DOI 10.36074/logos-21.06.2024.031

MASKING USAGE OF PROCEDURAL GENERATION IN VIDEO GAMES

Tovstochub Ihor[1]
Supervisor: Zinchenko Olha[2]

1. 3rd year student of AKI-123 group
State University of Information and Communication Technologies, UKRAINE

2. Doctor of Technical Sciences, Head of the Department of Artificial Intelligence
State University of Information and Communication Technologies, UKRAINE

In the video game development, procedural content generation (PCG) is a powerful technique for creating vast, dynamic game worlds. By using algorithms to generate game content on the fly, developers can create experiences that offer players a sense of discovery and novelty with each playthrough. Masking the usage of PCG is crucial for maintaining a cohesive and immersive gaming experience for players. If procedurally generated content is easily identifiable or stands out as distinct from handcrafted elements, it can disrupt the sense of immersion and suspension of disbelief that is essential for keeping gamers engaged. When PCG is effectively masked, players are unable to distinguish between procedurally generated and manually designed content, allowing them to remain fully immersed in the game world without breaking the illusion, at the same time allowing to decrease repetitive workload for the development team.

The primary task when masking procedurally generated content is to create an illusion of handcrafted content, where gamers are unable to detect the procedurally generated elements from those designed by human artists and level designers. Achieving this illusion requires a delicate balance between complexity of the algorithm and artistic direction, ensuring that the generated content follows the game's overall aesthetic and thematic elements, thereby maintaining the player's immersion in the game world.

In many games, developers employ a hybrid approach, blending handcrafted content with procedurally generated elements. This involves creating a framework of manually designed structures, environments, or narrative elements, which are then populated and enhanced by PCG algorithms. By anchoring the generated content within a carefully crafted foundation, developers can maintain a consistent artistic direction while introducing dynamic and unpredictable elements, providing gamers with a sense of exploration and discovery within a cohesive game world.

One prominent example of effective PCG masking can be found in the Diablo series. While the game worlds and environments are largely procedurally generated, the developers at Blizzard Entertainment have employed various techniques to seamlessly integrate these generated elements with handcrafted content. For instance, major story-driven areas and key locations are carefully designed by level artists, providing a grounded foundation for the game's narrative. These areas then transition into procedurally generated zones, such as randomized dungeons and caves, through the use of blending algorithms and transitional area designs. Additionally, the games utilize predefined art styles, texture sets, and architectural elements to ensure that the generated content adheres to the overall aesthetic and thematic direction of the Diablo universe, making it indistinguishable from manually crafted environments.

Another example is the No Man's Sky universe, which embraces a fully procedural approach to world generation. While the entire game universe is procedurally generated, the developers, Hello Games, have implemented various masking techniques to maintain coherence and plausibility. This includes the use of mathematical algorithms and noise functions to generate realistic planetary terrains, flora, and fauna, while adhering to predefined rules that ensure a consistent and believable science-fiction aesthetic. Furthermore, the game incorporates procedural narratives and quest systems, contextualizing the generated content within the game's lore and providing plausible explanations for the existence of various celestial bodies, alien structures, and lifeforms encountered by players. This seamless integration of procedural generation with narrative elements helps to mask the underlying algorithms, creating a vast and immersive universe for players to explore.

For any project that utilizes procedural generation, PCG usage should begin with establishing artistic constraints. To maintain a cohesive visual style, developers must set clear artistic constraints that guide the content generation algorithms. These constraints can include predefined color palettes, texture sets, architectural styles, and environmental themes, among others. By carefully curating these artistic assets and integrating them into the generation process, developers can ensure that the resulting content aligns with the game's intended aesthetic, making it indistinguishable from manually designed elements and preserving the player's sense of immersion [1].

Narrative integration and contextualization would be the next step in utilizing generated content in games. By contextualizing the generated content within the game's narrative and providing possible explanations for its existence, developers can create a sense of cohesion and believability for gamers. For example, in a fantasy RPG, procedurally generated dungeons could be attributed to ancient civilizations or magical phenomena, seamlessly integrating them into the game's lore and enhancing the player's perception of a detailed and immersive world.

In order to make implementation feel seamless, the developers should keep in mind the following things:

1. Environmental consistency - for level design/art purposes, algorithms must account for factors such as terrain generation, object placement, navigation mesh creation, and collision detection to maintain a plausible and coherent game world. Techniques like fractal terrain generation, constraint-based modeling, and L-systems can be employed to generate realistic and logically consistent environments that align with gamers' expectations and suspension of disbelief [2].

