The Importance of Context in Image Generation: A Case Study for Video Game Sprites

Roberto Gallotta, Antonios Liapis, and Georgios N. Yannakakis

Institute of Digital Games, University of Malta, Msida, Malta
{roberto.gallotta,antonios.liapis,georgios.yannakakis}@um.edu.mt

Abstract. In recent years text-to-image generative models have found a wide range of applications, including the generation of video game assets such as sprites. However, sprites are often reused throughout a game, leading to repetitive visuals. Creating custom sprites tailored to different in-game environments can be a resource-intensive task when done manually. In this paper, we address the challenge of generating visually appealing sprites that not only align with the overall theme of the game but also adapt to the varying environments within it. We investigate how different contextual details of these environments, provided as prompts to a text-to-image model, influence the resulting sprites. Our approach is demonstrated through sprite generation for the video game design tool *LLMaker*. Experiments using objective performance metrics confirm the effectiveness of our method, while a user study shows that the generated sprites resonate with players.

Keywords: Generative AI · Computational Creativity · Video Games.

1 Introduction

The creation of video games blends various artistic disciplines—such as visual art, sound design, narrative, and interaction design—to craft immersive experiences [17]. Designing visual assets such as characters, environments, and objects traditionally requires significant manual effort. Some of these elements can be represented as *sprites*: transparent two-dimensional images that are overlaid onto backgrounds. In old-school games, low-resolution sprites with limited color palettes were heavily used due to hardware constraints. Sprites were often reused throughout the game to save resources and development time. For example, each *Goomba* in *Super Mario Bros* (Nintendo, 1985) is the same sprite in the levels with warm tones as in those with colder tones (e.g. the underwater world). Even though most modern games use 3D assets, sprites remain popular in the independent game developer community. Commercially successful indie games such as *Hollow Knight* (Team Cherry, 2017) and *Darkest Dungeon* (Red Hook Studios, 2016) make use of sprites in a stylized manner. However, the limited number of sprites can lead to repetitive visuals, which may detract from player immersion. This issue highlights the need for more efficient, creative methods of generating diverse and unique assets without compromising quality.

Today the integration of AI-driven tools such as text-to-image generative models offers a novel approach to asset creation. These models generate images based on text descriptions (*prompts*), allowing game developers to quickly produce a wide variety of unique sprites, environments, and other visual elements. This significantly reduces the manual effort involved in asset creation. Stable Diffusion (SD) [23], DALL-E [22], and VQGAN-CLIP [4] are able to generate images from text thanks to multimodal models such as Contrastive Language-Image Pre-Training [21], which establish connections between text and images by learning from large datasets of paired data. These models enable more flexible and creative generation of assets that are not limited to predefined classes, offering a wide scope for generating diverse and intricate designs.

Despite these advancements, there are challenges associated with using generative AI in game asset creation. While SD and similar models can produce visually diverse content, the generated sprites may not always be consistent with their in-game surroundings. Ensuring that these AI-created assets integrate seamlessly with their in-game environment remains a key hurdle, but addressing this can lead to the generation of more immersive and customized assets.

In this work, we introduce the concept of *context*: additional details of the in-game environment that guide sprite generation by an SD model. We investigate how different contexts impact sprite generation within the video game design tool *LLMaker* [8]. Our paper contributes to the field of computational game creativity [17] by automating and evaluating a design task under specific requirements. Specifically, we generate a large number of sprites and evaluate them using automated image metrics. To further validate our findings and ensure the generated assets align with human taste, we conduct a user study. Finally, we release the source code for our experiments at `https://github.com/gallorob/sd_context_control` to facilitate replication and extension of our work.

2 Related Work

In this work we are interested in generating video game sprites that are consistent with their environment using Stable Diffusion (Section 2.1) in our domain of interest, *LLMaker* (Section 2.2).

2.1 Stable diffusion

Diffusion models [9,26] are a class of generative models that create images conditioned by a text prompt, starting from random noise. These models learn a parametrized function that can iteratively reverse the additive noise from the initially noisy image. Training these models starts by adding disruptive Gaussian noise to "clean" images progressively over multiple timesteps. Then, the model is trained to apply the correct denoising diffusion step to obtain back the original image [9]. The neural network architecture most commonly used in such settings is the U-Net [24], suitable for image-to-image translation tasks. Such a generative approach has been successfully applied to image synthesis [7], achieving better image fidelity and mode coverage than Generative Adversarial Networks and Variational Auto-Encoders.

