**Proceedings of the ASME 2024**
**International Design Engineering Technical Conferences**
**IDETC2024**
**August 25-28, 2024, Washington, DC, USA**

**IDETC2024-143923**

# DESIGNING FICTIONAL WORLDS FOR PLAY THROUGH LARGE LANGUAGE MODELS

**Ada-Rhodes Short[1]**

[1]University of Nebraska Omaha, Omaha, NE

## ABSTRACT

*This paper explores the application of Large Language Models (LLMs) in addressing the challenges of procedural generation in video game design, specifically tackling the Oatmeal Problem and the Bach Faucet Problem. These issues refer to the generation of endlessly similar, unengaging content and the devaluation of content due to its infinite producibility, respectively. By integrating design thinking, computational-cognitive heuristics, and prompt engineering with modern machine intelligence tools, we propose a novel approach termed Procedural Woodworking (PWw). This methodology aims to generate unique worlds and narratives through LLMs while enhancing player engagement by embedding a player's character into the game's narrative in a meaningful way. We utilize OpenAI's ChatGPT-4 to create custom GPTs for world-building and narrative generation, evaluating their novelty and logical consistency through comparative rating scales. Our results indicate that both Top-Down and Bottom-Up world-building methods can produce logically consistent and structurally coherent outputs. The Procedural Woodworking approach, in particular, shows promise in generating novel narrative content that aligns with pre-made human-designed fictional worlds, potentially mitigating the identified procedural generation problems. This study lays the groundwork for further investigation into how LLMs can enrich game design and player experience by offering a more personalized and value-driven narrative generation process.*

Keywords: Large-Language Models, Artistic and Creative Design, Generative AI, Video games

## NOMENCLATURE

Place nomenclature section, if needed, here. Nomenclature should be given in a column, like this:

| | |
|---|---|
| PWw | Procedural Woodworking |
| LLM | Large-Language Model |
| GPT | Generate Pre-training Transformer |

## 1. INTRODUCTION

Procedural generation is an increasingly common element of video game design, but while it promises to generate infinite adventures and endless expanses, it can often only produce endless meaninglessly different content with no novelty, or what one scholar called “oatmeal.” Additionally, there is the problem that even if something unique and superficially novel is created, it will still lack value due to the lack of scarcity and the ability of a user to generate more infinitely; this is described as the Bach Faucet problem.

This paper attempts to address both problems through a combination of design thinking, application of the Large-Language Model computationally-cognitive heuristics, prompt engineering, and modern machine intelligence tools. This approach is twofold. First the Oatmeal Problem is addressed by seeing to what extent a LLM can generate unique worlds and narratives. Second, an approach is put forward to address the Bach Faucet Problem by demonstrating how a player could use an LLM to write a character into the game's narrative through a process we call Procedural Woodworking.

### 1.1 Motivation

Videogames are an increasingly important part of popular culture, impacting those who play them and providing the basis for TV shows, books, movies, and major motion pictures. Approximately 70 % of American adults play video games, and the trend is moving upward [1]. It is also projected that in 2024 video games will generate $282.3 billion in revenue in the US [2]. Its seems like the design science community should probably start to take playing games more seriously.

The Oatmeal [3,4] and Bach’s Faucet [5] problems limit what can be done as bigger and more complex worlds are desired. Procedural generation is important because if it could be done well, it would allow for better labor practices, as smaller teams could achieve bigger artistic endeavors.

### 1.2 Broader Impacts

While this problem is focused on video gaming, narrative generation, understanding, and modification are important parts of the bigger picture of Machine Intelligence. In humans, storytelling is a critical part of how our brains process reality [6,7]. Improving machines' ability to understand stories could enhance their ability to navigate the complexities of the real world.

### 1.3 Background

Large-Language Models (LLMs) are an increasingly common approach to various problems [8,9]. LLMs are artificial neural networks that most commonly have a decoder-only transformer-based architecture, though many LLMs on the market are proprietary and do not publicly disclose their inner workings. One of the most prolific LLMs is OpenAI's chatGPT. ChatGPT is a Generative Pre-trained Transformer (GPT) which enables it to perform unsupervised language learning tasks with high accuracy and minimal training [10–12]. OpenAI's release of ChatGPT and tools for streamlining the creation of GPTs has led to a rapid expansion of applications of LLMs to a wide variety of applications [13].

