Research Statement

Sam Earle — NYU — sam.earle@nyu.edu

Research Vision

I work on open-ended learning in video games. This generally takes the form of systems that co-evolve games and game-playing agents, with games becoming increasingly challenging and complex as player agents become more general and robust.

My motivation is to build better, more general, more adaptive artificial intelligence. I'm doubtful that the current paradigm of large models trained on massive amounts of static data will be sufficient for generally capable AI. In particular, I suspect that without sufficient grounding in interactive environments, these models cannot have a true "world model", an *umwelt* that gives real meaning to the symbols they manipulate and enables their recombination into genuinely novel, creative new forms. As a result, I focus on systems that can automatically generate a plethora of rich embodied environments for learning agents. In their limit, I envision these open-ended systems generating environments that more closely resemble Artificial Life-like ecosystems (think games like *SimCity*, *MineCraft*, interactive Cellular Automata) than more conventional, single-objective video games (*Pong*, *Super Mario Bros.*, *Halo*). In this conception, these systems would be automatically searching through the space of environment substrates for the life-bearing needle in the haystack, using signals from agent learning to identify environments supporting complexification and emergence.

In what might seem like something of an ironic twist, then, my recent work moves toward the integration of these very LLMs and VLMs into the "driver's seat" of such open-ended environment generation loops. By using these models as mutation operators or fitness functions in the open-ended evolution of video game-like environments—e.g. by having them write and iterate on code describing game mechanics, or having them evaluate the interestingness of generated games based on playtraces—we can probe their capacity for genuine creativity. In turn, we can use these models to regularize open-ended search over radically unconstrained and flexible environment substrates, using them to guide agent-environment dynamics toward (or away from) human-interpretable semantic priors by varying degrees.

If we're optimistic about the capabilities of large pretrained models, we can aspire to automatically illuminate the space of games as we know them, and perhaps push beyond the frontier toward genuinely novel experiences. If it turns out that these models indeed lack the kind of embodied intelligence that leads to human-grade creativity, then training or fine-tuning them in the environments generated by these open-ended processes will provide a means of incrementally closing this gap. And if instead these models have some more fundamental limitation—e.g. in their architecture or learning algorithms—they may nonetheless provide a means of automatically identifying some human-recognizable signs of life from their more intelligent successors among the substrate.

Contributions to Date

My PhD research can be chronicled in three phases. In the first, I played a central role in establishing a new paradigm of AI-driven Procedural Content Generation (PCG), using reinforcement learning and quality diversity evolutionary approaches to train neural content generators that are both controllable with respect to functional gameplay-relevant metrics and fast at run-time. A key, recurring finding of this line of work was that—perhaps counter-intuively—the task of generating a globally coherent game level is best tackled by distributing it across a number of spatialy disparate, local agents.

The second phase leveraged this insight to design an efficient video game environment substrate based on local rewrite rules—showing how a broad swath of grid-based games can be encoded as cellular automata—and applied open-ended learning techniques to generate novel agent-environment dynamics within this substrate. With a vast number of seemingly complex and interesting but largely abstract and uninterpretable games and player agents on our hands, the question then became how to ground these environments in some human-interpretable semantics.

To this end, the third phase involved my pioneering a number of novel applications of large pre-trained models to PCG, finding ways to leverage their pre-existing knowledge while keeping them explicitly aligned with functional design constraints. To facilitate LLM-driven PCG in the context of open-ended learning, I extended our existing cellular automata-based environment substrate to be fully interoperable with a popular engine for grid-based puzzle games, allowing us to benchmark player agents on a large dataset of human-authored games, to render environments with visually interpretable sprites, and to leverage the innate coding abilities and language understanding of LLMs to generate entire game descriptions in the engine's human-readable domain-specific language.

Before my PhD, I published independent work that trained RL agents to play *SimCity* [Earle, 2019]. Incidentally, this foreshadowed much of my subsequent work: *SimCity*'s implementation largely reduces to a series of interacting cellular automata, it can be seen as an open-ended game with Artifical Life-like emergent properties and multiple objectives [Earle et al., 2021b], and the fully-convolutional networks that were the most capable in playing the game could themselves be seen as *Neural* Cellular Automata (NCAs).

