The Game That Builds Itself Around You

Benjamin Llados
WRIT 340 - Professor Townsend
Fall 2026

Abstract

Video games have used algorithms to generate worlds and content for decades, but generative AI is starting to add another dimension. This article explores how generative AI could move beyond assisting game developers and begin operating inside games themselves. From characters that develop personalities through player interactions to worlds that respond to player decisions, these systems introduce major possibilities for the gaming industry. This article examines some of the newest innovations in this space, the limitations they face today, and where the technology could go next. But these advances also create deeper questions about game design. Does more content actually make a game more meaningful? How could AI transform players into creators, and what would that mean for the relationship between the developer, the game, and the player?

Introduction

Imagine you are playing a fantasy game and wandering around a town you just arrived in. You see an old man playing guitar outside a church; he intrigues you, so you go talk to him. He tells you there's a secret tunnel hidden beneath the building. Curious, you search for the entrance, find it, and climb underground. Maybe there is treasure inside. Maybe there is something much worse. Either way, the strange part isn't what you find. It's that, in this future game, the tunnel didn't exist until your conversation gave the world a reason to create it. No game developer placed that exact passage beneath the church waiting for you to discover it. Your interaction helped cause it to appear, and what happens inside that tunnel can later change the town above it. The game is no longer simply revealing a world that was already there. This raises a question: what if a video game could create parts of itself because of what you did while playing it?

Games that create content while you play aren't actually new. Developers have been using algorithms for decades to create parts of a game rather than manually building each piece [1]. This is called procedural content generation, or PCG. Traditional procedural systems rely on rules, constraints, and objectives that developers explicitly design. For example, a Minecraft world generator follows a system built to decide what kinds of terrain and structures appear. Snow can only appear at certain heights or biomes; bamboo can only grow in jungles, etc.

Modern generative AI is another way of generating content, using models that learn patterns from existing content instead of relying solely on rules written by developers [2]. That does not mean procedural generation is just random while generative AI is somehow "creative." In fact, the two approaches can overlap, and modern systems can combine learned models with developer-written rules and constraints [1], [2]. The real difference is that learned systems can potentially respond to much richer inputs, opening the door for games to generate beyond where a player goes, but to what they say, ask for, and do.

Characters that Remember You

Imagine you meet a companion early in the game who is optimistic, trusting, and eager to help you. Over the next several hours, your actions say otherwise. You constantly lie to them, make reckless decisions, and ignore their advice. Eventually, instead of reacting to you the same way every time, they start becoming more guarded and suspicious. A simple decision tree might check how many times you have lied to a companion, then choose a trusting or suspicious response from options the developer wrote ahead of time. But generative AI opens up another possibility: letting a character's past experiences influence how it responds to you in the future. Researchers at Stanford and Google demonstrated one version of this idea with 25 generative agents placed inside a simulated town. The agents stored observations about what happened around them, retrieved relevant memories, formed higher-level reflections, and created plans that influenced their behavior as time went on [3]. The experiment only lasted two simulated days, so it does not prove that a game character could develop a believable personality across a 40-hour campaign. What it does show is the basic architecture for a character whose past can dictate its future.

Giving a character a memory isn't as simple as saving everything that has ever happened to it. Let's go back to that same companion; say they have traveled with you for twenty hours. By then, the game could have recorded hundreds or even thousands of conversations, decisions, and events. The companion can't consider every single interaction every time you speak to them. That would take an incredible amount of memory and hardware inaccessible to most gamers. For your choices to shape the character's behavior, the game needs to decide which memories matter for each response.

In the *Generative Agents* experiment, stored memories were instead ranked according to three factors: recency, importance, and relevance to the current situation [3]. The system would then retrieve the memories most useful for what the agent was doing at that moment. For example, when an agent was preparing for a party, memories related to the party were prioritized over unrelated observations from around the house [3]. This creates an important distinction: storing an event and knowing when to remember it are two separate engineering problems. A character may technically have a record that you betrayed them ten hours ago, but that memory only

matters if the system knows when to recall it. As Fig. 1 shows, that retrieved memory can then become one of several inputs that shape what the character says or does next.

