Designing Meaningful AI-Generated Dialogue: The Behaviour-Driven Conditional Prompting Framework for Serious Games

Kamiel de Visser[1], Anders Bouwer[2] and Ivo Swartjes[3]
[1]Grafisch Lyceum Utrecht, The Netherlands
[2]Amsterdam University of Applied Sciences, The Netherlands
[3]&ranj, Rotterdam, The Netherlands

kamieldevisser@gmail.com
a.j.bouwer@hva.nl
ivo@ranj.com

Abstract: Many serious games use dialogue between players and non-player characters (NPCs) to enhance learning. However, designing appropriate dialogue is often time-consuming for game developers. Recent advancements in artificial intelligence, particularly Large Language Models (LLMs), have made open-ended dialogue with virtual characters feasible, though managing it in educational contexts remains a significant challenge. This study explores how game designers can guide open-ended dialogue powered by LLMs to create meaningful educational conversations. Expert interviews and a review of existing approaches to implementing open-ended conversation in games led to the formulation of design requirements for a new framework. Based on these insights, the Behaviour-Driven Conversational Prompting (BDCP) framework was developed. This framework offers practical guidance for designers to create scenarios where behavioural learning objectives are achieved through structured dialogue. It combines lock-and-key narrative design, dynamic character prompts that dictate LLM-generated responses, and behaviour analysis prompts that assess player interactions. To validate the framework, a functional prototype called 'Detective Duck' was created. This detective-style 'whodunit' game has players solve crimes through open-ended conversation with AI-driven characters. Players encounter challenges such as persuading hesitant witnesses, or verifying alibis. These challenges can only be solved by demonstrating reasoning and conversational strategies relevant for detectives such as lateral thinking and persuasion. Upon demonstrating these behaviours, character prompts can be dynamically adjusted, ensuring that - only then - players receive key clues needed to advance the narrative. The framework and prototype were evaluated against the established design requirements and through expert interviews. Results were largely positive, indicating that the BDCP framework supports meaningful, open-ended dialogue aligned with educational objectives. However, some inconsistencies in dialogue coherence and adherence to designer intent were noted. Future work will focus on refining narrative consistency and enhancing adherence to designer intent by fine-tuning the language model, integrating AI-driven player feedback, and incorporating other game mechanics into the framework. These improvements would further strengthen the BDCP framework as a tool for designing serious games centered around open-ended conversation.

Keywords: Dialogue generation, Serious games, Game design framework, Large language models, Behaviour-driven conditional prompting framework

1. Introduction

Dialogue has played an important role in many digital games, from the early days of text adventures, where the system provided the players with verbal descriptions of the game world while graphical capabilities were still limited, to games like L.A. Noire (2011), which incorporates spoken dialogue by actors who show facial expressions, and may be lying or telling the truth (Domsch, 2017). In interactive drama games such as Façade (Mateas & Stern, 2005), the dialogue plays an important role in advancing the game's narrative. In serious games, dialogue also often plays an important role, especially when this is used as a game mechanic in relation to learning goals to be achieved in the game. Because serious games also typically allow players to learn by doing (Anastasiadis et al, 2018), the dialogue also needs to make sense in relation to actions that players and NPCs can perform in the game. In many cases, the dialogue is scripted to ensure alignment of the dialogue with the intended narrative and learning goals (van der Meij et al, 2020).

With Artificial Intelligence (AI), and in particular, Large Language Models (LLMs), showing major advancements in recent years (Liu et al, 2023), game designers now can also employ open-ended conversation. LLMs, like OpenAI's GPT models, are capable of processing written text and generating replies in a conversational manner, showing understanding of the subject matter. Their capabilities are still improving, showing major leaps between generations (OpenAI, 2024). Whereas previously game designers had to prepare every line of spoken word in a game, designers now have the option to implement a LLM into their game that generates responses instead. Instead of having players choose from predetermined dialogue options (as in the traditional 'dialogue-tree' style game), players can now openly converse with a LLM, allowing them to come up with their own line of questioning. This open-endedness of dialogue can offer several important advantages for serious game designers: First, it allows players to act as an agent with free will. This mitigates unintended scenarios where players resort to calculated guessing from the available options, instead of actively engaging in problem-solving. Second, using generative dialogue can reduce the workload of a designer: it can be a time-consuming process to create branching dialogue, as each branch in the path requires consideration of the outcomes, character reactions, and narrative coherence. Additionally, in a worst-case scenario, each branch in the path can exponentially increase the number of dialogue lines that need to be written.

