dessa forma, em busca de uma mudança que gerasse uma progressão, foi modificado o fundo com o personagem apenas inclinando-se para frente em vez de fazer o movimento de ir para cima e depois para baixo.

Após algumas pesquisas mais aprofundadas na ferramenta, foi descoberto que os prompts para essa funcionalidade específica não funcionavam da mesma maneira que os do resto da ferramenta. É recomendado utilizar prompts curtos e visuais, mencionando o movimento da câmera e o estilo visual (VIDU, 2025a). As sugestões foram feitas especificamente para o modelo Vidu Q1, porém, como citado anteriormente, o teste foi realizado no modelo Vidu 2.0.

Assim, durante o último teste, com o objetivo de gerar a animação da porta marrom em *side view* abrindo, é utilizado um prompt seguindo esse padrão. O resultado²⁸ foi satisfatório, porém ele teria que passar por alguns ajustes antes de poder ser aplicado no jogo. A animação fez o movimento da porta abrindo, porém sem mover a maçaneta de lugar.

Após o vídeo ter sido baixado, foi extraído seu *sprite sheet* pelo site ezgif para ser possível fazer os ajustes com mais facilidade. Após isso, foi aberta a imagem no Pixilart, transformando-a em uma verdadeira *pixel art* (Figura 69) cujo fundo foi removido. Após isso, a animação estava pronta para ser exportada para a ferramenta Pixel Lab, em que foi realizado o pós-processamento (detalhado na Seção B.8).

Figura 69 – *Sprite sheet* em *pixel* sem fundo da animação da porta em *side view* abrindo



Fonte: Elaborada pela autora.

Conclui-se que essa funcionalidade específica, ao trabalhar com *frames* definidos, suaviza

²⁸ <https://drive.google.com/file/d/1NfyI0P6tybE1VX_NelegRJdKDVumBqmA/view?usp=sharing>

a dificuldade da ferramenta em manter a consistência em 2D, observada na análise anterior. As falhas apresentadas foram relacionadas à imprecisão da interpretação do prompt e a uma aparente limitação com movimento de progressão não linear, como o pulo. A ferramenta demonstrou grande potencial e consistência para gerar um movimento fluído entre quadros-chave já desenhados, produzindo resultados que, embora precisem de ajustes finos, tornam o processo de animação mais eficiente.

B.11 Análise da Ferramenta Gemini Pro

A ferramenta Gemini Pro foi escolhida por sua capacidade de operar, entender e combinar diferentes tipos de informação, incluindo texto, imagem e vídeo (PICHA, 2023). Mais especificamente, para os testes, foi utilizado o Gemini 2.5 Pro, que é capaz de raciocinar nativamente por meio de seus pensamentos antes de responder, o que aumenta a performance e a precisão (KAVUKCUOGLU, 2025). Para a geração de imagens, foi integrado o modelo especializado Imagen 4, e para vídeos havia o Veo 3 (WOODWARD, 2025).

Para a geração e edição de imagens, foi possível anexar diversos arquivos, enquanto para a criação do vídeo só foi permitido o anexo de uma imagem. Além disso, foi oferecida a possibilidade de re-gerar as figuras com um limite alto, porém os vídeos tinham um limite diário de três gerações e não podiam ser refeitos sem escrever novamente o prompt.

Durante a análise, o objetivo variou de acordo com o sucesso dos testes e as necessidades do projeto. No total, foi tentado criar animações do personagem Pablo andando, pulando, abrindo a porta e de diferentes portas fechando. Como referência foram utilizadas as Figuras 3a, 43b, 48b, 48d e 70 para o Pablo, e as Figuras 3c e 60c para a porta.

Figura 70 – *Sprite 2* do personagem Pablo rotacionado 45 graus gerado e editado no Pixel Lab



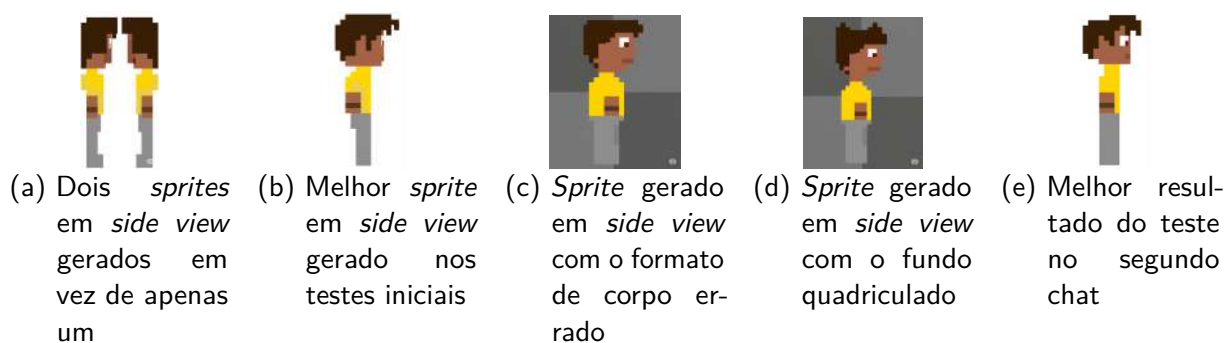
Fonte: Elaborada pela autora.

B.11.1 Geração do *sprite* em *side view*

Durante os primeiros testes, foi adicionada apenas a imagem do personagem em *front view* (Figura 3a) junto com o prompt instruindo para rotacionar o personagem em 90 graus. Os resultados mantiveram consistência com o personagem anexado e o estilo, com a maioria deles sendo precisos com a instrução, porém com a parte do rosto apresentando deformações,

principalmente em relação aos olhos (que estavam faltando ou muito estreitos). Apenas uma das imagens geradas foi imprecisa, rotacionando o *sprite* de duas maneiras diferentes (sentido horário e anti-horário) e mostrando ambos ao mesmo tempo (Figura 71a). Além disso, as imagens visivelmente não estavam no padrão *pixel perfect*, como é possível observar por meio do olho do personagem à esquerda na Figura 71a. A Figura 71b apresenta o melhor resultado gerado.

