

Char Sheets Generator: Em direção a uma ferramenta para geração automática de animações frame a frame

Vinícius Dantas Januário¹, Felipe Calegario¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

vdj@cin.ufpe.br, fcac@cin.ufpe.br

Abstract. *Creating animations for 2D games using frame-by-frame animation technique is a process that can be time-consuming and costly for game developers. This article presents a prototype of a tool built to automate this process. The tool uses a latent diffusion model published on the HuggingFace platform as a basis for generating pixel art characters from an input image. The tool was evaluated by a group of six professionals working in a game development studio in Recife. The results brought important considerations for the construction of a robust tool capable of automating this process.*

Resumo. *Criar animações para jogos 2D utilizando a técnica de animação frame a frame é um processo que pode ser demorado e custoso para desenvolvedores de jogos. Este artigo apresenta um protótipo de uma ferramenta construída para automatizar esse processo. A ferramenta utiliza um modelo de difusão latente publicado na plataforma Hugging Face como base para gerar personagens em pixel art, a partir de uma imagem de entrada. A ferramenta foi avaliada por um grupo de seis profissionais atuantes em um estúdio de desenvolvimento de jogos em Recife. Os resultados trouxeram considerações importantes para a construção de uma ferramenta robusta capaz de automatizar esse processo.*

1. Introdução

Desenvolver animações frame a frame é um processo demorado e custoso [Shuter 2020]. Dentro do contexto de desenvolvimento de jogos 2D para os tempos atuais, essa técnica de animação já não é mais predominante, tendo em vista outras possibilidades como o rigging 2D. Porém, para jogos com o estilo de arte tradicional (ex. pixel art), ou para quando se deseja ter mais controle sobre o movimento, essa técnica de animação costuma ser utilizada. Para desenvolvedores de jogos indie que possuem um orçamento baixo, o uso dessa técnica pode ser crítico, tendo em vista a quantidade de tempo que poderia estar sendo gasto para outras funcionalidades do jogo ou da quantidade de recursos financeiros para contratar animadores experientes. Tendo em vista essas limitações, esse trabalho apresenta a construção de um protótipo para uma ferramenta que gera animações (spritesheets) de personagens a partir de uma imagem fornecida pelo usuário, através de um arquivo ou desenho em canvas.

Para a geração das imagens essa ferramenta utilizou o PixelartSpritesheet_V.1 [Onodofthenorth 2022], um modelo de difusão latente construído usando o Dreambooth [Ruiz et al. 2022], uma técnica utilizada para customizar modelos de difusão latente

[Rombach et al. 2021]. Para tirar melhor proveito do modelo e criar a ferramenta foi necessário explorar diversas funções de geração, dentre elas a text-to-image, image-to-image e inpainting. A partir da função image-to-image se obteve os melhores resultados, possibilitando a criação de um pipeline de geração para uma simples aplicação usando a biblioteca gradio.

A ferramenta foi avaliada através de uma sessão de testagem realizada com um grupo de 6 pessoas que trabalham em diferentes segmentos da indústria de jogos (Desenvolvimento de Software, Arte e Quality Assurance). Após o uso da ferramenta, os usuários responderam a questionários e a uma entrevista semi-estruturada, utilizados para coletar dados quantitativos e qualitativos para a avaliação. Os métodos de análise quantitativos utilizados foram o SUS [Brooke 1995] e o UEQ [Laugwitz et al. 2008]. Para a análise qualitativa, foi utilizada a análise temática [Braun and Clarke 2006] em conjunto com métodos de codificação [Saldaña 2009]. A partir dos resultados obtidos, discutimos as limitações da ferramenta e suas correções necessárias, bem como seus aspectos positivos e o que pode ser adicionado para uma iteração mais robusta da ferramenta.

2. Problema

Este trabalho aborda o problema de geração de animações de personagens para jogos 2D feitos frame a frame, que no seu contexto de desenvolvimento é uma atividade intensiva e que demanda muito esforço [Hong et al. 2019] [Coutinho and Chaimowicz 2022]. É necessário desenhar diversas imagens representando cada etapa de movimento desejado ao personagem, e que escala na medida em que mais tipos de movimentos são requisitados (i.e. andar, pular, cair, etc). É um processo que costuma demorar e passar por várias revisões, sendo assim uma atividade repetitiva e exaustiva.

Além disso, é uma técnica que pode ser custosa, levando em conta a necessidade de se ter uma pessoa com boa experiência em animação no projeto de desenvolvimento [Shuter 2020], uma dificuldade que desenvolvedores indies e pequenos estúdios costumam encontrar devido a limitação de recursos.

Devido ao alto custo e complexidade, outras técnicas de animação acabam sendo mais vantajosas para esses desenvolvedores, como por exemplo o rigging 2D, na qual apenas um frame e um esqueleto do personagem precisa ser elaborado para realizar o seu movimento [Adobe 2022].

Apesar dessas vantagens, a animação frame a frame costuma ser necessária quando um jogo tem um estilo de arte mais tradicional e retrô, ou caso queira se ter um maior controle criativo sobre as animações [Shuter 2020].

Uma ferramenta que fosse capaz de automatizar esse processo, gerando rapidamente cada frame de uma animação específica, e que fosse fiel a uma descrição textual ou visual que represente o personagem que se deseja animar, poderia contribuir significativamente no ganho de tempo e na redução de custos para um projeto de desenvolvimento de um jogo.

3. Estado da arte

Nesta seção, será discutido o atual estado da arte para a geração de animações 2D através de IA e a inspiração para a solução proposta neste trabalho.

3.1. Soluções que utilizam GAN's

Abordagens utilizando redes adversárias gerativas (GANs) propõem modelos que geram animações, no qual a rede é treinada para receber uma imagem como entrada e devolver um spritesheet.

Coutinho e Chaimowicz [Coutinho and Chaimowicz 2022] descrevem uma rede baseada na arquitetura Pix2Pix que recebe uma pixel art frontal de um personagem e a rotaciona para uma das outras quatro direções (esquerda e.g.). Em seus experimentos com pequenos datasets produziram modelos com diversos graus de generalização, sendo o melhor deles em um dataset que foi construído de maneira restrita, mas que demonstrou resultados quase próximos da realidade no conjunto de testes.

Já Hong, Kim e Kang [Hong et al. 2019] propuseram uma arquitetura Multi Discriminadora (MDGAN) que recebe como entrada duas imagens, uma com informação sobre a animação desejada, em forma de esqueleto da personagem, e outra com informação sobre a cor e o corpo da personagem. O modelo proposto foi comparado com modelos de outras arquiteturas, a arquitetura MDGAN mostrou que esse processo de tradução image-to-image pode ser utilizado para criar animações, ainda que necessário mais exemplos de treinamento e informações adicionais de entrada como o esqueleto.

3.2. Modelos de difusão latente

Ferramentas de geração de imagem como Midjourney (www.midjourney.com) e front-ends para modelos de difusão latente (Stable Diffusion) [Rombach et al. 2021] podem ser utilizados na geração de assets em jogos. Tais ferramentas são capazes produzir imagens de alta qualidade a partir de texto e imagem. A geração das imagens no Stable Diffusion ocorre através de diminuição de ruído (denoise), no qual o modelo aprende a transformar ruído gaussiano aleatório em uma imagem dentro de uma representação latente, sendo esse processo condicionado à entrada textual fornecida.

É possível utilizar essas ferramentas para criar assets como sprites, backgrounds, ícones, retratos, personagens e entre outros [Ebert 2023]. Porém, a criação de animações com esses modelos só é possível a partir de outras técnicas de animação. A partir de um sprite de personagem gerado por IA, por exemplo, podemos animá-lo através do rigging 2D. Entretanto, essa etapa precisa ser realizada manualmente através de um software de animação 2D.

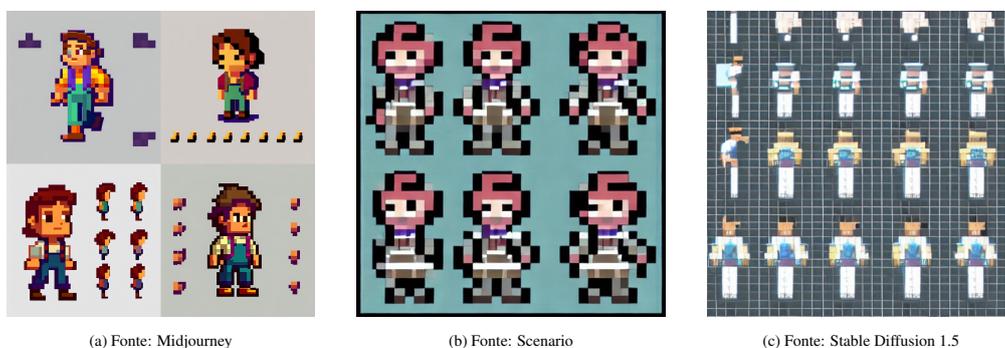


Figura 1. Spritesheets gerados por modelos de difusão

Utilizar apenas esses modelos para geração de animação frame a frame não traz

resultados satisfatórios. Como ilustra a Figura 1, se condicionados a prompts como “spritesheet”, “animation”, “pixelart” ou “2D”, os frames das imagens geradas aparecem em posições inconsistentes e sem espaçamento adequado, além de não existir a impressão de movimento e nem como visualizá-lo nessas ferramentas.

Existem também serviços gerativos, como o Scenario (www.scenario.com) e o Layer.ai (www.layer.ai), dedicados à geração de assets para games, e que provavelmente usam modelos derivados do Stable Diffusion utilizando técnicas de customização de modelos como o Dreambooth, que será discutido mais adiante [Ebert 2023]. Além de disponibilizar modelos com diferentes estilos para geração dos assets, essas plataformas permitem que o usuário treine modelos com o seu próprio estilo. Outra funcionalidade é a possibilidade de gerar essas imagens dentro de um canvas, funcionando como um editor de imagem através de IA, possibilitando que o usuário tenha um controle maior sobre as imagens geradas. Por disponibilizarem modelos treinados em diversos estilos, essas ferramentas conseguem gerar imagens consistentes e especializadas para a construção de assets. Porém, esses modelos ainda compartilham das inconsistências e dificuldades já apontadas ao se gerar animações.

