

2D Sprite Sheet Generator: uma ferramenta para geração dinâmica de animações de personagens utilizando Stable Diffusion

Daniel Acco Maroni¹, Rafael Rieder¹

¹Ciência da Computação – Universidade de Passo Fundo (UPF)
BR 285 Km 292,7 – São José – Passo Fundo, RS, Brasil

{169481, rieder}@upf.br

Abstract. *Traditional drawing methodologies used to create 2D character animation consume time because designers must manually create each character frame in modeling and animation software. In this context, this work aims to present **2D Sprite Sheet Generator**, a tool for automatically generating dynamic character animations. To this end, we used Stable Diffusion, a deep-learning model that uses text input to generate highly detailed and adaptable images. Based on the user input and specified details, the tool generates a sprite sheet of a 2D character and a sample GIF animation, all within less than three minutes. From a pilot study involving seven professionals in the gaming industry, the tool demonstrated satisfactory results, showing excellent acceptance and successful execution of its functionality. The main advantage of our approach is the innovation and potential of the solution, considering the market's lack of tools that utilize artificial intelligence to facilitate the generation of 2D characters. The proposed solution addresses a critical need in the industry, streamlining the animation creation process and offering a practical solution for developers and artists.*

Resumo. *O processo de criação de animações de personagens 2D utilizando metodologias convencionais de desenho costuma consumir um tempo considerável, levando em consideração que cada frame do personagem precisa ser criado manualmente em softwares de modelagem e animação. Neste contexto, o objetivo deste trabalho é apresentar o **2D Sprite Sheet Generator**, uma ferramenta para geração dinâmica de animações de personagens, de forma automática. Para tanto, utilizou-se Stable Diffusion, um modelo de aprendizado profundo que utiliza uma entrada de texto e gera imagens com grande detalhismo e adaptabilidade. Conforme a entrada do usuário e possíveis detalhes apontados, a ferramenta gera uma spritesheet de um personagem 2D assim como um GIF de amostra da animação, levando menos de três minutos. A partir de um estudo piloto com sete profissionais da área de games, evidenciou-se resultados satisfatórios no uso da ferramenta, mostrando ótima aceitação e um bom êxito na execução de sua funcionalidade. Como principal vantagem da abordagem, destaca-se a inovação e o potencial da solução, tendo em vista que o mercado carece de ferramentas que utilizem inteligência artificial para facilitar a geração de personagens 2D. A abordagem proposta atende a uma necessidade crítica na indústria, agilizando o processo de criação de animações e oferecendo uma solução prática para desenvolvedores e artistas.*

1. Introdução

Animação é a exibição de imagens rapidamente em sequência para criar uma ilusão de movimento [Harrison & Hummell 2010]. Em animação de filme tradicional, a taxa de atualização de quadros (*frames per second*, FPS) é geralmente de 24 quadros de animação por segundo. As televisões geralmente têm uma taxa de atualização de 50 FPS ou 60 FPS e, portanto, são compatíveis com 25 ou 30 quadros de animação por segundo, respectivamente. [Feijó & Badaró 2006].

Na animação cinematográfica tradicional, muitas vezes apenas 12 desenhos únicos são usados por segundo, com duas exposições para cada desenho [Silveira 2017]. Usar menos quadros reduz significativamente a quantidade de trabalho e é suficiente para a maioria das ações, porém 24 quadros individuais são necessários apenas para movimentos muito rápidos ou suaves [Assunção 2012]. A taxa de quadros usada em videogames varia dependendo do tipo de jogo e das capacidades do hardware. Nos jogos modernos, a taxa de quadros geralmente fica entre 30 e 60 quadros por segundo. Altas taxas de quadros são particularmente importantes em jogos de ação em ritmo acelerado.

De acordo com [Roberts 2012], existem duas maneiras principais de animar: direta e pose a pose. Animação direta significa desenhar a ação quadro a quadro, do início ao fim. Tal método resulta na animação criativa e espontânea, mas dificulta a manutenção de proporções e focar dentro de uma cena. Na animação pose a pose, o animador primeiro decide quais são os desenhos mais importantes do movimento e começa por desenhá-los. Os quadros intermediários são desenhados posteriormente. Este método dá controle completo sobre a cena, e o resultado é claro e poderoso, mas carece da espontaneidade de uma ação direta. Ambos os métodos devem ser utilizados para alcançar os melhores resultados possíveis [Rantala 2013].

A constante evolução dos recursos computacionais tem acelerado a animação e o processo de produção de animações, ampliando o campo de atuação para diversos tipos de mídia, não somente destinados a demandas de filmes de cinema ou televisão. Essa amplitude considera a criação de animações para jogos de computador e mídias de streamings e internet [Saputra et al. 2021]. Nesse contexto, artistas e agências criativas que antes restringiam seus talentos para um ou outro tornaram-se mais generalizados, devido à convergência digital. Técnicas de animação eficazes surgiram do artesanato tradicional de desenhos feitos à mão que, por sua vez, provêm de uma compreensão das Leis do Movimento de Newton [Kuperberg 2002].

Produzir uma animação 2D consome tempo, de alguns meses até vários anos, pois depende de diferentes fatores. De acordo com [Shahbazi 2024], uma animação de 30 segundos pode levar de dois a quatro meses para ser concluída, enquanto uma de 60 segundos pode exigir de até seis meses de produção. Já uma animação de 90 segundos pode levar nove meses ou mais para ser finalizada, especialmente se envolver visuais elaborados ou uma narrativa mais complexa. Assim, o tempo necessário para criar uma animação depende muito da duração e da complexidade da história a ser contada. Filmes da Disney antigos, como “A Pequena Sereia (1989)” e o “O Rei Leão (1994)” possuem, em média, 140 minutos de duração, e precisaram de cerca de três anos para serem finalizados.

