



The Ink Splotch Effect: A Case Study on ChatGPT as a Co-Creative Game Designer

Asad Anjum
asad.anjum@nyu.edu
New York University
Brooklyn, New York, USA

Yuting Li
yl7685@nyu.edu
New York University
Brooklyn, New York, USA

Noelle Law
noelle.t.law@gmail.com
Independent Researcher
Baltimore, Maryland, USA

M Charity
mlc761@nyu.edu
New York University
Brooklyn, New York, USA

Julian Togelius
julian@togelius.com
New York University
Brooklyn, New York, USA

ABSTRACT

This paper studies how large language models (LLMs) can act as effective, high-level creative collaborators and “musings” for game design. We model the design of this study after the exercises artists use by looking at amorphous ink splotches for creative inspiration. Our goal is to determine whether AI-assistance can improve, hinder, or provide an alternative quality to games when compared to the creative intents implemented by human designers. The capabilities of LLMs as game designers are stress tested by placing it at the forefront of the decision making process. Three prototype games are designed across 3 different genres: (1) a minimalist base game, (2) a game with features and game feel elements added by a human game designer, and (3) a game with features and feel elements directly implemented from prompted outputs of the LLM, ChatGPT. A user study was conducted and participants were asked to blindly evaluate the quality and their preference of these games. We discuss both the development process of communicating creative intent to an AI chatbot and the synthesized open feedback of the participants. We use this data to determine both the benefits and shortcomings of AI in a more design-centric role.

CCS CONCEPTS

• **Computing methodologies** → *Natural language generation; Information extraction*; • **Human-centered computing** → **User studies**; *Collaborative and social computing design and evaluation methods*.

KEYWORDS

LLMs, AI-assisted game design, co-creative game design, mixed-initiative creativity, procedural content generation, user study, Unity

ACM Reference Format:

Asad Anjum, Yuting Li, Noelle Law, M Charity, and Julian Togelius. 2024. The Ink Splotch Effect: A Case Study on ChatGPT as a Co-Creative Game Designer. In *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG 2024)*, May 21–24, 2024, Worcester, MA, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3649921.3650010>

1 INTRODUCTION

Despite the rising prominence of artificial intelligence (AI) in both research and retail products, it remains a sensitive and even taboo topic in creative circles¹²³. The surge of AI art has served as a further catalyst to this issue, and perhaps no other field embodies the fear and rejection of newer AI principles than game development⁴⁵⁶. It is a creative discipline comprised of artists, composers, designers, writers, actors, and programmers, all of which are feeling the reverberate effects of AI advancements. It is unsurprising, then, that a large portion of this industry is skeptical—even fearful—of the effect that AI can have.

What becomes important then, not just to the advancement of AI but to its acceptance, is drawing a clear distinction between what AI tools can provide and what still requires a human-refinement. If AI is simply a tool used to improve and expedite certain processes in creative fields, it should be inferior at making cohesive, expressive creative choices compared to a designer with a clear vision for a project. Moreover, if an assistant AI tool is given full autonomy over what decisions to make and how to implement them in a creative medium such as game design, the result should theoretically be functional and structurally sound, but lack creative vision. Consequently, this idea was put to the test—could a general AI tool such as OpenAI’s ChatGPT create a game with genuine, well-defined creative vision with little to no influence from the user? If so, how would those creative choices compare to ones made by a designer instead?

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
FDG 2024, May 21–24, 2024, Worcester, MA, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0955-5/24/05
<https://doi.org/10.1145/3649921.3650010>

¹<https://www.cnn.com/2022/10/21/tech/artists-ai-images/index.html>

²<https://www.forbes.com/sites/joemckendrick/2022/12/21/who-ultimately-owns-content-generated-by-chatgpt-and-other-ai-platforms/?sh=891092b5423a>

³<https://www.ign.com/articles/dragon-age-writer-dismisses-ai-generated-storytelling-bioware-david-gaider>

⁴<https://www.cbc.ca/radio/video-games-artificial-intelligence-1.6974408>

⁵<https://fortune.com/2023/07/25/video-game-studios-scared-ai-forcing-managers-study-machine-learning-offering-employees-7000-bounties-gala-sports/>

⁶<https://www.gameinformer.com/2023/11/22/its-undeniably-going-to-cost-people-jobs-inside-the-game-industrys-fight-over-ai>

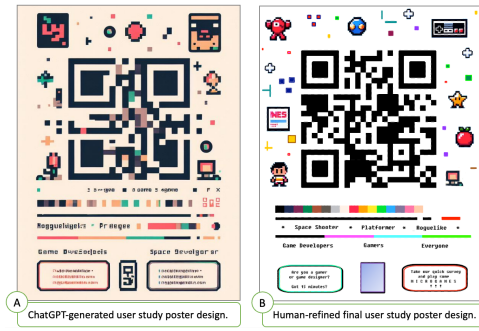


Figure 1: Leveraging LLMs for creative tasks can be highly efficient, however there still remains a need for human-refinement. Much like this poster, where a human refined the AI-generated art into something readable and more aesthetically pleasing, humans can use LLMs as a ‘muse’ to design better games.

This paper attempts to answer these questions by having game designers create a handful of common game archetypes with minimalist gameplay. We then compare their game feel, uniqueness, and cohesion to the same games made only using suggestions and code from LLMs. We asked participants to evaluate the quality, game feel, innovation, and overall preference of three games—a base game, a human-designed game, and an AI-suggested game—across three genres using a ranking scale questionnaire. The results of the questionnaire illuminated the strengths and weaknesses of both methods of pure human design versus AI-assisted design, and how a collaboration between human game designers and a general AI assistant could be beneficial or detrimental to the creative process.

2 BACKGROUND AND RELATED WORKS

2.1 Collaborative Game Design

Collaborative systems allow AI agents to interact in the development process with the game designer by generating or modifying some aspect of the game. In past works, these systems have typically been applied to level design [1] [3] or for developing entire games [13][5]. Computational creativity from these collaborative systems can be applied to any part of the game, such as level design, visuals, narrative, or gameplay [10]. The extent of interactivity of the agent can come in varying degrees of involvement; the AI can act as a guiding system, as an equal collaborator, or can simply model user intent [7]. The design process for these collaborative systems [14] and how they are used by their designers—whether by directly incorporating their content into the game [23] or as a source of inspiration akin to the artistic exercise of Bulletism [2] [6]—has become an active area of research. The benefits and drawbacks of these collaborative AI systems are critical to explore from a human designer and a researcher perspective, particularly as generative systems (e.g. text-to-image, LLMs) become more widely used. We use ChatGPT as an abstract AI-assisted content generator—one intended for the use of designing game feature and game feel content.

2.2 Designer Intent

For creative projects such as video games, the designer’s intent is expressed thoroughly through the different aspects of the game. This expression can take form in the game’s art, music, level design, narrative, and general gameplay—all of which contribute to the overall player experience and feel of the game. Automating game design while maintaining original game designer intent is a challenge, especially when given limited context and understanding on the AI’s part. Nelson et. al. first explored how the communication and discrepancies between AI game design assistants and human users affect the quality and experience of a game [13]. This was later expanded by Treanor et. al. to build entire games from user ‘ideas’ in a mixed-initiative process [21]. Various other works have looked to model and expand the creative intent of the designer and apply the output to generative game content [9] [8] [15] or games themselves [20] and then evaluate them on the player experience. In this experiment, we use ChatGPT to examine how well an LLM can understand and contribute to the game development process based on a human designer’s original intent. We prevent the LLM from being involved directly with the output product, and rather implement the game features and code suggestions through chat communication.

2.3 LLM-Assisted Game Content Generation

The development and growing accessibility of large language models (LLMs) for the use of content generation has become increasingly relevant in game design. Most of the use cases of LLMs in PCG—which use the iterations of the GPT and LLaMA models—are used to generate game assets and more abstract content such as suggesting game features [4]. Concretely, LLMs have been used to generate levels for previously researched PCG domains such as Sokoban [19] and Mario [16]. LLMs have also been used for narrative generation to facilitate story telling [17] and quest generation [22]. For data augmentation, these models can be applied to text-to-asset input prompts to train other generative models such as for sprites and levels [12].