Stable diffusion (SD) models are popular latent diffusion models [23] that can generate images from different input modalities, such as text only or a combination of text and images. This allows for different image editing and generation paradigms that are more controllable; for example, ControlNet [30] allows for the conditional generation of images starting from a text description and a "control" image that the model uses as starting point for the subsequent generation.

As these models are trained on a large corpus of images and text pairs, they rarely correctly capture certain niche styles or subjects. However, instead of retraining these models from scratch, low-rank matrix adaptation (LoRA) [10] can instill new knowledge into the models. Personalizing these models using LoRA is extremely easy, as it requires only a few annotated images.

2.2 LLMaker

LLMaker [8] is a mixed-initiative content creation [16] tool that leverages Large Language Models (LLMs), namely GPT-3.5-Turbo [2], to assist a human user designing content for a reverse dungeon crawler video game, *Dungeon Despair*. The game draws inspiration from *Darkest Dungeon* (Red Hook Studios, 2016), with heroes battling against enemies placed in rooms and corridors of a dungeon. As it is driven by an LLM, there are no constraints on the themes and styles of the rooms and corridors. The game is played in a side view perspective, with a single background image per room and one foreground transparent sprite per enemy, hero, treasure, and trap. This makes *Dungeon Despair* the ideal game to test our approach for customized sprite generation, as we can quickly iterate through a multitude of different types of rooms and enemy types.

3 Methodology

In this section we explain our concept of *context* for Stable Diffusion (Section 3.1) and how we can use it when generating assets for *Dungeon Despair* using *LLMaker* (Section 3.2).

3.1 What is Context?

Generating images with Stable Diffusion (SD) models starts with a text description of the subject. The more detailed the description, or *prompt*, the more accurate and tailored the resulting image will be. To generate sprites for a video game, the simplest prompt might include only the entity's name and description, along with specific *keywords* for a particular style (such as "trending on DeviantArt" or "Unreal Engine"). Once generated, the sprite can be placed anywhere in the game. However, while the same sprite may be reused in different environments, it might not always blend well with them. We propose incorporating environment-specific information *during* sprite generation, which we define as *context*:

Context is any (non-keywords) additional domain information described as text that is provided to a text-to-image model for a more customized image generation.

This definition suggests that different contexts, derived from the environment, can influence the final sprite in different ways. To better understand how context can influence sprite generation, it is important to recognize that environments within a game can vary significantly. Each environment is defined by its atmosphere and physical details, giving it a unique identity within the game. This information can be provided directly by the game or obtained by analyzing a snapshot of the environment. For example, in a game such as *Dungeon Despair*, sprites of the same entity can be placed in different rooms. A room's background is generated based on its name and description, representing its *semantics*. We can use this semantic information as context to generate a more tailored entity sprite for that specific room. However, since the background is generated by an SD model, it might visually differ from its original semantic description. We can instead provide a *caption*—a description derived directly from the image of the environment—as context instead. Both semantics and captions operate on a text description of the environment, but while semantics refer to the information that defines the environment, the caption is drawn from its visual representation. Beyond descriptions of the image, context can also be extrapolated from visual elements of the environment. One simple yet powerful way to match the sprite to its surroundings is to align the sprite's palette to the dominant *colors* of the environment. A sprite that shares this palette will blend in with its environment. While the dominant colors are a valuable property, since we are generating sprites using a SD model, we can enhance this process by incorporating the actual *background image* during sprite creation. This allows the model to integrate not just the dominant colors but also the shapes, lighting, and texture of the environment into the sprite's design. Each of these factors shape the player's perception of the environment, and as a result, they also influence the appearance of the sprites that populate it. By embedding these environmental details into the sprite generation process, we can ensure that the final sprite feels consistent and is better integrated into its environment.

In this work we inspect the following contexts:

None: The sprite is generated based only on its name and description;
Col: The sprite is generated using its name and description, and the dominant colors of the background image;
Sem: The sprite is generated using its name and description, and the name and description of the room;
SemCol: The sprite is generated using its name and description, the name and description of the room, and the dominant colors of the background image;
SemImg: The sprite is generated using its name and description, the name and description of the room, and the background image;
Cap: The sprite is generated using its name and description, and a generated description of the background image;

Fig. 1: An example of sprites generated for "*Mischievous Imp*" at different context levels, in two different rooms, sampled from the same run. Top: "*Submerged Arena*", bottom: "*Hieroglyphic Hallway*".

CapCol: The sprite is generated using its name and description, the generated caption of the background image, and its dominant colors; and
CapImg: The sprite is generated using its name and description, the generated caption of the background image, and the background image.