While the open-ended nature of generated dialogue is powerful, its strength also comes with weakness, particularly when it comes to maintaining control. How can a conversation be kept open-ended while also steering it in an educational direction? On one hand, the designer wants to push the player into completing an educational objective, but on the other hand, they want the player to retain agency of their actions. Therefore, this study aims to answer the following research question:

> *"How can serious game designers maintain balance between control over conversational*
>
> *topics and the open-ended nature of AI-generated dialogue?"*

The structure of the remainder of this paper is as follows: Section 2 discusses related work on dialogue generation in games; Section 3 introduces the Behaviour-Driven Conditional Prompting framework; Section 4 describes how this framework is implemented in a case study to create meaningful dialogue in a prototype game; Section 5 discusses evaluation of the framework using expert reviews; finally, Section 6 concludes this paper.

2. Related Work

Related work on dialogue generation can be found using various approaches, such as knowledge graphs, intentional lines, and generating dialogue and code together at game time.

2.1 Dialogue Generation Based on Knowledge Graphs

Ashby et al (2023) proposed a novel framework for quest and dialogue generation in role playing games. Using a knowledge graph-based approach, quests are generated that fit within the context of a game. Accompanying dialogue for those quests is then generated using a large language model, and delivered in-game through NPCs. The knowledge graph exists of nodes that represent in-game objects and entities (i.e., NPC: Trevor, or Resource: Gold), which are then linked through certain relations, such as 'wants' or 'made from'. From this knowledge base, quest descriptions are generated that are meant to align with the intent of the player's input using a cosine similarity approach. The quest description forms part of the prompt for generating a context-sensitive dialogue using a fine-tuned large language model. This approach can generate new content in game-time, but is restricted to a design-time based knowledge graph. The developer can maintain control over possible generated quests by adjusting the knowledge graph. However, this approach is labour-intensive and is limited by the quality of contents in the provided knowledge base.

2.2 Dialogues Based on Intentional Lines

Kerr and Szafron (2009) offer an approach that provides insights that can help a developer come up with new dialogue. First, they consider a dialogue to essentially be a collection of "intentional lines", i.e., "an abstract line with a specific intent", for example a *greeting line*, or a *farewell line* (Kerr & Szafron, 2009), which can be worded differently while serving the same intention. For each intentional line, a collection of 'actual dialogue' lines can be generated at design time that serve the same intent but have different social characteristics such as sophistication and disposition of the speaker. Intentional lines can be replaced by 'actual dialogue' by filtering on these characteristics at game time. Their approach offers a systematic way to design dialogue contents and organize the timeline within the dialogue, but does not generate actual dialogue lines at game-time.

2.3 Dialogue and Code Generation at Game-Time

Volum et al (2022) put the focus more on game time generation. Through the use of OpenAI Codex (Zaremba et al, 2021) an attempt is made to create NPCs that engage in open-ended conversations with players in the context of Minecraft (Minecraft, 2024). A player can talk to nearby NPCs through the means of the Minecraft chat interface. The player input is processed by the language model, which then generates a code-like response that is executed in the game. In the prompt to the Large Language Model, they include both the natural language commands and examples of code that needs to be generated to enable the NPC to act on the game state, using a technique called autoregressive prompting (Volum et al, 2022), also known as Retrieval-Augmented Generation (RAG) (Lewis et al, 2020). This approach still had some problems however, such as generated code calling non-existing functions, NPCs repeating themselves, especially in lengthier conversations, and NPCs presenting unreliable information due to recent prompt bias.

2.4 Analysis of Dialogue Generation in Existing Detective Games

In addition to the academic literature above, two existing detective games with open-ended conversation as a mechanic have been analyzed: Vaudeville ("Vaudeville on Steam", 2024) and WhoDunit (Kniberg, 2023).

