

Generative AI Processes for 2D Platformer Game Character Design and Animation

Shanhui Qiu^{1,a,*}

¹*Study of Interactive Art and Technology (SIAT), Simon Fraser University, Vancouver, Canada*
a. jgibson50992@student.napavalley.edu

**corresponding author*

Abstract: AI has the potential to revolutionize the time-consuming and technically complex process of 2D animation production. This paper specifically focuses on creating 2D character animations for platformer games using AI. While AI has made significant contributions to video animations, its application in 2D gaming animation is largely unexplored. Existing AI applications for animation mostly target video animations and lack effective control over randomness. Therefore, this paper explores the role of generative AI in 2D gaming animation, from character design to full animation. Software tools like ChatGPT, Midjourney, Stable Diffusion, and Unity are used to streamline the production process. The research aims to investigate the feasibility and potential of generative AI, with a focus on controlling randomness. By leveraging the unique features of each software, the study aims to enhance the production of 2D game animations. The final output will be an animated character in the “Idle” state, showcasing the potential of generative AI in 2D gaming animation.

Keywords: generative AI, ChatGPT, Midjourney, Stable Diffusion, Unity 2D

1. Introduction

The production of 2D character design and animation has traditionally been a time-consuming process that necessitates years of expertise in illustration. Despite advancements in computer graphics that have accelerated the production rate, the fundamental nature of 2D character design and animation creation has remained largely unchanged over the years [1-4]. However, the advent of Artificial Intelligence presents a pivotal turning point for this status quo. The primary objective of this paper is to employ “Generative AI” in the creation of 2D Platformer Game Character Design and Animation as a mediating bridge, exploring multiple possibilities for altering this inherent disadvantage. The possibilities alluded to in this article span multiple software platforms. The narrative incorporates the use of ChatGPT, Midjourney, the LoRa model and ControlNet within Stable Diffusion, Photoshop, and the Unity 2D Package. By synergistically combining the functionalities of these software in a modular fashion, an individual devoid of drawing expertise can still achieve satisfactory animation outcomes [5].

Furthermore, this paper will place particular emphasis on exploring the utilization methods of “Generative AI.” The characteristic feature of this tool is its “generative” nature, signifying that its outputs are inherently stochastic. Consequently, this introduces two divergent approaches for its application: First, leveraging the inherent randomness—meaning that irrespective of the user’s input, the tool autonomously interprets the data and produces an output, allowing the user to obtain

unexpected results. Second, controlling the randomness—to harness outputs that are utilizable, mastering the control of this stochastic nature becomes indispensable. As a result, understanding how to employ these two method-ologies and discerning the appropriate timing for their application emerges as critically important [6].

2. Character Design for 2D Platformer Game

Prior to initiating the animation process, it is essential to first conceptualize a character. The visual design of a game character encompasses numerous aspects; however, at this preliminary stage, it is sufficient to identify the character's fundamental attributes, such as age, gender, and appearance, which align with the game's thematic background. The authors are in the process of developing a 2D platformer game, and the character will be created in accordance with the narrative context of this game. The storyline centers on a futuristic city where the artist Eva and the AI entity Aion engage in a competitive endeavor to produce emotionally resonant art [7].

2.1. Character Design in ChatGPT

The initial step entails uploading the entire narrative content of the game to ChatGPT. This is implemented to regulate the inherent stochasticity of the tool, thereby ensuring that ChatGPT is well-informed about both the game's/character's background and the specific requirements set forth by the developer. Building upon this foundation, we instruct ChatGPT to generate ten alternative scenarios for the character “Eva,” leveraging the tool's randomization capabilities to yield a diverse array of options [8]. Two considerations are imperative in this phase: First, due to the persisting element of randomness in subsequent steps, it is advisable to focus on acquiring a general outline or primary features of the character rather than investing substantial time in intricate detailing. Based on the authors' experience, the characters generated in later stages often prove to be more engaging and offer a broader spectrum of options than preconceived notions of the character. Second, the number of generated scenarios should ideally not exceed ten, as exceeding this limit tends to result in homogenized outcomes [9]. Furthermore, these scenarios can be freely combined, implying that ten initial scenarios could potentially yield hundreds of unique combinations.

2.2. Secondary Character Design and Illustration in Midjourney

Upon obtaining a general character outline, the illustration process commences within Midjourney. The Prompt Template utilized within Midjourney is structured as follows:

[Art Type], [Artist], [Character Sheet, Multiple Expressions and Pose], [T-Shaped], [3/4 View + Three View], [Style], [Character Features and Clothing]

It is important to note that not every component of this Template is obligatory; its utilization is contingent upon individual requirements. This segment represents one of the optimal manifestations of leveraging and controlling stochasticity. The subsequent section will provide an analytical breakdown of the Template:

[Art Type]: This component is obligatory and provides the AI with explicit directives regarding the purpose and orientation of the image to be generated. In the given case study, the requisite field should be populated with “2D Game Character Design.”

[Artist]: This component is imperative for the purpose of ensuring that the AI generates images in a coherent style. This uniformity of style will also facilitate the generation of game skill/item icons as well as backgrounds. However, the content filled in does not necessarily have to be the name of an artist; it could also be a game, film, or other media. In the case study at hand, the authors have opted for Angus McKie, a renowned artist specializing in science fiction themes.