Nesse sentido, o objetivo deste estudo é apresentar o desenvolvimento do **2D Sprite Sheet Generator**, uma ferramenta que utiliza inteligência artificial (IA) para

geração dinâmica de animações de personagens 2D, de forma automática. Para tanto, a solução proposta emprega, como base, recursos do modelo de aprendizado profundo Stable Diffusion [Rombach et al. 2021, Rombach et al. 2024] para receber apenas uma descrição em texto como parâmetro, e gerar como saída N frames de uma única animação. O intuito desta abordagem é diminuir o tempo de criação de animações de personagens 2D e possibilitar o rápido desenvolvimento de animações por meio de recursos de IA.

Para tanto, o trabalho está organizado como segue: a Seção 2 mostra alguns trabalhos relacionados recentes do uso de IA no processo de animações; a Seção 3 apresenta os materiais e métodos utilizados, bem como descreve o desenvolvimento da solução proposta; a Seção 4 divulga os resultados obtidos com um estudo preliminar com profissionais da área e discute as vantagens e limitações do estudo; e a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

O estudo de [Xi & Chung 2023] analisa o impacto da tecnologia de IA na criação de personagens de animação, com foco no software de geração de imagens Midjourney¹. Comparado aos métodos tradicionais de pintura digital, o uso da IA está revolucionando a lógica de criação de personagens animados. Os resultados destacam como a IA, especialmente por meio do Midjourney, automatiza tarefas de design, oferece abordagens inovadoras e produz personagens visualmente atraentes e realistas.

Em resumo, [Xi & Chung 2023] mostram que a integração da IA na criação de personagens animados promove fluxos de trabalho mais eficientes, estimula a criatividade e eleva a qualidade geral das animações. Eles destacam que são ferramentas com potencial de impulsionar ainda mais a inovação na indústria de animação.

Já o estudo de [Serpa & Rodrigues 2019] aborda o desafio técnico e artístico do desenvolvimento de jogos, especialmente no contexto do aumento da demanda por trabalho artístico de alta qualidade. Explora-se a geração de ativos por meio de aprendizado profundo para reduzir os custos e esforços de desenvolvimento de jogos. Especificamente, o foco está na geração de sprites de arte pixelada a partir de esboços de arte linear, utilizando técnicas de tradução de imagem de ponta. O trabalho é situado no contexto do jogo “Trajes Fatais: Suits of Fate”, um jogo de luta 2D inspirado nos clássicos dos anos noventa.

Os resultados obtidos por [Serpa & Rodrigues 2019] indicam que a técnica de geração de ativos por aprendizado profundo é capaz de produzir sprites semelhantes aos criados pela equipe de artistas, demonstrando sua eficácia por meio de análises qualitativas e quantitativas, bem como avaliação por designers de personagens.

De forma similar a este trabalho, o trabalho de [Januário & Calegario 2023] também utiliza a Stable Diffusion para geração de personagens dinâmicos, além de outras técnicas similares. Porém, a diferença em relação a este trabalho está ligada ao formato de entrada de dados utilizado no momento da geração da saída. Nesse caso, a abordagem aguarda o envio de imagens de desenhos feitos a mão, para a criação automática de personagens no estilo PixelArt 2D.

¹<https://www.midjourney.com/home>

A ferramenta foi avaliada por um grupo de seis profissionais atuantes em um estúdio de desenvolvimento de jogos em Recife. De acordo com [Januário & Calegario 2023], os resultados mostram que a solução atende aos critérios de automação e rapidez de geração, entretanto a fidelidade com a descrição visual do personagem ainda precisa ser melhorada. Em relação a experiência do usuário, o estudo mostrou que os profissionais consideraram a aplicação atrativa, eficiente e estimulante, mas carece de ajustes para ser útil em atividades cotidianas de um estúdio gráfico.

O estudo de [Fukaya et al. 2023] busca preencher lacunas na compreensão das possibilidades da geração procedural de conteúdo para jogos. O trabalho observa que, por vezes, a pesquisa existente se concentra em métodos específicos para a geração de ativos gráficos, como personagens, nuvens, prédios ou vegetação, sem considerar amplamente as aplicações possíveis, tanto dentro quanto fora dos jogos. Para resolver esse problema, os autores realizam uma revisão sistemática da literatura, examinando 200 artigos aceitos, a fim de explorar as abordagens mais recentes para a geração de ativos gráficos e preencher lacunas de conhecimento.

Como resultado do trabalho de [Fukaya et al. 2023], um quadro conceitual é derivado da literatura para abordar essas lacunas e fornecer um guia para interessados descobrirem métodos e abordagens para suas necessidades, mapeando o processo de uso deles. Essa pesquisa oferece uma visão abrangente das possibilidades de geração de ativos gráficos, tanto para jogos quanto para outros contextos, facilitando a compreensão e a aplicação dessas técnicas na prática.

Levando em consideração os trabalhos relacionados encontrados, pode-se observar que o uso de ferramentas de IA, como Stable Diffusion, que suportam entrada de textos para geração de assets podem ser úteis para a indústria criativa. No caso desse estudo, uma ferramenta para geração de animações de personagens 2D, de forma totalmente dinâmica e de fácil utilização, se trata de uma nova forma de utilização da IA no meio criativo, que diferente de trabalhos anteriores como de [Xi & Chung 2023] e [Januário & Calegario 2023], é uma solução que pode auxiliar na criação dos elementos gráficos, buscando agilizar o processo, gerando por fim assets que podem ser otimizados ou empregados diretamente em diversas aplicações relacionadas.

