Ctrl + Create: Empowering Creative Control in AI-Driven Rapid Level Design

Arthur Baars
Utrecht University
Utrecht, Netherlands
arthurbaars@grachtwerk.nl

Fabian Akker
Breda University of Applied Sciences
Breda, Netherlands
akker.f@buas.nl

Angelos Chatzimparmpas
Utrecht University
Utrecht, Netherlands
a.chatzimparmpas@uu.nl

Johannes Pfau
Utrecht University
Utrecht, Netherlands
j.pfau@uu.nl

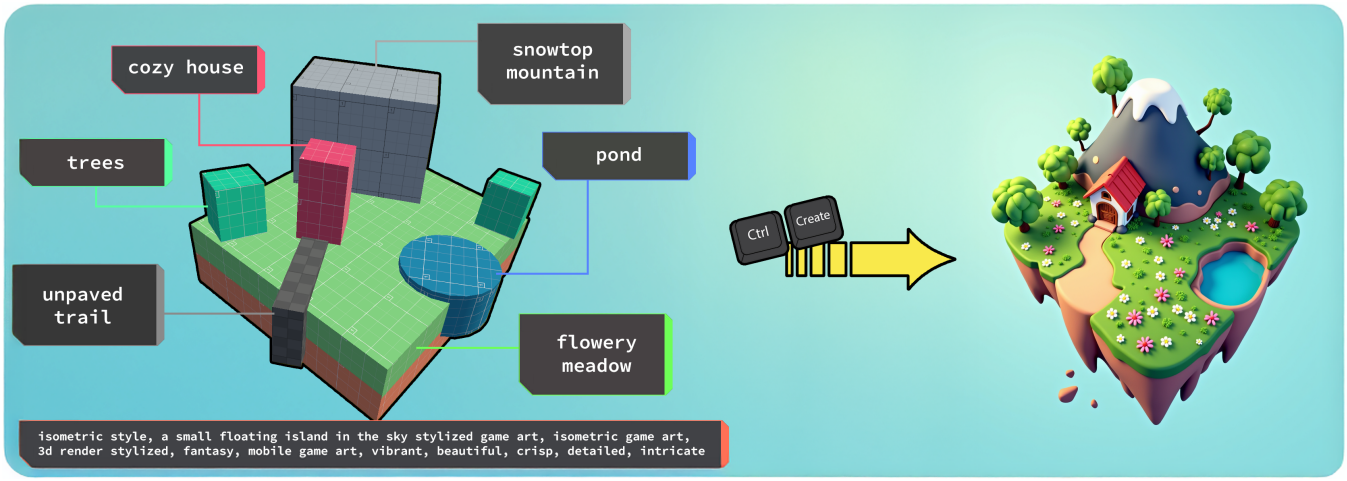


Figure 1: How to make sure a creative vision survives the many roles and iterations of game development? In light of rapid iteration and to close the gap between ideation and implementation as much as possible, visual generative AI approaches seem promising; yet, a significant part of control is inherently surrendered to the image model. In our proposed approach, rough level maps can be allocated with separate prompts, united with a global world style, and unified at generation time, aiming at elevating the level of control and adherence to the designer's vision.

Abstract

Level design is one of the most labor-intensive processes in video game development – especially because communicating a creative vision across designers, artists, writers, and gameplay engineers requires extensive iterative refinement. Traditionally, this involves rounds of *white boxing* and *set dressing*, which distributes labor effectively but limits rapid exploration and creative experimentation. Recent advances in AI-driven image generation offer timely opportunities to transform these workflows, but risk losing control to the model's interpretation and capabilities. We investigate the impact of generative AI on designers' control, expressiveness, and efficiency through a mixed-methods study (n=20), comparing drawing (full control), text-to-image (full AI), and our approach: a visualization pipeline combining generative AI with user-centered spatial control. It succeeded in enhancing visual expressiveness and sense of control over bare AI, matched manual drawing (without necessitating advanced skills), and enabled faster iteration – highlighting the potential of giving back control to creative visionaries.

CCS Concepts

• **Computing methodologies** → *Artificial intelligence*; • **Human-centered computing** → *User centered design*; *User studies*.

Keywords

Generative AI, Video Games, Level Design, Whiteboxing

ACM Reference Format:
Arthur Baars, Fabian Akker, Angelos Chatzimparmpas, and Johannes Pfau. 2026. Ctrl + Create: Empowering Creative Control in AI-Driven Rapid Level Design. In *Foundations of Digital Games (FDG '26), August 10–13,*

FDG '26, Copenhagen, Denmark

ACM ISBN 979-8-4007-2495-4/26/08
https://doi.org/10.1145/3815598.3815646

2026, Copenhagen, Denmark. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3815598.3815646

1 Introduction

Designing and prototyping 3D environments – especially game levels – is an inherently complex and iterative process. Core industry practices include *whiteboxing* levels during early phases of development, where layouts are blocked out with simple geometry[21]. Environmental artists subsequently produce 2D level paintovers of screenshots of these white boxed levels to establish visual coherence between functional play spaces and intended artistic direction. Unlike text labels or mood boards, paintovers ground artistic direction in the actual spatial layout — making it possible to evaluate whether the intended look and feel holds up in context. This process visualizes the final aesthetic presentation and provides team feedback[37]. However, these full concept art overpaints are time-consuming, and level designs are vulnerable to sudden narrative or gameplay changes, which can force extensive rework.

Such rework is not rare: as Karlsson et al. report in their cross-sectional study of game development workflows, levels are often "reworked or sometimes even scrapped far into production due to sudden updates in the narrative" [21]. This uncertainty forces designers to keep prototypes low in detail or risk wasted effort by refining assets too soon. As one designer wrote:

> **(Level designer 1) [21].** *"I spent weeks finalizing it [the white boxed level], before finally finding out that everything needed to be changed. Now I don't do detailed white boxes anymore, because I know I will have to throw away half of my work. However, that [the lack of white box details] makes it harder for the creative director to understand the level, which isn't good either."*

These problems illustrate a definite gap in current practices: the need for tools that allow for fast, iterative prototyping without compromising visual clarity or creative flexibility. While white boxing (also called a *grey box* or *block out*), defined as an outline of a game's level without any art assets attached, used for playtesting and prototypes [21] is a staple in the game development industry, the limitations in conveying detailed artistic vision during early phases of development often delay or complicate collaboration between design teams and other stakeholders. Misalignment between spatial layout and artistic vision are cheaper to catch early — yet if a level's artistic direction cannot be meaningfully visualized during white boxing, such misalignments might go unnoticed until significant effort has already been invested.

In this paper, we introduce an AI-driven visualization pipeline that integrates directly into a game engine, allowing designers to combine whiteboxing and full rendered overpaints within a single workflow (see Figure 1). Our key contributions are as follows:

- A pipeline that extracts segmentation, depth, and normal maps from editable 3D scenes in Unity;
- A method for conditioning these within a genAI model to produce stylized, spatially consistent 2D renderings;
- A real-time feedback loop that supports rapid iteration while maintaining artistic quality and direction;
- A mixed-methods user study with 20 participants, assessing the pipeline's impact on efficiency (time and effort per successful iteration) and creativity (perceived control and expressive capacity) in comparison to two deliberately chosen baselines: manual sketching, representing an upper bound of full user control, and out-of-the-box AI prompting, representing minimal direct control.

