Machine Learning for Playable Room Generation: A Modular Approach with VAE and PCG

Juan Paes[1], **Roberto Junio**[1,2], **Luis Cuevas Rodriguez**[1,2]

[1] Escola Superior de Tecnologia – Universidade do Estado do Amazonas
Av. Darcy Vargas 1200 – 69050-020 – Manaus – AM – Brazil

[2]Ludus – Laboratório de Tecnologia, Inovação e Economia Criativa da (EST/UEA)

{jcdcp.snf22, rjrg.pje24, lrodriguez}@uea.edu.br

Abstract. Introduction: *The growing complexity of roguelike games demands smarter content generation, but uncontrolled PCG can result in unbalanced and frustrating gameplay.* ***Objective:*** *This paper proposes a modular system that integrates PCG with Machine Learning (ML) to generate optimized, playable rooms for roguelike games.* ***Methodology:*** *The approach combines Cellular Automata for initial generation and a Variational Autoencoder (VAE) to refine layouts based on user-defined criteria.* ***Results:*** *The system achieved (85%) player approval for VAE-generated rooms, reduced generation time by (58%), and improved diversity, consistency, and gameplay accessibility.*
Keywords *Procedural Content Generation, Machine Learning, Roguelike, Variational Autoencoder, Map Generation.*

1. Introduction

The video game industry moves billions of dollars and continues to attract more developers and investments, especially in the Brazilian scenario, which has been gaining global visibility through both large brand support and independent initiatives [Mateus Omena 2025]. With the increasing complexity of games, production costs have also grown significantly, making it essential to adopt methods that optimize time and resources—especially for independent developers, who often lack external financial support [Se Yeon KIM e Kim, Seokkyoo 2024].

In this context, *Procedural Content Generation* (PCG) has established itself as an effective alternative, enabling the automated creation of rooms, characters, and objects through rules and algorithms. It reduces the need for manual design and promotes greater variety and replayability in games [Shaker et al. 2016, Silva et al. 2024].

One of the genres that most benefits from PCG is the roguelike, characterized by unique runs through randomly generated rooms. However, such randomness can cause severe balance issues. For players of the genre, rooms with poorly placed enemies or inaccessible items can result in frustrating or even unplayable experiences. Even small generation flaws directly compromise the challenge and enjoyment—core elements of these games [Rogers 2014, Craddock 2021].

This imbalance leads many designers to opt for fixed rooms, giving up the variety that procedural generation offers, since uncontrolled PCG can harm gameplay and alienate players, especially in games that require well-calibrated challenges [Silva et al. 2024].

Given this scenario, this work proposes the development of a tool aimed at roguelike game developers, capable of automatically generating rooms that strike a balance between randomness and playability. The proposal combines procedural generation techniques with *Machine Learning* (ML), allowing the creation of customized rooms based on user-provided parameters. The solution aims to support independent developers by offering a practical and flexible way to generate varied content without compromising game design, merging variety with control.

2. Related Work

Video games have become a significant research domain in recent years, particularly regarding the integration of PCG and ML. Combining these two technologies in game development has shown great potential to produce more balanced rooms without losing diversity [Shaker et al. 2016].

One strategy for procedural room generation integrated with ML, highlighted by Werneck and Clua, used the Unity ML-Agents toolkit to train agents that learn essential patterns in room design and replicate them when generating new environments. This approach strongly emphasizes gameplay and balance, leveraging Unity's 3D environment and reinforcement learning techniques to improve the quality of generated rooms [Werneck e Clua 2020].

The method presented by Minini et al. combines traditional constructive approaches with the Wave Function Collapse (WFC) algorithm, generating visually coherent 2D game rooms. The focus is on preserving local rules and patterns, which leads to aesthetically consistent rooms but offers less flexibility from the standpoint of emergent gameplay [Minini e Assuncao 2020].

Other authors, such as Gisslén et al., have proposed adversarial reinforcement learning approaches (ARLPCG), where a generator agent creates rooms and a solver agent attempts to beat them. This method enables adaptive challenges but relies on complex and computationally intensive training cycles, making it more suitable for games that require dynamic difficulty adjustment [Gisslén et al. 2021].

