Spring 2024

Unveiling New Realms: Enhancing Procedural Narrative Generation and NPC Personalization using AI

Tanishq Chawla
Southern Methodist University, tanishq.chawla.22@gmail.com

Unveiling New Realms: Post-Mortem

Enhancing Procedural Narrative Generation and NPC Personalization using AI

Tanishq Chawla
SMU Guildhall tchawla@smu.edu

Abstract
A System Design Framework used to train GPT - based agents can significantly help improve current Procedural Narrative and Character (NPC) Personality Generation Models in games. This approach can lead to a Richer Gameplay Experience that allows players to craft their own Personalized and Unique stories.

Keywords
Procedural Generation, Generative AI, GPT, Narrative Design, System Design, Character Personality Generation, System Architecture, Unity, LLM.

1 INTRODUCTION

Procedural narrative generation in games while being extremely effective currently faces several issues. Primarily the issues are: Lack of emotional depth, lack of control over the content generated, and repetitiveness. This distinguishes the narratives procedurally generated to those of handcrafted stories significantly. This leads to ultimately failing to connect with the player deeply, narrative incoherence (lack of logical flow in the story), and inappropriate or undesirable story elements being generated. This thesis aims to address these issues by harnessing the power of generative AI using a system design framework focusing on narrative structures and guardrails to enrich a player's experience in narrative rich games.

2 TERMS DEFINED AND THEORIES/ RESEARCH

2.1 Procedurally Generated Narrative

Stories that are created using keywords and algorithms that are completely driven by code.

"An automatic process that creates a narrative over time, where the narrative is not determined before the process begins." [1]

2.2 Artificial Intelligence (AI)

"Artificial intelligence (AI) refers to the simulation of human intelligence by software-coded heuristics." [2]

2.3 Generative Pre-Trained Transformers (GPT)

A Neural Network model is a computer program designed to detect and recognise patterns. It consists of a web of interconnected nodes or "neurons" similar to those found in the human brain. By making adjustments to the connections between these nodes, these models learn and improve over time to make better predictions or decisions when recognising patterns.

Transformers are a type of neural network model that specializes in processing or recognising patterns in sequences of data like sentences in a particular text. This makes transformers very powerful in natural language detection tools like GPTs.

What are GPTs?

"A Family of Neural Network Models that uses the transformer architecture and is a key advancement in AI. It is used to Power Generative AI Applications such as ChatGPT." [3]

2.4 Procedural Generation of Narrative Structures

A thesis study by Emily Rizzo, a software developer graduate from SMU Guildhall, cohort 27 [4], explores and experiments with methods of procedurally generating narratives. The study explores the comparison of good vs bad narrative generation, the risks and limitations that come along with procedurally generated narratives. Her methods of procedural narrative content generation primarily include:

- Directed graphs with nodes and edges to generate narratives to create sequentially progressing stories.
- Emphasizes logical story progression using XML-defined rules for node placement.
- Focuses on ensuring narrative validity and coherence through structured graph algorithms.
- Good narrative generation is marked by logical and coherent story progression, while bad narrative generation results from incoherence and lack of emotional depth.

2.5 ScripTale: Generation of Procedural Narrative

A thesis study by L.R. Galan, a student at Barcelona Tech, investigates procedural narrative generation in games, emphasizing the creation of narratives automatically as the game progresses [5]. The thesis aims to use a custom tool in to generate procedural narrative through dialogue systems with game characters. It also discusses the challenges in balancing procedural generation with structured narrative elements, ensuring coherent storylines.

2.6 "SceneCraft: Automating Interactive Narrative Scene Generation in Digital Games with Large Language Model

A study conducted by V. Kumaran, J. Rowe, B. Mott, and J. Leste, at North Carolina State University, discusses the use of large language models (LLMs) in automating narrative scene generation in digital games. The study details the design of SceneCraft framework, which interprets natural language instructions to generate game scenes aligned with authorial intent. SceneCraft highlights the framework's ability to generate branching conversation paths and dynamic NPC interactions, enhancing player engagement.