We find that our approach enhances visual expressiveness and the sense of control compared to pure AI generation, matches manual drawing without requiring advanced skills, and enables faster iteration. It is designed for creators who need to rapidly prototype 3D environments while preserving precise layouts and artistic intent – not to move set dressing earlier in the pipeline, but to make its visual direction communicable earlier, reducing the risk of misalignment down the line. These capabilities could also benefit filmmaking by reducing manual effort in pre-visualization, VR/AR design by enabling faster iteration with spatial coherence, and architectural visualization by accelerating style exploration without compromising accuracy. Across these contexts, our approach offers a practical way to overcome the persistent trade-off between efficiency, clarity, and flexibility in high-quality prototyping, which meets the requirements of usable AI in practice [26].

2 Related Work

Deterministic methods, which operate based on predefined rules and algorithms, lay the foundation for early AI contributions to game and level development. For example, Liapis et al.'s Sentient Sketchbook uses novelty search to iteratively assist designers with alternative level layouts [23]. More broadly, search-based procedural content generation (SBPCG) has been surveyed extensively, highlighting evolutionary and meta-heuristic methods as key early approaches [42]. Classic mixed-initiative tools such as Tanagra [35] demonstrate how reactive generators can collaborate with designers by filling in constraints like geometry and pacing [34].

Building on these foundations, non-diffusion machine learning approaches extended procedural generation into more adaptive forms. Summerville et al. provide a comprehensive survey of PCG via machine learning (PCGML), covering autoencoders, Markov models, and neural networks [36]. Generative adversarial networks (GANs) enabled advanced structural content, such as DOOM-like levels [39] and tile-based platform levels [16], but often struggle with controllability. Other approaches focus explicitly on control: Jiang et al. propose reinforcement learning for 3D level generation in Minecraft [19], and more recently, Baek et al. extend this line of work by combining reinforcement learning with natural language instructions, enabling levels to be generated under unseen prompts while maintaining controllability (IPCGRL) [5]. While Zakaria et al.'s Start Small framework shows how controllable generators can be trained without large datasets [41], the Controllable Path of Destruction system takes yet another angle, generating content by repairing destructive states (of game levels) with fine-grained control [33]. Mixed-initiative frameworks also extend beyond visual aesthetics: Baba is Y'all enables collaborative puzzle level design by combining evolutionary search with player constraints [8]. Furthermore, Alvarez et al.'s Interactive Constrained MAP-Elites also explores co-creative content generation, allowing designers to interactively shape feature dimensions and evaluate the expressiveness of generated levels [3].

More recently, in the rapidly developing realm of generative AI applications, approaches to enhancing level design practices are being swiftly overhauled — yet they remain sparsely disseminated and even more rarely evaluated scientifically [2]. Stable Diffusion [30] and ControlNet [43] allow conditioning on sketches, text prompts, or 3D blockouts, aligning generation more closely with designer intent. In industry, Tenitsky's Blender–Stable Diffusion workflow [4], Rendair AI's blockout-to-render service [29], and Krea Realtime's live sketch-to-render interface [22] illustrate this shift, though most still provide limited control over individual elements. Other platforms, such as PromeAI [28] and Typus.AI [38], provide similar sketch-to-render capabilities, sometimes with integration into existing design tools.

Alongside these industrial workflows, academic research also explores diffusion-based generation for interactive level and layout design. Schrum et al. [32] present text-to-level diffusion models for Super Mario Bros., which allow designers to build longer, playable levels by piecing together generated segments. VIPCGRL [6] captures designer intent by unifying text, sketches, and layouts in a shared representation (building on top of the aforementioned IPCGRL). Cho's Stable Diffusion Tools for Unreal Engine [11] enable text- and image-prompt generation directly in the editor but offer little geometric control. Similarly, Gnatyuk et al.'s Smart Brush [15] combines Stable Diffusion with ControlNet to inpaint terrain and textures on large 3D maps, focusing mainly on the iterative refinement of texturing. Chen et al. [9] propose a Unity workflow that transforms generated imagery into traversable 3D environments, while CinePreGen [10] uses diffusion with depth and pose data for cinematic pre-visualization. Surveys on asset generation [14] highlight that while diffusion workflows excel in aesthetics, controllability, and fidelity to whitebox geometry remain underexplored. While focusing on 2D layouts, Pfau et al. compared the efficacy of generating levels through LLMs or diffusion models - before transforming them to shippable level imagery through a custom, diffusion-driven pipeline [25]. Although not situated in game design, He et al. [17] explore yet another angle by applying a similar logic to urban planning. Their approach uses a stepwise, human-in-the-loop process where diffusion models help at different stages, from laying out roads to generating detailed buildings. What these studies have in common is a move toward more interactive and co-creative workflows, though they do not focus on the specific challenges of early-stage 3D whiteboxing.

While these systems focus on producing final renders, our approach differs by: (1) enabling selection and generation of individual scene components, (2) generating these results directly from Unity, (3) utilizing specialized LoRA[1] models that maintain whitebox-level fidelity, and (4) running on open-source models locally, ensuring full control over the generation process. This positions our system as both a design exploration tool and a final rendering solution, offering greater artistic control and workflow integration compared to existing cloud-based alternatives.

[1]Lightweight fine-tunes that adapt large diffusion models using small, low-rank weight updates rather than retraining the entire network. Originally proposed for language models [18], LoRA has since been adapted to image diffusion systems, where it enables efficient fine-tuning and rapid personalization (see e.g., DreamBooth [31] as a contrasting full fine-tuning approach).

To evaluate the effectiveness of generative AI workflows in level design and to test whether our solution can augment established practices and out-of-the-box AI, we pose the following research questions:

- **RQ1:** What is the impact on **task efficiency** when embedding visual generative AI into level design workflows?
- **RQ2:** To what extent is users' **perceived control** affected by visual generative AI assistance in level design?
- **RQ3:** How well can level designers express their **visual imagination** using visual generative AI?
- **RQ4:** What are level designers' required features and capabilities for such systems?

3 Approach

In this section, we outline our conceptual and technical approach to embed visual generative AI into level design while maintaining creative control as much as possible – accessible as open source[2].

3.1 Pipeline

We propose a visualization pipeline which consists of three main elements: The **front-end**, which is used by the level designer to interface with the pipeline, extract contextual information from the level, and provide the resulting visual imagery; The **relay**, which serves as a medium between any front-end and back-end using a many-to-many relationship; And finally, the **back-end**, which is a rendering service capable of visualizing 2D images using the contextual information provided by any front-end.

For each of the three elements, we defined a set of requirements that must be met for the element to be deemed viable. In the following sections, we will discuss each element in detail, including the requirements from a high-level perspective.

3.1.1 Front-end. The front-end can be incorporated into any tool or game engine usable for the early phases of level design and prototyping, but is only deemed viable given that the following requirements are met:

(1) Capability to adhere user-modifiable textual prompts to 3D objects residing in the scene or level of the tool/game engine.
(2) Capability to extract object contextual information from 3D objects residing in the scene or level of the tool/game engine.
(3) Capability to have the user specify and modify global contextual information in the tool or game engine.
(4) Capability to display 2D visual imagery provided by the front-end in the tool or game engine.

3.1.2 Relay. The relay serves as a dynamic intermediary that manages image generation requests between any front-end and any available back-end. Its primary function is to allocate requests from any front-end user to an appropriate and available back-end service, ensuring scalability and efficient resource utilization.

3.1.3 Back-end. The back-end is a rendering service that facilitates the guided generation of images through accompanying contextual information. We keep it similarly tool-agnostic, as it can be any service that generates imagery if it meets the following requirements:

[2]https://github.com/arthurb123/UnitySDCN

(1) Capability to dictate or influence the output of any self-generated image given varying contextual information.
(2) Capability to indicate whether the back-end is currently busy or is available to accept new work.
(3) Capability to serve any generated image on request from an external service.

3.2 Contextual Information

The contextual information consists of two main elements, the global context and the object context.

