Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2025)

Embedded Mechanics Generation

Johor Jara Gonzalez
[1]University of Alberta
11011-88 Avenue
Edmonton, AB T6G 2G5 Canada
jaragonz@ualberta.ca

Abstract

Developing game mechanics is challenging due to the need for intricate design and programming. Procedural Content Generation (PCG) is a prevalent aspect of modern video game development, enabling the generation of content via algorithms. Achieving the desired balance and player experience is a multifaceted challenge, with game mechanics playing a crucial role—requiring thorough testing, player feedback, and iterative refinement. This work explores automated approaches to mechanic generation and evaluation, drawing from Automated Game Design (AGD). I present methods for generating mechanics, reconstructing levels through level inpainting, and creating enemies that can only be defeated using newly generated mechanics. Comparative studies between reinforcement learning agents and traditional static agents such as A* show that RL facilitates more diverse and human-like mechanic discovery, while static methods remain more stable but less creative. Ongoing work integrates these techniques into environments where mechanics, levels, and enemies coevolve, enabling richer evaluation of gameplay dynamics. To assess alignment between generated content and designer intent, I propose Design Impact Accuracy (DIA) as a metric to measure how effectively new mechanics are supported within AI-generated levels and enemies.

Introduction

Game mechanics are the rules and basic actions that define a game (Hunicke et al. 2004). Mechanics encompass actions and controls available to players, such as rules that allow for and limit movement, facilitate resource gathering, and spending. Mechanics work in tandem with the game's content to shape the overall dynamics, influencing the challenges and experiences players encounter. By carefully integrating and refining these mechanics alongside other elements, developers can achieve their goals of creating engaging and balanced gameplay experiences.

Evaluating game mechanics is crucial for understanding their impact on player experience and ensuring they contribute positively to game balance and engagement. Traditionally, developers are responsible for creating game mechanics and testing how players interact with them, a process that can be both time-consuming and labour-intensive (Berg Marklund et al. 2019). To understand new mechanics, there must be game objects for them to interact with, demonstrating those mechanics through gameplay. Significant effort is required to ensure that mechanics interact well with one another and achieve a balanced gameplay experience. Therefore, I prioritize creating new game objects , such as levels and enemies Figure 1, to support gameplay that allows for a more comprehensive evaluation of the generated mechanics.

Automating the design and generation of game mechanics has been a consistent area of interest in game development (Cook, Colton, and Gow 2016; Treanor et al. 2012; Summerville et al. 2018; Todd et al. 2024). Currently, automated game mechanic generation researchers often use a static simulator like A* agent or an evaluation function to evaluate game mechanics (Cook, Colton, and Gow 2016; Guzdial and Riedl 2021). My proposal is to automate the generation of game mechanics using a novel generator and human-aligned evaluation methodologies. By human-aligned I indicated that the evaluation should not be made just via an A* agent, instead combining it with approaches that share features with human players.

I define Design Impact Accuracy (DIA) as a measure to gauge how well the output of the system aligns with the intended developer experience. A high DIA indicates strong alignment between the design goals and and the output, while a low DIA reveals discrepancies between the designer's intentions and the actual output of the system. For instance, if a level designer creates an area that players can fully navigate using existing mechanics, the level will have a high DIA value if that was the intention of the developer. Similarly, if a new level incorporates new mechanics that enable players to navigate effectively, it will also score high in DIA. Conversely, if a designer's level is challenging to navigate with these new mechanics and negatively affects the player experience, it will yield a low DIA value, if that was not the intention of the developer. For this a user study needs to be conducted in order to obtain more information on how intention works for this case.

Related Work

My work builds on prior research in procedural content generation (PCG) and reinforcement learning (RL) as an evaluator, particularly the use of search-based PCG meth-

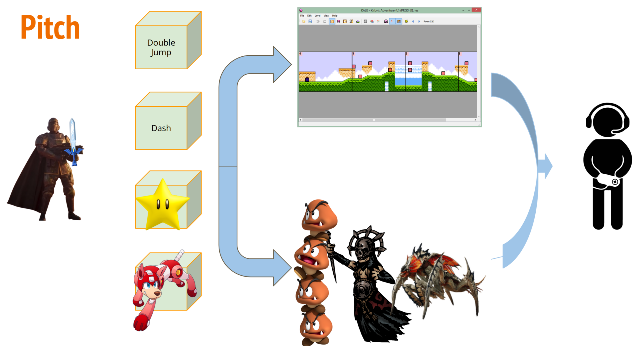

Figure 1: Pitch on how new mechanics need to interact with objects in order to be a good mechanic.