Figura 71 – *Sprites* em *side view* gerados no Gemini Pro



Fonte: Elaborada pela autora.

Como as deformações mais chamativas estavam na região da cabeça, nos testes posteriores foram anexadas as Figuras 43b e 48d para auxiliar especificamente na geração da cabeça. O resultado (Figura 71c) apresentou a cabeça menos deformada, porém apresentou o mesmo erro de formato no corpo que uma das imagens anexadas possuía, além de copiar o fundo com quadrados cinzas dessa referência.

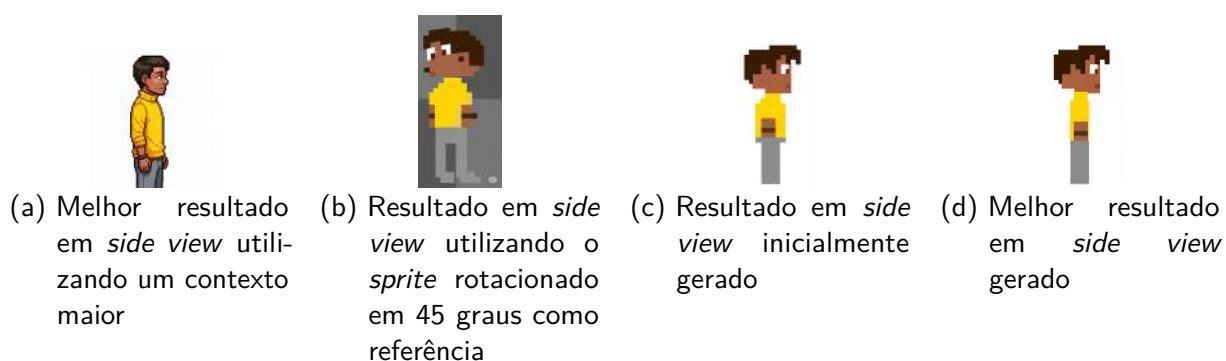
No prompt seguinte, a IA foi instruída a manter a cabeça da Figura 71c e o corpo da Figura 71b, porém a ferramenta não gerou um resultado preciso e ainda manteve o fundo quadriculado, como pode ser visto na Figura 71d.

Como a imagem do Pixel Lab estava claramente influenciando mais do que o esperado, foi criado um novo chat para novamente gerar imagens sem essa influência no contexto da IA. Também foi anexado o *sprite* em *front view* e escrito o prompt para rotacionar o personagem. Os resultados gerados foram satisfatórios, apresentando uma performance bem melhor em relação ao primeiro teste, apesar dos prompts serem quase idênticos, como pode ser visto na Figura 71e. Vale destacar que visivelmente nenhum resultado apresentou o padrão *pixel perfect*.

Numa tentativa de verificar se a ferramenta era capaz de produzir um resultado ainda melhor, foi usada a Figura 71e como referência de como as próximas rotações deveriam ficar. Os *sprites* gerados apresentaram uma performance pior, com erros na rotação e deformações no rosto. Mais alguns testes foram feitos para entender quais mudanças no prompt e nos arquivos enviados poderiam aumentar ou diminuir a qualidade dos *sprites* gerados. Adicionar a descrição do personagem não trouxe nenhuma mudança significativa na qualidade dos resultados, gerando inclusive outras imagens satisfatórias.

Escrever um prompt instruindo a ferramenta a fazer o *sprite* do personagem olhando para a direita e adicionar mais imagens de referências (Figuras 3a, 43b, 48b, 48d e 70), dando o contexto do que cada figura representa em diferentes mensagens, fez com que a IA gerasse um *sprite* inconsistente com o estilo de *pixel art* (Figura 72a). Ao observar o raciocínio da IA, foi descoberto que o Gemini apenas formula um prompt para ser enviado ao modelo especializado na geração de imagens, que na época era o Imagen 4. Com isso, foi levantada a hipótese de que, como os *sprites* de referência foram anexados numa mensagem anterior, o modelo de chat não estava passando essas figuras para o Imagen 4 na hora de gerar a imagem. Para confirmar isso, a ideia era comparar o raciocínio da geração consistente com a da inconsistente, porém em nenhuma das imagens consistentes havia a opção de mostrar os pensamentos. Dessa forma, não foi possível testar a validade da teoria.

Figura 72 – Mais *sprites* em *side view* gerados no Gemini Pro



Fonte: Elaborada pela autora.

Utilizar como referência o *sprite* rotacionado em 45 graus (Figura 70), com o prompt pedindo uma rotação de 45 graus para o personagem ficar em *side view*, gerou o personagem na visão incorreta, como pode ser visto na Figura 72b.

Foi feita uma última tentativa em gerar um *sprite* ainda melhor do que o anteriormente gerado, focando em redesenhar o personagem utilizando as imagens dele em *side view* como referência (Figuras 48d e 71e), ao invés de realizar a rotação. A ideia era manter a cabeça parecida com a do melhor *sprite* em *side view* gerado pelo Gemini Pro até o momento, bem como manter o corpo com relevo em ambos os lados, como acontecia no melhor *sprite* em *side view* gerado pelo Pixel Lab. O resultado gerado inicialmente (Figura 72c) foi editado na própria ferramenta por meio de outros prompts, formando a Figura 72d.

Posteriormente, essa imagem foi passada para o padrão *pixel perfect* utilizando o Pixilart. Ainda dentro do Pixilart, foram feitos pequenos ajustes para corrigir os erros gerados durante a conversão. Esse processo pode ser verificado na Figura 73. Por último, o *sprite* foi exportado para a ferramenta Pixel Lab, em que mais ajustes foram realizados (detalhados na Seção B.8).