3.2.1 Global Context. The global context contains miscellaneous information about the desired scene, besides rendering settings. The global context must contain the following information:

(1) **Background Description**
The background description is a textual description of desirable aspects, which should be applied to anything that is not a 3D object in the scene.
(2) **Negative Description**
The negative description is a textual description of undesirable aspects, which should be heeded during rendering, and not be visible in any resulting image.
(3) **Rendering Settings**
A set of generic rendering guidelines, primarily the image resolution in pixels.

The global context is a uniform form of data that is consistent across the entire pipeline. This can be guaranteed because the content of the global context is and should always be tool-agnostic. The only requirement the global context imposes on a front-end is that it is a mutable form of data, modifiable by the user.

3.2.2 Object Context. The object context contains information on all 3D objects that desire to be rendered. Any 3D object that should not be rendered must be excluded from the object context. The object context must contain the following information, regardless of the form in which it is presented:

(1) **Geometric information**
- Shape of the 3D object in the view of the image.
(2) **Spatial information**
- Position of the 3D object in the view of the image.
(3) **Orientational information**
- Rotation of the 3D object in the view of the image.
(4) **Descriptive information**
- Textual description of the 3D object adhered by the user.

3.3 Implementation

For the following implementation, a uniform format of the object context will be used. We opted for placing our evaluation into Unity as one of the most popular engines for game development, and using Stable Diffusion (the *Juggernaut XL* [20] checkpoint) in the backend, as at the time of evaluation, this allowed for the highest amount of supported modalities for controllability through ControlNet [43] (e.g., depth [40], normal [13], canny maps, and segmentation). Our generations feature a common configuration of 20 generation steps, a CFG of 5, a denoise factor of 1, and use the *dpmpp_2m_sde* sampler.

3.4 Front-end

For the implementation of the front-end, a custom Unity plugin available for any *Universal Render Pipeline* (URP) project starting from Unity version 2022.3.50f1 has been developed.

3.4.1 Extracting Object Context. To extract the required object context, we generate three types of images: segmentation maps, depth maps, and normal maps. These are captured using a custom component attached to the camera in the scene and together provide a comprehensive representation of each object's spatial, geometric, and orientational information.

Segmentation Map
A segmentation map is a binary image that represents each object in the scene as a 2D silhouette (see Figure 2). For every object that needs to be rendered, an individual segmentation map is generated, where:

- **White pixels (1)** indicate areas occupied by the object.
- **Black pixels (0)** indicate areas where the object is absent.

We end up with N segmentation maps, for $N = |S|$ where S is the set of objects that desire to be rendered.

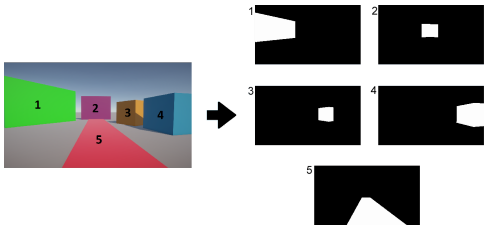


Figure 2: Example of produced segmentation maps ($N = 5$).

The segmentation map provides spatial information, indicating the object's location within the camera view. However, since this is a 2D projection, the object's depth/orientation are lost, requiring additional data from the depth/normal maps, such that we can represent the object's geometrical and orientational information.

Depth Map
A depth map is a greyscale image where pixel values represent the distance from the camera (see Figure 3). This is achieved by rendering the scene's depth buffer to a floating-point texture in the range $[0, 1]$:

- **White (1.0)**: the closest point(s) to the camera.
- **Black (0.0)**: the farthest point(s) from the camera.

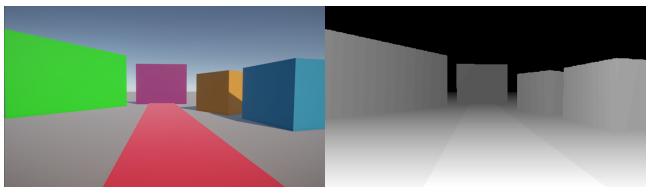


Figure 3: Example of a produced depth map (to the right).

Since the depth map covers the entire scene, combining it with the segmentation map allows us to determine depth values for

individual objects by comparing their pixels and checking which pixels are turned on (1) in the segmentation map. This provides a geometric representation of objects within the scene, as we now have the depth information of each object.

Normal Map

A normal map is an RGB image (see Figure 4), where the colour channels encode surface orientation in relation to the camera view:

$$Red(R) \rightarrow \hat{x}, Green(G) \rightarrow \hat{y}, Blue(B) \rightarrow \hat{z}$$

This is achieved by computing the normal vectors of all visible surfaces in the scene and storing them as colour values in a texture using a custom shader. Once again, by using the segmentation map to isolate object pixels, we can extract orientational information, describing the object's surface angles and facing direction.

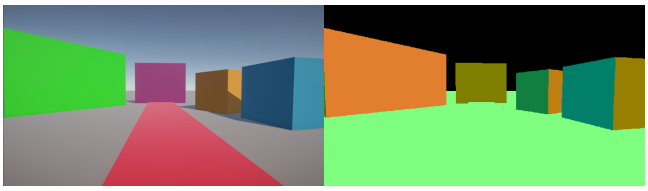


Figure 4: Example of a produced normal map (to the right).

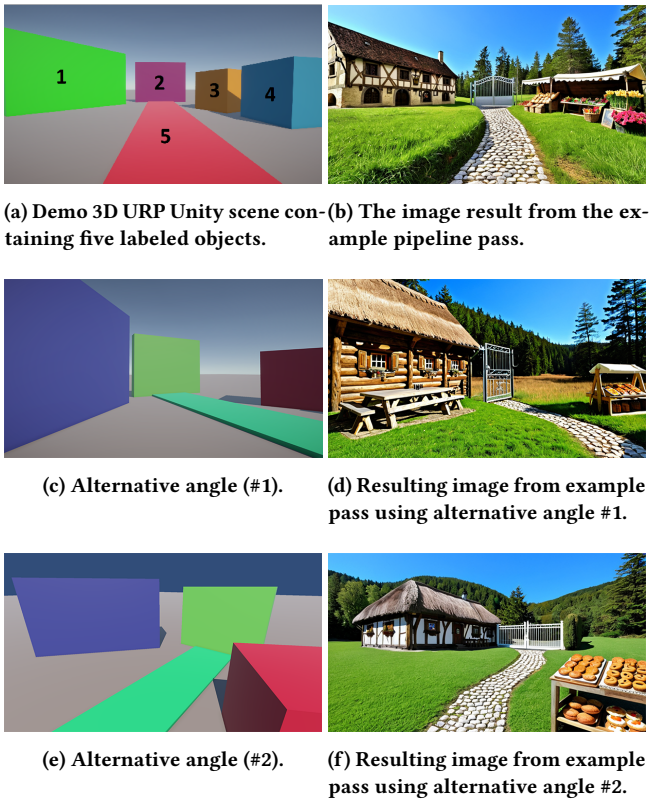
3.5 Back-end

For our back-end we used NodeJS [24] as the web server which handles requests from the front-end, while using ComfyUI [12] as our rendering service – relying on Stable Diffusion [30] as the text-to-image model combined with ControlNet [43] for the capability of using guided image generation.

3.5.1 Guided Image Generation. The back-end has the capability to dictate or influence the output of any generated image with respect to the contextual information as provided by the front-end. Using the customized ComfyUI plugin *ComfyUI Tooling Nodes* [1], we converted the segmentation maps into *masks* to create *regions* for the objects. These regions also take *conditioning* as input, adhering the object's descriptive information to the respective area of the latent image as defined by the mask. On top of this, through the usage of ControlNet, we differentiate between positive and negative conditioning nodes, and combine normal (see Figure 4) and depth maps (see Figure 3) as input to temporarily influence the conditioning. By doing so, the conditioning gets transformed to take the geometrical and orientational information of the scene into account when rendering an image. Combined with the region-based conditioning, as mentioned above, the back-end is capable of providing the required guided image generation.