3.3. Modelo de pixel art: PixelartSpritesheet_V.1



Figura 2. Imagens geradas pelo modelo PixelartSpritesheet_V.1

Um modelo produzido e publicado na plataforma Hugging Face pelo usuário Onodofthenorth [Onodofthenorth 2022] é capaz de gerar spritesheets de personagens animados em pixel art andando para quatro direções, para frente, para a direita, para trás e para esquerda, através de seus respectivos identificadores textuais (PixelartFSS, PixelartRSS, PixelartBSS e PixelartLSS). Cada identificador, se usado no prompt textual, define o tipo de animação a ser gerada, como ilustra algumas imagens da Figura 2.

Esse modelo foi produzido através do Dreambooth [Ruiz et al. 2022], que é uma técnica utilizada para personalizar modelos a partir de um conjunto de imagens de exemplo. Nesse processo, o modelo é treinado com poucas imagens para aprender um novo conceito, que é associado a um identificador textual único. Com isso, o modelo re-treinado com o Dreambooth consegue gerar imagens desse novo conceito aprendido em

contextos do domínio de saída do modelo original. Segundo Ruiz [Ruiz et al. 2022], 3 a 5 imagens são o suficiente para treinar o conceito, o que foi subvertido no modelo aqui apresentado. Segundo o que o autor relatou na plataforma Hugging Face e em postagens em outras redes sociais, ele foi treinado com 224 spritesheets, sendo 56 para cada conceito, utilizando personagens de mods do jogo Stardew Valley e chegando a utilizar até 16000 passos durante o treinamento, diminuindo a taxa de aprendizado a cada 3000 passos. As imagens geradas por esse modelo mostram que muito pouco do que era gerado no modelo original é levado em consideração, evidenciando um possível sobreajuste (overfitting) no modelo produzido. Essa característica de certa forma contribui para o modelo gerar animações com uma qualidade considerável e um espaçamento regular entre os frames, porém é algo que limita o controle do output gerado a partir do input textual, como será discutido nas experimentações da sessão a seguir. Esse modelo serviu como inspiração e base para o protótipo desenvolvido neste trabalho, se tratando de uma ferramenta de geração de animações.

4. Proposta de solução

Visando avançar na solução do problema discutido, este trabalho propõe elaborar e validar um protótipo de aplicação que possa automatizar o processo de criação de animações. A seguir vamos relatar o processo de construção do protótipo dividido em 3 partes. A Primeira aborda o processo de experimentação com o modelo anteriormente citado, o PixelartSpritesheet_V.1 [Onodofthenorth 2022], que foi necessário para chegar a um fluxo de geração adequado para o protótipo. Na segunda parte, será descrita as principais funcionalidades e elementos de interface da aplicação e por fim uma descrição da arquitetura e das tecnologias usadas, descrevendo o pipeline de geração dos assets.

4.1. Experimentação

Em todas as etapas do processo de experimentação foi utilizado o Stable Diffusion web UI [AUTOMATIC1111 2022], ferramenta desenvolvida pelo usuário AUTOMATIC1111 e publicada na plataforma Github. Essa ferramenta disponibiliza diversas features utilizadas para gerar imagens e treinar novos modelos com o Stable Diffusion.

Em um primeiro momento a experimentação se deu apenas com a função text-to-image, testamos os diferentes identificadores em conjunto com algumas descrições textuais, utilizando o Euler Ancestral ou o DDIM como Scheduler, e variando outros hiper-parâmetros de geração como a quantidade de passos (Steps) e CFG Scale (Classifier Free Guidance Scale).



Figura 3. Imagens geradas pela função text-to-image com prompt “PixelartRSS, a woman wearing a red dress”

Foi observado nos resultados, alguns ilustrados na Figura 3, que há uma constante inconformidade com o prompt de geração. Todas as imagens da Figura 3 foram geradas com a mesma configuração hiper parâmetros (prompt descrito na legenda) e seeds distintos. Apesar de haver uma imagem em conformidade com o prompt (Figura 3a) o restante das imagens apresentam poucas semelhanças semânticas. A partir desse ponto se optou por outras funções de geração de imagem que serão discutidas nas próximas subseções.

Antes de relatar a experimentação com essas funções, vamos descrever brevemente a influência de alguns dos hiper-parâmetros de geração das funções image-to-image:

- **Prompt:** Entrada textual que condiciona a geração.
- **Classifier Free Guidance Scale:** Controla a influência do prompt textual no processo de geração.
- **Steps:** Quantos passos do processo de diminuição de ruído aplicar na imagem.
- **Denoising strength:** Valor entre 0.0 e 1.0, controla a quantidade de ruído que é adicionado na imagem inicial. Em 0.0 a imagem gerada não é alterada com relação a imagem inicial, em 1.0 se permite mais variação que pode produzir imagens que não possuem uma consistência semântica com a imagem inicial.
- **Scheduler:** Algoritmo utilizado no processo de adição de ruído.

4.1.1. Inpainting

O inpainting é uma função capaz de gerar imagens a partir da modificação de uma região específica da imagem de entrada, essa região é delimitada por uma máscara, uma imagem monocromática que especifica as regiões a serem alteradas. A interface web para Stable Diffusion permite você desenhar a máscara na própria imagem ou fazer o seu upload. Na experimentação com o inpainting, a intenção foi observar se era possível gerar animações que se assemelham com o primeiro frame da imagem ao se ocultar os 3 últimos frames, correspondente a máscara desenhada na imagem como indicado na Figura 4a na região em preto, e na Figura 4b que representa apenas a máscara da geração, na região em branco.

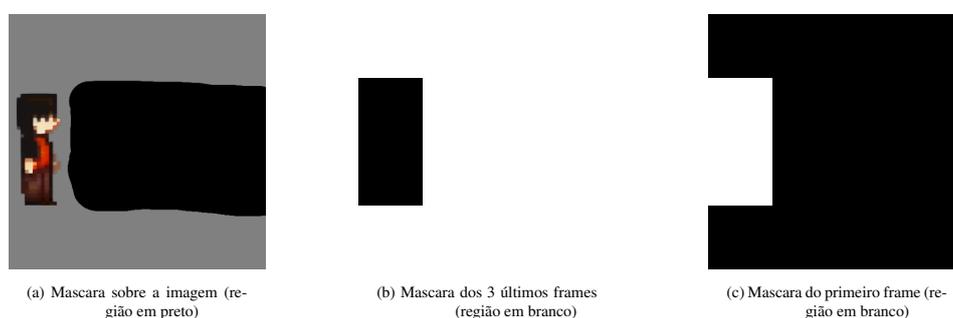


Figura 4. Máscaras utilizadas na experimentação

Como imagem de entrada foram testadas tanto frames que foram gerados pelo modelo ou frames que foram desenhados manualmente, como nas respectivas figuras 6 e 7. Nessa etapa, não houve uma experimentação muito a fundo dos hiper-parâmetros, se limitando ao uso do DDIM como scheduler e valores altos de denoising strength (entre 0.9 e 1.0), posteriormente foi verificado que a falta dessa experimentação no Scheduler fez

com que não se conseguisse tirar melhor proveito desse fluxo de geração. Imagens com denoising strength mediano (0.6) e DDIM como Scheduler produziam imagens borradas como ilustrado na Figura 5.



Figura 5. Imagem borrada, Scheduler=DDIM e Denoising Strength=0.6

Apesar dessas limitações, seguindo essa abordagem se obteve resultados como exemplificado nas figuras 6 e 7. Muito por conta do problema já mencionado (denoising strength alto), os resultados das duas figuras apresentaram resultados com o mesmo comportamento encontrado na função text-to-image, algumas imagens se conformam com melhor com o frame de entrada (figuras 6b, 6c e 7c), enquanto que outras não (figuras 6a, 6d e 7b).



Figura 6. Imagens geradas com a máscara de animação, utilizando como base frames gerados pelo modelo

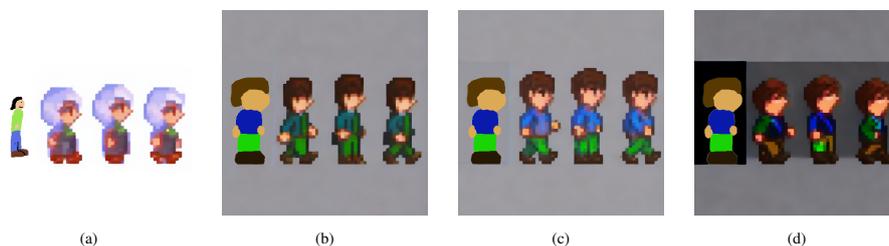


Figura 7. Imagens geradas com a máscara de animação, utilizando como base desenhos feitos manualmente

Em um segundo momento, se optou por testar essas imagens resultantes ocultando a região do primeiro frame (utilizando a máscara da Figura 4c), que obteve resultados que se conformaram melhor com os outros frames da imagem, como ilustrado na Figura 8. Para completar apenas um frame da animação essa técnica produziu bons resultados, porém a dificuldade de trazer essa conformidade para a animação como um todo fez com que se experimentasse outro fluxo de geração.

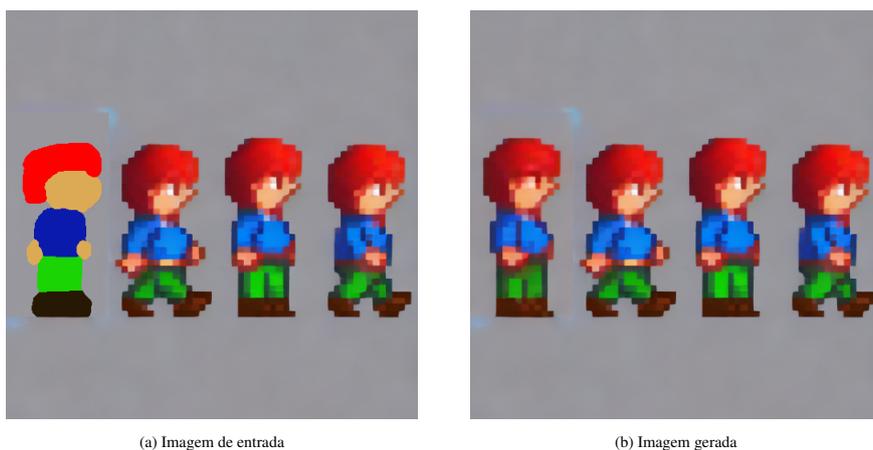


Figura 8. Geração utilizando máscara do primeiro frame (Figura 4c)

4.1.2. Image-to-image

A outra abordagem pensada para a geração das imagens foi utilizando a função image-to-image, que é semelhante ao inpainting, porém ela gera uma nova imagem e não se restringe a uma parte específica da entrada. Em um momento da experimentação percebemos que utilizar essa função, com a entrada sendo o desenho posicionado no primeiro frame da animação e um denoising strength mediano (0.5), como ilustrado na Figura 9, produziu uma imagem com um frame que se assemelha com as principais características do frame original. Esse resultado forneceu uma pista sobre como forçar o modelo a produzir toda a animação: replicar o frame de entrada em cada uma das posições em que são geradas os frames no modelo, em uma imagem de fundo cinza, como ilustra a Figura 10.