Vaudeville is a detective game in which players attempt to solve a murder mystery by engaging in open-ended conversation with AI-driven NPCs ("Vaudeville on Steam", 2024). The narrative of Vaudeville can be shaped by the game designers purely through character design. After creating a new character, and determining their personality and role in the story, the script that contains the plot of the mystery in Vaudeville is updated, and the character, personality and background information are fed to the Inworld character engine. If a game-specific event is required, example dialogue that should trigger that event is fed to the character engine.

WhoDunit is an experimental game created by Kniberg (2023), which uses multiple AIs to generate different types of content. For example, a mystery generator AI generates a mystery, which is fed to a GameMaster AI, which in turn generates the prompts that provide a Character AI with instructions on how to generate the dialogue. This separation of concerns allows more specific prompt engineering resulting in better fine-tuning of results. Whodunit is not an educational game, however, and focused rather on providing entertainment value.

3. The Behaviour-Driven Conditional Prompting Framework

Developing games with AI-generated dialogue poses challenges in steering the dialogue in a certain direction that fits the narrative, especially when gameplay depends on *impactful dialogue* that helps achieve goals in the game. To support game designers in creating meaningful conversations with NPCs using generative AI, a new framework has been developed, called the Behaviour-Driven Conditional Prompting (BDCP) framework, based on requirements derived from interviews.

3.1 Interviews

To derive requirements for the framework, interviews were carried out with five experienced designers working in the serious games industry (de Visser, 2024). Below is a summary of the main issues as discussed in these interviews:

No 'one size fits all' approach

When asked about their general approach to serious game design, it became evident that there is no approach that fits all games. Two serious games may have the same objective, but their clients may have different needs for the tone of voice or entertainment factor.

Scenario-Based Learning

When asked about design tools used to convey serious game objectives, the most popular design element was scenario-based learning using a dialogue tree. This approach allows designers to control what players may encounter, analyze play traces, and adjust scenarios to clients' needs.

Dialogue trees limit player agency

When asked about challenges in creating the above-named scenarios, it was clear that this approach comes with difficulties. For example, for each branch added to the plot, more dialogue needs to be written to cover it. More important, however, was the argument that dialogue trees limit player agency.

Incorporating open-ended conversation to regain player agency

As discussed, LLMs could improve player agency by granting players full control over how they converse, without limiting them to predetermined lines. A good starting place would be to write a narrative structure that allows for scenario-based learning, then feed that narrative to the LLM. With such a structure, it seems easier to write rules and test interactions with the model than with a dynamic narrative.

3.2 Requirements

Using the information from the interviews, a list of functional requirements was compiled that needs to be supported by the framework:

- Open Conversation: the design should deliver a game mechanic that allows players to openly converse with characters in a game, where responses from these characters are generated by a LLM.
- Structured Narrative Scenario Design: the designer should be able to create structured narrative-based conversational challenges in which players can demonstrate their behavioural skills.
- Responsive dialogue: the designer can influence a conversation's direction based on player behaviour.
- Behaviour in dialogue drives events: the designer can make dialogue events affect the game world.
- High-level control: the designer has high-level control over the output of the LLM, incl. parameters such as tone of voice, character backstory, response length, and knowledge of the game world.
- Syntactical and social rules: the designer must be able to define syntactical and social rules that determine how the LLM responds to player inquiries.

Besides these functional requirements, there are also three qualitative requirements to consider:

- Ease of use: the design must be easy to conceptualize, and allows for rapid development of new content.
- Adaptability to other serious contexts: the design must be adaptable to a multitude of serious game contexts for it to be considered widely useful.
- Avenues for player feedback: the design must include avenues to supply the player with feedback on their behaviour as part of the educational aspect of serious games.

3.3 The Framework

This section presents the Behaviour-Driven Conditional Prompting (BDCP) framework, that aims to support game designers in creating game scenarios in which serious game goals are achieved through dialogue with NPCs driven by LLMs. The name reflects the two core concepts of the framework:

- Behaviour-Driven: The framework allows designing scenarios that test a player's behaviour. Behaviour can range from social characteristics, e.g., emotional response, politeness and verbal sophistication, to conversational techniques, e.g., applying sales strategies or investigative questioning.
- Conditional Prompting: The player behaviour triggers conditions that alter the dialogue-generating prompts supplied to the large language model. For example, if the player shows a trustworthy attitude toward an NPC, the prompts for that NPC can be altered to make them share more information.