3.6 Example Pass

We start off with an example Unity scene containing five objects, each object adhered a unique label (cf. Figure 5a). Once the back-end receives a generation request, a dynamic ComfyUI workflow will be created. Using the described process, a pipeline pass using the example scene results in the generated image shown in Figure 5b. Simply by moving our camera we can generate images from other angles (see Figures 5c and 5e); some example passes from different camera angles can be seen in Figures 5d and 5f.



(a) Demo 3D URP Unity scene containing five labeled objects.

(b) The image result from the example pipeline pass.

(c) Alternative angle (#1).

(d) Resulting image from example pass using alternative angle #1.

(e) Alternative angle (#2).

(f) Resulting image from example pass using alternative angle #2.

Figure 5: Comparison of the Unity scene, pipeline output, and alternative results. The objects were labeled as (1) "a wooden tavern", (2) "a metal gate", (3) "a wooden market stall selling baked goods", (4) "a wooden market stall selling flowers", and (5) "cobblestone path".

4 Evaluation

Using the just implemented pipeline, we evaluate the efficiency, perceived control, visual expressivity and requirements of a target group of level designers.

4.1 Measures

Several types of data were collected during the evaluation. First, a **pre-test questionnaire** (see appendix) gathered demographic information, educational background, and prior experience with level design, game engines, and generative AI tools. These items included both open-ended questions and rating scales (0–100) assessing experience with level design, white boxing, set dressing, game engine use, and the prompting of generative AI.

Second, **task performance data** was recorded. Participants performed three design tasks (manual drawing, pipeline, and bare stable diffusion), and the wall-clock duration of each task and subtask was measured in seconds using a timer. Completion times for both the initial (P1) and revised (P2) level descriptions were stored for later analysis.

Third, a series of **intermediary questionnaires** were administered after each tool was used. These included Likert-type ratings

(0–100) of perceived control over the outcome, ability to express visual imagination, and ease of use, as well as open-ended reflections on bottlenecks and frustrations.

Eventually, a **final questionnaire** was administered after both tools were used. This required participants to rank the three approaches (drawing, bare stable diffusion, our pipeline) on perceived speed and required effort, and to provide ratings of ease of use. Open-ended questions asked participants to reflect on the extent to which each method enabled them to translate their visual imagination into a design, and to suggest possible improvements.

Summing this all up, the measures consisted of (a) background and prior experience, (b) recorded task durations, and (c) repeated self-reporting of perceived efficiency, creative expression, control, ease of use, besides qualitative feedback.

4.2 Procedure

We recruited ($n = 20$) participants and began with asking them to fill in a pre-test questionnaire to participate. This pre-test questionnaire was used to assess their prior experience with level design, (the prompting of) generative AI and game engines – besides recording some demographic information. After completing the pre-test questionnaire, the participants were presented with a printed-out rough map of a level (presented from five viewing angles, see Figure 6), and a paragraph of text describing a level:

> **Level Description #1.** *A wooden tavern with three windows sits to the left of a dirt path. To the right of the dirt path sit two market stalls, the most front one selling flowers and the most back one selling baked goods. The dirt path meets a cobblestone gate near the back of the level. The settlement is located in a forest, with a grass floor.*

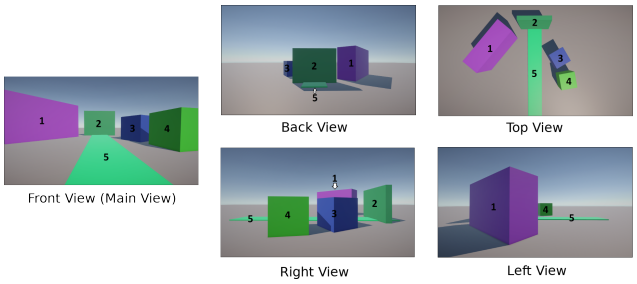


Figure 6: The rough map, presented from five different viewing angles to the participants (1 = tavern, 2 = gate, 3 & 4= market stalls, 5 = path).

The participants were first asked to create a *rough sketch* of their visualized level design based on the rough map and the text description, using a pencil and paper. As this is the method of most conceivable control, the created sketch serves as a baseline against which the pipeline and bare Stable Diffusion will be assessed. For all tasks, the front view was specified as the main view.
After drawing, using a latin-square approach, N participants were alternately and evenly (for $N = 20$, $N_A = 10$ and $N_B = 10$) assigned to one of two sequences: (A) using bare Stable Diffusion first, and the pipeline second; or (B) using the pipeline first, and bare Stable Diffusion second. For both tools, the participants were asked to generate a level design based on the same rough map and text description as before. No familiarization phase was administered prior to the main tasks; potential learning effects are acknowledged in Section 6.1. For both the pipeline and bare Stable Diffusion, two phases were executed. The first phase required the participant to recreate their level design using the current tool with the sketch as a control and reference. For the pipeline condition, participants were presented with a pre-white boxed scene in Unity matching the printed level map. After this phase, a second phase presented the participant with a slightly altered level description, which required them to adapt their level design. This phase was included to assess the flexibility of the tools. For both tools and both phases, participants were free to iteratively refine their text prompts across multiple generation attempts; no constraint was placed on prompt formulation or the number of revisions.

> **Level Description #2.** *A wooden tavern with three windows sits to the right of a cobblestone path. To the left of the cobblestone path sit two market stalls, the most front one selling flowers and the most back one selling baked goods. The cobblestone path meets a metal gate near the back of the level. The settlement is located in a forest, with a grass floor.*
> *(All red words are changed from the first description.)*

In between tools, the participants were asked to fill in an intermediary questionnaire, which required them to compare their perceived efficiency and creative output comparing the first used tool with the initial manual approach of drawing. After both tools were used, a final questionnaire was presented to the participants, which required them to compare the two tools with each other, and to rate their perceived efficiency and creative output.

(a) Manual sketches **(b) Bare SD** **(c) Our approach**

Figure 7: Three example outcomes for each condition, in no particular order or selection criteria.

4.3 Participants

20 participants have participated in the evaluation, which were recruited from the *Game & Media Technology* Master of Science program at *Utrecht University* (11) and the *Game Designer* and *Game Artist* Bachelor of Arts programs at *the Hogeschool voor de Kunsten Utrecht* (9). All of the participants had a background in game and/or

level design. Following the pre-test questionnaire, the participants featured in this evaluation had a mean level design experience of $M = 44.0 \pm 22.4$ (scale of $0 - 100$). Furthermore, the participants' experience with white boxing (M=34.2 ± 33.4) and with set dressing (M=23.15 ± 24.1) was rated as low-moderate and rather low respectively, suggesting a possible low familiarity with the practical aspects of level design. The participants rated their game engine experience (M=46.25 ± 24.5) and experience with generative AI (M=47.85 ± 28) as moderate. While most of these experience values are rather moderate, the high standard deviations indicate participants with rather high and low experience in the sample, which suits our evaluation with regards of assessing needs and attitudes of novices up to advanced users that all are interested in level design.

