CrawLLM: An LLM-Based Pipeline for Game Asset Generation

Marvin Zammit, Antonios Liapis *Member, IEEE*, Georgios N. Yannakakis *Fellow, IEEE*

***Abstract*—Procedural Content Generation (PCG) systems typically struggle to generate cohesive content across multiple domains. Large Language Models (LLMs) understand semantic relationships of game elements, at least when they are described in natural language. This paper investigates how LLMs and text-to-image models can generate narrative, visual, and gameplay content *coherently*. We leverage LLMs as a scaffold for an automated, theme-driven asset generation pipeline, enabling unique and scalable game experiences with no developer input besides setting up a game template. This paper introduces CrawLLM, a dungeon crawler with card combat mechanics, as a testbed for an LLM-driven game generation pipeline. A Mixtral 8x7B model generates game themes, guiding the creation of narrative content. Visual assets are produced via Stable Diffusion XL, using ControlNet and IP-Adapter modules to achieve game-ready formats. A user study conducted on snapshots of fully generated games indicates that the underlying semantic themes remain clearly discernible in many cases, although intended visual styles were less clear. This work demonstrates the potential of LLM-driven pipelines for PCG, while highlighting areas for improvement in content specificity.**

***Index Terms*—Procedural Content Generation, Image Generation, Narrative Generation, Large Language Models, Stable Diffusion, Automated Game Generation**

I. Introduction

Crafting content for digital games is a labour intensive process for several reasons. First, digital games are multifaceted and rely on narrative, visuals, audio, rules, levels, and more [1]; each of these facets requires its own skillset and mindset [2] to produce assets for. Secondly, such content requires technical processing, to format and optimise it in order to be usable by the game engine. Lastly, all of these assets must form a coherent ensemble that shapes and enhances the player experience [1]. Procedural content generation (PCG) for individual facets, such as levels, visuals or music, has been extensively explored [3]–[7]. However, it is very challenging to cross-evaluate artefacts from different facets; cross-modal [8] generation in games is under-researched [1]. Finally, human players have different aesthetics: no one set of game assets will be liked by all. PCG may provide a wider variety of content themes, or possibly personalised assets to match player preferences.

The advent of Large Language Models (LLMs) and their recent leap in performance and popularity makes them appealing for PCG. Their inherent instruction-following ability and high-level understanding of game elements and their functional interconnections position them as capable generators in several aspects of game development [9]. Their semantic input and output make them very accessible with little technical knowledge needed. However, using LLMs introduces some constraints in automated systems: for one, their outputs need to be parsed through consistent naming and formatting conventions by follow-up PCG components in a generative pipeline [1]. Similarly to LLMs, diffusion-based text-to-image generators—such as the Stable Diffusion [10] (SD) family—are able to generate images in a wide range of styles and high aesthetic quality. In conjunction with other models designed to condition SD output [11], [12], text-to-image generators are well-suited to generate visual assets for direct deployment within a game engine.

Inspired by recent advances in LLMs and text-to-image models, this paper introduces an LLM-driven pipeline capable of generating both textual and visual assets that are coherent with a theme—itself generated by the LLM. The pipeline produces narrative elements, characters, and locations, along with descriptions that drive text-to-image sub-processes for corresponding visuals. As a proof-of-concept, we developed a 2D dungeon crawler, CrawLLM, in which combat unfolds through card-based actions, demonstrating the generator's ability to support diverse visual design tasks.

This work can be viewed as an initial attempt at applying generative AI for automated 'cloning' or 're-skinning' of games [13]. Game clones (e.g. Diablo-clones) follow the original game fairly closely (mostly changing audiovisual assets), and are often considered "opportunistic and monetarily motivated" [14] when made by human creators. At the same time, the ambition of CrawLLM extends beyond cloning: by combining narrative generation with visual synthesis, the pipeline points toward a more general framework for holistic game generation [1]. Central to both cloning and game generation is the system's ability to maintain thematic coherence across narrative and visual elements. We evaluated this through a user study with 34 participants assessing the coherence of generated games to their overarching themes.

This paper contributes to PCG research in several ways:

1) It introduces CrawLLM, a novel dungeon crawler with card combat mechanics, which incorporates narrative elements, cards, character animations, and 2D tilesets.
2) It describes the CrawLLM generative pipeline, which generates new thematic elements to guide the generation of *game-ready* text and visual assets (see Fig. 1). This approach employs a fixed game template.
3) Asset generation is controlled by natural-language themes (Section III-B), enabling designers with minimal technical expertise to direct content, while developers can refine the underlying template. This yields controllable generation of games with consistent themes and loops.
4) It advances controllability and coherence in generative AI via ControlNet-guided visual refinement and a hierarchical multi-stage pipeline. Though demonstrated in CrawLLM,

This project has received funding from the Digital Technologies Programme project GameChanger (no: DTP-2025-18), funded by Xjenza Malta & MDIA.

M. Zammit, A. Liapis, G. N. Yannakakis are with Institute of Digital Games, University of Malta, Msida, Malta.

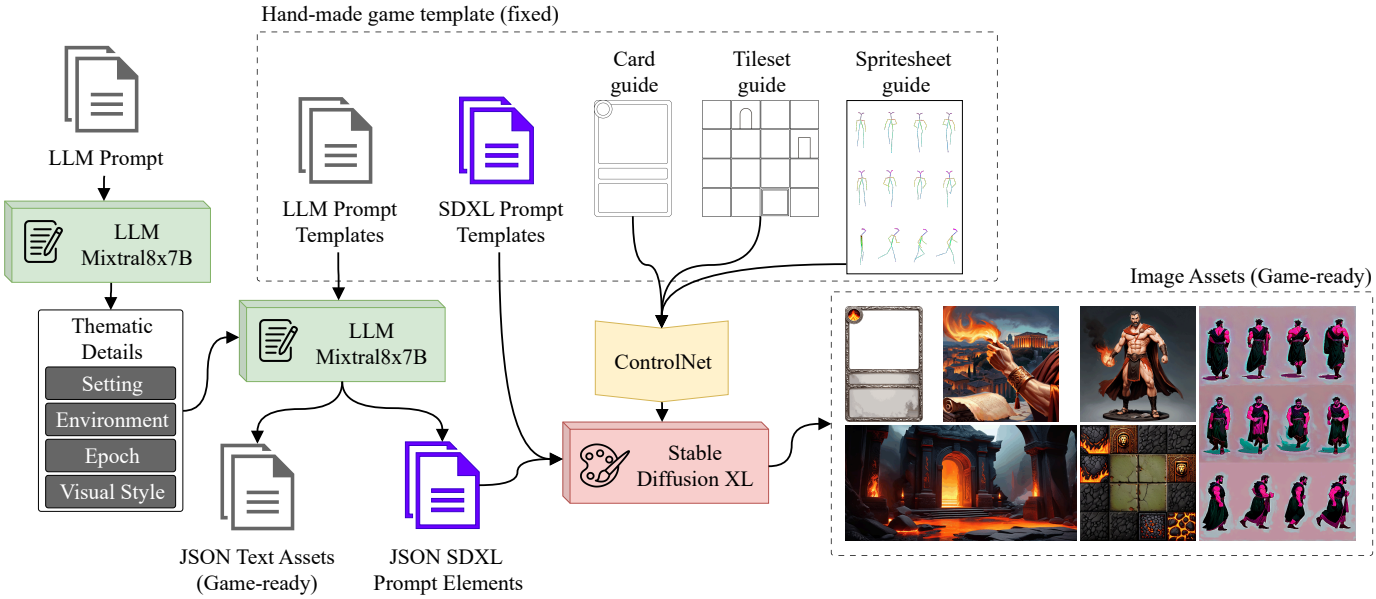


Fig. 1: The full CrawLLM pipeline, illustrating the predefined elements in the template created specifically for the game. The LLM generates both text assets for the game as well as prompts for image generation.

these methods generalise to diverse types of assets, including tilesets, animations, and narrative.

5) It proposes a new evaluation protocol in which participants infer the intended theme (against semantically related alternatives) based on generated game assets as integrated in the game.

II. Related Work

Since the focus of this paper is on leveraging recent advances in generative AI for procedural content generation, we review work on the above topics and their intersection.