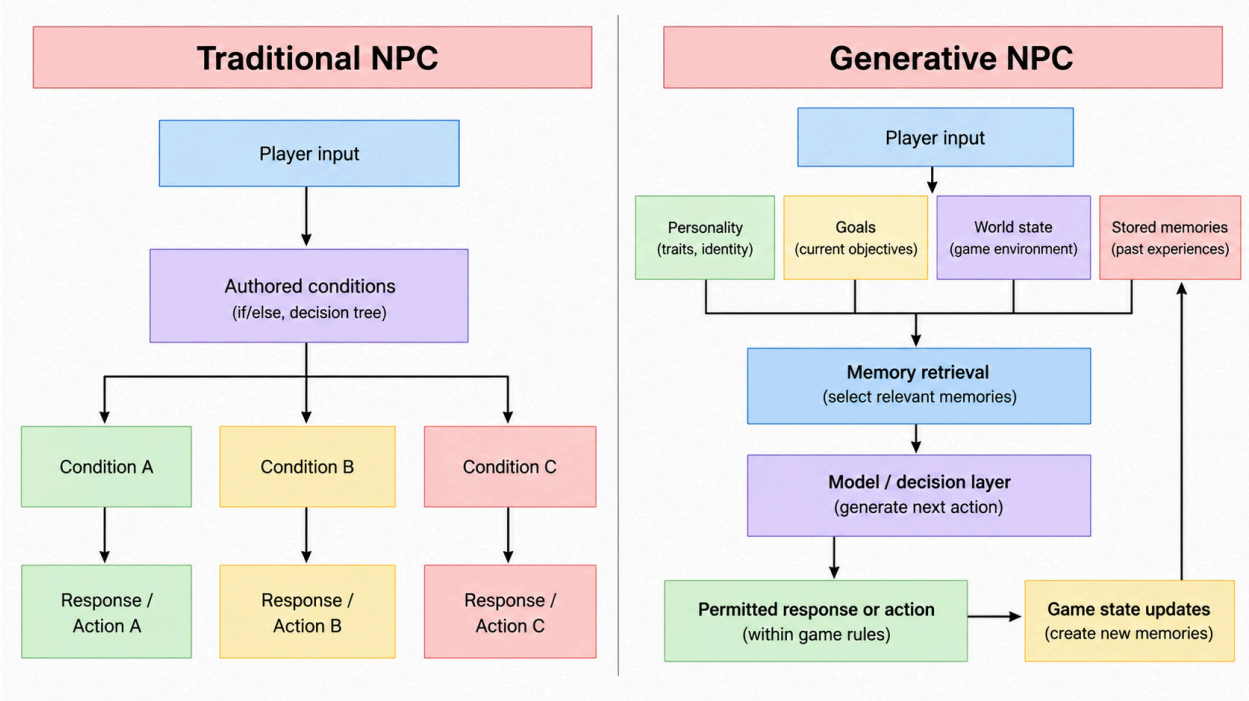


Figure 1. Traditional vs. generative NPC decision pipelines [3], [4].

Even if an AI character can remember what happened, there is still another problem: saying it wants to do something doesn't mean the game knows how to make it happen. Imagine you ask your companion to climb onto the roof of a nearby building. A language model could generate the response, "Sure, follow me," but the game still needs to know where the roof is, whether the character can reach it, what animation to use, and how that action changes the state of the world. NVIDIA's ACE framework shows this type of integration problem, describing the need for AI characters to synchronize with "global and dynamic game states" and connect their reasoning to the tools and actions the game supports [4]. This is one of the main bottlenecks in AI NPCs; they can't just be a chatbot dropped into a game. Their words and decisions have to connect back to the core mechanics that make the game work.

World Generation

If generative AI can make characters more responsive, the next question is what happens when the world itself starts responding too. For example, this past summer I went on a trip to Croatia and fell in love with Dubrovnik. As a game designer, I can turn this love for a city into a new world. Thinking back to what made me love it, I remember: narrow stone streets, massive castle walls, warm street lights, turquoise water, painted windowsills, flower beds, and people sitting

outside their doorsteps. Now imagine taking those same details and asking for whatever game I want inside a world generated exactly like Dubrovnik. Maybe I turn it into a racing game, where go-karts race around the city at sunset, or even a tactical shooter, where the tight corners create moments of suspense. Instead of choosing from a world that a developer has already built, you could help describe the kind of place you want to explore. That possibility is still far beyond what today's systems can reliably create, but researchers are already working on models that can generate interactive environments in response to what a player does, letting your imagination guide you to the places you most want to see.