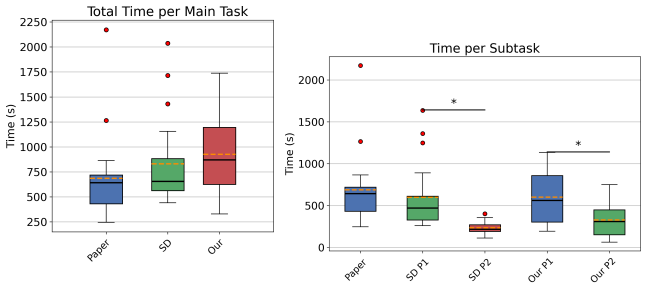
5 Results

Figure 7 outlines sample outcomes of the manual, bare Stable Diffusion (SD) and our approach. The time taken for each main task is shown in the boxplot in Figure 8a, while the time taken for each subtask (P1 being the first level description, P2 being the second level description) is shown in the boxplot in Figure 8b.

Paired-samples t-tests were performed to evaluate any significant differences between the times taken. No significant difference emerged between our approach and bare SD when comparing the overall completion times of the main tasks ($p = 0.43$), neither specifically for the initial iteration (P1: $p = 0.98$). However, comparing between the two design iterations with regards to our approach (P1 & P2), a significant difference was found ($t(19) = 3.82, p < 0.05, d = 1.02$). The same trend was observed comparing the two design iterations for bare SD, for $t(19) = 4.58, p < 0.05, d = 1.29$. There was no statistical significance between bare SD and our approach for the second iteration (P2: $p = 0.07$).

Table 1: Participants' reported perception of being in *control* and expressing *visual* imagination per condition.

Condition	$M_{Control}$	$SD_{Control}$	M_{Visual}	SD_{Visual}
Paper	**81.7**	±23.54	**64.55**	±23.18
SD	26.95	±15.95	39.9	±24.67
Our	**66.9**	±16.66	**71.8**	±14.68



(a) Time taken per main task.

(b) Time taken per subtask.

Figure 8: Task completion times. Boxes indicate the interquartile range (±1.5 around the median (–)), mean (- -), outliers (•), and significant differences (*).

Results for the participants' perceived feeling of being in control and perceived ability to express their visual imagination are reported in Table 1, and visualized in Figure 9. Performing paired-samples t-tests revealed significant differences in the perceived feeling of being in control and expressing visual imagination between the manual drawing and both AI-based approaches. Participants reported significantly greater feelings of control with both manual drawing ($t(19) = 7.98, p < 0.05, d = 2.72$) and our approach ($t(19) = 14.59, p < 0.05, d = 2.45$) compared to the bare SD approach, with large effect sizes. The difference between manual drawing and our approach did not reach statistical significance ($p > 0.05$). The ability to express visual imagination was rated significantly higher for both the manual approach of drawing ($t(19) = 3.07, p < 0.05, d = 1.03$) and our approach ($t(19) = 5.97, p < 0.05, d = 1.57$) compared to bare SD. Again, no significant difference was found between paper and our approach for expressing visual imagination ($p = 0.32$).

We also compared the prior experience with the (prompting of) generative AI and ability to express visual imagination, to assess if people with more or with less experience benefit more from our approach (as compared as to standard generative AI approaches). The (Pearson) correlation between prior experience with generative AI and expressing visual imagination was found to be $r = 0.46$, $p < 0.05, d = 1.03$ for bare SD and $r = 0.30, p = 0.20, d = 0.63$ for our approach, which indicates a less steep learning curve or required expertise to arrive at the desired imagery.

Complementary to the objective measures, we also asked participants to rank the three approaches in terms of perceived **speed** ($M_{Paper} = 1.9 \pm .91, M_{SD} = 2.5 \pm .76, M_{Our} = 1.6 \pm .5$), and perceived **effort** ($M_{Paper} = 1.85 \pm .75, M_{SD} = 2.4 \pm .82, M_{Our} = 1.75 \pm .79$) – where 1 is the fastest or least effort, and 3 is the slowest or most effort. Regarding speed, a Friedman test revealed a statistically significant difference among the conditions ($\chi^2(2) = 8.40, p < 0.05$). Post-hoc Wilcoxon signed-rank tests with Bonferroni correction showed that participants perceived our approach as significantly faster than bare SD ($p < 0.05, d = 1.77$). No significant difference was found between manual drawing and our approach ($p = 0.77$), nor between manual drawing and bare Stable Diffusion ($p = 0.28$). In terms of required effort, a similar Friedman test showed no significant differences ($\chi^2(2) = 4.90, p = 0.09$). When asked about perceived ease of use between bare SD and our approach (on a scale of 1-100), no significant difference appeared ($p = 0.79$, $M_{SD} = 75.5 \pm 21.6$, $M_{Our} = 76.3 \pm 17.7$).

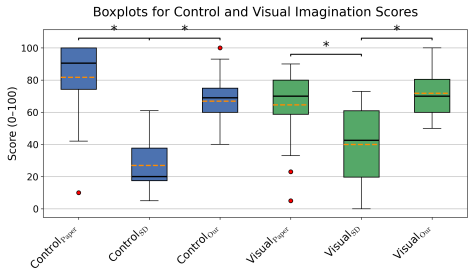


Figure 9: In control and visual expression scores.

On the qualitative data, an inductive thematic analysis was conducted following Braun and Clarke's (2006) six-phase framework [7]. All open questions from the questionnaires were read multiple times and codes were generated manually during the readthroughs. Themes were constructed based on recurring patterns across participants' responses (included in Appendix Table 2).

6 Discussion

In the following, we discuss the outcomes of Section 5, in the context of the initially posed research questions.

6.1 Task Efficiency

One goal of the evaluation was to determine to what extent visual generative AI, and especially our tailored approach, improves the *efficiency* of early-stage level design. To address this, we ground our reasoning on the recorded wall-clock time and perceived effort per successful design iteration; Where a successful design iteration is defined as a design that meets the requirements of the level description, and is deemed as satisfactory by the designer.

6.1.1 Task Timing and Iteration. The completion times of the first design task (P1) did not significantly differ between manual drawing, our approach, and bare Stable Diffusion. However, in the second iteration (P2), which required revising the existing level design due to slight changes in the level description, both our approach and bare SD were completed significantly faster than in the first round (P1) with large effect sizes. This improvement in speed may be partially due to participants' growing familiarity with the task and tools, in addition to any possible tool-specific advantages.

Participants' responses (see Table 2) revealed themes of lack of control over generated outputs (19/20) and objects not being rendered (10/20) while using bare SD. As a result, an overall observation was that participants were quick to accept lower quality and incomplete (thus unsuccessful) level designs compared to the first iteration (P1) out of frustration, which could have led to artificially lower time measurements. The qualitative feedback from participants underlines that they often felt that bare SD did not meet their expectations, highlighting the tendency to accept lower quality outputs more readily than with our approach. When asked what bottlenecks or frustrations the participants experienced during the usage of bare SD, some reported opinions along the lines of this example:

(P3). *"I got nothing close to what I asked for. It was really frustrating to have an idea of the picture in my head with the description in front of you, but when rendered it looked nothing like the idea I had in my head."*

As for our pipeline, although participants also outed themes of lack of control over generated outputs (15/20) and objects being merged or missing (6/20), participants generally expressed a theme of improved spatial control (16/20) throughout the design process compared to bare SD. Participants were able to position objects more effectively and thus felt more in control over the level design process, which likely contributed to the observed reduction in completion time for the second iteration (P2).

Since no significant difference in wall-clock time was found between manual drawing and our approach during the initial design task (P1), both approaches required a similar mean amount of time to complete the initial design. However, for the second level design (P2), which involved revising the design based on a changed level description, our approach showed a clear reduction in completion time with a large effect size. This indicates not only statistical significance but also a substantial practical advantage in iterative tasks when using our pipeline, and is underlined by the significantly faster perceived speed as per participants' rankings. Partly, this might reflect a general learning or familiarization effect with our tool, which is fair to assume, as in industrial practice, onset novelty effects would not be of practical consideration anyway.