[Character Sheet, Multiple Expression and Pose]: This component is obligatory and serves to instruct the AI to clear the background while simultaneously obtaining various poses and expressions for the character. This facilitates a more comprehensive understanding of the character in question. Although these elements will not be utilized within the context of the 2D platformer game.

[T-shaped]: This component is obligatory. Employing this term can directly yield character poses that are applicable for use in a 2D platformer game. However, at the time of the character's initial creation, the authors had not yet discovered this term and continued to use "3/4 View + Three View," with results as illustrated in Figure 1 and Figure 2.

[3/4 View + Three View]: With the acquisition of the "T-shaped" term, this component is no longer obligatory. Through empirical testing, the authors have found that AI exhibits a misunderstanding of the 3/4 view commonly used in 2D platformer games. As can be observed in Figure 1 and Figure 2, specifically in image 4, the 3/4 view is misconceived as representing the character from a rear 3/4 perspective, whereas a frontal 3/4 view is what is actually required. This section will assume a specialized role in some of the subsequent methodologies discussed.

[Style]: This component is obligatory and serves to establish the overarching stylistic tone for the character. In the case study under consideration, the authors have elected to populate this field with "Futuristic, Cyberpunk."

[Character Features and Clothing]: This section will be populated with the principal characteristics of the character as generated by ChatGPT. However, one also has the option to maximize the utilization of stochasticity by allowing Midjourney to autonomously complete this section based on the predetermined style, thereby enabling the free generation of the character.

Under this Template, the finalized Prompt for the character would be:

2D Game Character Design, by Angus McKie, character sheet, Multiple Expression and Pose, t-shaped, 3/4 view + three view, full body, Futuristic, cyberpunk, short hair, wearing tights adorned with gemstones, wearing gloves --ar 1: 1



Figure 1: other character design demonstration by Artist Mattias Adolffson.



Figure 2: the character this article is going to use for animation.

It is noteworthy that, the “T-shaped” prompt is an efficacious construct that the authors discovered in later stages and was not employed during the initial creation of this character.

3. Animation Methods

Upon the completion of character design, the process transitions to the phase of 2D animation production. Within the scope of this article, we shall elucidate the process of generating intricate animations by harnessing a collaborative approach involving multiple software platforms.

3.1. 2D Animation by Human Skeleton Method

Human Skeleton is predominantly utilized for skeletal binding in 3D characters but is infrequently acknowledged for its applicability in producing exceptional animations for 2D characters as well. Currently, software solutions in this domain include Blender’s 2D Animation Tool, the Unity Engine 2D Animation Package, and Cartoon Animator 5. For the purposes of this paper, the Unity Engine 2D Animation Package will be selected for demonstration.

3.2. Image Preprocessing for Unity Engine 2D Animation Package

As can be discerned from Figure 2, certain areas of the character are notably lacking in detail, such as the hands and the gemstones on the shoulders. Prior to progressing to the subsequent phase, it is imperative to address these deficiencies in detail. The character images employed must be devoid of blurriness and disorganized details.

To this end, the authors will undertake three distinct steps:

Step 1. The authors will utilize the “Custom Zoom” feature within Midjourney to acquire a more detailed and possibilities representation of the character. In this step, it is pivotal to exercise caution in not exceeding two iterations of “custom zoom” on the character. Specifically, upon completing the first “custom zoom,” one should refrain from continually applying a second “custom zoom” on the image produced by the first. Exceeding this limit of two iterations compromises Midjourney’s ability to maintain control over the character and significantly amplifies the likelihood of textual artifacts appearing.

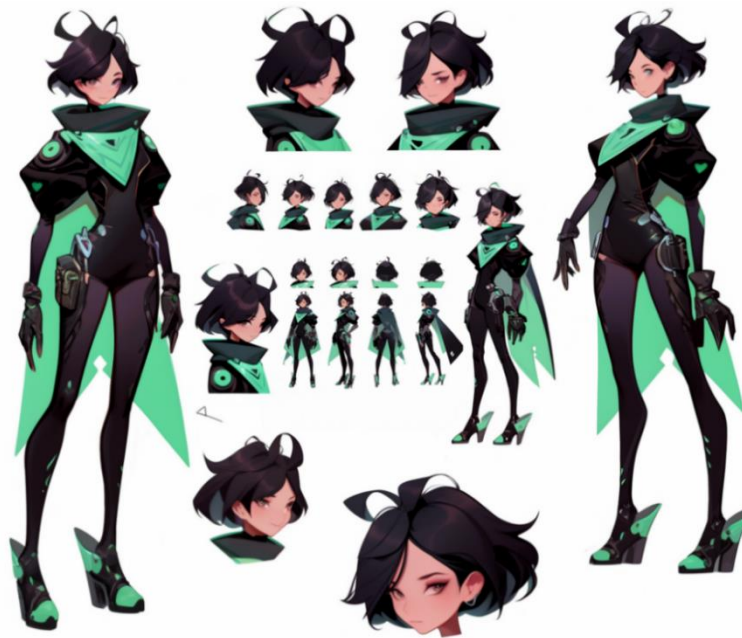


Figure 3: Utilizing the “custom zoom” feature yields a “T-shaped” pose suitable for application in a 2D Platformer Game.