Unlike these approaches, this work proposes a modular system that combines the classical technique of PCG based on Cellular Automata with a Machine Learning model based on a Variational Autoencoder (VAE). This combination aims to strike a balance between randomness and control in the design of roguelike game rooms.

Before arriving at the proposed solution, we explored other ML techniques, such as Decision Trees, which proved inadequate for preserving the spatial structure of rooms. These attempts reinforced the need for a more robust model, leading to the adoption of the VAE as the core component of the system.

The architecture follows a stepwise pipeline that supports both initial random generation and refinement based on learned patterns. Although the technical details of the implementation are presented in the next section, it is worth noting that the focus of this proposal is to provide a lightweight, accessible, and parameterizable solution, particularly suitable for independent developers.

This approach stands out for avoiding complex tools such as 3D agents or computationally heavy algorithms (e.g., ARLPCG, Wave Function Collapse), prioritizing efficient and reproducible techniques that can run on modest computational resources.

Together with the approaches discussed above, our proposal highlights the potential of combining PCG and ML to address challenges in game content creation, such as the trade-off between structure and randomness, the provision of appropriate challenges, and the enhancement of player experience. In the specific context of roguelikes, this work seeks to offer a practical and efficient solution to one of the genre's central issues: the inherent imbalance of traditional procedural generation.

3. Development

This work proposes the development of a system for the automated and balanced generation of rooms for games in the roguelike genre. The architecture combines classical PCG algorithms with ML techniques, aiming to ensure variability without compromising playability. The system was implemented in Python and is structured into three main modules: (i) initial room generation via cellular automata, (ii) filtering and encoding based on user-defined criteria, and (iii) generation of new optimized rooms through learning models. The modular architecture ensures separation of concerns, scalability, and ease of maintenance, as shown in Figure 1:

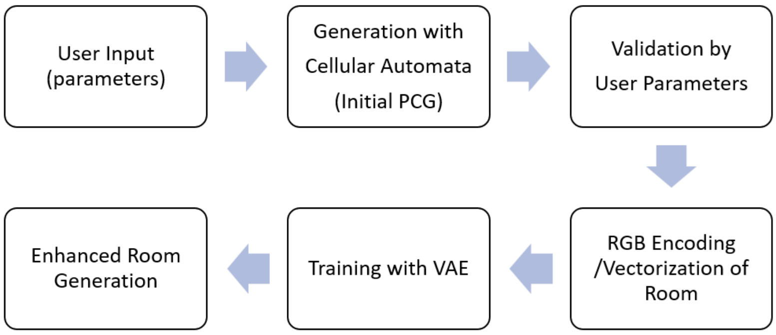

Figure 1. Overview of the system's operational flow
Source: Authors' own work

3.1. Initial Generation with Cellular Automata

The procedural generation of rooms is carried out using an algorithm based on Cellular Automata, chosen for its efficiency in creating organically connected environments with pseudo-natural appearance. This technique is widely used in roguelike games for generating caves and labyrinthine structures and proved suitable for representing playable rooms composed of walls, floor, items, enemies, and the player, as follows Figure 2:

Each generated room consists of a two-dimensional matrix where cells are iteratively updated based on neighborhood rules. The input parameters allow defining the density of walls and the spatial configuration of elements, directly contributing to the game's flexibility and replayability.

3.2. Room Validation and Encoding

After the initial generation, each room is evaluated against a set of user-defined criteria, which serve both to ensure gameplay viability and to guide the training of the VAE. These

