Using Artificial Intelligence for Generating Concept Artwork for Video Games: An Exploration of Stable Diffusion Through ComfyUI

Patrick Lewis

This assessment will explore the use of Artificial Intelligence (AI) in practice through its application to the pre-production of a video game example. The video game example is a strategy simulation fantasy game in the spirit of 'King of Dragon Pass' developed by A Sharp (A Sharp., 1999), a game with a wealth of characters and races allowing us to fully explore the ability of Stable Diffusion and ComfyUI to generate a choice of characters. This use-case will allow this assessment to the Text-to-Image form of art generation through the creation of character concept art for the playable and non-playable characters.

The combination of these processes builds into a condensed "Game Art Pipeline," with the pre-production components described as the following: "Idea Generation," "Visual Style Development," "Sketch," "Colour Key", "Painting", "Final Concept Art", "3D Model Planning" (Shahbazi, N., 2023). Guided by this pipeline we can determine the effectiveness of the AI in practice, its ease of use, as well as its accessibility to an audience.

This is a key area to test and apply AI applications to due to the length and expense involved in the pre-production of video games, as well as other areas such as Film and Television. The process can last "few weeks to several months or even years" (Clayton, P., 2024), therefore shortening the time required for mundane tasks such as generating as much concept art as possible regardless of how much will be used, and for testing how characters look with slight variations in weapon, pose, costume will result in a greater value-to-budget as the artists can spend more time on details and the final products. Companies such as N-iX Game and VR Studio (2023) are prime examples of pre-production being outsourced to save time and resources for smaller studios who are focused more on the game development aspects.

In order to use the image generation techniques effectively and efficiently we will first breakdown Stable Diffusion and ComfyUI in a technical overview. This overview will be an exploration into the mechanics of Stable Diffusion, a breakdown of how ComfyUI functions, and a deep-dive into the parameters used for image generation.

Technical Overview

This process will utilize three distinct elements: the **AI image generation method**, a **graphical user interface** within which to use this method and model, and the **parameters** used to generate the images.

Method - Stable Diffusion

The chosen AI image generation method is Stable Diffusion, developed by Stability AI (2023). Stable Diffusion has been chosen as it is an open-sourced model available on GitHub (CompVis, 2022) and offers its latest available models for download onto a machine, whilst other models such as Dall-E 3 (OpenAI, 2024) and Midjourney V6 (Midjourney, 2024) require subscription payments for full access or offer limited free access.

Stable Diffusion stands above the others because of its accessibility. In an article about the pros and cons of this, Vincent (2022) describes this as significant, highlighting "there's a full-fat version of the model that anyone can download and tinker with," allowing the user freedom in creating a workflow suitable to their needs and their style of work.

At its core, Stable Diffusion is a latent diffusion model. Latent diffusion models are related to regression models which "express the relationship between a dependent variable and one or more independent, or regressor, variables (Bartholomew, D.J. et al. 2002), although they operate on a different principal. This is useful when testing how prompts affect the generated images, as these are "probabilistic models that can generate high-quality images by starting with random noise and gradually transforming them into realistic images through a diffusion process" (Aguimar Neto, A, 2023). This diffusion process may highlight the importance of the prompt and its clarity, as the iterative method used implies the model gradually becomes less random in order to match specified prompt.

Chang et al. (2023) describes the pipeline of diffusion models as involving a "forward process and a reverse process to learn a data distribution, as well as a sampling procedure to generate novel data." This pipeline contains a neural 'backbone' which is described as being "realized as a time-conditional UNet" (Rombach, R. et al. 2022). This UNet follows "the typical architecture of a convolutional network" (Ronneberger, O. et al. 2015) which is fundamental in Stable Diffusion's capabilities in image synthesis.

The forward and reverse processes of the diffusion model, supported by the UNet as its neural backbone, are fundamental to the model's ability to generate images from text prompts. The process of learning from noise and reconstructing data allows Stable Diffusion to effectively transform abstract prompts into detailed and realistic images.

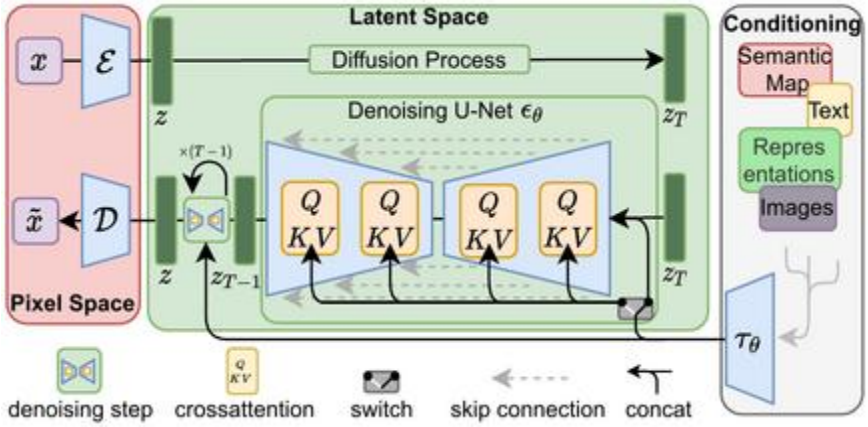

Figure 1: diagram of the process used by Stable Diffusion (CompVis. Figure, 2022)

Graphical User Interface - ComfyUI

Currently the most popular interface for using Stable Diffusion is 'WebUI' created by Automatic1111 (2023). It runs from your personal machine and loads a browser interface where you can tweak parameters, input positive and negative prompts, and determine the size and quality of the generated images. For this assessment we are evaluating 'ComfyUI' (comfyanonymous, 2024), a graphical user interface (GUI) created by an employee of Stable Diffusion, 'comfyanonymous', originally in order to help himself "learn how Stable Diffusion works" (Yubin. 2023).

This GUI based on its in-depth options and readability, allowing the user to understand and follow each step Stable Diffusion takes when generating an image. ComfyUI itself is a "node-based GUI" which allows the user to created workflows "by chaining different blocks (called **nodes**) together" (Stable Diffusion Art, ComfyUI 2023). Figure 2 shows basic workflow which presents the default image generation workflow provided with ComfyUI through connecting nodes. Each node represents customisable parameters, which we will soon assess in-depth.

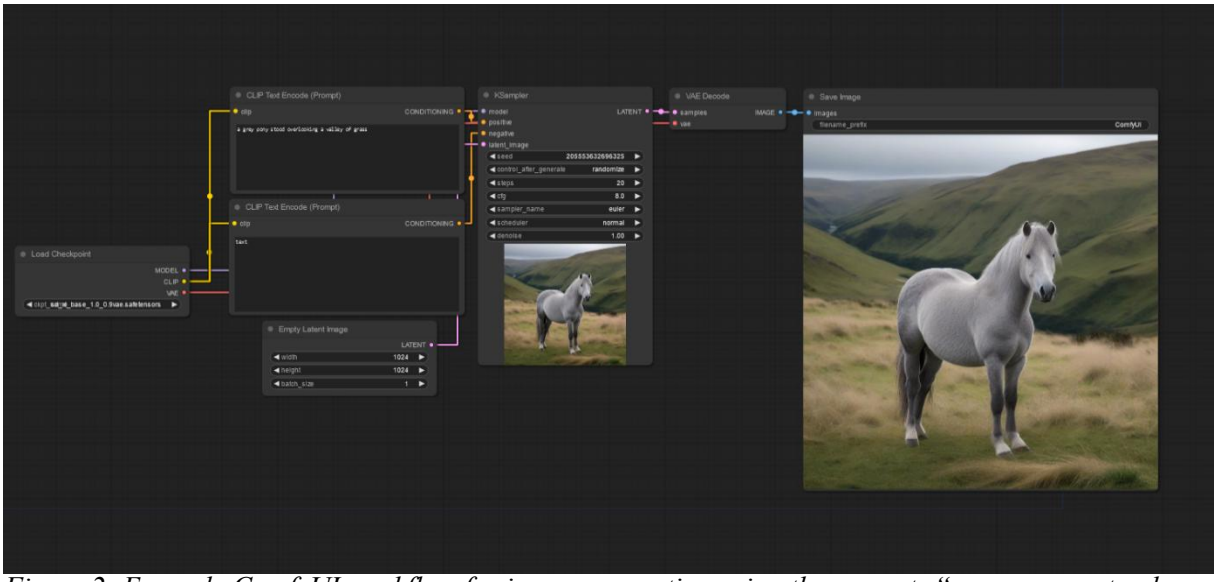

Figure 2: Example ComfyUI workflow for image generation using the prompt: "a grey pony stood overlooking a valley of grass".