Conclui-se que o Gemini Pro demonstrou um desempenho superior em relação às

Figura 73 – Processo de edição do melhor *sprite* em *side view* no Pixilart



Fonte: Elaborada pela autora.

ferramentas anteriores na geração do *sprite* em *side view*, mantendo alta consistência com o personagem e o estilo em quase todos os testes, e sendo preciso na interpretação dos prompts na maioria dos casos. Apesar disso, os resultados não possuem o padrão *pixel perfect* e a eficácia da IA é sensível ao contexto da conversa, o que pode exigir a criação de novos chats para evitar uma repetição de erros. Embora a edição de imagens diretamente na plataforma seja limitada ao uso da IA, a capacidade dessa ferramenta em gerar um desenho base coeso e de alta qualidade solidifica seu papel na eficiência na produção de poses para auxílio no processo de animação.

B.11.2 Geração do *sprite* em *back view*

Para a geração do *sprite* em *back view*, foi usado como referência o *sprite* em *front view* (Figura 3a) e o *sprite* em *side view* (Figura 73c). Inicialmente, o modelo interpretou errado a instrução de fazer o personagem virado para o norte, gerando o *sprite* em *front view* com deformações de pixels no rosto, que aparentavam formar um sorriso.

Na interação seguinte, foi especificado melhor que o personagem deveria estar de costas. Os resultados gerados foram todos satisfatórios, mantendo alta consistência com o estilo e o personagem, como pode ser visto na Figura 74a. A olho nu, os *sprites* parecem manter o padrão *pixel perfect*, sendo facilmente exportados para aplicativos de edição de *pixel art*.

Apesar disso, foi notado que os tons de cores não eram exatamente idênticos aos do *sprite* original, sendo necessário que a imagem passasse por essa correção. Assim como foi feito com o *sprite* em *side view*, a imagem foi exportada para o Pixilart, como pode ser visto na Figura 74. Após isso, a imagem foi colocada na ferramenta Pixel Lab, no qual foi realizado o pós-processamento (detalhado na Seção B.8).

Em geral, os testes para a geração do *sprite* em *back view* revelaram uma capacidade de consistência ainda maior do que a observada anteriormente. O modelo foi capaz de simular o padrão *pixel perfect* com alta fidelidade a olho nu, um feito impressionante para um modelo de IA generativo que não é especializado nesse estilo, solidificando ainda mais o Gemini Pro

Figura 74 – Processo de edição do melhor *sprite* em *back view* no Pixirart



(a) Melhor *sprite* em *back view* gerado pelo Gemini Pro antes de converter para *pixel perfect*



(b) *Sprite* em *back view* após conversão para pixels no Pixilart

Fonte: Elaborada pela autora.

como a ferramenta mais eficiente para a criação de poses base para animações 2D.

B.11.3 Geração do *sprite sheet* do ciclo de caminhada

No teste inicial, foi anexado o personagem em *front view* (Figura 3a) e o melhor *sprite* gerado até aquele momento em *side view* (Figura 72c), junto com a descrição do personagem e o contexto de cada imagem. Apenas na mensagem posterior foi enviado o prompt instruindo a ferramenta a gerar o *sprite sheet* com 16 *frames*. Isso foi feito para verificar se o erro de consistência (descrito na Seção B.11.1) ocorreria novamente.

Como esperado, os resultados demonstraram inconsistência com o estilo, como se estivessem usando como base apenas as características do personagem e ignorando as imagens de referência. Isso contribuiu para a hipótese anterior, sobre o modelo de chat estar enviando apenas o prompt textual para o Imagen 4 por causa do envio das imagens numa mensagem diferente. Para consolidar essa teoria, mais testes foram realizados utilizando um prompt idêntico com a imagem anexada na mesma mensagem. Porém, antes de abordar esses novos experimentos, é importante ressaltar outras características presentes em todas as figuras geradas. Apesar da IA ter feito 16 quadros, como aconteceu durante a análise de outras ferramentas, o mesmo *sprite* era repetido com mudanças não significativas, sem formar o movimento de caminhada.

Diante desse contexto, foi anexada a melhor imagem do personagem em *side view* gerada (que naquele momento era a Figura 72d) e repetido o mesmo prompt de antes. De maneira inesperada, os resultados apresentaram consistência variável com o estilo. Além disso, o número de quadros gerados foi diferente do valor instruído, possuindo diversos *frames* repetidos, porém eventualmente mudando a pose para avançar o movimento (o que não ocorreu na geração anterior).

Analisando a fundo, foi levantada a teoria de que os resultados foram influenciados pelo contexto já existente da ferramenta, pois ambas as fases de testes foram realizadas no mesmo chat. Isso explicaria o motivo de algumas imagens geradas manterem a consistência com a figura anexada, enquanto outras apresentavam a mesma inconsistência do experimento anterior. Para eliminar essa interferência, foi criado um novo chat, no qual replicou-se de maneira exata o teste inicial (utilizando as Figuras 3a e 72c), exceto pelo fato de que os textos (descrição, contexto e prompt) foram unidos em uma só mensagem.

Os resultados apresentaram uma consistência maior do que os testes anteriores, mantendo características específicas das referências, porém perdendo o estilo de *pixel art* e com diversas deformações no *sprite*. O erro de *frames* sem mudanças significativas diminuiu, porém foram gerados mais do que 16 quadros e algumas posições apareceram de forma imprecisa. Em alguns casos, o personagem era desconstruído ou parecia fazer ações diferentes do que apenas andar. Em geral, apesar de mostrarem uma semelhança maior com o estilo da imagem de referência, os resultados foram menos precisos, com mais deformações e menos satisfatórios quando considerada apenas a descrição textual do personagem.

Nos testes seguintes, voltou-se a anexar apenas a melhor imagem gerada em *side view* até aquele momento (Figura 72d). Quando era usado o mesmo chat, geraram-se *sprites* mais parecidos com o de referência e com levemente menos deformações, porém ainda mantendo as demais características comentadas anteriormente. Entretanto, ao se utilizar um novo chat, a consistência do personagem e do estilo aumentou de forma satisfatória na maioria das figuras geradas, com um número menor de deformações em comparação com os outros resultados. De maneira inesperada, essa estratégia gerou, em certos momentos, sequências de imagens com os *sprites* em vez de uma imagem com o *sprite sheet*, além de apresentar uma maior repetição de quadros sem mudança significativa.