With the BDCP framework, game designers develop conditional prompts that are used to generate dialogue through use of a language model. The language model takes on two roles in this framework:

- Character AI - In this role, the model is used to generate open-ended conversation between the player and characters in the story. The Character AI is prompted with details such as information about the characters, context, rules that dictate what they can or cannot talk about, and a conditional objective.
- GameMaster AI - The GameMaster AI is tasked with performing checks pertaining to the dialogue between the player and characters, e.g., "Has the player named <NPC> as a possible suspect?". The yes/no reply is parsed by the GameMaster and the result triggers the associated in-game events.

The BDCP framework splits the scenario design up into four phases: Goal, Lock and Key Scenario, Conditional Prompts, and Events. A schematic overview of the framework including the four phases is shown in Figure 1.

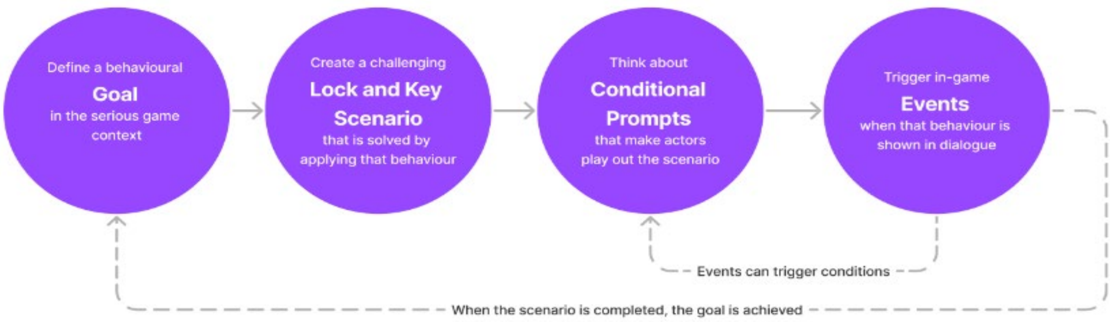

Figure 1: A schematic overview of the four phases of the BDCP framework

Details on how the framework supports the four phases in the context of a case study are given in Section 4.

4. Case Study: Generating Conversations with NPCs in a Detective-Style Game

In order to test the applicability of the BDCP framework, a detective-style game called 'Detective Duck' was developed as a prototype, implementing the framework's four phases, described in Section 4.1 to 4.4. Section 4.5 describes technical implementation choices and Section 4.6 presents a resulting example dialogue.

4.1 Phase one: Designing Behavioural Learning Goals

The serious game context for the 'Detective Duck' prototype concerns law enforcement investigations. As a hypothetical client, we imagined a law enforcement agency with a shortage of detectives ordering a dialogue-driven game that engages players to use their problem solving skills and encourages them to consider this line of work. Behavioural techniques relevant to professional investigators were selected, inspired by examples found in literature (van der Sleen, 2009):

- Persuading a hesitant person of interest - When a person is hesitant (for logical or emotional reasons) to provide information, persuasion is required.
- Investigating a timeline of events - For example, by asking the right questions, investigators can gather evidence that supports or dismantles an alibi.
- Handling emotional characters - Emotions can get high in a criminal investigation, and knowing how to handle such situations is important, e.g., by showing emotional understanding, and building trust.
- Connecting the dots - Using lateral thinking, corroborating different clues can result in new theories and leads to investigate.
- Persuade a confession through conclusive evidence - Being able to persuade a culprit to confess is important to close the case.

4.2 Phase Two: Designing Lock and key Scenarios Around the Behavioural Learning Goals

4.2.1 Locks and keys

The term lock-and-key mechanism refers to a mechanism that unlocks access to parts of a game (Adams & Dormans, 2012). Locks and keys may be physical, or more conceptual in nature. Scenarios can lock player progress behind required player actions, which may require mastery of certain behaviours.

Ashmore (2006) lists the functional properties that keys and locks typically have in games:

- *Keys have little impact on low-level gameplay*, but can strongly affect high-level gameplay.
- *Old keys are still useful*. A key becomes more important when it can or must be used multiple times.
- *Locks are often shown before the key appears*, to make players want to find out what lies behind.
- *Sometimes, locks are not always immediately visible*. In contrast with the previous point, sometimes a lock can be hidden behind another lock, allowing for places already visited to seem new.

