NO PREP

AI in Video Games

NPCs and Procedural Worlds

Includes: Student reading, case study, differentiated questions, teacher key

MORE LESSONS LESS PLANNING · MLLP · FREE TODAY · FREE TOMORROW · FREE FOREVER

Name: Period: Date:

AI in Video Games: NPCs and Procedural Worlds

The Other Players You've Always Ignored

In nearly every video game you've ever played, there is a hidden layer of artificial intelligence operating constantly in the background. Non-player characters — the shopkeeper who offers you supplies, the guard who patrols the castle, the enemy soldier who takes cover when you open fire — are all controlled by game AI. This form of AI is older than most people realize: the ghosts in Pac-Man (1980) each had individual behavior rules that made them behave differently from each other. Pac-Man's AI is primitive by modern standards, but the design principle it embodies — programming distinct, responsive behavior into game characters — is still at the heart of every major game release today. Game AI is a discipline separate from the broader AI development happening at companies like Google and Anthropic. Game developers don't need

Name: **Period:** **Date:**

AI in Video Games: NPCs and Procedural Worlds

The Other Players You've Always Ignored

In nearly every video game you've ever played, there is a hidden layer of artificial intelligence operating constantly in the background. Non-player characters (the shopkeeper who offers you supplies, the guard who patrols the castle, the enemy soldier who takes cover when you open fire) are all controlled by game AI. This form of AI is older than most people realize: the ghosts in Pac-Man (1980) each had individual behavior rules that made them behave differently from each other. Pac-Man's AI is primitive by modern standards, but the design principle it embodies (programming distinct, responsive behavior into game characters) is still at the heart of every major game release today. Game AI is a discipline separate from the broader AI development happening at companies like Google and Anthropic. Game developers don't need their NPCs to be genuinely intelligent; they need them to appear intelligent and behave in ways that make the game fun and challenging. That distinction matters, because it means game AI is often built from tricks and clever illusions rather than genuine machine learning.

How NPC Behavior Is Built

The most common tool for creating NPC behavior is the finite state machine (FSM). An FSM defines a set of states a character can be in (patrolling, investigating, attacking, fleeing, idle) and a set of rules that trigger transitions between states. A guard NPC might: patrol a route (state 1) until it detects the player (transition trigger), then switch to alert mode (state 2), then attack (state 3) if the player gets within a certain range, then switch to fleeing (state 4) if health drops below 20%. More sophisticated games use behavior trees, which are hierarchical structures that let designers build complex, layered decision logic. A behavior tree might define a soldier NPC that first checks if it has ammo, then if it has cover nearby, then if

allies are present, and makes different decisions based on the answers to all three questions. Modern shooters like the Halo series, The Last of Us, and games in the Far Cry franchise have used behavior trees to create enemies that seem to communicate with each other, use flanking maneuvers, and adapt to the player's tactics: behaviors that feel remarkably intelligent even though no machine learning is involved.

Procedural Generation: When AI Builds the World

A separate and increasingly important form of game AI is procedural generation, using algorithms to create game content automatically rather than designing it by hand. The most famous modern example is No Man's Sky (2016), a space exploration game in which 18 quintillion unique planets exist, each with its own terrain, weather, plant life, and fauna. No human designed any of these planets. They are all generated by algorithms that combine parameters (temperature, gravity, atmospheric density) to produce coherent, explorable worlds. Minecraft is another landmark example. Its terrain generation algorithm creates an infinite world with mountains, caves, oceans, deserts, and forests that feel natural rather than random. Dungeons & Dragons-style games like Hades, Dead Cells, and Spelunky use procedural generation for level layouts, enemy placement, and item drops. This ensures that no two playthroughs are identical. This technique reduces development cost (you don't need to hand-design every room) while dramatically extending replayability.

