

# **An Analysis of the Variability, Reliability, and Control Provided by Level Generation Techniques for 2D Platform Games**

**COMP110 - Computer Architecture Essay**

1507290

December 6, 2015

There are a number of diverse approaches to level generation for 2D platform games, each placing emphasis on a different aspect of level generation. Chunk-based approaches tend to favour control and variation, whilst rhythm-based approaches favour playability and enjoyability. There are also evolutionary approaches that use genetic algorithms, which when combined with constraint satisfaction provide a high level of control and reliability. For an independent games developer with limited resources, generating creative, high quality levels with minimal human input is essential. This essay will present an analysis of the suitability of each approach.

## **1 Introduction**

Procedural content generation for 2D platform games is a relatively new field [1]. Many approaches to this problem have been developed, each with

different characteristics [2]. Chunk-based approaches, such as Occupancy-regulated Extension (ORE) [3], prioritise variability. Rhythm-based approaches, such as Launchpad [4], place an emphasis on playability and enjoyment. Evolutionary approaches, such as Sorenson et al.’s [5], provide a high level of control and reliability. For an independent games company who desire creative control yet do not possess the resources to hand design each level, the most desirable level generator would be one that requires minimal human input yet offers a high level of control. It should also generate interesting and playable levels consistently.

## 2 Comparison

### 2.1 Overview

Smith et al. [4] base their level generation tool, Launchpad, on the concept of rhythm. They define rhythm as being an ideal sequence and timing of button presses; for example, a series of jumps. They use a two-tiered, grammar-based approach, as illustrated in Figure 1, to ensure that rhythm is always present.

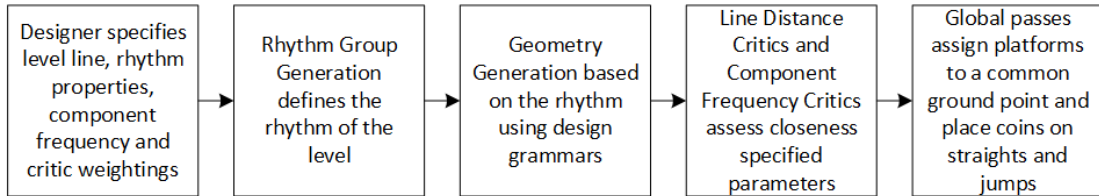


Figure 1: Launchpad’s Level Generation Process

Sorenson et al. [5] use a genetic algorithm combined with constraint satisfaction to generate levels, as illustrated in Figure 2. Their fitness function is inspired by rhythm, but unlike Launchpad, rhythm is defined as being alternating

periods of high and low difficulty. Levels with a high fitness value are able to pass on their genetic information, resulting in levels gradually being improved. Constraint satisfaction is also used to ensure playability and provide additional control.