3. Materiais e Métodos

Para o desenvolvimento do **2D Sprite Sheet Generator**, foram utilizadas diferentes ferramentas e técnicas. As próximas seções detalham a respeito.

3.1. Ferramentas

Como principal motor da aplicação e gerador das imagens dinâmicas, optou-se por utilizar a interface web “Stable Diffusion web UI”² (Figura 1), recurso que simplifica a utilização do modelo Stable Diffusion. Esta interface permite adicionar extensões necessárias e configurar diversos parâmetros de forma manual, oferecendo flexibilidade e controle sobre o processo de geração de imagens. Além disso, a “Stable Diffusion web UI” possui uma API integrada, que foi utilizada nas requisições da aplicação para facilitar a automação e a integração dos processos de geração de imagens dinâmicas diretamente na plataforma web da aplicação proposta.

²<https://github.com/AUTOMATIC1111/stable-diffusion-webui>

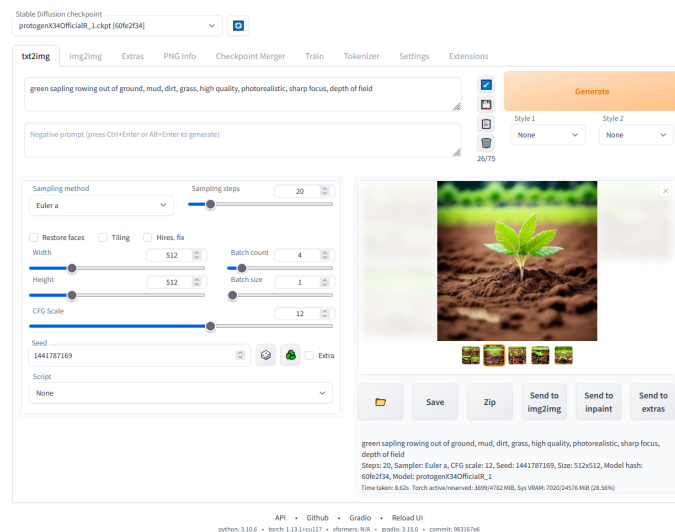


Figura 1. Amostra visual da interface Stable Diffusion web UI.

Quanto as extensões utilizadas na “Stable Diffusion web UI”, optou-se pela utilização das seguintes:

- sd-webui-controlnet: para utilizar na aplicação a rede neural *controlnet*;
- sd-webui-api-payload-display: para mostrar o *payload* de requisições feitas pela interface.

Para a geração dos personagens em diferentes estilos, foram utilizados alguns modelos já treinados como checkpoints da ferramenta de Stable Diffusion. São eles:

- Dreamlike Photoreal 2.0 ³, para imagens realistas;
- Pixelart Difussion x1 ⁴, para imagens no estilo *pixelart*;
- SPYBG’s Toolkit ⁵, para imagens no estilo *cyberpunk*.

Esses checkpoints fornecem uma base robusta de padrões para a geração de personagens, garantindo consistência e qualidade nas saídas. Ao utilizar esses modelos pré-treinados, foi possível acelerar o processo de criação e garantir que os personagens gerados atendam aos requisitos estéticos e funcionais desejados para a aplicação.

Para que seja possível gerar *spritesheets*, tanto de humanoides, quanto de animais quadrúpedes, foram utilizados dois modelos distintos de ControlNet, que atuam na geração da saída, modificando-a para o tipo desejado. Estes são os modelos:

- control_openpose-fp16 ⁶, para humanoides;
- control_sd15_animal_openpose_fp16 ⁷, para animais quadrúpedes.

Estes modelos são especificamente ajustados para lidar com as diferentes morfologias e movimentos dos humanoides e quadrúpedes, garantindo que as animações sejam

³<https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0>

⁴<https://civitai.com/models/277680/pixel-art-diffusion-xl>

⁵<https://civitai.com/models/4118>

⁶https://huggingface.co/webui/ControlNet-modules-safetensors/blob/main/control_openpose-fp16.safetensors

⁷https://huggingface.co/huchenlei/animal_openpose/blob/main/control_sd15_animal_openpose_fp16.pth

precisas e realistas. Cada modelo do ControlNet é treinado com dados específicos para identificar e replicar as características e comportamentos únicos de cada tipo de criatura, permitindo a criação de *spritesheets* de alta qualidade adaptados às necessidades de diversos projetos de animação.

Para conceber a interface da página web que recebe a entrada do usuário e mostra a saída obtida, optou-se pela linguagem de programação Python ⁸, que oferece recursos para a criação de páginas web, e tem forte suporte à ferramentas de IA. Para o roteamento da página web, foi utilizado o pacote Flask ⁹, um microframework que permite o rápido desenvolvimento de aplicações web em Python, e integra-se também com bibliotecas de IA. A combinação dessas ferramentas facilita a construção de interfaces web intuitivas e responsivas, processando entradas dos usuários e exibindo resultados de maneira eficaz.

Como a imagem de saída gerada possui sempre um fundo colorido, e a aplicação precisa de imagens transparentes, a ferramenta Backgroundremover ¹⁰ foi adicionada ao projeto. Ela utiliza IA para detectar e remover fundos de imagens e vídeos, permitindo que apenas o objeto principal permaneça visível. Ela analisa a imagem pixel por pixel, identificando os contornos do objeto principal e separando-o do fundo. Isso é especialmente útil para criar animações e gráficos com fundos transparentes, facilitando sua integração em diversos cenários e aplicações sem a interferência de elementos indesejados.