A. Procedural Content Generation and Game Generation

PCG has traditionally targetted specific facets of games, such as levels [15]; while less popular, PCG for other facets such as game visuals [5], game audio [6], game narrative [16], and game rules [7] has also matured. However, combining generators of different facets is challenging [1]: establishing coherence between content of different modalities is non-trivial, and heuristics for such cross-modal evaluations have been historically sparse. A common way to address the generation of games is to either generate everything in one go, e.g. evolving both the game rules and the game board in Ludi [7], or to generate everything sequentially: the previous generative step constrains the possibility space of the next generative step. The latter is called a *generative pipeline* or a *top-down orchestration process* [1]. The generation of complete games has often been possible due to such a pipeline, e.g. starting from a human-provided concept—as simple as the protagonist's name [17], [18] or as complex as a concept relationship graph [19]—and generating individual game components one at a time. Other systems have used declarative logic programming to generate games that are guaranteed to be consistent with a desired interpretation [20]. In other cases, game generation occurred by generating different facets independently and tying them to a high concept, or leaving some aspect of the process (such as level generation) independent from the game concept [21].

As noted in Section I, the term *game generation* in PCG academia [15], [22] has often been associated with the automated discovery of new game mechanics—e.g. in platformer games [23], [24]. Early successes in game generation such as the abstract two-player board games generated by Ludi [7] may have influenced this association. However, we follow [1] and their more holistic view of games as a confluence of visuals, music, narrative, level- *and* game-design (e.g. rules) rather than only the latter. We argue that game generation software such as DATA Agent [25] can create unique player experiences by changing the characters, visuals and locations while keeping a pre-defined core game loop. CrawLLM takes this approach a step further, generating the general theme of each game and then a number of visual and text assets, used as proto-narrative [1], in a feed-forward generative pipeline controlled by modern generative AI methods.

Past game generation research has often focused on variants of existing games such as chess [26], [27] or game genres such as platformers [23], [24]. Their process is often approached through automatic mechanic discovery [7], [23], [24], [28]. Few examples instead focus on changing the player experience through textual and graphical assets, such as in early versions of the ANGELINA platformer game generator [21]. Dungeon crawlers have also been targeted extensively in PCG research [29], [30] from the perspective of level generation. CrawLLM adopts the approach of text and visual assets generation and applies them to a novel dungeon crawler with generated levels, adding card combat mechanics to assess the pipeline's ability to generate a diverse range of assets.

B. Gen AI: Large Language Models and Text-to-Image Models

Recent advances in Generative AI (Gen AI) exhibit potential in PCG. Cross-modal evaluation is now possible through deep learning models such as OpenAI's CLIP [31] or FLAVA [32], which are zero-shot classifiers mapping text and images to the same latent vector space. These models can evaluate coherence between images and text, but their applications can do even more. Text-to-image generative algorithms have seen a leap in quality, especially diffusion models [10] trained on large corpora of image sets. Stability AI's initiative to release its Stable Diffusion (SD) family of such models led to widespread use by both the scientific community and enthusiasts. ControlNet [11] was developed soon after, and offered better control over the generated image beyond the input text prompt. ControlNet works by replicating parts of the SD model's neural network while freezing the original network's gradient; this allows the replicated part to adapt to specific control input images (e.g., edge or depth maps). This ensures precise control without altering the core model's learned features. However, each image is individually generated and the model retains no awareness of prior inference iterations. This makes it difficult to produce multiple images of e.g. the same character or location, as required in a sequential visual narrative. An IP-Adapter [12] works with SD by conditioning the model with additional input from an image, helping it retain key visual elements (e.g style or structure) during inference.

Diffusion models have been used to generate game assets, such as 2D spritesheets for character animations [33]. However, the process involves manual intervention and multiple refinements, with little automated control on the generated poses themselves. Other work on an automated pipeline for character generation [34] required training of a pose-control model for the specific art style and character type, restricting its use in a more general game generation pipeline.

In this work we utilise the SDXL model [35] from the SD family, as it offers a wide selection of compatible ControlNet and IP-Adapter models which may be readily applied to it. Although larger and more performant text-to-image generation models are available, these do not all offer compatible conditioning models that are necessary for the proposed pipeline.

While text-to-image models seem able to address both the challenge of coherence evaluation and the challenge of visual asset generation, coherence can be ensured in more ways. LLMs understand relationships between semantic concepts and can generate text constrained by the instructions provided—also in text. LLMs are transformer-based auto-regressive models trained on extremely large corpora of tokenised text to predict the next token. OpenAI's ChatGPT family [36] has become predominant in research, but competing models have been released openly, such as Meta's LLAMA [37] family, and Mistral's own models [38]. These LLMs demonstrate a high level of understanding not just of language, but also of a broad range of subjects [39].

C. Gen AI and Procedural Content Generation

Gen AI approaches have already shown potential for use in games in multiple roles [9], [40], and have already been used for PCG in games [41], [42]. Prior research has explored how Gen AI models can create game assets [43]. In some cases, such as Genie [44], this has extended to generating entire games—including the underlying engine. Other work has focused on using LLMs to design levels and rules [42], [45]. StoryAgent [46] demonstrates a top-down approach: from a single-line prompt, an LLM constructs a hierarchical story structure (characters, scenes, timeline), which is then expanded in a bottom-up phase that generates and assembles assets such as text, images, and music. Unbounded [47] generates an 'infinite' sandbox life simulator based on a player-defined character appearance and personality; game mechanics, narratives, and environments are produced dynamically during play by an LLM fine-tuned on game data. At each step, new mechanics or narrative events are generated in response to players' textual input. Visual assets are produced through diffusion models, with consistency managed by IP-Adapters [12]. These approaches typically rely on either models trained explicitly for the task or manually integrating generated assets into the target engine.

LLMaker [48] is similar to our approach in combining LLMs, Stable Diffusion, and ControlNet to create a (side-view) dungeon crawler similar to *Darkest Dungeon* (Red Hook Studios, 2016). Unlike CrawLLM, however, LLMaker is a human-in-the-loop authoring tool rather than a fully automated generator of a playable game. In LLMaker, users guide the process through natural language, and the LLM translates their intent into appropriate generative function calls [48].

This paper is distinct from prior work in key ways. First, it employs pre-trained models without task-specific fine-tuning, enabling a zero-shot pipeline. Second, asset generation is fully automated, allowing direct incorporation of outputs into the game engine without human intervention. Finally, much of the literature has favoured large proprietary systems such as GPT-4 [36] for their strong performance; this paper demonstrates that smaller open-source models can also be effective, and can run locally on hardware that is more readily accessible.

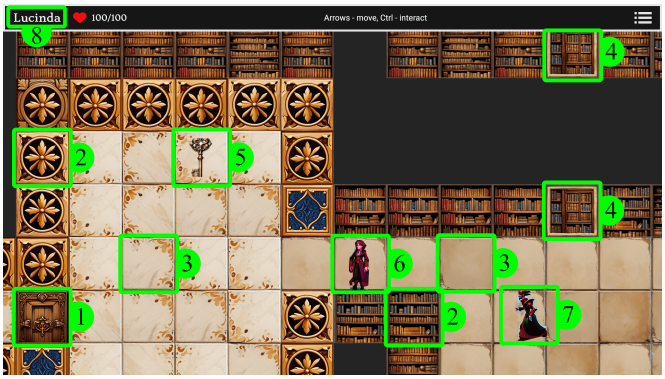
III. CrawLLM

CrawLLM is a procedurally generated dungeon crawler game developed in the Unity game engine.It features a fully defined game template (see Fig. 1) which specifies the player's experience as well as the game's expected assets and their format (described in Section III-A) and a generative pipeline for producing all the assets in a top-down, step-by-step fashion [1]. Specifically, LLMs produce the high-level descriptions of the entire game, which then guide and constrain follow-up generative steps to produce the user-facing text assets and visual assets.