Figura 9. Geração utilizando a função image-to-image

Esse processo possibilitou a realização de experimentos com valores de denoising strength mais baixo. A seguir será apresentado o resultado de uma parte do estudo realizado avaliando a influência de cada um dos hiper-parâmetros na geração das imagens ao se utilizar essa técnica, em um caso usando um frame de uma imagem gerada pelo modelo (Figura 10a) e em outro usando um desenho elaborado manualmente (Figura 10b).



(a) Frame gerado pelo modelo

(b) Desenho feito a mão

Figura 10. Imagens de entrada com frames replicados



**Figura 11. Teste de Denoising Strength 1: Prompt="PixelartRSS",
Sampler=Euler a, CFG scale=7, Steps=20, Seed=4131043249**



**Figura 12. Teste de Denoising Strength 2: Prompt="PixelartRSS",
Sampler=Euler a, CFG scale=7, Steps=20, Seed=663664100**



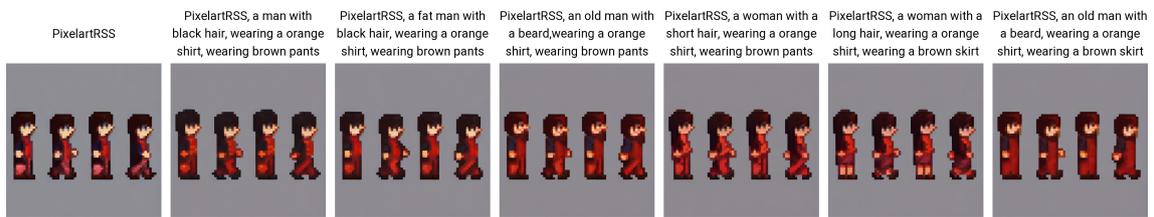
**Figura 13. Teste de Scheduler 1: Prompt="PixelartRSS",
Denoising Strength=0.65, CFG scale=7, Steps=20, Seed=4131043249**



**Figura 14. Teste de Scheduler 2: Prompt="PixelartRSS",
Denoising Strength=0.65, CFG scale=7, Steps=20, Seed=663664100**



**Figura 15. Teste de Prompt 1: Scheduler="Euler a",
Denosing Strength=0.65, CFG scale=7, Steps=20, Seed=4131043249**



**Figura 16. Teste de Prompt 2: Scheduler="Euler a",
Denosing Strength=0.65, CFG scale=7, Steps=20, Seed=663664100**



Figura 17. Teste de CFG Scale 1: Prompt="PixelartRSS, a man with brown hair, wearing a blue shirt, wearing a green pants", Denosing Strength=0.65, Scheduler="Euler a", Steps=20, Seed=4131043249



Figura 18. Teste de CFG Scale 2: Prompt="PixelartRSS, a man with black hair, wearing a orange shirt, wearing brown pants", Denosing Strength=0.65, Scheduler="Euler a", Steps=20, Seed=663664100

Com relação ao parâmetro denoising strength, constatamos o dilema entre ter uma imagem totalmente parecida com a original, que não gera uma animação, e uma imagem que possui movimento mas não tem correlação com o input original. Isso pode ser observado tanto nas figuras 11 e 12, onde o movimento não é aparente para valores de denoising mais baixo e para valores mais altos as cores originais de algumas partes do personagem vão aos poucos se degenerando. Para a geração das imagens valores na faixa de 0.6 mostraram ter um bom equilíbrio entre o movimento do personagem e o conteúdo original.

Sobre o prompt, as figuras 15 e 16 mostram que pequenos aspectos como o roupas, barba, cabelo e físico dos personagens podem ser alterados através do prompt, ainda que em alguns casos isso gere deformidades e que em termos de cor se prevaleça o conteúdo da imagem original. Para o CFG Scale percebemos que valores muito altos deformam as imagens geradas, como ilustrado nas últimas imagens das figuras 17 e 18. Por fim, as figuras 13 e 14 ilustram alguns dos Schedulers testados.

Dentre outros aspectos observados durante todo o processo de experimentação foi: a necessidade de se colocar um fundo cinza na imagem de entrada, pois a cor de fundo influencia na imagem gerada, como ilustrado anteriormente na Figura 7d, e no caso dos desenhos para uma boa geração da imagem o rascunho teria que se assemelhar com o formato das imagens geradas, onde gerações como a do desenho da Figura 7a mostrou poucos resultados semelhantes ao conteúdo original.

A tarefa de estabelecer um critério não subjetivo para a avaliação da qualidade das animações se mostrou difícil e não foi levada adiante nesse trabalho. Avaliando o resultado desses estudos, para o protótipo se optou deixar editável alguns desses hiper parâmetros com os seguintes valores padrão: Denoising Strength (0.65), CFG Scale (7.0), Steps (20) e Prompt ("PixelartRSS"). Para o Scheduler se optou por deixar fixo o algoritmo Euler Ancestral (Euler a). O protótipo e sua arquitetura serão discutidos nas próximas seções.

4.2. Protótipo desenvolvido

O protótipo desenvolvido consta com duas funcionalidades principais: Geração de animações a partir de uma imagem (via upload de arquivo) e geração de animações a partir de um desenho sobre um canvas. Ambas compartilham as mesmas opções de geração, que serão discutidas no decorrer desta seção.

Como ilustra a Figura 19, a aplicação possui duas abas, cada aba define qual modo de geração das animações, por upload de imagem ou por desenho. Elas estão localizadas no canto superior esquerdo e quando o usuário inicializa a ferramenta ele se encontra por padrão na aba de upload. Nessa função, o usuário pode inserir uma imagem que ele possua em sua máquina ou selecionar algumas das imagens já predefinidas que a aplicação dispõe como ilustra a Figura 21, que será discutida mais adiante. Logo após a imagem de input se localiza o botão Generate spritesheet em destaque (laranja), ele é o botão responsável por acionar a geração da animação.

Após esse botão se localiza as opções de geração, como ilustra a Figura 20. Cada opção se encontra com o seu respectivo valor padrão. Ela é comum as duas funções da aplicação e é dividida nos seguintes blocos:

Char Sheets Generator

The screenshot shows the top section of the Char Sheets Generator interface. It features two tabs: 'Upload image' (active) and 'Draw image'. Below the tabs is a large dashed box labeled 'Input image' with the text 'Drop Image Here - or - Click to Upload'. A prominent orange button labeled 'Generate spritesheet' is positioned below the input area. Underneath, there are several controls: a checked checkbox for 'Remove background from input image'; an 'Animation type' section with radio buttons for 'Left to right' (selected), 'Forwards', 'Right to left', and 'Backwards'; a 'Seed' input field containing '-1' and a checked 'Randomize seed' checkbox; and two buttons: 'Use last seed' and 'Reset generation values'. To the right, there are text input fields for 'Prompt' and 'Negative prompt'. At the bottom, there is an 'Advanced settings' section with a downward arrow.

Figura 19. Tela superior: Função de upload

This screenshot displays the 'Advanced settings' section of the Char Sheets Generator. It contains three sliders: 'Denoising strength' set to 0,65, 'Sampling steps' set to 20, and 'CFG Scale' set to 7. Below each slider is a descriptive text box: 'Denoising strength' explains that 0 means no change and 1 means an unrelated image; 'Sampling steps' notes that lower values can produce bad results and higher values take longer; 'CFG Scale' states that lower values are more creative and higher values may deform the image.

Figura 20. Tela mediana: Opções de geração

- **Remove background from input image:** Checkbox que dá a opção de remover o background da imagem fornecida como entrada.
- **Animation type:** Seleção única entre múltiplas opções (Radio), responsável por apontar qual tipo de animação será gerado, referente as 4 opções fornecidas pelo modelo.
- **Seed, randomize seed e Use last seed:** Conjunto que regula qual seed de geração será utilizado. Randomize seed é um checkbox que se estiver marcado a opção de seed estará fixa em -1, e o seed escolhido para a geração será aleatório. Se Randomize seed estiver desmarcado o usuário pode editar o seed utilizado na geração, que será sempre o que estiver no campo. Assim que o checkbox é desmarcado, um valor aleatório é sorteado para o seed. O botão Use last seed possibilita que o usuário fixe o último seed utilizado na geração.
- **Prompt e Negative prompt:** Caixas textuais que permite o usuário escrever as respectivas opções textuais de geração. O prompt, como já discutido anteriormente, atua condicionando o processo de geração com o seu valor semântico, enquanto que o Negative prompt atua de maneira reversa, condicionando a não aparição de seu valor.

Input image	Prompt	Negative prompt	Animation type	Seed	Randomize seed	Remove background from input image
	brown hair, blue shirt, green pants		Left to right	3467755142	false	true
			Left to right	2825266081	false	true
	orange shirt, brown pants		Left to right	23112	false	true
	a old man, blue hat, long white beard, dark robes		Left to right	249298562	false	true
	a man with brown hair, wearing an orange shirt, wearing blue pants		Forwards	3397323969	false	true
	a woman, purple shirt, blue overall		Forwards	1381991609	false	true
			Forwards	864049657	false	false
	black shoes, red skirt	green shirt	Forwards	864049657	false	true

Figura 21. Tela inferior: Função de upload

- **Reset generation values:** Botão que restaura todos os valores padrão de geração, que inclui toda a seção das opções de geração.
- **Advanced settings:** Possui 3 valores alteráveis por sliders e que representam os hiper-parâmetros de geração, Denoising strength, Sampling steps e CFG Scale, já discutidos na seção anterior a esta. Cada valor possui uma descrição textual localizada logo abaixo do slider, ela explica brevemente como atua o respectivo parâmetro na geração. Além disso, é importante destacar que todo o bloco de Advanced settings se encontra em um menu sanfonado (Accordion), que por padrão esconde as opções avançadas e mostra elas caso o usuário clique na região de Advanced settings.