The combinations of contexts were chosen to balance environmental descriptions (through semantics or caption) with visual details (through colors or the image itself). Other permutations, such as COLIMG or SEMCAPCOL, were omitted as they would introduce redundant information. An example of the effects of each context on the generation of sprites for the same entity in two different rooms is shown in Figure 1.

3.2 Generating via Context

We now illustrate the sprite generation pipeline for each context. We employ a fine-tuned SD model, namely A-Zovya RPG Artist Tools v4[1], and two LORAs (Necro Sketcher[2] and DarkestDungeon[3]), to obtain enemies' sprites and rooms backgrounds that closely resemble the style of *Darkest Dungeon* (Red Hook Studios, 2016). We also use StabilityAI's VAE[4], as recommended in the user guide of A-Zovya RPG Artist Tools v4. All room backgrounds are generated at a resolution of 1024×512 pixels, and each enemy sprite is generated at a resolution of 512×768 pixels. The Compel [5] library weighs different parts of the prompt. As SD models do not operate on transparent images, we also rely on the *Rembg* [6] Python library to remove the background of generated sprites.

For the None context, we make use of the following prompt template (see Figure 2):

[1] Available at: `https://civitai.com/models/8124?modelVersionId=250344`
[2] Available at: `https://civitai.com/models/70147/darkest-dungeon-style-or-necro-sketcher-or-lora`
[3] Available at: `https://civitai.com/models/65324/darkestdungeon`
[4] Available at: `https://huggingface.co/stabilityai/sd-vae-ft-mse-original`

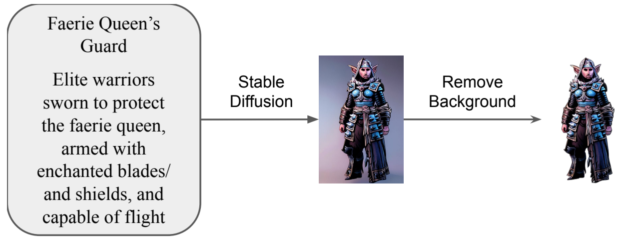


Fig. 2: The pipeline for generating images using the None context.

```
(full body)+++ {entity name}: ({entity description})++, (flat
empty background)+++, masterpiece++, highly detailed+
```

Each of the following contexts alters the above prompt by adding the context information [5].

In the Col context, we extract the dominant colors of the room image to include in the prompt. We do this by first quantizing the image to 32 colors, and matching them with the closest standard HTML colors. We pick the 3 most common colors in this image, and obtain their names (from the HTML colors).

In the Sem context, we include the name and the description of the environment in the prompt. This information is provided by *LLMaker*, as it is used to generate the background image of the room (see Fig. 1).

When generating the image of the room, it is possible that the final image does not match the prompt fully. This is particularly the case when the description of the room contains many details, which rarely end up appearing in the generated SD output. For this reason, the Cap context works directly on the image. We use the pretrained vision and language BLIP model [15] to generate a caption of the image, which is then included in the prompt.

The contexts SemCol and CapCol include the dominant colors information along with descriptions of the environment, either from its semantics or from the generated caption.

Finally, SemImg and CapImg generate the final sprite directly on the room image, introducing detailed information to the SD model. These two contexts however come at an increased computational cost: the sprite requires two passes of SD to be generated, instead of just one as in the previous contexts. The sprite is first generated using either the semantics or the caption of the room, and then processed to obtain both the alpha mask and the edges mask. The alpha mask is simply the non-zero pixels on the alpha channel of the image, and the edges mask is a grayscale copy of the sprite image containing the sprite edges as identified with the Sobel edge-detection filter. We then crop part of the room image and feed it, along with the edges mask, to an inpainting SD model. The same prompt used in the first step (either semantics or caption) is used again. The final sprite is then obtained by masking the background image using the

[5] All prompts are available at `https://github.com/gallorob/sd_context_control/blob/main/generator.py#L241`.