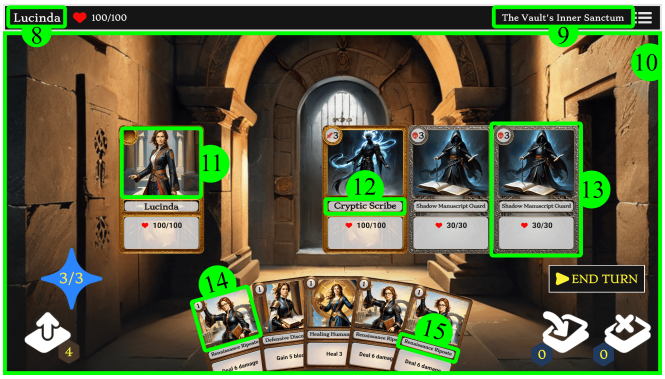
A. The Game

CrawLLM is a dungeon[1] crawler with top-down 2D visuals. The "dungeon" consists of rooms and corridors grouped into locations; up to 5 locations may exist within the dungeon. The

[1]We use the term "dungeon" here loosely based on the tropes of the genre, but apart from labyrinthine and fairly constrained structures, the "dungeon" in each game of CrawLLM might be anything—not necessarily a castle or medieval prison.



(a) Dungeon Crawling view



(b) Card Combat view

Fig. 2: Gameplay elements within a CrawLLM game session. Labelled elements are door (1), wall (2), floor (3), and secret door (4) tiles, key (5), PC (6), and enemy (7) sprites, PC (8), location (9), enemy (12), and action card (15) names, and location (10), PC (11), card frame (13), and action card (14) images. The game depicted is *"Luminary's Legacy: The Renascence Vault"*.

dungeon is generated anew on each playthrough (as described in Section III-C) but using text and visual assets generated beforehand (see Section III-B).

When the player starts the game, they receive a short message introducing their player character (PC), their situation and goal. At that point, the player moves their PC around the 2D map in an effort to reach the dungeon's exit and complete the game. To achieve this, the player has to pass through different locations (see Fig. 2a) which consist of corridors and rooms. Rooms can be separated with a door: doors may either be locked (requiring one or two keys to open) or they may close behind the PC, acting as a one-way portal (entering from the other side is impossible). Keys are found in the dungeon and picked up by colliding with them; there are no other pickup items. To increase the navigational challenge, secret doors may appear in some rooms; once discovered, they turn into regular corridors connecting two secret doors. Each location may have one type of enemy and two types of minions (tied to that enemy). The enemy appears in the final room of that location, while minions may be found in other rooms of this location. Enemies or minions roam within a room following simple movement patterns. If the player collides with a roaming opponent, they

TABLE I: Cards in CrawLLM, with their in-game effect; the original card title listed is subsequently adapted by the LLM.

Original title	Effect	Cost	Frame
Basic Attack	Deal 6 damage	1	Bronze
Fast Attack	Deal 3 damage	0	Bronze
Basic Block	Gain 5 block	1	Bronze
Basic Heal	Heal 3 HP	1	Bronze
Basic Draw	Gain 2 cards	0	Silver
Gain Mana	Gain 2 mana	0	Silver
Increase Max Health	Increase max HP by 5	2	Gold
Increase Strength	Attacks do 4 more damage	2	Gold
Life Steal	Deal 5 damage, heal 5 HP	2	Gold

enter a new view to conduct card-based combat.

The Card Combat view (see Fig. 2b) shows the PC and current enemies as cards (along with their hit points and other statuses) and the player's action cards. The enemy may be accompanied by minions during card combat (see Fig. 2b), even though those are not visible in the dungeon crawling view. The number of additional minions depends on the danger level of the room, which is controlled by the dungeon generator (see Section III-C). Combat is turn-based; the player chooses *action cards* up to their maximum man, while enemies take actions randomly from a pool of actions depending on their type. Victory in combat gives the player a new action card among three options. The initial player deck is predefined for all games, but the awards are randomly chosen from all possible cards encoded in the system. Cards' frame colours (bronze, silver or gold) hint at the power of the card. All cards' in-game effects and costs are predefined, and are listed in Table I; however, their titles and card art (including the card's frame) are generated according to the game's theme. Indicatively, in Fig. 2b the "Basic Attack" title of the card in Table I becomes "Renaissance Riposte" (see 15 in Fig. 2b).

The number of assets that must be generated for a complete game depend on the number of locations in the dungeon. Since in this paper the dungeon generator can add up to 5 locations in the level, the in-game text assets that must be generated include the names of 1 PC, 5 enemies, 10 minions and 9 action cards as well as 1 introductory message and 1 completion message. In terms of visual assets, those include 3 card frames (one per colour), 16 spritesheets and 16 portraits for card combat (1 PC, 5 enemies, 10 minions), 9 pieces of action card art, 5 card combat backgrounds, 45 wall tiles (9 per location), 20 floor tiles (4 per location), 20 key sprites (4 per location), 10 door tiles and 5 secret door tiles. All of these need to be externally coherent with the game's setting but also internally coherent in the context of their use (e.g. walls matching the floors of a location). We detail the generation of all these assets in Section III-B.

B. Generative pipeline

The scope of this paper was to implement a pipeline based on LLMs which can orchestrate [1] the generation of all necessary text and visual assets for CrawLLM, thereby re-theming it to different settings. To do this, we first generate overarching semantic information (see Section III-B1) that is not shown to the user directly but is used as additional prompts both to

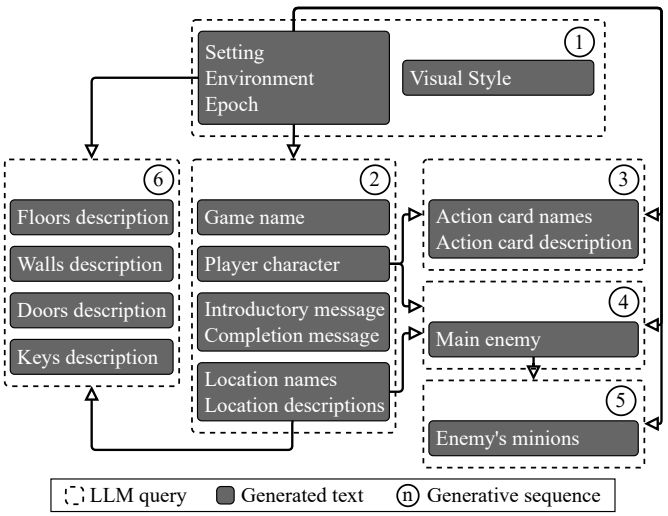


Fig. 3: The sequence of text generation via Mixtral 8x7B. Arrows indicate the reuse of generated text as part of the prompts in the following stages. Steps 4, 5, 6 are repeated for each location in the game (five times in the current paper).

control player-facing text (see Section III-B1) and visuals (see Section III-B2). Based on extensive testing, this study uses the Mixtral 8x7B model for all LLM-based text generation; the model runs locally on our machines through the Ollama[2] platform. The prompts used for both text and image generation are being made openly available on an online repository[3].

The overall structure of the CrawLLM generative pipeline is outlined in Figure 1.

1) Generation of semantic information: Thematic details are represented as text, and are generated first by the LLM in steps (see Fig. 3). As noted by [49], time and space are central to narrative design in games. Accordingly, three *thematic details* were defined, namely the *setting* (e.g., "A cursed carnival during a sandstorm"), the *environment* (e.g., "Tilted rides, ghost tents, dunes and ruins"), and the *epoch* (e.g., "Late 20th century"). To guide generation of visuals, a fourth thematic detail is the artistic *style* (e.g., "Watercolor, whimsical"). These four thematic details are all generated together by a single prompt, which outputs a JSON-formatted response.

The thematic details (except visual style) guide the generation of a number of text assets which are shown to the player—even if not in full. With appropriate prompt templates which are instantiated with the setting, environment and epoch details of this game, the LLM generates the game name, details on the player's character (their name, species, gender, and description), an introductory message shown to the user when the game starts, and a completion message shown when the player reaches the dungeon's exit. We also generate five appropriate locations (their name and description) which will be used to create location-appropriate details in next steps. It is worth noting that while the game name, introductory and completion messages are shown to the player in full, the PC's details and location descriptions are never shown to the player (only the PC's name and location name) but are necessary as prompts for the next generative steps.

The PC details, coupled with the thematic details (except visual style) are used in additional prompts for generating the action cards of the game; the details of each card as described in Table I are also provided in this prompt. The LLM generates action card names for each of these (replacing the original title of Table I) as well as the action's description, which is not shown to the player but impacts the generation of its card art detailed in Section III-B2.