Para gerar o resultado final em formato de arquivo GIF, incluiu-se na ferramenta proposta o uso dos recursos do ImageMagick ¹¹. O ImageMagick é uma suíte de software capaz de criar, editar e converter imagens em uma ampla variedade de formatos. Com ele, é possível combinar várias imagens, ajustar cores, aplicar efeitos e otimizar a qualidade do GIF a ser gerado. A flexibilidade e a robustez do ImageMagick também permitem ajustes adicionais de parâmetros, como tempo de transição entre quadros e compressão, garantindo que o produto final seja eficiente e visualmente atraente.

Para teste e validação da ferramenta, pessoas da área foram convidadas a testar utilizando um link onde a ferramenta estava sendo disponibilizada. Para isso, utilizou-se o túnel do Tailscale ¹², que permite a criação de uma rede privada segura, facilitando o acesso remoto e garantindo a privacidade dos dados durante o processo de teste.

Por fim, para testar a funcionalidade dos *spritesheets* gerados pela ferramenta, utilizou-se a plataforma Unity ¹³, onde foi possível criar personagens 2D e suas devidas animações utilizando como base os *spritesheets* gerados. A Unity, conhecida por sua versatilidade na criação de jogos e animações, permitiu validar a qualidade e a aplicabilidade dos *spritesheets*, garantindo que eles funcionassem corretamente dentro de um ambiente de desenvolvimento real.

3.2. Hardware e Sistema Operacional

Para implementação da ferramenta e geração das saídas de teste, a configuração utilizada foi a seguinte:

⁸<https://www.python.org/>

⁹<https://flask.palletsprojects.com/en/3.0.x/>

¹⁰<https://github.com/nadermx/backgroundremover>

¹¹<https://imagemagick.org/index.php>

¹²<https://tailscale.com/>

¹³<https://unity.com/pt>

- CPU: AMD Ryzen 5 5600X 6-Core Processor 3.7Ghz;
- GPU: NVIDIA Geforce RTX 3060 12GB;
- RAM: 32GB DDR4 3666Mhz;
- Sistema Operacional Ubuntu Linux 22.04 LTS.

O hardware escolhido desempenhou um papel crucial na eficiência e na rapidez do processamento das animações. Utilizou-se um processador de alto desempenho, com múltiplos núcleos para lidar com a carga de trabalho intensa e paralelizar tarefas. A memória RAM disponível garantiu que grandes arquivos de imagem e vídeo pudessem ser manipulados, em parceria com os recursos da aceleradora gráfica. E a GPU foi empregada para acelerar os processos de renderização e de remoção de fundo, beneficiando-se de suas capacidades de processamento paralelo.

Para criação do servidor onde a ferramenta esteve disponível para testes e validação dos *testers* convidados, utilizou-se uma máquina com as seguintes configurações:

- CPU: Intel(R) Core(TM) i7-7700 CPU @ 3.6GHz;
- GPU: NVIDIA GeForce GTX 1070 8GB;
- RAM: 16GB DDR4 3666Mhz;
- Sistema Operacional Ubuntu Linux 22.04 LTS.

3.3. Desenvolvimento

A Figura 2 ilustra o funcionamento da ferramenta. Primeiramente, a interface web da Stable Diffusion é inicializada a partir de um interpretador de comandos (modo terminal) utilizando a flag `--api`. Esta flag habilita a opção de API, permitindo a integração com ferramentas de terceiros para a geração de personagens. Após a inicialização, a interface permanece em modo standby, aguardando requisições para serem processadas. Este modo de operação permite uma maior flexibilidade e eficiência no processamento de dados, garantindo que as solicitações dos usuários sejam atendidas de forma rápida e precisa. A utilização da API facilita a automação e a escalabilidade do sistema, tornando-o uma solução robusta para projetos que exigem geração dinâmica de conteúdo visual.

Na parte de frontend, para receber a entrada do usuário, foi desenvolvida uma página web utilizando Flask. Nesta interface, conforme ilustrado na Figura 3, há diversos elementos de seleção que permitem ao usuário escolher diferentes tipos de geração, como estilo, tipo de personagem e um *prompt* personalizado que será utilizado para a criação de personagens totalmente dinâmicos, conforme a entrada do usuário. Esta abordagem proporciona uma experiência interativa e personalizada, facilitando a criação de conteúdo visual adaptado às necessidades específicas de cada usuário.

Quando uma requisição é realizada pela interface, a função de geração do personagem é executada pelo Python. Inicialmente, todas as informações fornecidas pelo usuário são processadas, criando um corpo de requisição (*body JSON*) a ser enviado para a interface da Stable Diffusion, responsável pela geração de uma imagem contendo nove frames de animação do personagem 2D.

Para gerar uma animação em que todos os nove frames sejam distintos e sigam um mesmo fluxo de movimento, utiliza-se a técnica ControlNet. Esta técnica consiste em usar uma imagem que define pontos específicos onde a Stable Diffusion pode desenhar. No