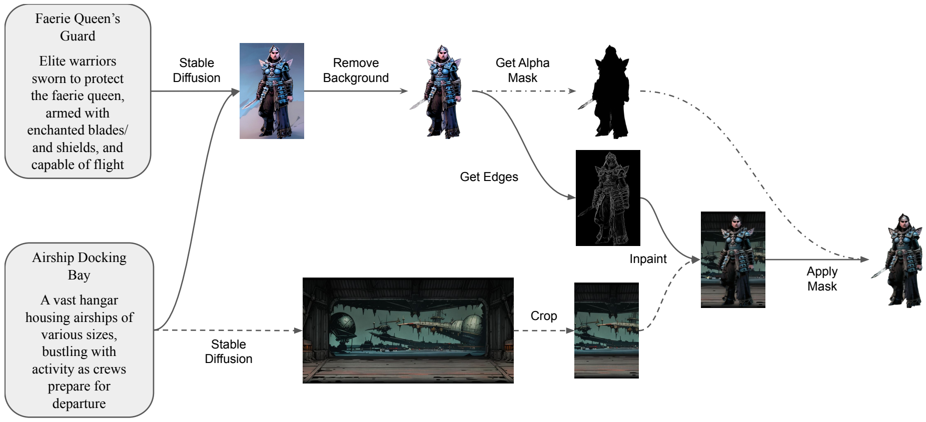


Fig. 3: The pipeline for generating images using the SemImg context. Solid lines indicate operations that concern sprite generation, dashed lines for background image, and dash-dotted lines for masking.

alpha mask obtained in the previous step. A diagram for the SemImg case of this pipeline is presented in Figure 3.

4 Experimental Protocol

In this section we define the experiments we carried out with the goal of evaluating both the quality and the cohesion of the sprites. We first conducted an automated evaluation of a large sample of generated sprites (Section 4.1), and then evaluated a small subset of these images via a user study (Section 4.2).

4.1 Automated Evaluation

In this work we aim to assess whether varying contexts leads to more consistent and different sprites being generated, and their impacts on image quality.

We measure *diversity* between sprites using the Learned Perceptual Image Patch Similarity (LPIPS) metric [31]. LPIPS is a pre-trained deep learning model that measures perceptual similarity in images, based on human perception rather than pixel-wise differences. In this work, we use LPIPS to assess whether an enemy sprite is tailored to its environment by comparing it to a sprite generated with the same context but for a different room.

A key concept introduced in Section 3.1 is that sprites tend to feel more visually consistent when they share the dominant colors of their surroundings. We quantify *consistency* by calculating the ratio of shared colors between an enemy sprite and its background image, after quantizing both images to 32-color palettes based on each images' colors. A higher consistency implies that the sprite shares more dominant colors with the environment.

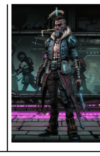
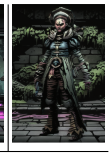
Metric		
Quality	34.17	36.17
Consistency	44%	44%
Complexity	26%	29%
Diversity	0.36	

Table 1: Example of metrics' values for the same entity ("Mad Tinkerer") in different environments ("Neon Alley" and "Serpentine Labyrinth") at SemImg context.

We assess the sprites' *quality* via the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [13,19] model. This pre-trained model analyzes natural scene statistics in the spatial domain to detect distortions such as blurriness, noise, and compression artifacts. BRISQUE provides a quantitative measure of how well an image meets desired visual fidelity standards. A high BRISQUE value indicates that the sprite has minimal noise or blurriness.

When evaluating sprites' quality we are also interested in measuring their *complexity*. One effective way to measure image complexity [18,29] is using the Holistically-Nested Edge Detection (HED) model [27], a deep learning approach that detects edges at multiple scales and fuses them into a highly detailed edge map. In this work, we quantify sprite complexity as the percentage of edges detected by a HED model within the sprite.

We include an example of these metrics computed on two sprites in Table 1. We employ these metrics to evaluate a large collection of sprites generated with different contexts. We use *LLMaker* to obtain a list of 25 thematically distinct rooms, and for each room we generate a thematically accurate enemy. We then proceeded to generate all possible combinations of room and enemy pairs, and generate the enemy sprite for each pair using all contexts. We repeat this generation pipeline for 10 randomized runs. In the end, we obtain a total of 50.000 unique sprites that we can evaluate automatically with the above metrics.

4.2 User Evaluation

While evaluating the sprites with objective metrics can be computationally efficient, we do not know if these metrics evaluate sprites in the same way a human would. We conduct a user study to assess whether users' perceptions of sprite consistency differ from our established metrics, and whether there are any discrepancies between their preferences and our measure of image quality.

To answer these questions, we designed a questionnaire using Google Forms. We defined two separate sections in the questionnaire: in the first section (Q1), users are asked to evaluate images based on their consistency with the environment ("Which image best fits with its surroundings?"), whereas in the second