Each location has one enemy with two minions; those are generated through separate LLM calls that include details of this location in the prompt. Along with location details, the prompt includes the PC details and the thematic details (except visual style) to generate the main enemy details (its name, species, gender, and description). Details for two minions (as above) per enemy are generated through a single LLM call that includes the main enemy details, the location details, the PC details and the thematic details (except visual style). As above, enemies' and minions' details are used for the generation of visuals but are not explicitly shown to the player; only their names in Card Combat view are shown (see Fig. 2b).

Finally, each location's semantic details (along with thematic details bar visual style) are used to generate semantic descriptions for the walls, floors, doors, and keys of this location. These are not shown to the player but inform the generation of the dungeon crawler tileset (see Section III-B2).

2) Generation of visuals: Unlike the semantics generated in the previous step (see Section III-B1), all generated visuals are player-facing. Some are used in the dungeon crawling view (e.g. tilesets, PC and enemy animations) while some are used in the card combat view (e.g. action cards and location background). Due to the complex nature of the asset generation pipelines of dungeon crawling views, we detail those in dedicated sections below and present all visual generation processes for card combat at the end of Section III-B2.

All visual assets are generated using a community fine-tuned Stable Diffusion XL (SDXL) [35] model called Artium V2.0[4] due to its versatility across various art styles. In addition, there are a number of ControlNet [11] models available for SDXL which constrain the generated image to either follow the contours of a predefined line drawing, or follow a predefined pose for a character. An IP-Adapter [12] is also available for this model, which retains character consistency from a provided reference image when generating new images. We note that SDXL generates fairly high-resolution assets (best results are with 1024×1024 pixels) which is above what a simple tile-based level layout would require; however, we keep high-resolution results for all aspects of CrawLLM due to the improved image quality. Note that all prompts for SDXL for all visual asset generations include the visual style details generated via LLMs (see Section III-B1), and we will not repeat this information in the explanations that follow. All line drawings and masks mentioned below were designed ad hoc by the authors and are part of the game template (see Fig. 1).

[2]https://ollama.com/

[3]https://github.com/m-a-r-v-i-n/CrawLLMPrompts

[4]https://civitai.com/models/216439/artium

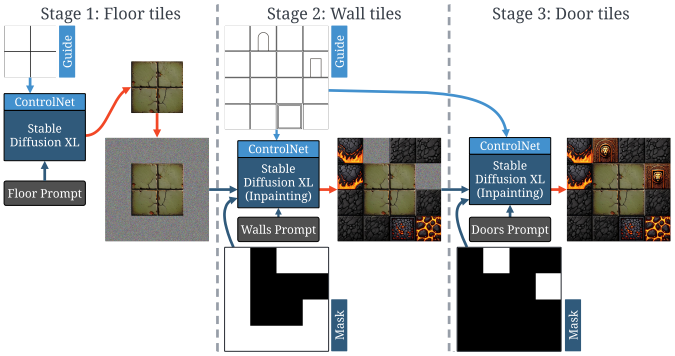


Fig. 4: The tileset generation process carried out in 3 stages. Tiles from the game *"Aeternum Ruptura"*.

a) Tileset Generation: Each location has floors, wall and door tiles that must be arranged in a 2D grid, acting as a background for the player to navigate. Generating appropriate visuals that clearly demarcate traversable areas (floors) from impassable areas (walls) and interactable tiles (doors and secret doors) is challenging and thus requires some elaboration.

In initial trials, the floors, walls, and door tiles were generated separately as individual images. Although the results were acceptable when observed individually, the floor tiles were often too intricate or the walls offered little contrast to them, and demarcation of the traversable areas was unclear. In follow-up trials, each location's tileset was instead generated as a single image with specific parts of the image representing different tile types, using ControlNet based on a custom-made outline for this task. However, the SDXL model was unable to correctly distinguish between regions intended to be walls and those intended to be floors.

In this version of CrawLLM, each location's tileset is generated as a single image in three stages as shown in Figure 4. Initially, we provide a line drawing as a guide for the central 4 floor tiles to the SDXL via ControlNet, together with the semantic description for floor tiles (see Section III-B1). The generated image with the floor tiles is then extended to the full tileset size, with random noise in unpainted areas. This noisy tileset image is provided to the model, along with wall descriptions (semantics), a line drawing as a guide to ControlNet, and a mask identifying the regions of the walls for inpainting (see Fig. 4). The resulting image of this stage has almost all tiles (including the secret door, which should look like a wall) generated except for two tiles which are re-painted with random noise. These two tiles are populated with two door variants in the final stage. The updated noisy tileset image is provided to the model, along with door descriptions (in the prompt), a line drawing guide for ControlNet and a mask identifying the two door regions for inpainting, to produce the final result shown at the far right of Figure 4.

Note that when creating the final visual of the dungeon (see Fig. 6), each wall tile is chosen at random among the 9 wall textures of the tileset, each floor tile is chosen at random among the 4 floor textures of the tileset, and each door is chosen at random between the 2 door textures. This leads to more visual variety within a location, but may lead to some dissonant results when tiles of the same type are very different.

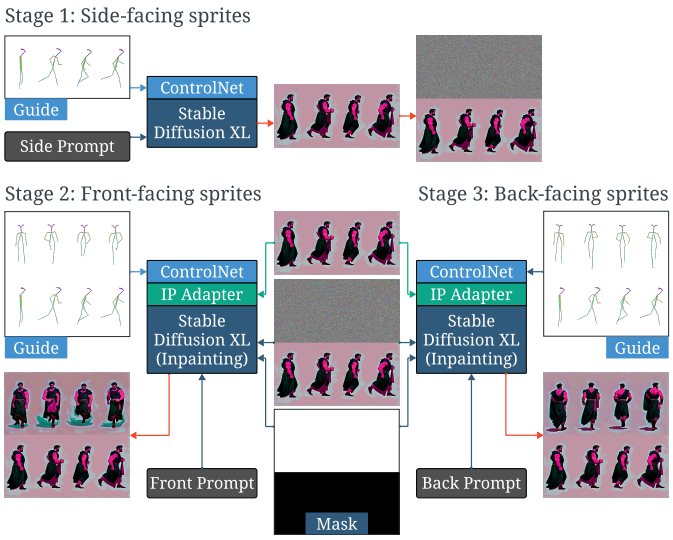


Fig. 5: The spritesheet generation process carried out in 3 stages. The back, front, and side-facing sprites are combined into a single image with all 12 poses. PC spritesheet from the game *"Aeternum Ruptura"*.

b) Generation of Character and Opponent Animations: Since the PC, enemies and minions move around in the Dungeon Crawling view, it is important to create movement animations along the four cardinal directions. These animations are done through *spritesheets*, which consist of four frames of movement along three directions (front, back, and left[5]). A spritesheet requires high consistency between frames, and a controlled pose and positioning in each.

To generate each spritesheet, a guide image with predefined OpenPose poses [50] was produced in Unity: a 3D humanoid model was captured at different stages throughout an idle and a jogging animation. Points and lines corresponding to the various joints and bones (as specified in the OpenPose standard) were drawn at each key frame of the required animation (see Fig. 5). The resulting images were collated into a single image as a reference.

While initial trials attempted to use this spritesheet guide via ControlNet to generate the spritesheet in one go, the generator often mistook back movement with front movement. The generation is therefore split into three stages, one for each pose orientation (see Fig. 5): prompts always include the character's details (PC, enemy, or minion) as generated in Section III-B1. In the first stage the guides for the side-facing poses are provided as a single ControlNet reference. The resulting image is then extended vertically with random noise. The front-facing poses are added to the ControlNet guide image, and an IP-Adapter with the generated side-facing sprites as reference was added to the SDXL model. A mask for inpainting the front-facing sprites in the freshly extended region is also provided to SDXL, and the text prompts direct it to draw the same character in front-facing poses. The process is repeated to generate the back-facing poses from the side-facing

[5]Right movement is a simple reflection of the left animation.

