Personalized the Behavior of Virtual Characters Based on Cognitive Architecture and Large Language Model

Yihan Ping

Georgia Institute of Technology
Atlanta, Georgia 30318
yping7@gatech.edu

Abstract

Immersive storytelling and games play a pivotal role in AR/VR development. This paper delves into the realm of turning concepts like those seen in Westworld into a reality. Specifically, it examines how cognitive architecture and the use of Large Language Models (LLMs) can enhance the behavior of virtual characters, thereby fostering a more interactive and personalized gaming experience. Knowledge-based AI and cognitive architecture can effectively model NPCs to simulate human cognitive processes, resulting in realistic decision-making and emotional modeling. On the other hand, LLMs excel in enhancing natural language interaction and narrative summarization. When combined, cognitive architecture and Large Language Models have the potential to generate dynamic narratives, adapt game difficulty levels in real-time, and ultimately provide players with a personalized gaming experience.

Introduction

Nowadays, the development of AR/VR has captured the attention of numerous investors and technology professionals, cementing its position as one of the future trends. Companies like Meta, Sony, Epic games etc are involved in the development of AR/VR games. Head-Mounted Display devices, spatial tracking technologies, and motion capture technologies enhance the game experience, making it more immersive and interactive. Immersive storytelling and games play a pivotal role in AR/VR development. Within virtual games, there are numerous virtual characters and AI-driven entities. Currently, the behavior of many Non-Playable Characters (NPCs) is scripted in advance. The prevailing methods used to define NPC behavior include Rule-Based Systems, Finite State Machines, and Behavior Trees. Consequently, they tend to exhibit identical responses to different players, which diminishes the engagement level of VR games. The objective of AR/VR games is to heighten immersion and realism. If a game continues to employ the previous game design model, it might be using resources excessively for AR/VR. To fully exploit the immersive potential of AR/VR games, enhancing the interactivity of NPCs is imperative. The research of personalizing NPC behavior using knowledge-based AI and LLM can bring billions of profits to the game industry.

Cognitive architecture is one of the areas within knowledge-based AI and is used to model human thought and reasoning, attempting to mimic cognitive processes in humans. Two prominent cognitive architectures are Soar and ACT-R. Cognitive architecture can be used to enhance immersive storytelling and games. NPCs can analyze the behavior and skill level of players and make appropriate next moves. From the player's perspective, NPC behavior could be more realistic in terms of actions and emotions, making the game more engaging.

Additionally, Large Language Models (LLMs) are currently a popular topic designed to understand and generate human language. In AR/VR games, LLMs can enable more natural and dynamic conversations between players and NPCs. NPCs can understand natural language from players and make the narrative and dialogue more realistic. LLMs can also be used to summarize the game experience and provide suggestions for the next level of the game.

This article will focus on how cognitive architecture and LLMs can enhance NPC behavior.

Related Work

Warpefelt et al. presented a proposal to evaluate and improve NPCs' social capabilities but failed to fully implement it due to technical challenges. McCollum et al. describe the use of the iGENTM architecture to control NPCs through cognitive and emotional models. The objective of interacting with NPCs is to learn culturally appropriate behaviors. This article also outlines the NPC behavioral taxonomy, emotional modeling method, and architecture design. However, this model struggled to facilitate facial animation for NPCs during complex interactions.

Ravenet et al. propose that current NPCs have limited abilities to exhibit emergent emotional and social behaviors, relying heavily on scripted rule-based interactions. Consequently, there is a need to integrate emotional models with AI tools. Rivera st al. develop enemy NPC design patterns for shooter games. However, the patterns focus just on enemies, not friendly and neutral NPCs. This model lack of variation.

Nathaniel et al. introduce the KNUDGE dataset, which contains 159 NPC dialogue trees with complex branching structures and ontological constraints, such as quest objectives and character biographies. They also employ a knowledge-constrained dialogue generation task and a dialogue writer model, a neural generation model. They use supervised learning on the KNUDGE dataset, as well as context learning with GPT-3, to generate NPC dialogues in games. However, their work has some limitations: KNUDGE dataset only collects NPCs from side quests, not major NPCs; their models do not consider human performance in game quality dialogues based on automatic and human evaluations; it is challenging to evaluate designer-written quests when the content is outside the original games; and using a complex graph-based method for branch structures with multiple nodes may be more suitable than relying solely on a language model for simplifying complex dialogue trees.

Methodology

In this research, Cognitive Architecture, with specific emphasis on the SOAR framework, will serve as the foundational framework for modeling cognitive processes within the context of Knowledge-Based AI. The methodology employs feature-value language to represent knowledge and leverages chunking as a learning technique.