Unlike previous works, where the output of the LLMs pertained to a single aspect of the game, we apply the output of the LLM in the context of the entire game design process. With this system, we use ChatGPT’s GPT-3.5 model to suggest game features, suggest game feel elements, and to generate code in the C# language for the Unity game engine. We combine both the abstract and concrete content generation capabilities and apply them directly to the game design process.

3 METHODOLOGY

To test the suggestive game design capabilities of ChatGPT, a series of 2D games for three game genres—arcade space shooter, platformer, and turn-based roguelike—were designed. Each genre had three games: a baseline game; a pure human-designed game; and a game made from ChatGPT suggestions. The games were designed with minimalist graphics, where players and NPCs were made up of simple shapes and colors, to prevent any visual bias across genres.

These genres were chosen to test ChatGPT’s capabilities for general game domain knowledge and how well it could apply context-relevant design suggestions. The platformer was chosen due to its

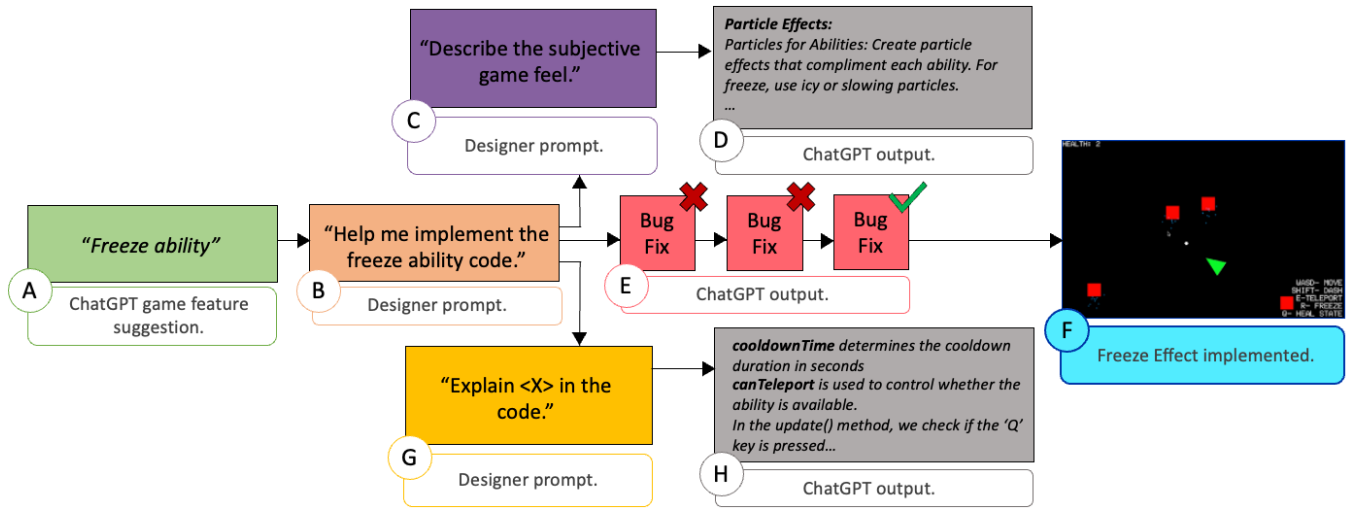


Figure 2: Example of the freeze ability being implemented for the space shooter genre game with guidance from ChatGPT. The freeze ability is (A) first suggested by ChatGPT from an initial prompt. (B) The user then asks ChatGPT for help with implementing the code in Unity engine in C#, (E) the output code from ChatGPT is attempted to be implemented 3 times by the user until it is successfully integrated into the game without error. The user also asks for (G) explanations about the generated code and (C) further suggestions to add game feel.

popularity for Unity game design tutorials. The space shooter was chosen to represent a classic genre for arcade games. The roguelike was chosen as a genre of game with the most complex mechanics (i.e. procedurally generated maps and AI behaviors) and a different time signature (being turn-based). These selections attenuated potential bias in the user study from participants who had a preferential genre of game to play.

The AI trials mimicked a situation where a developer had a basic gameplay loop in mind. These designers would then turn to ChatGPT for a comprehensive tutorial of how to implement their idea and what mechanics or abilities could be added to improve gameplay. We assume for these experiments that the developer had blind faith in the ideas presented to them, and if a range of items or ideas were ever provided, they would simply ask which of the options was most recommended. ChatGPT was prompted for everything from high level subjective questions (e.g. “What abilities should I add to my game?”) to low level code implementations (e.g. “Can you provide code and instructions on how to apply this in Unity?”) to add features and improve the game feel of each game. Author Steve Swink describes “game feel” as “the tactile, kinesthetic sense of manipulating a virtual object” and “the sensation of control in a game” [18]. The authors performed the role of novice human designers for this study and directly implemented ChatGPT’s responses (without regard to personal opinion of the output) into the games. In addition, improvements to game feel and general bug fixes were limited to suggestions exclusively from ChatGPT. A number of limitations were placed on prompting ChatGPT to ensure the prompter’s bias or influence was kept to a minimum. The prompter was only allowed to request code implementation for up to 5 different features for each game. If a bug arose from any of the generated code, up to 3 attempts could be made to ask the program how to fix the error, after which its failure to do so would

be noted. The prompter could request 1 general explanation of the provided code and could ask once which game feel elements should be associated with the feature.

The following subsections describe the distinguishing features and creative development process from implementing the 9 games.

3.1 Space Shooter



Figure 3: Screen-captures of the 3 shooter games created for the experiment.

The prompt for the first game was to “Create a fast paced, top-down shooter with 3 special abilities. Each ability should have a short cooldown. The goal is to survive for as long as possible while enemies randomly spawn into the scene and chase the player”.

3.1.1 Base Game. The base shooter game included standard top-down movement as well as a short ranged omni-directional dash on a brief cooldown. The player would be oriented towards the mouse position on screen, and a button press would fire a bullet in the same direction. Pressing and holding the mouse button would fire bullets at a constant rate. Enemies would spawn into the edge of the scene at set intervals, from any one of 9 evenly spaced spawn points. Enemies would follow the player’s position at a constant speed and destroyed when they either collided with the player, in

which case the player’s health would drop, or shot by a bullet. The game featured a black background and simple geometric shapes for the player, bullets, and enemies, with no particle effects or generally accepted elements of game feel. The only user interface (UI) element was a countdown on the top left indicating the player’s current health.

3.1.2 Human-Designed Game. The initial design process began with adding general game feel elements to the base game. This included shaking the screen any time an enemy was hit, adding appropriate sound effects for shooting and destroying enemies, and triggering an explosive particle effect when enemies were killed. During playtesting, we felt that taking damage was not effectively communicated to the player, despite the health counter updating on the UI. We resolved this by incorporating a short, screen-wide glitch effect during enemy-player collision. The more explicit visual cue influenced future decisions regarding visual design and ramping difficulty, and its clarity in frantic scenarios supported a more chaotic gameplay loop. This constant re-assessment of the game’s direction through multiple iterations of testing and refinement was vital to the human design process. It presented an important factor in delineating between human and AI driven games: the ability to ask *why*.

A screen-wide freeze ability—which would stop all enemies currently on screen in an instant—was first implemented into the game. In its original state, enemies would suddenly stop moving when the associated key was pressed, offering little to no feedback to the player. As the first of many changes to rectify this, a strong screen shake effect was added to signify that a powerful attack had been activated. Frozen enemies would also turn light blue as opposed to their standard red, a faded blue vignette would appear along the edges of the screen accompanied by the sound of ice cracking and forming, and destroying frozen enemies would trigger a new “shattered” sound effect. Each of these additions were intended to give the ability a distinctly “cold” and elemental nature.

Once this ability had been set, it felt natural to continue along the “elemental” theme for the game. A sweeping fire attack—which destroyed enemies on impact—and a strong outwards wind blast—which pushed all enemies away from the player—were implemented to complement the freeze ability. Both were given their own screen-wide colored vignettes to match the freeze effect (orange for fire, white for wind) as well as appropriate sound effects and specialized particle effects. The wind ability pushed white lines from the center outwards, and the fire ability produced an arc of flames around the player.