ComfyUI is an accessible application which can be downloaded onto your machine from its GitHub repository at https://github.com/comfyanonymous/ComfyUI, which is then run off a .bat file that opens a browser window within which you are free to modify and create workflows. These workflows can be saved onto your machine for future usage. As Andrew comments in his tutorial on how to install ComfyUI, it is a "lightweight" application which "results in lower memory usage" (Installation Guide, 2023) compared to Automatic1111's 'WebUI'. The machine used to run this application uses an Intel Core I7, 16GB RAM, and an NVIDIA GeForce RTX 2060, and for 20 iteration steps, each iteration of the image in figure 2 was generated in an average of 1.30 seconds, therefore totalling at around 30 seconds an image. This average iteration time will depend heavily on the 'checkpoint' used as well as any other extra steps such as custom nodes, which will be explained in due course.

Two vital additions enhance the usability of ComfyUI: the ComfyUI 'Manager' (ltdrdata, 2024), and the vast options of custom nodes available to either download from GitHub, or through the 'Manager' menu. The 'Manager' allows for the browsing and installation of custom nodes as shown in figure 3, as well as the ability to update these nodes and ComfyUI itself, negating the requirement to keep track of every update from a wealth of custom nodes, model checkpoints and VAE checkpoints.

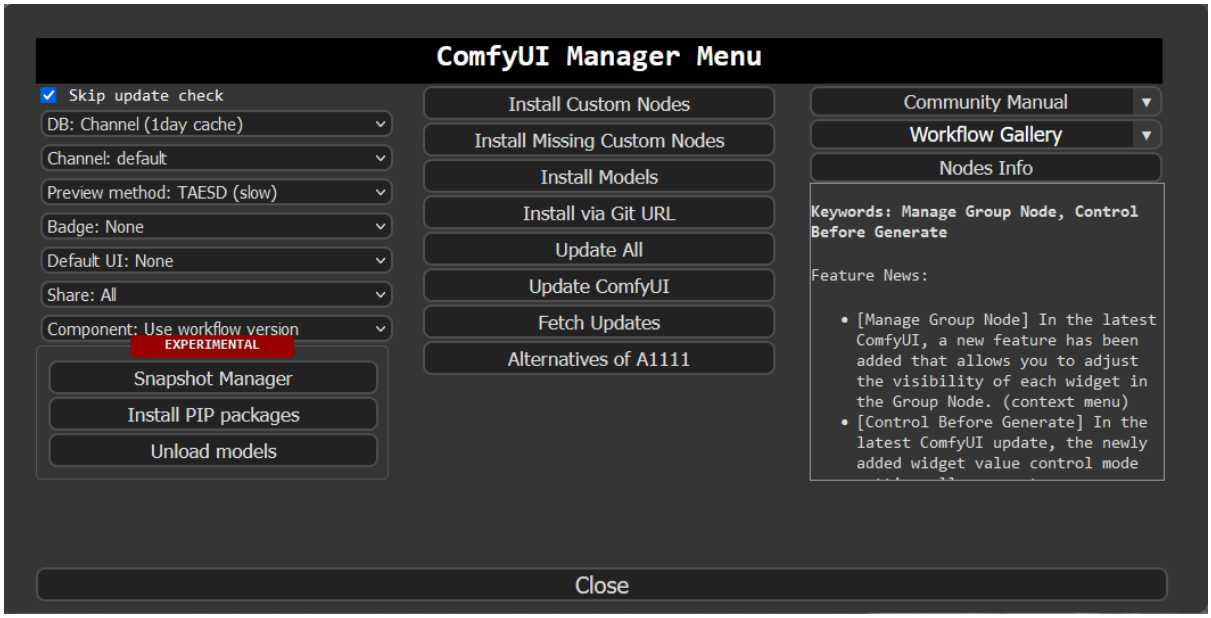

Figure 3: ComfyUI 'Manager' Menu

Custom nodes allow much deeper control of parameters such as with the addition of ControlNets, LoRas, IPAdapters and nodes modified for better-efficiency, with examples of the above shown in figure 4. For this assessment, only the 'Efficiency Nodes for ComfyUI Version 2.0+' is used.

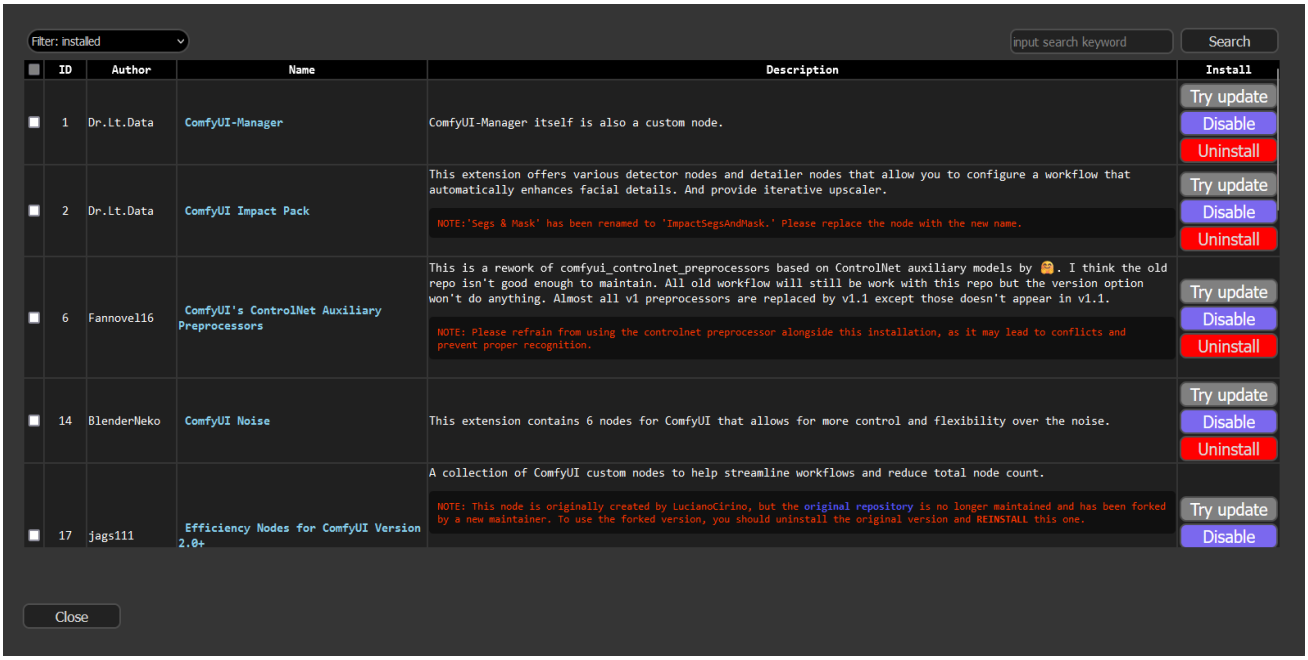

Figure 4: View of installed Custom Nodes via the 'Install Custom Nodes' option in figure 3.

Although this level of control and customization suits a user looking to generate images for a specific purpose, the variance in workflows may cause confusion and limit readability for different use cases. As