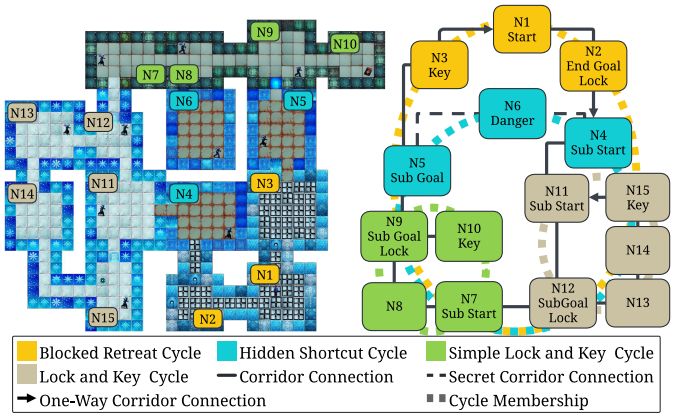


Fig. 6: A sample generated dungeon (left), and the graph representation which generated it (right). The nested cycle structure is delineated on the graph. The game depicted is *"Spirits of the Ice: Arctic Research Unbound"*.

set, with the respective back-facing guides and prompts. The three parts are then combined into a single spritesheet.

The final step of the process requires the removal of the background in the image. The text prompts throughout the generative process include instructions to draw the images on a blank background. *Rembg*[6], an open-source Python library for background removal based on a pre-trained neural network, was used to add transparency to the final image. In rare cases, some peripheral items in the image were erroneously cropped out as background.

c) Generation of other Dungeon Crawling assets: The only assets not covered above are key sprites, which are generated using the location-based key descriptions (and visual style) as SDXL prompts, with an additional post-processing step for background removal via *Rembg*.

d) Generation of Card Combat assets: Most assets for the card combat are fairly simple text-to-image prompts, requiring only the thematic details and the relevant asset details: location details for the background image, enemy details for the card art of the enemy, PC details for the card art of the PC, and action details (name and description) for the card art of the PC's action cards. The only complex generative pipeline is for the card's frame. Three card frames are generated, colour-coded to signify the rarity of the card (gold, silver and bronze). Card frame prompts include all thematic elements (setting, epoch, environment, and visual style), which is not the case for any other visual asset. Each card has several strictly delimited areas for the card art, descriptive or functional information, and a smaller area for icons showing the intended actions for opponents' cards. To generate this card frame, a line drawing guide was used for all three frames to guide SDXL via ControlNet.

C. Dungeon Generation

The dungeon layout is procedurally generated at the start of each playthrough. The layout follows a cyclic structure [51], inspired by the algorithm introduced in *Unexplored* (Ludomotion, 2017), and results in solvable puzzles involving keys and locked doors. CrawLLM uses a number of hand-crafted cycles adapted from [52]. Nodes can define the goal room (e.g. where the enemy or locked door is located), whether the room contains a key, or its danger level (which determines how many minions are added in card combat). During generation, a random cycle from the collection kickstarts the process, and is then expanded by replacing a node in this cycle with a full cycle (a node of which may again be replaced with a full cycle and so on). The generative process ends when some termination conditions are met (to ensure that routing is still possible on a 2D grid); cycles are then grouped as locations (see Fig. 6). The maximum number of cycles in a dungeon is five (thus the need for 5 location names and assorted assets in Section III-B) but can be fewer if graph generation terminates early, as in Figure 6.

While the cyclic dungeon graph is fairly simple to generate, placing the nodes and paths onto a 2D grid is non-trivial and must be done iteratively. In CrawLLM, nodes are translated into rectangular rooms and edges are represented as (straight or right-angled) corridors. Placement starts from the innermost layer of cycles, with one node chosen and placed as a room at the grid's origin point. Connected rooms are then placed on a grid around it, and this process is repeated recursively for all remaining rooms. Since CrawLLM is designed to be purely two-dimensional, with no overlapping corridors or rooms, this process might occasionally be unable to place some corridors between rooms. However since this graph-routing process is rapid, it can be restarted until a possible placement is found. Figure 6 shows how the graph layout is translated into a 2D grid, with appropriate tiles and visual elements.

D. Game Collection

The above pipeline was applied to generate 20 different themes and assets for each. All resulting games were playable; the first author played through 10 of these games to produce screenshots during dungeon crawling and card combat which were used for the user study described in Section IV.

IV. User Evaluation

In order to assess the coherence and relevance of the generated assets within the game context, a user study was conducted. The design of the study is described in Section IV-A and its results in Section IV-B

A. Experimental Protocol

The study aimed to address two research questions: (a) whether the thematic details which guided every aspect of the generation of both text and visuals were clearly discernible in the final result and (b) whether the final results were appealing to human players. To answer these questions, we leveraged the game collection of 20 games (see Section III-D) and produced one stimulus per game to show users: the first 5 games were used to produce Dungeon Crawling screenshots (D1-D5), the next 5 games were used to produce Card Combat

[6]https://github.com/danielgatis/rembg

screenshots (C1-C5), and from the next 5 games we collected text descriptions of one enemy (E1-E5). This means that 5 games were not shown to users, but their thematic details were still used to potentially confuse users (as described below). The games were selected in the order they were generated, and there was no cherry-picking or preferential choice for any of these screenshots, text descriptions, or survey questions.

For each stimulus, we collected relevant thematic details of that game: setting, environment, epoch and visual style for dungeon crawling screenshots, setting, epoch and visual style for Card Combat screenshots (since the environment was not visible), and setting and epoch for enemies' text descriptions. For each of these thematic details, we selected the 3 nearest respective thematic details among the remaining 19 games. Distance in this case was computed by the Foundation Language And Vision Alignment (FLAVA) [32] model based on the game's own thematic detail in question (e.g. Setting). FLAVA can map images and text to the same latent space, thereby making it possible to quantify distances between vector embeddings of text and images; moreover it is orthogonal to OpenAI's CLIP [31] similarity which is used for generation of images in Stable Diffusion. In this way, each stimulus had 4 options for setting (one correct coming from this game's thematic details), 4 options for epoch, etc.

The study was designed to be carried out online, using Google Forms. A link for participation was distributed through the authors' social media channels. The form was prefaced by an informed consent form followed by questions about the user's game experience (but no demographic questions). The form dedicated one page per stimulus: the stimulus was shown first, then the user was requested to choose the appropriate setting, epoch, environment or visual style (depending on the stimulus) from a list of 4 options shown as a multiple-choice question. The order of options was randomised in advance and was the same for all participants. A final question per stimulus asked users to rate its appeal on a 5-item Likert scale ("The Screenshot is visually appealing" or "The Description is entertaining to read"). The study was approved by the University of Malta Research Ethics Committee. We highlight that in this study, the users do not play the generated games, but are only presented with static screenshots. This protocol assumes that assessing coherence between final visual output and theme can be done on static images (and text) with less cognitive load than while playing, and the process is more controllable for analysis as the authors have full knowledge of what the users observed. Furthermore, allowing respondents to play the game would expose them to a large number of diverse assets which would have facilitated the identification of the thematic descriptors. For example, in the introductory text shown to the players, there may be phrases or words which also appear in the theme's setting, environment, or epoch. This would skew the results of this survey, as it was designed to assess whether each set of assets (Dungeon Crawling visuals, Card Combat visuals, and pure text) individually represents each thematic descriptor accurately or not.

Specifically, the hypotheses of the experiment (aligned with the choice of stimuli) are listed below:

H1 The ability of the human evaluators to identify the intended thematic details that guided the generation process is significantly above chance level.

H2 The mean appeal of the generated game assets, as rated by human evaluators, is significantly higher than the neutral midpoint on a Likert scale.

These hypotheses respectively formalise the research questions posed at the beginning of this section.

TABLE II: A summary of the user study results. The accuracy of the answers is shown together with the entropy over the responses for that question. An asterisk (*) denotes statistically better outcomes (higher rating, higher accuracy, lower entropy) than a random selection.