With the game mostly complete, playtesting and refinement were once again responsible for many small but important changes to give the game its final look and feel. Cooldowns for abilities were visually indicated on the UI and each skill had a different cooldown time based on how powerful the ability was. To add to the slightly frenetic nature of the game, a slight amount of screen shake was invoked every time the player fired a bullet. A faded red vignette would appear when the player was hit, and the colors of both the background and the enemies would change upon taking damage.

3.1.3 ChatGPT Guided Game. The initial task for the ChatGPT guided game was to recreate the entirety of the base game, noting any failures or differences in implementation from the original. It

was able to accurately rebuild most the game from scratch, with detailed descriptions of how to add objects, code, and components such as line renderers in Unity. In some instances, it actually employed better practices than the base game, including setting up a more robust enemy spawning system. However, the program did run into roadblocks. For instance, it failed to correctly diagnose why the player’s rotation did not exactly match the mouse position. While the process was fairly harmonious, proper execution required a more in-depth understanding of Unity and the general game making process.

Once the base game was set, we asked ChatGPT to suggest a list of potential abilities for a “top-down, spaceship shooter game”. It provided a list of 20 options, including:

- **Teleportation:** Instantly jump to a different location on the screen, useful for escaping dangerous situations or repositioning strategically.
- **Nanobot Repair:** Activate a healing ability that slowly restores your ship’s health over time.
- **Time Manipulation:** Slow down or temporarily freeze time to dodge bullets, aim precisely, or gain a tactical advantage.

While the code for each ability was provided on request, a number of issues were met in the process. For instance, teleportation did not achieve the desired effect that ChatGPT itself outlined in its description, as it would often teleport the player directly to an enemy or to the same position they already were. A later issue with the same ability would cause the player to disappear every time it was activated. Despite multiple attempts to solve this problem, ChatGPT was not able to resolve the issue, and the problem was fixed manually by recognizing the player’s location on the z-axis was being changed during teleportation.

The model struggled most with incorporating game feel elements for the abilities. It often gave vague instructions for adding particle and sound effects, contrary to ChatGPT’s in-depth explanation on utilizing the line renderer component (used as an accessory to the dash function in the base game). Basic suggestions such as having the teleport look like “a burst of stars or particles that appear and disappear quickly” were provided, but without detailed instructions on how to achieve them. As a result, the final game contained a significantly lower count of additional game feel elements than the human-designed counterpart. The ChatGPT log for the space shooter genre game can be accessed here⁷.

3.2 Platformer



Figure 4: Screen-captures of the 3 platformer games created for the experiment.

⁷<https://chat.openai.com/share/f807763a-6a25-4fd3-85de-adb5f1de8159>

The prompt for the second game was to “Create a simple platformer where you need to reach the flag to win. Develop 3 transformative mechanics to the player which change their platforming. Design a short level that takes these abilities into account”.

3.2.1 Base Game. The base platformer game contains minimalist controls and visual design. It contains only horizontal movement, gravity, and a jumping mechanic to allow the player to traverse floating platforms found in the level. The player must reach the flag object found in the level to complete the game. The level is designed similar to that of a pyramid, where platforms are placed intermittently vertically and horizontally in the space. If the player falls off the level, they are respawned to the original starting point. Like the base space shooter game, the base platformer game contains no particle effects, sound effects, or any other elements of noticeable game feel. The only UI is a transparent screen that tells the player when they have won the game.

3.2.2 Human-Designed Game. Once again, the design process began with implementing a layer of general game feel elements to the base game. The original jump function would sometimes cause the player to fall when attempting to jump at the edge of the platform. This was due to a lack of “coyote jumping”, which describes the phenomenon of a player’s ability to jump for a split second after moving off of the platform. An innate understanding of platforming design allowed responsiveness to be improved.

“Transformative mechanics” in the platformer prompt was interpreted by the designer as the player having shape-shifting abilities. Much like the elemental abilities in the shooter, this allowed for a level of coherence in potential abilities while sacrificing their range of options. Shrinking and growing abilities were both added, each with their own control inputs. When small, the player could traverse thin crevices, but suffered from a much smaller jump. When big, they would be unable to pass through certain environments, but were able to jump proportionally higher. The surrounding level was altered to encourage the use of these abilities, and the camera scaled accordingly to keep the player in frame. Particle effects were added to both transformations, with a clump of small particles moving towards the player during the “growing” animation and particles moving outwards during the “shrinking” animation.

The final ability allowed the player to extend their jumps by descending at a slower rate than normal. This could only be activated while the player was in the air, and would instantiate a small object mimicking the shape of an umbrella or hang glider on top of the character model. A small effect was added to the top of the player, where particles would push out upwards and then drop down off to the side, as if the “hang glider” was protruding directly from the top of the player’s head. Additional improvements such as linearly interpolating camera zoom for smoother movement rounded out the remainder of the game.

3.2.3 ChatGPT-Guided Game. While ChatGPT was once again able to recreate the base game eventually, its poor understanding of the game’s context became an even bigger issue for the platformer. It would often provide code for a 3D project- despite multiple reiterations that the game was in 2D- and was incapable of producing a reliable method of following the player with the camera. It also listed a time manipulation ability (similar to that found in the space

shooter, which would pause all enemies on screen for a brief period of time) as one of its 3 suggestions for “transformative mechanics”. This was despite the game not featuring any enemies. As a result, only 2 abilities had any real functionality in the final game.

The first of these abilities was size manipulation, which was notably similar to the first two abilities in the human-designed game. The implementation was generally seamless, albeit missing many additional effects present in the human version and only featuring two states: “normal” and “giant”. The second ability allowed the player to flip gravity, however this only worked in one direction when implemented exactly as described. This led to another manual fix by the prompter, who recognized the need to flip the player character over the x-axis when on the roof for the code to operate correctly.

Unique to the platformer game, ChatGPT was also asked to give suggestions for level design given the transformative mechanics at play. These suggestions were generally indeterminate, but occasionally relevant, such as advising that the game have a section which was too narrow for the player to fit through if they were in their “giant” state. It did not, however, suggest that the level should feature an area that was not traversable by the “normal” state, which meant the final game could be completed without using the size manipulation ability altogether. Game feel suggestions were once again minimal, apart from a UI element which displayed the player’s size through a slider at the top left of the screen. The ChatGPT log for the platformer genre game can be accessed here⁸.

3.3 Roguelike



Figure 5: Screen-captures of the 3 roguelike games created for the experiment.

The prompt for the roguelike game was to “Create a turn based stealth roguelike where the player must collect a key and then head to an exit to escape, all while avoiding patrolling enemies. The player can collect or recharge up to 3 abilities, but they must be stealth related. The player can also engage in direct combat with enemies”.

3.3.1 Base Game. The base roguelike game takes elements from classic dungeon-crawler games. Rooms of a random size are procedurally generated and connected with corridors using a binary-space partition algorithm. The player’s and enemies’ movement and combat are turn-based, where each step the player enacts the next turn of the game. The player can move in any directional space (up,down,left,right) so long as there is a floor tile. The player can also skip a turn and remain on the same tile. To win the game, the player must find a randomly placed key, survive combat from enemies, and then use the key to unlock a randomly placed door. If the

⁸<https://chat.openai.com/share/e1813d47-d568-475e-9f5f-5975cc42fce0>

player loses too many health points, they lose the game. Enemies are randomly placed in the rooms, move in a random direction every 3 turns, and follow the player when they are within 5 spaces of them.

3.3.2 Human-Designed Game. The roguelike separated itself from the previous two genres largely due to its turn-based nature, which impacted the approach to improving the game. As the first of 3 stealth mechanics, the player would become invisible for a number of turns. Enemies would lose sight of the player and would continue on random patrol paths, even if the player was in proximity. The ability was coupled with a vanishing sound effect and semi-translucent, cloud-like particle effect at the player’s position. The player would then turn translucent themselves for as many turns as the effect held, after which they would turn opaque and attract enemy attention once again. The player was able to attack enemies in the invisible state, but this alerted the enemy to their position and instantly removed the invisibility effect.