A parte inferior da função de upload é composta por uma tabela de opções com inputs pré-definidos, como ilustra a Figura 21. Cada coluna representa um parâmetro da interface a ser subscrito e os elementos da tabela se dispõem verticalmente. Quando o usuário seleciona alguma das linhas as opções Input image, Prompt, Animation type, etc. são sobrescritas com os valores estabelecidos.

A função de desenho consta um tipo diferente de input, como ilustrado na Figura 22. Ela dispõe de um canvas editável com as mesmas proporções de um frame gerado pelo modelo. O usuário pode desenhar na região do canvas, editar a cor ou tamanho do pincel e reverter as ações no canvas, tudo através dos ícones localizados no canto superior direito.

Seguindo também o exemplo da Figura 22, assim que o usuário desenha, ajusta as opções de geração e clica no botão de gerar o spritesheet, uma região antes oculta aparece logo após o botão principal e antes das opções de geração. Essa região contém 4 saídas, a Raw spritesheet, que contém a saída bruta do modelo e sem nenhuma alteração, a Spritesheet, a mesma imagem só que com o fundo cinza removido, a Animation, um gif repre-



Figura 22. Tela superior: Função de desenho

sentando a animação gerada, e por fim um arquivo zip com uma opção de download, ele contém todas essas imagens de saída mais 4 imagens com cada frame separado. Enquanto a imagem é gerada e não está pronta, em cada uma dessas partes aparece uma animação de carregamento. Esse processo ocorre da mesma forma para a função de upload.

Por último na parte inferior da função de desenho, existe uma seção com 4 imagens para servir de referência ao usuário, como ilustra a Figura 23.

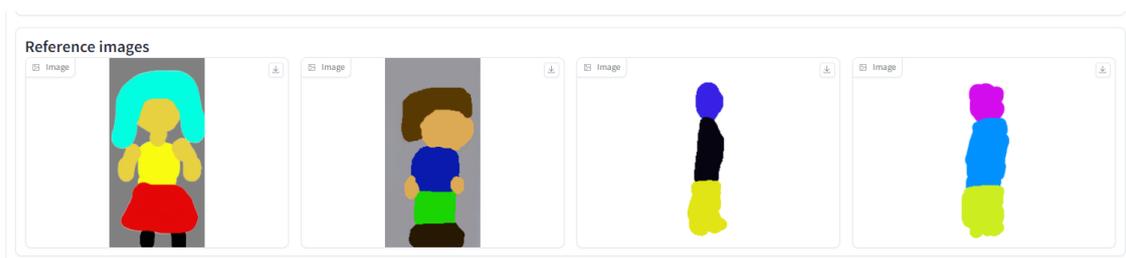


Figura 23. Tela inferior: Função de desenho

4.3. Implementação

Essa seção apresentará os aspectos técnicos relacionados à arquitetura da aplicação e o fluxo de geração das animações. Para o desenvolvimento da interface e processamento se utilizou o gradio [Abid et al. 2019], uma biblioteca Python utilizada para a criação de aplicações web voltada para demonstração de modelos de machine learning. Utilizamos a própria linguagem Python e outras bibliotecas para realizar as tarefas de processamento dos assets, que serão discutidas adiante. Para realizar a geração das imagens com modelo em questão, utilizamos a mesma ferramenta usada na fase de experimentação, a Stable Diffusion web UI [AUTOMATIC1111 2022], ela possibilita o seu uso através de uma API, possibilitando assim a comunicação com a aplicação gradio, que provê todas as informações necessárias para a geração. A imagem da Figura 24 ilustra a arquitetura da ferramenta.

A região em laranja representa a aplicação gradio, com essa biblioteca é possível definir em um script comum tanto os elementos de interface, que executam do lado do cliente e definem as ações da aplicação, quanto às funções que realizam o processamento

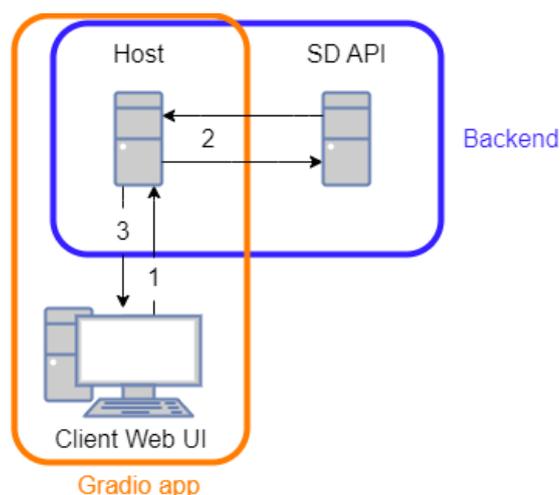


Figura 24. Arquitetura da aplicação

dessas mesmas ações definidas na interface. O host inicia a aplicação gradio, que expõe um link de acesso para uma interface web. Em azul estão destacados os endpoints que atuam no backend da aplicação. A seguir será apresentado o fluxo principal de geração de uma animação dentro da aplicação, como apontado na Figura 24:

1. **Processamento dos dados de geração:** Assim que o usuário interage com a interface e ativa a geração do spritesheet, os dados fornecidos como imagem de entrada e parâmetros de geração são pré-processados e enviados para host. As imagens são redimensionadas para um tamanho predefinido (128x256 pixels), o equivalente ao tamanho de um frame na imagem gerada. Em seguida o parâmetro Remove background define se o fundo do frame de entrada deve ser removido, caso sim a ferramenta gera novo frame sem fundo. Após essa etapa o frame é utilizado para construir a imagem de entrada a partir do processo já descrito na seção de experimentação: Gerar uma nova imagem com fundo cinza e replicar o frame nas posições relativas de cada frame na imagem a ser gerada (Figura 10). Além do processo de preparação da imagem de entrada, o prompt de geração é montado através da concatenação do atributo Animation type com o prompt textual inserido na caixa de entrada.
2. **Geração da imagem pela api:** Com os dados de geração processados, o host faz uma requisição post para o endpoint Stable Diffusion API, passando os valores de geração no corpo da requisição com o formato específico da API. Como resposta da requisição o host recebe a imagem gerada.
3. **Processamento dos assets:** Após a geração da imagem a última etapa é montar um arquivo zip com os assets gerados, que inclui o Raw input (imagem sem alterações), o spritesheet processado, frames individuais e animação (gif). Como ilustra a Figura 25, as imagens geradas possuem dimensão 512x512, os 4 frames são dispostos horizontalmente a distâncias aproximadamente iguais a 128 pixels. Verticalmente os primeiros 128 pixels e os últimos 128 pixels correspondem a espaço cinza vazio (região vermelha). Como a figura mostra, o Spritesheet vem de um recorte da área em verde, e para criar os frames individuais recortes das regiões de 128x256 pixels (regiões em azul) são feitos horizontalmente em sequência. Em seguida o gif de animação é montado com os frames individuais, todos os arquivos

citados são comprimidos em um único arquivo zip que compõem os outputs gerados pela aplicação. Por fim o Host retorna para a interface do cliente as imagens e o arquivo gerado.



Figura 25. Recortes de espaço vazio (vermelho), spritesheet (verde) e frames individuais (azul).

Para as tarefas de processamento de imagem foi utilizada a biblioteca Pillow, rembg para remoção de background, zipFile para zipar os arquivos e requests para a realização de requisições.

Duas dificuldades ficaram como pendências do processo de implementação: Ajustar o tamanho do canvas da função de desenho (Figura 22), algo que para nós era muito pequeno. A outra é que nem sempre o modelo gera imagens que se enquadram perfeitamente nos recortes apontados pela Figura 25, sendo recorrente o desenho de algum frame invadir o de outro subjacente, essa situação é exemplificada em alguns pixels que aparecem na borda direita da animação em um gif que aparece na Figura 22. Ambos os problemas serão abordados mais adiante na avaliação.

5. Avaliação

Para avaliar a ferramenta foram realizadas sessões de testagem com pessoas que atuam na indústria de jogos em 3 segmentos distintos: Desenvolvimento de Software, arte e QA (Quality Assurance). Além de avaliar a usabilidade do protótipo, nesse processo se buscou entender qual o valor de uso da ferramenta para cada um desses 3 perfis, buscando coletar feedbacks e validar a inclusão de novas funcionalidades.

Nas seções subsequentes serão descritas as metodologias de análise utilizadas, como foi estruturada as sessões de avaliação e os resultados obtidos.

5.1. Métodos de análise quantitativa

5.1.1. System Usability Scale (SUS)

O SUS (System Usability Scale) [Brooke 1995] é um questionário utilizado para medir a usabilidade de um sistema considerando 3 aspectos: Efetividade, eficiência e satisfa-

ção. Possui 10 afirmações onde o usuário responde numa escala de 1 a 5 se concorda ou discorda totalmente da afirmação. É um questionário curto e simples, recomenda-se que seja aplicado logo após o uso da ferramenta. O SUS produz um número (Score) de 0 a 100 para a usabilidade do sistema, sendo 68 uma média consolidada pela comunidade que aplica o questionário. Valores abaixo da média podem indicar a necessidade de rever a usabilidade do sistema. Consiste nas seguintes perguntas:

1. Eu acho que gostaria de usar este sistema frequentemente.
2. Eu achei o sistema desnecessariamente complexo.
3. Eu achei que este sistema foi fácil de usar.
4. Eu acho que eu precisaria do suporte de uma pessoa com conhecimentos técnicos para ser capaz de usar este sistema.
5. Eu achei que várias funções neste sistema estão bem integradas.
6. Eu achei que existiram várias inconsistências neste sistema.
7. Eu imagino que a maioria das pessoas aprenderia a usar este sistema rapidamente.
8. Achei o sistema muito complicado de usar.
9. Me senti confiante usando o sistema.
10. Eu precisaria aprender muitas coisas antes que eu pudesse me sentir a vontade em usar este sistema.

5.1.2. User Experience Questionnaire (UEQ)

O UEQ (User Experience Questionnaire) [Laugwitz et al. 2008] é um questionário utilizado para medir a experiência do usuário de produtos interativos. As escalas do questionário cobrem uma impressão abrangente da experiência do usuário, elas são:

- **Atratividade (Attractiveness):** A impressão do usuário sobre o produto, se ele gosta ou não.
- **Perspividade (Perspicuity):** O produto é fácil de se familiarizar e de aprender a usar.
- **Eficiência (Efficiency):** O usuário consegue resolver as tarefas sem esforço desnecessário.
- **Confiabilidade (Dependability):** O usuário consegue se sentir no controle da interação com o produto, ele é seguro e previsível.
- **Estimulação (Stimulation):** O produto é empolgante, motivador e divertido de usar.
- **Novidade (Novelty):** O design é criativo e captura o interesse do usuário.