3 METHODOLOGY

3.1 System Design Framework

Using a structured system design and a clear data content generation format helps GPT models create game narratives that are relevant and ready-to-use in narrative driven games.

- **Chunking:** Breaking the information down into chunks since they provide designers macro control over the outcome, the information is easier to parse, and finally provides a more accurate generation of content with GPTs to avoid hallucinations - AI hallucinations happen when an AI model, inaccurately generates information as fact. This occurs because these tools are programmed to predict the most likely word sequences for a given query, without the capability to logically evaluate or identify factual errors. Essentially, AI hallucinations arise when the AI strays in its attempt to respond accurately due to a larger number of word connections it finds in a bigger query.
- **Narrative generation example:** 3 Act structure. Asking the LLM or GPT model to generate content on a per act basis, with each act having dependencies on the act previously generated.

3.2 Guided information generation format

Providing the GPT model with strict guidelines and file format instructions to generate information.

- **JSON:** JSON (JavaScript Object Notation) is a lightweight data-interchange file format that is easy for humans to read and write and for machines to parse and generate, using text to represent simple data structures and associative arrays.
- For this study, the GPT models were asked to generate data in a structed Json format. The GPT model was provided with Json files with definitions and variables, that it uses to fill in the details which are then plugged into the artifact directly as text objects.

3.3 Well defined datasets

Providing detailed and well-defined data sets to the GPT model allows it to generate unique and engaging stories dynamically.

- **In the current artifact, the GPT model is provided with a set of Json files-** Narrative setting, NPC personalities, Overarching Narratives, Plot Generation Model, and Card definitions. Each of these files consists of a requirement, definitions of those requirements, and a Json structure of the data that is to be generated and returned.
- The GPT model generates and returns data in a sequential order of the data sets provided, each with dependencies on the previously generated information to maintain coherence and logical flow of the narrative.

3.4 Guardrails for content generation and story focus

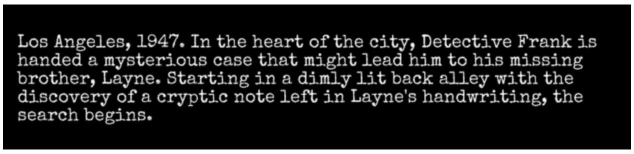
Guardrails – Set of instructions provided to the GPT model in a prompt structure to maintain narrative coherence and avoid inappropriate information to be presented when generating data.

Along with each data set, the GPT is given a general definition of the guardrails it is supposed to follow to keep the story on track and to ensure that the data generated dynamically during the playthrough is appropriate and relevant.

These guardrails are the key elements or rule sets that the GPT adheres to. The guardrails also allow designers to maintain consistency in data generation pertaining to the game's requirements.

Few examples:

- Overarching narrative must have an Introduction, 3 cases- interdependent on each other, NPCs associated with each case, and a resolution.



[8]

- NPCs traits and personalities must be generated based on the Narrative Setting of the case.
- NPCs must maintain their personalities when conversing with the player.
- All the data generation should be in the realm of the narrative and real-world information should not be generated at any given point, even if the player requests it. In case, it is requested, the NPC must respond according to its personality and the narrative

setting of the game to drive the player back into the story of the game.

- No inappropriate data should be generated even if requested.
- Data generated must always be in the requested JSON structure format, and no text must be explained. This guardrail ensures formatted content to be generated which can then be parsed as variables in game code.
    - For example, to generate the narrative setting of the story a contextual prompt is provided along with a strict Json structure that is required as a response.
    - Contextual Prompt: Generate a Narrative Story Setting film noir-style detective role-playing game setting, specifying the type of location, a specific location, the general time period, and a precise date.
    - Response Instruction: Return nothing but JSON like this and don't reply with anything else, don't explain the text!
    - Desired JSON Structure example:

```
{
    "location_city":"...",
    "specific_location_in_relation_to_story":"...",
    "time_period":"...",
    "precise_date":"...",
    "narrative_story_setting_description":"..."
}
```