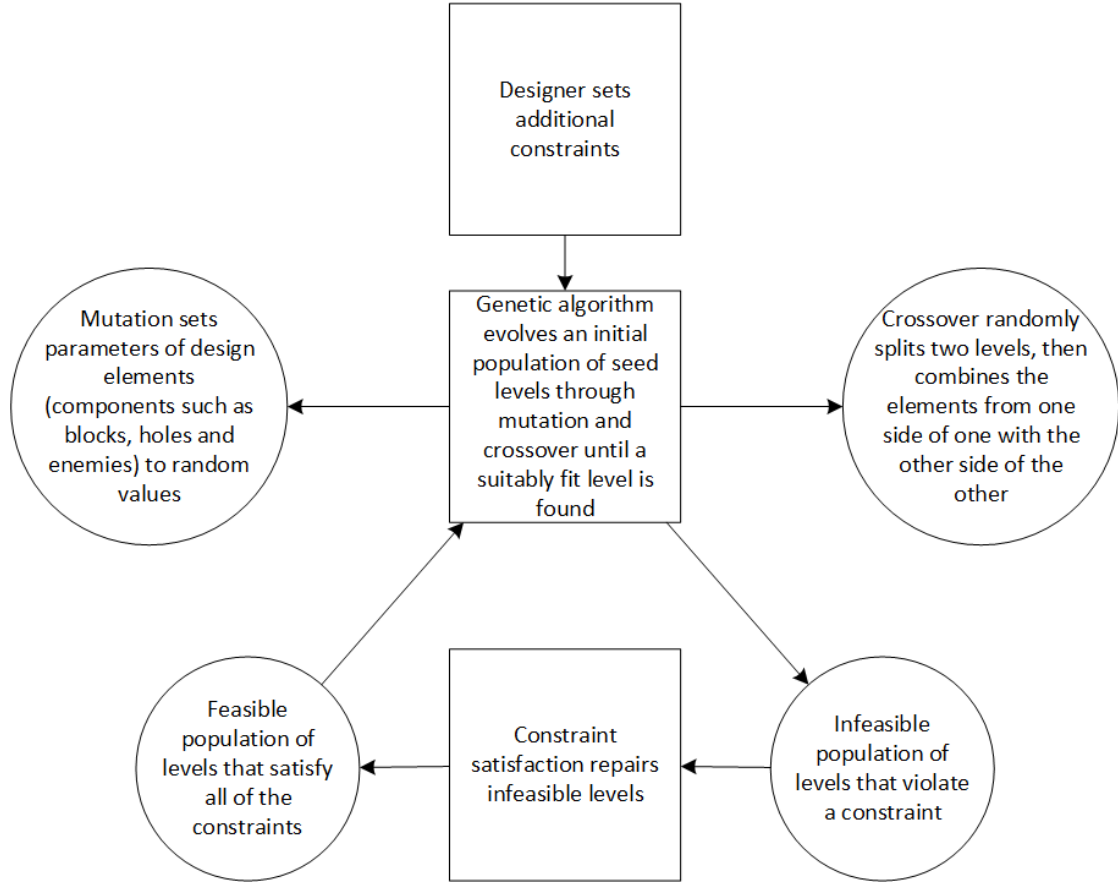


Figure 2: Sorenson et al.'s Level Generation Process

Mawhorter and Mateas [3] take a different approach to level generation. ORE uses pre-authored chunks of level and stitches them together based on anchor points that represent places the player can occupy, as illustrated in Figure 3. Unlike the other two algorithms, ORE has no strict measures in place to ensure playability, as it was designed to prioritise variability.

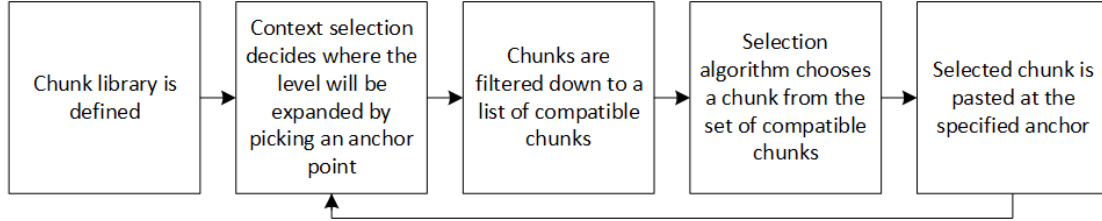


Figure 3: Level Generation Process using Occupancy Regulated Extension

## 2.2 Human Input and Control

Launchpad [4] requires minimal effort whilst offering high levels of control. It allows the rhythm, the path of the level, and the frequency of each component to be specified. A large number of levels are generated before a pool of the best matches is presented. In practice, this means that Launchpad may be better for aiding the design of levels, rather than generating them in-game. Even if the closest matching level could automatically be chosen, generating a large pool of levels would extend the time taken.

Although other approaches using genetic algorithms have been developed [6], Sorenson et al.’s [5] hybrid offers more direct control. In addition to allowing component numbers to be restricted, the difficulty of the generated levels can be adjusted by changing one parameter. Furthermore, levels can be generated around pre-authored level sections. This means that level profiles for each part of the game could be specified, allowing control over progression whilst maintaining variation. This could even allow levels to adapt to the player, by reducing the difficulty parameter if the player repeatedly loses<sup>1</sup>.

Of the three methods, ORE [3] requires the most human effort. Level chunks must be hand-authored, meaning that the output is greatly influenced by the quality of the chunks. Additionally, in order to achieve the desired

---

<sup>1</sup>Although genetic algorithms are usually too slow to generate levels in-game, it is possible to find suitable levels quickly if the seed levels are well-designed [7].

result, an understanding of how the algorithm works is required.

## **2.3 Flexibility and Variability**

Launchpad’s [4] two-tiered approach allows levels with the same rhythm to have many geometric representations. The levels tend to be very linear and have little pattern variation [2], however, which could result in them lacking replayability. Additionally, the emphasis on rhythm means that the generated levels are only appropriate for dexterity-based platform games.

This is also the case with Sorenson et al.’s algorithm [5], as the fitness function is modelled on challenge and difficulty with respect to skill. On the other hand, there is the scope to adjust the fitness function. If required, a fitness function based on a different model could be developed, allowing levels suitable for other types of platform games to be generated. Furthermore, their definition of rhythm is looser, allowing levels to be less linear and repetitive.

