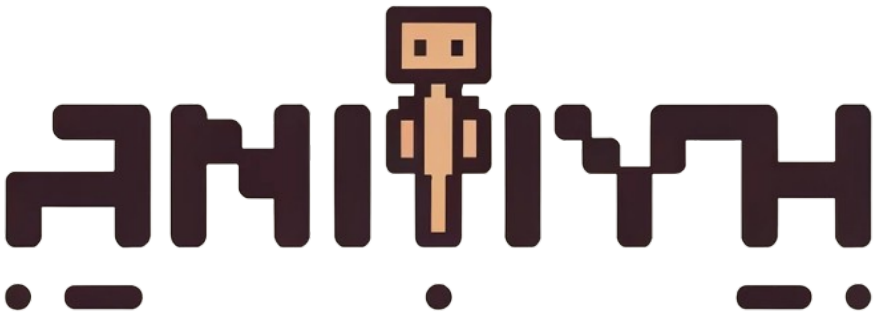
Animyth - Final Report

A tool for assisting creation of 2D animation for game development

by Sky Hou & Sherry Xu



Word Count: 1975
Penalty: 0%

Introduction

Animyth revolutionizes game development by empowering teams, particularly those with limited artistic skills, with the ability to quickly transform textual descriptions into dynamic sprite sheets. This innovation leverages Large Language Models and image generation technologies, addressing a critical bottleneck in game design. Animyth not only streamlines the creative process but also democratizes game development, making it accessible and efficient regardless of resource limitations. It exemplifies the potent role of machine learning in bridging the gap between concept and visual creation.

System Illustrations

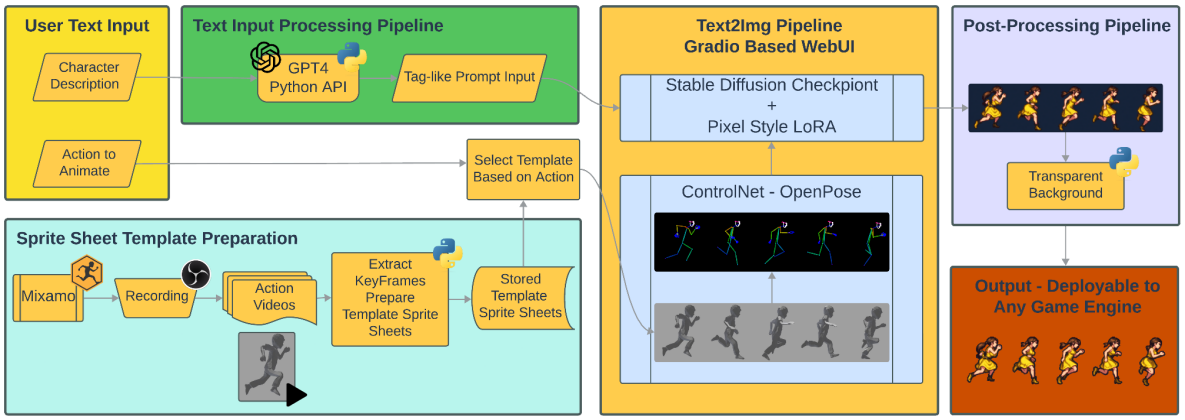


Figure 1. Animyth System Diagram

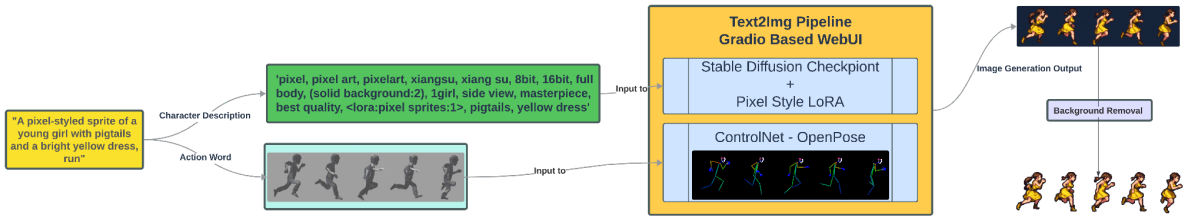


Figure 2. Animyth Data Flow Example

Background

The "Tune-A-Video" method is pertinent to our project as it showcases the potential of diffusion models in generating video from text, which is a step towards our goal of creating animated sprites for games. While it offers a framework for ensuring content consistency, what it may lack is the specific optimization for the unique constraints and stylistic requirements of game animation, such as the need for loopable sequences and actions that correspond to gameplay mechanics [1].