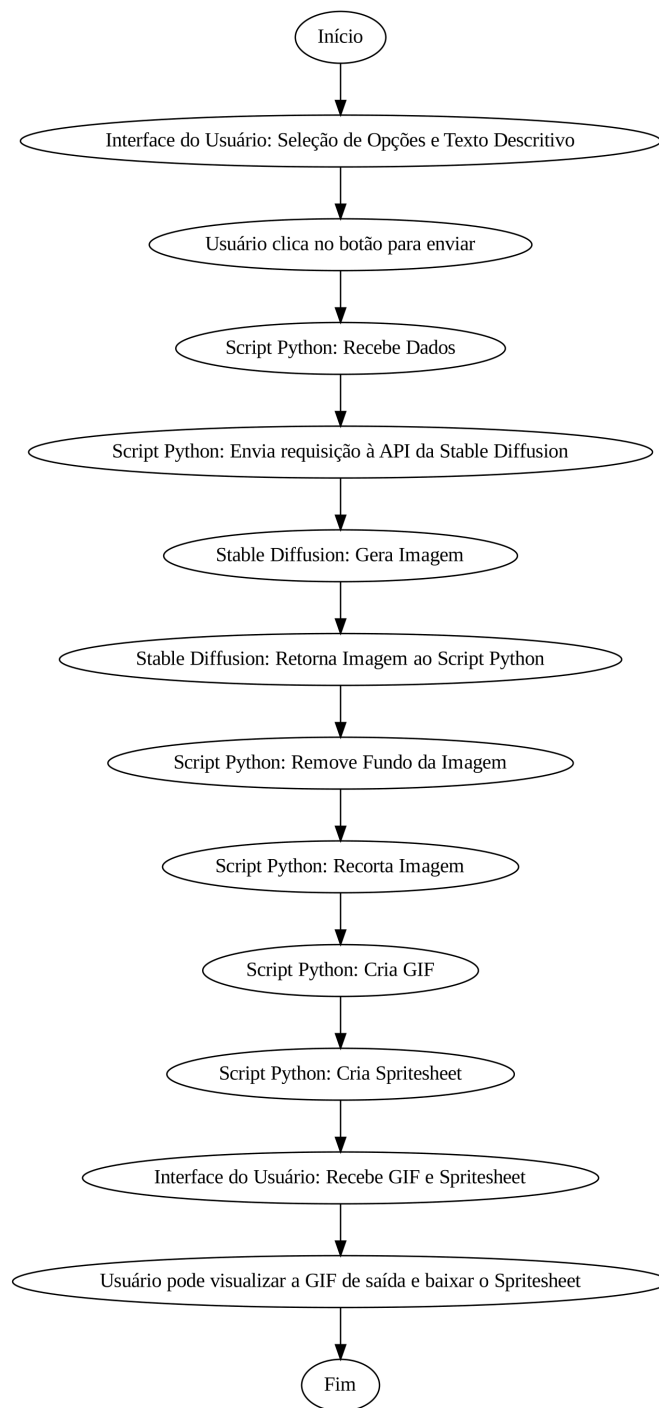


Figura 2. Método de funcionamento da ferramenta 2D Sprite Sheet Generator.

caso, como estão sendo criados apenas personagens humanoides e quadrúpedes, foram utilizadas apenas imagens de esqueleto desses tipos. Na Figura 4 e na Figura 5, é possível observar como são os esqueletos e obter uma ideia da estrutura da imagem final.

Para criação dos esqueletos utilizados na geração dos personagens, optou-se por uma técnica que utiliza a ControlNet da interface da Stabble Diffusion, juntamente com um *preprocessor* do tipo "*openpose*", e o modelo chamado "*control_openpose-fp16*". Ao

Gerador de SpriteSheets em 2D via *Stable Diffusion*

Como usar:

Digite características que você gostaria que seu personagem 2d possuísse, em inglês, por exemplo: "old man, with a black leather coat", e selecione qual estilo de geração você gostaria, para gerar um animal, selecione o tipo "animal", no final será gerado um gif de amostra do seu personagem e um botão para baixar sua spritesheet que pode ser importada para qualquer ferramenta de criação de jogos

Observações:

Como se trata de uma ferramenta experimental que ainda está em fase de testes, existe grande possibilidade que algum frame tenha cores diferentes e que existam problemas com membros como braços e pernas, então caso isto venha a acontecer tente gerar novamente utilizando mais detalhes na entrada

Texto de entrada (prompt):

ex: wizard, grey beard, green boots ...

Tipo de personagem:

Humanoide

Estilo:

Realista

Ação:

Andando

Gerar

Figura 3. Amostra visual da interface do 2D Sprite Sheet Generator

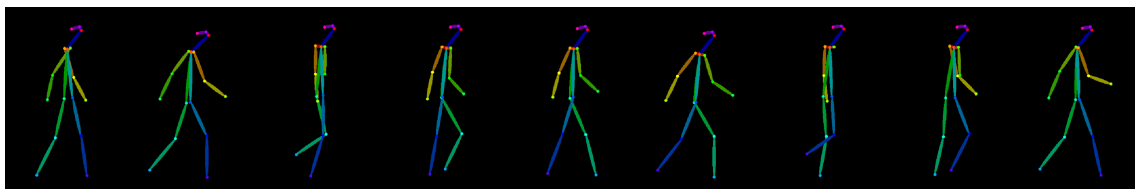


Figura 4. Esqueleto de humanoides utilizado na ControlNet.



Figura 5. Esqueleto de quadrúpedes utilizado na ControlNet.

inserir uma imagem de um personagem 2D utilizando estas configurações, um esqueleto é gerado imitando a mesma pose do personagem utilizado. Após realizar este processo para N frames de um único personagem, basta juntar os esqueletos em uma única imagem, que por fim será utilizada na geração da *spritesheet* final.

Após geração da *Stable Diffusion*, uma imagem de saída Figura 6 é gerada, retornando ao processo Python que em seguida deve realizar alguns ajustes para que a animação final seja criada. Primeiro, a imagem é recortada em nove partes iguais, considerando que cada frame possui o mesmo tamanho. Em seguida, utilizando a ferramenta do *backgroundremover*, o fundo de cada frame é alterado para transparente.



Figura 6. Imagem de saída da Stabble Diffusion.

Por fim, utilizando o *Imagemagick*, um arquivo GIF de amostra é criado à partir de todos os frames transparentes, além da *spritesheet* de saída final que basicamente consiste na união de todos os frames transparentes em uma mesma imagem.