[6]

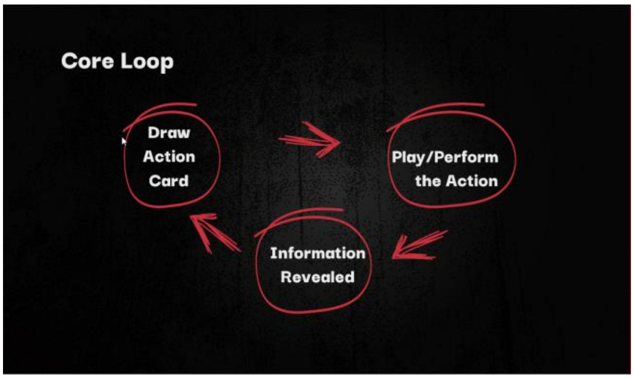
4 ARTIFACT

The artifact is a digital crime mystery card game where the objective is to solve crimes to unravel the core narrative or the story. The overarching narrative, sub-plots, characters, and their individual personalities are generated by AI. Interaction with NPCs generates the core story and NPC dialogs on the fly, creating a unique experience for each player.

[8]

4.1 Core Gameplay Loop

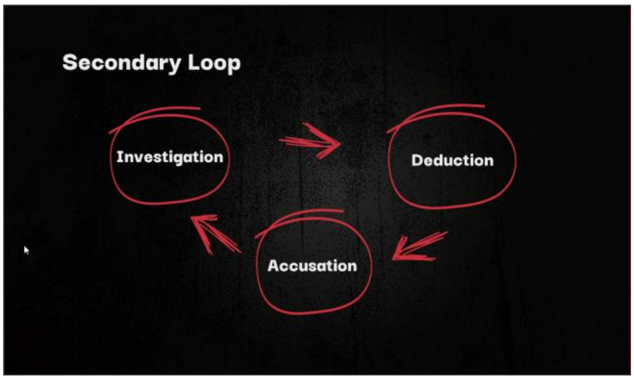
- The player draws action cards.
- The player performs the actions.
- New Information is revealed.



[7]

4.2 Secondary Gameplay Loop

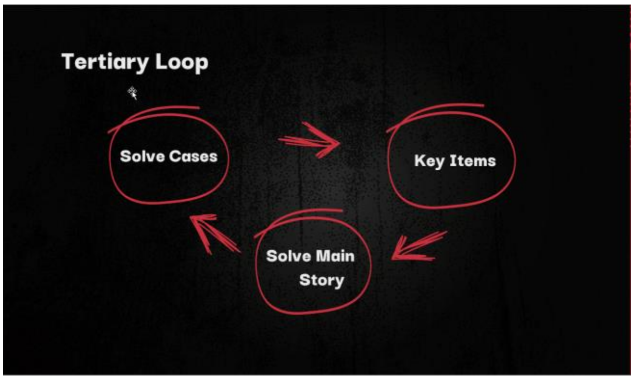
- Based on the information revealed, the player conducts the investigation.
- The player then makes their deduction.
- And finally, the player makes an accusation to find the culprit of the crime.



[7]

4.3 Tertiary Gameplay Loop

- The player solves multiple cases.
- Each case rewards the player a key item.
- This key item is associated with the main story of the game.
- Gaining access to multiple key items help the player unravel the main story of the game.