Analisando e comparando todos os resultados, compilados no Quadro 14, é consolidada a hipótese sobre a importância de anexar as imagens de referência na mesma mensagem do prompt. A provável causa desse comportamento é que o modelo de chat não retém o conteúdo visual de mensagens anteriores que não tiverem relação com uma geração da imagem, dessa forma, não podendo enviar a figura ao formulário a instrução para o Imagen 4.

Quadro 14 – Resumo dos experimentos de geração do *sprite sheet* no Gemini Pro

| Bateria de teste | Teste 1 | Teste 2 | Teste 3 | Teste 4 | Teste 5 |
|---|---------------------------------------|------------------------------------|------------------------------------|------------------------------------|---|
| Sprites anexados | | | | | |
| Método de prompt | Imagem na mensagem anterior ao prompt | Imagem na mesma mensagem do prompt | Imagem na mesma mensagem do prompt | Imagem na mesma mensagem do prompt | Imagem na mesma mensagem do prompt |
| Contexto do chat | Nenhum | Teste 1 | Nenhum | Teste 3 | Nenhum |
| Melhor resultado | | | | | |
| Pior resultado | | | | | |
| Semelhança com as imagens de referência | Extremamente baixa | Variando entre muito alta e baixa | Alta | Alta | Variando entre alta e extremamente alta |
| Consistência com o estilo | Baixa | Variando entre alta e baixa | Média | Alta | Muito alta |
| Qualidade do <i>sprite</i> baseada na descrição | Alta | Média | Muito baixa | Baixa | Média |
| Precisão com o prompt | Média | Baixa | Baixa | Baixa | Média |
| Nível de deformação | Muito baixo | Médio | Muito alto | Muito alto | Médio |
| Taxa de repetição | Extremamente alta | Média | Média | Baixa | Alta |
| Taxa de incoerência da caminhada | Muito baixa | Média | Muito alta | Alta | Alta |

Fonte: Elaborada pela autora.

Os experimentos também revelaram que a ferramenta Gemini Pro apresenta dificuldades em gerar sequências de imagens com pequenas variações. A IA tende a repetir quadros e poses, introduzir deformações ou perder a precisão da progressão do movimento ao tentar criar os múltiplos *sprites* de uma animação. Conclui-se que, embora excelente para a criação de *sprites* específicos, o Gemini Pro não se mostrou capaz de formar o *sprite sheet* de uma animação, falhando em compreender o contexto temporal de um ciclo de caminhada.

B.11.4 Geração da animação de caminhada

Como comentado anteriormente, a geração de vídeo era limitada a três animações por dia. Esse fator fez com que, antes mesmo de ter o *sprite* final em *side view* do personagem, fossem realizados testes para produzir o movimento de caminhada. Portanto só era possível anexar uma única imagem e o prompt tinha que ser reescrito novamente a cada resultado gerado. Todos os vídeos produzidos possuem som, porém o foco da análise foi especificamente no conteúdo visual produzido, sem levar em conta o material sonoro.

Durante os testes iniciais, foi utilizado o *sprite* do personagem em *front view* como imagem a ser transformada em vídeo. Os resultados²⁹ gerados mantiveram em grande parte a *pixel art*, criaram precisamente o movimento de caminhada e formaram o *sprite* em *side view* consistente com o estilo. Porém, foram observadas incongruências em relação a características físicas específicas e aos tons de cores. Um detalhe interessante de ser ressaltado foi que, diferente das outras ferramentas de vídeo, a animação não deformou completamente a *pixel art*, como pode ser visto na Figura 75, onde está circulada em vermelho a diagonal incoerente com *pixel art* e circulada em azul a diagonal pixelizada de forma coerente. Apesar da alta qualidade, a inconsistência chamativa na aparência fez com que esses vídeos não fossem considerados satisfatórios para a animação no jogo.

Figura 75 – Quadro da animação de caminhada gerada no Gemini Pro



Fonte: Elaborada pela autora.

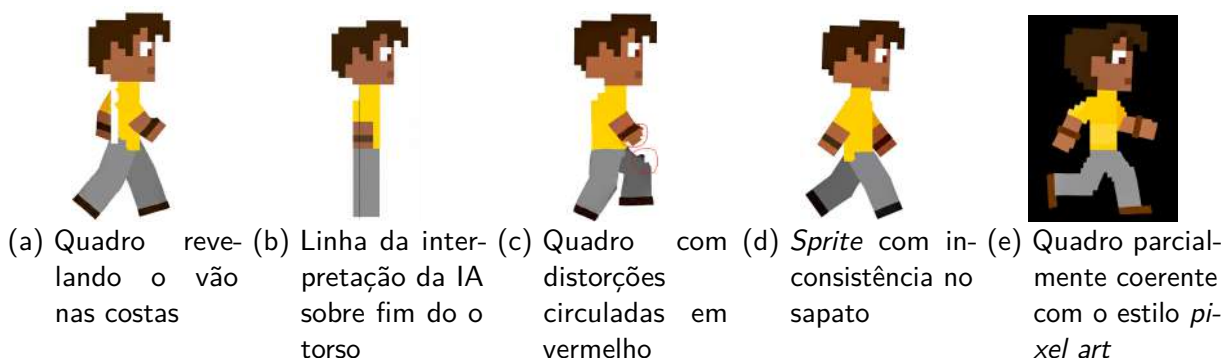
No teste seguinte, foi anexado o melhor *sprite* em *side view* que havia até aquele momento (Figura 71e), fazendo com que não fosse preciso gerar um *sprite* em *side view* junto com o vídeo, sendo necessário produzir apenas a animação. O resultado³⁰ foi extremamente consistente com a referência e apresentou o movimento correto de andar. A animação deforma a *pixel art* de modo a formar diagonais incoerentes, porém essa falha não foi muito evidenciada. Apesar disso, o vídeo possuía um erro grave que tornava o mesmo inadequado para uso: durante a animação, quando o braço se move, é revelado um buraco nas costas. Isso pode ser visto na Figura 76a.