The "VideoControlNet" framework, utilizing diffusion models with ControlNet for video translation, aligns with our text-to-image generation work by advancing the application of diffusion technology to create coherent sequences. It demonstrates how initial frames guide subsequent frame generation for fluid motion, a principle applicable to our aim of generating consistent imagery from text [2].

Animyth draws inspiration from the cutting-edge field of text-to-continuous-motion generation, as seen in innovative methods like 'Tune-A-Video' and 'VideoControlNet'. The project leverages the text processing prowess of GPT-4 [3] in conjunction with stable diffusion models to develop a system that adeptly generates coherent images from textual descriptions.

Input Dataset

Our project's dataset, designed for Animyth, consists of 24 manually created samples describing pixel-styled characters performing actions typical in platformer games. It features 8 unique characters, each depicted in 3 key actions: idle, run, and jump. An example description is

'A pixel-styled sprite sheet of a girl with short hair and black hoodie, idle'

emphasizing character appearance and action.

Due to the generative design of Animyth, our dataset serves exclusively as input. Traditional labeled or test datasets are not required in this context. The effectiveness of our model is instead assessed through a specialized method that quantifies qualitative aspects, which will be elaborated on later. The process of creating this dataset ensures that the descriptions authentically mirror potential user inputs while rigorously testing the model's capability to produce a variety of sprite sheets.

Architecture and Software

Animyth is designed to convert textual character descriptions and actions into detailed sprite sheets, seamlessly integrating into game engines.

Detailed Component Overview (Refer to Figure 1)

User Text Input Block: Captures the character description and action word, serving as the starting point for the sprite generation process.

Text Processing Block: Transforms the character description into a tag-like prompt using GPT4 [3]. This conversion is key for aligning the input with the capabilities of the Stable Diffusion model.

Sprite Sheet Template Preparation Block: Employs Mixamo animation [4] keyframes to derive sprite sheet templates. These templates are critical for guiding the generation process, ensuring that the final output aligns with the specified action.

Text to Image Block with Stable Diffusion and LoRA within Stable Diffusion WebUI: This critical component of Animyth's image generation process uses a custom-configured Stable Diffusion model, housed within the Stable Diffusion WebUI [5]. LoRA (Low-Rank Adaptation) [6] is seamlessly integrated into this model, enhancing its ability to finely tune and control the style of the generated images, a feature particularly effective for preserving the pixel-art aesthetic. The Stable Diffusion WebUI serves as the interactive interface for this entire process, facilitating the integration and operation of the model with the user's input. This synergistic combination of the Stable Diffusion model, LoRA, and the WebUI interface enables the creation of high-quality, stylistically consistent images that are closely aligned with the input prompts.

ControlNet and OpenPose Synergy: These two advanced tools bridge the gap between the sprite sheet templates and the final image generation. ControlNet leverages the power of neural networks to guide the image generation process, ensuring the poses and actions in the generated images match those in the sprite sheet template [7]. OpenPose complements this by providing accurate pose detection and alignment, enhancing the fidelity of the character movements and poses in the generated sprites [8].

Post-Processing Block: A Python script removes backgrounds from the generated images, a vital step in preparing the sprites for direct use in game engines, enhancing the usability and integration capability of the final outputs.

Data Flow

Figure 2 illustrates Animyth's data flow, matching the color scheme of Figure 1 for clarity. It starts with user text input being transformed into a tag-like prompt while simultaneously selecting a sprite sheet template based on an action word. These elements are input into the Stable Diffusion WebUI for image generation, aligning with the template's poses. The final step involves background removal, resulting in the creation of a ready-to-use sprite sheet.