Machine Learning Enters the Game

Traditional game AI uses hand-coded rules. Machine learning game AI — still relatively rare in commercial games — can actually learn from data. DeepMind's AlphaGo (2016) and AlphaZero (2017) are the most famous examples of machine learning applied to games: an AI trained by playing millions of games against itself until it surpassed the world's best human players at Go, chess, and shogi. These achievements were milestones in AI research broadly, not just in gaming. Commercial game companies have begun experimenting with machine learning for NPC behavior, most notably in racing games where AI drivers can be trained to race rather than scripted to follow a line. Electronic Arts has published research on using reinforcement learning to train NPCs in FIFA to play soccer with human-like patterns. Unity and Unreal Engine — the two dominant game development platforms — both now offer built-in machine learning toolkits for developers who want to

experiment with learning-based NPCs. The frontier that excites game developers most is using large language models to give NPCs genuinely dynamic conversations. Current NPCs can say perhaps 50-200 scripted lines. An NPC powered by a language model could have an open-ended conversation about anything — remembering what you've told it, responding to questions the developer never anticipated, and evolving its relationship with the player over time. Prototype demos of this technology appeared in 2023 and 2024, though the computational cost and content control challenges of deploying LLM-driven NPCs in commercial games remain significant.

Why This All Matters Beyond Gaming

Game AI is not just an entertainment technology. Many of the foundational techniques in real-world AI were developed or refined in gaming contexts. Reinforcement learning — the technique behind AlphaGo's world-beating performance — is the same technique used to train real-world robotics systems and to fine-tune large language models. Procedural generation techniques are used in architectural design, drug discovery, and simulation training for autonomous vehicles. The entertainment industry has always been one of the earliest and most creative adopters of new AI capabilities — and what gets tested in games often migrates into the broader world.

Procedural Generation and the Illusion of Infinite Worlds

Modern open-world games face a fundamental production problem: players explore far more content than any development team could hand-craft in reasonable time. Procedural generation solves this by using algorithms to create content mathematically rather than manually. No Man's Sky generated over 18 quintillion unique planets using a system of mathematical rules that determined terrain shapes, atmospheric conditions, flora, and fauna. Minecraft's world generation uses a technique called Perlin noise to produce terrain that looks organic and varied despite being mathematically generated. The critical design challenge is ensuring that procedurally generated content feels meaningful rather than arbitrary. Games solve this through layered systems — a procedural algorithm generates the landscape, but handcrafted story beats, landmarks, and structured encounters are then placed within it. The AI handles the infinite canvas; human designers ensure the player has reasons to explore it.

Adaptive Difficulty and Machine Learning NPCs

Traditional NPC behavior is scripted: if the player does X, the NPC does Y. This produces predictable opponents that experienced players learn to exploit quickly. Game developers are experimenting with machine learning to create NPCs that adapt to individual player behavior over time. Some commercial games now use machine learning to dynamically adjust difficulty based on player performance, ensuring the challenge remains engaging without explicit difficulty settings. These systems analyze metrics like death rate, time spent on specific sections, and accuracy to infer whether a player is struggling or breezing through content. The goal is a game that feels tuned specifically to each player's skill level without the player ever touching a difficulty slider. This approach represents a meaningful shift from fixed difficulty curves toward personalized challenge systems that respond to actual player behavior rather than a preset assumption of skill progression.

Key Information: AI in Video Games

Detail	Information
Oldest Game AI Example	Pac-Man (1980) — each ghost had individual behavior rules
Finite State Machine	Defines NPC states (patrol, attack, flee) and rules for switching between them
Behavior Trees	Hierarchical decision logic used in Halo, The Last of Us, Far Cry for complex NPC behavior
Procedural Generation	Algorithms that auto-generate game content — No Man's Sky has 18 quintillion unique planets
Machine Learning in Games	DeepMind AlphaGo/AlphaZero surpassed human players at Go, chess, and shogi
LLM-Driven NPCs	Prototype technology (2023-2024) enabling open-ended NPC conversation; not yet commercially deployed at scale
Key Platforms	Unity and Unreal Engine both offer ML toolkits for developers