4.2.2 Scene 1

For scene 1, the designer has selected the goal to persuade a hesitant person of interest, and created a lock-and-key scenario for this objective, shown in Table 1. The player's task is to investigate a crime: what happened to Bubbles the Bunny's muffins, which disappeared the day before the yearly bakery contest? The player has to convince Bubbles to consider characters' potential motives. Bubbles is initially hesitant to suspect anyone, but if the player convinces Bubbles to do so, he will reveal that Rustles the Raccoon came second last year, and thus may potentially have motive to sabotage Bubbles this year.

Table 1: Overview of the lock, key, and door in Scene 1 of the game

Lock	Key	Door
Bubbles won't mention Rustles the Raccoon as a possible suspect.	Instruct Bubbles to think who could have had motive to steal his muffins.	Rustles the Racoon becomes available as a character to interview.

As shown in Table 1, narrative progression in scene 1 is locked: The player cannot leave this area before he gets a lead. When the player demonstrates the desired behaviour, Bubbles' prompt is altered, and he shares the information required to open the door to the next scene. In Figure 2, the design process so far is schematized.

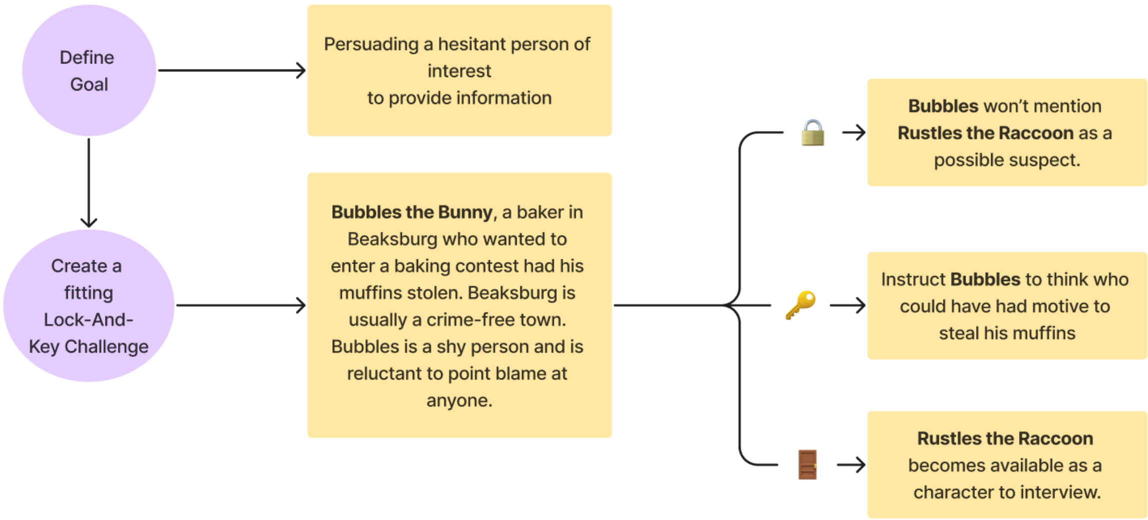

Figure 2: A schematized overview of the design process of scene 1 so far

Subsequent scenes in the game are more complicated, often dealing with multiple goals of investigating a timeline and handling emotional characters, which may also have their own side goals.

4.3 Phase Three: Designing Conditional Prompts for the Lock and key Scenarios

In phase three, conditional prompts are created for each scenario to generate dialogue through character AI. Depending on players' behaviour, these prompts are altered to reveal new information and provide character AI with new rule sets to follow. First, a generic prompt is designed, providing context and response rules to the AI (i.e. 'Respond like a character in a video game'). Then, character-specific information is slotted in (i.e. 'You are Bubbles, a shy baker'). Finally, appended to the prompt are Optionals: these are interchangeable based on player behaviour and the driving factor behind the BDCP framework. Figure 3 illustrates how Optionals reveal key information that progresses the narrative.