Para ilustrar, seguem alguns exemplos de produtos finais gerados pela ferramenta proposta:

- Na (Figura 7), pode-se observar uma *spritesheet* de saída de um leão, utilizando a palavra chave “lion” e o modelo de geração de imagens do estilo *realista*.



Figura 7. *Spritesheet* de saída utilizando o estilo “realista”, com o *prompt* “lion”.

- Na (Figura 8), pode-se observar uma *spritesheet* de saída de um leão futurista, utilizando a palavra chave “lion” e o modelo de geração de imagens do estilo *cyberpunk*.



Figura 8. *Spritesheet* de saída utilizando o estilo “cyberpunk”, com o *prompt* “lion”.

- Na (Figura 9), pode-se observar uma *spritesheet* de saída de um gato laranja, utilizando a palavra chave “orange cat” e o modelo de geração de imagens do estilo *pixelart*.



Figura 9. *Spritesheet* de saída utilizando o estilo “pixelart”, com o *prompt* “orange cat”.

- Na (Figura 10), pode-se observar uma *spritesheet* de saída de um vampiro, utilizando a palavra chave “vampire” e o modelo de geração de imagens do estilo *realista*.



Figura 10. Spritesheet de saída utilizando o estilo “realista”, com o prompt “vampire”.

3.4. Avaliação

Para validação da ferramenta, foi conduzido um estudo piloto usando amostragem por conveniência. Sete pessoas foram convidadas para realizar testes de geração de personagens, bem como avaliar a facilidade e a utilidade da ferramenta. Os participantes voluntários são artistas e desenvolvedores da área de games.

Para o experimento, um túnel foi criado utilizando a ferramenta do Tailscale, onde um domínio foi gerado para acesso remoto. Como a geração com Stable Diffusion utiliza grande recurso da placa de vídeo, os testes ocorreram de forma agendada, onde cada pessoa podia testar em um dia específico de forma que não existisse concorrência da máquina que estava sendo utilizada na aplicação.

O procedimento dos testes foi o seguinte: duas tarefas específicas foram enviadas aos *testers*, bem como um formulário do Google Forms ¹⁴ onde posteriormente responderam um questionário de aceitação de tecnologia - TAM [Davis et al. 1989] e um conjunto de perguntas abertas.

As tarefas pré-definidas foram estas:

- Gerar inicialmente um personagem simples, um leão. Para isso, colocando no *prompt* a palavra “*lion*”, como tipo de personagem um “*animal (quadrupede)*” e selecionando o estilo que desejar;
- Gerar um segundo personagem simples, um duende. Para isso, colocando no *prompt* a palavra “*hobgoblin*”, como tipo de personagem um “*humanoide*” e selecionando o estilo que desejar.

Após a execução destas tarefas, os *testers* foram orientados a gerar animações de livre escolha, utilizando palavras-chave ou expressões em Língua Inglesa, separados por vírgula, sendo o mais específico possível. Por exemplo: “*golf player, green t-shirt, red pants*”.

No questionário TAM, foram apresentadas afirmações relacionadas à facilidade de uso e a utilidade da ferramenta, utilizando uma escala Likert de cinco pontos, onde 1 representa “discordo totalmente” e 5 representa “concordo totalmente”. Em seguida, respondiam três perguntas abertas para relatar, subjetivamente, aspectos positivos e negativos da aplicação em relação à usabilidade e à utilidade, bem como registrar elogios, críticas e sugestões de melhoria.

4. Resultados e Discussões

As subseções seguintes apresentam os resultados obtidos nos testes de validação da ferramenta, seguidos de análise e discussão dos mesmos.

¹⁴<https://www.google.com/forms/>

4.1. Resultados

A Tabela 1 apresenta as afirmações utilizadas no questionário TAM de aceitação da tecnologia, e a distribuição das avaliações e as médias obtidas. Considerando a média geral, observa-se que a ferramenta proposta apresenta avaliações muito boas para a facilidade de uso. Por outro lado, nota-se que as avaliações são inferiores para a utilidade da solução.

Perguntou-se também se os usuários *testers* recomendariam a ferramenta para outros profissionais da área, onde 57,1% (quatro votos) responderam que sim, e 42,9% (três votos) responderam que não.

Quanto às perguntas abertas, verificou-se respostas similares, apontando como principais aspectos positivos a facilidade e simplicidade da **2D Sprite Sheet Generator**, bem como o potencial dela como ferramenta geradora de sprites. Por outro lado, os aspectos negativos relatados se concentram na demora na geração dos personagens, a falta de mais opções de animação para geração (como ações de pular e abaixar) e, principalmente, a qualidade da geração de personagens - que ocasionalmente podem ter membros do corpo ou cores não condizentes com o *prompt* de entrada do usuário.

4.2. Discussões

Observando a Tabela 1, nota-se uma grande variação nas respostas, principalmente em perguntas relacionadas à utilidade e desempenho da ferramenta. Provavelmente, esse comportamento pode ter relação com os recursos disponíveis do servidor de testes, que possui menos memória de vídeo (8 GB) em comparação com a máquina utilizada no desenvolvimento (12 GB). Dessa forma, a qualidade dos sprites, a assertividade da requisição, e o tempo consumido para geração do conteúdo pelos *testers* acabou não corresponderam às expectativas deles.

Isto foi perceptível não somente pela avaliação dos questionários, mas também comparando resultados obtidos nos testes realizados durante o desenvolvimento da ferramenta usando o mesmo *prompt* de entrada. É possível observar na Figura 11 uma *spritesheet* gerada utilizando a máquina com menos capacidade, resultando em frames menos consistentes em cores e posição de membros. Por outro lado, a Figura 12 foi gerada utilizando a máquina de capacidade superior. Nota-se que os frames possuem melhor consistência, além de melhor qualidade de saída.