6.1.2 Effort and Ease of Use. There was no statistically significant difference in perceived effort between the approaches. Despite this, participants' qualitative feedback often contained themes of missing quality-of-life features in our approach, such as undo/redo functionality (10/20) and copy/paste functionality (7/20), among others. This suggests that participants recognized pain points of our initial version of the tool, which could be addressed in future iterations, potentially lowering the perceived effort.

Ease of use ratings support this interpretation. Participants rated our approach as similarly easy to use, compared to bare SD, with both achieving high mean scores. Therefore, targeted improvements to the user experience of our approach could potentially enhance ease of use and perceived effort without sacrificing its advantages.

6.1.3 Observations on Manual Drawing. One recurring theme in participant feedback, regarding the manual approach of drawing, was that its effectiveness depended heavily on individual drawing ability. Participants frequently reported themes such as a lack of drawing skill (13/20), difficulties with perspective (8/20) and a limited ability to revise (5/20) in their responses. When asked what bottlenecks or frustrations they experienced during the manual drawing task, they uttered statements like:

(P8). *"As someone who is not good at drawing/drawing I struggled to replicate what I had envisioned in my head. Due to this, I often erase, then redraw, then erase again, wasting a lot of time for what should be a simple sketch. Perspective is very tricky to draw correctly."*

These constraints may not matter in all contexts, but they do become relevant in iterative workflows where quick prototypes, rapid changes and visual clarity are important.

- **RQ1:** What is the impact on **task efficiency** when embedding visual generative AI into level design workflows?

Towards **RQ1**, these findings partially support the hypothesis that our approach can improve early-stage efficiency, especially in iterative workflows. Notably, we cannot conclude the same for the bare application of visual generative AI, which often did not reach satisfying results, or would require excessive amounts of generations before a compelling outcome would appear by chance. While initial task completion times were comparable across all three methods, the observed reduction in time during the second iteration suggests that the pipeline may be particularly valuable in scenarios which require quick revisions. While the perceived effort did not yield statistically conclusive results, the observed improvements in time and perceived speed align with the goals of early-level design, where speed through iteration is essential. Further research with larger samples and more complex, iterative

tasks is recommended to strengthen these findings and to better assess the pipeline's impact in real-world industry settings.

6.2 Creative Control

The second core aim of this evaluation was to assess the extent to which visual generative AI, and our approach in particular, can support and enhance *creative output* during early-stage level design. Creative output being defined here as the designer's ability to visually express their internal mental image, within the constraints of the task, measured in this case by the perceived feeling of being in control during the design process, and the perceived ability to express visual imagination.

6.2.1 Feeling of Control. Participants reported clear differences in how in-control they felt during the design process. Both manual drawing and our approach led to significantly higher feelings of control compared to bare SD with large effect sizes indicating substantial impacts. A straightforward reason for the advantage of our approach over SD was due to its support for spatial and object-level adjustments, as expressed by participants which outed themes of improved spatial awareness over bare SD (16/20). In contrast, the lack of such control in the bare SD setup might have led to unpredictable outcomes, making it harder for users to guide the design process and thus feel in control.

It is worth noting that both our approach and bare SD were used in two level design iterations (P1 & P2), while manual drawing was only used for the first level design iteration (P1). This additional exposure over two design iterations instead of only one, may have helped participants feel more familiar with the AI-based tools over time, potentially increasing their perceived control and ability to express their visual imagination during the second iteration. Yet, we assumed that the (inherent) drawing capabilities would not be significantly impacted by the study setup.

6.2.2 Expressing Visual Imagination. Significant differences between our approach and SD (as well as between manual drawing and SD) highlight that both our approach and drawing enabled participants to express their visual imagination more effectively than bare SD, which simultaneously suggests that generative AI is not an out-of-the-box improvement, but our pipeline yields a practically meaningful improvement comparable to full-control drawing.

Besides featuring the highest mean score, our approach also exhibited the lowest standard deviation (14.68), while manual drawing showed a significantly higher standard deviation (23.18). This suggests that participants' experiences with manual drawing were more varied compared to our approach, potentially reflecting individual differences. This can be supported by the finding that more than half of the participants (13/20) reported a lack of drawing skill as an obstacle to successfully translate their mental image into a sketch, as expressed for example by:

> **(P10).** *"What works well about the manual control [drawing] is things being exactly what you make them: you are 100% in control. However, I was hindered by my inability to bring certain imaginations to life due to lack of drawing skill."*

This indicates that the efficacy of conveying creative ideas is (as hypothesized) very dependent on an individual's drawing skills in this situation. To evaluate whether tools of visual generative AI could reduce this gap, and make the way from ideation to creation more accessible, we next investigated if these conditions were so dependent on an individual's expertise as well. The prior experience with generative AI was found to significantly and substantially influence *visual expression* scores in the bare SD condition, but not for our approach. This suggests that our approach might allow a broader audience to realize creative ideas in this context. Above that, no statistically significant correlation was found between prior experience and the *perceived control* for our approach, which might underline its ease of use. In addition, and regardless of their prior prompting experience, participants reported being significantly more able to express their visual imagination through our approach (compared to bare SD). Thus, while additional experience might enhance usability and the overall perceived feeling of control, we follow that our approach allowed users to express their visual imagination without requiring and relying on extensive prior knowledge on the prompting of generative AI.

- **RQ2:** To what extent is users' **perceived control** affected by visual generative AI assistance in level design?

With respect to **RQ2**, we accept that it is challenging to compete with the full control over the output when drawing freely. Yet, when considering this as a theoretical upper bar, the naïve use of visual generative AI such as Stable Diffusion has shown a drastic loss of control – while our solution of a configurable 3D pipeline was not significantly different from full-control drawing.

- **RQ3:** How well can level designers express their **visual imagination** using visual generative AI?

Towards **RQ3**, we conclude that while drawing remains a viable method for the expression of visual imagination (especially for those with a significant drawing skill), our approach has proven to be accessible for great creative expression. This is due to the reduced reliance on drawing skill compared to the manual approach of drawing, and the reduced reliance on (the prompting of) generative AI compared to the bare SD approach. Altogether, this potentially renders it a valuable addition to early-stage level design workflows, particularly in collaborative settings where multiple designers may have varying levels of drawing skill or generative AI expertise.

6.3 Designers' Desires

- **RQ4:** What are level designers required features and capabilities for such systems?

Judging from the qualitative responses, participants weighed advantages and drawbacks of all conditions. They were mostly agreeing on the idea that manually sketching ideas takes a lot of time and skill, while potentially losing considerable detail and accuracy. Tools that would make this step more accessible to a broad audience of creatives are desirable, yet "only" using out-of-the-box generative AI (such as SD) takes more control over content and structure away than it yields (aesthetic) benefits. While a number of quality of life features were criticized in our initial implementation of our approach, and a one-to-one mapping of ideas to canvas is still far fetched, the majority of participants agreed that it considerably improved control over the spatial structure of the desired outcome. We report the most frequently desired features in the appendix.

7 Limitations & Future Work

While we draw a number of promising conclusions out of the previous analyses, we also acknowledge a series of limitations and how to overcome those in the following.

Participants. The participants in this study were students with backgrounds in game development and level design, which aligned well with potential future users of our proposed approach. The chosen audience demonstrated a clear benefit from using our approach, particularly in scenarios that required rapid iteration or for users with a lack of drawing skills or low prior experience with (the prompting of) generative AI. This study provides valuable insights into how the pipeline can support less experienced or early-career developers and designers, a group that will likely make up a significant portion of the target audience in the coming years. While this served as a valuable demographic, further research should explore whether the observed benefits also extend to more experienced, professional level designers and game developers.