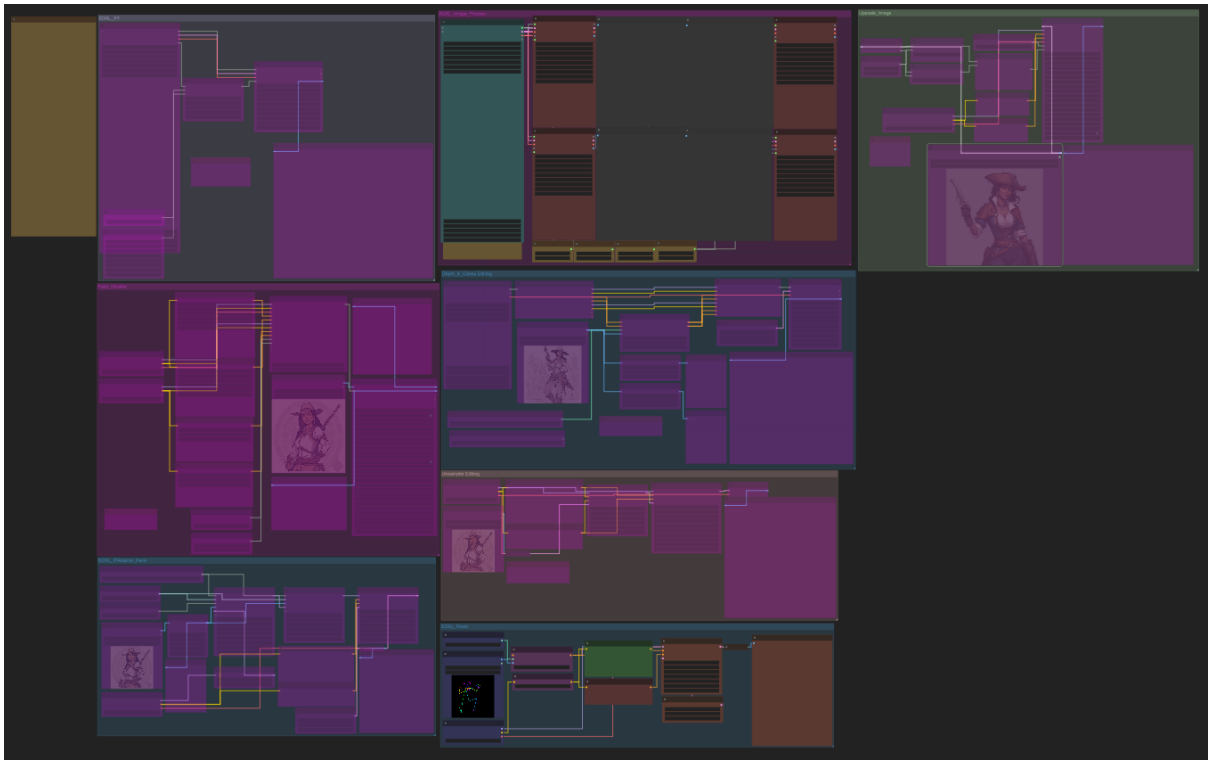

Figure 5: A large custom workflow with nearly 100 nodes and connections.

Andrew mentions, an "inconsistent interface" and "too much detail" can affect the user experience negatively as "average users don't need to know how things are wired under the hood" (Beginner's Guide, 2023), and due to this technology being on the cutting edge the constant updates will inevitably lead to redundant workflows, some of which may be huge and difficult to fix (figure 5).

Therefore, an in-depth knowledge of the parameters is required as these determine the nodes used and number of connections within a workflow, plus the ability to understand how to customise parameters based on a specific task, such as creating concept art.

Parameters

To efficiently use Stable Diffusion for image generation, there are a number of parameters which require understanding. Using the same workflow from figure 2 we can break down the basic nodes.

Load Checkpoint: A load checkpoint node (figure 6) contains the 'MODEL' used for image generation, the 'CLIP', and the 'VAE'. The model has been trained on a dataset, so the resulting generated images will be based

Figure 6: The 'Load Checkpoint' node

purely on that training data. The 'CLIP' (Contrastive Language-Image Pretraining) is a text encoder which converts the text prompt into a format which the format understands. The Variational Auto Encoder (VAE) affects how the image is transformed when sent in and out of the latent space. Most models come with their own VAE 'baked' into them, whilst others require a VAE to be connected via a separate node.

Models

The majority of 'MODELS' are custom-made by users wanting to achieve specific styles in their generations. These models contain UNets. For example, 'Realistic Vison v6' (SG161222, 2023) is trained to generate photoreal images replicating digital and film photography. 'GhostMix v20' (digiplay, 2023) is designed to blend realism and illustration with a lot of references from classic anime and animation. SDXL (Stable Diffusion XL, 2024) has been released officially by Stable Diffusion as a model designed for an 'across the board' generation of images, meaning there is no specific style. To present their differences, figures 7-9 have each been generated using the exact same seed, parameters, positive prompt **-a photo of a modern hero-**, and negative prompt **-text, nudity-**, the only difference being that they each use a different model. We can determine the strengths of each model from these examples: Realistic Vision v6 is good at generating images of real-looking people in a photoreal style, GhostMix v20 generates illustration-like images fit for a book or concept art, and SDXL creates a photoreal image of a blend of modern heroes, the primary seeming to be The Mandalorian. These examples provide a quick look at the difference only change one parameter, the model, can make,

Figure 7: Realistic Vision v6

Figure 9: SDXL 1.0

due to the data these models are trained on. A simple explanation on how a training dataset determines the output can be summed up as "a model won't be able to generate a cat's image if there's never a cat in the training data." (Stable Diffusion Art, 2023), therefore the user must understand the training data a model has been trained on if they want to fully utilise the models in their workflow.

CLIP

'Contrastive Language-Image Pretraining' was developed by OpenAI and released in 2021 (OpenAI., 2021) and comes ingrained into the checkpoint loader in ComfyUI. As Radford explains, the "CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples" (Radford, A., 2021), which now when deployed means the CLIP breaks the natural language used in positive and negative prompts (figure 10) down into a format the model can read. This process means greater control over the generated image as "it embeds images and text to the same latent space, thus allowing us to apply language-guided image manipulations" (Ramesh, A., 2024). To display this control, we can test how an image generation changes based on the CLIP conditioning. We will use four conditions (figure 11): 'Concatenate' two prompts, 'Combine' two prompts, 'Average' two prompts, and choose a value of the 'SetTimestepRange' for each prompt. Figures 10 & 16 show the result of the standard workflow for prompting. The negative prompt remains the same for each of the following generations. Concatenate adds the second prompt onto the end of the first, treating both texts as 'one' and generates one base noise from which the image is generated (figures 12 & 17). Combine generates a base noise from each prompt, then combines this noise into one base noise from which the image is generated (figures 13 & 18). Average uses the strength modifier to determine the weight of each prompt and then averages, generating the base noise from which the image is generated (figures 14 & 19). SetTimestepRange acts in a similar function to averaging, however it gives increased control to the user as they are able to modify the start and end of when each prompt affects the final generated image. Shown in figure 20, the modifier

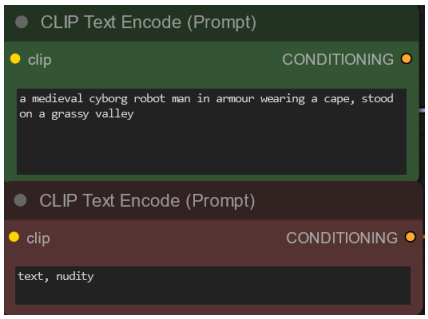

Figure 10: Standard Positive (Green) & Negative (Red) Prompting

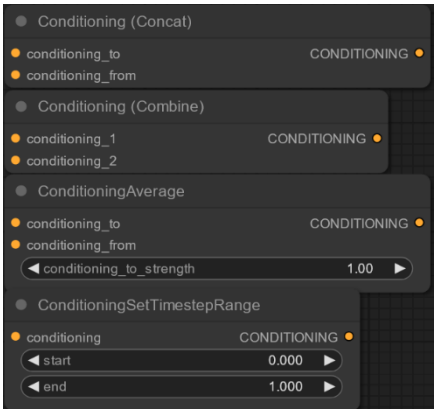

Figure 11: Four CLIP conditioners

'start' refers to when the point which that prompt begins affecting the final image, with '0' being '0%' and '1.00' being 100%, whilst 'end' represents the same but for when the prompt stops affecting the image.

Figures 16-23 show how the format and conditioning of the prompt can both subtly and majorly affect the final image generated, even with just minor tweaks. An interesting observation is how a greater weight towards the "rocky valley" prompt in figures 21 & 22 cause the "cyborg robot man" to become sharper with red eyes and perhaps 'eviler' in appearance, while "grassy valley" generates it as smoother with blue

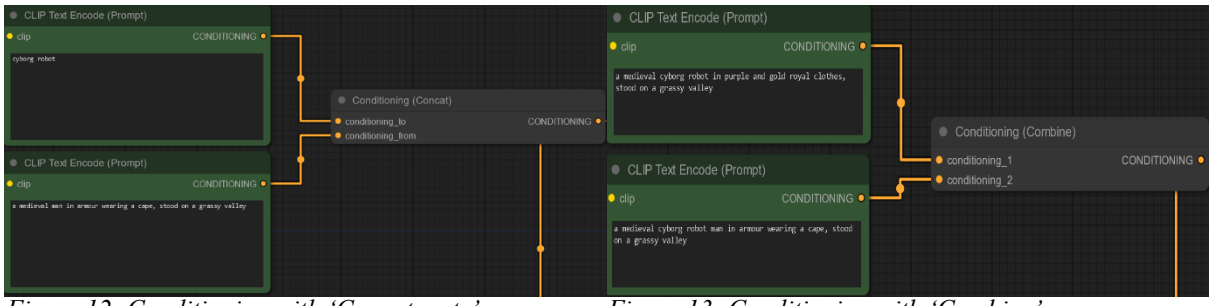

Figure 12: Conditioning with 'Concatenate'

Figure 13: Conditioning with 'Combine'