Analisando a imagem de referência, foi teorizado que a maneira em que o *sprite* estava desenhado foi responsável por fazer com que a IA entendesse que a linha do corpo acabava antes (Figura 76b), o que criava um espaço vazio entre o torso e a perna, formando o vão.

²⁹ <https://drive.google.com/drive/folders/1rbBwuVsgvShD8JoruLJN_poLJRwVSOcO?usp=sharing>

³⁰ <<https://drive.google.com/file/d/1dQZF4nImDsFU4jw68rXUGffW1hfGMWA/view?usp=sharing>>

Figura 76 – Quadros das animações de caminhada geradas no Gemini Pro



Fonte: Elaborada pela autora.

Para evitar essa falha, o *sprite* apresentado na Figura 72c foi usado como referência no teste posterior. O resultado³¹ gerado foi satisfatório, apresentando alta consistência e precisão. Apesar disso, o vídeo adicionou sapatos pretos no personagem e apresentou distorções na região dos joelhos e das mãos (Figura 76c), além da animação deformar a *pixel art* fazendo-a perder a coerência.

Como foi rapidamente gerada uma animação adequada, os experimentos seguintes focaram em testar se a ferramenta mantém esse alto padrão na geração, e se era possível criar um vídeo que não tivesse nenhuma distorção. Para esses testes, uma versão aprimorada do *sprite* anterior (Figura 72d) foi utilizada como referência. Os resultados³² também foram satisfatórios, apresentando alta consistência e precisão, além de não apresentar as distorções citadas anteriormente. Porém, ainda foram encontrados pequenos erros de consistência e apesar disso, um dos vídeos gerados mostrou-se mais adequado do que os prévios, com apenas uma única inconsistência visível (Figura 76d).

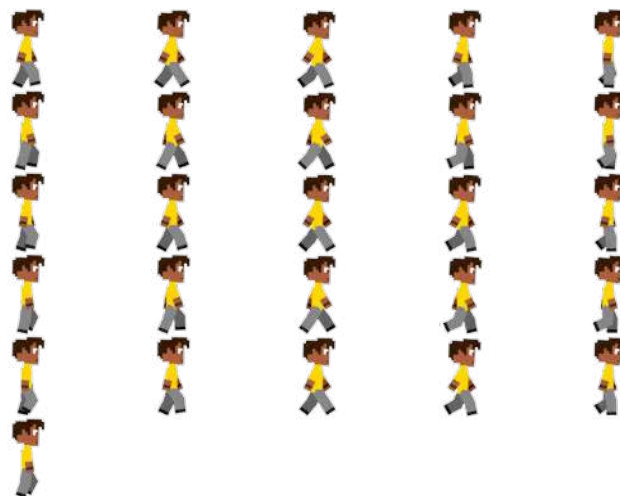
Comparando todos os vídeos obtidos, foi possível perceber que as animações geradas usando o personagem em *front view* como referência apresentavam menos incoerências na *pixel art* em relação aos que usavam um *sprite* em *side view*. Foi levantada a hipótese de que a IA consegue gerar de maneira mais precisa a animação em *pixel art* quando a referência está no padrão *pixel perfect*. As imagens em *side view* não estavam nesse padrão pois tinham sido feitas por IA.

Como havia outras animações para serem geradas, foi decidido fazer a implementação do melhor resultado no jogo. Para isso, o vídeo foi transformado em um *sprite sheet* usando a ferramenta ezgif. Além disso, também foi necessário remover o fundo da imagem para que o mesmo não aparecesse durante o jogo. Para isso, foi utilizada a ferramenta removebg³³, cujo resultado pode ser consultado na Figura 77.

³¹ <<https://drive.google.com/file/d/1Bi-5QSThXMqe6zONrDluai8ukYTAYkVj/view?usp=sharing>>

³² <https://drive.google.com/drive/folders/19e9IRIDhn1UIBP_wyuXw2AlhyZP0OnSC?usp=sharing>

³³ <<https://www.remove.bg/upload>>

Figura 77 – *Sprite sheet* do ciclo de caminhada sem fundo

Fonte: Elaborada pela autora.

Posteriormente, o *sprite* em *side view* foi ajustado e aprimorado no Pixel Lab, o que tornou ultrapassada a imagem usada na animação de caminhada. As diferenças não eram muito grandes, de maneira que o vídeo ainda fosse adequado para ser usado no jogo. Ainda assim, novos testes foram realizados com a Figura 53c de referência. Com isso, os resultados³⁴ não foram melhores do que a animação já implementada, pois todos apresentaram mais erros de consistência que o anterior, em específico fazendo o bracelete ganhar um relevo que dava uma vaga sensação de 3D. Diversos prompts diferentes foram usados em busca de corrigir essa falha, porém não houve sucesso. Também é importante notar que alguns dos resultados conseguiram parcialmente manter a coerência da *pixel art* durante a animação (Figura 76e). Isso consolida a hipótese levantada antes sobre a IA ter mais facilidade em animar de maneira pixelada ao usar um *sprite* no padrão *pixel perfect*, como foi o caso durante esses testes.

De maneira não planejada, uma animação de caminhada satisfatória com o *sprite* final foi obtida durante os experimentos para gerar a animação do personagem abrindo a porta, conforme será detalhado na Seção B.11.6.

A ferramenta Gemini Pro demonstrou uma alta capacidade de precisão e consistência para a geração de animações 2D, sendo parcialmente capaz de lidar com o estilo de *pixel art*. Algumas pequenas distorções e inconsistências podem ser encontradas nos vídeos, porém são detalhes pouco significativos que podem ser corrigidos através de um ajuste manual, trazendo um ganho de tempo em comparação a desenhar o personagem do zero para cada *frame*. Esse processo de edição, entretanto, torna-se mais complicado pelo fato de que a ferramenta não apresenta um editor embutido.