Prompt engineering is the process of designing text prompts to be used with generative AI tools to create consistent and desirable outputs [14,15]. A prompt within this context is any natural language input to generate an AI model. When working with a GPT specifically there are a few different forms of prompts that get used in different ways. The most obvious form of a prompt is the message put into the chat box. This is often just referred to as a prompt as it prompts the GPT directly. In addition to the text entered into the chatbox, ChatGPT allows for additional attachments to be included with message prompts. Message prompts and their attachments are only read and responded to once. The next most common form of prompt is called Instructions. Instructions describe what the GPT should do generally in natural language. This could include describing what it should do in certain circumstances, what it should model its behavior off of, keys for interpreting specialized message prompts, or multistep directions on complex operations. The directions are referenced every time the GPT generates a response. The final form that can be introduced to GPTs is Knowledge Files. Knowledge Files are natural language files that are uploaded to the GPT and trained into the model. During the initial training the GPT effectively "skims" the contents of the knowledge files and incorporates that information into its responses to every prompt. The GPT can also search the Knowledge Files in more detail if prompted with Instructions or a Message Prompt to do so.

Procedural generation is the process of creating game data algorithmicly [16]. The earliest examples were mostly roguelike games such as Rogue and Beneath the Apple Manor. The approach has two major advantages: 1) it allows for a lot of gameplay or graphical elements to be created without taking up more memory, and 2) it allows for unique randomized gaming experiences. While some form of procedural generation has been around for 40 years, it began to receive increased attention in the 2010s when games like Dwarf Fortress and Minecraft allowed for massive, infinitely unique gaming experiences to be created by small indie developers. This lead to a massive elevation and proliferation of procedural generation in the industry [16–18]. However, this proliferation quickly highlighted a major problem with precedural generation which is infinite random variation does not create a necessarily meaningful gaming experience. This problem has come to be referred to as The Oatmeal Problem [3,18]. Recently, game scholars have identified a second issue with procedural generation: regardless of how beautiful the generated content is, it loses artistic value if the player knows they can just create more instantly. This is referred to as the Bach Faucet problem [5].

## 2. MATERIALS AND METHODS

To perform this work, OpenAI's ChatGPT 4 was used to create custom GPTs. The GPTs, along with a prompt-engineered script, were used to generate six fictional worlds, which were rated using a comparative rating scale for novelty and logical consistency. Finally, a new prompt-engineered script was created for writing a player-generated character into an existing game world and narrative.

### 2.1 Large Language Model

The LLM used for the work was OpenAI's ChatGPT-4. This was used along with the MyGPT feature to generate three separate GPTs for fictional world-building and narrative generation. The first GPT was given knowledge files and instructions on how to perform a Top-Down world-building method. The second GPT was given knowledge and directions to perform a bottom-up world-building method. The third GPT was given knowledge and instructions on how and when to employ each of the methods as well as knowledge of a pre-made human-created fictional world for it to work with.

The knowledge files and instructions are in the APPENDIX 1, along with a link to access the GPTs for review.

### 2.2 Prompt Script

Prompt-engineered scripts were created to generate consistent results for evaluation. We made three scripts for the study. The first script was used to walk through the Top-Down world-building process. The second script was used to walk through the Bottom-Up world-building process. The third script was created to walk through the Procedural Woodworking process. All of the prompt scripts used in this paper can be found in APPENDIX 1.

### 2.3 Comparative Rating Scales

To evaluate the procedurally generated worlds a form was created to rate each world on logical consistency and novelty. Both of the scales were from 1 to 3. For the logical consistency scale we assigned codes of 1-Inconsistent, 2-Fuzzy Logic, and 3-Consistent. For the novelty scale we assigned codes of 1-Derivative, 2-Fuzzy, and 3-Novel. The raters were instructed to review the world summary documents found in APPENDIX 2 and to rate the worlds on novelty across 19 categories and consistency across 16 categories. The entire tool used to rate the world can be found in APPENDIX 3. For this paper, the raters were two undergraduate computer science students with extensive experience with video games and fantasy narratives. One student was male, and the other was female.