The second ability—a smoke bomb—followed a similar structure, briefly limiting enemy sight lines and using a cloudy particle effect. An area around the player was enveloped by smoke, and all enemies inside the effect’s radius would completely stop. Unlike invisibility, enemies were not able to identify the player’s location in the smoke, even when the player was directly attacking them. This added benefit, along with the wider area of effect, made the smoke bomb significantly more effective than invisibility. This was offset by a longer cooldown. Finally, the player would be able to use shurikens for long-ranged combat. While the shuriken ability was available, a line renderer would indicate which direction the player was aiming.

Each ability was given distinct sound effects, and clashing sword sounds were added to the face-to-face combat from the base game to maximize feedback. Enemies would explode into a burst of blood when killed and cooldown timers were added to the user interface. To incentivize combat, killing enemies shortened all cooldown times by a sizeable amount.

3.3.3 ChatGPT Guided Game. Unlike the other 2 genres of games, ChatGPT was unable to recreate the original design of the base game for the roguelike genre. It could not understand the ‘turn-based’ nature of the game and apply it to a real-time Unity game engine context. The LLM was also unable to correctly implement the binary-space partition algorithm for the rooms and the turn-based enemy AI movement. This could be due to the lack of context for the input prompts sent to the LLM, the Unity game engine, or the miscommunication between user intent and what ChatGPT was capable of creating. Therefore, the original base game was used as a starting point for implementing the feature suggestions of the LLM. The ChatGPT log for the roguelike genre game can be accessed here⁹.

Like the human-designed version of the roguelike, ChatGPT suggested to implement a temporary invisibility feature, where the player could walk by enemies without engaging in combat or be attacked in retaliation. It also suggested to implement an energy system into the game. With this energy system, players would need a certain amount of energy points in order to engage in combat and turn invisible. This energy could be restored by

waiting turns. In addition to the random chance to regain health from defeating an enemy, ChatGPT suggested that the player should have a chance to increase their energy restoration rate, decrease their energy consumption, or increase the invisibility time duration as an alternate reward.

Two suggested features from ChatGPT—a shortcut algorithm which would allow the player to take alternate paths from the generated rooms and an enhanced enemy AI patrolling feature—could not be implemented due to bugs and a context misunderstanding in the code provided by ChatGPT. Finally, ChatGPT suggested to add game feel elements that included character animations (implemented in the form of slight idle movement), a UI indicator for when the player became invisible, color changing effects on the enemy for both when they detected the player and on the player for when they were low on health, and proximity-based sound effects when enemies moved on the map.

4 EXPERIMENT DESIGN AND USER STUDY

4.1 Website

A website was hosted on Github Pages to serve as a hub for users to play the games. The games were anonymized and the ordering of the base, human-designed, and ChatGPT-suggested games were shuffled so they could be evaluated by participants for quality, feel, and gameplay. Users would select a genre of game and then be given the anonymized games (labeled A-C) to play in order. This experiment setup was inspired by Mazzone’s art and AI collaborative output experiments, which asked participants to evaluate AI generated art in a Turing Test style experiment [11]. The games’ pages included instructions for how to play and control them below the game window as well as arrows to navigate between games. The participants could play the games for as long as they liked. After 2 minutes, a button would flash orange on screen—either the ‘next’ button for the user to play the next game in the genre or the button to the user study form—to encourage users to evaluate all of the games in the study.

4.2 Survey

We sent out the study as a Google Form survey via Twitter posts, flyers (one of which was featured earlier in this paper as Figure 1 as a co-designed ChatGPT-human creation), and Discord servers. For the first section of the form, we asked general demographic questions concerning the gaming habits, their game design experience, and experience with AI. The demographic survey questions are listed in Table 1. The following sections of the survey were divided based on the three genres of games (space shooter, platformer, and roguelike). The participants did not have to play all 3 genres, but were required to play all 3 games in the genre of their choice. After they played the games on the hub website, participants were asked to rank the games (described as they were on the website as game A, B, and C for anonymity) based on their aesthetic and qualitative evaluations. Finally, the participants were asked to give optional feedback on the games they played. We received a total of 45 responses.

⁹<https://chat.openai.com/share/32afedb3-6ef5-4578-81e7-a81874acdacf>

Table 1: Demographic survey questions asked in the user study separated by general topic.

Q#	Question
Q1	On average, how often do you play video games?
Q2	What are some of your favorite genres of video games to play?
Q3	What type of games do you play more? Independently made games or triple-A games?
Q4	What is your level of game design experience?
Q5	What kinds of games do you design?
Q6	Have you ever submitted a game or prototype to a game jam (either leisurely or for an assignment)?
Q7	Have you ever used an AI-assisted content generation tool?
Q8	On average, how often do you interact with a large language model?

5 USER STUDY RESULTS

5.1 Demographic Information

All of the participants in the study played video games (2 or more hours a week) with 48% of participants playing 8 or more hours on average a week. We asked participants about the types of games they played in the context of publishers: independent (i.e. games published on user-based platforms such as Newgrounds or Itch.io and released by small indie studios such as Supergiant, Innersloth, Developer Digital, etc.) and/or triple-A games (i.e. games published largely on console systems and released by large game studios such as Nintendo, FromSoftware, Riot, etc.) 57% of participants reported to equally play both independent and triple-A titles and 31% played independent games more frequently. 53% of the participants had never designed a game, while 31% had between 1-11+ years of experience making games. 42% of those who had experience with game design had submitted a game or prototype to a game jam, while 24% reported an attempt to submit to a jam but had never finished a game for it. 64% of participants had experience using and interacting with an AI-assisted content generation tool (e.g. text-to-image generators, LLMs). 62% of participants reported interacting with a large language model between 1 to 4 hours a week.

We asked participants to list the genres of games they like to play and, if they designed games, to list the genres they like to design. Platformer games were tied with puzzle games for the most commonly made game genre and the 5th most played genre. Roguelikes were the 4th most frequently made genre and the 11th most played genre. While arcade games were not explicitly said as a genre, we could consider the space shooter game an action game. Action games were the 2nd most played game and the 2nd to least most designed game. From this, we can gather that players were most familiar with action games, while game designers were more experienced with designing platforming games. Roguelikes would not be as familiar to players or designers.

5.2 Game Ranking

For each game genre, we asked participants to rank the three anonymized games from “Best”, “Mid”, and “Worst” based on the following 6 categories of criteria: C_1 , *overall preference*; C_2 , *game feel*; C_3 , *innovation*; C_4 , *thematic cohesion*; C_5 , *most interesting abilities*; and C_6 , *visual presentation*. For example, for the “innovation” category a participant can rank the games as “Best” = A, “Mid” = C, and “Worst” = B for a given genre. Participants were not required to play all 3 genres of games, and some participants chose not to answer every question, therefore each question has a varying number of responses. Our experimental hypothesis was that the human-designed game would outrank the ChatGPT-guided game while both would outrank the baseline game.

To examine the difference in votes across the 3 games for each category, we created a scoring system based on the following equation $S_G = 3V_B + 2V_M + V_W$ where V_B is the number of votes the game received for being “Best” in the category, V_M is the number of votes the game received for being “Mid” in the category, and V_W is the number of votes the game received for being the “Worst” of the category. The “Best”, “Mid”, and “Worst” votes are weighted with values of 3,2,1 respectively and summed to equal the final score S_G for a given game G where $G \in [\text{Human game, ChatGPT game, Base game}]$. Table 2 shows the scores received in the category out of all of the votes (each category received different numbers of total votes). The platformer genre had the most variance in ranking, with 2 categories (game feel and visual presentation) going against the hypothesis. The space shooter genre only had one category deviate (innovation). The roguelike category contained no scores that differed from the hypothesis. Two statistics tests – Friedman and Kruskal-Wallis – were performed on all of the participants and the two demographic groups – designer and non-designer. The table results are shown in the Appendix with $\alpha = 0.05$.