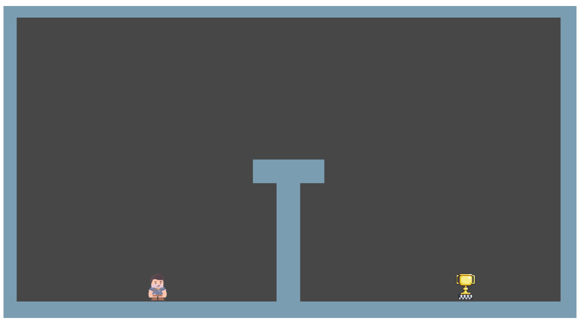

Figure 2: Environment for the generations of mechanics

ods and automated game design (AGD) systems like Mechanic Miner (Cook et al. 2013; Cook, Colton, and Gow 2014, 2016). While traditional methods rely on static agents like A* for evaluating mechanics, I follow work in PCGRL (Khalifa et al. 2020) to use RL agents that better reflect human-like learning and adaptability.

Work to Date

Level Inpainting Research

In my prior level inpaining work (Gonzalez and Guzdial 2023), I explored the reconstruction of existing game levels through "Level Inpainting", a technique inspired by image inpainting (Salem 2021). This approach allows for the restoration or completion of incomplete areas within a game level. It is also highly relevant to the study of embedded mechanics generation, as the reconstruction and adaptation of levels can significantly influence how mechanics are implemented and perceived in different contexts.

I adapted two techniques from image inpainting for level inpainting: the Autoencoder and U-Net architectures. I aimed to maintain the structural coherence of the original game content while effectively restoring missing sections. The U-net slightly outperformed the autoencoder in accurately capturing unique tile locations in the VGLC dataset.

Mechanic Maker 2.0

In my previous work (Gonzalez, Cooper, and Guzdial 2023), I utilized Reinforcement Learning to evaluate generated

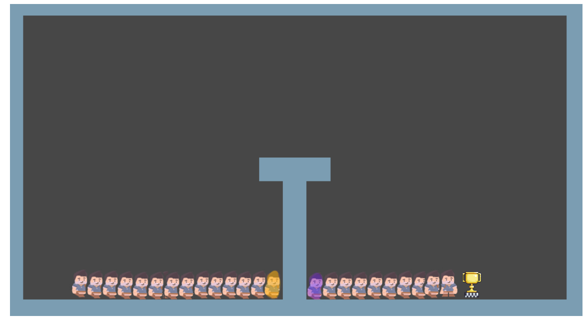

Figure 3: Mechanic that uses speed to go through the wall to reach the goal by glitching unity environment

rules within a search-based approach for Automated Game Design (AGD). The motivation behind this research was to address the limitations of traditional AGD methods, which often rely on static agents or objective functions (such as the A* algorithm) to evaluate game mechanics. These methods lack the ability to account for human learning and adaptation during gameplay.

To investigate this Mechanic Maker 2.0 was created. It is a 2D platformer environment in Unity, replicating Mechanic Miner (Cook et al. 2013). The environment features a player starting on the left, a trophy on the right, and a "T"-shaped obstacle in between, see Figure 2. Mechanic Maker 2.0 uses public movement-related variables, such as Jump Force, Speed, and Position, to facilitate rule generation. Initially, players can only move left, right, and jump, but these actions are insufficient to overcome the obstacle. New rules must be generated to help the agent reach the goal.

The study involved a comparative analysis between RL and A* agents applied to rule evaluation. The results demonstrated that the RL agent facilitated the discovery of a wider variety of rules, potentially more aligned with human intuition. However, there was a trade-off. While RL explored unconventional solutions, it also exhibited a tendency to "hack" the fitness function, sometimes leading to unintended rule modifications as shown in Figure 3. On the other hand, the A* agent was reliable and consistent, providing rules that did not change but lacked the variety seen in the RL-evaluated rules, which I consider more creative.

Enemy Generation

In my current research, I am focusing on generating enemies that can only be defeated using new mechanics, mechanics that in this case are outputs from Mechanic Maker 2.0. To achieve this, I am exploring the use of different approaches from traditional development that will be hand made by developers. The process involves using a neural network to generate a 4x4 matrix which represents an enemy's collision map, where the enemy is represented by a one-hot encoded format with different values corresponding to different kinds of collision components. This is integrated into a reinforcement learning environment that trains the agents to use the new mechanics, or using a A* agent environment that classify which parts are reachable, allowing for an evaluation

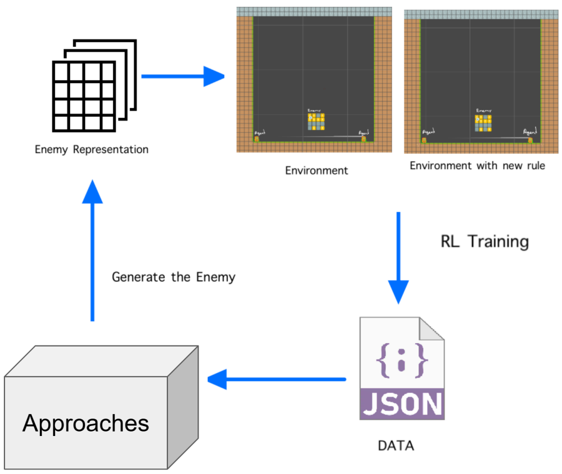

Figure 4: The general loop of enemy generation where the approaches changes for the experiments using NeEF, the RL classifier, or the A* classifier

of how effectively they can defeat the enemies. The goal is to assess the effectiveness of the new mechanic by comparing the agents' performance. The process iteratively refines the enemy, with a goal that the enemies created can only be defeated with the new mechanic see Figure 4 to see a visualization the workflow that I use to generate the enemies.