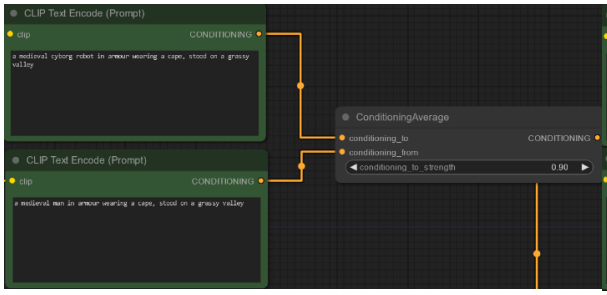

Figure 14: Conditioning with 'Average'

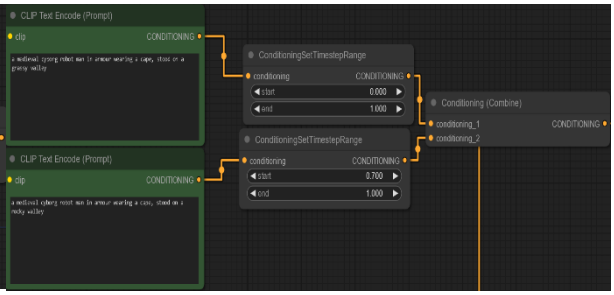

Figure 15: Conditioning with 'SetTimestepRange'

Figure 16: Standard Result

Figure 17: 'Concat' Result

Figure 18: 'Combine' Result

Figure 19: 'Average' Result

Figure 20: SetTimestepRange

Rocky: Start 0.700, end 1.000. Grass: Start 0.000, end 1.000

Figure 21: SetTimestepRange

Rocky: Start 0.000, end 1.000. Grass: Start 0.700, end 1.00

Figure 22: SetTimestepRange

Rocky: Start 0.000, end 1.000. Grass: Start 0.200, end 1.000

Figure 23: SetTimestepRange

Rocky: Start 0.000, end 1.000. Grass: Start 0.000, end 1.000

VAE

The choice of variational autoencoder results in very minor changes in detail for generated images. Their importance lies much more in their necessity for the generation of images. A standard "autoencoder is a special type of neural network that is trained to copy its input to its output" (TensorFlow., 2023), whilst a variational autoencoder "is a type of likelihood-based generative model" (Kingma, D.P. and Welling, M., 2014), and it is this 'likelihood' element which allows a VAE freedom in generating varied images from a single output, perhaps eluding to the idea of a 'creative' neural network. The use of a VAE provides image generation models such as Stable Diffusion a unique edge as this means everything happens within the 'latent' space, where the VAE and the CLIP operate together.

Latent

The latent space contains a 'dense representation' of the image before it is 'decoded' into a pixelated image. Within ComfyUI, the size of the generated image is a parameter set within the 'Empty Latent Image' node (figure 24). An advantage of this process is the memory efficiency, as the generated image is not transformed into pixels until the decoding process and all the high-computation processes such as determining how the image will look pixel by pixel is performed in this 'dense representation.' Utilising a VAE, those major and minor changes are made within the laten space dependent on both the MODEL used and the CLIP conditioning. A final note on the latent is the 'batch_size' modifier in figure 24. Whilst keeping all other parameters static, this determines the number of images generated in this 'batch.' It is an effective way to test a fixed set of parameters with a fixed seed

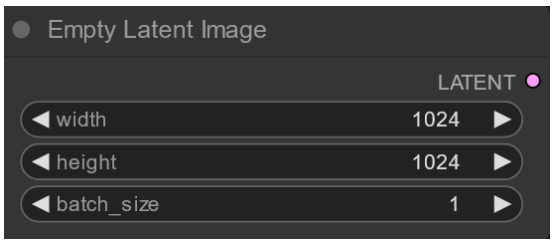

Figure 24: Empty Latent Image node

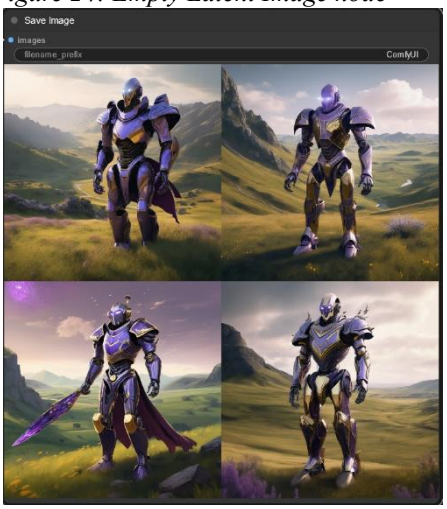

Figure 25: Generated Images with batch_size of '4'

value and evaluate the results, such as in figure 25 with the same parameters as figure 18, only the 'batch_size' is now '4'.

KSampler

The KSampler is the engine through which all nodes will eventually feed their process into for the KSampler to produce the final latent image ready for decoding. Blenderneko's 'ComfyUI Community Manual provides an excellent breakdown of each of these parameters (Blenderneko., 2023), with the most used parameters being the noise_seed, steps, cfg, sampler name, and scheduler. A great process to generate a number of images using different samplers or schedulers in order for the user to determine their favoured sampler/scheduler combination is through the use of an XY Plot, an example of which can be seen in figure 27. This XY Plot shows six different seed images generated with six different samplers with the scheduler 'karras.' This can be a starting point in understanding which combination generates a user's preferred 'look,' however a downside is on a machine without up-to-date hardware, this process can take up to 53 minutes to complete.