#	Accuracy (Entropy)				Mean
	Setting	Environment	Epoch	Style	Rating
D1	79%* (0.37*)	68%* (0.56*)	44%* (0.76*)	50%* (0.89*)	2.5
D2	82%* (0.34*)	94%* (0.16*)	15% (0.3*)	47%* (0.89*)	3.4*
D3	32% (0.79*)	15% (0.61*)	0% (0.67*)	47%* (0.5*)	2.4
D4	53%* (0.8*)	74%* (0.52*)	62%* (0.73*)	27% (0.99)	3.2
D5	41%* (0.94)	29% (0.53*)	9% (0.73*)	68%* (0.61*)	3.2
C1	65%* (0.65*)	–	18% (0.81*)	71%* (0.58*)	4.0*
C2	15% (0.51*)	–	15% (0.66*)	35% (0.94)	3.9*
C3	88%* (0.32*)	–	0% (0.76*)	9% (0.82*)	3.6*
C4	97%* (0.10*)	–	94%* (0.19*)	100%* (0*)	3.6*
C5	94%* (0.19*)	–	100%* (0*)	15% (0.78*)	3.8*
E1	77%* (0.56*)	–	0% (0.37*)	–	3.1
E2	79%* (0.52*)	–	91%* (0.26*)	–	3.4*
E3	0% (0.59*)	–	77%* (0.46*)	–	3.4*
E4	79%* (0.46*)	–	68%* (0.67*)	–	3.4*
E5	82%* (0.39*)	–	85%* (0.37*)	–	3.5*

B. Results

A total of 34 responses were collected through convenience sampling. Most participants were gamers: only 6% claimed they never play video games while 47% play regularly. Most participants (61%) enjoy playing dungeon crawlers, while 21% answered that they never played this genre. Similarly, 62% of participants enjoy playing card-based video games, and 12% had never played any.

We process the results in line with the hypotheses, and compute metrics for each stimulus separately. To address H1 and assess coherence, we measure *accuracy* as the ratio of participants' responses that matched the game's thematic details over all responses. We also measure Shannon's *entropy* [53] normalised to the number of options; if entropy is low, users mostly agreed on the same choice (even if it was the incorrect one, if accuracy is low). To address H2 and assess appeal, we compute the *mean rating* from all participants' ratings on the 5-item Likert question regarding appeal (see Section IV-A).

We assess statistical significance (at $p < 0.05$) assuming a random choice: for ratings, we test significance based on the 95% confidence intervals of the mean rating against the baseline Likert score of 3. For accuracy, we use the binomial test assuming a 1 in 4 chance of choosing the right option [54] and for entropy we use the χ^2 goodness of fit [55] which tests whether the users' answers fit an even distribution among all options. The Benjamini–Hochberg False Discovery Rate correction [56] was applied to all the above tests to limit false positives, as it offers greater sensitivity than more restrictive approaches.

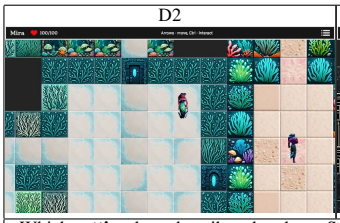
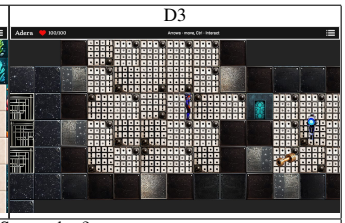
Table II presents the summary of the above performance metrics per stimulus (in the order that they were presented to

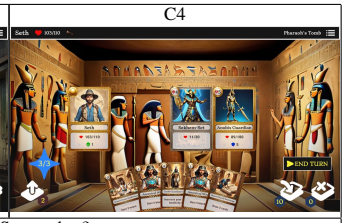
participants). Most stimuli were found to be appealing, with ratings statistically above the baseline (Likert score of 3) in 10 of 15 cases. Card combat screenshots were overall rated to be far more appealing than other stimuli; dungeon crawler screenshots had more varied ratings, with 2 of 5 screenshots ranking significantly below the baseline (D1, D3). In terms of accuracy, we observe a variety in results. On the one hand, accuracy was significantly higher than the random baseline (25%) in 30 of 45 questions. On the other hand, accuracy differed markedly across stimuli. While setting questions had the highest accuracy on average (64%) there were still 3 instances out of 15 where accuracy was not significantly higher than random chance—including a case where all 34 participants guessed the setting incorrectly (for E3). Epoch questions received the least accurate answers (average accuracy of 43% and 6 of 15 cases not significantly above random chance) followed closely by visual style questions (average accuracy of 45% and 4 of 10 cases not significantly above random chance). We also observe that in many questions participants tended to choose different options: 28 of 45 questions had over 0.50 normalised entropy score, and 7 of 45 over 0.80. While this is not surprising in cases where the correct option was rarely picked (e.g. for style of C3 which has an accuracy of 9% and an entropy score of 0.82), the pattern persisted even when accuracy was significantly above chance levels (e.g. for style of D1 with 50% accuracy and an entropy score of 0.89). While entropy scores heavily depended on the stimulus, we observe that questions on visual style were more confusing, with a higher average entropy (0.70) compared to other questions (the average entropy of which fluctuated around 0.50).

In order to elucidate on the differences between users' response patterns in different questions, we highlight some indicative examples of dungeon crawling screenshots (D2, D3) and card combat screenshots (C3, C4) in Table III.

As shown in Table II, D2 exhibited a high accuracy for setting and environment, while D3 showed low accuracies in both. For D2, all 6 users that did not guess the correct setting ("A mystical underwater city") chose "A subterranean realm of mushroom forests and crystal caves" instead. This is surprising, since only 2 users did not guess the correct environment ("Coral reefs, sunken temples, deep sea trenches"). Therefore, while almost all users identified the marine environment (likely due to marine fauna on the dungeon walls), some of those users estimated that the setting is subterranean rather than marine-themed (e.g. "A sunken pirate city in a stormy sea" or "A mystical underwater city"). With regards to epoch, both D2 and D3 had very low accuracies; in D3 no user guessed the correct epoch. Admittedly, the options for the epoch questions in both stimuli were unhelpful: most users chose "fantasy" for D2 (85%) while for D3 users were split between "science fiction" and "space age" (50% and 41% respectively). Users likely recognised established terms used e.g. in literature (fantasy, science fiction). It is arguably difficult to depict concepts such as "Ancient" and "Near future" (the correct choices for D2 and D3 respectively) both visually and semantically in a way that clearly differentiates them from other competing terms such as "Medieval" and "Science fiction" respectively. More importantly, however, the setting seems dissonant from the

TABLE III: Example stimuli and questions from the user study. Checked options are the ones used to generate the game shown (and should ideally be guessed by the users).

D2	D3
	
Which **setting** best describes the above Screenshot?	
☐ A subterranean realm of mushroom forests and crystal caves ☐ A sunken pirate city in a stormy sea ☒ A mystical underwater city ☐ A lost colony on an alien world	☐ A lost colony on an alien world ☒ A drifting asteroid mining facility ☐ A subterranean realm of mushroom forests and crystal caves ☐ An abandoned space station overrun by mutants
What best describes the **environment** in the above Screenshot?	
☐ Flooded streets, shipwrecks, treasure troves ☐ Trap-filled corridors, massive machinery, hidden chambers ☒ Coral reefs, sunken temples, deep sea trenches ☐ Trap-filled corridors, mummies and scarabs, hidden treasures	☐ Trap-filled corridors, massive machinery, hidden chambers ☐ Zero gravity corridors, malfunctioning machinery, airlocks and vents ☒ Gravity-less corridors, cryogenic chambers, dark caves ☐ Trap-filled corridors, mummies and scarabs, hidden treasures
Which **epoch** is depicted in the above Screenshot?	
☐ Fantasy ☒ Ancient ☐ Medieval ☐ Renaissance	☐ Lost civilization ☐ Space age ☐ Science-fiction ☒ Near future
What best describes the **visual style** in the above Screenshot?	
☐ Semi-abstract, vivid alien landscapes ☐ Digital matte painting, cold color palette ☒ Stylized realism, bioluminescent color palette ☐ Mystical realism, moody	☐ Science-fiction, gritty ☐ Retro-futuristic, vibrant and nostalgic ☒ Science-fiction, minimalist ☐ Retro-futuristic, neon-lit