One way researchers are approaching this idea is through world models. Instead of starting with a completely built game world, these models use what the player is currently seeing along with the action they take to predict what should happen next. If the player moves forward, turns left, or interacts with something, the model will generate the next view in response. Google's Genie 3 demonstrated this kind of interactive generation at 720p and 24 frames per second, with Google reporting that generated environments could remain visually consistent for several minutes [5]. Another system, Microsoft's WHAM, was trained to predict sequences that include both images of a game and controller actions, allowing those actions themselves to become part of the model's inputs to predict what comes next [6]. These systems can start to look like playable worlds, but they are still far from complete game engines. They still struggle with long-term consistency, precise control of elements, and maintaining everything that has happened in the world over time [5], [6].

Google DeepMind notes that during longer Genie 3 generations, "inaccuracies tend to accumulate over time" [5]. Microsoft's WHAM has also tested whether visual changes remain present in later generated frames, showing that models can preserve some edits over short periods [6]. But visually remembering a change for the next few seconds is different from recording that event as a part of the world's permanent state.

For example, you destroy the only bridge connecting two towns. The game now has much more to remember than just whether the bridge still exists. Can NPCs still cross the river? Do merchants have to reroute? Does trade between the two towns slow down? How does this affect the quests in the game? Who knows you destroyed it, and are they mad?

Games already use persistent data systems to "store data that needs to persist between sessions," meaning information can survive even after a player leaves and comes back later [7]. But storing the fact that a bridge is gone doesn't automatically create all of the consequences that should follow from it. These questions are what separate a fun generative AI project from a fully developed game. Those consequences still have to connect to navigation, characters, quests, and the story itself. Earlier interactive systems such as *Façade* approached a similar narrative problem by using authored story beats that were selected and sequenced during play [8]. A truly

generative world would have to coordinate many of these systems at once. Showing the bridge collapse is just the starting point; making the rest of the world understand what that collapse means is the true problem.

Generate a Car

To understand why that is so difficult, shrink the problem down from an entire world to a single object: a car. Generating an image of a sports car is relatively straightforward, but what happens when you ask the game to let you drive it? Roblox is already experimenting with this distinction through what they call 4D generation, which they say "adds the dimension of interactivity" to the generated 3D objects [9]. In its 2026 beta, Roblox reported that players in their 4D generation game called *Wish Master* could generate vehicles and then actually drive or fly them inside the game [9]. For generative gaming, creating something convincing to look at is only the first step. It also has to convince the game that it is actually a car.

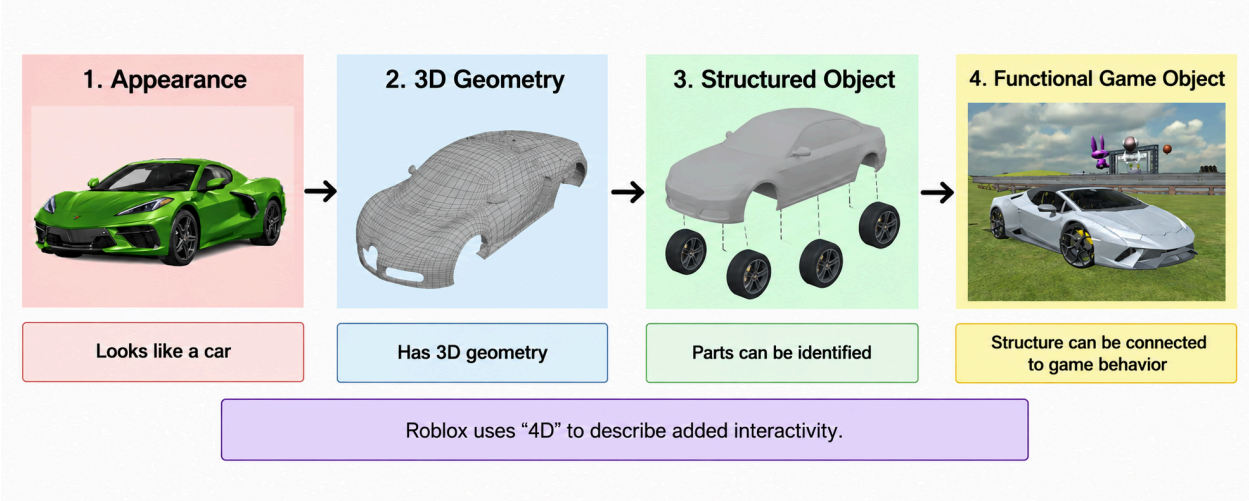


Figure 2. From generated car appearance to functional game object [9]–[11].