The fundamental premise of Cognitive Architecture, as posited by Newell and Simon [5], is that behavior is an emergent property resulting from the interaction between architectural structures and the content within them. In our context, this implies that the design of the architectural framework plays a pivotal role in generating behavior, with the specific behavioral outcomes being influenced by the nature of the knowledge represented within.

The Cognitive Agent Function serves as a pivotal component in this methodology. It maps a sequence of perceptual inputs to corresponding actions, effectively bridging the gap between sensory input and behavioral response. SOAR, a well-established cognitive architecture for deliberation, will be utilized as the foundation for the cognitive agent function. SOAR was initially developed by John E. Laird, Allen Newell, and Paul S. Rosenbloom[6].

At its highest level, the SOAR architecture comprises two primary memory components: long-term memory and working memory. Within the long-term memory component, diverse forms of knowledge are stored. SOAR distinguishes three primary types of knowledge: Procedural, Semantic, and Episodic.

Episodic knowledge pertains to specific instances or occurrences of events. In the context of our game-related content, episodic knowledge corresponds to various in-game scenarios, such as NPC interactions during dining or shopping.

Semantic knowledge encompasses generalizations in the form of conceptual frameworks and models of the game world. In our context, it relates to the overarching models governing NPC behavior and the conceptualization of NPCs themselves.

Procedural knowledge encompasses the "how-to" aspects of performing tasks or actions. Within our game-related content, procedural knowledge pertains to the actions undertaken by NPCs and their dialogues with players. These procedural rules are stored in the long-term memory and are formally represented as production rules.

In summary, this research capitalizes on the SOAR cognitive architecture to model and simulate cognitive processes in the context of Knowledge-Based AI. The architecture's three knowledge types, Procedural, Semantic, and Episodic, are instrumental in defining and regulating NPC behavior within the gaming environment. Procedural knowledge, in particular, is codified in the form of production rules and is integral to the generation of dynamic in-game behaviors.

Experiments

In order to elucidate the fundamental principles of our proposed approaches, we shall employ the fighting game "NARUTO SHIPPUDEN: Ultimate Ninja STORM Trilogy" as an illustrative example. Within this game, I, as the player, have the prerogative to select a character, such as NARUTO, and to designate the NPC adversary, in this instance, Sasuke, with whom I will engage in combat. The subsequent discussion shall revolve around the utilization of the SOAR cognitive architecture within the context of this game, specifically focusing on the cognitive processes governing NPC Sasuke's decision-making and actions.

NPC Sasuke possesses a repository of internal knowledge encompassing predefined goals and objectives. Furthermore, Sasuke has the capacity to assimilate external information from the game environment. For instance, NPC Sasuke can perceive pertinent details, including but not limited to, the dimensions of the battle arena, the spatial positioning of both the player and the NPC, the ongoing score, the player's actions, the chakra status, and the vitality status.

Conceptually, NPC Sasuke's decision-making process can be depicted in Figure 1. Our methodology involves the utilization of feature-value representations to encapsulate the knowledge necessary for NPC Sasuke's decision-making. We ascribe to NPC Sasuke the goal of achieving a post-action state whereby the player's health points mirror those of NPC Sasuke, within a specified range. To explicate, if NPC Sasuke's health points are denoted as 'x', our goal for the player's health points ('y') will be defined that y is larger or equals to -10% x, and y is smaller or equals +10% x, with the likelihood of achieving each point within this range uniformly distributed.

To provide a comprehensive understanding of the cognitive agent's decision-making process, Figure 2 shall delineate the specific knowledge elements that inform NPC Sasuke's actions, with formal definition of the game state presented in the same figure. The procedural rules governing NPC Sasuke's decision-making shall be depicted in Figure 3.

Notably, SOAR employs a chunking mechanism to resolve impasses. For instance, when two conflicting rules become concurrently activated within a given context and no immediate resolution is apparent, SOAR initiates a historical event search to identify analogous situations. Should such a similar situation be identified, whether it yielded the desired outcome or not, the associated event shall be encapsulated