Overall, participants almost consistently rated the human-designed games higher than the ChatGPT-suggested games, and both games were better than the control baseline game with no added game features.

5.3 Participant Feedback and Comments

While optional, we received a total of 22, 20, and 19 comments for the space shooter, platformer, and roguelike games respectively. 55% of the comments left were from participants that reported to have some experience designing games. We performed simple word frequency analysis on the comments using SpaCy¹⁰ to look for commonality in themes and criticisms concerning the game. Some critiques were directed towards individual games while others critiqued the games at a general level. For readability, we refer to the games the users critiqued as their true labels for the following paragraphs.

For the space shooter games, 91% comments left by participants concerned the player “ability”, while 70% of the comments pertained to the “feel” of the game. Participants commented positively on the “cool animations” and “interesting”-ness of the abilities of the game. Many participants even offered suggestions for improvement on the abilities such as adding more enemies and changing the rate of fire. However, some participants had issues with the controls and

¹⁰<https://spacy.io/>

Table 2: Scores for each game and genre using the formula S_G and category criteria Models with the highest score are colored in green, second highest in yellow, and lowest in red.

	Space Shooter			Platformer			Roguelike (%)		
	Human	ChatGPT	Base	Human	ChatGPT	Base	Human	ChatGPT	Base
Overall Preference	110	88	48	97	77	57	87	77	64
Game Feel	108	85	48	86	73	75	92	71	65
Innovation	93	95	49	104	76	48	93	65	52
Thematic Cohesion	91	80	63	98	66	58	79	66	65
Most Interesting Abilities	101	92	47	103	85	46	96	65	55
Visual Presentation	107	83	50	98	56	68	83	75	58

accessibility of the game. Other found some of the sound effects “harsh” and the overall gameplay “easy” or “boring” and the abilities “unnecessary”. Some participants had positive comments about the ChatGPT-suggested game.

For the platformer games, all of the comments concerned the “feel” of the game and half concerned the “level”. Unfortunately, a majority of the comments were negative. Some participants stated the controls were too “confusing” and “horrendous” – particularly in the ChatGPT-suggested and the human-designed platformer games. Although they found the abilities and mechanics “interesting” and “floaty”, many people stated that the games were “not polished”. Some participants said that the base game, while many participants stated had better controls, had “boring” level design. One participant even suspected that “the games may have been by an AI” but could not identify which.

Lastly, the roguelike games were more varied in responses and had only 26% comments pertaining to the feel of the game and 21% related to the “time” of the game. Many of the comments related to the “feedback” given by the game to players and the general “combat.” However, there were some complaints about “broken levels” in the games, a possible generation error from the PCG levels. Many participants explicitly stated that they most enjoyed the free-movement of the shuriken ability from the human-designed game. Many participants stated that the movement of the players and enemies felt “stiff” and commented their “confusion” for the gameplay and saying the game was “unclear”.

6 DISCUSSION

6.1 Human-ChatGPT Game Genre and Game Engine Knowledge Base

As the game developers and designers for the study, the overall quality of the games were naturally limited by our skill levels. ChatGPT, as a general large-language model, can only hallucinate “reasonable” game design decisions and regurgitate relevant code snippets. The Unity3D engine has a large breadth of online resources available to developers through forum sites, in addition to online tutorials and open source code projects that provide code examples. With this information, ChatGPT can suggest code that is relevant to the task at hand and contextual enough for the Unity engine. Had this experiment been done in an engine with fewer online resources or one which does not rely on text-based programming such as Scratch, ChatGPT’s suggestions may not have been as helpful.

Understanding of common tropes, important gameplay elements, and basic problem solving was also an important factor in the human-designed games which separated them from their AI counterparts. Inferring the need for “coyote jumping” in the human designed platformer simply because the jump “felt off” was not a service ChatGPT could provide. In situations where problems would arise in the code implementation, ChatGPT would often struggle with suggesting fixes without pointed prompting from the designer, which required a deeper understanding of programming and certain aspects of Unity. For instance, when recreating the base game of the space shooter, ChatGPT provided the correct method for destroying enemies on collision with the player, but failed to call the method at any point in the script. While the prompter immediately recognized the issue and how to fix it, ChatGPT was unable to identify the problem itself despite being given numerous descriptions of the error.

In many instances, the prompter was forced to incorporate elements against their better judgment in an attempt to role-play as an individual who lacked any design instincts. In some cases, those instincts made their way into the game design without the prompter realizing. Enemy spawn points in the AI-recreated shooter were instinctively placed off-screen so they would not spawn directly into the middle of the scene. This was not suggested by ChatGPT, but was a knee-jerk impulse from the prompter. Divorcing the game-making process from the designer’s instincts entirely to force sole-dependence on ChatGPT’s output proved challenging in itself.

ChatGPT performed best when it was prompted for abstract, high level game feature suggestions. These feature suggestions were genre relevant and specific, occasionally matching the features implemented for the human-designed games. Time manipulation in the space shooter, size manipulation in the platformer, and invisibility in the roguelike were all present in both versions of each game. These suggestions were helpful and sometimes aligned with our creative visions for the game. However, to maintain consistency, we had to rely on ChatGPT’s code interpretation and attempt to implement it directly into the Unity engine.

Conversely, ChatGPT struggled with context, particularly when suggesting code snippets to implement into the game. For the roguelike game, it would not account for the turn-based movement of the game and would recommend real-time movement and actions instead. It also struggled with more complex code concepts such as procedural content generation and enemy AI movement. For the shooter and platformer games, it struggled with Unity’s coordinate system, passing incorrect values for rotation and position

and sometimes mixing code for 2D and 3D games. Therefore, we would recommend ChatGPT as a tool for conceptual game ideas or for writing pseudocode over direct code implementation in a game engine.

6.2 Design Approach

The initial design approach from the human designer included the use of multiple online resources. This included instructions on how to create particle effects and toggle camera shake. While this may bring into question the designers usefulness—even perhaps blurring the lines between this approach and one lead by AI—an important distinction comes in the designer’s ability to pursue the right ideas, and their proclivity to remix existing ideas which better serve their creative vision. For instance, the glitch effect indicating damage taken in the human-designed shooter was the result of a multi-step process: scouring online for chromatic aberration effects; happening upon a tutorial for glitching UI elements; and implementing a modified version of the glitch to cover the screen when the player was hit. This effect also influenced the decision to change the color palette of the game each time the player was hit, almost as if the player was shifting to a different dimension when attacked. The process began with searching for a basic color shifting script, but transformed into something else entirely which better suited the designer’s goals. This made for the early stages to be less of an exercise in creating games elements from scratch, and rather in curating them in an appropriate way. The LLM, by contrast, would remain within the confines of what was asked of it, and could not replicate any semblance of the same vision.

All three human-designed games notably displayed similar patterns during ideation. The designer would often produce an output which took context into account, maintaining a consistent theme and tying each element together with complimenting visual effects. Using splash-screen color effects when activating each ability in the shooter, for instance, or developing abilities which would reasonably fall inside the definition of “shapeshifter” for the platformer. However, they were also often susceptible to a narrow outlook following the initial brainstorming stage. In the case of the shooter, every ability was a screen-wide offensive weapon and no real consideration was given to defensive tools or ones that affected movement. Essentially, while the designer would often attempt to initially “think outside the box”, once they committed to an idea they would then continue along a similar line of thinking for the remainder of the ideation process. While this may certainly be in part the result of the designer’s experience level, this consistently observable behavior presents an interesting dichotomy between the human and AI approach.

The LLM offered a much wider range of suggestions without falling victim to the same pattern. The user study indicates that although the human platformer demonstrated much stronger thematic cohesion, one of the more interesting ideas- gravity manipulation- came from the AI platformer. Depending on the needs of the project, either metric may be more relevant than the other, but having access to both would certainly allow the developer to make the most informed decision.