Roblox's current generation pipeline shows how many layers sit between an object that looks like a car and one that the game can actually use. In Roblox's model-generation system, one car format called Car5 breaks it down into five separate pieces: one body and four wheels [10]. Those pieces give the game structure it can work with, but they still don't make the car drive. Roblox documents a separate step where a behavior script, a small piece of code attached to a game object that tells it how to act, is attached to the generated model and adapted to its geometry [10].

Roblox's newer CubePart research takes this idea a step further. Instead of generating one complete 3D object and treating it as a single piece, the system can generate the object with specific parts in mind, such as the body and wheels of a car [11]. That helps the object map user

inputs to parts, rather than attaching them to a whole object. The system is still far from perfect. Researchers found cases where generated parts overlapped or where the model placed a named part in the wrong location [11]. A car could look convincing to a player while the game is still lost on what a wheel's function is. As Fig. 2 shows, generating a car's appearance, geometry, structure, and behavior are all separate problems the system has to solve.

Looking beyond a single generated object, researchers have argued that a generator should be judged by its expressive range, or the "style and range of levels that it can create," rather than just by the impressive examples a company puts forward [12]. In other words, seeing one incredible generated object doesn't tell us anything about the depth of a system. Generative AI can make a game feel almost limitless in what you can ask for, but every request still has to connect to the mechanics the game knows how to simulate. The limit isn't what the AI can describe; it's what the game can actually do.

What Creates Meaning?

Even if games overcome these technical limitations, there is another question: does generating more content actually make a game better? Imagine walking through a city where every single NPC can hold a completely different conversation with you. You can ask them about their childhood, their fears, the city around them, anything you want. At first, you might think that's incredible, but in reality, how long would it take for you to get bored? Game writer Emily Short has discussed a similar problem through the idea of "bowls of oatmeal": a generator can create many outputs that are technically different, while still feeling basically the same to the player [13]. The oats might be arranged differently in every bowl, but you're still eating the same breakfast. The challenge isn't just producing more dialogue; it's making that dialogue connect to a deeper meaning. Generative AI could give these NPCs far more to say, but that doesn't automatically make it meaningful.

AI has a unique role in games because players have been helping shape the experience long before AI entered the picture. A generated image or song can exist as a finished piece before anyone sees it, but a game only becomes an experience once someone interacts with its systems. The MDA framework for game design breaks this relationship down into three parts. It starts with mechanics, the rules and systems built by developers; then moves to dynamics, what happens when those systems are actually played; and finally to aesthetics, the experience and emotions produced by the player [14]. Research on player agency makes a similar point, describing agency as "a phenomenon involving both player and game" [15]. That raises the question: are games unusually suited to AI because the audience was already helping create the experience?

That relationship between player and game dates way back to early gaming. Think back to Minecraft, for example. One player could be peacefully building a house on the surface while his friend might immediately dig straight down into lava. They are playing the same game with the same rules, but their experiences are extremely different because of the choices they made. Game scholar Jesper Juul describes this as “simple rules combining, leading to variation” [16].

Generative AI could push that relationship even further. The developers would still create the systems that make the experience possible. The generative model could create a specific character, conversation, or space within those boundaries. But your choices as a player would help create that experience [8], [15]. As Fig. 3 shows, none of those pieces has to replace the others. The most revolutionary part of generative AI in gaming isn’t the excess content created. Instead, it’s the relationship created between the developer and player.