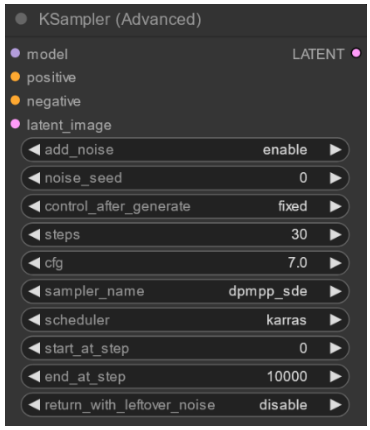

Figure 26: Advanced KSampler node

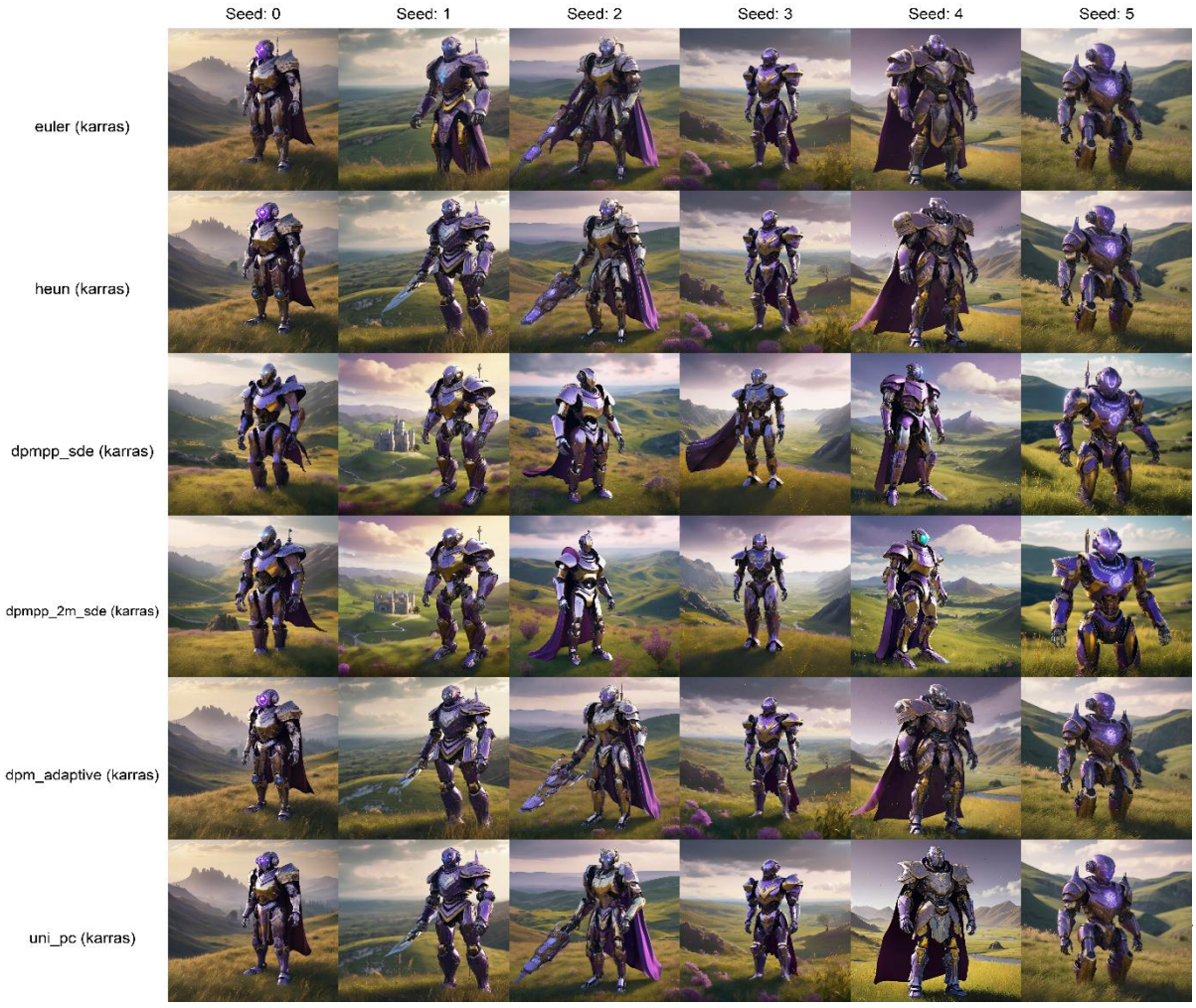

Figure 27: XY Plot of a batch of 6 images generate by 6 different samplers

Refiner

The refiner does exactly what is says on the tin: it refines an image. It is a valid addition developed by Stable Diffusion on "specialized on high-quality, high resolution data" which "improves sample quality for detailed backgrounds and human faces" and has been evaluated with an audience whose "results demonstrate the SDXL with the refinement stage is the highest rated choice" (Podell, D. et al. 2023). The process differs from a regular workflow, as you need to duplicate the original workflow shown in figure 2 and replace the base model with a refiner, shown in figure 26. The refiner KSampler takes the image generated from the inputted steps from the base model's KSampler as its latent image and iterates (x) more times based on the user's choice. Figure 28 is generated with ten steps in the SDXL refiner model (Stable Diffusion XL, 2024) resulting in sharper grass and flowers, more detailed valley cliffs and background rocks, and addition of minor armour details such as carvings on the chest compared to figure 27. This shows that when requiring extra detail, a refiner is a solid addition to the workflow.

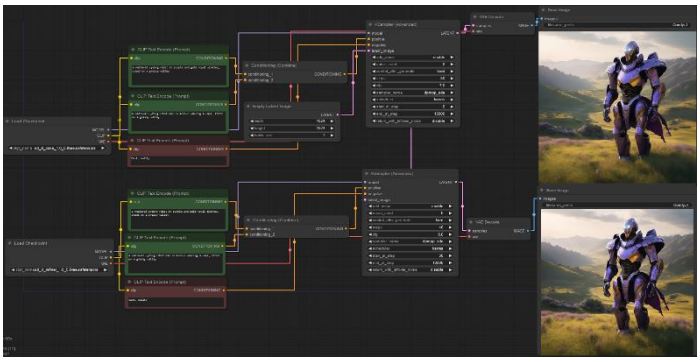

Figure 28: Workflow with the addition of SDXL refiner

Figure 29: Image generated using base SDXL model

Figure 30: Image generated with addition of SDXL Refiner Model

Technical Specifications

Returning to our goal of creating concept art, the model used for this process will be the 'SDXL' Model. This is due to its advantages in "Legible text," "Better human anatomy," "Artistic styles," "Shorter prompts," and "Improved composition" (Stable Diffusion XL, 2024) as well as the overall "Higher image quality" and "Higher native resolution" (Stable Diffusion Art, 2023). As **figure 2** shows, research suggests that overall SDXL performs far better than the 'SD 1.5' and 'SD 2.1' models, with a greater increase when combined with a 'refiner'.

It is noted that the addition of a "refinement model" leads to the "hampering accessibility and sampling speed," (Podell, D. et al. 2023) which should be a consideration when deciding whether the refiner is necessary for this process.

In summary, using Stable Diffusion as our method, with SDXL as our model, and ComfyUI as our GUI, we will utilise both default and custom nodes through the 'Node Manager' in order to efficiently generate text-to-image generations with the goal of creating concept art for characters in our game.

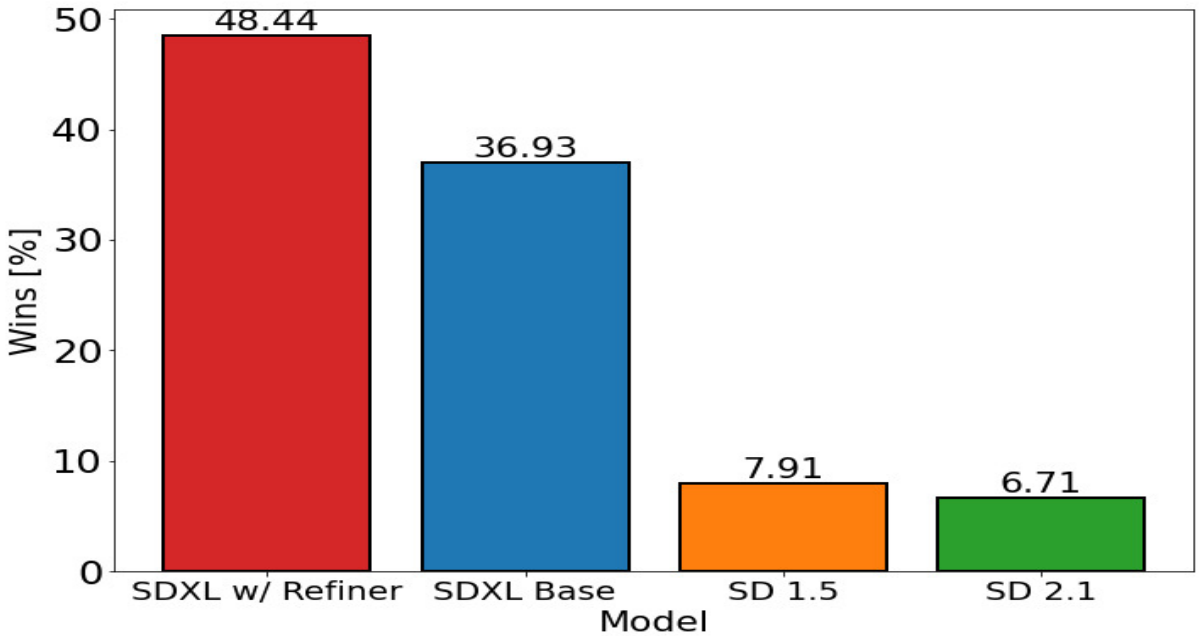

Figure 31: Comparing user preferences between checkpoints (Podell, D. et al. 2023)

Text-To-Image

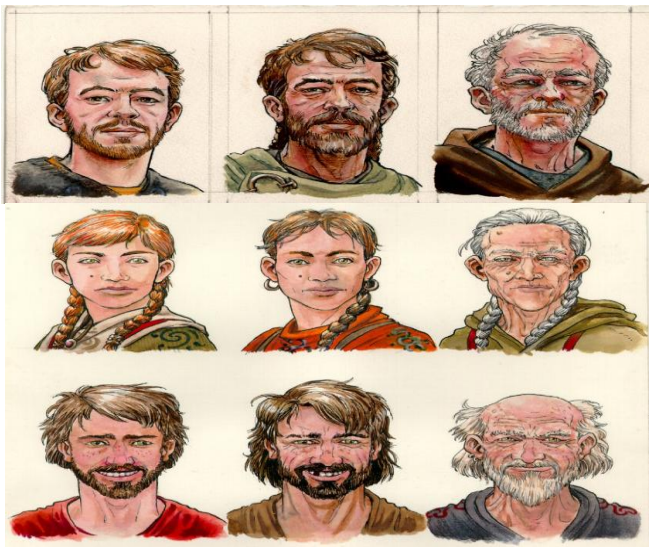

Figure 32: Human character artwork from King of Dragon Pass (Gaudiano, S., 2023)

This process of generating ideas is vital to lay down the foundation of everything we will do from this point on. We can use concept artwork and screenshots as references for the prompts we will generate. The main inspiration is King of Dragon Pass, from which we can garner a dictionary of keywords, terms, and phrases. Rather than upload an image to receive a prompt, which can be done using Dall-E 3, which takes away the human element of idea generation, we will make this dictionary based on a collection of images. This process is organic due to the fact that every individual has their unique view, understanding, and ideas of everything they see. A hundred people will see different things in a single painting, and these unique elements make up what we can determine as their 'style.' Using examples such as figures 31, & 32, we can choose our keywords for prompting our characters.