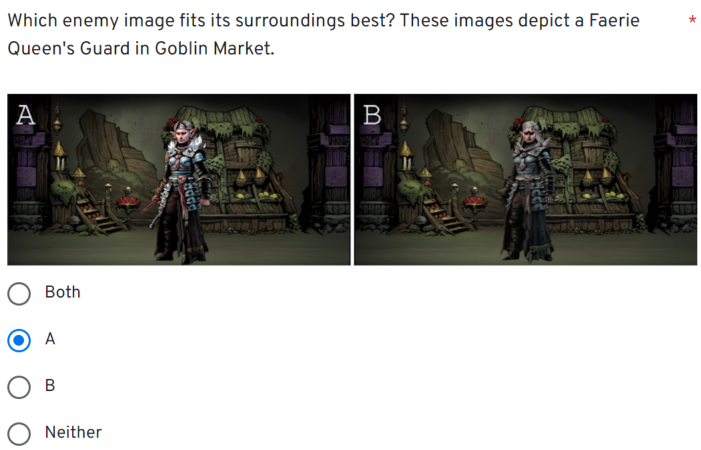


Fig. 4: An example question from the survey for Q1. The user is unaware of the contexts employed in the images generation, but knows what entity is shown and in what kind of environment.

section (Q2) users are asked to express their preference between images ("Which image do you like best?"). Each section is comprised of 20 questions. Each question shows two images (A, B) that differ only by context. The user can give their response via 4-alternative forced-choice (4-AFC) [11]: *Both*, *A*, *B*, and *Neither*. An example question from the questionnaire is shown in Figure 4. The images were randomly sampled from the collection of images generated as discussed in Section 4.1. To avoid fatiguing users or having to create multiple questionnaires, we limited the number of pairs to 20 per section. We sample the sprites ensuring that all contexts are chosen at least once per section.

The above process was approved by the University Research Ethics Committee of the University of Malta before data collection commenced. Participants gave informed consent to participate in the user study. No demographic data was collected. Responses to the questionnaire were anonymized and processed under European GDPR guidelines. We disseminated the questionnaire internally and advertised it over social media (LinkedIn, X).

5 Results

This section presents the results obtained for both the automated evaluation in Section 5.1 and the user study in Section 5.2.

5.1 Results for Automated Evaluation

We computed each metric for each context, averaging across all runs. We then ranked each context based on each metric. We used an Elo-like rating system

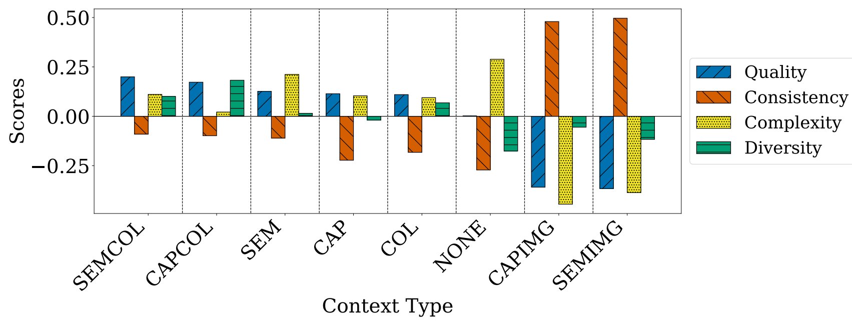


Fig. 5: The context scores for each metric, ordered by quality.

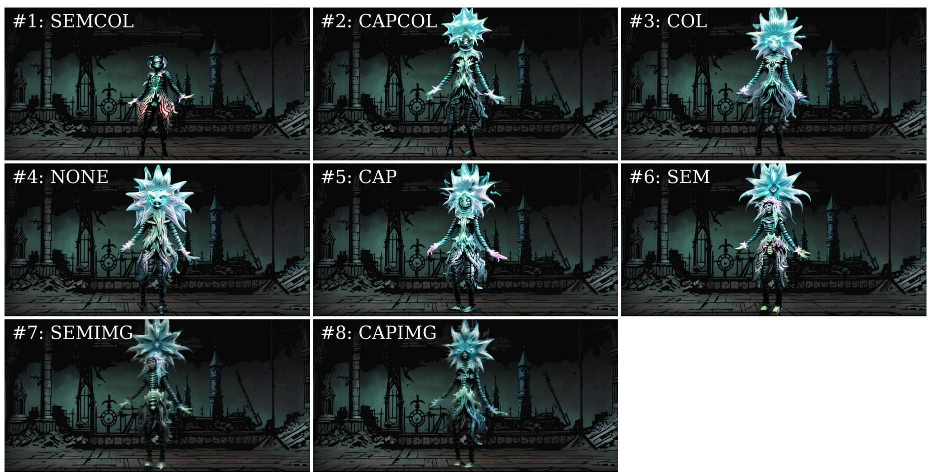


Fig. 6: Ranking example for the best sprite generated according to quality. SemImg and CapImg sprites are more blurry than all other contexts.

when evaluating all possible combinations of room, entity, and context: for each metric, we increased the score of the context with the better value in that metric, and decreased the score of the other. This scoring technique allowed us to rank contexts relatively to each other, highlighting their difference in performance. We present the final normalized scores (between -1 and 1) in Figure 5.