6.3 ChatGPT as an Ink Splotch

The exercise of analyzing structureless blotches of ink and extracting meaning or images from them is common in multiple fields. In psychology it takes the form of the Rorschach test, where patients identify the images they see within the ink blotches. This is then analyzed to understand their pattern of thinking. The artistic process of Bulletism, where artists will splash a canvas with ink and use whichever images they see as inspiration for their next work[2], is commonly practiced by fledgling artists or those who have reached a creative impasse. In separating the human-driven and AI-driven approaches to game design and then removing the barrier between the two, it became apparent that LLMs such as ChatGPT could fit a similar role for game designers.

In each genre, the user study indicates enough votes for ChatGPT games in the “most interesting ideas” field to suggest that LLMs can suggest compelling enough ideas to explore further. ChatGPT’s ability to provide inspiring design choices was also noted by the human designers themselves after reading its suggestions. Notably, the human designer became aware of their over-focused design process for the top-down shooter when reading AI suggestions such as:

- **Energy Shield:** Activate a protective shield that absorbs a certain amount of damage (a defensive ability).
- **Afterburner boost:** A short burst of speed that allows the player to quickly evade enemy fire or close the gap between them and their targets (a movement ability).
- **Repair drones:** Release drones that repair your ship or restore its shields over time (a passive ability).

The human designer acknowledged they had not considered non-offensive options for abilities, and nearly instantly began to absorb these suggestions into their “elemental” theme for the game. The energy shield suggestion was reimagined as a protective barrier of rocks and dirt, and the afterburner as an expulsion of dark matter. The human designer continued to curate the game and understand the importance of theme and consistency, while the AI provided an uninhibited list of recommendations to jumpstart that creative process.

7 FUTURE WORK

The crux of this experiment was to synthesize a situation which would not often occur outside of the context of research. This was done to isolate the procedures and explore the merits of each without cross contamination. In the future, we would like to tackle experiments which better recreate real-life scenarios of human-AI creative collaborations and how they impact both the developers and the audience. One such study could involve instructing experienced developers to use LLMs in a similar fashion for creative inspiration, but permitting full control over which suggestions they would take and how they would implement them. This would then be recreated multiple times by different developers with the same prompt, with developer feedback indicating the usefulness of the process and the end products being examined for homogeneity. It may also be pertinent to test multiple different LLMs on the same premise, and expand the creative liberties of the models by inciting their help for level design, encounter design, and a more diverse range of gameplay features, particularly in different game engines

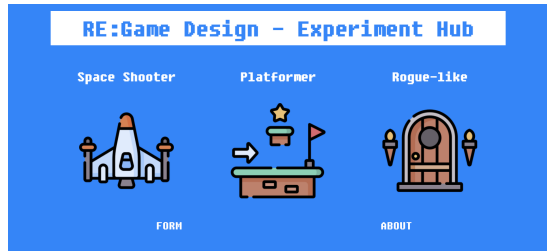
with less online resources such as Godot, CryEngine, and even private, proprietary engines.

8 CONCLUSION

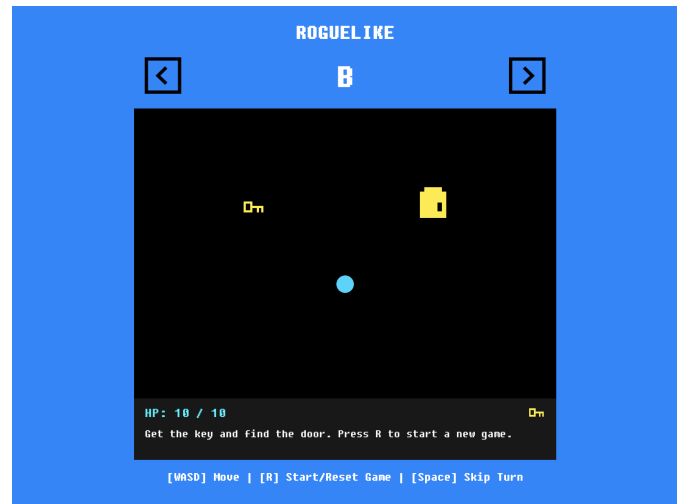
In developing a minimalist game focused on unique game mechanics and maximizing game feel, then reproducing the process using an LLM, the resounding response from a panel of unbiased, blind judges was that the AI driven games could not compare to those made by humans. In all but one metric that was measured by the survey, the human designed games—which had been given the same restrictions and general time constraints as their AI counterparts—ranked the highest. While ChatGPT was able to produce fully satisfactory and structurally sound games in a leading role, it had no direct ability to replace a human designer’s creative goals, understanding of important context, and ability to identify what “felt” good to a player. These findings reject the notion that AI could act as a sufficient replacement for humans in game design, despite their ability to be highly effective tools in that process. However, certain observations from each game- including their ability to innovate and employ interesting gameplay features- indicated that the AI games had enough appeal for players to want to see them explored in full. These results, compounded by the human designer’s own acknowledgment of their shortcomings in ideation and the expanded outlook they gained through exploring ChatGPT’s suggestions, indicate that AI can have an effective role in the creative process of game making without undercutting the work of game developers.

REFERENCES

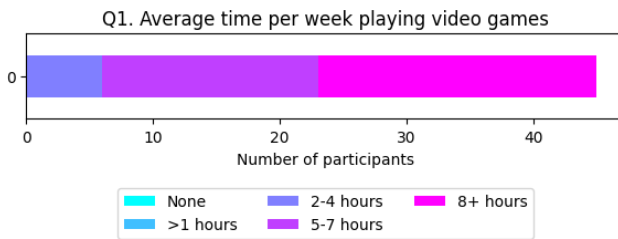
- [1] Alberto Alvarez, Steve Dahlsgog, Jose Font, Johan Holmberg, Chelsi Nolasco, and Axel Österman. 2018. Fostering creativity in the mixed-initiative evolutionary dungeon designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 1–8.
- [2] Team Artsapien. 2023. The art of bulletism: Understanding Salvador Dali’s vision. <https://artsapien.com/2023/02/the-art-of-bulletism-understanding-salvador-dalis-vision/>
- [3] Debosmita Bhaumik, Ahmed Khalifa, and Julian Togelius. 2021. Lode Encoder: AI-constrained co-creativity. In *2021 IEEE Conference on Games (CoG)*. IEEE, 01–08.
- [4] M Charity, Yash Bhartia, Daniel Zhang, Ahmed Khalifa, and Julian Togelius. 2023. A Preliminary Study on a Conceptual Game Feature Generation and Recommendation System. *arXiv preprint arXiv:2308.13538* (2023).
- [5] Michael Cook, Simon Colton, and Jeremy Gow. 2016. The angelina videogame design system—part i. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 2 (2016), 192–203.
- [6] GDC. 2023. . YouTube. <https://www.youtube.com/watch?v=fOqLuWml0UY>
- [7] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
- [8] Kristian Hjalton, Steffen Christophersen, Julian Togelius, and Mark J Nelson. 2015. Game mechanics telling stories? An experiment.. In *FDG*. Citeseer.
- [9] Antonios Liapis, Georgios Yannakakis, and Julian Togelius. 2013. Designer modeling for personalized game content creation tools. In *Proceedings of the AAAI Conference on AIIDE*, Vol. 9. 11–16.
- [10] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2014. Computational game creativity. ICCG.
- [11] Marian Mazzone and Ahmed Elgammal. 2019. Art, creativity, and the potential of artificial intelligence. In *Arts*, Vol. 8. MDPI, 26.
- [12] Timothy Merino, Roman Negri, Dipika Rajesh, M Charity, and Julian Togelius. 2023. The Five-Dollar Model: Generating Game Maps and Sprites from Sentence Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19. 107–115.
- [13] Mark J Nelson and Michael Mateas. 2008. An interactive game-design assistant. In *Proceedings of the 13th international conference on Intelligent user interfaces*. 90–98.
- [14] Nathan Partlan, Erica Kleinman, Jim Howe, Sabbir Ahmad, Stacy Marsella, and Magy Seif El-Nasr. 2021. Design-driven requirements for computationally co-creative game AI design tools. In *Proceedings of the 16th International Conference on the Foundations of Digital Games*. 1–12.
- [15] Martin Pichlmair and Mads Johansen. 2021. Designing game feel: A survey. *IEEE Transactions on Games* 14, 2 (2021), 138–152.
- [16] Shyam Sudhakaran, Miguel González-Duque, Claire Glanois, Matthias Freiberger, Elias Najarro, and Sebastian Risi. 2023. Prompt-Guided Level Generation. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. 179–182.
- [17] Yuqian Sun, Zhouyi Li, Ke Fang, Chang Hee Lee, and Ali Asadipour. 2023. Language as reality: a co-creative storytelling game experience in 1001 nights using generative AI. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19. 425–434.
- [18] Steve Swink. 2008. *Game feel: a game designer’s guide to virtual sensation*. CRC press.
- [19] Graham Todd, Sam Earle, Muhammad Umair Nasir, Michael Cerny Green, and Julian Togelius. 2023. Level Generation Through Large Language Models. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*. 1–8.
- [20] Julian Togelius and Jurgen Schmidhuber. 2008. An experiment in automatic game design. In *2008 IEEE Symposium On Computational Intelligence and Games*. IEEE, 111–118.
- [21] Mike Treanor, Bryan Blackford, Michael Mateas, and Ian Bogost. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the The third workshop on Procedural Content Generation in Games*. 1–8.
- [22] Susanna Värtinen, Perttu Hämäläinen, and Christian Guckelsberger. 2022. Generating role-playing game quests with gpt language models. *IEEE Transactions on Games* (2022).
- [23] Veera Vimpari, Annakaisa Kultima, Perttu Hämäläinen, and Christian Guckelsberger. 2023. "An Adapt-or-Die Type of Situation": Perception, Adoption, and Use of Text-To-Image-Generation AI by Game Industry Professionals. *arXiv preprint arXiv:2302.12601* (2023).