Figure 32: Event & Beast character artwork from King of Dragon Pass available on the Steam Community Screenshot Page

Prompt Engineering

The layout of the prompts can be set out in the following order, with examples of each:

Style: **Illustration, Video Game Concept Art, Watercolour, Ink, Traditional Fantasy, Sketch**

Subject: **Male/Female, Young, Middle-Aged, Old, Race, Skin-Colour, Time-Period**

Action: **Facing Camera, Sneaking, Standing Proud, Fighting, Pulling Bow, Angry, Conniving**

Shot Type: **Close-Up, Portrait, Full Body Shot**

Characteristics: **Hair-style, Hair Colour, Facial Hair, Scars, Tattoos, Jewellery, Clothing**

The approach which seems best is to start with as small a prompt as possible and gradually add to it. Once the user finds a rough idea of what they like, they can test how different samplers handle that prompt. This can be seen in action on figures 33-40, where the positive prompt of

"Video Game Concept Art, Detailed Sketch
portrait, young medieval woman"

is used, with the line between them, to generate the sketches of a medieval woman. Now we add to the prompt word by word and generate the image with a lower step count of 20 to save time and computation. For our purposes, DMP_Adaptive will be used with the Karras scheduler.

Figure 33-36: Samplers in order from left to right: Euler, Heun, LMS, Uni_PC

Figure 37-40: Samplers in order from left to right: DPM_Adaptive, DPMPP_SDE, DPMPP_2_SDE,

After some tweaking, our result is figure 41 which has gone through 20 iterations of the base SDXL model and 5 iterations of our SDXL refiner. It has been generated using the SetTimestepRange CLIP conditioner, as the watercolour and ink styles heavily influenced the overall image as shown in figure 42. The prompt including the art styles has its 'start' set to 0.050 (figure 43), and this miniscule modifier means the image that would be generated without this style, a more realistic image but still focused on a 'video game concept art', can take hold and be the base on which the art styles can be applied. Keeping this seed provides consistency as we can see in figure 44 where we can generate a male character with only a minute change.

Figure 41: Our generated image (flipped) compared to a character from King of Dragon Pass

Figure 42: Result of both positive CLIP prompt without CLIP conditioning

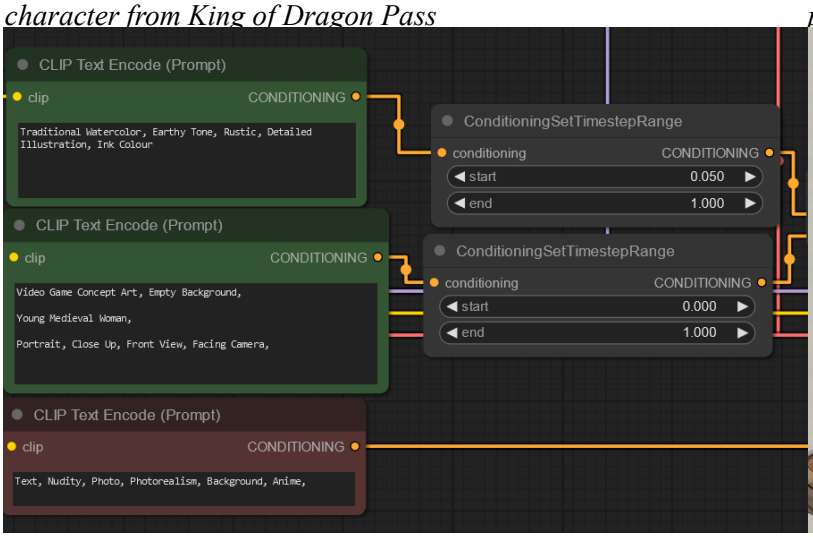

Figure 43: CLIP Prompts with conditioning in workflow to generate figure 41

Figure 44: Generated image with "Man," replacing "Woman" in the prompt

With this setup we can generate the following series of concept art which we can use to evaluate the consistency and efficiency of Stable Diffusion and ComfyUI: Portrait Art for playable characters; Art for in-game events; and art for meeting other races (nicknamed Beastfolk).

Figure 45: Four character portraits generated, with the only change between gender being 'Man' or 'Woman'

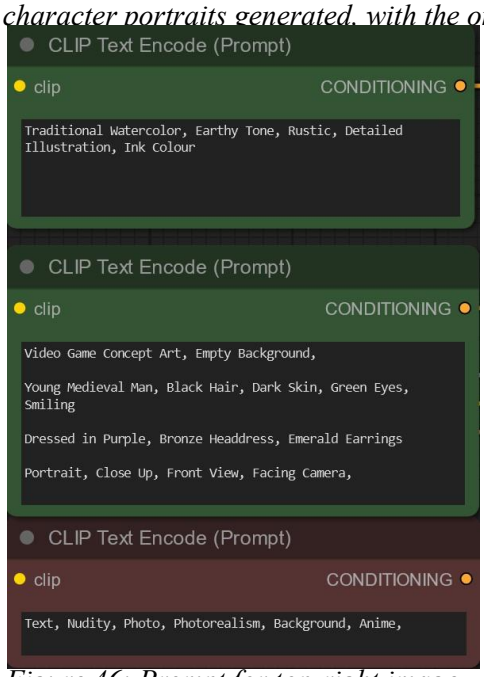

Figure 46: Prompt for top-right image

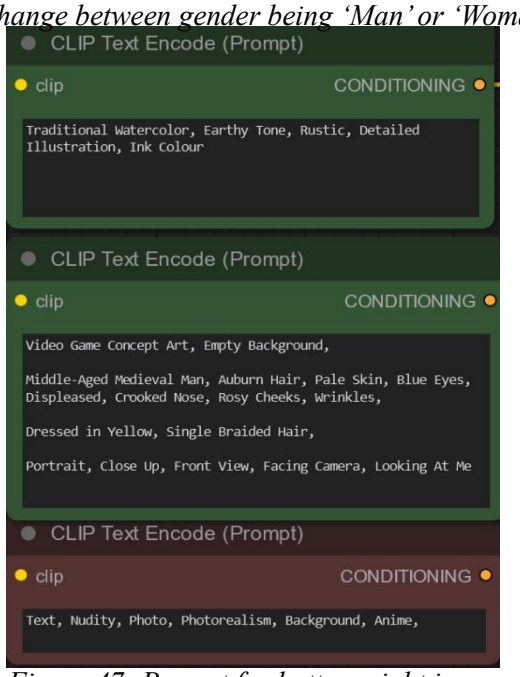

Figure 47: Prompt for bottom-right image

Event 1:

At first a deep rumbling of the earth… Then like thunder the mighty white ox Ulfhir glowing in red runes rampaged through our market! **13** of our tradesmen perished beneath His hooves, whilst a number of visiting merchants sustained injury. Ulfhir vanished into the hills leaving a glowing red trail of dirt and blood. Your priest Yoldor speaks "Perhaps this is a sign our sacrifices to the Bull God Ulf, Lord of the Harvest, did not sate his hunger? If we do not rectify this tragedy, foreign merchants may not return, and our trade will stagnate."

Figure 48: Rampaging ox 'Ulfhir'

- **Bribe the merchants to return** *(Lose Silver)*
- **Conduct a public sacrifice to the Bull God Ulf in the market square** *(Lose Cows)*
- **Send out Weaponthanes to track and slay Ulfhir** *(30/70 Success/Loss)*
- **Do nothing, these things happen...** *(-Trade Income)*

Event 2:

Thieves! Who would have thought the 'royal' descendants of Ancient Dragons could stoop so low as to steal our precious poultry? Cotter Tysan caught this Forgotten in the act of prying open his coop. Undeterred, they demand respect and that no harm be done to them lest we want to feel the wrath of their tribe. *"*SQUAWK* They are not for eating… not by your kind! We can trade you… let us save all… no… half of our kin every year… and we will give *SQUAWK* comfy feather pillows!"*

Figure 49: A chicken thief from the 'Forgotten Tribe'

- **Execution! We will mount his head in our great hall** *(Ecstatic Population)*
- **Send a message, cut off their hand** *(Happy Population)*
- **Set up this chicken trade with their tribe** *(Concerned Population)*
- **Let them go with a warning and keep all your chickens** *(Unhappy Population)*