Because the number of reviewers was low (n=2), the results will be classified into five categories. This will allow for comparison while not overemphasizing a small fractional difference between quality that may be a statistical anomaly. Scores will be classified as Overall Unsuccessful, Moderately Unsuccessful, Ambiguous Result, Moderately Successful, and Overall Successful. Table 1 summarizes the classification ranges.

*Table 1: Classification of Rating Results*

| Classification | Range |
|---|---|
| Overall Unsuccessful | $1 \leq \text{rating} \leq 1.33$ |
| Moderately Unsuccessful | $1.33 < \text{rating} \leq 1.66$ |
| Ambiguous Result | $1.66 < \text{rating} \leq 2.33$ |
| Moderately Successful | $2.33 < \text{rating} \leq 2.66$ |
| Overall Successful | $2.66 < \text{rating} \leq 3$ |

### 2.4 Development Process

The development of this method began by first setting if a GPT had previously been created for world-building. The most used GPT at the time was World Builder [CITE] and had 200 conversations logged at the time. We began out evaluation by selecting the suggested prompt of "Can you create a new world for me?" which was answer by listing out prompts on individuals aspects of world-building. I then tested the other recommended prompt "How do the inhabitants of your world communicate" and I was again met with a response describing how the human user could answer that question. As neither of these responses was generative and were more aimed at a walk-through of the world-building process, I moved on to my own development.

My first attempt was largely inspired by the book 101 World Building Prompts [19], but lacked the depth and nuance of the books approach. The instructions that I entered for the GPT were:

*Perform the following steps to create a fictional fantasy world*

*1. Generate a Fantasy Conceit:*

*A fantasy conceit describes how the world you are creating deviates from the real world we are all familiar with. A fantasy conceit should be classifiable as exsecting, unchanged, divergent, or additive.*

*Exsecting conceits occur when something is removed from the world. For example, a world without standing water, electricity, the moon, or horses.*

*Unchanged conceits are when the world mimics the real world, but the names of individuals or locations are changed.*

*Divergent conceits are when the world takes a different course from the real world in one major way, but things otherwise remain the same. For example, it could be a world where humanoids evolved from dinosaurs instead of apes, or it could be a world where the Allies lost WWII.*

*Additive conceits are when something is added to the world that does not exist in the real world. For example it could be a world with dragons, magic, faster than light travel, or impossible technologies.*

*Based on that information randomly generate a conceit and use that conceit to complete the following steps. Briefly describe the fantasy conceit in 1-3 sentences.*

*1. Describe the World:*

*Based on the selected fantasy conceit, describe how each of the following topics would be changed from the real world. Briefly summarize the differences in a paragraph for each topic.*

*The topics are:*

*a) Geography*

*b) Biology*

*c) Physics/Magic*

*d) Metaphysics*

*e) Technology*

*f) Culture*

*2. Describe Analog Cultures:*

*Based on your answers to step 1 and 2 select three analog cultures that might exist in the world. Analog cultures are real-world societies and time periods that inspire the creation of the fictional world's inhabitants. Analogue cultures should be selected that would hypothetically be able to thrive in the fictional world. Briefly describe why each analog culture was selected and what their strengths and weaknesses would be when trying to survive in the fictional world.*

*3. Create Fictional Cultures:*

*For each of the selected analog cultures, create a new fictional culture that is better adapted for living in the fictional world. This process should consist of the following sub-steps*

*a) consider how the culture's weaknesses in the world would have influenced their culture and history*

*b) consider how the culture's strengths in the world would have influenced their culture and history*

*c) list 3-5 ways that the culture would have adapted to respond to their strengths and weaknesses in the world*

*d) describe the culture's primary motivating values and morals within the fictional world*