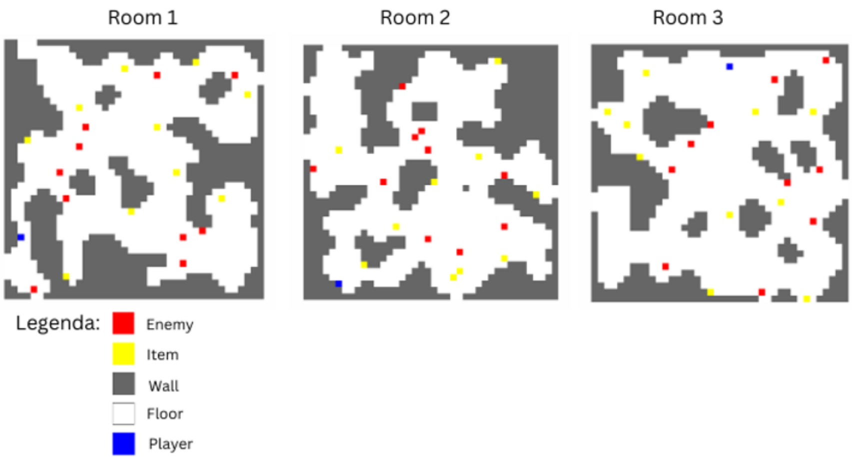

Figure 2. Visual examples of rooms generated with Cellular Automata
Source: Authors' own work

criteria are configurable and were designed to be easily adjustable through the system interface, enabling developers to tailor room generation according to their design goals. The specific parameters used for validation and training in this implementation were:

- Percentage of walls (obstacle density);
- Number of enemies and items;
- Minimum distance between the main character and enemies;
- Minimum distance between items and enemies.

Rooms that do not meet the requirements are discarded. Those that satisfy all conditions are then transformed into a vectorized matrix representation, using both numeric and RGB encoding for each game entity. This encoding is essential to enable the training of machine learning models.

3.3. Learning with Variational Autoencoder (VAE)

Given the limitations encountered with simpler models, we opted for a VAE, a neural network architecture capable of learning compact and expressive latent representations of input data. The VAE was trained using RGB representations of the previously filtered valid rooms.

The network is composed of a convolutional encoder that reduces the dimensionality of the rooms into a latent space, and a decoder that reconstructs new rooms from samples in that space. In this architecture, the convolutional filters play a fundamental role in learning spatial features from the rooms. These filters extract and abstract patterns such as wall structures, open paths, and object placement. By progressively encoding these features, the model is able to compress the spatial configuration into a latent representation that captures essential design traits. This enables the decoder to generate new room layouts that maintain spatial coherence, structural consistency, and gameplay viability.

During inference, the system can generate novel rooms by sampling different latent vectors while maintaining coherence with learned patterns and adhering to established balance criteria, as shown in Figure 3:

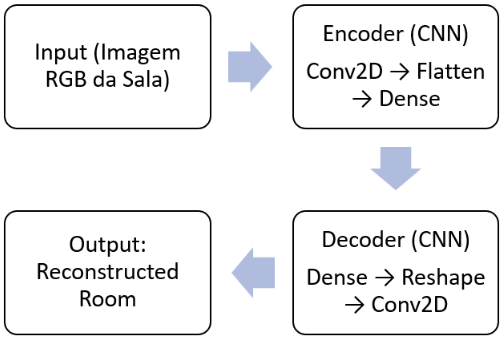

Figure 3. Internal workflow of the VAE model
Source: Authors' own work

3.4. Optimized Generation and Adjustments

The new rooms generated by the VAE are again verified against the user-defined parameters, ensuring that the randomness remains controlled and functional. The system also allows manual adjustments and room revalidation by the developer (user of the tool), offering additional flexibility to the creative process. Furthermore, the tool outputs the room matrix in a format ready to be loaded into the user's game engine of choice.

3.5. Interface and Visualization

Finally, the system provides a simple interface for parameter input and automatic room generation. Rooms can be visualized both as textual matrix representations and as RGB images, with distinct colors assigned to each entity. This facilitates rapid prototyping and visual evaluation of the generated content.

4. Evaluation

The goal of this evaluation was to assess the effectiveness of the proposed system in generating balanced rooms for roguelike games, while respecting user-defined criteria and preserving the diversity typically offered by procedural methods. The analysis focused on acceptance rate, compliance with defined parameters, and visual coherence of the generated environments.