4.4 Phase Four: Coupling In-Game Events to the GameMaster AI

This section describes how the designer can design questions to be fed to the GameMaster AI, and use the answers to these questions to alter the previously described conditional prompts. The GameMaster is used to perform GMChecks. A GMCheck contains a question concerning the conversation, that is answerable by yes/no. In scene 1 of the prototype, there are two checks that need to be performed. The first check is related to the 'key' in that scene. If the LLM considers the conversation between the player and the character, and answers "Yes" to this check, then the prompt for Bubbles' character AI is altered to allow him to suspect Rustles the Raccoon. The next check is related to the 'door' in that scene. If the locked information was not provided to the player yet, and the conversation ended too soon, there is no way for the player to advance the narrative. Only when this door check succeeds, the next scene is unlocked. In Figure 3, a schematic overview is provided of the GameMaster's GMChecks in Scene 1 and their effects on the relevant conditional prompts.

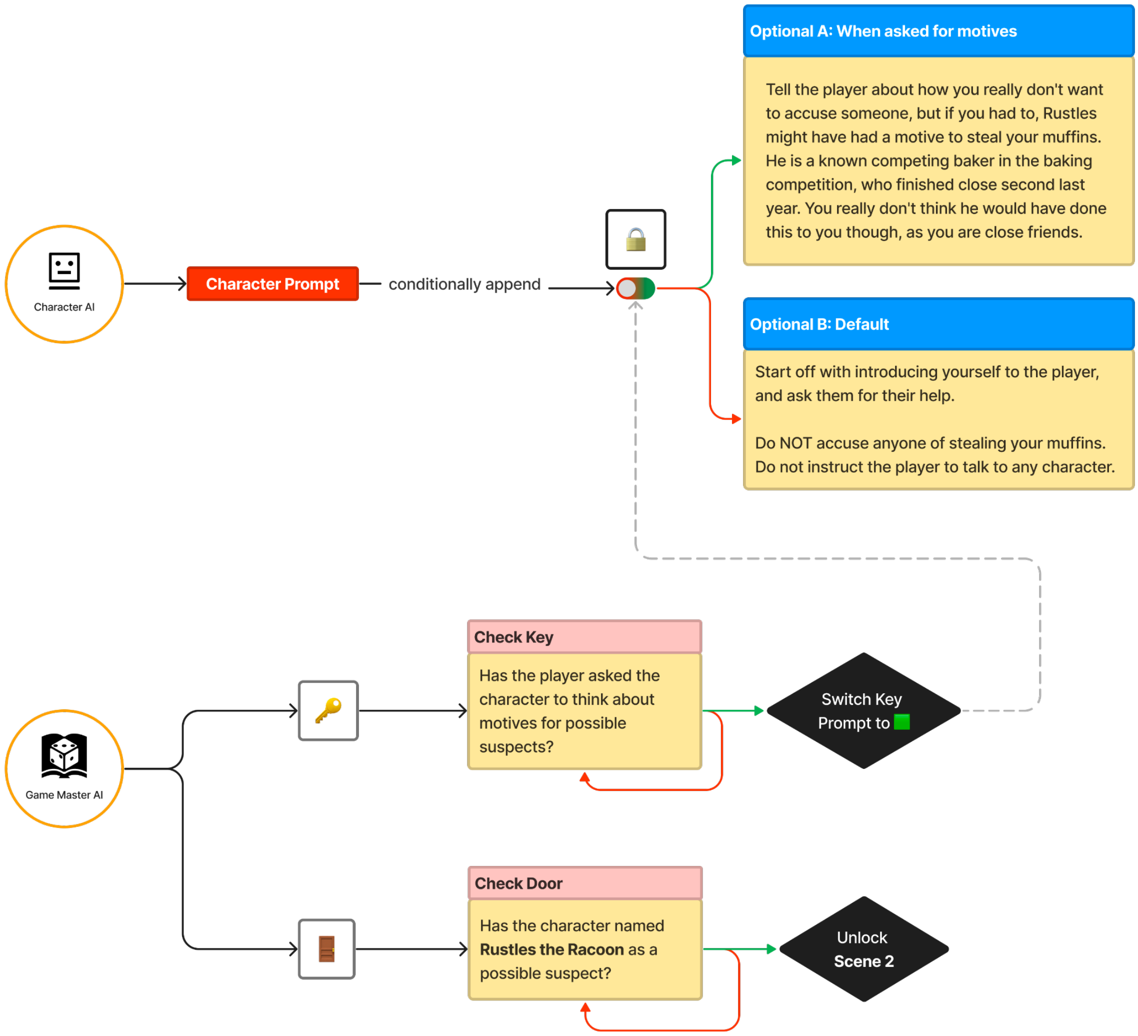

Figure 3: The GameMaster's GMChecks for Scene 1 and their influence on the conditional prompts. Optional B is appended by default, until a condition is met, and optional A is appended instead