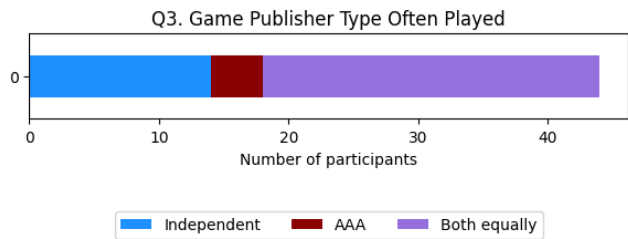
(a) Screen-capture of the website where users could play each of the 9 games and provide feedback.



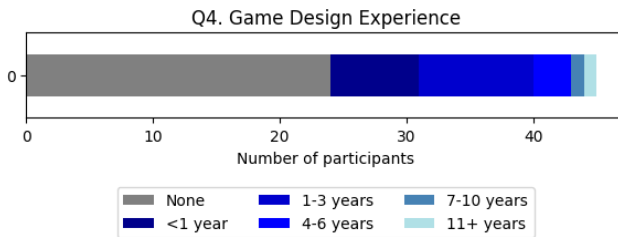
(b) The base roguelike game labeled as 'B' on the website. Instructions and links to the form and home page are located below the game.



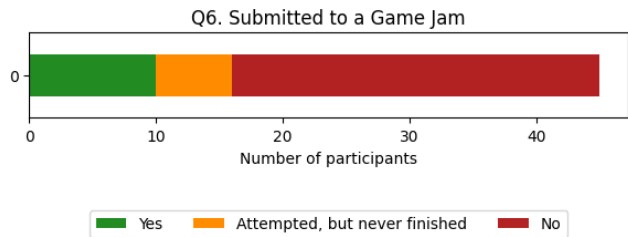
(a) Average time per week participants played video games (Q1)



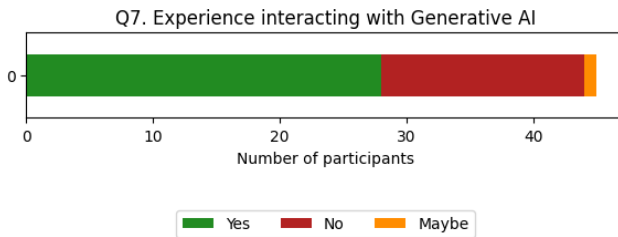
(b) Report of the types of games often played by participants (Q3)



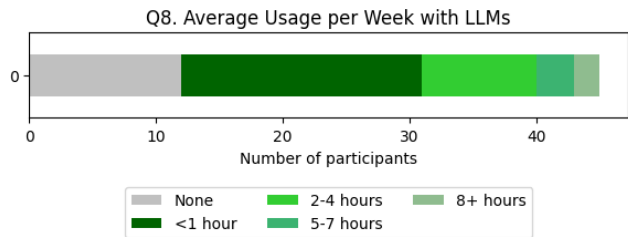
(a) Level of game design experience of participants by years (Q4)



(b) Experience submitting a prototype game to a game jam (Q6)



(a) Level of game design experience of participants by years (Q7)



(b) Experience interacting with LLMs averaged on a weekly basis (Q8)

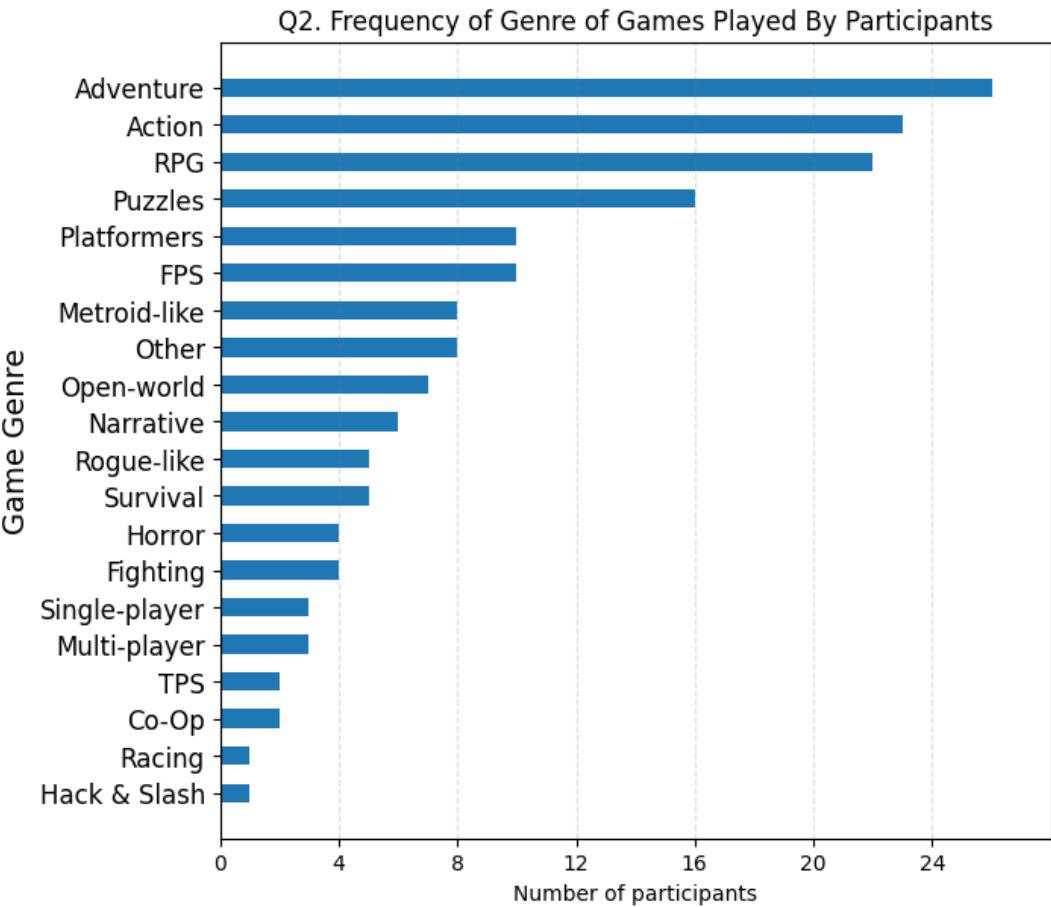


Figure 6: Frequency of preferential genres of games to play from all participants.