*4. Develop the Cultural History*

*For each of the fictional cultures, you will need to create a fictional cultural history that describes how they developed and explains why they hold the values and norms* *that they have. These should be inspired by the analog cultures but can partially or wholly diverge from the analog culture. For each fictional culture, the process should consist of the following sub-steps:*

*a) Describe where in the fictional world the culture lives*

*b) Describe how, why, and when the culture was founded*

*c) Describe who the primary founders of the culture were*

*d) Describe what major events occurred in the culture between the time of its founding and now*

*e) Describe how those major events changed or influenced the culture*

*f) Describe the hierarchy within the culture*

*g) Describe how the culture is governed and by who*

*h) Assign a name to the fictional culture that is reflective of its history, development, geography, or other important cultural aspect*

*5. Establish Cultural Relationships*

*You need to describe how each of the three cultures relates to each other. Describe how they relate to each other in a few sentences.*

I then prompted the GPT with "Generate a new fictional world for me." This resulted in the GPT following the world creation prompts as described, but it did not complete the process. I was discovered after some experimentation that if I prompted it with "please continue" that it would automatically move on to the next step until the entire process was completed. I used this method to generate three test worlds. These outputs can be found in APPENDIX 4. While this was generally successful, it seemed to fixate on a few specific tropes and analog cultures, and the formatting of the results was inconsistent.

I continued to iterate on the instructions, and took careful notes on the outputs and their formatting. When the results were consistent, I tried starting the process with a new prompt "*Generate a fictional world for me based on the conceit that superheroes are real. The analog cultures should all be based. off of real-world 21st century cultures.*"

The GPT successfully created a fictional world that followed the prompt. This validated two assumptions: 1) the GPT can work with defined constraints, 2) given a constraint it can break out of its default design fixations. A log of this conversation can be found in APPENDIX 5.

I then turned my attention towards recreating these results using a Bottom-Up method. The overall steps for this methods instructions were:

1. *Generate a fantasy conceit*
2. *Generate a central character based on the fantasy conceit*
3. *Generate ten characters with connections to the main character*
4. *Generate ten more characters with a connection to any character that has been generated already*
5. *Generate ten more characters with a connection to any character that has been generated already*
6. *Based on the characters you have established so far, generate a settlement for them to line in*
7. *Generate a fictional history for the settlement*
8. *Generate a culture for the settlement to exist within*
9. *Generate two more cultures that exist in the world at the same time as the culture previously described*
10. *Describe how the three cultures relate to and interact with each other*
11. *Describe the world the cultures exist in*

The full instructions can be found in APPENDIX 1. This approach benefited greatly from the previous work on the Top-Down method and reached a functional level quickly.

The next step was to improve the functionality and prompt the creation of consistent outputs. To do this we took our rough directions and expanded on them into a full document. This was uploaded as a knowledge file to the GPT. A more streamlined version of the instructions was then created that pointed towards the knowledge file as needed. Finally a block was added to the instructions that described how outputs should be formatted after the world-building process was completed. The full knowledge files and directions for the two GPTs can be found in APPENDIX 1.

The specialized Top-Down and Bottom-Up World Building GPTs were then used to generate three fictional worlds each. The fictional worlds were then evaluated using the Comparative Rating Scales described in Section 2.3.

### 2.5 Procedural Woodworking

The goal of Procedural Woodworking (PWw) was to create a method for a player created character to be written into the game narrative and have unique experiences. The goal of this portion of the work was the address the Bach's Faucet problem of procedural generation by prompting the player to value the world more highly by playing off of their effort heuristic [20]. The approach targets the effort heuristic in the same way that including a single egg in a cake mix improves the perceived quality of the product. By having the player personally craft the materials that seed the procedural generation, they will hopefully perceive the end product as more value and unique.

The general approach to the PWw process is to first have human designers craft a fictional world and narrative and then have a human player create a player character to exist in that world. The LLM then uses a combination of Bottom-Up and Top-Down methods as needed to fill in the gaps in the world and create environments, characters, and quests tailored to the player character's backstory.

The term Woodworking refers to taking an individual's backstory and “blending it into the woodwork”.