Playing *SimCity* in some ways mirrors the process of designing a video game level, so it made sense to adapt the representation and architecture I had used in a collaboration that introduced PCG via RL [Khalifa et al., 2020] with a focus on level design. Unlike symbolic, "Good Old Fashioned AI"-flavored approaches (e.g. L-systems, answer set programming, binary space partitioning and random walks) that require designers to use domain-specific knowledge to design constructive recipes for generating content [Haugeland, 1989, Togelius et al., 2011], and data-hungry PCG via ((un)supervised) Machine Learning, which makes no guarantees about the playability/functionality of generated content [Summerville et al., 2018], PCGRL is data-free and only requires designers to express high-level heuristics or constraints about the desired content, resulting in generators that can efficiently adapt to a broad distribution of functional objectives at runtime [Earle et al., 2021a].

PCGRL has seen considerable uptake in the AI and Games community [Shu et al., 2021, Rupp et al., 2023, Zakaria et al., 2022], interest from industry , and has influenced the design of algorithms that use open-ended learning to train robust game-playing agents using automatically-generated curricula of levels [Parker-Holder et al., 2022]. We've observed that limiting agents' observation windows to small local patches [Earle et al., 2024a], increasing the number of agents [Earle et al., 2025a], and in the most extreme case using quality diversity to evolve archives of NCA-based agents that model level design as a fully-distributed and parallelized morphogenetic process [Earle et al., 2022] all lead to more globally coherent and scalable design strategies. In a complementary project that trained NCAs to solve pathfinding problems over mazes via supervised learning [Earle et al., 2023], we sought to illuminate these models' empirical effectiveness, showing by construction that they could directly implement search-based pathfinding algorithms. The same intuition can explain their success in the related problem of grid-based level generation.

Noting the surprising expressive power of NCAs coupled with their extreme simplicity and efficiency when run on the GPU, I endeavored to design a game description language that used them as an implementation substrate. The goal was to push open-ended reinforcement learning beyond training agents on distributions of *levels* for a single game, and toward the exploration of a vast space of evolvable *mechanics*, while optimizing speed so as to support research at an academic scale. The result was *Autoverse* [Earle and Togelius, 2025], a GPU-accelerated game engine centered on symbolic pattern rewrite rules capable of expressing mazes, *Sokoban*-likes, dungeon-crawlers, and arbitrary interactive cellular automata environments for training embodied player agents.

Using the effective depth of successful search-based player agents as a stable proxy for environment complexity and difficulty, and evolving curricula of increasingly complex *Autoverse* environments according to this objective, we showed that agents trained on later checkpoints from evolution generalized better to held-out environments. (Warm-starting player agents by imitation learning on trajectories from search similarly improved performance.) Meanwhile, the fully online approaches to curriculum generation from prior work often led to degraded generalization. This discrepancy is perhaps owing to the uniquely expansive space of tasks presented by *Autoverse*, which calls for more careful maintenance of a stable archive of prior tasks to to inhibit catastrophic forgetting. This opens up salient lines of future work in developing objectives for automatic environment generation that leverage signals from hybrid search- and learning-based player agents.

At the same time, amidst the surge in capabilities of large pretrained models, and acknowledging the visual "soupiness" of *Autoverse*'s generated environments, I began to explore methods for LLM/VLM-driven PCG in an effort to take advantage of the human priors latent in these models. In *DreamCraft* [Earle et al., 2024b] (developed during an internship at Meta FAIR; awarded best paper at FDG), I modified a text-conditioned Neural Radiance Field to output distributions over *MineCraft* block types—making careful use of annealed quantization and differentiable rendering of game-specific assets—to allow users to generate in-game structures from free-form prompts. To address functional constraints, I introduced domain-specific loss terms based on differentiable computation of proportion of block types and invalid adjacencies (prohibiting e.g. floating sand or water). In *DreamGarden* [Earle et al., 2025c] (developed during an internship at Microsoft Research; awarded best paper at CHI), I developed a system that dynamically orchestrated LLMs, VLMs, and diffusion-based 2D and 3D asset generators to iteratively design simulations in Unreal Engine from free-form prompts. The system included a GUI that exposed its hierarchical development plan as a tree supporting real-time editing, and a preliminary user study probed the system's co-creative potential and the particular design affordances of this novel interface.