4.1. Methodology

The evaluation followed commonly adopted guidelines in works that combine PCG and machine learning in games, which recommend objective metrics for rule conformity and qualitative analysis of generated structures [Shaker et al. 2016].

A total of 500 initial rooms were generated using the cellular automata algorithm. After filtering based on predefined criteria — such as number of enemies, wall density, and minimum distances between entities — 152 rooms (30.4%) were classified as valid and used to train the Variational Autoencoder (VAE) model.

After training, the VAE model was used to generate new rooms via latent space sampling. These new instances were once again validated using the same criteria to ensure consistency. Additionally, qualitative comparisons were made between rooms generated by the VAE and those generated by Cellular Automata, focusing on visual aspects such as connectivity, spatial distribution, and visual noise, as discussed in procedural map analysis literature [Rogers 2014, Craddock 2021].

This dual approach aimed to ensure that the generated rooms were not only random, but also playable and meaningful, respecting fundamental design principles of roguelike games such as fair progression, challenge diversity, and spatial coherence.

4.2. Results

The VAE demonstrated strong generalization capabilities, with over 80% of the generated rooms accepted according to the defined criteria. RGB-based visualizations revealed coherent layouts with good internal connectivity and functional distribution of elements.

To complement the quantitative analysis, a survey was conducted with 20 experienced players, each with an average of over 12 hours of weekly gameplay exclusively in the roguelike genre, ensuring familiarity with game mechanics and typical map patterns. Participants compared pairs of rooms—one generated using cellular automata (procedural baseline) and one generated using the VAE model. The goal was to collect subjective feedback on perceived quality and playability. The results were as follows, Figure 4:

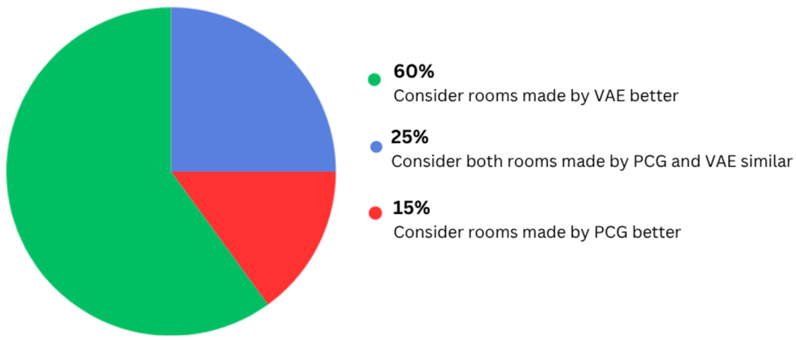

Figure 4. Player opinions on room quality
Source: Authors' own work

- 12 participants (60%) considered the VAE-generated rooms as improvements over the procedural ones;
- 5 participants (25%) found both approaches similar, with no significant advantage;
- 3 participants (15%) preferred the rooms generated via traditional procedural techniques.

These results reinforce the effectiveness of the proposed approach, showing that 85% of the evaluators perceived the VAE-generated rooms as superior or equivalent.

This qualitative perception aligns with the technical validation results and highlights the positive impact of the VAE approach on room structure from the player's point of view. Thus, the proposed system combines objective computational validation with subjective usability feedback, strengthening its practical potential in roguelike game design.

4.3. Complementary Quantitative Analysis

In addition to evaluating the acceptance rate of the generated rooms, complementary metrics were computed to enrich the system assessment and offer a broader perspective on structural and computational behavior. These analyses aimed not only to verify whether the rooms met the predefined criteria, but also to understand their internal organization and performance characteristics.