Detail	Information
Real-World Connection	Reinforcement learning from games is used in robotics and LLM training

Case Study: The Guard Who Wouldn't Stay Fooled

Tyrell is 15 and a serious gamer. He's been playing a stealth game called 'Covenant,' and he's stuck. There's a guard NPC he needs to sneak past, but every trick that worked on other guards in the game isn't working on this one. He tries the distraction throw: tosses a rock to lure the guard away. Works the first time. He tries it a second time — same spot, same rock. The guard investigates but returns faster and seems noticeably more alert. Third attempt, same location: the guard doesn't move at all and instead starts sweeping the exact area where Tyrell is hiding. Frustrated, Tyrell posts in a gaming forum: 'Is this guard AI bugged or is it actually learning from my behavior?' A veteran player responds: 'It's a behavior tree. After the first distraction, a flag triggers that makes the guard treat that entire area as elevated risk. He's not truly learning — there's no model updating here — but he IS adapting to a new internal state. Those are very different things, and that distinction matters a lot when we talk about real AI.'

1. Explain the difference between 'learning' and 'adapting' as the veteran player describes it. Why does that distinction matter when we talk about AI intelligence?

2. The reading says game AI is often built from 'tricks and clever illusions rather than genuine machine learning.' Do you think it matters whether an AI is truly learning or just appearing to? What are the implications of that distinction?

Multiple Choice

1. What is a finite state machine (FSM) in the context of game AI?

a) A system that defines NPC behavior states and the rules that trigger transitions between them

b) A procedural algorithm that generates random enemy placements in each game level

c) A scoring and difficulty adjustment system that responds to player performance metrics

d) A machine learning system that trains its NPCs by running millions of simulated self-play games

2. What made DeepMind's AlphaGo and AlphaZero significant milestones?

a) They created the first procedurally generated game world with over a trillion unique locations

b) They demonstrated that game-derived AI techniques could be used to control autonomous vehicles in development

c) They were the first game AIs ever created, marking the birth of the game development AI field

d) They used machine learning to surpass the world's best human players at Go and chess without hand-coded rules

3. What is one reason LLM-driven NPCs have not yet been widely deployed in commercial games, despite promising demos?

a) Players in surveys said they preferred scripted dialogue over open-ended conversations

b) Game consoles lack the hardware to run language models at all

c) The technique was patented and is available only to a single developer

d) The computational cost and content control challenges remain significant obstacles

Matching

Write the letter of the correct definition on the line next to each term.

Term	Definition
1. ___ Non-Player Character (NPC)	A) A system defining a fixed set of character states and the rules that trigger movement between them
2. ___ Procedural Generation	B) A hierarchical decision structure that allows NPCs to check multiple conditions and make layered choices
3. ___ Behavior Tree	C) A game character controlled by AI rules rather than a human player
4. ___ Reinforcement Learning	D) An algorithmic process that auto-generates game content such as terrain, levels, or enemies rather than designing them by hand
5. ___ Finite State Machine (FSM)	E) A machine learning technique where an AI learns by trial and error, receiving rewards for correct actions

Fill-in-the-Blank

1. The game _____ Sky contains 18 quintillion unique procedurally generated planets.

2. DeepMind's _____ was a machine learning system that surpassed the world's best human Go player in 2016.

3. The reinforcement learning technique developed and refined in gaming is also used to train _____ systems and fine-tune large language models.

Short Answer

Answer each question in 2-4 complete sentences.

1. Explain what procedural generation is and why it is valuable for game developers. Use at least one specific example from the reading.

2. The reading says game AI is 'a discipline separate from the broader AI development' happening in tech companies. What does this mean, and what is the key difference in what each type of AI is trying to achieve?

3. The reading argues that gaming has been an important testing ground for AI techniques that later move into the real world. Do you find this argument convincing? Explain using at least one specific connection from the reading.