C3	C4
	
Which **setting** best describes the above Screenshot?	
☐ An enchanted forest on a floating island ☒ A lost colony on an alien world ☐ A subterranean realm of mushroom forests and crystal caves ☐ A mystical underwater city	☐ An ancient Roman city swallowed by a volcano ☒ An ancient labyrinth beneath a desert pyramid ☐ A forgotten underground library ☐ A lost colony on an alien world
Which **epoch** is depicted in the above Screenshot?	
☐ Near future ☐ Science-fiction ☒ Space age ☐ Lost civilization	☒ Ancient Egypt ☐ Medieval ☐ Fantasy ☐ Science-fiction
Which **epoch** is depicted in the above Screenshot?	
☒ Semi-abstract, vivid alien landscapes ☐ Science-fiction, minimalist ☐ Retro-futuristic, vibrant and nostalgic ☐ Dreamlike, surreal	☐ Grimy steampunk, surrealistic ☐ Semi-abstract, vivid alien landscapes ☒ Egyptian relief, earthy tones ☐ Science-fiction, gritty

epoch: mystical underwater cities are not associated with an ancient era (except via the most abstract connections) while asteroid mining facilities are not expected in the near future. The issue is not one of possibility but one of tropes: tropes such as mystical underwater cities abound in fantasy literature and asteroid mining facilities are staples of science fiction movies. Both D2 and D3 had the same accuracy (47%) for style, which despite being modest, was significantly above chance levels; however, entropy scores for the two stimuli differ widely. For D2, style answers were distributed among the incorrect options (the most popular incorrect choice being "Semi-abstract, vivid alien landscapes" at 26%). For D3, all users that did not choose the correct style ("Science-fiction, minimalist") instead chose "Science-fiction, gritty" (53%). Once again, we note the tendency towards the keywords "science fiction" (evident in the choice for epoch, above). The example highlights a more general issue with the experimental protocol: by choosing the semantically closest settings among the remaining 19 games, in many cases almost identical options were shown to the user. While we omitted exact duplicate options from the questionnaire, two of the style options here are only slightly different; similar examples include options for epoch being "Gothic Victorian" and "Victorian era" for D5. While there is significant evidence from both the quantitative results of Table II and the examples of Table III that epoch and style were difficult to depict in an in-game screenshot, we revisit the limitations of the experimental protocol in Section V. To finish the comparison between D2 and D3, we note that D2 had the highest rating across all dungeon crawler screenshots (3.4) while D3 had the lowest (2.4). We surmise that the noisy background for the floor tiles of D3 was responsible for the low rating, while D2 had flat colours for floor tiles and sharp contrasting colours for wall tiles. The organic shapes in the walls might also be perceived as pleasant. It is worth noting that while the visual style for D2 is reasonably retained via "bioluminescent color palette", the noisy floor tiles are not representative of a "minimalist" visual style described in the prompt. We revisit this limitation in Section V.

For the card combat views, C3 and C4 are interesting in that the setting is easily determined by the users for both stimuli, but for C3 neither epoch nor style are guessed correctly. The epoch options for C3 are identical to those for D3 (except the order): no user chose the correct "space age", with almost equal picks between "near future" (41%) and "science-fiction" (38%). This is an interesting observation, considering that for D3 the correct "near future" option for epoch was never chosen while "space age" (which was correct for C3) was chosen by 41% of users in D3. Barring any ordering effects, the users' behaviour is difficult to explain. We hypothesise that the background image for C3, which shows an indoor view of a warehouse. does not indicate the "space age" due to a lack of astronomical or starship elements (which are admittedly not necessary since the Space age is a time period rather than a location). The visual style options, however, could explain why users rarely (9%) chose the correct "Semi-abstract, vivid alien landscapes" (none of which are present in the background image of C3) and instead mostly chose "science fiction, minimalist" (50%) instead. As above, we are reminded that visual style is difficult to convey unless both the semantic description is very specific and SDXL can find appropriate ways to depict this. A perfect case where both conditions are met is C4: the style "Egyptian relief, earthy tones" is much more specific than "science-fiction, minimalist" (D3), but also easily conveyed visually in all respects in the background of C4. All users guessed the correct visual style for C4, the only case where this occurred in all screenshots used for this user study.

V. Discussion

The online user study asked 34 participants to rate game content (screenshots and text descriptions) and guess the thematic details used to generate them. The goal of the user study (see Section IV-A) was to assess whether the original theme prompts are recognizable in the final game-ready content (H1) and whether the generated assets are appealing (H2). Participants rated 10 out of 15 screenshots or descriptions significantly above neutral ratings; we thus consider H2 at least partially validated. While the card combat view was always found appealing, the dungeon crawling view could be severely impacted by noisy backgrounds (e.g., floor tiles). Participants accurately guessed the thematic details from a screenshot or a text description in 30 out of 45 instances (significantly above random chance levels); H1 can thus be considered at least partially validated. However, it was observed that visual style and epoch were difficult to convey visually in-game. Through qualitative analysis of the images and user responses, a snippet of which is included in Section IV-B, it can be argued that the background of the card combat view played a major role in conveying the thematic details. The thematic details were typically conveyed successfully only when all such details were highly specific (e.g., when describing a particular place rather than a general genre trope).

The user study highlighted some limitations of CrawLLM which were already evident from our internal playthroughs of the games. Generating a consistent tileset was challenging and required many revisions. The three-step process shown in Fig. 4 overcomes several limitations of earlier trials, but is still prone to producing noisy background tiles as evidenced in some of the generated games (e.g. C3 in Table III). Future work could address this by using image metrics such as complexity [57] and colourfulness [58] to choose the least noisy option from the many generated tiles of each tileset (see Fig. 4) instead of randomly selecting among all options. Other limitations pertain to the experimental protocol itself, which led to near-duplicate options which confused users (e.g. options for style in Table III). Future user studies should perhaps remove near-duplicates and show only closest neighbours above some minimum distance threshold. A more critical limitation, however, pertains to the way that some thematic elements control text and image generation. Evidently, in many cases the visual style description had no (visible) impact in the generation of many visual assets, be it tilesets or card combat backgrounds. We hypothesise that this may be a limitation of the prompting method: visual style is among the last words in most prompts for visuals generation, and may fall outside the context length of SDXL's CLIP model

[31] or may be overshadowed by earlier parts of the prompt (e.g. the description). However, it may also be a limitation of the LLM that generates the visual style description. Some visual styles are easier to act upon in follow-up steps (e.g. "Egyptian relief") than others (e.g. "semi-abstract"), but the LLM that generates the visual style can not ascertain that—in part because it is unaware of the purpose of its output. In future work, a way to address this could be to request adjustments by the LLM to the visual style description in cases where results are improper (e.g. if tiles are too complex as described above). Future work could also explore the impact of reducing information passed on to the SDXL model: for example, location descriptions may be too much information for generating a background image for card combat, and the location title and visual style alone could provide enough control so that there is meaningful transfer in style. Similar concerns can be raised about the epoch thematic detail: perhaps omitting epoch altogether could lead to more concise generation of follow-up assets—highlighting more important elements such as the setting instead. Finally, a limitation of the current paper is that we did not focus on analysing enemy descriptions in the same way that we analysed screenshots in Table III. In part, this is because enemy descriptions are not shown directly to players in CrawLLM. Moreover, most enemy descriptions received significantly above average ratings (and in most cases high accuracies in guessing the thematic details) and thus we focused on the mixed results from screenshots instead.

Another limitation pertains to the user study, which relied on task-specific ad-hoc metrics rather than validated perceptual scales [59]. While appropriate for the exploratory goals of this study, future work could incorporate established aesthetic judgment metrics or reliability analyses to further strengthen claims about perceived quality. The playability of the CrawLLM games also remain to be tested in actual playtest sessions—with or without coherence and perceptual quality questions.

Future work should refine the gameplay loop of CrawLLM, providing more complex interactions (and decision-making) to the player and more opportunities for impactful generation. This can be done via more complex enemy actions in combat (to leverage mechanics generation), via rewards for combat (e.g. themed action cards based on the enemy defeated, or dialogue with the defeated enemy for narrative generation), or via spatial puzzles during dungeon crawling (to elevate level generation). The pipeline can also be applied to different game genres, and also extended to accommodate generation across more modalities, such as audio. These additions would elevate CrawLLM, which currently relies fully on a pre-scripted gameplay loop, to a true *game generation* pipeline with more unexpected (and, ideally, thematically coherent) outcomes.

VI. Conclusion

This paper introduced CrawLLM, a novel pipeline that generates coherent game assets using LLMs and text-to-image models. The system produces narrative and visual components for a dungeon crawler game with card-based combat, operating on a feed-forward fashion to create playable games without iterative refinement [1]. User evaluations indicated that the generated games maintained thematic consistency, with participants easily identifying settings and environments in most cases, although epochs and visual styles were often too broad or ambiguous. CrawLLM highlights the potential of LLM-driven content creation across narrative and visual elements. Our user study also revealed limitations, such as the LLM's tendency to produce vague definitions and the text-to-image models' struggles with less common visual styles. Despite its capabilities, the feed-forward methodology may not consistently yield optimal results without refinement. We highlighted potential ways that iteration can be re-introduced as part of the generative process in Section V.