Drawing on Paper. It is important to note that while drawing manual sketches is not a standard practice for professional level design, it was deliberately chosen in this study as a low-barrier method for participants to visually communicate their ideas without requiring specialized software skills. Posing manual drawing as the first condition before the Latin square evaluation might have introduced some bias, but we explicitly opted for this as it presented the lowest barrier of entry to engage visual imagination in participants.

Our Approach. A common issue participants reported was the lack of quality of life features, such as the ability to undo changes (10/20), operate the camera in more ways (9/20), and copy and paste objects (7/20). This made it difficult for participants to quickly iterate on their designs, as they had to manually adjust each object instead of being able to undo or redo changes, or create new objects from scratch. We acknowledge that this was an artificial bias introduced by letting participants operate in a (simplified) 3D environment rendered in Unity, instead of in the Unity Editor directly. We made this deliberate choice to not overwhelm participants with the full complexity of working in the Unity Editor, but published our tool as an Editor plugin, so that more experienced designers can make use of the creative control while not refraining from Unity's polished user experience.

Generative AI. A common issue that was more often observed than explicitly reported (6/20) was that the image generation often merged separate, unique objects into one single entity, making it difficult to control or adjust individual elements. This issue arises from the latent space representations used by Stable Diffusion, where diffusion models tend to conflate elements that are either spatially close or semantically similar. Another issue, again more frequently observed than reported, was that objects would sometimes not be present in the final output (6/20), due to a limited resolution or lack of sufficient training data about objects/styles present in the model.

Future Work. One successive avenue for future work is to refine the current implementation of our approach and enhance its usability, including the features previously indicated. More importantly, we aim to better integrate concept artists into the level design workflow, in subsequent participatory development sessions.

To address issues like object merging and missing elements, future work could test higher resolution outputs or integrate an intermediary upscaling phase if rendering times allow for it. Many small alterations can be made to the current implementation and evaluated – such as trying different diffusion models, different ControlNet models, different rendering settings – or the usage of a wholly different rendering service for the back-end, instead of a (latent) diffusion-based approach. Eventually, multi-modal models and pipelines are promising to establish a connection between the mechanical and the artistic game design [27] – which renders us positive that we can close the gap between design and implementation even further in the future.

8 Conclusion

Especially in larger productions of video game development, the early phases of level design are labor-intensive tasks involving many differently skilled roles, which makes communicating and establishing coherent creative visions a formidable challenge. In this paper, we set out to investigate whether out-of-the-box visual generative AI (Stable Diffusion) can aid in this process, and propose a custom, designer-centric tool pipeline as a Unity plugin on top.

A mixed-methods user study of twenty participants with a background in game development and/or level design compared our approach to the traditional approach of drawing on paper and a basic txt2img Stable Diffusion workflow. Results showed that our approach enabled participants to quickly iterate on level designs, while being able to express their visual imagination more clearly than with the basic Stable Diffusion workflow – while also feeling a much stronger sense of control. Compared to traditional drawing on paper, our approach performed similarly in terms of perceived visual expression and control, without relying on drawing skill. This suggests that the tool can offer an accessible alternative to manual drawing, while avoiding some of the unpredictability and limitations of txt2img AI prompting, and without requiring or being reliant on extensive prompting skills.

Furthermore, both our pipeline and bare Stable Diffusion interfaces were equally easy to use for participants, suggesting that introducing spatial control features did not have an adverse effect on usability. Although effort rankings did not significantly differ between methods, insights from qualitative answers highlighted various obstacles for participants with the manual approach of drawing on paper, underscoring the possible practical advantages of AI-driven tools in creative prototyping contexts.

While the results are promising, the study also had limitations. All participants were students with a background in game or level design belonging to the target audience, but their prior experience was quite limited. Ideally, future studies would involve game developers and/or level designers stemming from the professional industry with overall more experience. Additionally, during evaluation, our approach lacked quality of life features like undo/redo and more flexible camera controls which could have affected the perceived usability. The underlying diffusion model also introduced visual inconsistencies, such as missing, merged or misplaced objects, indicating that there is still room for technical improvements as a whole.

Despite these limitations, this research provides a useful foundation for integrating generative AI into real-world level design

workflows. The proposed pipeline demonstrates potential for enhancing early-stage level design and prototyping by streamlining the white boxing process through spatial iteration. This eventually has the potential to support the set dressing phase as well, as it enables designers to quickly visualize and refine stylistic elements, all while maintaining creative freedom and spatial consistency.

References

[1] Acly and Arthur Baars. 2025. ComfyUI Tooling Nodes. https://github.com/arthurb123/comfyui-tooling-nodes. Accessed: 25 November 2024.

[2] Sultan A Alharthi. 2025. Generative AI in game design: Enhancing creativity or constraining innovation? *Journal of Intelligence* 13, 6 (2025), 60.

[3] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. 2022. Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions. *IEEE Transactions on Games* 14, 2 (June 2022), 202–211. doi:10.1109/tg.2020.3046133

[4] Anton Tenitsky. 2024. Combining 3D and AI: Blender & Stable Diffusion Workflow. https://www.udemy.com/course/combining-3d-and-ai-blender-stable-diffusion-workflow. Accessed: 6 September 2025.

[5] In-Chang Baek, Sung-Hyun Kim, Seo-Young Lee, Dong-Hyeon Kim, and Kyung-Joong Kim. 2025. IPCGRL: Language-Instructed Reinforcement Learning for Procedural Level Generation. arXiv:2503.12358 [cs.AI] https://arxiv.org/abs/2503.12358

[6] In-Chang Baek, Seoyoung Lee, Sung-Hyun Kim, Geumhwan Hwang, and KyungJoong Kim. 2025. Human-Aligned Procedural Level Generation Reinforcement Learning via Text-Level-Sketch Shared Representation. arXiv:2508.09860 [cs.AI] https://arxiv.org/abs/2508.09860

[7] Virginia Braun and Victoria Clarke and. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. arXiv:https://www.tandfonline.com/doi/pdf/10.1191/1478088706qp063oa doi:10.1191/1478088706qp063oa

[8] Megan Charity, Ahmed Khalifa, and Julian Togelius. 2020. Baba is Y'all: Collaborative Mixed-Initiative Level Design. 542–549. doi:10.1109/CoG47356.2020.9231807

[9] Yuanbo Chen, Yixiao Kang, Yukun Song, Cyrus Vachha, and Sining Huang. 2024. AI-Driven Stylization of 3D Environments. arXiv:2411.06067 [cs.CV] https://arxiv.org/abs/2411.06067

[10] Yiran Chen, Anyi Rao, Xuekun Jiang, Shishi Xiao, Ruiqing Ma, Zeyu Wang, Hui Xiong, and Bo Dai. 2024. Cinepregen: Camera controllable video previsualization via engine-powered diffusion. *arXiv preprint arXiv:2408.17424* (2024).

[11] Ok Hue CHO. 2023. A Study on Effective Unreal Interactive Level Implementation Using Stable Diffusion. *Journal of The Korean Society for Computer Game (KSCG)* 36, 2 (2023), 89–94. doi:10.22819/kscg.2023.36.2.010

[12] ComfyUI. 2023. ComfyUI. https://comfyui.org/. Accessed: 25 November 2024.

[13] Eugeoter. 2025. Noob SDXL ControlNet Normal. https://huggingface.co/Eugeoter/noob-sdxl-controlnet-normal. Accessed: 12 February 2025.

