Adaptive AI for Games Using NARS

Introduction

My name is Nahom Walelinge, and this report summarizes my project, Adaptive AI for Games Using NARS (Non-Axiomatic Reasoning System), conducted under the guidance of Dr. Pei Wang at Temple University. The goal of this project was to explore how artificial intelligence, specifically a general-purpose reasoning system like NARS, can be applied to video game environments to create dynamic, adaptive in-game behavior.

Background

Traditional game AIs often rely on scripted behaviors or finite state machines that lack adaptability. These methods typically perform well within the narrow constraints of a pre-defined environment but fail when unexpected situations arise. NARS offers a promising alternative due to its ability to reason with insufficient knowledge and resources, making it more aligned with real-time and evolving gameplay conditions.

What makes NARS especially interesting to me is its general-purpose intelligence design. It is not trained on large datasets like neural networks but instead simulates human-like reasoning using a belief-desire-intention framework. This fits particularly well in environments like games, where players constantly create new, unforeseen situations.

Project Objectives

The main objectives of this project were:
- To understand the core architecture and reasoning model of NARS.
- To integrate NARS into a simplified game environment.
- To test how NARS can handle dynamic decision-making and adaptation in response to player behavior.
- To reflect on the difference in behavior between rule-based and reasoning-based AI systems in games.

Methodology

I started by studying the fundamentals of the NARS framework using the documentation and examples provided by Dr. Wang. After setting up the development environment, I created a simple 2D grid-based game prototype using Python, where an AI-controlled agent must collect resources and avoid hazards.

I then integrated NARS as the decision-making engine, replacing any hardcoded logic. I encoded game events and world states as Narsese input, and the AI agent would use NARS to generate goals, learn from new input, and revise its strategies over time.

How NARS Works

NARS operates based on a few core principles:
- Non-axiomatic logic: It assumes knowledge is always incomplete, and conclusions are uncertain.
- Narsese language: Input is given in a formalized syntax that resembles logic but includes uncertainty.
- Dynamic priority management: Goals and beliefs have confidence levels and priorities, which get revised as the system gathers new evidence.
- Inference rules: The system uses inheritance, revision, choice, and question-based inference to derive new beliefs and respond to input.

This methodology allowed my game agent to "think" in a way that's more human-like than deterministic AI scripts.

Results

The AI demonstrated adaptive behavior in scenarios where traditional systems would struggle. For example, when the reward structure or obstacles changed mid-game, the NARS-based agent adapted more efficiently without needing a reconfiguration or retraining phase. The agent learned to prioritize certain resources over time and avoid paths that repeatedly led to failure.

Example Gameplay Scenario

In one test case, the agent was initially trained to collect "blue gems" which were placed near the bottom of the map. Midway through the game, I changed the environment to replace blue gems with "red gems" placed near hazards. A traditional rule-based agent would continue seeking blue gems or crash when it couldn't find them. However, the NARS agent observed the new rewards, inferred their value, and began adapting its pathfinding to target the red gems—while learning to avoid the hazards along the way. This real-time adaptation without rewriting the AI script is a strong indicator of the flexibility of NARS.

Figures and Code Snippets

Figure 1: Simple Game Grid
Diagram showing a 2D grid with player, resources, and hazards.

Code Snippet (Narsese Input):
```
<(*,agent,resource) --> goal>.
<(*,hazard,agent) --> avoid>.
```

Code Snippet (Python integration):
```
nars_input = "<(*,agent,resource) --> goal>."
nars.run(nars_input)
```

Challenges

One of the main challenges was learning the syntax and reasoning patterns of Narsese. It took time to properly structure inputs and interpret outputs from NARS in a way that made sense in the game environment. Additionally, integrating NARS into an interactive loop required creative thinking about time management, decision latency, and feedback from the environment.

Another issue I encountered was performance. As the number of beliefs increased, the reasoning cycle took longer. I had to experiment with ways to limit memory and cycle usage to maintain responsiveness in the game.

Related Work and Applications

Other adaptive AI approaches in games include:
- Behavior trees: Modular, but still largely rule-based.
- Goal-Oriented Action Planning (GOAP): Popular in AAA games but assumes a fixed goal hierarchy.
- Reinforcement Learning (RL): Powerful but requires extensive training data.

Compared to these, NARS offers reasoning without extensive training and adapts through interaction rather than repetition. Outside gaming, NARS has been proposed for use in robotics, conversational agents, and autonomous decision-making systems.

Conclusion

Working on this project has deepened my understanding of general AI frameworks like NARS and their practical applications. It also made me appreciate the complexity involved in building AI that can reason and adapt under real-time constraints. I believe that further development in this area could lead to more immersive and intelligent gaming experiences.

Future Work

In the future, I plan to expand this project by introducing more complex game elements, such as enemy agents, cooperative AI behavior, or narrative decision-making. I also want to explore ways to optimize NARS performance in larger-scale environments.

References

Wang, P. (2013). Rationale of Non-Axiomatic Logic. Journal of Experimental & Theoretical Artificial Intelligence.
Wang, P. (2006). Non-Axiomatic Logic: A Model of Intelligent Reasoning. World Scientific.
NARS Development Community. (2025). OpenNARS Documentation. Retrieved from https://github.com/opennars/opennars
Isla, D. (2005). Handling Complexity in the Halo 2 AI. Game Developers Conference.
Millington, I., & Funge, J. (2016). Artificial Intelligence for Games (2nd ed.). CRC Press