For this approach's initial test, a graduate research assistant generated a character concept and followed a prompt script, shown in APPENDIX 1. The research assistant was not given any context or information on the world as to not limit their creativity. The results of this output were rated using the method described in Section 2.3 and evaluated additionally for consistency with the human-generated materials.

After this first test, it was found that [RESULT], which we discuss further in section 3. However, the content generated diverged too far from the human-designed elements, so we determined that this approach was currently infeasible. The issues potentially arose from two sources. First, the LLM does not have persistent knowledge of human design elements because they were uploaded to the conversation as an attachment, not a knowledge file. Second, the character description included elements potentially inconsistent with the designed world.

A second attempt was made, in which, in addition to the knowledge files on world-building, a knowledge file was included describing the human-designed setting, and a few prompts were slightly modified to encourage better drawing on the source materials. The prompt script was then followed again. This time it stayed closer to the described world, but shifted the setting forward to 2050, well after the described plot. Additionally, it didn’t include either the main character’s most or least favorite person in the generated community members. This indicated that including the human-designed world into the knowledge files works better, but does not solve the problem entirely. The chatlog can be found in APPENDIX 5.

A third attempt was made, with changes to the prompts to be more specific to the human-created fictional world and minor changes to the player character for consistency. Up until this point the assumption has been that the prompt scripts need to be as generic as possible to allow for repeatability. However, at this point we are looking at applying what has been learned to a specific use case and therefore it makes sense to tailor the prompts to be more specific. Generally, this involved changing the prompts to reference materials in the human-generated knowledge files specifically, but in some cases more responsive prompts were used to insure that the GPT was performing the task as desired. The prompt script and the chatlog can be found in APPENDIX 1 and 5. The changes made to the player character were very minor. They included appending “*and part-time superhero through the Monument Now app*” to the occupation entry of “barista” and changing the faction entry to “*Monumental gig economy worker*.” In a final working game, the changes could be justified by automatically appending the text or by allowing a few more limited options from a drop-down with explanations of their implications. This attempt was rated comporably to the first attempt on novelty and logic and importantly kept the results bound within the human generated world enough to allow for implementation within an actual game.

## 3. RESULTS AND DISCUSSION

To analyze the results the mean was taken of the ratings across 19 logical consistency categories and 16 novelty categories. Table 2 summarizes the mean ratings.

*Table 2: Raw Results*

| Method | Trial | Logic Rating | Novelty Rating |
|---|---|---|---|
| Top-Down | 1 | 2.84 | 2.31 |
| | 2 | 2.76 | 2.53 |
| | 3 | 2.95 | 1.97 |
| | **Combined** | **2.85** | **2.27** |
| Bottom-Up | 1 | 2.82 | 2.03 |
| | 2 | 2.71 | 2.09 |
| | 3 | 2.76 | 2.19 |
| | **Combined** | **2.76** | **2.10** |
| PWw | 1 | 2.86 | 2.79 |
| | 2 | 2.93 | 2.18 |
| | **Combined** | **2.89** | **2.49** |

As the number of ratings in each category was small (n=2) and subjective, the numerical ratings were then classified on a scale of Overall Unsuccessful, Moderately Unsuccessful, Ambiguous Result, Moderately Successful, or Overall Successful. This allowed for broad comparison without looking at minute differences between results that were not statistically significant. Table 3 summarizes these results.

*Table 3: Classified Results*

| Method | Trial | Logic Rating Classification | Novelty Rating Classification |
|---|---|---|---|
| Top-Down | 1 | Overall Successful | Ambiguous Result |
| | 2 | Overall Successful | Moderately Successful |
| | 3 | Overall Successful | Ambiguous Result |
| | Combined | **Overall Successful** | **Ambiguous Result** |
| Bottom-Up | 1 | Overall Successful | Ambiguous Result |
| | 2 | Overall Successful | Ambiguous Result |
| | 3 | Overall Successful | Ambiguous Result |

| | Combined | Overall Successful | Ambiguous Result |
|---|---|---|---|
| PWw | 1 | Overall Successful | Overall Successful |
| | 2 | Overall Successful | Ambiguous Result |
| | Combined | **Overall Successful** | **Moderately Successful** |

In general, all of the approaches did well, though this is to be expected as extensive iteration was performed before testing. All three approaches scored high on logical consistency and ended up in the Overall Successful category on every trial. On the novelty rating, there was much more room for improvement. With each method scoring an Ambiguous Result at least once. Looking at the combined results, the PWw performed better on average at novelty, with a mean rating of Moderately Successful. Notably, the PWw is the only method that scored Overall Successful in novelty during one of its trials. These results are not definitive, but they highlight the potential for the PWw method and justify further time and resource investment in development.