In terms of generation time, the traditional procedural method based on Cellular Automata required an average of 0.12 seconds per room, while the VAE model, once trained, reduced this to just 0.05 seconds per room. The total training time for the VAE was approximately eight minutes using an NVIDIA RTX 3060 GPU. With respect to wall density, rooms generated procedurally exhibited an average of (44.8%) ± (3.1%), whereas those produced by the VAE showed a slightly lower mean of (43.6%) ± (2.4%), indicating greater structural consistency. Regarding spatial distribution entropy—measured using Shannon entropy per room—the VAE achieved a slightly higher average (2.95 ± 0.09) compared to the procedural baseline (2.83 ± 0.12), suggesting increased diversity in the arrangement of elements while maintaining coherence. For connectivity, defined as the ratio between the largest connected component and the total number of accessible cells, procedural generation reached 0.91, while the VAE maintained a comparable value of 0.89, reflecting robust internal structure in both approaches. Lastly, when evaluating spatial accessibility through the number of viable paths—calculated using the A* algorithm between the player and all items or enemies—The VAE outperformed the baseline procedural method (traditional Cellular Automata) with an average of 92 viable paths, compared to 86.

These quantitative insights reinforce the effectiveness of the VAE-based approach in ensuring not only valid room generation, but also improvements in structural diversity, internal coherence, and gameplay accessibility.

4.4. Systematic Comparison between the Methods

The main differences observed between the two generation methods can be observed on Table 1:

Table 1. Comparison Between Methods

Metric	Traditional Procedural	VAE
Average generation time (s)	0.12	0.05
Average wall density (%)	44.8 ± 3.1	43.6 ± 2.4
Average spatial entropy	2.83 ± 0.12	2.95 ± 0.09
Average connectivity	0.91	0.89
Accessibility of paths (%)	86	92

These data reinforce that the VAE-based model not only generates valid rooms, but also produces more connected, accessible, and diverse environments, while maintaining fidelity to the patterns learned during training. Furthermore, the gain in generation time can be significant in games that require real-time generation or multiple executions.

4.5. Limitations

Although the VAE model has proven effective in generating coherent and playable rooms, some important limitations should be considered. The quality of the generated rooms is directly tied to the quantity and diversity of the valid examples used during training. Since only a fraction of the 500 initially generated rooms met the validity criteria, the model was trained on a relatively limited dataset, which may impact its ability to generalize.

Additionally, there is a risk of overfitting, especially in low-data scenarios. While KL divergence — a metric that measures how much the learned latent distribution deviates from a target distribution — was used as a regularization technique, other strategies such as early stopping, dropout, or data augmentation were not implemented, leaving the model potentially vulnerable to memorizing specific patterns. Another limitation is the simplicity of the room elements used in training—restricted to walls, floors, players, enemies, and items—without incorporating more complex features such as doors, keys, puzzles, or enemies with conditional behaviors.

This constraint limits the system's applicability to games with more intricate mechanics. Finally, even after training, generating valid rooms still relies on a trial-and-error process until all positional and distance criteria are satisfied. While this process is faster than traditional procedural generation, it remains an area that could benefit from further optimization.

5. Discussion

The results obtained validate the proposal of combining classical PCG techniques with machine learning to address the recurring issue of structural imbalance in roguelike games. The use of cellular automata for initial generation proved advantageous due to their simplicity and ability to produce organic forms with low computational cost.

During the system's development, an initial attempt was made using a Decision Tree model, given its status as a classical supervised learning technique with low computational cost and high interpretability. However, the results showed serious limitations in preserving the spatial structure of rooms, leading to disorganized and unplayable maps. These methodological shortcomings highlighted the need to adopt more robust models capable of handling spatial patterns, which ultimately motivated the selection of the VAE as the main learning architecture. Replacing the Decision Tree with the VAE was a critical turning point in the system's success. Although decision trees offer interpretability, they fall short in dealing with spatial and local dependency problems inherent in room layouts. In contrast, convolutional neural network-based techniques, such as VAEs, are capable of capturing complex spatial patterns, allowing for greater control over the characteristics of generated rooms.

By enabling the generation of rooms adjusted to user-defined parameters, the system becomes a versatile tool for developers—particularly independent ones—who often lack resources to internally develop balanced PCG systems. The proposed approach also stands out for its computational lightness and independence from specific game engines such as Unity or Unreal, enhancing its accessibility and integration into diverse development workflows. Despite these positive points, it is important to recognize that the VAE model has relevant limitations. The system's performance is sensitive to the quantity and quality of valid rooms used in training. A reduced number of examples may limit its ability to generalize relevant patterns, compromising the diversity of new rooms generated. Strategies such as data augmentation, database augmentation, and model variations (e.g., Beta-VAE) could be investigated to mitigate this problem.