Step 2. The authors will leverage the Vary (Region) feature within Midjourney for the purpose of redrawing or substituting specific details of the character. In this phase, it is not imperative to execute the step on a blank canvas. For example, the location of the portrait in Figure 3 can be redrawn to represent the character’s hand, as illustrated in Figure 4. Alternatively, more detailed images of the head can be obtained, as illustrated in Figure 5.

Example prompt for Figure 4: 2D Game Character Design, by Angus McKie, Character hands, intricate detail hands, Futuristic, cyberpunk, wearing gloves.



Figure 4: Revise the character hand.



Figure 5: Redrawing of the character’s facial details by using Step 2 method.

It is imperative to note that people must not rigidly adhere to a single outcome in the aforementioned two steps. Rather, the generative and stochastic nature of the process should be leveraged to yield a broader array of choices. The preceding two steps can be conceptualized as generating the original assets intended to replace the blurry segments, whereas step 3 serves as the actual implementation of said replacement.

Step 3. By leveraging Photoshop and Photoshop AI, the meticulous processing of character details is achieved. This specific processing entails the utilization of a clipping tool to segment each constituent of the character's body, with select segments necessitating substitution with content generated in Step 1 and Step 2. Subsequently, these segmented constituents are amalgamated, as illustrated in Figure 6. It is important to note that, given the context of a 2D Platformer Game, it is requisite to adjust the character's legs to be positioned at a uniform elevation. Upon completion of the detailed character processing, the author will segregate the character's head, body, hands, and feet within Photoshop, situating each in distinct layers. Following the exportation of the file in .PSD format, the subsequent procedure involves transitioning to the Unity Engine for animation development. As depicted in Figure 7, the mandatory layer order that must be adhered to for.psb format files imported into Unity.

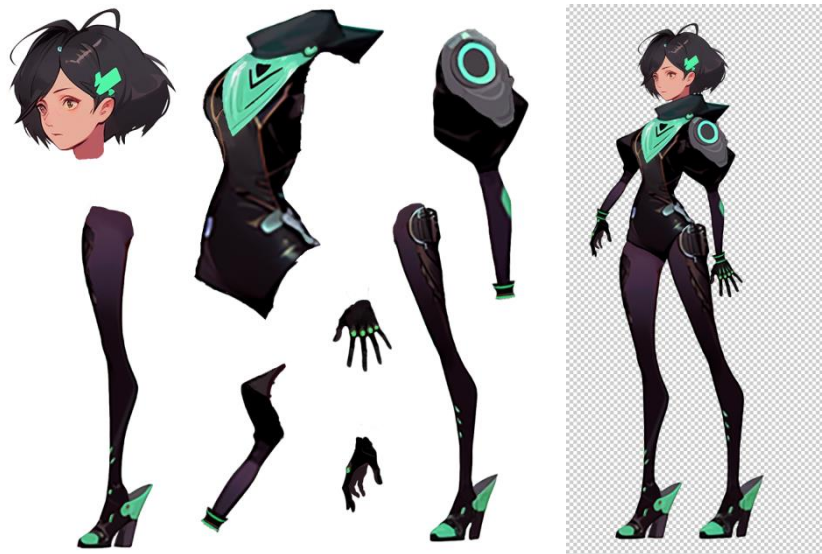


Figure 6: This constitutes the final result of the character design process.

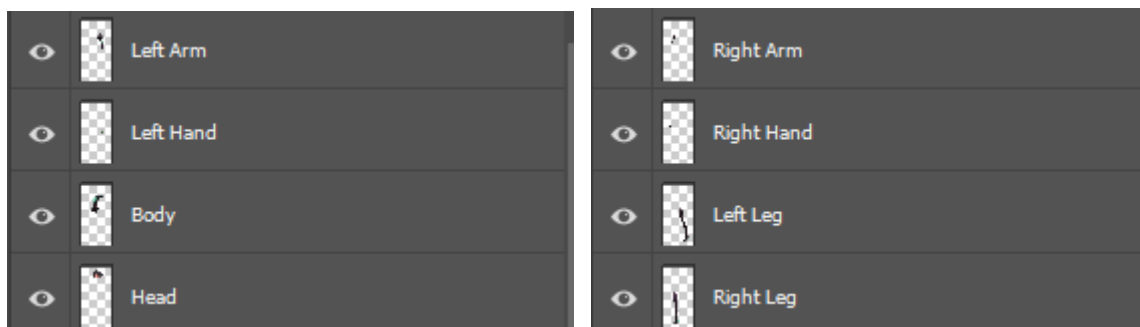


Figure 7: The character is deconstructed into its constituent layers, which are subsequently organized in hierarchical sequence.

3.3. Training in Stable Diffusion LoRA and Hand Depth Techniques for Advanced 2D Animation (Optional)

The inclusion of an “Optional” designation in section 4.1.2 is predicated on the understanding that if all animation requirements of the user can be fully met in section 4.1.1, this step may be circumvented. Here, “fully met” refers to the complete reliance on Midjourney for generating all variations in hand movements, facial expressions, and other details during animation production. The author aspires to acquire multiple variations of hand movements, such as a clenched fist during running, which Midjourney could not adequately address. Consequently, the authors opt to incorporate section 4.1.2 to attain such movements. Within 4.1.2, the initial step involves using LoRa training within Stable Diffusion to obtain a hand that is thematically consistent with the character, ensuring sufficient control and stylistic similarity over the generated hand. This is followed by employing Depth ControlNet within Stable Diffusion to ensure the generation of the requisite hand movements.