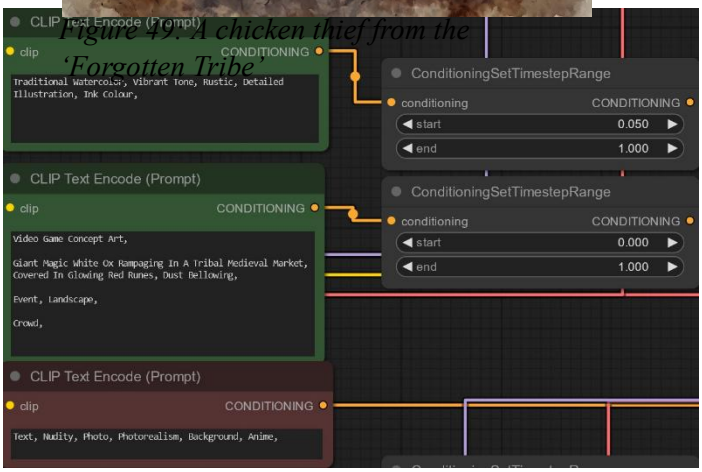

Figure 50: Prompt for figure 48

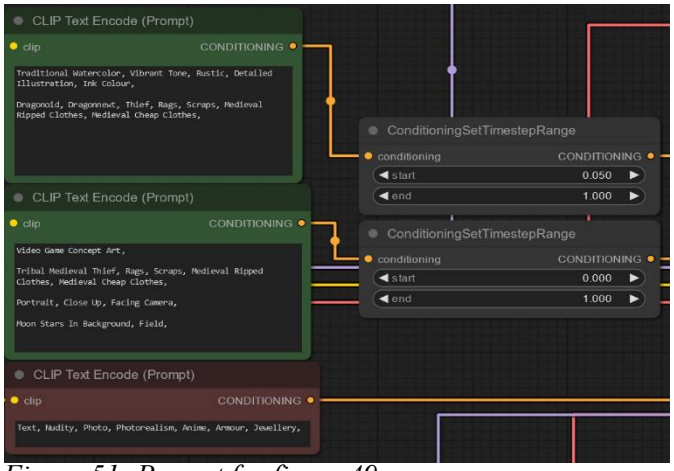

Figure 51: Prompt for figure 49

Beastfolk:

Figure 52 (top row): Three generated variations for the 'Frogfolk'
Figure 53 (top row): Three generated variations of the 'Children'

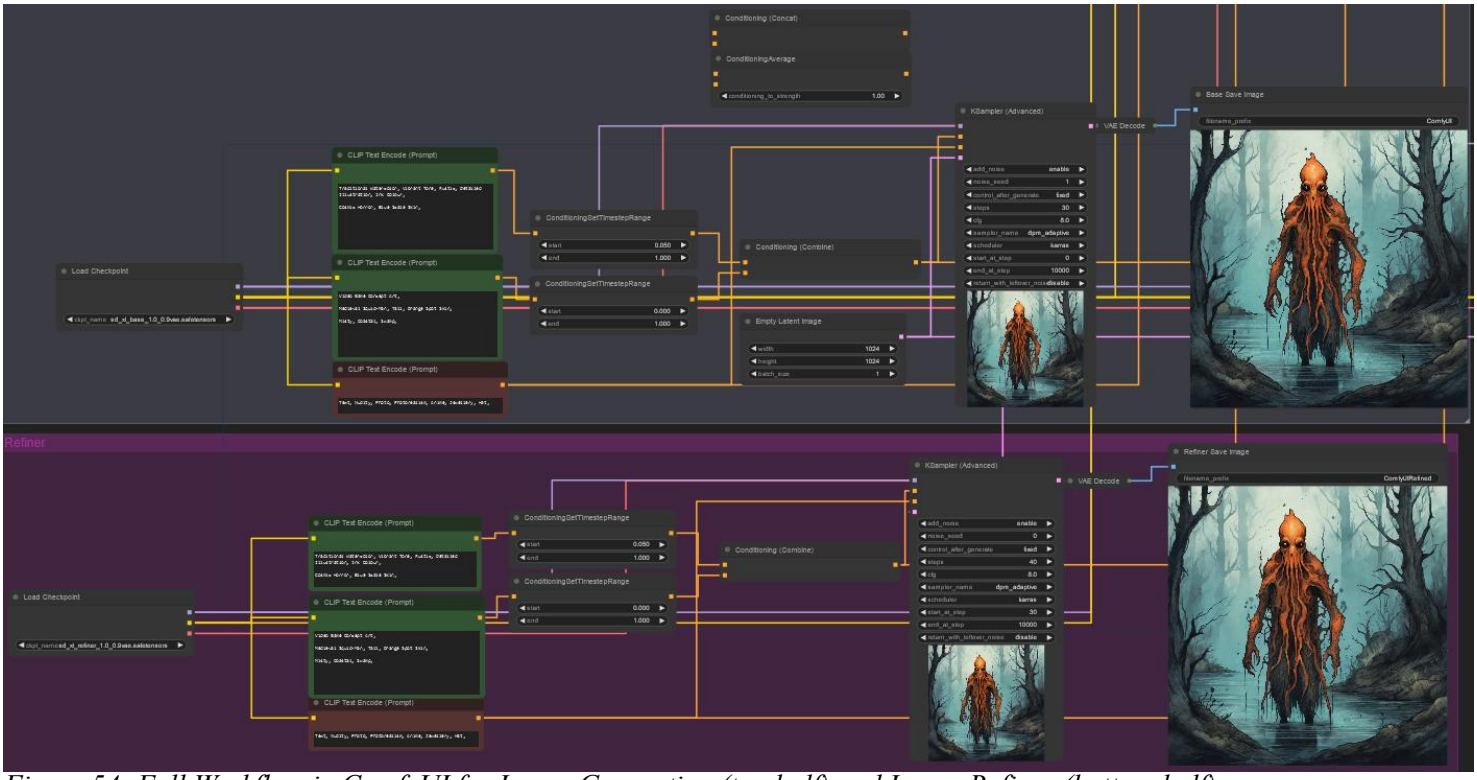

Figure 54: Full Workflow in ComfyUI for Image Generation (top half) and Image Refiner (bottom half)

Evaluation

This assessment has explored the use of Stable Diffusion with ComfyUI in the generation of concept artwork in a similar style to an existing game and genre, using the Text-To-Image capabilities.
The generated images in this process all follow a consistent ink and watercolour art style, fit into an early medieval/tribal setting, with high fantasy elements whilst grounded with familiar, semi-realistic details. Due to this, it is clear this is an effective process to achieve the goal of the assessment. Idea Generation has produced a template of sorts for keywords to use for prompts, which have allowed for consistency across the whole group of generated images.

On efficiency, overall, the results took 2 minutes per image on average using clip conditioning and refiner. On a better machine, this can be cut down dramatically. Due to the edits required, a generation ever 2 minute is not suitable due to the need to often generate over 20 images. For example, figure 48 of Ulfhir was generated 39 times before this image was chosen. This is due to the number of parameters that can be tweaked, particularly the prompt, cfg, and steps. A word can change the entire composition, such as in

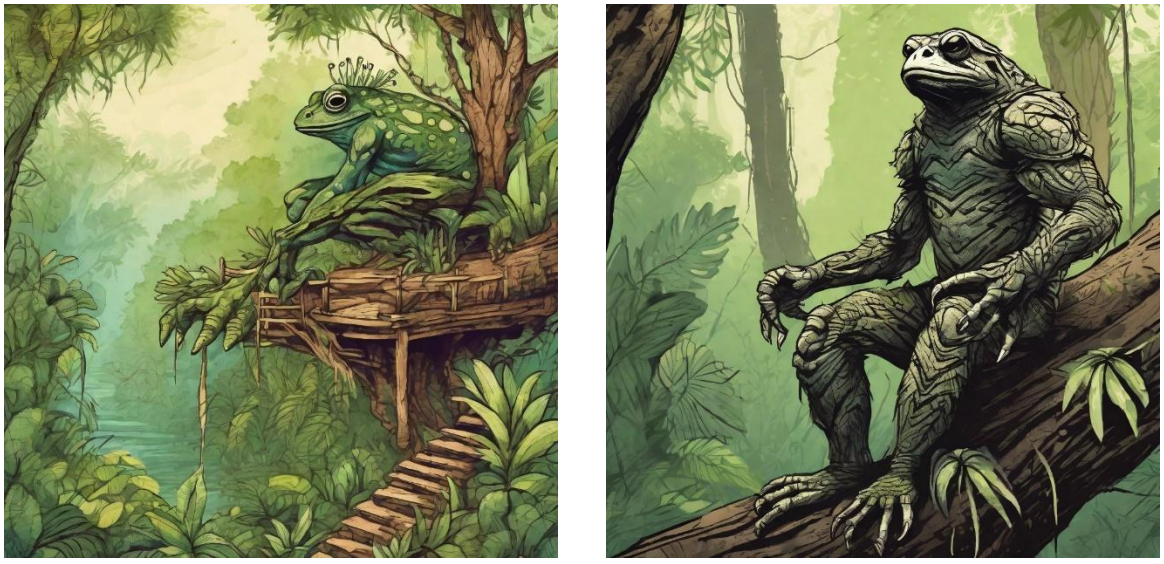

Figure 55: Comparison of two generated image with (left) and without (right) "Shield" in the prompt

figure 55 where the word "Shield," causes the image to shift into a different composition whilst not even generating a shield anywhere in the image. Often a change in the cfg scale can produce interesting differences which deserve exploration, such as in figure 56. Here the scale is changed from 7.00 to 6.00, allowing more freedom for the model to generate visuals that are less strict to the prompt. On the other hand, the final image chosen (figure 54) had a cfg of 8.0, which shows that a stricter value can generate

preferable results. Therefore, on the machine used for this assessment the efficiency would not be suitable for work, however on a more up-to-date machine with a modern graphics card this would be a much quicker process.