Furthermore, the current system is limited to the representation of rooms with a static structure and simple elements. Its application in games with more complex mechanics — such as keys, doors, puzzles, traps, and conditional events — would require significant adaptations. Such elements involve logical and temporal relationships that can hardly be learned from RGB matrix representations alone. Hybrid approaches that integrate symbolic or graph-based representations, or even contextual attention techniques in neural networks, could be explored to expand the scope of the system.

Another point of concern is the possibility of overfitting, especially considering the limited number of valid samples. Despite the use of KL divergence for regularization, the VAE may end up memorizing specific patterns instead of learning robust latent representations. Complementary techniques such as early stopping, dropout, cross-validation, and analysis of reconstruction variance could be used in future versions of the system to diagnose and prevent this effect.

These limitations provide room for future improvements and also demonstrate the need to explore hybrid approaches that combine deep learning with symbolic representation, especially in games with complex interaction rules. The limitation in representing mechanics such as keys and doors, which involve cause-and-effect relationships, suggests that the exclusive use of matrix representations may not be sufficient.

Finally, the systematic comparison between methods revealed that VAE not only outperforms the traditional procedure in diversity and connectivity, but also reduces the average generation time per room. These gains position the approach as a viable alternative for practical use in rapid prototyping and adaptive content development in digital games.

6. Conclusion

This work presented the development and evaluation of a system capable of procedurally generating balanced rooms for roguelike games by integrating classical algorithms with modern machine learning techniques.

The combination of cellular automata and a Variational Autoencoder proved effective in ensuring both variability and control in the generation process. The proposed approach addresses a recurring problem in roguelike games: unbalanced randomness, which can significantly impact player experience. The system stands out for being modular, lightweight, customizable, and accessible to independent developers.

As future work, we propose integrating the system with game engines to enable direct application within development pipelines. Additionally, we aim to extend the system to support the interconnection of multiple rooms, allowing the generation of larger structures such as complete dungeons. Another direction includes diversifying the set of in-room elements by incorporating conditional entities such as doors, keys, and interactive objects—features that are currently absent but critical for representing more complex game mechanics. Moreover, we recognize the limitations posed by the small player sample used in the evaluation of room quality. Future studies should seek broader participation, potentially by publishing the system in game development forums to attract a more diverse and substantial player base for feedback.

References

Craddock, D. L. (2021). *Dungeon Hacks: How NetHack, Angband, and Other Rougelikes Changed the Course of Video Games*. CRC Press.

Gisslén, L., Eakins, A., Gordillo, C., Bergdahl, J., e Tollmar, K. (2021). Adversarial reinforcement learning for procedural content generation. In *2021 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

Mateus Omena (2025). Indústria de games no Brasil 'vira o jogo' no cenário global — e com apoio das grandes marcas.

Minini, P. e Assuncao, J. (2020). Combining constructive procedural dungeon generation methods with wavefunctioncollapse in top-down 2D games. *Proceedings of SBGames*, pages 27–29.

Rogers, S. (2014). *Level Up! The guide to great video game design*. John Wiley & Sons.

Se Yeon KIM e Kim, Seokkyoo (2024). A Research on Machine Learning Agent in Rogue-like game. *Journal of The Korean Society for Computer Game*, 37(1):33–39.

Shaker, N., Togelius, J., e Nelson, M. J. (2016). *Procedural Content Generation in Games*. Computational Synthesis and Creative Systems. Springer International Publishing, Cham.

Silva, B. C. D., Rodrigues Maia, J. G., e Viana De Carvalho, W. (2024). Procedural content generation in pervasive games: state of affairs, mistakes, and successes. *International Journal of Pervasive Computing and Communications*, 20(3):345–364.

Werneck, M. e Clua, E. W. (2020). Generating procedural dungeons using machine learning methods. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 90–96. IEEE.