With the goal of bringing these LLM-driven PCG strategies to open-ended learning loops, I extended *Autoverse* to be interoperable with PuzzleScript, a popular online engine for grid-based puzzle games with a concise but expressive domain specific language based on rewrite rules. The resultant framework, *PuzzleJAX* [Earle et al., 2025d], stands on its own as a novel GPU-accelerated benchmark for player agents. Many of the human-authored games in the online corpus are simple in terms of their state space—making them easy to solve with simple tree search—but deceptive in terms of their reward landscape—making them difficult for off-the-shelf RL methods. Their niche, often somewhat alien semantics (e.g. *Collapsoban*, in which nearby walls gradually close in on the player) seem to make them very challenging for frontier LLMs.

Most importantly, *PuzzleJAX*'s graphics and its domain specific language—both designed to be as minimal as possible without sacrificing expressivity—make it amenable to the study of LLM/VLM-guided open-endedness in an academic setting. And a vast online corpus of human-authored games provides a benchmark for the success of these open-ended processes. We've shown that LLMs can generate some functional PuzzleScript games given careful prompting and automated player feedback [Earle et al., 2025b]. But the greater question is whether, starting from something as simple as *Sokoban*, an LLM can iterate on game descriptions to rediscover the kind of inventive mechanics in the human PuzzleScript corpus. We expect that efficiently gathering feedback from a diversity of AI player will be crucial to identifying the "sweet spot" of complexity and difficulty of games generated by such a process. Perhaps the games generated along the way can be used to refine the creative problem-solving abilities of large models, leading to more general, embodied intelligence.

My near-term goal is to use *PuzzleJAX* to push the frontier of open-ended learning in mainstream AI venues, first establishing it as a challenging benchmark for RL and LLM player agents, then developing techniques for automatically generating curricula of environments for improving the performance of these agents.

At Sakana AI, I'm focusing on a pared-down benchmark of VLMs' capacity for open-endedness. *Picbreeder* [Secretan et al., 2008] was an online platform that allowed users to colloboratively generate images via interactive evolution of Compositional Pattern Producing Networks. The discovery of novel/recognizable images involved highly non-linear, curiosity-driven exploration; and given that intermediate phenotypes were not saved, there is no chance of VLMs having memorized these trajectories. By placing VLMs in the role of human users, and assessing their relative coverage of the space of meaningful images and the latent representations of evolved networks [Kumar et al., 2025], we can begin to quantify these models' capacity for open-endedness.

Future Work

By integrating large pretrained models with open-ended reinforcement learning algorithms and artificial life-like substrates and game environments, I will establish a research program poised to be at the forefront of the next wave of breakthroughs in artificial intelligence. While the near future of AI development is highly unpredictable, the family of methods that I'm developing—in which LLMs guide the generation of increasingly complex virtual worlds—will enable crucial contributions given a number of divergent possible outcomes. Such systems would equally allow us to reap the benefits of generally capable large pretrained models as generators of boundless creativity, to push them past the limitations of their training data by fine-tuning them in a growing array of procedurally-generated embodied environments, and to automatically discover substrates from which entirely new forms of intelligence may emerge.