³⁴ <https://drive.google.com/drive/folders/16ljPYloPZtI81zKZ3kOz3I3qkPX078_9?usp=sharing>

B.11.5 Geração da animação de pulo

Para a geração da animação de pulo, foi possível utilizar a versão final do *sprite* em *side view* (Figura 53c). O resultado inicial³⁵ alcançado apresentou alta consistência com o *sprite*, conseguindo gerar uma animação parcialmente coerente com o estilo de *pixel art*. No início do vídeo, o personagem começa sem olho, porém o mesmo é gerado de forma precisa antes do movimento. Formaram-se distintos pulos no mesmo vídeo, cujos braços e as pernas ficavam de maneiras diferentes. Além disso, as mãos se deformam ao longo do vídeo.

No teste posterior, tentou-se incluir o deslocamento horizontal do pulo na animação através de um prompt que instruía a edição do vídeo já gerado sem anexo de uma nova imagem de referência. O resultado³⁶ não se moveu para o lado, porém apresentou menos deformações na mão, manteve o movimento do pulo constante e preciso, sem movimentos extras das pernas e dos braços. Além disso, a coerência com a *pixel art* continuou e apenas a consistência do bracelete diminuiu. Dessa forma, sua qualidade foi considerada melhor do que a da animação anterior.

Depois foi notado que o deslocamento deve ser realizado pela movimentação física do objeto no Unity, cujo *sprite* fica mudando de acordo com o *frame*, de forma que sempre esteja centralizado com o colisor. Assim, não é necessário que o personagem se mova de um ponto A para um ponto B no vídeo gerado, pois essa movimentação apenas torna mais complicado o processo de manter o *sprite* visível na mesma posição do objeto.

Apesar do conteúdo já ser satisfatório, mais experimentos foram realizados em busca de um resultado com menos falhas. Porém, todos os vídeos gerados³⁷ apresentaram mais erros de consistência nas mãos e nos braceletes. Além disso, formaram-se movimentos exagerados durante o salto. Dessa forma, a Figura 78 foi considerada o melhor resultado. O *sprite sheet* do vídeo foi extraído através do ezgif. Essa imagem foi cortada de forma a conter a animação de um único pulo e o seu fundo foi removido, como pode ser vista na Figura 78b.

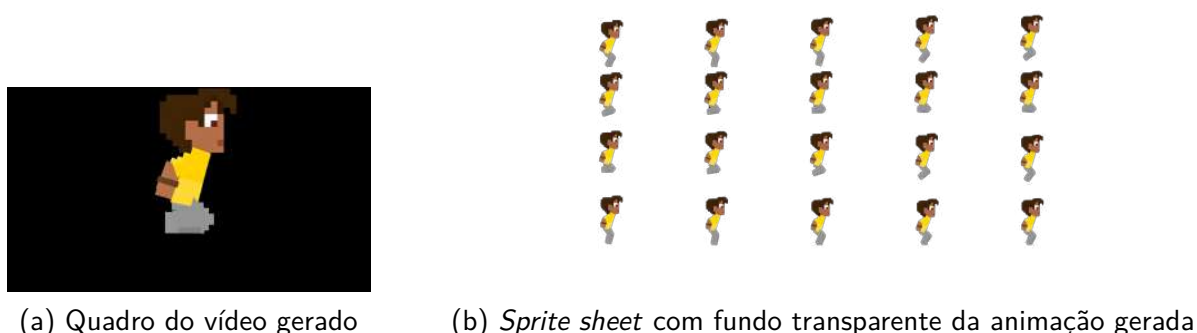
Analisando todos os resultados, é possível notar que o uso do *sprite* no padrão *pixel perfect* fez com que a ferramenta conseguisse parcialmente manter uma coerência com o estilo de *pixel art* durante a animação, fazendo quadrados diagonais em vez de uma reta diagonal em certas partes do corpo. Essa tentativa de manter a animação como *pixel art* possui suas imperfeições, sendo feita apenas uma reta diagonal de um pixel independente do ângulo.

³⁵ <https://drive.google.com/file/d/131HVD9P7_fZnAPsjHeYBHitE1ipKgKCG/view?usp=sharing>

³⁶ <<https://drive.google.com/file/d/1XS2euWjdv9dG-pvYUYqbuyQoUXlcfjfo/view?usp=sharing>>

³⁷ <https://drive.google.com/drive/folders/1nC_Mn9xHSIVC97XQD9yduLMBAohsexgx?usp=sharing>

Figura 78 – Melhor resultado da animação de pulo gerada no Gemini Pro



(a) Quadro do vídeo gerado

(b) *Sprite sheet* com fundo transparente da animação gerada

Fonte: Elaborada pela autora.

B.11.6 Geração da animação do personagem abrindo a porta

Foi criada uma animação do personagem abrindo a porta. Como só era possível anexar uma imagem, a ideia era criar a animação apenas do movimento do personagem, sem a porta realmente presente em cena. Diversas tentativas de prompt foram feitas, todas indicando que a porta não deveria aparecer durante o vídeo. Porém, todos os resultados³⁸ desenharam a porta a ser aberta, com distintos níveis de precisão do movimento, consistência do personagem e coerência da animação com o estilo. Vale ressaltar que as portas geradas não possuíam nenhuma animação de abertura, apenas sendo deslizadas para o lado até saírem da tela. Analisando esses resultados, foi notado que quando a instrução descrevia com mais detalhes a ação de abrir a porta, mais a IA ficava criativa durante esse movimento, a ponto de gerar ações extras desnecessárias para o personagem.

Posteriormente, foi descartada a necessidade dessa animação ser criada, por causa do funcionamento e ambiente do jogo, cujo personagem não conseguiria interagir com um objeto que está ao lado dele, apenas em frente. Em vez disso, a porta em *side view* se abre automaticamente quando o *sprite* do personagem se aproxima, sem nenhuma animação extra do personagem. Porém, uma das animações geradas apresentou o personagem andando antes de abrir a porta, com um movimento extremamente preciso utilizando a versão final do *sprite*, como pode ser visto na Figura 79a. Esse vídeo foi visto como candidato para ser usado na funcionalidade Animação para Animação do Pixel Lab (detalhada na Seção B.8.2) com o objetivo de gerar uma animação de caminhada mais consistente.