O questionário divide algumas dessas escalas como pragmáticas (Perspividade, eficiência e confiabilidade), que dependem do objetivo final do produto, e hedônicas (Estimulação e novidade), que não dependem do objetivo final do produto. O questionário consiste de 26 itens, cada um é associado a uma das escalas avaliadas e cada item possui dois termos com significados opostos, o usuário responde numa escala de -3 a 3 onde -3 representa a resposta mais negativa, 0 a resposta neutra e 3 a resposta mais positiva.

Para o processamento e análise dos dados do questionário foi utilizada uma ferramenta de benchmark [Schrepp et al. 2017] que calcula as métricas (médias e variâncias) das respostas por escala e expõe os resultados em um gráfico. Esse benchmark oferece

uma classificação do produto para cada uma das escalas avaliadas pelo questionário, e a cada ano ele é atualizado. A Figura 26 ilustra essa classificação em um gráfico, ela é construída a partir intervalos percentuais do banco de resposta, que são representados por:

- **Excelente:** Entre o intervalo dos 10% melhores.
- **Bom:** 10% dos resultados são melhores e 75% dos resultados são piores.
- **Acima da média:** 25% dos resultados são melhores e 50% dos resultados são piores.
- **Abaixo da média:** 50% dos resultados são melhores e 25% dos resultados são piores.
- **Ruim:** Entre o intervalo dos 25% piores.

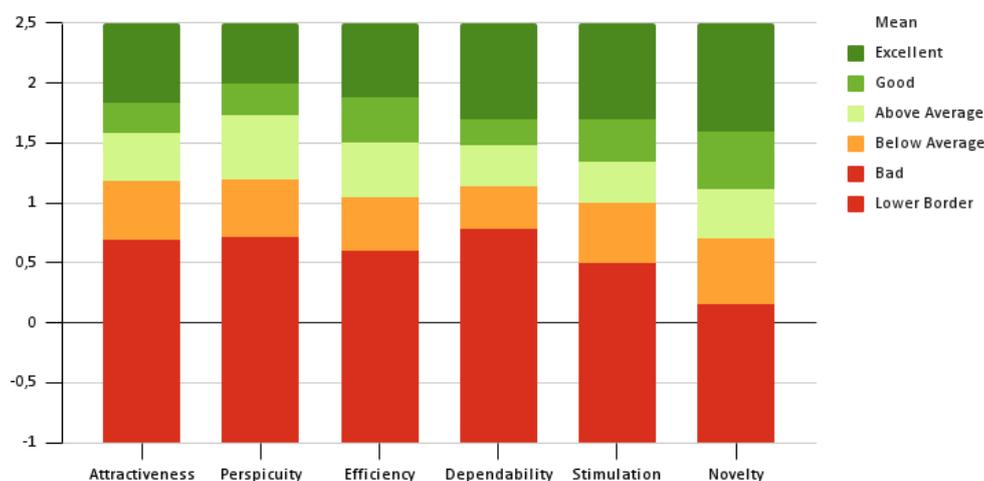


Figura 26. UEQ Benchmark

5.2. Métodos de análise qualitativa

5.2.1. Análise temática

A análise temática [Braun and Clarke 2006] é um método utilizado para analisar temas e padrões recorrentes em dados qualitativos. Durante o processo de análise se faz um agrupamento de respostas semelhantes que possuem origem distintas. Esse processo foi aplicado aos dados obtidos nas sessões de avaliação e os temas foram identificados a partir do que os participantes falaram, e não sobre uma interpretação do mesmo. Os temas encontrados serão discutidos na seção de resultados.

5.2.2. Métodos de codificação

Para auxiliar o agrupamento no método de análise temática, foram utilizados códigos [Saldaña 2009], que são palavras ou sentenças curtas que capturam a essência sobre uma porção de dado textual ou visual. Para este trabalho foram estabelecidos códigos de maneira prévia (Top-down) à análise dos dados e outros códigos emergiram durante o processo (Bottom-up). Esses códigos serão descritos na seção de resultados.

5.3. Sessão de avaliação

As sessões de teste foram realizadas remotamente no período de 22/03/2023 a 27/03/2023. A coleta de dados durante a sessão foi feita através de gravações em vídeo, aplicação de questionários (SUS e UEQ) e por fim uma entrevista semi-estruturada. A seguir será apresentado o perfil demográfico dos participantes e o roteiro seguido nas sessões de avaliação.

5.3.1. Participantes

As sessões foram realizadas com 6 participantes, todos trabalham em um mesmo estúdio de desenvolvimento de jogos de Recife em 3 áreas distintas: Desenvolvimento de software, arte e QA (Quality assurance). A tabela 1 a seguir mostra o perfil demográfico de cada participante.

Participante	Idade	Profissão	Tempo de experiência
P0 (D0)	22	Desenvolvedor de software de jogos	2 anos
P1 (Q0)	23	QA Tester	2 anos
P2 (A0)	26	Artista 2D/3D	4 anos (em jogos)
P3 (A1)	31	Artista de games	5 anos (1 ano em jogos)
P4 (Q1)	24	QA	2 anos (Como QA)
P5 (D1)	21	Desenvolvedor de jogos / full stack	4 anos

Tabela 1. Perfil dos participantes

Note que para além da identificação convencional (P0, P1, ..., etc.), foi designado um outro identificador para o participante destacando sua área de atuação: D para desenvolvedor, A para artista e Q para quality assurance. Essa identificação será predominantemente utilizada durante a análise.

5.3.2. Roteiro

O roteiro da sessão de avaliação é dividido em 4 partes: Apresentação, teste, aplicação de questionários e entrevistas. As tarefas da etapa de teste foram elaboradas pensando na cobertura de todas as principais funcionalidades e elementos da ferramenta. Para as perguntas da entrevista, elaboramos questionamentos visando compreender as forças e as fraquezas da ferramenta, bem como entender o valor de uso para o usuário e a opinião sobre a inclusão de novas funcionalidades. As 5 primeiras perguntas da entrevista foram adaptadas da Estratégia Blue Ocean para pesquisa. Na parte da entrevista os trechos entre parênteses representam perguntas ou falas opcionais, e perguntas prefixadas com {Artista} foram feitas apenas para os participantes que possuíam o perfil de artista. A seguir é apresentado o roteiro utilizado durante a pesquisa:

1. Apresentação

- 1.1 **Introdução:** Cumprimentar e agradecer o participante. Apresentar os principais pontos da seção: Teste, aplicação de questionários e entrevista. Pedir o consentimento da gravação da reunião e pedir o compartilhamento de tela apenas durante o uso da aplicação. Reforçar que a qualquer

momento o participante pode optar por não participar da pesquisa e que ele pode se sentir à vontade para falar e responder o que quiser durante a entrevista.

1.2 **Breve apresentação da ferramenta:** É dito: “O Char Sheets Generator é uma ferramenta de inteligência artificial, capaz de gerar spritesheets com animações de um personagem andando com um determinado estilo, a partir de uma imagem fornecida como entrada. Essa imagem pode ser um arquivo de imagem contendo o frame ou um próprio desenho que pode ser feito na aplicação.”

1.3 **Explicar o processo de testagem:** Explicar como vai ocorrer o processo de testagem. Dizer que a intenção é interferir o mínimo possível, mas caso necessário serão feitas intervenções para o andamento da testagem. Por fim, disponibilizar o link de acesso.

2. Testagem

2.1 **Exploração livre:** Navegar entre as duas abas principais e se familiarizar com os principais elementos da interface e usar a ferramenta como bem entender, pode ser feita de maneira livre dentro de 2 minutos.

2.2 **Usar a função de upload com um preset:**

- Dentro da função de gerar uma imagem a partir de upload, selecione uma das opções de geração pré-definidas e gere uma imagem.
- Após a geração, tente baixar e abrir as imagens geradas.
- Se você tentar gerar a imagem novamente, você vai observar que a mesma imagem será gerada. Altere as configurações para produzir uma imagem diferente a cada geração.
- Experimente gerar uma imagem com outro tipo de animação. Assim que achar que já experimentou o suficiente apenas avise.

2.3 **Usar a função de desenho:**

- Dentro da função de gerar uma imagem a partir de um desenho, experimente fazer um desenho simples de um personagem e gere spritesheets do mesmo. Para o desenho, você pode se basear em alguns dos desenhos de referência.
- Dentro dessa mesma imagem, experimente utilizar as funções textuais para alterar a geração da imagem. Você pode tomar como base alguns dos exemplos definidos na opção de upload. Assim que achar que já experimentou o suficiente apenas avise.

2.4 **Usar a função de upload experimentando as opções avançadas:**

- Volte para a função de upload. Agora tente gerar personagens a partir de alguma imagem de personagem 2D que você possua. Caso não tenha, utilize a sexta imagem (Maru Stardew Valley) dos exemplos.
- Se familiarize com as configurações avançadas e experimente alterar seus valores. Assim que achar que já experimentou o suficiente, apenas avise.

3. Aplicação de questionários (SUS e UEQ)

4. Entrevista

4.1 Para você, o que deveria ser eliminado da ferramenta (tem hoje, mas não deveria ter mais)?

- 4.2 Para você, o que deveria ser reduzido da ferramenta (tem hoje, mas deveria ter menos)?
- 4.3 Para você, o que deveria ser mantido na ferramenta (tem e deveria continuar do jeito que está)?
- 4.4 Para você, o que deveria ser ampliado na ferramenta (tem hoje, mas deveria ter mais)?
- 4.5 Para você, o que deveria ser adicionado à ferramenta (não tem hoje, mas deveria passar a ter)?
- 4.6 Você usaria essa ferramenta durante o processo de desenvolvimento de algum jogo? Se sim, como? (Quais seriam as possíveis dificuldades de integração?) (Quais seriam as vantagens de usar esse sistema?)
- 4.7 {Artista} Caso você possuísse apenas um frame inicial de um personagem e tivesse que animá-lo, como normalmente você faria?
- 4.8 {Artista} Caso você possuísse apenas uma ideia conceitual de um personagem e tivesse que criar a base ou um protótipo dele, como normalmente você faria?
- 4.9 Considerando o processo de criação usando apenas com a função de upload da ferramenta, como você avalia a qualidade dos assets gerados? Quais foram as dificuldades?
- 4.10 Considerando o processo de criação usando a função de desenho manual do personagem, como você avalia a qualidade dos assets gerados? Quais foram as dificuldades?
- 4.11 Quais ferramentas de trabalho você usa no cotidiano no desenvolvimento de jogos?
- 4.12 O que você acha sobre uma possível integração das funcionalidades da ferramenta apresentada com alguma ferramenta que você já use no cotidiano, caso use alguma (Game engine, editor de imagem, etc)?
- 4.13 (O que você pensa sobre a expansão de estilos e tipos de animação que a ferramenta suporta?)
- 4.14 Espaço para comentários, depoimentos e feedback sobre o uso da ferramenta.