Para ser possível rodar o **2D Sprite Sheet Generator** em uma máquina com menos memória de vídeo, havia apenas duas possibilidades: diminuir consideravelmente o tamanho das imagens de entrada e saída, o que resultaria em gerações com muito menos qualidade e assertividade, ou, como foi optado, utilizar a flag `--lowvram` no processo da Stable Diffusion. Essa abordagem culminou em gerações mais demoradas, pois a memória precisava ser esvaziada toda vez que uma nova geração era feita. Assim, o modelo precisava sempre ser recarregado, alocado, e com nova ativação neuronal para proceder com novas predições.

Esse fato acaba diminuindo a assertividade das gerações finais, que podem apresentar problemas como membros faltantes ou cores inconsistentes entre os frames da animação. Caso a flag não fosse utilizada, o processo da Stable Diffusion retornaria exceções durante cada geração, devido a um fenômeno conhecido como “*out of memory*” ou “*memory overflow*”, que é basicamente a falta de memória livre durante o processamento dos dados.

Tabela 1. TAM: distribuição das avaliações e médias gerais dos participantes, avaliando aspectos de facilidade de uso percebida e utilidade percebida.

Afirmativa	Escala 1	Escala 2	Escala 3	Escala 4	Escala 5	Média±DP
Quanto à Facilidade de Uso Percebida:						
Usar o 2D Sprite Sheet Generator é uma boa ideia para gerar sprites automaticamente.	0	1 (14,3%)	1 (14,3%)	2 (28,6%)	3 (42,9%)	4,00±1,07
Usar o 2D Sprite Sheet Generator facilita o processo de geração de sprites.	0	0	2 (28,6%)	2 (28,6%)	3 (42,9%)	4,14±0,83
É fácil acessar os recursos do 2D Sprite Sheet Generator e aplicá-los na geração de sprites.	0	0	0	3 (42,9%)	4 (57,1%)	4,57±0,49
É fácil aprender a usar o 2D Sprite Sheet Generator.	0	0	0	1 (14,3%)	6 (85,7%)	4,86±0,35
Quanto à Utilidade Percebida:						
O 2D Sprite Sheet Generator é preciso na criação de personagens.	2 (28,6%)	1 (14,3%)	3 (42,9%)	0	1 (14,3%)	2,57±1,29
O 2D Sprite Sheet Generator apresenta um bom desempenho, retornando gerações em tempo aceitável.	2 (28,6%)	1 (14,3%)	2 (28,6%)	0	2 (28,6%)	2,86±1,55
O 2D Sprite Sheet Generator é importante e adiciona valor às tarefas de geração de sprites.	1 (14,3%)	1 (14,3%)	1 (14,3%)	2 (28,6%)	2 (28,6%)	3,43±1,40
O 2D Sprite Sheet Generator é útil como ferramenta de apoio para geração de sprites.	1 (14,3%)	2 (28,6%)	0	0	4 (57,1%)	3,57±1,68

Para evitar a flag `--lowvram` e, consequentemente, melhorar a qualidade das gerações, é preciso utilizar uma placa de vídeo com especificações superiores, para que exceções como “*memory overflow*” não ocorram.

A ferramenta demonstrou uma disparidade significativa entre a facilidade de uso e a percepção de utilidade. Embora a interface seja intuitiva e de fácil navegação, a utilidade foi avaliada de forma menos favorável. Esse contraste ocorreu porque a máquina utilizada para os testes pelos usuários era inferior à utilizada durante o desenvolvimento. Como resultado, as gerações de imagens foram mais lentas e os resultados menos satisfatórios, o que impactou negativamente a avaliação da aplicação em termos de utilidade. A performance do sistema, portanto, está diretamente ligada à capacidade da máquina utilizada, influenciando a experiência do usuário e a percepção de valor da aplicação.



Figura 11. Spritesheet de saída utilizando o estilo “realista”, personagem de tipo humanoide, com o prompt “lion”, utilizando a máquina de capacidade inferior e a flag `-lowvram`.



Figura 12. Spritesheet de saída utilizando o estilo “realista”, personagem de tipo humanoide, com o prompt “lion”, utilizando a máquina de capacidade superior.

Apesar dos resultados apresentarem respostas negativas sobre a utilidade e desempenho da ferramenta, é possível ressaltar alguns pontos positivos, como a usabilidade da interface, que possui uma rápida curva de aprendizado e proporciona uma boa experiência de usuário, além do grande potencial da ferramenta.

Quanto aos aspectos positivos observados na Tabela 1, destaca-se o item “É fácil aprender a usar o **2D Sprite Sheet Generator**”, onde 85,7% dos *testers* atribuíram a maior avaliação na escala Likert. Dentre os feedbacks recebidos sobre a usabilidade, destacam-se comentários como “A ferramenta é super simples de usar. Após ler rapidamente a descrição e utilizá-la uma vez, já é possível entender todo o funcionamento”, “O prompt em formato texto deixa a ferramenta muito aberta para gerar qualquer tipo de sprite, facilitando a utilidade em qualquer tipo de projeto”, “A barra de progresso ajuda muito pra saber o tempo que os sprites vão demorar para serem gerados”.