[7]

4.4 Initialization:

- During the initialization of the game, the GPT integrated into the game, generates data in JSON formats which is then used to populate text objects that are presented to the player.
- An overarching narrative is generated based on predefined parameters, for example, the narrative in the artifact is about a detective who is searching for their lost brother. Here, we are defining the premise of the narrative. The GPT model takes this information and generates content adhering to the 3-act structure and develops 3 core narrative beats for the story. Each Narrative beat act as a separate case, that the player solves, which leads to the resolution of the player finding their brother.
- Based on each narrative beat, a card deck consisting of various card types are generated. Each card consists of definitions and a JSON structure. These details are reference points for the GPT model to generate data. The GPT model looks at these definitions and returns a response in a JSON structure.
- Case Cards: These cards represent the cases to be solved. Each case card has a brief description of the crime, a list of suspects, and a solution key(the solution is hidden from the player but created during data generation).

[8]

- Suspect cards: These cards represent the individuals involved in the case. Each suspect card has a set of alibis, motives, and connections to evidence cards.

[8]

- Action Cards:

[8]

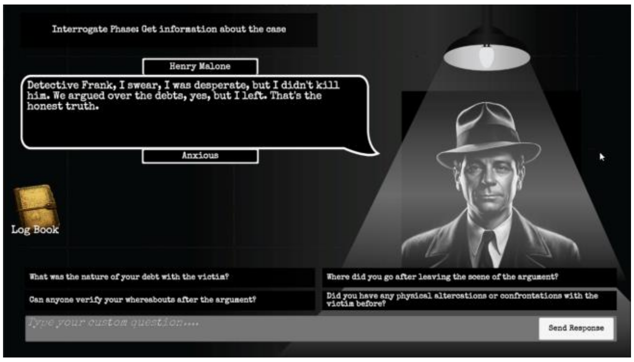
  - Interrogate: This card allows the player to question a suspect to gather more information about the case.
  - Search Crime Scene: This card enables the player to revisit the crime scene and look for any overlooked clues or evidence.
  - Consult Forensics: Sometimes, a closer look is needed. This card allows the player to get a forensic analysis on a piece of evidence, revealing details that aren't visible to the naked eye.

4.5 Gameplay Phases

The player is presented with a case card, 3 suspect cards, and a deck of action cards, and evidence cards.

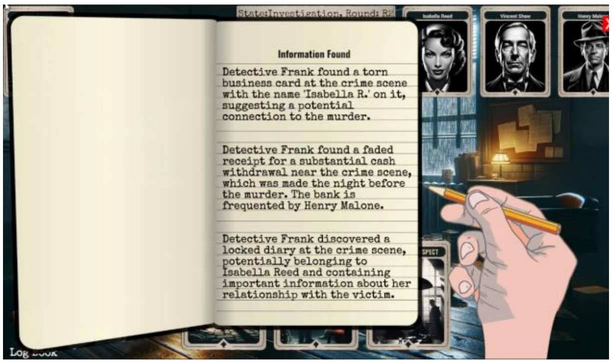
- Investigation Phase: The player draws 3 action cards. By playing each action card, the player executes its action, which reveals new information about the case. New information might be released through interrogation, search crime scene, and consult forensics.

[8]



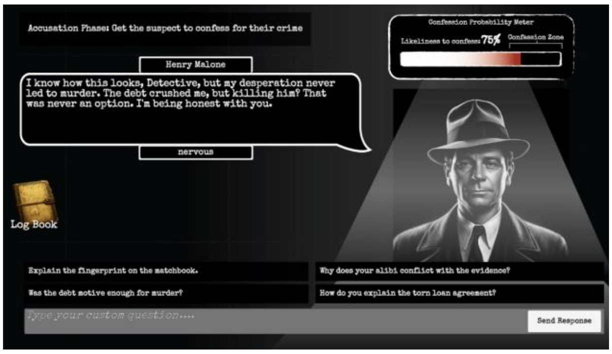
[8]

- Deduction Phase: The player draws and plays another action card. This is the final draw. The player can now use the information revealed in all the stages, to deduce and piece together the mystery.



[8]

- Accusation Phase: Once the deduction is made, the player chooses one of the 3 suspects to accuse of committing the crime in the case at hand. If a correct accusation is made the player must draw a confession out of the culprit of the case by going into a conversation with the accused suspect. If an incorrect accusation is made, the player must solve the case again.