5. **Finalização:** Coleta de dados demográficos e agradecer novamente o participante.

5.4. Resultados

Nesta seção serão apresentados os resultados obtidos com os métodos de análise quantitativa e qualitativa abordados.

5.4.1. Análise quantitativa

Com relação aos resultados do questionário SUS, a tabela 2 apresenta as respostas e o SUS Score de cada participante, bem como a média. Apesar da média ter sido boa (acima de 68) 3 participantes obtiveram um resultado abaixo da média, entre eles dois desenvolvedores e um artista.

Os resultados do UEQ estão ilustrados na Figura 27 e na tabela 3. Os pontos do qual a ferramenta teve um desempenho ruim ou abaixo da média foram a confiabilidade e perspicuidade. Os outros pontos ficaram acima da média, com destaque para a estimulação que quase alcançou a categoria bom.

Participante	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
D0 (P0)	3	2	4	4	4	1	3	1	4	3	67.5
Q0 (P1)	5	2	4	1	5	2	5	1	5	1	92.5
A0 (P2)	3	4	3	2	5	2	4	4	2	2	57.5
A1 (P3)	3	1	4	1	5	3	5	1	4	1	85
Q1 (P4)	4	2	4	2	3	2	4	2	4	1	75
D1 (P5)	4	1	3	2	3	4	4	2	4	2	67.5
Média											80

Tabela 2. Resultados do questionário SUS

Escala	Média	Desvio padrão	Categoria	Interpretação
Atratividade	1,42	1,1	Acima da média	25% dos resultados são melhores, 50% dos resultados são piores
Perspiciuidade	0,83	0,54	Abaixo da média	50% dos resultados são melhores, 25% dos resultados são piores
Eficiência	1,29	0,58	Acima da média	25% dos resultados são melhores, 50% dos resultados são piores
Confiabilidade	0,75	1,18	Ruim	Está no intervalo dos 25% piores resultados
Estimulação	1,29	0,71	Acima da média	25% dos resultados são melhores, 50% dos resultados são piores
Novidade	0,71	0,19	Acima da média	25% dos resultados são melhores, 50% dos resultados são piores

Tabela 3. Resultados do benchmark UEQ (Tabela)

5.4.2. Análise Qualitativa

O processo de análise temática foi realizado de maneira cíclica e iterativa. Primeiro, seguindo o método de codificação, definimos previamente um conjunto de códigos que representam elementos da aplicação. Os códigos foram atribuídos a trechos relevantes das entrevistas. Com o decorrer da análise e emergindo a necessidade, se incluiu mais codificações. Esses códigos são:

- **Positivo:** Utilizado para classificar um trecho positivamente.
- **Negativo:** Utilizado para classificar um trecho negativamente.
- **Interface:** Comentários acerca da interface.
- **Imagem:** Comentários relacionados a qualidade dos assets gerados.
- **Upload:** Comentários destinados a função de upload.
- **Desenho:** Comentários destinados a função de desenho.

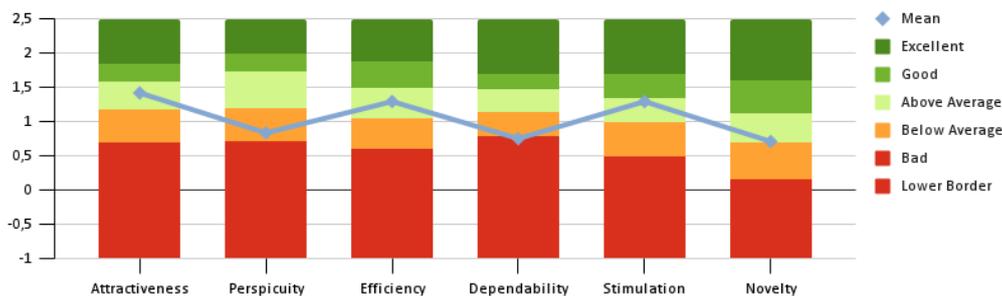


Figura 27. Resultados do benchmark UEQ (Gráfico)

- **Funcionalidade:** Comentários gerais relacionados a qualquer aspecto funcional da aplicação.
- **Utilização:** Comentários sobre como o participante utilizaria a ferramenta.
- **Ferramentas:** Comentários relacionados a qual ferramentas de trabalho o participante usa no cotidiano.
- **Criação:** Respostas relacionadas ao processo de criação de animações (criação de arte).

Os 4 últimos itens da lista correspondem aos códigos estabelecidos durante o processo de análise. A partir da codificação das falas dos participantes, foi facilitado o processo de agrupamento e identificação de padrões e temas recorrentes. A seguir serão abordadas as temáticas identificadas, com trechos selecionados das entrevistas. Cada trecho é identificado com o formato (XY:Z), onde X é uma letra correspondente a profissão do participante, Y é seu número identificador e Z é o número da pergunta pelo qual o trecho foi obtido. Aqui se utiliza a identificação do participante por profissão, como descrito na seção 5.4.2 (tabela 1).

- **Dificuldade de controlar o processo de geração**

Os entrevistados apontaram diversas dificuldades com relação ao controle do processo de geração dos spritesheets. 4 Participantes (D1, A0, A1, Q0) relataram ter problemas ao usar o prompt, destacando que não houve muita correspondência no que foi escrito textualmente com o resultado final (“Não necessariamente o que eu digitei no prompt saiu no resultado”(D1:10), “apesar de ser muito legal essa ideia dos prompts ..., pra conseguir fazer o funil ideal que você quer às vezes demora um pouquinho”(A0:9) e “A impressão que eu tive quando eu desenhei a personagem foi que o prompt do desenho não foi muito a valer consideração”(A1:9)), o entrevistado D0 pediu uma melhor explicação da ferramenta sobre os prompts (“Eu acho que talvez uma explicação melhor sobre como funcionam esses parâmetros de texto”(D0:5)). No caso particular do entrevistado A1 foi importante observar um comentário relacionado a dificuldade de gerar personagens com tom de pele mais escuro (“o input inicial gerado foi a personagem de Stardew Valley, que ela tem dark skin, mas tipo, querendo ou não, os prompts só geravam resultados de personagens que não tinham a mesma cor de pele”(A1:9) e “as que geraram de pele escura, as outras coisas que tinham no prompt acabou tendo que ser alterada, conseguiu a pele ser mais parecido, mas em compensação a cor do cabelo ficou diferente, a roupa não ficou o que devia estar escrito no prompt.”(A1:9)).

- **Qualidade das imagens geradas**

Além da dificuldade de possuir um controle sobre as imagens geradas, como mencionado anteriormente, dois entrevistados (D1 e Q1) perceberam no gif de animação que não necessariamente os frames das animações geradas estão em posições corretas, que ocasiona as falhas percebidas (“eu observei que ele gera algumas falhas no sprite”(Q1:9) e “As dificuldades eu acho que hoje vindo é muito por conta que ele não gera o tamanho da sprite certinho.”(D1:6)). O participante D0 destacou que a qualidade dos assets é bem variável utilizando a função de desenho (“acho que a qualidade deles é bem variável, tem uns que parecem bem legais, tem outros que são bem esquisitos”(D0:10)) e especificamente na função de upload, onde se utilizou um asset próprio e diferente das imagens parecidas com o modelo, apontou que a ferramenta não gerou assets que se conformam ao original (“acho que os assets de fato que ele gerou não são muito bons porque parece que eles não conformaram direito com o meu asset original.”(D0:9)). Para esses dois participantes, D0 e A1, foram informados durante a entrevista que a aplicação se conforma melhor com entradas que possuem um formato parecido com as imagens geradas pelo modelo, eles sugeriram tornar isso explícito de alguma forma na aplicação (“se a opção de desenhar tivesse algum grid, alguma coisa do tipo para me guiar, de que a mão tem que vir mais ou menos aqui, talvez melhore.”(A1:9) e “adicionar algum tipo de orientação sobre o tipo de imagem que ele espera ou que o desempenho dele é melhor”(D0:14)). Logo após esse apontamento, o participante A1 relatou que conseguiu gerar uma boa imagem (“Eu fiz outro desenho que gerou um sprite muito bom levando em consideração de onde vai a mão, de onde vai o corpo.”(A1:14)).

Um ponto criticado pelo entrevistado A0 foi as opções avançadas de geração, sendo complicado de entender e de certa forma inútil porque facilmente alterando alguns valores a imagem gerada vira ruído (“pra mim até foi (complicado), porque eu tive que ler e parar pra raciocinar o que era.”(A0:1) e “se eu quero que no final ele gere uma imagem de qualidade, alguns valores aqui se eu tirar pra aumentar a imagem gerada vira só ruído, ... acaba gerando uma imagem inútil pra mim.”(A0:1)). Dois participantes (D0 e D1) enxergaram as opções avançadas positivamente (“eu acho que são interessantes”(D0:3) e “a parte de configurações adicionais, eu acho que seria interessante aumentar as configurações,”(D1:4)) porém, o participante D1 ressaltou que isso poderia ser simplificado, como uma espécie de filtro aplicado a imagem (“adicionar algum filtro que eu possa mudar, ... tipo o filtro do Instagram, ... se eu quisesse gerar uma imagem em preto e branco (por exemplo)”(D1:4)).

Mas superando as dificuldades do processo de geração, uma parte dos entrevistados (Q0, Q1, D1) reconheceu positivamente a qualidade dos assets gerados (“é muito massa a ferramenta. ... isso do estilo de arte eu gostei bastante, porque ele gera uma arte bem bonitinha.”(D1:14), “Ele gera um asset bem legal a partir do prompt e do formato”(Q1:10) e “gostei bastante do resultado.”(Q0:10)).