4.5 Technical Implementation

The prototype was built in Unity3D (Unity, 2024), using GPT-3.5-turbo. GPT-4 was also considered but 3.5-turbo seemed to provide sufficient responses and its lower response time contributed to more fluent gameplay. A ChatGPT Wrapper library (RageAgainstThePixel, 2024) was used to interact with the ChatGPT API. A custom editor extension allows designers to preview, arrange, and alter prompts, and attach corresponding GMChecks. The GameMaster AI was instructed to reply with Yes/No answers, which allows for simple boolean parsing. Based on the checks, the ProgressionManager triggers UnityEvents (to update dialogue prompts, progress the story, unlock new objectives, etc.) before generating a new character reply, allowing the game to respond directly to player behaviour. More details on the implementation are described by de Visser (2024).

4.6 Example Dialogue

An example dialogue taking place in scene 1 is displayed in Figure 4, featuring automatically generated text for the NPC Bubbles, while the player enters the text for the main character, Detective Duck. The player encourages Bubbles to think about who could have had a motive for the crime, which by design results in an accusation towards Rustles the Raccoon.

Figure 4: Example dialogue from the game. Text for Bubbles is generated by the system described in Section 4.5, while the player enters the text for Detective Duck

5. Evaluation

To evaluate the usability and usefulness of the framework, another series of interviews was held with five game designers to gather feedback on the framework in relation to the qualitative requirements. An overview of all feedback points related to these requirements, as discussed in these interviews, is presented below:

- *Ease of Use:* The BDCP framework is conceptually simple to use, and provides designers with a way of quickly designing new game scenarios; Further improvements, such as expanding the Editor tooling, are needed to make it practical in real-world use, however.
- *Adaptability to other contexts:* The framework is deemed suitable for any serious game context regarding soft skills; The framework should incorporate other game mechanics to teach hard skills.
- *High-level control over conversation:* The framework offers high-level control over the response of characters through scenario design and textual prompting, using syntactical and social rules; Suggestions to enhance the level of control include allowing for multiple outcomes to one scenario.
- *Avenues for player feedback:* The framework already includes several opportunities to give feedback to the player; Suggestions for improvement include introducing an additional character for guidance, and a third role for the language model to provide feedback and an end-game debriefing.
- *Derailment of conversation:* Some designers experienced the conversation derailing from the intended narrative; This issue should be addressed for real-client use; In its current state it is still usable for rapid prototyping to impress clients.

More details on this evaluation, and an analysis of the framework to investigate to what extent the functional design requirements listed in Section 3.2 have been satisfied, are described by de Visser (2024).

6. Conclusions

The main aim of this study was to understand how serious game designers can maintain high-level control over open-ended conversation between a player and NPCs powered by large language models, so these conversations can be used to achieve serious game goals. To meet the design requirements for this challenge, the Behaviour-Driven Conditional Prompting (BDCP) framework was developed. It supports designers in creating scenarios that require players to demonstrate soft skills relevant to the serious game context through conversations with in-game characters driven by a large language model.

The BDCP framework provides designers with guidelines on how to create challenging ’lock-and-key’ scenarios, and shows how these scenarios can be realized using the language model in two different roles: the character AI, which generates dialogue between players and NPCs through prompts that contain conditional information, and the GameMaster AI, which is responsible for analyzing the dialogue. The GameMaster AI adjusts the conditional prompts dynamically based on observed player behaviour to steer the conversation in the direction intended by the game designer.