3.3.1. Image Processing Prior to LoRa Training

It is requisite to employ the methodology delineated in Step 2 of section 4.1.1 to acquire a sufficient variety of hand movements. These generated hand representations must be relatively clear, devoid of errors (such as missing fingers), and exhibit a degree of generalizability to account for a range of hand movements. The image requirements for such an endeavor are estimated to range between 15 to 25 images. Extract all generated hand movement images within Photoshop, ensuring that the background of each image consists of a singular, unblemished color. Modify the resolution of the images to 300 DPI. Adjust the dimensions of the images to conform to one of the three specifications: 512x512, 512x768, or 768x512, contingent upon the specific content of the image in question, as shown in Figure 8. Place all the processed images into a designated folder intended for training purposes.



Figure 8: Utilize hand movements generated via Midjourney.

3.3.2. Assigning Tags to Images

This step necessitates the tagging of each individual hand image. WD 1.4 Tagger, an AI-powered tagging program, can be employed to streamline this process. The parameter setting as shown in Figure 9. Upon the completion of Tagger’s operation, one merely needs to manually augment or reduce the tag content according to specific requirements. Figure 9 illustrates the parameters of

WD1.4 Tagger. The ‘path’ is designated as the storage location for the images. Within ‘Additional_tags,’ the term ‘Character Hand’ must be included as the triggering keyword for LoRa.

Tagger

path	<input type="text"/>
threshold	<input type="range"/>
additional_tags	<input type="text"/>
interrogator_model	wd14-convnextv2-v2
replace_underscore	<input checked="" type="checkbox"/>
escape_tag	<input checked="" type="checkbox"/>
batch_output_action_on_conflict	copy

Figure 9: Configuration of Tagger Parameters.

3.3.3. Setting Parameters for LoRa Training

Upon the completion of the tagging process, the training of the LoRa model can commence. The estimated range of training steps is between 1,500 and 2,500. In the case study, approximately 15 to 25 images of hands are required, for which the author has prepared 17. Each image undergoes 12 training iterations, resulting in a maximum training epoch of 10. Thus, the total number of training steps is calculated as $17 \times 12 \times 10 = 2040$. Given that the learning rate scheduler employs cosine with restarts, the Unet Learning Rate is set at $1e-4$, and the Text Encoder Learning Rate is set at $1e-5$. The Network_dim parameter is configured at 32, and the Network_alpha parameter is set at 16. In Figure 10 below, The range from 2 to 10 refers to a total of 10 epochs, during which the objective is to maintain the loss value below 0.08. Owing to the high-quality image preprocessing, the initial value already falls below 0.08. Nevertheless, we can still observe a significant drop from 0.054 to 0.048 [10]. One can anticipate favorable outcomes during epochs 8, 9, and 10.



Figure 10: Based on the data feedback from the chart.

3.3.4. Hand Depth Maps

Within the Civitai platform, locate the package titled ‘900 Hands Library for Depth Library [11].’ These depth maps can be employed via ControlNet to confer an extraordinary level of control over hand movements. This package encompasses more than 900 distinct hand movements (Figure 11).



Figure 11: The Depth Map showcases an array of distinct hand movements, which are unattainable through generation via Midjourney.

Step 6: This step is instituted for the evaluation of the LoRa model’s efficacy. Given the ten epochs, each epoch might manifest varying results corresponding to different control levels exerted by LoRa. Displayed below are both the Positive Prompt and the Negative Prompt. Within the Negative Prompt, terms such as “badhand,” “bad-picture-chill,” “ng_deepnegative,” and “EasyNegative” are all embeddings:

Positive Prompt: character hand, right hand, transparent background, gloves, black gloves, solo, objectification

Negative Prompt: (((badhandv4))), NSFW, (worst quality:2), (low quality:2), (normal quality:2), lowres, ((monochrome)), ((grayscale)), bad-picture-chill-75v, ng_deepnegative_v1_75t, EasyNegative, (((character)))

In Figure 12, is a quick test for the LoRa models by using an x/y/z plot. The x-axis represents the epoch values, ranging from 1 to 10 and None (As indicated in the notation above the figure), while the y-axis represents the control intensity, spanning from 0.8 to 1.0. In Figure 13, is an enlarged image corresponding to an epoch of 10 with a control intensity of 1.0. Despite the presence of imperfections, they can be swiftly rectified using ImageToImage Inpainting. As this is a test image, detailed elaboration is not provided herein.



Figure 12: From the visual representation, it is evident that epochs 8, 9, and 10, with a control intensity of 1.0, all yield favorable outcomes.



Figure 13: Presented here is an enlarged image corresponding to an epoch of 10 with a control intensity of 1.0.

Step 7: This phase is dedicated to procuring the requisite depth map, essentially entailing the design of character movements. Currently, the authors necessitate a raised-hand movement for the “idle” animation and a clenched-fist movement for both running and jumping actions. At this juncture, it becomes essential to integrate the hand movements into the character within Photoshop, aligning each hand with its corresponding movement (Figure 14).