• **Display da função de desenho**

Três participantes (Q0, Q1, D1) mencionaram que o tamanho da tela utilizada para desenhar o personagem é pequena e que poderia ser aumentada (“aumentar esse campo do desenho, porque ele tá bem pequeno”(Q1:5), “aumentar o tamanho da área do desenho, que hoje eu acho que é um pouco pequeno.”(D1:1) e “opção pra aumentar quando tem um desenho, pra fazer um desenho melhor e mais detalhista.”(Q0:4)). 3 Participantes (D1, A0 e A1) mencionaram que o editor de imagem poderia ter outras

funcionalidades mais cômodas e que facilitariam o desenho (“talvez essa geração, esse desenho de imagem, ele ser também em pixel art, talvez facilite a geração, ou então talvez fique mais cômodo.”(D1:1) e “algumas ferramentas bem pick and paint em geração de formas simples para facilitar e agilizar o processo da pintura.”(A0:5)).

- **Outros aspectos negativos da interface**

Alguns outros aspectos negativos relacionados a interface foram levantados. Dois entrevistados (A0, Q1) relataram que precisavam ficar rolando a página durante o uso (“a ferramenta no momento tá ocupando muito espaço, então tem que dar Scroll pra escrever alguma coisa e apertar o botão Scroll pra cima de novo”(A0:14)), esses mesmos participantes sugeriram mudanças na disposição dos elementos de interface para eliminar esse problema e destacar elementos importantes da interface (“juntar o filtro de animação com o prompt, tipo ficar mais destacado, perto desse campo principal, que é onde a pessoa gera o sprite, tipo do lado, acho que ajudaria, que a pessoa ficava comparando o que ela inseriu e o que tá gerando.”(Q1:4) e “Toda essa parte de upload imagem poderia ser de um lado fixo, ... Enquanto toda essa parte de cima tipo input, raw, spritesheet, animation, tudo isso poderia ficar meio que empilhado na direita, e na esquerda ficar todas essas configurações que você quer fazer”(A0:14)), o participante A0 também reforça que havia muito espaço vazio (em branco) (“espaço em branco tem muito”(A0:14)) e que o espaço da galeria poderia se tornar outra aba (“o lugar da galeria ... talvez testar ela ficando em cima, com mais uma aba, dando as três opções pra pessoa”(A0:14)). Outro participante (D0) relatou que havia barras de progresso redundantes na aplicação (“quando você clica pra gerar o spritesheet, ele tem 3 barras de progresso, na verdade 4 barras de progresso que são redundantes, eu não acho que precisa disso.”(D0:2)).

- **Uso da ferramenta**

Todos os entrevistados responderam que utilizariam a aplicação. Três participantes apontaram que usariam a ferramenta para a construção de uma prova de conceito ou protótipo (“Eu acho que isso é bom para prototipagem, ... acho que para isso seria uma boa aplicação, para conceito e protótipo”(D0:6), “eu usaria as sprites geradas pelo site para fazer um teste de implementação e validação de prova de conceito.”(D1:6) e “Talvez para protótipos, para validar a prototipação e para gerar NPCs de maneira mais rápida”(A1:6)). Um dos participantes (A0) mencionou que usaria a ferramenta se pudesse utilizar sua própria galeria como fonte para o modelo de IA (“Se eu usasse, usaria na condição de usar minha própria galeria de fonte para a inteligência.”(A0:6)) e que no estado atual da ferramenta poderia utilizar como base ou ideia para a criação de um personagem (“mesmo que eu não use ele 100% como tá, ... já é uma ajuda mesmo, ... aqui ele (ferramenta) já tá me dando praticamente pronto uma ideia.”(A0:12)). Dois participantes afirmaram que utilizariam os assets gerados do jeito que a ferramenta entrega (“Usaria diretamente o que ela gera, porque eu não saberia modificar pra ficar melhor do que já tá.”(Q1:6) e “Sim, ... apenas fazendo o drag and drop das imagens.”(Q0:6)).

- **Outros aspectos positivos da aplicação**

Três participantes (D0, A0, A1) relataram que a ferramenta é simples e intuitiva (“acho que foi tudo bem intuitivo”(D0:14) e “Eu gostei dessa questão de ser simples”(A0:14)). 3 entrevistados (D0, D1, A0) destacaram positivamente a galeria de inputs

de referência, que poderiam ser ampliada na aplicação “Eu gostei muito dessa galeria aqui de baixo”(A0:4) e “A parte dos presets, ... te ajuda bastante a já fazer novas coisas sem precisar tanto trabalho”(D0:3). Todos entrevistados destacaram pelo menos um aspecto positivo da aplicação (geral ou específico), afirmando ser legal ou interessante.

- **Integração com outras ferramentas**

Todos os entrevistados de alguma forma veem positivamente a integração das funcionalidades da ferramenta proposta com alguma outra ferramenta que usam no cotidiano. Dentre as ferramentas que os participantes usam em sua rotina de trabalho estão game engines (Unity e Unreal), IDE's e editores de texto (Visual Studio, Visual Studio Code, Vim), hospedagem de código fonte (Github), editor de imagem ou animação (Adobe Photoshop, Adobe Illustrator, Aseprite, Dragon Bones) e modelagem 3D (Blender). Apenas um entrevistado (Q1) afirmou que não usa nenhuma ferramenta específica (“Eu não uso nenhuma, ... Apenas testo as funcionalidades já no jogo implementado”(Q1:11)).

Sobre os comentários acerca da integração, dois participantes (D0 e D1) chegaram a sugerir formas de como isso poderia ser realizado (“Eu acho que é interessante, ... daria para fazer algum tipo de editor ou plugin de engine em cima dessa ferramenta web.”(D0:12) e “Eu acharia muito interessante se tivesse como a ferramenta gerar um prefab, por exemplo no Unity, que o prefab já gerasse meio que com um controle de estado, uma máquina de estado.”(D1:12). Outro participante (A1) relatou que isso seria interessante para a geração rápida de assets para game engines (“vai ser interessante ter plugins de certa forma que eu possa gerar esses assets de maneira mais rápida”(A1:12)), e outros dois participantes (Q1 e A0) mencionaram que ajudaria no processo de criação e que nem tudo precisaria ser feito do zero (“Ajudaria bastante a iniciar o processo, ... já cria o personagem e depois a pessoa melhora com base nisso.”(Q1:12) e “Ia ser uma mão na roda, ... por assim dizer eu não tenho que fazer tudo do zero.”(A0:12)).

- **Adição de outros tipos de animação e outros estilos**

Três entrevistados (Q0, A0, D1) apontaram que a ferramenta poderia incluir mais tipos de animação (“os tipos de animações, deveria ter acho que pulando, atacando, essas coisas.”(Q0:5), “no futuro fazer alguns tipos de animações diferentes, de ações pré-definidas, de pular, atacar e outros tipos de animações.”(A0:5) e “poderia existir também a geração de Idle (Tipo de animação), que foi mais o que eu fiquei sentindo falta.”(D1:13)). 2 participantes (Q0 e Q1) gostariam de ver a possibilidade de gerar animais ou objetos “Uma adição seria isso, gerar várias coisas, não só humanos.”(Q0:14) e “Seria acrescentar coisas não-humanas e objetos também.”(Q1:13). 2 entrevistados (D0 e A1) veem de maneira positiva a ferramenta possuir mais estilos além de pixel art (“aumentaria o valor dessa plataforma, seria interessante. Até porque se eu quiser testar um protótipo ... que não seja pixel art eu não tenho muito como usar isso aqui.”(D0:13) e “Seria massa ..., nesse momento essa ferramenta meio que transforma o meu desenho sempre no estilo de desenho já fixo, que é esse de pixel assim.”(Q1:13))

5.5. Discussão

Nesta seção será apresentada uma discussão dos resultados obtidos. Ela foi dividida nos 3 pontos a seguir.

5.5.1. Limitações da ferramenta

Ao se observar tantos os resultados quantitativos e qualitativos é nítida a principal limitação da ferramenta: o controle sobre as imagens geradas. Os resultados do benchmark do questionário UEQ mostram que a confiabilidade teve uma média baixa e que entrou na categoria ruim. Os depoimentos convergem com esses resultados, relatando a dificuldade de controlar o processo de geração. Essa limitação é sobretudo uma limitação do modelo. Como visto na experimentação, gerar as animações através apenas do texto traz resultados inconstantes, e o que a solução proposta faz é jogar a responsabilidade do controle na imagem de entrada, com a entrada textual sendo secundária e com pouca influência no processo. Isso implica em outra limitação discutida na implementação, que é a necessidade da imagem de entrada se adequar ao espaço de geração do modelo, com relação a cor e forma de cada parte do personagem. Isso ocorre devido ao sobre-ajuste (overfitting) do modelo, essa característica acaba sendo uma faca de dois gumes: Por um lado, consegue gerar os spritesheets com uma boa qualidade dentro do seu espaço de geração, porém, limita bastante as opções de controle sobre o que é gerado. A solução proposta até consegue tirar um melhor proveito do modelo, permitindo pequenas alterações via prompt, mas ainda limita bastante a possibilidade de controle sobre o que é gerado. Ainda nesse tema, outro ponto é o viés sobre o qual os dados foram treinados, com o exemplo da dificuldade de gerar personagens com o tom de pele escuro como um dos participantes relatou. Outro problema identificado tanto por nós na experimentação quanto por alguns participantes foram as “falhas” que ocorriam com certa frequência nas animações geradas, onde os pixels de um frame invadem o espaço de outro frame.

Sobre essas limitações, o que pode ser alterado agora na ferramenta, sem mudar o modelo ou processo de geração, é apresentar essa limitação de forma adequada para o usuário. Uma possibilidade é a ferramenta já incluir um conjunto de desenhos simples e validados, como nos exemplos dos presets, e a partir daí oferecer uma simples edição das cores de cada parte do personagem, através de preenchimento. Outra possibilidade é transformar a função de upload em uma espécie de função text-to-image. Isso poderia ser feito através de opções de múltipla escolha na interface, onde o usuário vai escolhendo a cor de cada parte do personagem e por baixo dos panos a geração ocorrer normalmente com a função image-to-image, sendo a ferramenta responsável por juntar em uma imagem de entrada as cores de cada parte. Junto a essas mudanças as opções avançadas (hiperparâmetros) e o prompt poderiam ser retirados da interface, reduzindo a complexidade da ferramenta e fazendo com que ela seja mais fácil de se familiarizar, algo que foi apontado como abaixo da média no benchmark UEQ (Perspicuidade). Para a questão da delimitação dos frames, poderia ser feita uma opção de edição pós-geração onde o usuário pudesse editar onde se localiza cada frame na imagem, que seria editar as zonas em azul na Figura 25, por exemplo.