Using this framework, a prototype detective-style game was developed, called ‘Detective Duck’. The framework and the resulting prototype were evaluated by five professional game designers against the design requirements. The evaluation produced promising results, with mostly positive feedback and useful suggestions for potential improvements. The BDCP framework was considered intuitive to understand, enabling game designers to quickly create new scenarios. The framework was also considered to be easily adaptable to other serious game contexts. To be viable for real-world use cases, however, the technical implementation still needs further work.

uggestions for future work include developing UX/UI tools to support scenario creation, improving the implementation by enabling character AI to interact with the game through code, fine-tuning the language model with example conversations to improve narrative consistency, extending the framework with AI to provide feedback on player progress, incorporating additional game mechanics into the BDCP framework, and summative evaluation of the framework and resulting games.

Ethics declaration: No ethical clearance was necessary for the research reported in this paper.

AI declaration: The paper provides a dialogue featuring AI-generated text in Figure 4, created by the system described in Section 4. The text clearly indicates which parts are generated and how they were created, in order to demonstrate and analyze the use of AI in dialogue-driven game design, related to the topic of this paper, i.e., research into conversational systems for games using AI generation.

In addition, OpenAI’s tool ChatGPT was used as a writing aid for this paper, to detect writing errors and suggest possible repairs, and to help summarize sections of text. Such suggestions and summaries were subsequently reworded by human authors. AI was not used to directly author text in the paper, except for the AI-generated dialogue fragments mentioned above.

References

Adams, E., & Dormans, J. (2012). Game mechanics: Advanced game design. New Riders.

Anastasiadis, T., Lampropoulos, G., & Siakas, K. (2018). Digital game-based learning and serious games in education. International Journal of Advances in Scientific Research and Engineering, 4 (12), 139–144.

Ashby, T., Webb, B. K., Knapp, G., Searle, J., & Fulda, N. (2023). Personalized quest and dialogue generation in roleplaying games: A knowledge graph-and language model-based approach. Proceedings of CHI 2023, 1–20.

Ashmore, C. (2006). Key and lock puzzles in procedural gameplay. Master's thesis, Georgia Institute of Technology, April 2006.

de Visser, K. B. (2024). Maintaining control over open-ended conversation: An exploration into designing dynamic dialogue to produce meaningful conversations for serious games. Bachelor Thesis HBO-ICT, Amsterdam University of Applied Sciences.

Domsch, S. (2017). Dialogue in video games. In Dialogue across media (pp. 251-270). John Benjamins Publishing Company.

Kerr, C., & Szafron, D. (2009). Supporting dialogue generation for story-based games. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 5 (1), 154–160.

Kniberg, H. (2023, October). Whodunit - ai game development on steroids [online]. https://blog.crisp.se/2023/08/26/henrikkniberg/whodunit-ai-game-development-on-steroids.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., et al. (2023). Summary of chatgpt-related research and perspective towards the future of large language models. Meta-radiology, 1(2), 100017.

Mateas, M. & Stern, A. (2005). “Procedural authorship: A case-study of the interactive drama Façade”. In Digital Arts and Culture, 2005.

Minecraft (2024). Minecraft homepage [online]. https://www.minecraft.net/.

OpenAI (2024). GPT-4 [online]. https://openai.com/research/gpt-4.

RageAgainstThePixel (2024). OpenAI [online]. https://github.com/RageAgainstThePixel/com.openai.unity.

Unity (2024). Unity homepage [online]. https://unity.com/.

van der Meij, H., Veldkamp, S., & Leemkuil, H. (2020). Effects of scripting on dialogues, motivation and learning outcomes in serious games. British journal of educational technology, 51(2), 459-472.

van der Sleen, J. (2009). A structured model for investigative interviewing of suspects. Handbook of psychology of investigative interviewing: Current developments and future directions, 35–52.

Vaudeville on Steam. (2024). [online]. https://store.steampowered.com/app/2240920/Vaudeville/.

Volum, R., Rao, S., Xu, M., DesGarennes, G., Brockett, C., Van Durme, B., Deng, O., Malhotra, A. and Dolan, W. B., (2022). Craft an iron sword: Dynamically generating interactive game characters by prompting large language models tuned on code. In Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (pp. 25-43).

Zaremba, W., Brockman, G., & OpenAI. (2021, August) [online]. https://openai.com/blog/openai-codex.