Outro aspecto positivo encontrado ao se observar a Tabela 1, é o feedback recebido sobre a pergunta “O **2D Sprite Sheet Generator** é útil como ferramenta de apoio para geração de sprites.”. Nesse quesito, mais da metade dos participantes (57,1%) concordam plenamente na afirmação, apesar das limitações levantadas. De acordo com um dos *testers*, “o aspecto positivo é que ela pode dar um bom começo para o artista e depois ele só precisa corrigir os erros”. Isso indica que a ferramenta pode ser utilizada no início da criação dos personagens, como forma de inspiração para o artista desenvolvedor. Essa avaliação também corrobora com o teste realizado antes do estudo piloto, onde importou-se sprites em projeto na Unity, e foi possível criar personagens 2D e animações a partir dos *spritesheets* gerados pela **2D Sprite Sheet Generator**.

Por fim, após recebimento e análise de todo o feedback recebido dos *testers* da aplicação, destaca-se como potencial da ferramenta **2D Sprite Sheet Generator** sua capacidade de entregar um gerador de personagens 2D que funciona de forma dinâmica, utilizando do grande potencial da IA generativa. E, segundo feedback dos *testers*, inova em um segmento que está carente de soluções no mercado de desenvolvimento de jogos,

e pode ser utilizada facilmente por equipes de desenvolvimento pequenas e em encontros de desenvolvedores de jogos, como a Global Game Jam ¹⁵.

5. Conclusão

Este trabalho apresentou o **2D Sprite Sheet Generator**, uma ferramenta para geração dinâmica de animações de personagens utilizando Stable Diffusion. O objetivo de propor esta abordagem foi oferecer uma aplicação capaz de diminuir o tempo de criação de animações de personagens 2D e possibilitar o rápido desenvolvimento de animações por meio de recursos de IA. Um estudo piloto foi conduzido com artistas e desenvolvedores da área de games, considerando aspectos de aceitação da tecnologia proposta.

Com a realização deste estudo e desenvolvimento da ferramenta, verificou-se que a utilização de ferramentas de IA como a Stable Diffusion na criação de componentes visuais dinâmicos, como personagens 2D, possui grande potencial futuro e pode revolucionar o mercado criativo. De acordo com os *testers* convidados, o **2D Sprite Sheet Generator** é simples de usar, intuitiva e permite facilmente gerar animações. Eles também comentaram que a exibição na interface e a opção de download do spritesheet completo é uma ideia muito boa e bem executada.

Os participantes do experimento também perceberam que a ferramenta ainda apresenta muitos problemas de utilidade, como precisão, desempenho e assertividade nas gerações finais. Além disso, destacaram que ela ainda carece de mais opções de geração, como novas ações (pular, abaixar, etc.) e outros estilos de personagens. Observou-se que muitos destes problemas tem relação com a disponibilidade de hardware para execução plena do modelo. Ademais, os participantes reforçaram que a solução é ainda incipiente e que, com ajustes finos e melhoria na capacidade de treino, a ferramenta pode se tornar um grande trunfo na área de games.

Para trabalhos futuros, recomenda-se melhorar os seguintes pontos:

- Adicionar mais opções de ações, como pular, abaixar, correr, entre outras;
- Adicionar mais frames na animação do personagem para melhorar a fluidez do movimento (para isso, recomenda-se utilizar um hardware superior ao deste estudo, que suporte o processamento de imagens maiores);
- Encontrar uma forma de melhorar a assertividade das gerações, que tendem a apresentar problemas como membros do corpo do personagem 2D em posições incorretas, como braços à frente do corpo quando deveriam estar atrás, por exemplo;
- Melhorar o tempo de geração, que provavelmente está relacionado com a potência da máquina utilizada e a falta de técnicas de otimização;
- Validar a ferramenta novamente utilizando um hardware superior e com uma amostra maior de *testers*.

Referências

Assunção, F. G. d. (2012). *Animação em Camadas com Quadros-Chave Espaciais e Edição de Trajetórias*. PhD thesis, Universidade Federal do Rio de Janeiro.

¹⁵<https://globalgamejam.org>

- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management science*, 35(8):982–1003.
- Feijó, B. & Badaró, P. (2006). *Conceitos e modelos para um sistema brasileiro de produção de conteúdo digital*. PUC.
- Fukaya, K., Daylamani-Zad, D., & Agius, H. (2023). Intelligent generation of graphical game assets: A conceptual framework and systematic review of the state of the art. *arXiv*.
- Harrison, H. L. H. & Hummell, L. J. (2010). Incorporating animation concepts and principles in stem education. *Technology and Engineering Teacher*, 69(8):20.
- Januário, V. D. & Calegario, F. (2023). Char sheets generator: Em direção a uma ferramenta para geração automática de animações frame a frame.
- Kuperberg, M. (2002). Character animation fundamentals developing skills for 2d and 3d character animation.
- Rantala, T. (2013). *Animation of a High-Definition 2D Fighting Game Character*. PhD thesis, Kajaani University of Applied Sciences, School of Business.
- Roberts, S. (2012). *Character animation fundamentals: developing skills for 2D and 3D character animation*. Routledge.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2024). Stable diffusion.
- Saputra, D. I. S., Manongga, D., & Hendry, H. (2021). Animation as a creative industry: State of the art. In *2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 6–11. IEEE.
- Serpa, Y. R. & Rodrigues, M. A. F. (2019). Towards machine-learning assisted asset generation for games: A study on pixel art sprite sheets.
- Shahbazi, N. (2024). How long does it take to create 2d animation? Disponível em <https://pixune.com/blog/how-long-does-2d-animation-take/>. [Acessado em Maio 2024.].
- Silveira, T. Q. V. d. (2017). Seiren: curta animado infantil que discute representatividade social. Technical report, Universidade de Brasília, Trabalho de Conclusão de Curso (Bacharelado em Desenho Industrial).
- Xi, C. & Chung, J. (2023). A study on character design using [midjourney] application. *International Journal of Advanced Culture Technology*, 11:409–414.