We find that the inpainting process significantly hinders both the sprites' quality and complexity: CapImg and SemImg rank last in terms of both quality and complexity, as shown in Figure 5. This is likely due to the fact that inpainting often results in low-detail images, which can be exacerbated by using an edge mask with smooth edges (*e.g.*, non-binary masks) that don't clearly separate areas that have to be preserved from areas that have to be painted over[6]. As a result, our final sprites tend to be blurry and lacking in fine details (see Figure 6).

[6] This behavior is usually controlled by the strength of the inpainting process.

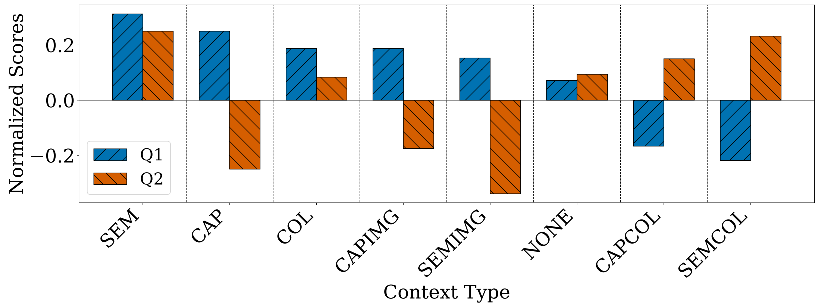


Fig. 7: The normalized context levels scores for both Q1 and Q2, ordered by Q1.

In contrast, integrating the background image has a clear positive impact on the consistency of the sprites. Both CapImg and SemImg consistently outperform other contexts, even when the dominant colors of the background image are included (Col, CapCol, and SemCol). CapCol and SemCol are ranked third and fourth respectively for consistency, but rank first and second in terms of diversity. This is because CapCol and SemCol introduce more details in the prompt, resulting in more unique images (compared to Sem, Cap, or Col alone). Additionally, CapCol and SemCol yield higher quality images compared to their counterparts that do not include color information, likely due to the additional structural information provided by the dominant colors.

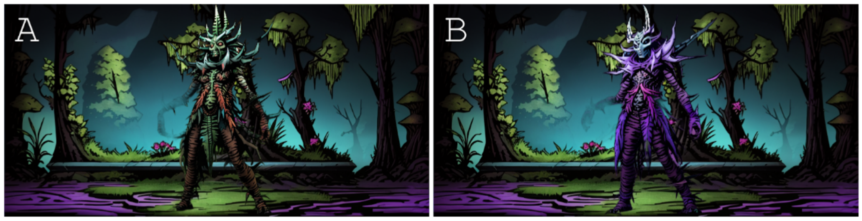
5.2 Results for User Study

After letting users participate in the questionnaire for two weeks, we collected a total of 16 responses.

Similarly to the scoring process introduced in Section 5.1, we increase the score of a context if it was picked by the user, and decrease it otherwise. We increase the score of both contexts when the user picks the "Both" option, and decrease the score of both options when the user picks the "Neither" option. Due to the original unbalance of stimuli (each context appears a different number of times in the questionnaire), we also scale the scores of each context by its frequency. We present the normalized final scores of the user study in Figure 7.

From these results, we see that user responses do not align with the metrics we measured. We can use Kendall's tau [14] to measure the correlation between rankings, setting the significance threshold to $p < 0.05$. Kendall's tau is a robust tool for measuring correlation between rankings that does not assume normally distributed data, and is particularly appropriate for small sample sizes as in this case. We found no significant correlation between rankings for Q1 and Q2, nor between Q1 or Q2 and any of the objective metrics. However, we found that users did not pick at random for either sections of the questionnaire. The Sem context was overall the most picked option both for sprite coherence and personal preference. CapImg and SemImg, while scoring high in terms of coherence, were

Which enemy image fits its surroundings best? These images depict a Thorned Treant in Enchanted Glade. *



(a) 10 out of 16 participants voted for A (None context) over B (Col context).

Which enemy image do you like more? These images depict a Mutant Enforcer in Forgotten Throne Room. *

(b) 12 out of 16 participants voted for B (SemCol context) over A (Col context).

Fig. 8: Examples of questions from the survey where participants showed a strong preference for one context over another, for Q1 (Figure 8a) and Q2 (Figure 8b).

ranked quite low in terms of personal preference. Again, we assume this is a result of the inpainting process generating blurry images.