Figure 56: Comparison of two generated image where cfg is 7.0 (left) and 6.0 (right)

The workflow is relatively simple once the user gains an understanding of the parameters available. They do not require a wealth of custom nodes. For this workflow used (figure 54) only the ComfyUI Manager is used which allows for a preview of the generated image under the KSampler. The workflow is viewable all on one screen, and for zooming a node can be pulled closer, for example the KSampler can be pulled beside the three CLIP Prompt Inputs, allowing for an easy preview of each change that is made while it is generated. If an image being generated is far away from the result the user is looking for, they can cancel this round of generation, make their changes, and repeat. Therefore, based on the user understanding the parameters, this is an accessible process for Text-To-Image Generation.

Originally this assessment was going to cover Image-To-Image and Image-To-Video, however the scale of understanding parameters alone in the technical overview, as well as the positive results produced by Text-To-Image has mean there is no time or space to fit those experiments in. These two processes would be the next steps on from this evaluation. ComfyUI has the capabilities for Image-To-Image generation, in particular Face-Swapping, using Depth Maps, Canny Edge Maps, Colour Maps all for the composition of images, as well as In-Painting and using Mask Layers. In addition, there are nodes and

workflows for video generation which max out at 3 seconds currently, but their capabilities will improve rapidly in the near future.

In conclusion, Stable Diffusion Models used with ComfyUI offer a fantastic option for image generation which are accessible for anyone due to its open-source nature, ease of use for a user with an understanding of parameters, although a machine will low-mid end specs may struggle with efficiency due to the time taken per image generation. The results created are fit to pass onto an artist to use as a mood board, early concept art, and general further idea generation for a strategy simulation fantasy in the spirit of 'King of Dragon Pass.'

Bibliography

A Sharp. 1999. King of Dragon Pass. [Video game] Available at: http://www.a-sharp.com/kodp/ [Accessed 18 January 2024].

Shahbazi, N. (2023) '2D & 3D Game Art Pipeline – Step by Step Guide', Pixune, 18 May. Available at: https://pixune.com/blog/game-art-pipeline/ (Accessed: 11/01/2024).

Clayton, P. (2024) 'How long does it take to make a video game?', Indivisible Gaming. Available at: https://indivisiblegame.com/how-long-does-it-take-to-make-a-video-game/ (Accessed: 11/01/2024)

N-iX Game & VR Studio. (2024) N-iX Game & VR Studio. Available at: https://gamestudio.n-ix.com/gaming/ (Accessed: 11/01/2024)

Stability AI. (2024) Stability AI. Available at: https://stability.ai/ (Accessed: 28/11/2023).

CompVis. (2022) Stable Diffusion. Available at: https://github.com/CompVis/stable-diffusion (Accessed: 28/12/2023).

OpenAI. (2024) DALL-E 3. Available at: https://openai.com/dall-e-3 (Accessed: Accessed: 28/12/2023).

Midjourney. (2024) Midjourney. Available at: https://www.midjourney.com/home (Accessed: Accessed: 28/12/2023).

Vincent, J. (2022) 'AI image generation: Stable Diffusion, explained', The Verge, 15 September. Available at: https://www.theverge.com/2022/9/15/23340673/ai-image-generation-stable-diffusion-explained-ethics-copyright-data (Accessed: 11/012024).

CompVis. Figure (2022) Stable Diffusion. Available at: https://github.com/CompVis/latent-diffusion/tree/main/assets (Accessed: 28/12/2023).

Bartholomew, D.J. et al. (2002). The Analysis and Interpretation of Multivariate Data for Social Scientists. 1st ed. Boca Raton: Chapman & Hall/CRC, p.145.

Aguimar Neto, A. (2023) 'What is Latent Diffusion in AI?', Medium. Available at: https://medium.com/@aguimarneto/what-is-latent-diffusion-in-ai-43aa1ad4f71e (Accessed: 11/01/2024)

Chang, Z., Koulieris, G., Shum, H.P.H. (2023) 'On the Design Fundamentals of Diffusion Models: A Survey'. arXiv. Available at: https://arxiv.org/pdf/2306.04542.pdf (Accessed: 11/01/2024).

Rombach, R. et al. (2022) 'High-Resolution Image Synthesis with Latent Diffusion Models'. arXiv, p4. Available at: https://arxiv.org/pdf/2112.10752.pdf (Accessed: 11/01/2024).

Ronneberger, O. et al. (2015) 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. arXiv. Available at: https://arxiv.org/pdf/1505.04597v1.pdf (Accessed: 11/01/2024).

AUTOMATIC1111. (2023). 'stable-diffusion-webui'. Available at: https://github.com/AUTOMATIC1111/stable-diffusion-webui (Accessed: 18/01/2024).

Comfyanonymous. (2024) 'ComfyUI'. Available at: https://github.com/comfyanonymous/ComfyUI (Accessed: 28/12/2023).

Yubin. (2023) 'Beginner's Guide to ComfyUI for Stable Diffusion'. Available at: https://aituts.com/comfyui/ (Accessed: 08/01/2024).

Andrew, Installation Guide. (2023). 'How to install ComfyUI.' Available at: https://stable-diffusion-art.com/how-to-install-comfyui/ (Accessed: 08/01/2024).

Ltdrdata. (2024) 'ComfyUI-Manager'. Available at: https://github.com/ltdrdata/ComfyUI-Manager (Accessed: 28/12/2023).

Andrew, Beginner's Guide. (2023) 'Beginner's Guide to ComfyUI', Available at: https://stable-diffusion-art.com/comfyui/ (Accessed: 11/01/2024).

SG161222. (2023) 'Realistic_Vision_V6.0_B1_noVAE', Available at: https://huggingface.co/SG161222/Realistic_Vision_V6.0_B1_noVAE (Accessed 15/01/2024).

Digiplay. (2023) 'GhostMix'. Available at: https://huggingface.co/digiplay/GhostMix (Accessed 15/01/2024)

Stable Diffusion XL. (2024) 'Stable Diffusion XL'. Available at: https://stablediffusionxl.com/ (Accessed: 28/12/23).

Stable Diffusion Art. (2023) 'Stable Diffusion Models: a beginner's guide'. Available at: https://stable-diffusion-art.com/models/ (Accessed: 11/01/2024).

OpenAI. (2021) 'CLIP'. Available at: https://github.com/openai/CLIP (Accessed 19/01/2024)

Radford, A. et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. [online] Available at: https://arxiv.org/abs/2103.00020 [Accessed 15/012024].

Ramesh, A. et al. 2024. Hierarchical Text-Conditional Image Generation with CLIP Latents. [online] ar5iv. Available at: https://ar5iv.org/html/2204.06125 [Accessed 15/01/2024].

TensorFlow, (2023). Autoencoder. [online] TensorFlow. Available at: https://www.tensorflow.org/tutorials/generative/autoencoder [Accessed 18 January 2024].

Kingma, D.P. and Welling, M., 2014. Auto-Encoding Variational Bayes. [online] ArXiv. Available at: https://arxiv.org/abs/1312.6114 [Accessed 18 January 2024].

Podell, D. et al. (2023). SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. Stability AI, Applied Research. Available at: https://arxiv.org/pdf/2307.01952.pdf. (Accessed: 11/01/2024).

Blenderneko, 2023. KSampler Advanced. [online] ComfyUI Community Manual. Available at: https://blenderneko.github.io/ComfyUI-docs/Core%20Nodes/Sampling/KSampler%20Advanced/ [Accessed 28/12/2023].

Gaudiano, S., (2023). King of Dragon Pass. [online] Available at: https://www.gaudianoart.com/kodp [Accessed 18 January 2024].