[8]

- The player must complete 3 cases successfully to unravel the core story of the game.

5 TECHNICAL IMPLEMENTATION

5.1 Development of the Artifact in Unity3D

5.1.1 Game Design and UI:

- A technical document envisioning the game's mechanics, system architecture, and UI, was developed. The document outlined the integration of the GPT model – Details of data handled offline by the game code versus the data generated by the GPT model.
- The Technical document was a living document, i.e. was consistently updated and modified based on the nature of the research project.
    - The document details information/prompts in the JSON schema.
    - Game Card Mechanics
    - System Architecture – Detailing the actions handled by the game code, versus the data generated by the LLM.
    - Gameplay Phases
    - NPC Personality Generation – based primarily on the Big 5 Personality traits.
    - NPC Dialog Generation formats.
    - Mock-ups of the UI and gameplay screens of the game.

5.1.2 Game's System Architecture:

- The game architecture consists of a harmonious system of data handling and data generation between the game code, and the requests sent to the LLM. There are "handshakes" at multiple stages of the gameplay (in real time) between the game code and the LLM. The game code acts as a data base for retaining and managing the data required to display narrative content to the player.
- Most of the player actions request data generation. The game code requests apt data in the format of JSON schemas. Once the data is returned by the LLM, the game code then populates this information into variables and presents them to the player.
- The data requested by the game from the LLM is implemented through a server API that connects the game to the LLM.
- An example of this "handshake" can be seen when the player play's an action card such as Search Crime Scene – The LLM would return a JSON structure containing variables like – Key Item Found: "Knife", Event Summary: "You ruffle through the pages of the book and find a knife shaped cutout. Sitting in the cut out is a bloody knife." These variables are then parsed by the game code and plugged into the game UI to present to the player.

5.1.3 JSON Schema:

- The JSON structures and the guardrails were established and finetuned by creating custom ChatGPT models for testing data generation outputs.
- Since GPT models can interpret and write code, the GPT model was asked to generate content in JSON structures.
- The GPT models were also given prompt in a natural language format – not adhering to the JSON structure. The results in this case were appropriate to the game content, but inconsistent in the formatting. This created issues with parsing data as game code.
- Hence, the choice of sticking to a JSON schema was established.

5.2 Issues faced during the development.

5.2.1 LLM Memory Constraints:

- The researcher encountered difficulties in maintaining a consistent and coherent narrative due to the language model's limited ability to remember past interactions, leading to potential narrative discontinuities.
- Challenges in integrating complex story arcs that span multiple interactions, as the model lost track of previously established plot points or character developments.
- Mitigation: A custom memory handler server was built and plugged into the game as an API, that allowed the game code to request data from a server that retained all the information generated by the LLM. This allowed the prompts to be customized with apt information plugged in when requesting data.

5.2.2 Token Count Issues:

- The researcher faced limitations with the maximum token count that could be processed in a single prompt, restricting the length and depth of narrative content that could be generated at once.
- This constraint also affected the ability to provide detailed backstories or complex narrative explanations within a single interaction, necessitating workarounds to split content across multiple prompts.
- Mitigation: The memory handler consisted of another LLM powered summarizer. This summarizer would condense the information into a short summary and plug that into the prompt. This not only allowed the prompts to be small when requesting data, but also provided contextual data for relevant information to be provided staying under the token count limit.

5.2.3 Irrelevant Data Generation:

- Instances where the AI produced content that, while grammatically correct, was irrelevant or out of context for the intended narrative, requiring manual review and correction.
- Difficulty in ensuring that generated narratives remained within the thematic boundaries of the game, occasionally leading to anachronisms or tone inconsistencies. Mitigation: Flash Hacking scenarios using Custom ChatGPT models allowed the researchers to fine tune prompts and guardrails. During these flash hacks, bizarre scenarios were given to the GPT model to force it to generate irrelevant data. Based on its responses, the guardrails and prompts were finetuned, giving it strict instructions which allowed it to stay on track to generate appropriate content.