When examining the PWw more closely, it was also important to determine whether the GPT-generated elements fit within the bounds of the human-designed world. This was examined three separate times with modified conditions. First, we examined the PWw approach with a human-designed world file uploaded as a prompt attachment. Next, we uploaded the human-designed files to the GPT's knowledge. In both of these cases the prompts used were generic and there was no guidance on the player character creation. In the third case, we uploaded the human-designed world files to the GPT's knowledge and we modified the prompts and the player character to fit the PWw process better. For the character, this involved limiting the faction selection to one of the human-designed factions and appending the occupation to also include an in-game role in addition to the character's day job. Table 4 summarizes the iteration properties.

*Table 4: Tested PWw Iterations*

| PWw Iteration | Human-Design Files | Prompts and Player Character |
|---|---|---|
| 1 | Attached to Prompt | Unmodified |
| 2 | Uploaded to Knowledge | Unmodified |
| 3 | Uploaded to Knowledge | Modified |

The first iteration was largely outside of the human-designed world and greatly changed the time period, themes, logic, and history of the setting. The second iteration was better and stayed within the general themes and logic established by the human-designed world but still changed the time setting. The third iteration successfully matched the themes, world logic, history, and time period of the human-designed world. Additionally, the third iteration included more elements of the player character information into the world and generated non-player characters (NPCs) consistent with the player character's backstory. Table 5 summarizes these result.

*Table 5: World-Building Elements Successfully Incorporated by PWw*

| Iteration | Themes | World Logic | History | Time Period | Origins Included |
|---|---|---|---|---|---|
| 1 | No | No | No | No | No |
| 2 | Yes | Yes | Yes | No | No |
| 3 | Yes | Yes | Yes | Yes | Yes |

### 3.1 Discussion

The results of this study are generally very positive and showed that both Top-Down and Bottom-Up world-building methods can be automated through a GPT and used to produce consistent and highly structured outputs. Additionally, those methods can be combined with prompt engineering to meld a pre-made human-designed fictional world with player-generated content.

As this work is in its infancy, these results are preliminary but promising. While significant progress towards a consistently working PWw that could be embedded in a playable game was made, it is important to note that progress on this problem will likely not be linear. This is highlighted by an observation made by one of the reviewers, who pointed out that the final iteration of the PWw presented was the most consistent with the world but also had the simplest relationships.

## 4. CONCLUSION

This study aimed to see if a GPT could generate novel and logically consistent narrative experiences. The major identified issues with procedurally generated content in the past have been that it suffers from a lack of meaningful and intentional difference (The Oatmeal Problem) and that if infinite new content can be generated, it loses its perceived value (The Bach Faucet).

This paper approached this problem by employing a novel method we call Procedural Woodworking (PWw). Procedural Woodworking uses a GPT to combine a human game developer-designed fictional world and narrative elements with player-generated content. Underlying this approach is a combination of Top-Down and Bottom-Up world-building methods commonly employed by writers, game designers, and dungeon masters to craft unique and meaningful narratives.

Over the paper, the PWw was shown to successfully generate novel and logically consistent narrative content that fits within the narrative of a pre-made human-design fictional world. At this stage, it shows great potential to address the Oatmeal Problem. Additionally, by allowing the player to dictate elements of the procedurally generated world, we hope to sidestep the Bach Faucet problem by drawing on the player's innate effort heuristic. We believe that this method has the potential to do that, but we have not yet performed a test that is capable of determining to what extent that is achieved.