Para isso, foi extraído o *sprite sheet* do vídeo usando o ezgif, que depois foi transformado para o padrão *pixel perfect* através do Pixilart, selecionando apenas o trecho de interesse da animação e removendo o fundo. As Figuras 79b e 80 demonstram o processo da transformação do *sprite sheet*.

³⁸ <<https://drive.google.com/drive/folders/1b-mW7EVslZTTQWzP5O5M7vZzSb3UrFpH?usp=sharing>>

Figura 79 – Resultado da animação de abrir porta gerada no Gemini Pro



(a) Quadro do personagem andando durante a animação de abrir porta

(b) *Sprite sheet* da animação de abrir porta gerada no Gemini Pro

Fonte: Elaborada pela autora.

Figura 80 – Imagem após remoção do fundo



Fonte: Elaborada pela autora.

B.11.7 Geração da animação das portas abrindo

Foram criadas animações dos *sprites* de diferentes portas (Figuras 3c e 60c) abrindo, em pontos de vista distintos. Durante os testes iniciais, os experimentos focaram na produção da animação da porta A em *front view* (Figura 3c). O prompt usado inicialmente foi simples e direto, apenas apontando o que é o *sprite* e requisitando a animação específica. Baseado nos erros do resultado gerado, foram adicionados detalhes mais específicos na instrução com o intuito de corrigir os erros. A consistência do *sprite* se manteve alta durante todo o experimento, apesar de não apresentar uma animação coerente com o estilo *pixel art* e os resultados iniciais apresentarem imprecisões no movimento por causa da interpretação incorreta sobre a posição da maçaneta, e o lado para o qual a porta deveria abrir.

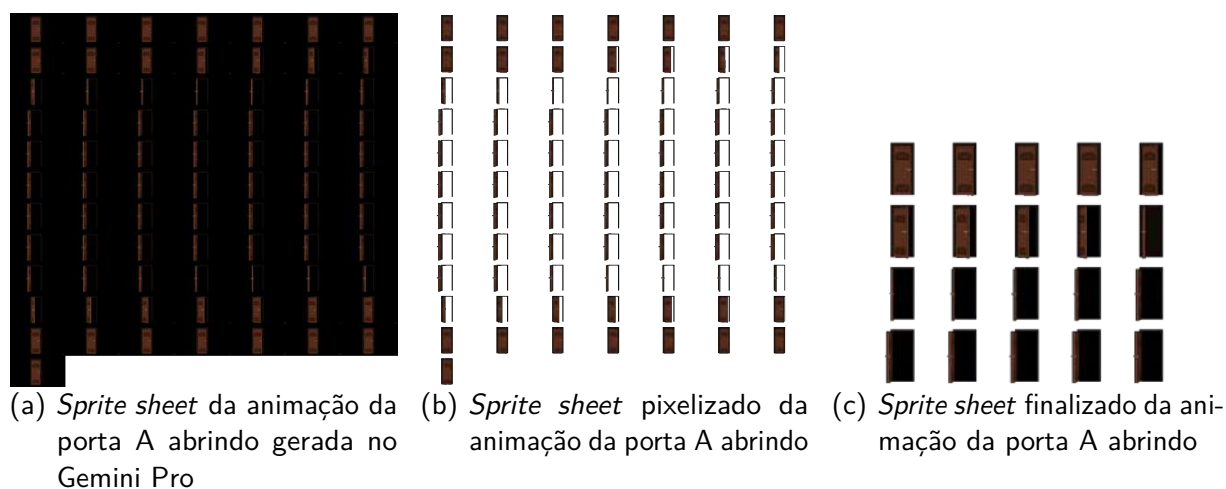
Essa estratégia mostrou-se extremamente efetiva, sendo possível gerar uma animação satisfatória em apenas três resultados³⁹. O *sprite sheet* do melhor resultado foi obtido pela ferramenta ezgif, como pode ser visto na Figura 81a. O fundo foi removido utilizando a ferramenta Photoroom⁴⁰ e o resultado foi transformado em *pixel perfect* pela ferramenta Pixilart (Figura 81b).

Nesse mesmo editor, foi aperfeiçoada a animação para manter a porta com tamanho

³⁹ <https://drive.google.com/drive/folders/10Whp-LvVw9JMa_Pc7ZjOHjllu-G5htA6?usp=drive_link>

⁴⁰ <<https://app.photoroom.com/create>>

Figura 81 – Melhor resultado da porta A abrindo



Fonte: Elaborada pela autora.

correspondente ao do *sprite* original durante todo o movimento de abertura. Isso foi feito colocando o *sprite* da porta fechada lado a lado com o primeiro quadro da animação e adicionando modificações conforme às mudanças dos *frames*. A Figura 81c apresenta o *sprite sheet* final da animação.

Esse processo mostra como a IA é apenas uma ferramenta para auxiliar o desenhista na produção da animação, criando uma base para ser editada e aprimorada pelo artista, fazendo com que ele não tenha que desenhar cada *frame* do zero e oferecendo uma referência visual personalizada de como cada quadro da animação deve ficar, de forma a ainda serem permitidas customizações e detalhes específicos visionados pelo desenhista.

A fase de testes posterior focou na geração da animação da Porta B abrindo em *side view* (Figura 60c). Durante o jogo, a porta vai abrir para o lado oposto ao personagem e para a esquerda do mesmo, de forma que o *sprite* da porta ficaria na frente do *sprite* do personagem quando ele passar por ela. Uma das grandes dificuldades durante essa etapa foi conseguir explicar de maneira clara como a porta deveria abrir.