Key Parameters and Techniques

For detailed system parameters and configurations, please refer to the Appendix: Stable Diffusion WebUI Details.

Prompt Engineering For Text Processing

The text processing block in Animyth is key to converting detailed character descriptions into tags for image generation with Stable Diffusion models. This conversion is vital for a user-friendly interface, especially for new users unfamiliar with specific tags or finding direct tag input intimidating.

Evolution of the Prompt Engineering Process

We employed prompt engineering for both GPT4, to convert sentences into tag-like prompts, and for the Stable Diffusion model, to optimize image generation based on these text inputs.

Based on our input dataset, the character descriptions were straightforward, allowing for the generation of desired tags without the need for chain-of-thought prompting. However, we recognize that more complex character designs might benefit from this approach in the future.

A significant part of our development involved iterating to identify the most effective tags. While many tags were suggested by the Stable Diffusion model provider, we found that not all were universally applicable. Notably, adjusting tags like '(((solid background)))' to '(solid background:2)' improved consistency in background generation, and altering '((side view))' to 'side view' diversified character action representations. See Appendix for more information about the tags.

Prompt

"This is my input example
'A pixel-styled sprite sheet of a boy wearing white hoodie and black shorts, run'

output should look like
'pixel,pixel art,pixelart,xiangsu,xiang su, full body,(solid background:2),1boy, side view, masterpiece,best,quality, white hoodie, black shorts, run'

You may add your own tags to enhance the appearance of the character like the colour of hair or clothes.

Always include "pixel, pixel art, pixelart, xiangsu, xiang su, 8bit, 16bit, full body, (solid background:2), side view, masterpiece, best quality" in the output.

If it is a girl or a boy, you should say 1girl or 1boy"

Quantitative Evaluation

To provide an objective assessment of Animyth, we employed a comprehensive evaluation rubric (see Appendix), targeting key aspects such as character consistency, animation smoothness, the necessity for manual editing, and how well outputs match descriptions. This rubric operates on a five-point scale, enabling us to quantitatively analyze each sprite sheet in a systematic and unbiased manner. This approach is particularly useful for our project, as it quantifies qualitative aspects, facilitating clear comparisons and data aggregation in a context where traditional quantitative metrics are challenging to establish. Each sprite sheet undergoes a thorough review against these criteria, resulting in an overall quality rating that reflects both the model's strengths and areas needing improvement.

Average Scores Across Key Aspects (see Appendix for more details):

- **Character Consistency:** 4.5
- **Animation Smoothness:** 4.3
- **Necessity for Manual Editing:** 3.9
- **Matching Description:** 4
- **Overall:** 4.2

Qualitative Insights from Model Outputs

Our model's performance was thoroughly evaluated using the rubric, with each criterion rated from 1 to 5. For instance:

> **Optimal Performance:** The sprite sheet shown in Figure 3 scored perfectly. This demonstrates the model's capability in accurately generating sprites from straightforward descriptions.

Figure 3. "A pixel-styled sprite of a young girl with pigtails and a bright yellow dress, jump" (c8 a3)

Challenges in reliability given simple input: The example shown in Figure 4 highlighted some inconsistencies. It scored 3.5 for character consistency and 3 for the need for manual editing, indicating occasional unreliability even with simpler inputs.

Figure 4. "An elderly man with a white beard and a fishing vest, run" (c7 a2)

Complex Character Handling: The robot example shown in Figure 5 scored below 3 in both editing necessity and description match, underscoring difficulties with complex or non-human characters.