We include the two questions, one per each section, where most users voted for one context over the other option. In Figure 8a, we can actually see one point of failure of SD using the Col context: one of the colors in the room image was purple, and the generated sprite was mostly purple, resulting in an inconsistent image. On the other hand, the None context generated a sprite that fit with the environment well enough already, matching the green and brown hues of the background image. In terms of preference, in Figure 8b it seems that the users preferred the more vibrant sprite. This choice pattern however was not consistent in other sprite pairings for the Q2 section.

6 Discussion

From our results, we find that there is no single context that maximizes our proposed metrics. Based on our objective metrics, however, both SemCol and CapCol are solid contexts to employ in the generation of diverse and high quality sprites. According to users, however, Sem context instead is the ideal context level to generate such sprites.

While both findings are valuable, there are multiple reasons for this discrepancy. Mainly, the scale of the user study was small, with few participants evaluating a limited subset of generated images. Despite efforts to minimize bias and randomize stimuli, different pairs of sprites might have yielded more informative results. In terms of the automated evaluation, ad-hoc metrics such as BRISQUE can be unreliable for cartoon-like images, as these metrics are typically trained on photorealistic images. Conversely, humans are able to evaluate images regardless of their visual style. It is worth noting that our findings are specific to the chosen style, as we used a custom SD model fine-tuned on RPG art and a LoRA for the style of *Darkest Dungeon* (Red Hook Studios, 2016). Different models or LoRAs would likely yield different results. In an example use-case [8] of the *LLMaker* tool using the Sem context for sprite generation, our system generates a "Punker Capybara" enemy that is a humanoid rather than resembling a giant capybara. This would not be the case with LoRAs finetuned for photorealistic image generation. In preliminary work, we tried generating the same entities and rooms using a non-finetuned SD model, Runwayml's SD 1.5[7], but we found it unreliable to use: multiple entities missed entire sections or were duplicated, and the same entity would be generated with different styles when varying just the context. We were also unable to compare our generated assets with human-authored sprites, which could have offered valuable feedback for evaluating the performance of our automated generation process as a whole. Finally, when sprites share dominant colors with their environment, they can blend in and become difficult for players to distinguish. Using complementary colors instead would create a striking visual contrast, keeping the sprites distinct without disrupting the overall visual harmony.

The concept of contexts can be relevant to different game development pipelines that personalize their graphical assets, as explored in recent work [3,28]. However, this personalization tends to be more effective when driven by the users themselves [32]. Beyond games, the use of context—understood as additional information provided to a system—has already been applied in areas such as user interface personalization [25] and alternative communication tools [12]. In music, personalization has been investigated through text-to-music latent diffusion models [20], though without including context into prompts. One promising but underexplored avenue of research lies in chatbot personalization: by incorporating user information as context, chatbot agents could offer more tailored interactions, moving beyond reliance on manually crafted personas [1].

7 Conclusions

In this work we introduced the concept of *context*, defined as different levels of detail that can be included in the prompt that controls a text-to-image model.

[7] Previously available at `https://huggingface.co/runwayml/stable-diffusion-v1-5`, now removed.

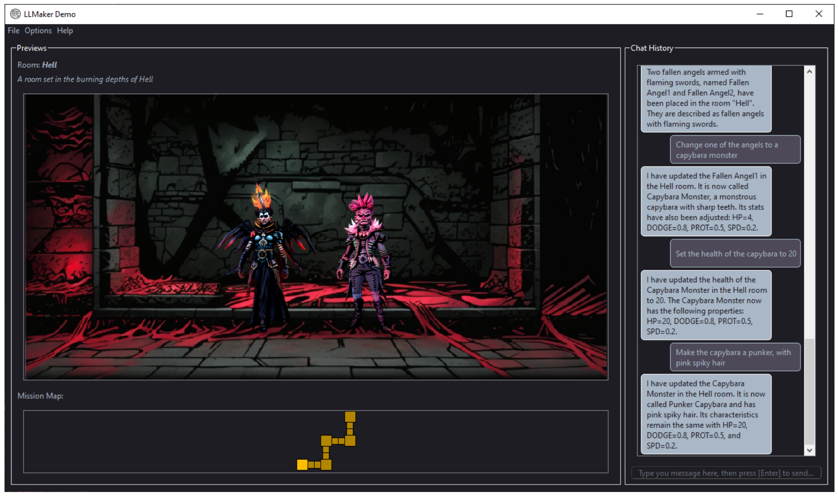


Fig. 9: An example room generated with SEM context in the graphical user interface of *LLMaker*. The interaction is from test case T5 of [8].