2. Balance - content generation algorithms must consider gameplay balance and progression, ensuring that the generated content provides a consistent level of challenge as well as not ruin handcrafted experiences for gamers. By incorporating difficulty curves, enemy spawning patterns, and resource distribution algorithms, developers can create dynamic and engaging experiences that feel carefully crafted and balanced, allowing players to immerse themselves in a fair and rewarding gameplay loop [3].

3. Reproducibility and replayability - the ability to reproduce or replicate specific game experiences for many gamers is highly valued. When masking procedurally generated content, developers must ensure that the generated content is reproducible, allowing players to revisit particular areas, challenges, or events with consistent results. This can be achieved through techniques like seeded random number generation, which ensures that the same input consistently generates the same output, preserving the gamer's ability to share and recreate memorable moments [4].

As the field of procedurally generated content continues to evolve, new techniques and technologies are emerging that offer promising ways for masking the usage of procedural generation in video games, further enhancing gamer's experience and immersion.

Machine learning techniques, particularly Generative Adversarial Networks (GANs), have shown significant potential in generating realistic and coherent game content. By training these models on existing game assets and data, developers can generate content that closely mimics the style and quality of handcrafted elements, making it increasingly difficult for gamers to distinguish between the two. This can lead to more immersive gaming experiences, as the line between procedurally generated and handcrafted content becomes increasingly blurred [5].

The concept of "procedural worlds" is gaining traction, where entire game environments are generated procedurally, rather than relying on individual handcrafted elements. These worlds are then further enhanced and populated by PCG algorithms. By embracing a fully procedural approach, developers can potentially eliminate the need for masking, as the entire game world is generated algorithmically, offering gamers a truly unique and ever-changing experience with each playthrough.

Adaptive and evolutionary algorithms are also being explored as a means to improve the coherence and plausibility of procedurally generated content. These algorithms can continuously adjust and refine the generated content based on player feedback, gameplay data, and predefined constraints, ensuring that the generated elements evolve to better align with gamer expectations and the game's overall design. This adaptive approach can lead to more personalized and engaging gaming experiences, as the content dynamically adapts to each player's preferences and playstyle [6].

The effective masking of procedural content generation in video games is a tough challenge that requires a combination of algorithmic optimization, artistic direction, and narrative cohesion.

From an algorithmic perspective, techniques such as constraint-based modeling, fractal generation, and machine learning approaches like generative adversarial networks (GANs) can be employed to generate content that adheres to predefined artistic and structural constraints, ensuring visual and environmental consistency.

Narrative integration and contextualization through techniques like AI-based quest and storyline generation can seamlessly integrate procedurally generated elements into the game's overarching narrative, enhancing plausibility and player immersion.

By synergistically combining these elements, developers can create game worlds that seamlessly blend procedurally generated and handcrafted content, creating a suspension of disbelief and immersive gameplay experience that successfully hides procedural generation methods.

REFERENCES:

[1] Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Cham : Springer International Publishing, 2016. URL: https://doi.org/10.1007/978-3-319-42716-4.
[2] A Survey on Procedural Modelling for Virtual Worlds / R. M. Smelik et al. Computer Graphics Forum. 2014. Vol. 33, no. 6. P. 31–50. URL: https://doi.org/10.1111/cgf.12276.
[3] Search-Based Procedural Content Generation: A Taxonomy and Survey / J. Togelius et al. IEEE Transactions on Computational Intelligence and AI in Games. 2011. Vol. 3, no. 3. P. 172–186. URL: https://doi.org/10.1109/tciaig.2011.2148116.

[4] Procedural Content Generation via Machine Learning (PCGML) / A. Summerville et al. IEEE Transactions on Games. 2018. Vol. 10, no. 3. P. 257–270. URL: https://doi.org/10.1109/tg.2018.2846639.

[5] Procedural content generation for games / M. Hendrikx et al. ACM Transactions on Multimedia Computing, Communications, and Applications. 2013. Vol. 9, no. 1. P. 1–22. URL: https://doi.org/10.1145/2422956.2422957.

[6] Super mario evolution / J. Togelius et al. 2009 IEEE Symposium on Computational Intelligence and Games (CIG), Milano, Italy, 7–10 September 2009. 2009. URL: https://doi.org/10.1109/cig.2009.5286481.