### 4.1 Future Work

The next step is to study further how effective PWws are at addressing the Oatmeal and Bach Faucet problem in a larger study. This study will be performed by implementing the PWw method into a character creator and seeing how allowing players to introduce elements into the procedural generation process influences perceived value. After a better understanding of how well the method addresses the Bach Faucet problem is gained, a full game will be developed that uses a Generative Adversarial GPT duo to generate quality content more consistently.

## ACKNOWLEDGEMENTS

The author would like to acknowledge the undergraduate research assistants Evie Harvey and Lawrence Wilson, the graduate research assistant Lauren Bertelsen, and senior colleague Dr. Matt Hale for their help with this paper.

## REFERENCES

[1] "U.S. Video Game Player Share 2022," Statista [Online]. Available: https://www.statista.com/statistics/499703/share-consumers-ever-play-video-games-by-age-usa/. [Accessed: 12-Mar-2024].

[2] "Video Games - Worldwide | Statista Market Forecast," Statista [Online]. Available: https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide. [Accessed: 12-Mar-2024].

[3] Compton, K., and Mateas, M., 2015, "Casual Creators.," *ICCC*, pp. 228–235.

[4] "So You Want to Build a Generator...," Tumblr [Online]. Available: https://galaxykate0.tumblr.com/post/139774965871/so-you-want-to-build-a-generator. [Accessed: 12-Mar-2024].

[5] Karth, I., and Compton, K., 2023, "Conceptual Art Made Real: Why Procedural Content Generation Is Impossible," *Proceedings of the 18th International Conference on the Foundations of Digital Games*, ACM, Lisbon Portugal, pp. 1–3.

[6] Aldama, F. L., 2015, "The Science of Storytelling: Perspectives from Cognitive Science, Neuroscience, and the Humanities."

[7] Herman, D., 2007, "Storytelling and the Sciences of Mind: Cognitive Narratology, Discursive Psychology, and Narratives in Face-to-Face Interaction," Narrative, **15**(3), pp. 306–334.

[8] Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D., 2024, "Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey," ACM Comput. Surv., **56**(2), pp. 1–40.

[9] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R., 2023, "A Survey of Large Language Models."

[10] Floridi, L., and Chiriatti, M., 2020, "GPT-3: Its Nature, Scope, Limits, and Consequences," Minds Mach., **30**(4), pp. 681–694.

[11] OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, Ł., Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, Ł., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Peres, F. de A. B., Petrov, M., Pinto, H. P. de O., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I.,

 

Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C. J., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B., 2024, "GPT-4 Technical Report."

[12] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J., 2023, "GPT Understands, Too," AI Open.

[13] "Introducing GPTs" [Online]. Available: https://openai.com/blog/introducing-gpts. [Accessed: 12-Mar-2024].

[14] Ekin, S., 2023, *Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices*.

[15] Marvin, G., Hellen, N., Jjingo, D., and Nakatumba-Nabende, J., 2024, "Prompt Engineering in Large Language Models," *Data Intelligence and Cognitive Informatics*, I.J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, eds., Springer Nature Singapore, Singapore, pp. 387–402.

[16] Hendrikx, M., Meijer, S., Van Der Velden, J., and Iosup, A., 2013, "Procedural Content Generation for Games: A Survey," ACM Trans. Multimed. Comput. Commun. Appl., **9**(1), pp. 1–22.

[17] Short, T., and Adams, T., 2017, *Procedural Generation in Game Design*, CRC Press.

[18] "'No Man's Sky' Is Like 18 Quintillion Bowls of Oatmeal" [Online]. Available: https://www.vice.com/en/article/nz7d8q/no-mans-sky-review. [Accessed: 12-Mar-2024].

[19] Presley, M., 2020, *101 Worldbuilding Prompts (Forgining Fantasy Realms)*.

[20] Kruger, J., Wirtz, D., Van Boven, L., and Altermatt, T. W., 2004, "The Effort Heuristic," J. Exp. Soc. Psychol., **40**(1), pp. 91–98.

## APPENDICES

Appendix 1-5 can be found on GitHub at:

https://github.com/Ada-Rhodes/Papers/tree/main/IDETC2024A

or by Scanning the following QR Code