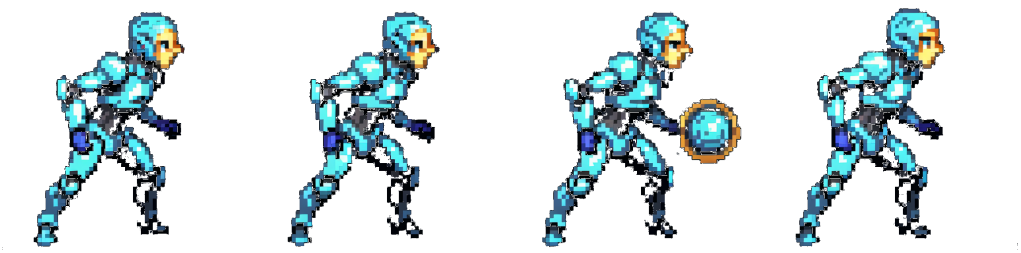


Figure 5. "A pixel-styled sprite of a robot with sleek silver design and blue accents, idle" (c6 a1)

(Refer to the Appendix for detailed evaluation results.)

These examples illustrate the model's varied performance across different character types and actions. The model demonstrates proficiency in generating sprites from straightforward descriptions. However, it faces challenges when dealing with complex or intricately detailed characters, often requiring additional post-processing or enhanced prompt engineering to produce more detailed and accurate tags. The qualitative assessments, when converted into quantitative scores, offer a comprehensive understanding of where the model thrives and where it requires further development.

Discussion and Learning

The performance of Animyth has been notably impressive, especially considering the simplicity of its input text and the high quality of the output sprite sheets. However, our evaluation has highlighted some areas for improvement and learning opportunities for future projects.

Key Observations:

1. **Template Limitations:** Currently, Animyth generates sprite sheets based on fixed templates, leading to similar actions across different characters. This limits the diversity in character behaviors, a critical aspect in game design. To address this, incorporating tools like ActionGPT [9] could diversify action templates, allowing for a broader range of character movements and behaviors.
2. **Background Removal Challenges:** The simplicity of our background removal method has occasionally resulted in imperfections, such as tiny holes in the characters. A more sophisticated approach, potentially using contour-based methods, could enhance the quality of the final sprite sheets.
3. **Areas Needing Improvement:** The model has scored lower in areas such as 'Matching Description' and 'Need for Manual Editing'. The former could be attributed to the limited range of tags used during the training of the text-to-image model, suggesting a need for more diverse and suitable stable diffusion checkpoints. The latter is somewhat expected, given the necessity of human intervention to refine outputs and address copyright concerns.

Future Enhancements:

- **Adopting an All-in-One Sprite Sheet Template:** This would likely improve consistency across animations, ensuring uniformity in character portrayal.
- **Automating with Stable Diffusion WebUI API [5]:** Implementing this API could make Animyth fully autonomous, streamlining the process from input to final output.
- **Developing a Robust Evaluation Rubric:** A more structured evaluation framework is needed to quantitatively and qualitatively assess the system's performance. This would enhance the credibility of results and provide clear guidance for improvements.

In conclusion, while Animyth has shown promising results, these insights and potential enhancements pave the way for creating more versatile, efficient, and user-friendly tools for indie game developers.

Individual Contributions

- Sky's contribution to the project
    - Recorded the action videos from Mixamo
    - Implemented python scripts to process the action videos and obtain the template sprite sheets
    - Implemented a python script for background removal
    - Setup the text-to-image pipeline
        - Setup everything essential to generating the sprite sheet, including downloading models, setup softwares, etc.
        - Obtaining and tuning the tags to the stable diffusion model
    - Generated the 24 outputs
    - Came up the aspects in the rubric
    - Evaluated 12 model outputs
- Sherry's contribution to the project
    - Collected the dataset (24 samples) for the project
        - Manually created 8 character's descriptions
        - Create combinations with the 3 actions
    - Implement the Text Input Processing pipeline
        - Iteratively prompt engineered GPT4 to convert text inputs to tag-like prompts
        - Implemented a python script to make the process autonomous and controllable
    - Designed the evaluation rubric given the aspects Sky provided
    - Evaluated 12 model outputs

References

[1] J. Z. Wu et al., “Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation,” *Papers With Code*, March 17, 2023, Available: https://paperswithcode.com/paper/tune-a-video-one-shot-tuning-of-image [accessed Dec. 8, 2023].