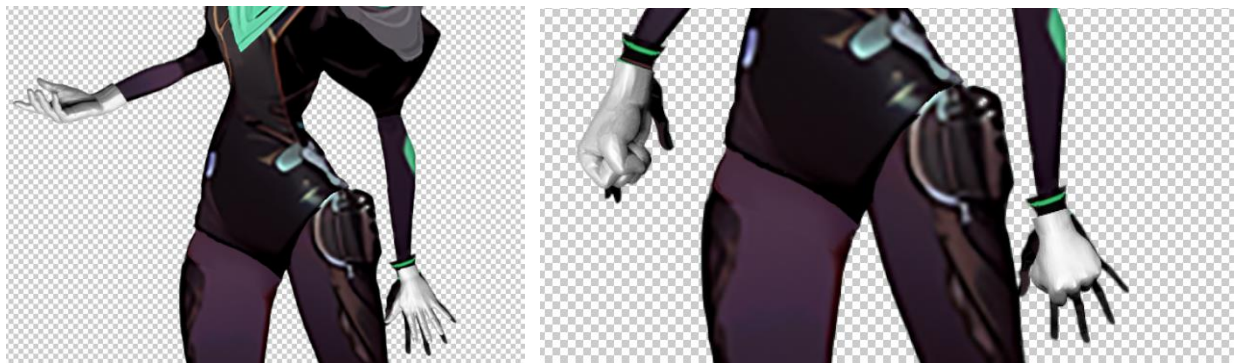


Figure 14: From the Illustrations, we can ascertain the necessary depth maps, specifically denoted by the white images attached to the character in the two presented figures.

Step 8: In this section, we will revisit Step 3 and Step 4 from section 4.1.1 and make the necessary preparations for transitioning into Unity. Diverging from Section 4.1.1, it is required to simultaneously attach multiple distinct hand movements to a single arm, as illustrated in Figure 15.

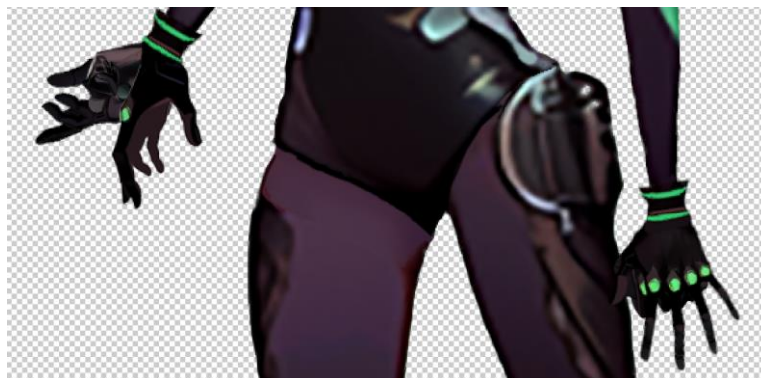


Figure 15: Diverging from previous procedures, we will superimpose all the required hand movements together.

3.4. Unity Engine 2D Animation Package Workflow

Upon entering Unity, it is requisite to download and install the 2D Animation, 2D IK, and 2D PSD Importer packages from the Package Manager. Fortuitously, Unity has now integrated an option that encompasses all 2D packages for user convenience. Additionally, 2D IK has been subsumed within the 2D Animation Package [12].

Import the .PSB file into Unity and open the Sprite Editor. Prior to advancing to the subsequent phase, it is imperative to ascertain that all components are duly prepared, encompassing aspects such as the hierarchical order of layers and the boundaries delineating the display range of the sprite. As illustrated in Figure 16. Following this, one should navigate from the Sprite Editor to the Skinning Editor in preparation for animation processes.

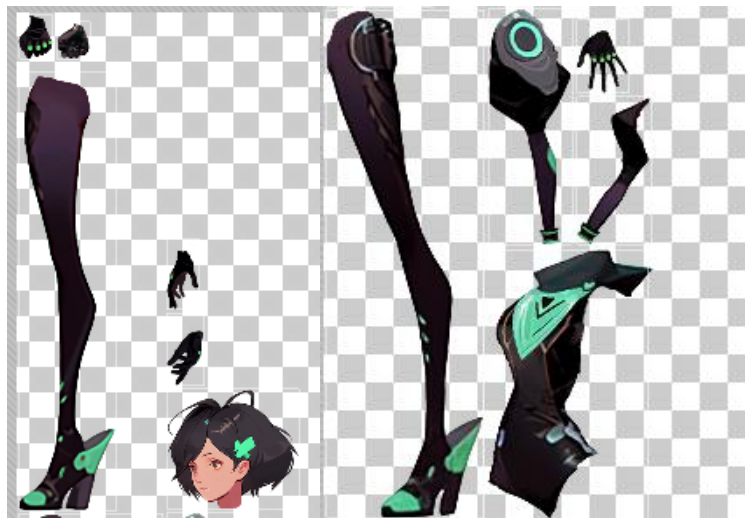


Figure 16: Images of all body parts show in Sprite Editor.