ORE [3] could potentially generate levels appropriate for other types, such as exploration-based platform games with multiple paths. Chunks can be designed with any desired result in mind, whilst the algorithm does not need significant adjustment since the playability of levels is not assessed. Additionally, significant pattern variation and complex structures within generated levels can be observed [2], which could have a positive impact on the replayability of levels generated by ORE.

## **2.4 Playability and Design**

Launchpad [4] and Sorenson et al.’s [5] approach have measures in place to ensure playability. Launchpad has a physics engine that is aware of the movement speed and jump height of the player, which is used when

generating the level geometry to ensure that the resulting level is possible to complete. Sorenson et al. specify the playability requirements in the form of constraints, meaning that unplayable levels are repaired before being returned to the population of acceptable levels.

ORE's [3] focus on variability means that it has no measures in place to guarantee playability. This makes ORE unsuitable for generating levels in-game, as the player may be presented with an unbeatable level. An additional algorithm could be used to determine whether the level is playable before presenting it to the player, but that would use additional resources and extend the time taken.

Smith et al. and Sorenson et al. also ensure that some form of theoretical 'fun' is present in generated levels. Both of their definitions of fun are inspired by the concept of 'flow', as described by Csikszentmihalyi [8, 9]. The fitness function used in Sorenson et al.'s algorithm is based on their comprehensive model of fun [10], ensuring that the population evolves towards 'fun' levels. Smith et al. apply their definition of fun in the algorithm that generates the rhythm groups.

ORE is not based on any model of fun. This means that quality of the resulting level relies both on how well the chunks have been designed and the implementation of the chunk selection algorithm. With chunks designed by the authors of the algorithm, ORE came fourth in the *2010 Mario AI Championship* [7], whilst Sorenson et al. managed to place third, despite their general approach.

### 3 Conclusion

Whilst ORE provides great control over the design of a generated level, Sorenson et al.'s approach and Launchpad provide greater flexibility and

reliability with little effort. Sorenson et al.’s approach perhaps provides greater control than Launchpad. Parameters can easily be altered and level sections can be designed by hand if desired. The concept of ‘fun’ has been comprehensively modelled, ensuring that generated levels are enjoyable. This is demonstrated by their reasonable placing in the *2010 Mario AI Championship* [7]. Overall, Sorenson et al.’s approach would allow a small independent developer to reliably produce fun and playable levels with minimal effort, whilst providing a high level of creative control.

## References

- [1] K. Compton and M. Mateas, “Procedural level design for platform games,” in *Proceedings of the Second Artificial Intelligence and Interactive Digital Entertainment International Conference (AIIDE)*, Marina del Rey, CA, June 2006, pp. 109–111.
- [2] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, “A comparative evaluation of procedural level generators in the Mario AI framework,” in *Proceedings of the 9th International Conference on the Foundations of Digital Games*, Ft. Lauderdale, FL, 2014.
- [3] P. Mawhorter and M. Mateas, “Procedural level generation using occupancy-regulated extension,” in *IEEE Symposium on Computational Intelligence and Games (CIG)*, Copenhagen, Denmark, Aug 2010, pp. 351–358.
- [4] G. Smith, J. Whitehead, M. Mateas, M. Treanor, J. March, and M. Cha, “Launchpad: A rhythm-based level generator for 2-D platformers,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 1, pp. 1–16, March 2011.

- [5] N. Sorenson, P. Pasquier, and S. DiPaola, “A generic approach to challenge modeling for the procedural creation of video game levels,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 229–244, Sept 2011.
- [6] F. Mourato, M. P. dos Santos, and F. Birra, “Automatic level generation for platform videogames using genetic algorithms,” in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, Lisbon, Portugal, 2011, pp. 8:1–8:8.
- [7] N. Shaker, J. Togelius, G. Yannakakis, B. Weber, T. Shimizu, T. Hashiyama, N. Sorenson, P. Pasquier, P. Mawhorter, G. Takahashi, G. Smith, and R. Baumgarten, “The 2010 Mario AI championship: Level generation track,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 4, pp. 332–347, Dec 2011.
- [8] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. New York, NY: Harper Perennial, 1991.
- [9] G. Smith, M. Treanor, J. Whitehead, and M. Mateas, “Rhythm-based level generation for 2D platformers,” in *Proceedings of the 4th International Conference on Foundations of Digital Games*, Orlando, FL, 2009, pp. 175–182.
- [10] N. Sorenson and P. Pasquier, “The evolution of fun: Automatic level design through challenge modeling,” in *Proceedings of the First International Conference on Computational Creativity (ICCCX)*, Lisbon, Portugal, 2010, pp. 258–267.