[2] Z. Hu and D. Xu, “VideoControlNet: A motion-guided video-to-video translation framework by using diffusion model with ControlNet,” *arXiv.org*, August 3, 2023, Available: https://arxiv.org/abs/2307.14073 [accessed Dec. 10, 2023].

[3] OpenAI, "GPT-4," OpenAI, March 14, 2023. [Online]. Available: https://openai.com/research/gpt-4.

[4] "Mixamo," Adobe Systems Incorporated, 2023. [Online]. Available: https://www.mixamo.com/

[5] AUTOMATIC1111, "Stable Diffusion WebUI," GitHub repository, 2022. [Online]. Available: https://github.com/AUTOMATIC1111/stable-diffusion-webui

[6] E. Hu, Y. Shi, P. Liang, and R. P. Adams, "LoRA: Low-Rank Adaptation of Large Language Models," in arXiv preprint arXiv:2106.09685, 2021. [Online]. Available: https://arxiv.org/abs/2106.09685.

[7] lllyasviel, "ControlNet," GitHub repository, [Online]. Available: https://github.com/lllyasviel/ControlNet

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8765346.

[9] S. S. Kalakonda, S. Maheshwari, and R. K. Sarvadevabhatla, “Action-GPT: Leveraging Large-scale Language Models for Improved and Generalized Action Generation,” arXiv preprint [Online]. Available: https://arxiv.org/abs/2211.15603, 2022.

Appendix

Tag Explanation

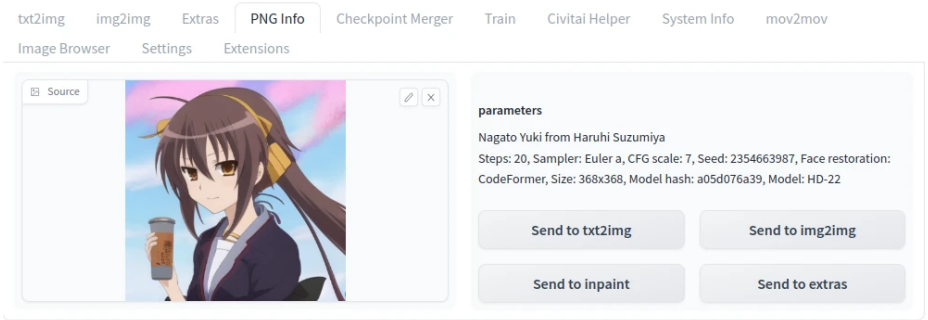
Tags in General: Tags are keywords or phrases that you input into the model. Each tag represents a characteristic, feature, or attribute that you want the model to consider when generating an image. For instance, tags can specify style (like 'pixel art'), content (like 'dragon'), or attributes (like 'red'). The model uses these tags to understand what elements should be present in the generated image.

Parentheses Around a Tag: Putting parentheses around a tag is a way to give that tag more emphasis or weight. When a tag is enclosed in parentheses, the model gives it higher priority compared to other tags. This means the model will try harder to ensure that the features represented by that tag are prominently included in the generated image. For example, if you use '(dragon)', the model understands that the dragon element is particularly important in your requested image.

Colon and Number (e.g., :2): Adding a colon and a number after a tag is a method of further specifying the weight or influence of that tag. The number typically indicates the degree of emphasis. A higher number means greater emphasis. This syntax is often model-specific and depends on how the developers have programmed the model to interpret these numbers. In some models, this might mean that a tag with ':2' is twice as influential in the image generation process as it would be without the number.

Stable Diffusion WebUI Details

Users can obtain the specific configuration used for outputed sprite sheets available in our project's GitHub repository. by uploading it to the PNG info tab in the WebUI.