References

[1] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, "Orchestrating game generation," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019.

[2] S. Colton, "Creativity versus the perception of creativity in computational systems," in *Proceedings of the AAAI Spring Symposium: Creative Intelligent Systems*, 2008.

[3] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, "PCGRL: Procedural content generation via reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020.

[4] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural Content Generation via Machine Learning (PCGML)," *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018.

[5] A. Liapis, "Exploring the visual styles of arcade game assets," in *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt)*. Springer, 2016.

[6] P. E. Hutchings and J. McCormack, "Adaptive music composition for games," *IEEE Transactions on Games*, vol. 12, no. 3, pp. 270–280, 2020.

[7] C. Browne and F. Maire, "Evolutionary game design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, 2010.

[8] H. Chen, G. Ding, X. Liu, Z. Lin, J. Liu, and J. Han, "IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[9] R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius, and G. N. Yannakakis, "Large Language Models and Games: A Survey and Roadmap," *IEEE Transactions on Games*, vol. Early Access, 2024.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[11] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," *arXiv preprint arXiv:2302.05543*, 2023.

[12] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models," *arXiv preprint arXiv:2308.06721*, 2023.

[13] D. Şen, H. T. Küçükkaykı, and E. Sürer, "Automated game mechanics and aesthetics generation using neural style transfer in 2D video games," *Journal of Information Technologies*, vol. 14, no. 3, pp. 287–300, 2021.

[14] T. Phillips, ""Don't clone my indie game, bro"': Informal cultures of videogame regulation in the independent sector," *Cultural Trends*, vol. 24, no. 2, pp. 143–153, 2015.

[15] A. Liapis, "10 years of the PCG workshop: Past and future trends," in *Proceedings of the FDG Workshop on Procedural Content Generation*, 2020.

[16] J. Ryan, M. Mateas, and N. Wardrip-Fruin, "Generative character conversations for background believability and storytelling," in *Proceedings of the 1st International Joint Conference of DiGRA and FDG*, 2016.

[17] G. A. B. Barros, M. C. Green, A. Liapis, and J. Togelius, "Who killed Albert Einstein? From open data to murder mystery games," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 79–89, 2019.

[18] M. Cook and S. Colton, "A Rogue Dream: Automatically generating meaningful content for games," in *Proceedings of the AIIDE Workshop on Experimental AI in Games*, 2014.

[19] M. Treanor, B. Schweizer, I. Bogost, and M. Mateas, "The micro-rhetorics of Game-o-Matic," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2012, p. 18–25.

[20] A. Summerville, C. Martens, B. Samuel, J. Osborn, N. Wardrip-Fruin, and M. Mateas, "Gemini: Bidirectional generation and analysis of games via ASP," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2018, pp. 123–129.

[21] M. Cook, S. Colton, and A. Pease, "Aesthetic considerations for automated platformer design," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2012.

[22] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[23] M. Cook, S. Colton, A. Raad, and J. Gow, "Mechanic Miner: Reflection-driven game mechanic discovery and level design," in *Proceedings of the Applications of Evolutionary Computation*, 2012.

[24] A. Zook and M. Riedl, "Automatic game design via mechanic generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[25] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, "DATA agent," in *Proceedings of the 13th Conference on the Foundations of Digital Games*, 2018.

[26] Y. Rabii and M. Cook, ""Hunt Takes Hare": Theming games through game-word vector translation," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2024.

[27] J. Kowalski, A. Liapis, and Ł. Żarczyński, "Mapping chess aesthetics onto procedurally generated chess-like games," in *Applications of Evolutionary Computation*. Springer, 2018.

[28] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 2008.

[29] A. Summerville and M. Mateas, "Sampling Hyrule: multi-technique probabilistic level generation for action role playing games," in *Proceedings of the AIIDE Workshop on Experimental AI in Games*, 2015.

[30] B. Lavender and T. Thompson, "Adventures in Hyrule: Generating missions & maps for action adventure games," in *Proceedings of the Foundactions of Digital Games (Extended Abstracts)*, 2015.

[31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning*, 2021.

[32] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "FLAVA: A foundational language and vision alignment model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[33] S. Qiu, "Generative AI processes for 2D platformer game character design and animation," in *Proceedings of the 2nd International Conference on Interdisciplinary Humanities and Communication Studies*, 2023.

[34] Q. Sun, Q. Luo, Y. Ni, and H. Mi, "Text2AC: A framework for game-ready 2D agent character (AC) generation from natural language," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024.

[35] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller *et al.*, "SDXL: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[36] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[37] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix *et al.*, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[38] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[39] P. Kumar, "Large language models (LLMs): Survey, technical frameworks, and future challenges," *Artificial Intelligence Review*, vol. 57, no. 260, 2024.

[40] D. Yang, E. Kleinman, and C. Harteveld, "GPT for games: An updated scoping review (2020-2024)," *IEEE Transactions on Games*, 2025, early Access.

[41] G. Todd, S. Earle, M. U. Nasir, M. C. Green, and J. Togelius, "Level generation through large language models," in *Proceedings of the International Conference on the Foundations of Digital Games*, 2023.

[42] C. Hu, Y. Zhao, and J. Liu, "Game generation via large language models," in *Proceedings of the IEEE Conference on Games*, 2024.

[43] Z. Wu, Z. Chen, D. Zhu, C. Mousas, and D. Kao, "A systematic review of generative AI on game character creation: Applications, challenges, and future trends," *IEEE Transactions on Games*, vol. Early Access, 2025.

[44] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. J. Shi, E. Hughes *et al.*, "Genie: Generative interactive environments," in *Proceedings of the International Conference on Machine Learning*, 2024.

[45] S. Earle, A. Khalifa, M. U. Nasir, Z. Jiang, G. Todd, A. Banburski-Fahey, and J. Togelius, "ScriptDoctor: Automatic generation of PuzzleScript games via large language models and tree search," in *Proceedings of the IEEE Conference on Games*, 2025.

[46] D. Li, S. S. Sohn, S. Zhang, C.-J. Chang, and M. Kapadia, "From words to worlds: Transforming one-line prompts into multi-modal digital stories with LLM agents," in *Proceedings of the ACM SIGGRAPH Conference on Motion, Interaction, and Games*, 2024.

[47] J. Li, Y. Li, N. Wadhwa, Y. Pritch, D. E. Jacobs, M. Rubinstein, M. Bansal, and N. Ruiz, "Unbounded: A generative infinite game of character life simulation," in *Proceedings of the International Conference on Learning Representations*, 2025.

[48] R. Gallotta, A. Liapis, and G. N. Yannakakis, "Consistent game content creation via function calling for Large Language Models," in *Proceedings of the IEEE Conference on Games*, 2024.

[49] H. Wei, J. Bizzocchi, and T. Calvert, "Time and space in digital game storytelling," *International Journal of Computer Games Technology*, vol. 2010, 2010.

[50] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 172 – 186, 2019.

[51] J. Dormans, "Cyclic Generation," in *Procedural Generation in Game Design*, T. X. Short and T. Adams, Eds. Taylor & Francis, CRC Press, 2017, pp. 83 – 96.

[52] S. Victory, "Cyclic dungeon generation," https://sersavictory.itch.io/cyclic-dungeon-generation, accessed 11 Oct 2024.

[53] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[54] Y. Dodge, "Binomial test," in *The Concise Encyclopedia of Statistics*. Springer New York, 2008, pp. 47–49.

[55] ——, "Chi-square goodness of fit test," in *The Concise Encyclopedia of Statistics*. Springer New York, 2008, pp. 72–76.

[56] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 2018.

[57] P. Machado, J. Romero, M. Nadal, A. Santos-del Riego, J. Correia, and A. Carballal, "Computerized measures of visual complexity," *Acta Psychologica*, vol. 160, pp. 43–57, 2015.

[58] D. Hasler and S. Suesstrunk, "Measuring colourfulness in natural images," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2003.

[59] M. Hassenzahl, M. Burmester, and F. Koller, "AttrakDiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer qualität," in *Mensch & Computer*. Vieweg+Teubner Verlag, 2003, p. 187–196.