There is no more compelling a medium for showcasing the creative capacities of AI than in the generation of video games, interactive simulations, or embodied virtual environments. The systems I develop are thus well-positioned to have high impact in mainstream discourse around AI, and to surface pressing questions around the ethics and risks of automating creative production. By extending these systems with user interfaces (e.g. allowing designers to interact with a genealogical tree of generated games in PuzzleScript's online editor), we can assess their potential as co-creative tools, evaluate their impact on designers, and collect human baselines allowing us to quantify the creative capacity of their fully autonomous counterparts. This will allow us to establish general principles for LLM-driven open-ended systems, exploring e.g. the relative advantages of various multi-agent collaborative configurations, of imbuing agents with diverse "personalities" or specializations via prompting or fine-tuning, and identifying settings in which these systems fall prey to communication breakdown, enter deletrious feedback loops, or exhibit unsafe behavior. These insights may in turn inform efforts to effectively organize large-scale, purely human collaboration.

The environments I generate may extend well beyond recreational video games to domains demanding varying degrees of functionality and immersiveness, from abstract coding problems, financial simulations, and autonomous driving tasks, to simulations for defense, robotics or training social workers. By carefully designing expressive and concise substrates, representations, and heuristics for these various domains, and guiding environment generation with pretrained models, we can generate a virtually boundless space of relevant tasks in which to train and evaluate humans, specialized agent models, or to push the generalization abilities of large pretrained models themselves. While considerable progress has been made in training black-box world models from unconstrained video data, there is an untapped competitive advantage to be found in leveraging existing domain-specific languages and symbolic PCG methods in open-ended environment generation, allowing for greater efficiency, interpretability and reliability, particularly when designing them as tools targeting domain experts.

These open-ended systems can also bring a novel perspective to Artificial Life, and the study of "bottom-up" forms of intelligence, in which the boundaries of the agent agent are not predetermined by the system's designer but rather emerge as a result of underlying environment dynamics. While such an approach is appealing in terms of its biological plausibility, its success is notoriously difficult to measure. Recent work has shown that ALife substrates (e.g. (neural) cellular automata) can be guided toward semantic targets using vision-language models [Kumar et al., 2024]. More generally, we can use such models to automatically identify novel or familiar forms and behaviors within environments (across varying timescales and spatial resolutions) without predefined labels, using them to detect and quantify initial signs of life and intelligence. By combining this with LLM-driven mutation over relevant substrates and DSLs, we can conduct a guided, automatic search to shed light on what forms of substrate, initial conditions, and interaction dynamics best facilitate emergent complexity and self-organization. In the long-term, we can scale this approach to evolve new forms of artificially intelligent agent that are inherently embodied and robust with respect to dynamic and complex environments.

References

Sam Earle. Using fractal neural networks to play simcity 1 and conway's game of life at variable scales. In *Proceedings of the AIIDE Workshop on Experimental AI in Games*, 2019. `https://arxiv.org/pdf/2002.03896`.

Sam Earle and Julian Togelius. Autoverse: Evolving symbolic neural cellular automata environments to train player agents. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 175–178, 2025.

Sam Earle, Maria Edwards, Ahmed Khalifa, Philip Bontrager, and Julian Togelius. Learning controllable content generators. In *2021 IEEE Conference on Games (CoG)*, pages 1–9. IEEE, 2021a. `https://ieeexplore.ieee.org/document/9619159`.

Sam Earle, Julian Togelius, and Lisa B Soros. Video games as a testbed for open-ended phenomena. In *2021 IEEE Conference on Games (CoG)*, pages 1–9. IEEE, 2021b.

Sam Earle, Justin Snider, Matthew C Fontaine, Stefanos Nikolaidis, and Julian Togelius. Illuminating diverse neural cellular automata for level generation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 68–76, 2022. `https://dl.acm.org/doi/pdf/10.1145/3512290.3528754?casa_token=BG2NZAQaENMAAAAA:zL5jvRHRqr6RUm9EQO-iJg8Tr-nSFY4EMiREwqSilHm2bY-dUZY0dDRFdZTN78HS3uAgor8dFdkQCA`.

Sam Earle, Ozlem Yildiz, Julian Togelius, and Chinmay Hegde. Pathfinding neural cellular automata. *arXiv preprint arXiv:2301.06820*, 2023. `https://arxiv.org/pdf/2301.06820`.