Before commencing, a reminder is pertinent. For those unfamiliar with Unity, inquiries can be directed towards Unity Muse. Unity Muse is an AI language model integrated within the Unity Engine. When one is uncertain about how to achieve a desired effect, Unity Muse can offer valuable solutions.

3.4.1. Set up Sprite Library

Given the presence of animations transitioning through various hand states of the character, it is imperative to exert control over the various layers of the character's body within the sprite library. In the Unity Project window, one is to right-click and opt for Create > 2D > Sprite Library Asset. Upon opening the Sprite Library Asset, set hand sprites (fist and palm) as distinct entries. This maneuver facilitates the swapping between hand states amidst the animation process (Figure 17).

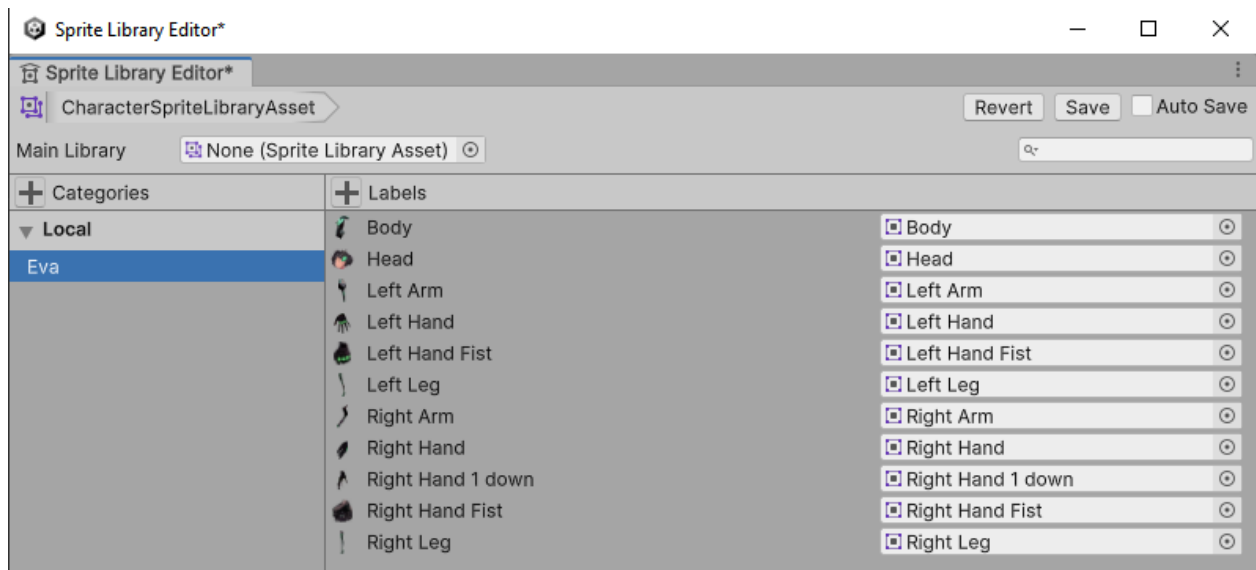


Figure 17: To create a new category, utilizing the name of our character, Eva, and subsequently import the .psb file into the labels.

3.4.2. Set up Character GameObject Before Binding Skeleton

Drag the .psb file into the Unity Hierarchy. This action will generate a GameObject containing child GameObjects for all the body part sprites. Subsequently, rename the GameObject to “Character” and append the Sprite Renderer and Sprite Library components to this GameObject (Figure 18).

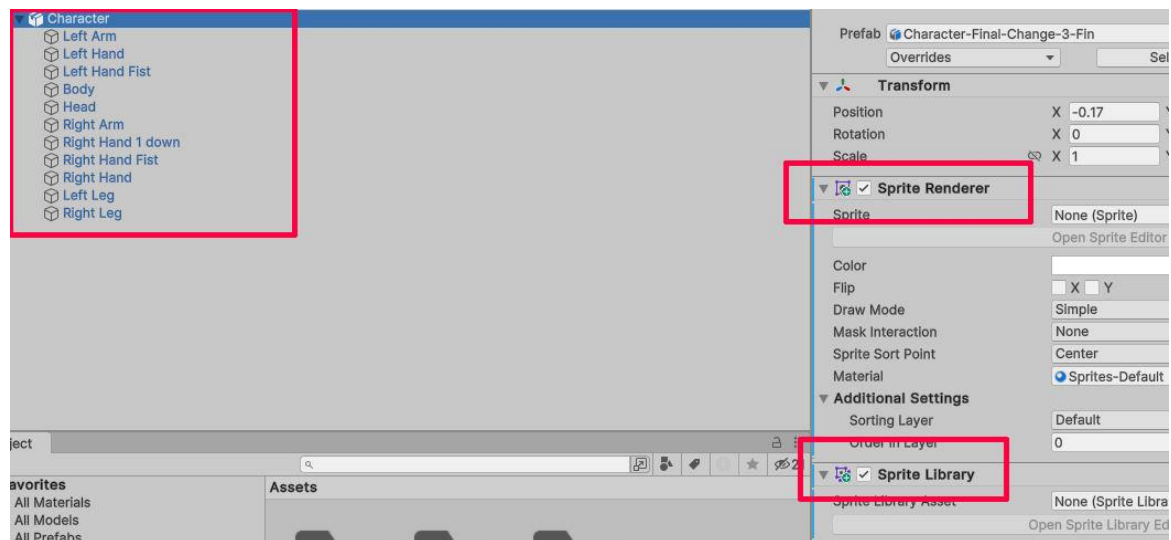


Figure 18: Set up screenshot in the Unity interface.