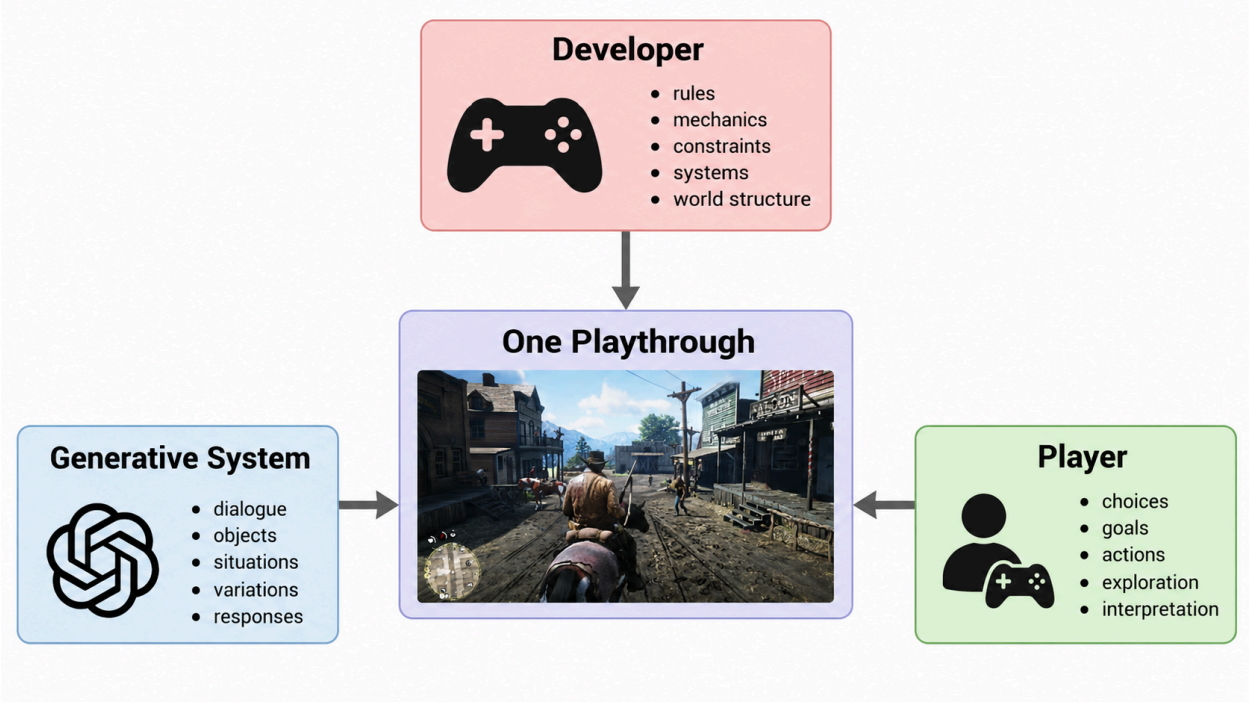


Figure 3. Developer, AI, and player shaping a single playthrough [8], [15], [16].

Think back to that secret tunnel underneath the church. Imagine you find something inside, return to the surface, and tell a friend playing the same game what happened. They search for the old man, but he isn’t there. There is no tunnel under their church either. Eventually, they ask you, “Was that actually part of the game?” In the future game described here, yes: the tunnel would be part of your playthrough, even though it wasn’t placed there ahead of time. The systems discussed in this article show pieces of that possibility, but generating the tunnel, making it playable, and preserving its consequences would all have to work together.

If you wanted to explore it with your friend, the game would need to preserve the tunnel in a shared world. It would also need boundaries on how one player's choices change that world for everyone else. The goal would be to give players more room to create while keeping the game enjoyable to play together.

Playing a game could look a little more like helping create one.

References

[1] J. Togelius, N. Shaker, and M. J. Nelson, “Introduction,” in *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Cham, Switzerland: Springer, 2016, pp. 1–15. [Online]. Available: https://www.pcgbook.com/chapter01.pdf

[2] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural content generation via machine learning (PCGML),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, Sep. 2018, doi: 10.1109/TG.2018.2846639. [Online]. Available: https://arxiv.org/abs/1702.00539

[3] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative agents: Interactive simulacra of human behavior,” in *Proc. 36th Annu. ACM Symp. User Interface Software and Technology (UIST)*, 2023. [Online]. Available: https://arxiv.org/abs/2304.03442

[4] P. Singh, “Build on-device AI companions with the NVIDIA ACE Game Agent SDK and Unreal Engine 5 plugins,” *NVIDIA Technical Blog*, Jun. 16, 2026. [Online]. Available: https://developer.nvidia.com/blog/build-on-device-ai-companions-with-the-nvidia-ace-game-agent-sdk-and-unreal-engine-5-plugins/

[5] J. Parker-Holder and S. Fruchter, “Genie 3: A new frontier for world models,” *Google DeepMind*, Aug. 5, 2025. [Online]. Available: https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/