Sam Earle, Zehua Jiang, and Julian Togelius. Scaling, control and generalization in reinforcement learning level generators. In *2024 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2024a. `https://ieeexplore.ieee.org/iel8/10645493/10645533/10645598.pdf?casa_token=g23BP9hbfsMAAAAA:G64eklRMMHkiGbxWNu9hlUCFPOKO-sF21FFVL4X-auoXgxoGiyJ9iMC06l4JUl8bRT5T9uQ--_g`.

Sam Earle, Filippos Kokkinos, Yuhe Nie, Julian Togelius, and Roberta Raileanu. Dreamcraft: Text-guided generation of functional 3d environments in minecraft. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–15, 2024b. https://dl.acm.org/doi/pdf/10.1145/3649921.3649943?casa_token=zJqB9eIRLbUAAAAA:Ct0nl08pA9RPjEvxM4f31Ee-2iCqXTOyiOpq6gyN4OA9jxzXY8el8Oi15BOtUOogu8QMPircjiBoUQ.

Sam Earle, Zehua Jiang, Eugene Vinitsky, and Julian Togelius. Video game level design as a multi-agent reinforcement learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 21, 2025a. URL https://arxiv.org/abs/2510.04862.

Sam Earle, Ahmed Khalifa, Muhammad Umair Nasir, Zehua Jiang, Graham Todd, Andrzej Banburski-Fahey, and Julian Togelius. Scriptdoctor: Automatic generation of puzzlescript games via large language models and tree search. In *2025 IEEE Conference on Games (CoG)*, pages 1–5, 2025b. doi: 10.1109/CoG64752.2025.11114269.

Sam Earle, Samyak Parajuli, and Andrzej Banburski-Fahey. Dreamgarden: A designer assistant for growing games from a single prompt. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2025c. https://dl.acm.org/doi/pdf/10.1145/3706598.3714233?casa_token=LG8aydLkhYUAAAAA:Lr4tcOnuYbB27R-uE41GObI9RBhS-Sf2EqV_dL_7yDHX4UaIHID2nF-D2dcKx99lnUtjoOCLGOIM7Q.

Sam Earle, Graham Todd, Yuchen Li, Ahmed Khalifa, Muhammad Umair Nasir, Zehua Jiang, Andrzej Banburski-Fahey, and Julian Togelius. Puzzlejax: A benchmark for reasoning and learning. *arXiv preprint arXiv:2508.16821*, 2025d.

John Haugeland. *Artificial intelligence: The very idea.* MIT press, 1989.

Ahmed Khalifa, Philip Bontrager, Sam Earle, and Julian Togelius. Pcgrl: Procedural content generation via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 95–101, 2020. https://ojs.aaai.org/index.php/AIIDE/article/download/7416/7341.

Akarsh Kumar, Chris Lu, Louis Kirsch, Yujin Tang, Kenneth O Stanley, Phillip Isola, and David Ha. Automating the search for artificial life with foundation models. *arXiv preprint arXiv:2412.17799*, 2024.

Akarsh Kumar, Jeff Clune, Joel Lehman, and Kenneth O Stanley. Questioning representational optimism in deep learning: The fractured entangled representation hypothesis. *arXiv preprint arXiv:2505.11581*, 2025.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR, 2022.

Florian Rupp, Manuel Eberhardinger, and Kai Eckert. Balancing of competitive two-player game levels with reinforcement learning. In *2023 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2023.

Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1759–1768, 2008.

Tianye Shu, Jialin Liu, and Georgios N Yannakakis. Experience-driven pcg via reinforcement learning: A super mario bros study. In *2021 IEEE Conference on Games (CoG)*, pages 1–9. IEEE, 2021.

Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.

Yahia Zakaria, Magda Fayek, and Mayada Hadhoud. Procedural level generation for sokoban via deep learning: An experimental study. *IEEE Transactions on Games*, 15(1):108–120, 2022.