[14] Kaisei Fukaya, Damon Daylamani-Zad, and Harry Agius. 2025. Intelligent Generation of Graphical Game Assets: A Conceptual Framework and Systematic Review of the State of the Art. *Comput. Surveys* 57, 5 (Jan. 2025), 1–38. doi:10.1145/3708499

[15] Vitaly Gnatyuk, Valeriia Koriukina, Ilya Levoshevich, Pavel Nurminskiy, and Guenter Wallner. 2025. In the Blink of an Eye: Instant Game Map Editing using a Generative-AI Smart Brush. arXiv:2503.19793 [cs.CV] https://arxiv.org/abs/2503.19793

[16] Matthew Guzdial, Nicholas Liao, and Mark Riedl. 2018. Co-Creative Level Design via Machine Learning. arXiv:1809.09420 [cs.AI] https://arxiv.org/abs/1809.09420

[17] Mingyi He, Yuebing Liang, Shenhao Wang, Yunhan Zheng, Qingyi Wang, Dingyi Zhuang, Li Tian, and Jinhua Zhao. 2025. Generative AI for Urban Design: A Stepwise Approach Integrating Human Expertise with Multimodal Diffusion Models. arXiv:2505.24260 [cs.AI] https://arxiv.org/abs/2505.24260

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[19] Zehua Jiang, Sam Earle, Michael Cerny Green, and Julian Togelius. 2022. Learning Controllable 3D Level Generators. arXiv:2206.13623 [cs.AI] https://arxiv.org/abs/2206.13623

[20] KandooAI. 2025. Juggernaut XL Checkpoint. https://civitai.com/models/133005?modelVersionId=782002. Accessed: 12 February 2025, Version: juggXIByRundiffusion.

[21] Tobias Karlsson, Jenny Brusk, and Henrik Engström. 2023. Level Design Processes and Challenges: A Cross Section of Game Development. *Games and Culture* 18, 6 (2023), 821–849. arXiv:https://doi.org/10.1177/15554120221139229 doi:10.1177/15554120221139229

[22] Krea. 2024. Realtime. https://www.krea.ai/realtime. Accessed: 6 September 2025.

[23] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. 2013. Sentient Sketchbook: Computer-aided game level authoring. In *International Conference on Foundations of Digital Games*. https://api.semanticscholar.org/CorpusID:2103833

[24] NodeJS. 2009. NodeJS. https://nodejs.org/en. Accessed: 25 November 2024.

[25] Johannes Pfau, Ian Baverstock, Martin de Ronde, Michiel Mol, and Jasper Vis. 2026. Generative Remastering: LLM versus Diffusion-Based Level Generation, Target Style Realization, and Autonomous Augmentation. In *2026 IEEE Conference on Games (CoG)*. IEEE, 1–8.

[26] Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. 2020. The Case for Usable AI: What Industry Professionals make of Academic AI in Video Games. In *Extended abstracts of the 2020 annual symposium on computer-human interaction in play*. 330–334.

[27] Johannes Pfau and Panagiotis Vrettis. 2026. From LLM-Driven Trading Card Generation to Procedural Relatedness: A Pokémon Case Study. *arXiv preprint arXiv:2604.27972* (2026).

[28] PromeAI. 2024. PromeAI. https://www.promeai.pro/. Accessed: 6 September 2025.

[29] Rendair. 2024. Rendair. https://www.rendair.ai/. Accessed: 6 September 2025.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. *CoRR* abs/2112.10752 (2021). arXiv:2112.10752 https://arxiv.org/abs/2112.10752

[31] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. arXiv:2208.12242 [cs.CV] https://arxiv.org/abs/2208.12242

[32] Jacob Schrum, Olivia Kilday, Emilio Salas, Bess Hagan, and Reid Williams. 2025. Text-to-Level Diffusion Models With Various Text Encoders for Super Mario Bros. arXiv:2507.00184 [cs.LG] https://arxiv.org/abs/2507.00184

[33] Matthew Siper, Sam Earle, Zehua Jiang, Ahmed Khalifa, and Julian Togelius. 2023. Controllable Path of Destruction. arXiv:2305.18553 [cs.AI] https://arxiv.org/abs/2305.18553

[34] Adam Smith and Michael Mateas. 2011. Computational Caricatures: Probing the Game Design Process with AI. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 7 (10 2011), 14–18. doi:10.1609/aiide.v7i3.12478

[35] Gillian Smith, Jim Whitehead, and Michael Mateas. 2010. Tanagra: A mixed-initiative level design tool. (06 2010). doi:10.1145/1822348.1822376

[36] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. 2018. Procedural Content Generation via Machine Learning (PCGML). arXiv:1702.00539 [cs.AI] https://arxiv.org/abs/1702.00539

[37] The Level Design Book. 2024. The Level Design Book. https://book.leveldesignbook.com/process/env-art. Accessed: 9 September 2025.

[38] Typus.AI. 2024. Typus.AI. https://typus.ai/. Accessed: 6 September 2025.

[39] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam Smith, and Sebastian Risi. 2018. Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. arXiv:1805.00728 [cs.AI] https://arxiv.org/abs/1805.00728

[40] Xinsir. 2025. ControlNet Depth SDXL 1.0. https://huggingface.co/xinsir/controlnet-depth-sdxl-1.0. Accessed: 12 February 2025.

[41] Yahia Zakaria, Magda Fayek, and Mayada Hadhoud. 2023. Start small: Training controllable game level generators without training data by learning at multiple sizes. *Alexandria Engineering Journal* 72 (June 2023), 479–494. doi:10.1016/j.aej.2023.04.019

[42] Mar Zamorano, Carlos Cetina, and Federica Sarro. 2023. The Quest for Content: A Survey of Search-Based Procedural Content Generation for Video Games. arXiv:2311.04710 [cs.SE] https://arxiv.org/abs/2311.04710

[43] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543 [cs.CV] https://arxiv.org/abs/2302.05543

A Qualitative Themes

Table 2: Themes and occurrences from thematic analysis on the qualitative answers, sorted by approach and major theme.

Theme	Occurrences	Coding
Paper: Lack of skill	13	Struggles to represent mental image, perceives drawing ability as poor, frustration with drawing quality, reluctance to commit to details
Paper: Difficulty with perspective	8	Inconsistent spatial proportions, trouble placing objects in depth, avoids angled or 3D views
Paper: Difficulty iterating	5	Erasing and redrawing repeatedly, iteration seen as time-consuming, paper format limits experimentation
Paper: Difficulty with details	6	Omits finer textures or features, felt lacking time for intricate elements, detail added via text annotations
SD: Lack of control over output	19	Output lacks expected completeness, limited influence on object position and orientation, cannot fix single feature precisely
SD: Missing Objects	10	Specified objects excluded from result, parts of prompt not being recognized, objects replaced with others
Our Approach: Lack of control over output	15	Output lacks expected completeness, cannot finetune small details, lack of object-level locking
Our Approach: Improved spatial awareness	16	Object are more accurately positioned, layout matches mental image, immediate visual response to changes, support for viewpoint alteration
Our Approach: AI Merging Objects	6	Overlapping shapes collapse together, proximity affects render logic, semantic distinctions not preserved
Our Approach: AI Missing Objects	6	Small shapes ignored in output, detail prompts partially omitted
Our Approach: Missing QoL - Undo/Redo	10	Actions cannot be reversed, mistakes require rework, undo expected but absent
Our Approach: Missing QoL - Copy/Paste	7	Cannot reuse existing objects, rebuilding repeated structures manually
Our Approach: Missing QoL - Better Camera Movement	9	Camera movement feels slow or clunky, rotation controls unintuitive, difficult to reposition view efficiently, navigation slows iteration speed