Essas possibilidades teriam que ser re-validadas, mas ainda assim seriam soluções que não atacam a verdadeira raiz das limitações, que é o modelo utilizado para a geração. Ainda é necessário mais pesquisa e experimentação com outras técnicas de geração de imagens e modelos, uma possibilidade será apresentada na conclusão.

Para os apontamentos negativos aos elementos da interface, com o destaque ao tamanho do display e disposição dos elementos, devem ser corrigidos ou repensados vi-

sando a melhoria da usabilidade e da experiência do usuário em uma nova iteração da ferramenta.

5.5.2. Pontos positivos

Todos os participantes comentaram que usariam a ferramenta de alguma forma durante o processo de desenvolvimento de algum jogo. Um dos participantes destacou a rapidez e não houve comentários negativos com relação a velocidade de geração, o que reflete o benchmark UEQ que classificou a eficiência como acima da média. Formas distintas de uso da aplicação foram relatadas, os desenvolvedores utilizariam para a prototipação e prova de conceito, para os artistas há interesse na prototipação (geração de NPC's) ou no uso como fonte inspiração ou base para construção de personagens, bem como a possibilidade de usar a própria arte como fonte para a inteligência artificial. Os QA's foram os únicos participantes que afirmaram que usariam os assets da forma como a ferramenta entrega, isso pode ter sido em razão da falta de experiência com criação de arte. Esses resultados mostram que para desenvolvedores e artistas essa ferramenta possui um valor, porém ainda como um auxiliar no processo de desenvolvimento de um jogo, não propriamente substituindo o processo de animação. Para os participantes que não possuem muita experiência com arte (QA tester) esse valor é ainda maior, com a ferramenta sendo mais protagonista no desenvolvimento das animações do jogo. Esse padrão pode se relacionar com os resultados do questionário SUS (tabela 2), as avaliações dos QA testers estão entre as mais altas. Porém, há limitações com essas considerações levando em conta o estilo de imagem que a ferramenta gera, que como um dos participantes mencionou só daria para se utilizar em um jogo com o estilo de pixel art. Entre os outros parâmetros que obtiveram um desempenho positivo no UEQ foram a atratividade e a estimulação, que condiz com boa parte dos participantes reforçando que a ferramenta era simples, intuitiva, que era legal e interessante de se usar. Esses aspectos positivos devem ser levados em consideração em iterações futuras da aplicação.

5.5.3. Inclusão novas funcionalidades para uma próxima iteração da ferramenta

Além da redução da complexidade da interface e da necessidade de desenvolver modelos voltados para a geração de animação, essa entrevista também conseguiu levantar mais funcionalidades a serem incluídas na construção de uma ferramenta mais robusta. Todos os participantes demonstraram interesse na integração com alguma outra ferramenta. Para desenvolvedores seria muito útil que houvesse a possibilidade de, em uma game engine como Unity por exemplo, gerar um prefab com todas as animações do personagem prontas e com as transições já definidas. Para os artistas seria útil usar as imagens geradas como base ou inspiração para uma outra arte. A inclusão de outros tipos de animação, a geração de outros elementos como animais e objetos bem como a adição de outros estilos de imagem também são bem vindas para essa aplicação. A partir desses resultados é possível elaborar uma lista de funcionalidades adicionais que a ferramenta pode ter e que agilizaria ainda mais o processo de geração de animações.

5.6. Limitações da pesquisa

Essa pesquisa foi feita com uma quantidade reduzida de participantes (6 pessoas), todos profissionais que atuam na área de desenvolvimento de jogos em um pequeno estúdio de Recife. O público participante pode ser considerado jovem (média de 24.5 anos) e com pouco tempo de experiência (média de 3.17 anos aproximadamente). Os resultados dessa pesquisa pode ter vieses e para ter um recorte demográfico mais abrangente seria necessário entrevistar mais pessoas, que incluísse profissionais de diversos estúdios e também desenvolvedores indie, abordando também um público mais velho e com mais experiência.

6. Conclusões

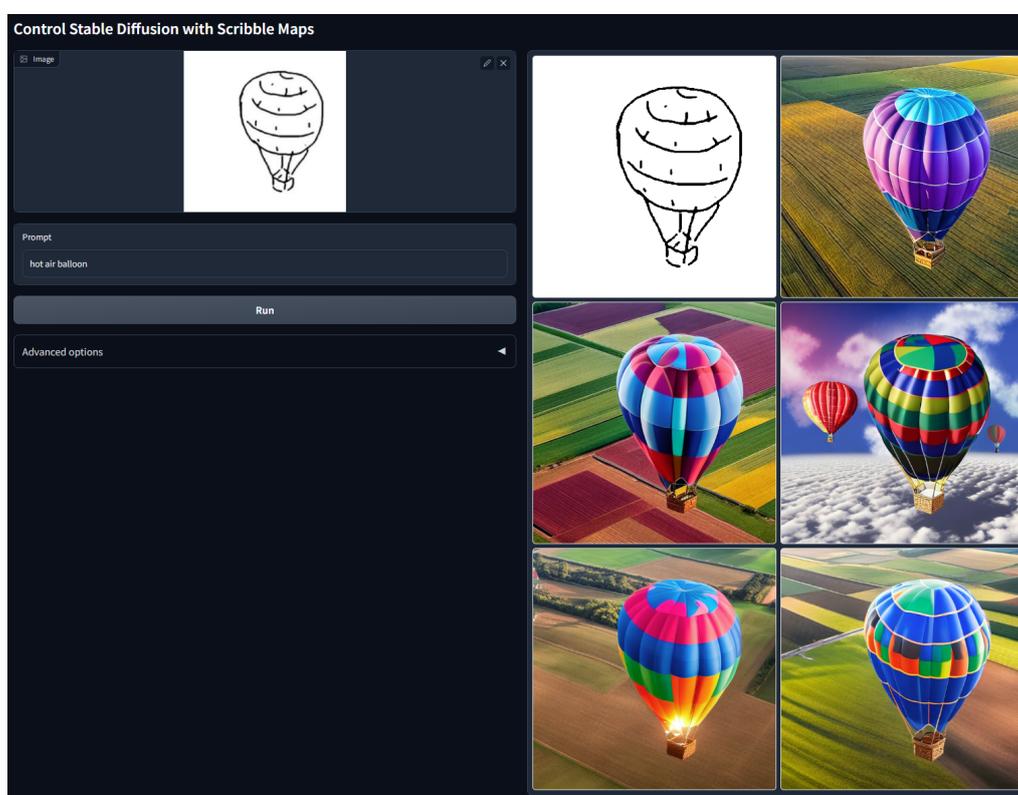


Figura 28. Exemplo de geração da ControlNet utilizando rabiscos

Este trabalho conseguiu, dentro das limitações discutidas, validar parcialmente um protótipo de uma aplicação que usa IA para a geração de animações frame a frame. A ferramenta atende aos critérios de automação e rapidez de geração, entretanto a fidelidade com a descrição visual do personagem ainda não foi atingida, se limitando ao reduzido espaço de geração do modelo utilizado. Para a experiência do usuário, utilizando como critérios as escalas do UEQ, a ferramenta pode ser considerada atrativa, eficiente e estimulante, mas não é confiável e perspicaz. As avaliações SUS abaixo da média e depoimentos coletados durante as entrevistas indicam problemas na usabilidade e a necessidade de ajustes na interface da aplicação. Durante o processo de avaliação, conseguimos elucidar o valor de uso dessa ferramenta para diferentes perfis de profissionais atuantes na indústria de jogos, bem como as alterações e adições necessárias para uma iteração mais robusta da aplicação.

Como trabalhos futuros, vemos a necessidade de se experimentar mais técnicas de geração de imagem para animações. O ControlNet [Zhang and Agrawala 2023] é uma possível alternativa para as limitações atuais da ferramenta. Trata-se de uma estrutura de rede neural usada para controlar modelos de difusão, permitindo com que eles suportem condições adicionais de entrada, como bordas, mapas de segmentação, poses humanas, rabiscos, entre outros. Esse parâmetro adicional permite que a rede gere imagens que correspondem à forma fornecida como entrada. A Figura 28 representa um exemplo de uso dessa rede treinada com o Stable Diffusion 1.5, onde o rabisco do balão em um canvas é utilizado como entrada e a mesma forma é mantida nas imagens geradas. Seria necessário experimentar o ControlNet em conjunto com modelos de Stable Diffusion treinados para gerar assets, e caso essa abordagem traga bons resultados para animações seria necessário implementar um novo fluxo para a aplicação, bem como uma nova rodada de testes para verificar se as limitações discutidas ainda persistem.

Referências

- Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., and Zou, J. (2019). Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*.
- Adobe (2022). Rigging and skeletal animation: what it is and how it works. <https://www.adobe.com/uk/creativecloud/animation/discover/rigging.html>. Acesso em 15 de abril de 2023.
- AUTOMATIC1111 (2022). Stable diffusion webui. <https://github.com/AUTOMATIC1111/stable-diffusion-webui>. Acesso em 10 de Abril de 2023.
- Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3:77–101.
- Brooke, J. (1995). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Coutinho, F. and Chaimowicz, L. (2022). Generating pixel art character sprites using gans.
- Ebert, D. (2023). 2d asset generation: Ai for game development #4. <https://huggingface.co/blog/ml-for-games-4>.
- Hong, S. Y., Kim, S., and Kang, S. (2019). Game sprite generator using a multi discriminator gan. *KSII Trans. Internet Inf. Syst.*, 13:4255–4269.
- Laugwitz, B., Held, T., and Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. volume 5298, pages 63–76.
- Onodofthenorth (2022). Onodofthenorth/sd_pixelart_spritesheet_generator. https://huggingface.co/Onodofthenorth/SD_PixelArt_SpriteSheet_Generator. Acesso em 16-12-2022.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.
- Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. (2022). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation.

Saldaña, J. (2009). The coding manual for qualitative researchers.

Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Construction of a benchmark for the user experience questionnaire (ueq). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4:40–44.

Shuter, G. (2020). Frame-by-frame animation: A complete guide. <https://www.twine.net/blog/frame-by-frame-animation-complete-guide>. Acesso em 15 de abril de 2023.

Zhang, L. and Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models.