Evaluation Rubric

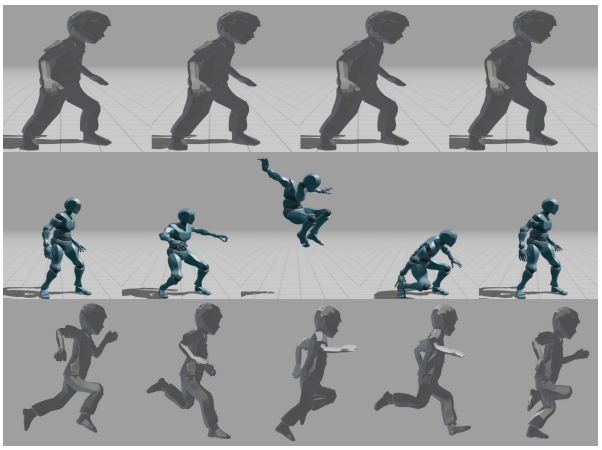
Aspect	Excellent (5 Points)	Good (4 Points)	Satisfactory (3 Points)	Needs Improvement (2 Points)	Poor (1 Point)
Character Consistency Across Frames	The character maintains consistent proportions, colors, and details across all frames.	Minor inconsistencies in details or colors, but overall consistent.	Noticeable inconsistencies in a few frames, but does not significantly impact the overall appearance.	Several inconsistencies are present, affecting the character's appearance.	Major inconsistencies across frames, significantly impacting the character's appearance.
Animation Smoothness	Movement is fluid and natural across frames.	Generally smooth with minor choppiness in places.	Movement is adequate but lacks fluidity in some sequences.	Frequent choppiness, impacting the quality of animation.	Jerky and unnatural movement, significantly disrupting the animation flow.
Need for Manual Editing	No manual editing is necessary; the sprite sheet is perfectly usable as is.	Minimal manual editing is needed; the sprite sheet is mostly ready for use.	Some manual editing is required, but it's relatively minor and manageable.	The sprite sheet needs noticeable manual editing, but some elements may be salvageable.	The generated sprite sheet requires extensive manual modification, such as background removal or significant detail editing.
Matching Description	The sprite sheet closely and accurately matches the input description or design, meeting the intended criteria.	The sprite sheet generally matches the input description, with only minor deviations.	The sprite sheet somewhat matches the input description but lacks accuracy in some aspects.	The sprite sheet has some elements that align with the description but largely misses the mark.	The generated sprite sheet bears little to no resemblance to the input description or intended design.

Evaluation Results

		Character Consistency Across Frames	Animation Smoothness	Need for Manual Editing	Matching Description	Overall Score
c1	a1	5	4	4	3	4
c1	a2	4	4	4	2.5	3.625
c1	a3	4.5	4	3.5	3	3.75
c2	a1	5	5	4.5	3.5	4.5
c2	a2	5	4.5	4.5	4.5	4.625
c2	a3	5	5	4.5	5	4.875
c3	a1	5	5	5	5	5
c3	a2	5	4.5	4.5	5	4.75
c3	a3	4.5	5	4	5	4.625
c4	a1	5	5	5	5	5
c4	a2	4.5	4	3	5	4.125
c4	a3	3	3.5	2	4	3.125
c5	a1	4	5	4	5	4.5
c5	a2	4.5	3.5	4.5	4.5	4.25
c5	a3	4.5	4	4	3	3.875
c6	a1	4.5	3	2	2.5	3
c6	a2	5	3	2.5	2.5	3.25
c6	a3	3.5	3	3	3	3.125
c7	a1	5	5	4.5	4.5	4.75
c7	a2	3.5	5	3	3	3.625
c7	a3	3	3	3	3	3
c8	a1	5	5	5	5	5
c8	a2	4	4	4.5	4.5	4.25
c8	a3	5	5	5	5	5
	Average	4.458333333	4.25	3.895833333	4	4.151041667

c# means the character id
a# means the action {a1=idle, a2=run, a3=jump}

All-In-One Sprite Sheet Example



Permission

Team Members	Permission to post video	Permission to post final report	Permission to post source code
Sky Hou	Yes	Yes	Yes
Sherry Xu	Yes	Yes	Yes