[6] A. Kanervisto et al., “World and human action models towards gameplay ideation,” *Nature*, vol. 638, pp. 656–663, Feb. 2025, doi: 10.1038/s41586-025-08600-3. [Online]. Available: https://www.nature.com/articles/s41586-025-08600-3

[7] Roblox, “Data stores,” *Roblox Creator Hub*. Accessed: Sep. 27, 2026. [Online]. Available: https://create.roblox.com/docs/cloud-services/data-stores

[8] M. Mateas and A. Stern, “Structuring content in the Façade interactive drama architecture,” in *Proc. Artificial Intelligence and Interactive Digital Entertainment Conf. (AIIDE)*, vol. 1, no. 1, pp. 93–98, 2005. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/18722

[9] A. Singh, “Accelerating creation, powered by Roblox’s Cube foundation model,” *Roblox Newsroom*, Feb. 4, 2026. [Online]. Available: https://about.roblox.com/newsroom/2026/02/accelerating-creation-powered-roblox-cube-foundation-model

[10] Roblox, “Model generation,” *Roblox Creator Hub*. Accessed: Sep. 27, 2026. [Online]. Available: https://create.roblox.com/docs/parts/model-generation

[11] Y. Zhu, K. Deng, J.-P. Fauconnier, I. Navarro, D. Li, A. Pun, Y. Zhang, P. Zhuang, X. Sun, M. Agrawala, K. Bhat, and T. Zhou, “CubePart: An open-vocabulary part-controllable 3D generator,” *SIGGRAPH 2026*, 2026. [Online]. Available: https://arxiv.org/abs/2605.28763

[12] G. Smith and J. Whitehead, “Analyzing the expressive range of a level generator,” in *Proc. Workshop on Procedural Content Generation in Games*, 2010, doi: 10.1145/1814256.1814260. [Online]. Available: https://users.soe.ucsc.edu/~ejw/papers/smith-pcg-2010.pdf

[13] E. Short, “Bowls of oatmeal and text generation,” *Emily Short’s Interactive Storytelling*, Sep. 21, 2016. [Online]. Available: https://emshort.blog/2016/09/21/bowls-of-oatmeal-and-text-generation/

[14] R. Hunicke, M. LeBlanc, and R. Zubek, “MDA: A formal approach to game design and game research,” in *Proc. AAAI Workshop on Challenges in Game AI*, 2004. [Online]. Available: https://www.cs.northwestern.edu/~hunicke/MDA.pdf

[15] N. Wardrip-Fruin, M. Mateas, S. Dow, and S. Sali, “Agency reconsidered,” in *Proc. Digital Games Research Association Conf. (DiGRA)*, 2009. [Online]. Available: https://cs.uky.edu/~sgware/reading/papers/wardripfruin2009agency.pdf

[16] J. Juul, “The open and the closed: Games of emergence and games of progression,” in *Computer Games and Digital Cultures Conference Proceedings*. Tampere, Finland: Tampere University Press, 2002, pp. 323–329. [Online]. Available: https://jesperjuul.net/text/openandtheclosed.html

Supplementary

Further Reading

1. J. S. Park et al., "Generative Agents: Interactive Simulacra of Human Behavior."
2. J. Parker-Holder and S. Fruchter, "Genie 3: A New Frontier for World Models."
3. A. Singh, "Accelerating Creation, Powered by Roblox's Cube Foundation Model."

Multimedia Resources

- J. S. Park et al., "Generative Agents: Smallville Simulation Replay."
- Google DeepMind, "Genie 3: Creating Dynamic Worlds That You Can Navigate in Real-Time." Video demonstration.
- Y. Zhu et al., "CubePart: Interactive 3D Gallery and Behavior Demonstrations."
- Big A, "Everything Is Everything Now." Video.

Search Terms

Generative AI, video games, procedural content generation, non-player characters (NPCs), AI memory, world models, 3D object generation, player agency, interactive storytelling, game design.

About the Author

Benjamin Llados is a senior at the University of Southern California's Marshall School of Business, majoring in business administration with a minor in artificial intelligence applications from the Viterbi School of Engineering. His love of video games began when he received a Nintendo Wii for his fifth birthday. Outside of gaming, he enjoys playing soccer and tennis and is passionate about using his education and career to create meaningful change in communities that need it most.