Table 3: Aggregated percentage of votes split for the space shooter genre. Cells with less than half of the majority vote and that go against the hypothesis are colored in gray.

Space Shooter			
	% [Best = Human] Vote	% [Mid = ChatGPT] Vote	% [Worst = Base] Vote
Overall preference	70.7	70.7	85.4
Game feel	77.5	72.5	87.5
Innovation	57.5	66.7	87.2
Thematic Cohesion	53.8	35.9	56.4
Interesting Game Features	62.5	60	90
Visual Presentation	77.5	72.5	90

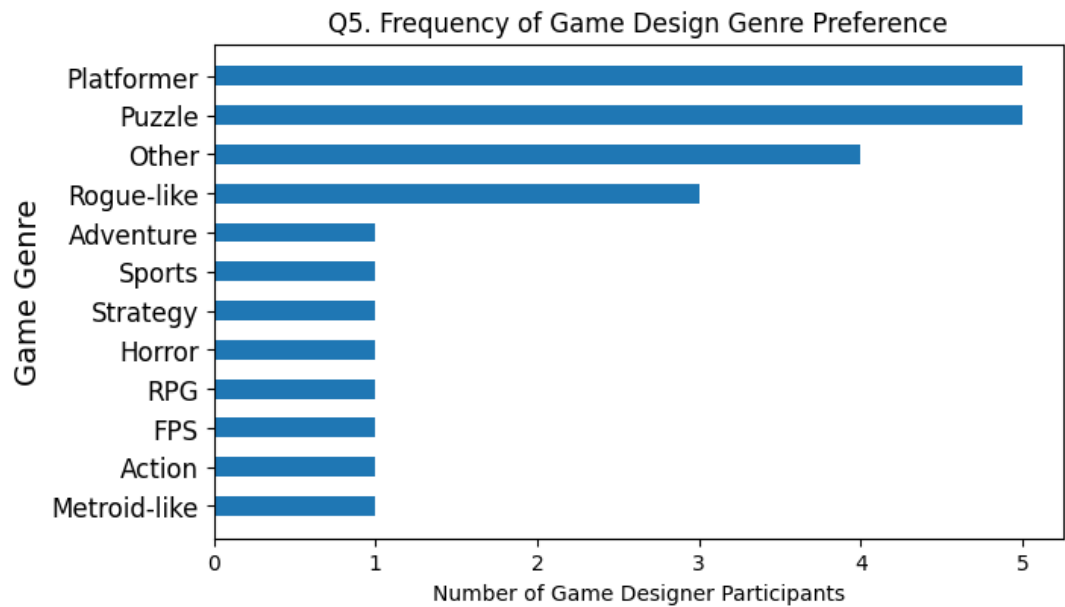


Figure 7: Frequency of reported genres designed from game designer participants.

Table 4: Aggregated percentage of votes split for the platformer genre. Cells with less than half of the majority vote and that go against the hypothesis are colored in gray.

Platformer			
	% [Best = Human] Vote	% [Mid = ChatGPT] Vote	% [Worst = Base] Vote
Overall preference	64.1	47.4	52.6
Game feel	48.7	48.7	33.3
Innovation	71.1	55.3	78.9
Thematic Cohesion	64.9	40.5	51.4
Interesting Game Features	59	43.6	87.2
Visual Presentation	67.6	32.4	32.4

Table 5: Aggregated percentage of votes split for the roguelike genre. Cells with less than half of the majority vote and that go against the hypothesis are colored in gray.

Roguelike			
	% [Best = Human] Vote	% [Mid = ChatGPT] Vote	% [Worst = Base] Vote
Overall preference	71.4	50	60.5
Game feel	63.2	65.8	63.2
Innovation	80	71.4	74.3
Thematic Cohesion	57.1	68	51.4
Interesting Game Features	83.3	72.2	69.4
Visual Presentation	63.9	61.1	61.1

Table 6: Statistic Tests on the Space Shooter Genre games based on all participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Space Shooter (All Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	70.50	78.32	54.076	24.02	71.00	88.34
P-val	4.89e-16	9.81e-18	1.80e-12	6.07e-06	3.81e-16	6.53e-20
F	47.4	52.66	36.37	16.15	47.74	59.4211
P-val	5.09e-11	3.66e-12	1.26e-08	0.0003	4.29e-11	1.24e-13

Table 7: Statistic Tests on the Platformer Genre games based on all participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Platformer (All Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	21.36	4.46	66.75	24.19	62.31	13.58
P-val	2.30e-05	0.1075	3.20e-15	5.60e-06	2.95e-14	1.13e-03
F	14.37	3.00	44.89	16.27	41.90	9.14
P-val	7.58e-04	0.22313	1.78e-10	2.93e-04	7.98e-10	1.04e-02

Table 8: Statistic Tests on the Roguelike Genre games based on all participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Roguelike (All Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	20.03	21.94	51.85	12.48	52.59	27.49
P-val	4.47e-05	1.72e-05	5.51e-12	1.95e-03	3.80e-12	1.07e-06
F	13.47	14.76	34.89	8.40	35.39	18.50
P-val	1.19e-03	6.25e-04	2.65e-08	1.50e-02	2.07e-08	9.61e-05

Table 9: Statistic Tests on the Space Shooter Genre games based on the designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Space Shooter (Designer Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	20.03	21.94	51.85	12.48	52.59	27.49
P-val	4.47e-05	1.72e-05	5.51e-12	1.95e-03	3.80e-12	1.07e-06
F	13.47	14.76	34.89	8.40	35.39	18.50
P-val	1.19e-03	6.25e-04	2.65e-08	1.50e-02	2.07e-08	9.61e-05

Table 10: Statistic Tests on the Platformer Genre games based on the designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Platformer (Designer Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	11.57	0.18	33.60	10.17	28.64	3.13
P-val	3.08e-03	9.12e-01	5.06e-08	6.19e-03	6.04e-07	2.09e-01
F	7.88	0.13	22.88	6.93	19.50	2.13
P-val	1.95e-02	9.39e-01	1.08e-05	3.12e-02	5.83e-05	3.44e-01

Table 11: Statistic Tests on the Roguelike Genre games based on the designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Roguelike (Designer Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	23.01	27.16	41.70	11.75	41.70	22.21
P-val	1.01e-05	1.26e-06	8.83e-10	2.81e-03	8.83e-10	1.50e-05
F	15.65	18.47	28.35	8.00	28.35	15.13
P-val	4.00e-04	9.75e-05	6.97e-07	1.83e-02	6.97e-07	5.20e-04

Table 12: Statistic Tests on the Space Shooter Genre games based on the non-designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Space Shooter (Non-Designer Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	33.17	46.06	36.41	22.70	40.42	48.22
P-val	6.26e-08	9.94e-11	1.24e-08	1.18e-05	1.67e-09	3.38e-11
F	22.45	31.18	24.67	15.36	27.36	32.67
P-val	1.33e-05	1.69e-07	4.40e-06	4.61e-04	1.14e-06	8.06e-08

Table 13: Statistic Tests on the Platformer Genre games based on the non-designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Platformer (Non-Design Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	13.64	6.05	30.51	11.81	30.75	15.75
P-val	1.09e-03	4.87e-02	2.37e-07	2.73e-03	2.10e-07	3.81e-04
F	9.24	4.10	20.67	8.00	20.82	10.67
P-val	9.86e-03	1.29e-01	3.25e-05	1.83e-02	3.02e-05	4.83e-03

Table 14: Statistic Tests on the Roguelike Genre games based on the non-designer group participant responses (KW = Kruskal-Wallis Test Statistic; F = Friedman)

Roguelike (Non-Design Participants)						
	overall	feel	innov.	cohesion	interest.	present.
KW	1.86	1.86	13.15	5.36	14.89	12.57
P-val	3.94e-01	3.94e-01	1.40e-03	6.84e-02	5.86e-04	1.87e-03
F	1.26	1.26	8.94	3.65	10.11	8.53
P-val	5.32e-01	5.32e-01	1.14e-02	1.61e-01	6.37e-03	1.41e-02