3.4.3. Binding Skeleton for the Character

In this section, the process will encompass three steps. The first step involves creating bones for the character, aiming to reconstruct the main skeletal structure of the human body. Once set up, as depicted in the first image of Figure 20, the principal locations and modes of human movement can be discerned through the bones. The second step is skinning, a method of controlling the pixels in the character drawing. For example, when moving the hand, this ensures that no finger detaches from the palm, as demonstrated in the second image of Figure 20 upon completion of the settings. The third

step pertains to weight, indicating the scope and magnitude controlled by the bones. For instance, if the strength and range controlled by the bones are excessive, moving the character's head would lead to the entire body following suit, as illustrated in the third image of Figure 20 upon completion of the settings (Figure 19).

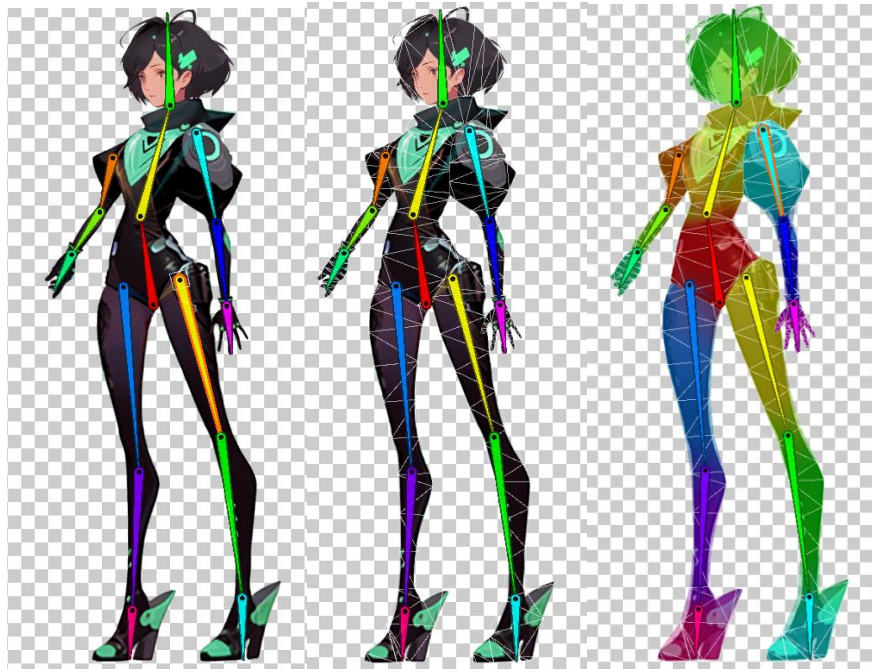


Figure 19: The first image illustrates the configuration of bones; the second image demonstrates the skin settings; and the third image displays the weight settings.

3.4.4. Animation for Idle

Firstly, it is necessary to create an Animation Controller for the character and attach it to the character GameObject's Animator component. Then, select the character game object and create an animation clip. Set keyframes to create animation by rotating bones and activate or deactivate the hand styles.

As illustrated in Figure 20 and Figure 21 below, the six images below represent six keyframes, each spaced by 10 frames. Upon completing the animations for these six keyframes, after 20 frames, it will transition from image 6 back to image 1, indicating the hand being raised and then returning to its original position. That is, a total of 11 keyframes, 100 frames, with the entire animation spanning approximately 4 seconds. It is noteworthy that, at image 3, the palm of the character's hand turns from facing downwards to facing upwards.

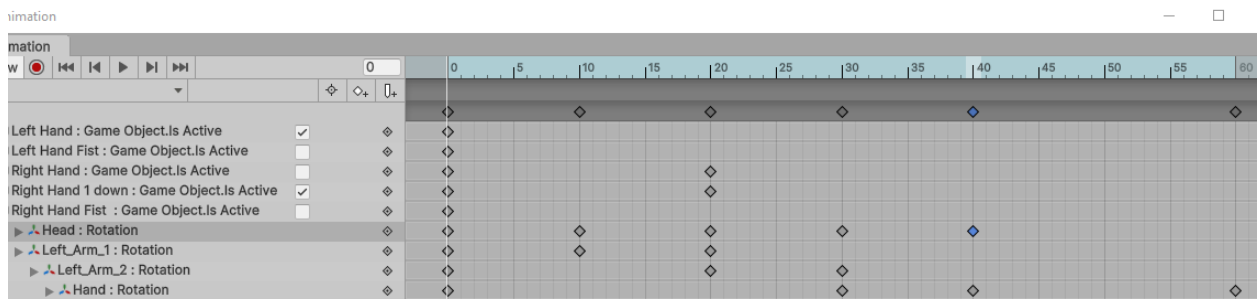


Figure 20: This section of the Animation Editor displays all the keyframes prior to the commencement of the return to the original position.