We applied different contexts in a pipeline to generate both high-quality and visually consistent sprites for a video game, *Dungeon Despair*. We tested our approach with an objective evaluation based on multiple metrics computed on generated sprites, and validated our findings by conducting a user study on a subset of these sprites. While the two evaluations resulted in different contexts being ranked higher, they both highlighted strengths and shortcomings of each context. We hope that this work will inspire more research in personalization of generated content in different domains and modalities.

Acknowledgements

This project has received funding from the Malta Council for Science and Technology (MCST) through the SINO-MALTA Fund 2022, Project OPtiMaL.

References

1. Ait Baha, T., El Hajji, M., Es-Saady, Y., Fadili, H.: The power of personalization: A systematic review of personality-adaptive chatbots. SN Computer Science **4**(661) (2023)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)

3. Böffel, C., Würger, S., Müsseler, J., Schlittmeier, S.J.: Character customization with cosmetic microtransactions in games: Subjective experience and objective performance. Frontiers in Psychology **12** (2022)
4. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In: Proceedings of the European Conference on Computer Vision. pp. 88–105. Springer Nature Switzerland (2022)
5. damian0815: Compel. `https://github.com/damian0815/compel` (2023)
6. danielgatis: Rembg. `https://github.com/danielgatis/rembg` (2023)
7. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems **34**, 8780–8794 (2021)
8. Gallotta, R., Liapis, A., Yannakakis, G.N.: Consistent game content creation via function calling for large language models. In: Proceedings of the IEEE Conference on Games (2024)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
10. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
11. Jäkel, F., Wichmann, F.A.: Spatial four-alternative forced-choice method is the preferred psychophysical method for naïve observers. Journal of Vision **6**(11), 13–13 (2006)
12. Kane, S.K., Linam-Church, B., Althoff, K., McCall, D.: What we talk about: designing a context-aware communication tool for people with aphasia. In: Proceedings of the International SIGACCESS Conference on Computers and Accessibility. p. 49–56. Association for Computing Machinery (2012)
13. Kastryulin, S., Zakirov, J., Prokopenko, D., Dylov, D.V.: PyTorch image quality: Metrics for image quality assessment. arXiv preprint arXiv:2208.14818 (2022)
14. Kendall, M.G.: A New Measure of Rank Correlation. Biometrika **30**(1-2), 81–93 (1938)
15. Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of the International Conference on Machine Learning (2022)
16. Liapis, A., Smith, G., Shaker, N.: Mixed-initiative content creation. In: Shaker, N., Togelius, J., Nelson, M.J. (eds.) Procedural Content Generation in Games: A Textbook and an Overview of Current Research, pp. 195–214. Springer (2016)
17. Liapis, A., Yannakakis, G.N., Togelius, J.: Computational game creativity. In: Proceedings of the International Conference on Computational Creativity (2014)
18. Machado, P., Romero, J., Nadal, M., Santos, A., Correia, J., Carballal, A.: Computerized measures of visual complexity. Acta Psychologica **160**, 43–57 (2015)
19. Mittal, A., Moorthy, A.K., Bovik, A.C.: Blind/referenceless image spatial quality evaluator. In: Proceedings of the Asilomar Conference on Signals, Systems and Computers (2011)
20. Plitsis, M., Kouzelis, T., Paraskevopoulos, G., Katsouros, V., Panagakis, Y.: Investigating personalization methods in text to music generation. arXiv preprint arXiv:2309.11140 (2023)
21. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)

22. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: Proceedings of the International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8821–8831. PMLR (2021)
23. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. arXiv preprint arXiv:2112.10752 (2022)
24. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical image computing and computer-assisted intervention (2015)
25. Schaub, F., Könings, B., Lang, P., Wiedersheim, B., Winkler, C., Weber, M.: Prical: context-adaptive privacy in ambient calendar displays. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing. p. 499–510. Association for Computing Machinery (2014)
26. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of the International Conference on Machine Learning (2015)
27. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of IEEE International Conference on Computer Vision (2015)
28. Zammit, M., Liapis, A., Yannakakis, G.N.: CrawLLM: Theming games with large language models. In: Proceedings of the IEEE Conference on Games (2024)
29. Zammit, M., Liapis, A., Yannakakis, G.N.: MAP-Elites with transverse assessment for multimodal problems in creative domains. In: Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMusArt) (2024)
30. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE International Conference on Computer Vision (2023)
31. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (2018)
32. Zhu, J., Ontañón, S.: Player-centered AI for automatic game personalization: Open problems. arXiv preprint arXiv:2102.07548 (2021)