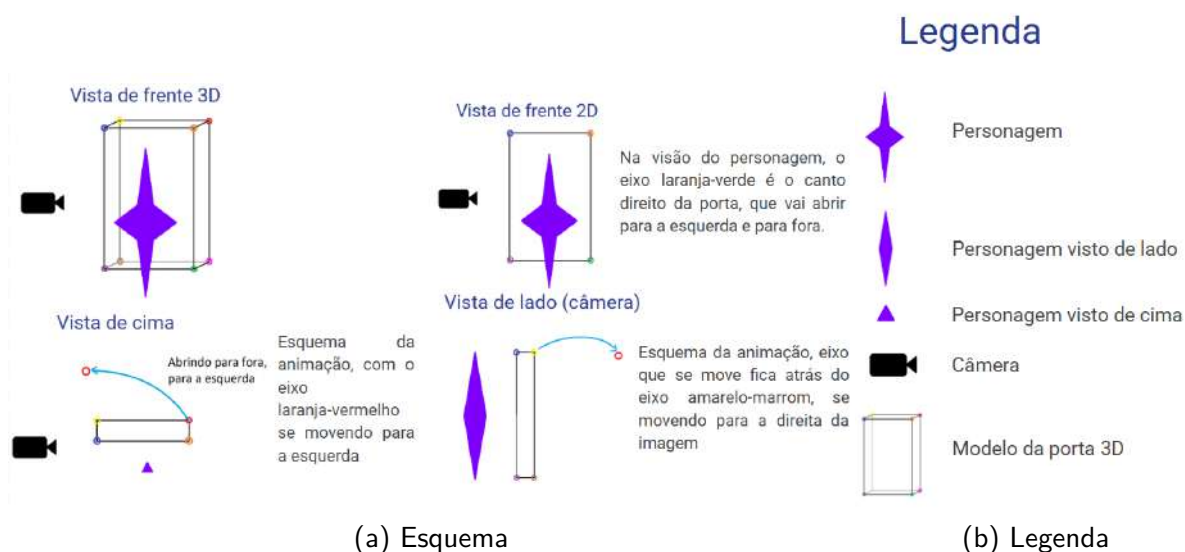
Apesar da porta abrir para a esquerda do personagem, no prompt foi instruído para a porta abrir para a direita. O motivo disso foi que o lado esquerdo do personagem equivale ao lado direito da imagem em *side view*, como detalhado na Figura 82

O prompt inicial era focado em descrever a porta, o ambiente e o quadro final. O resultado⁴¹ apresentou erros na compreensão do *sprite*, formando uma porta dupla e em vista frontal ao mesmo tempo em que a abertura ocorria. A instrução foi ajustada, removendo qualquer trecho que gerasse ambiguidade, e mais testes foram realizados. O novo vídeo⁴² gerado corrigiu a inconsistência do *sprite*, porém criou um movimento impreciso, realizando a

⁴¹ <<https://drive.google.com/file/d/1bqLKpgjRTf3mpunUhvB8464SkbnOmlh/view?usp=sharing>>

⁴² <<https://drive.google.com/file/d/1TkVeqaEhS6mMNMmAYHgM2NDxr674SkwTs/view?usp=sharing>>

Figura 82 – Esquema mostrando animação da porta abrindo em diferentes ângulos

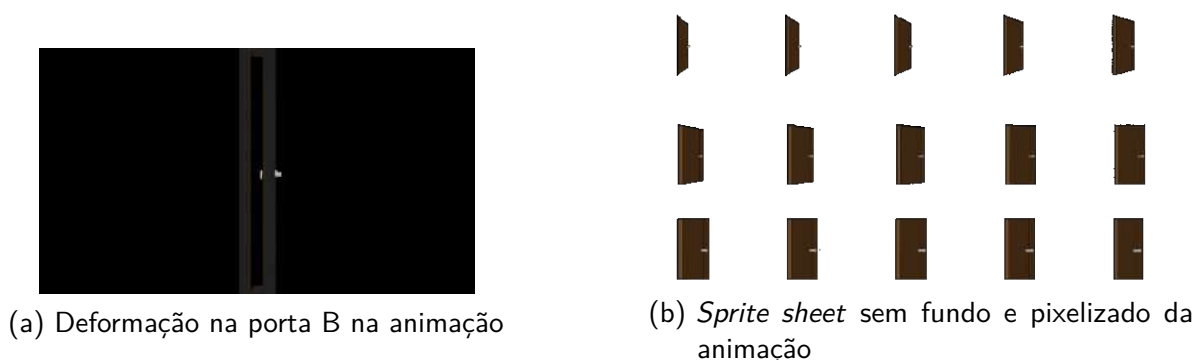


Fonte: Elaborada pela autora.

abertura em uma direção distinta da desejada.

Mais edições foram feitas ao prompt, focando na descrição da animação e na especificação da direção de cada movimento. Os resultados⁴³, porém, continuaram a apresentar as mesmas falhas na precisão do *sprite* e do movimento. Após mais algumas tentativas, a instrução visou forçar a IA a manter o eixo frontal fixo, além de repetir a descrição do movimento. A maior parte dos vídeos gerados⁴⁴ continuou a não formar a movimentação precisa. Porém, uma das animações⁴⁵ fez a abertura na direção correta após uma rápida deformação (Figura 83a), que poderia ser removida com uma leve edição.

Figura 83 – Melhor resultado da porta B abrindo no Gemini Pro



Fonte: Elaborada pela autora.

Para isso, foi extraído o *sprite sheet* do vídeo, utilizando a ferramenta ezgif. Essa

⁴³ <https://drive.google.com/drive/folders/1fwdzRZ7PL-uzQBVbJo5bKBHL_hjchn1V?usp=sharing>

⁴⁴ <<https://drive.google.com/drive/folders/1PQiVTqdHLN8fW6p62TypZ2la-WW8ynjs?usp=sharing>>

⁴⁵ <https://drive.google.com/file/d/1bBW7_HzSdrtaU5TI3Lj8auSoJJEMxYSC/view?usp=sharing>

imagem foi convertida para o padrão *pixel perfect* através do Pixilart, no qual também foi removido o fundo e cortado o trecho para manter apenas os *frames* sem deformação, como mostra a Figura 83b. Após isso, a animação é exportada para o Pixel Lab para mais ajustes (detalhado na Seção B.8.3).