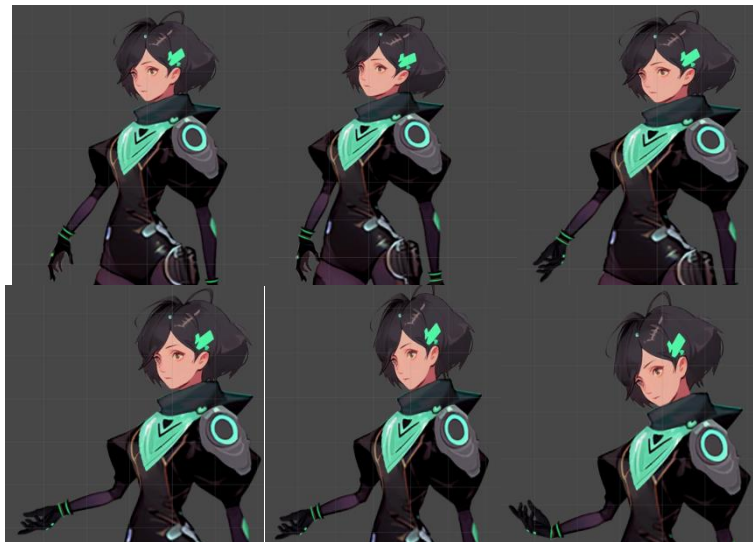


Figure 21: Since the lower half of the body does not move, we only capture the movement of the upper half.

4. Conclusion

In this paper, the author has successfully created an animation sequence without engaging in any manual drawing by utilizing a combination of various software tools. The paper demonstrates how ChatGPT and Midjourney can be employed for character design, and how Midjourney, Stable Diffusion, and Photoshop can be used for the preparatory work of character animation, culminating in the creation of animations using the skeletal animation technique in Unity. Through the exploration and experimentation of this case study, it can be inferred that the amalgamation of AI with various software tools has the potential to entirely supplant traditional artists and the customary procedures of character design. Beyond this, AI also demonstrates the capacity to replace conventional methods of creating animations, such as the traditional animation method – the pose-to-pose technique.

However, the adoption of this integrated workflow still presents certain barriers. For instance, during the process of character skeleton binding in Unity, the author, not being a professional in skeletal animation production, faced appreciable challenges. This implies that this workflow still necessitates the incorporation of specialized knowledge for effective execution.

Hence, there still exists potential for further simplification within this process. A plausible approach could be utilizing OpenPose from Stable Diffusion to capture various skeletal samples during character movement and subsequently employing video and Ebsynth to generate animations. In such scenarios, the stages of skeleton binding and animation creation in Unity could be circumvented.

This exploration of streamlining and integrating diverse technologies will continue to be the focal point of the author's ongoing research, aiming to unveil the expansive possibilities and capabilities of generative AI in the domain of 2D game animation.

References

- [1] Asraf, & Idrus, S. Z. S. (2020). *Hybrid Animation: Implementation of Two-Dimensional (2D) Animation*. *Journal of Physics. Conference Series*, 1529(2), 22093.
- [2] Game, Civitai. https://civitai.com/search/models?sortBy=models_v2&query=game
- [3] Tobias Fischer (2023), *Stable Diffusion Sprite Sheets - New Exodia Automation Method*. <https://www.youtube.com/watch?v=mBA-0mZLeTQ>

- [4] Samuelblue, *Onodofthenorth-SD_PixelArt_SpriteSheet_Generator*. https://huggingface.co/spaces/Samuelblue/Onodofthenorth-SD_PixelArt_SpriteSheet_Generator
- [5] Vranica. (2023). Advertising (A Special Report) --- AI Has Madison Avenue Excited -- and Worried: Using ChatGPT, Midjourney, DALL-E 2 and other artificial-intelligence tools to create advertising campaigns can save time and money. But it also introduces new risks for agencies and clients. *The Wall Street Journal*. Eastern Edition.
- [6] Belmonte, Millán, E., Ruiz-Montiel, M., Badillo, R., Boned, J., Mandow, L., & Pérez-de-la-Cruz, J.-L. (2014). Randomness and control in design processes: An empirical study with architecture students. *Design Studies*, 35(4), 392–411. <https://doi.org/10.1016/j.destud.2014.01.002>
- [7] Liao. (2023). A.I. May Help Design Your Favorite Video Game Character. *New York Times* (Online).
- [8] Baker. (2023). ChatGPT™ / Pam Baker. John Wiley & Sons, Inc.
- [9] Ahn. (2023). On That Toy-Being of Generative Art Toys. *M/C Journal*, 26(2). <https://doi.org/10.5204/mcj.2947>
- [10] Shilpa, Kumar, P. R., & Jha, R. K. (2023). LoRa DL: a deep learning model for enhancing the data transmission over LoRa using autoencoder. *The Journal of Supercomputing*, 79(15), 17079–17097.
- [11] wywywywy (2023.9), *sd-webui-depth-lib*. <https://civitai.com/models/67174/900-hands-library-for-depth-library-or-controlnet>, <https://github.com/wywywywy/sd-webui-depth-lib>
- [12] Bonzon, T. (2023, February 14). Rig a 2D animated character in Unity. *GameDev Academy*. <https://gamedevacademy.org/rig-a-2d-animated-character-in-unity/>