To generate enemies tailored to specific mechanics, my work explores three different approaches. The first is conceptually inspired by Neural Radiance Fields (NeRFs) (Mildenhall et al. 2021), which use a neural network to represent a 3D scene by predicting colour and density at each point in space based on a set of 2D images. This iterative refinement process forms the conceptual basis for the first approach, which I refer to as the Neural Enemy Field (NeEF). In NeEF, a neural network generates a 4x4 matrix representation of an enemy, where each part is encoded using one-hot values. An RL agent is used to interact with the enemy, collecting collision data that informs how the network refines the enemy layout—specifically to ensure it can only be defeated using the newly introduced mechanic. The second approach uses an A* agent to classify which parts of the enemy are reachable with the new mechanic. This reachability data is then used to directly modify the enemy structure, ensuring vulnerable areas can only be accessed with that mechanic. Finally, a separate RL-based classification approach uses the collision data gathered during training in a similar manner to the second approach's A* agent. to classify patterns in enemy shapes that correspond to successful or unsuccessful interactions.

I am currently applying these methods to EvoGym, a large-scale benchmark for co-optimizing the design and control of soft robots. As this is ongoing work, the methods and outcomes are still evolving, and further refinements are expected as experimentation continues.

Future Work

My future work plan is divided into three main directions. First, I aim to upgrade the mechanic generation process by enhancing the search algorithm to improve both its efficiency and the quality of the results. Second, I will focus on map generation. As previously mentioned, new mechanics need to interact with in-game elements, primarily enemies and level structures. In the full development cycle, the goal is to ensure that the generated levels are designed in such a way that they can only be traversed effectively by using a newly generated mechanic. And finally Design Impact Accuracy (DIA) will serve as a central metric to evaluate how well AI-generated content, specifically levels and enemies, aligns with a game designer's intended use of new mechanics. DIA measures the degree to which generated content supports or reflects the goals and vision behind a selected mechanic, not only in terms of functional playability but also in terms of perceived designer intent. To assess DIA from a human-centred perspective, I propose a user study where participants act as game designers by selecting a predefined mechanic and evaluating content generated for it. Through direct comparison between AI-generated levels and human-authored or sampled alternatives, followed by surveys and level design tasks, I will analyze whether the AI's output complements the mechanic as intended. The results, will provide insight into the system's ability to adapt to human design goals, and establish DIA as a meaningful, measurable bridge between AI generation and human creativity in game development.

References

Berg Marklund, B.; Engström, H.; Hellkvist, M.; and Backlund, P. 2019. What empirically based research tells us about game development. *The Computer Games Journal*, 8: 179–198.

Cook, M.; Colton, S.; and Gow, J. 2014. Automating game design in three dimensions. AISB.

Cook, M.; Colton, S.; and Gow, J. 2016. The angelina videogame design system—part ii. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(3): 254–266.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16*, 284–293. Springer.

Gonzalez, J. J.; Cooper, S.; and Guzdial, M. 2023. Mechanic Maker 2.0: reinforcement learning for evaluating generated rules. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, 266–275.

Gonzalez, J. J.; and Guzdial, M. 2023. Reconstructing existing levels through level inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, 276–283.

Guzdial, M.; and Riedl, M. O. 2021. Conceptual game expansion. *IEEE Transactions on Games*, 14(1): 93–106.

Hunicke, R.; LeBlanc, M.; Zubek, R.; et al. 2004. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, 1722. San Jose, CA.

Khalifa, A.; Bontrager, P.; Earle, S.; and Togelius, J. 2020. Pcgrl: Procedural content generation via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 95–101.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Salem, N. M. 2021. A Survey on Various Image Inpainting Techniques. *Future Engineering Journal*, 2(2).

Summerville, A.; Martens, C.; Samuel, B.; Osborn, J.; Wardrip-Fruin, N.; and Mateas, M. 2018. Gemini: Bidirectional generation and analysis of games via asp. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 123–129.

Todd, G.; Padula, A.; Stephenson, M.; Piette, É.; Soemers, D. J.; and Togelius, J. 2024. GAVEL: Generating Games Via Evolution and Language Models. *arXiv preprint arXiv:2407.09388*.

Treanor, M.; Schweizer, B.; Bogost, I.; and Mateas, M. 2012. The micro-rhetorics of Game-O-Matic. In *Proceedings of the International Conference on the Foundations of Digital Games*, 18–25.