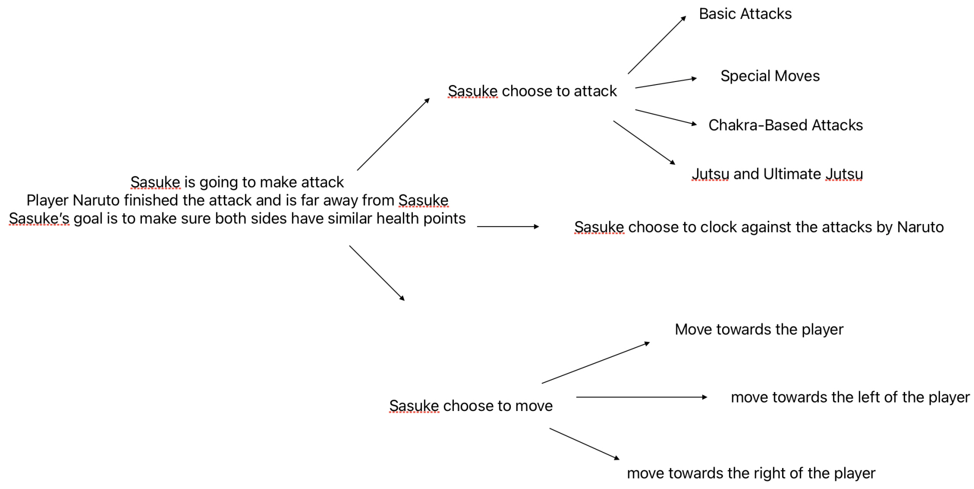

Figure 1

Features	Values
Health points of the player(max 10)	8
Health points of the adversary(max 10)	6
Distance between player and adversary(m)	3
Player's action	Basic Attacks
Player attack hit success rate	0.7

Figure 2
.

into the current rule, thus facilitating decision-making in the present context.

At the end of the game, NPC Sauke could talk with the player by using LLM, for example Chat-GPT API. We can put information like the performance of the player in this round and specific action the player do well or bad. And NPC could record if the player is doing better than expectation or worse, in this case, NPC could adjust the difficulty value in the next round.

Towards the conclusion of the gaming session, a novel and interactive element can be introduced wherein NPC Sasuke engages in a post-game dialogue with the player, facilitated through the utilization of Language Models (LMs) such as the Chat-GPT API. This dialogue serves as a mechanism for providing the player with personalized feedback, enriching the player's overall gaming experience.

In this context, NPC Sasuke serves as an informant, conveying specific insights regarding the player's performance during the preceding round. This feedback entails a comprehensive evaluation of the player's actions, encompassing commendation for proficient execution of certain maneuvers

If health points of the player is larger than adversary's,
Then suggest attack

If health points of the adversary is larger than player's and player's action belongs to attack,
Then suggest defense operator

If health points of the adversary is larger than player's and player's action belongs to defense,
Then suggest Jutsu operator

If suggest attack, and the distance between player and adversary is larger than or equals to 10,
Then suggest long-distance attack

If suggest long-distance attack, and player attack hit success rate >= 0.7,
Then suggest Ultimate Jutsuoperator

If suggest long-distance attack, and player attack hit success rate < 0.7,
Then suggest Jutsu operator

If suggest attack, and the distance between player and adversary is smaller than 10,
Then suggest short-distance attack, Basic Attack operator

If suggest short-distance attack, and player attack hit success rate >= 0.7,
Then suggest Chakra-Based Attacks operator

If suggest short-distance attack, and player attack hit success rate < 0.7,
Then suggest Basic Attack operator

If only one operator has been selected,
Then send operator to the motor system

Figure 3
.

and constructive critiques highlighting areas for potential improvement.

Furthermore, NPC Sasuke undertakes the pivotal role of assessing the player's performance through predefined performance expectations. By meticulously recording whether the player surpassed or fell short of these predetermined benchmarks, NPC Sasuke is equipped to dynamically calibrate the difficulty level for subsequent rounds. This adaptive approach ensures that the gameplay remains optimally challenging, fostering a delicate equilibrium between player engagement and attainability.

The integration of the Chat-GPT API into this interactive framework enables a seamless and natural mode of communication between NPC Sasuke and the player, contributing to a more immersive and rewarding gaming milieu. Through the conveyance of pertinent performance metrics and the subsequent fine-tuning of gameplay difficulty, this multi-faceted engagement elevates the overall gaming experience.

Environmental Setup

References

[1] Rivera, Gabriel, Kenneth Hullett, and Jim Whitehead. 2012. "Enemy NPC Design Patterns in Shooter Games." In Proceedings of the DiGRA FDG Joint Conference.

[2] Santarelli, Thomas, Christopher McCollum, Charles Barba, and John Deaton. 2004. "Applying a Cognitive Architecture to Control of Virtual Non-Player Characters." In Proceedings of the 2004 Winter Simulation Conference, edited by R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, 883–890. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

[3] Warpefelt, Henrik and Harko Verhagen. 2016. "A Model of Non-Player Character Believability." In Proceedings of the Social Believability in Games Workshop.

[4] Ravenet, Brian, Florian Pecune, Mathieu Chollet, and Catherine Pelachaud. "Emotion and attitude modeling for non-player characters." Emotion Modeling (2016): 44-56.

[5] Anderson, J.R. (2010). Cognitive Psychology and Its Implications. New York, NY: Worth Publishers.

[6] John E. Laird, Allen Newell, Paul S. Rosenbloom. "SOAR: An architecture for general intelligence",Artificial Intelligence,Volume 33, Issue 1,1987,ISSN 0004-3702