5.2.4 Data Generation Formatting:

- Challenges in getting the AI to consistently output data in a structured format (like JSON) that could be directly utilized by game systems without further processing or correction.
- Needed to implement additional layers of prompt engineering, parsing, and validation to ensure that the AI-generated content adhered to the expected data structures and formats for game integration.
- Mitigation: Flash hacking scenarios using custom GPT models also allowed us to create prompts that requested data in exact JSON structures as required by the game code. This was key as the link between the game and the LLM was established using these JSON schemas.

5.2.5 NPC Personality Adherence:

- Ensuring that AI-generated dialogues for NPCs remained true to their predefined personalities and backgrounds was challenging, with occasional deviations that disrupted player immersion.
- The AI sometimes generated responses that, while plausible, did not align with the character's established personality traits or the narrative tone, necessitating manual adjustments or the development of more sophisticated filtering mechanisms.
- Mitigation: Establishing guardrails was pivotal to maintaining NPC personalities. With each dialog requested to be generated, contextual information along with the NPC's pre-established personalities were plugged into the prompts. This allowed the GPT model to generate information exactly as desired.

6 CONCLUSION AND LEARNINGS

This thesis provides a structured and unique framework for using LLMs to generate procedural narratives and NPC personalities in an interactive turn-based role-playing game (RPG). The system is built around gameplay loops, which allow designers to harness the creative power of LLMs by giving them a rigid structure to generate dynamic, coherent storylines in real time. By incorporating the Big Five personality qualities into NPCs, this thesis creates a sophisticated character interaction model that is supplemented by a bespoke memory system for NPC behavior, guaranteeing that character replies are context-aware and do not reinforce stereotypes.

The thesis also brought about a new approach to the game production cycle which comprises of a research and development phase that explains the operations of GPT models and their role in generating dynamic content, as well as an improved debugging procedure. The system's architecture incorporates ethical considerations, with constant monitoring methods in place to tackle model bias based on training data to prevent harmful ideologies being promoted, while maintaining diversity and inclusiveness in NPC generation.

The research consisted of flash hacks and ablation tests conducted using the demo game, "Dark Shadows," which allowed the researchers to improve the gameplay experience by implementing guardrails that allowed AI generated content that adheres to the core narrative and gameplay guidelines. The thesis thus opens new avenues for future researchers to explore advancements in interactive narratives and game design using custom trained GPT models.

7 REFERENCES

[1] R. C. Rivera and C. M. Chris Martens, "Procedural narrative generation," GDC Vault: Procedural Narrative Generation, https://www.gdcvault.com/play/1024143/Procedural-Narrative (accessed Jan. 30, 2024).

[2] J. Frankenfield, "Artificial Intelligence (AI): What it is and how it is used," Investopedia, https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp (accessed Jan. 30, 2024).

[3] "What is GPT," Amazon: What is GPT?, https://aws.amazon.com/what-is/gpt/ (accessed Jan. 30, 2024).

[4] L. Ripoll Galan, "Scriptale: Generation of procedural narrative," Pàgina inicial de UPCommons, https://upcommons.upc.edu/handle/2117/128119?show=full (accessed Jan. 30, 2024).

[5] V. Kumaran, J. Rowe, B. Mott, and J. Lester, "Scenecraft: Automating interactive narrative scene generation in digital games with large language models," Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, https://ojs.aaai.org/index.php/AIIDE/article/view/27504 (accessed Jan. 30, 2024).

[6] Screenshot taken from a json structured created by Tanishq Chawla in Sublime text.

[7] Screenshot taken from a power point presentation created by Tanishq Chawla in MS PowerPoint.

[8] Screenshot taken from the Dark Shadows artifact developed by Tanishq Chawla and Zixin Zhuang in Unity3D