

A Comparison of Procedural Content Generators

COMP110 - Computer Architecture Essay

AR185160

November 30, 2015

This paper will evaluate and compare three articles; A Rhythm-Based Level Generator for 2-D platformers[1], Procedural Content Generation Using Occupancy-Regulated Extension[2] and Procedural Content Generation Using Patterns as Objectives[3]. These articles all have their own techniques to procedurally generate content for a 2-D platformer. This article aims to provide a recommendation for the most appropriate technique if a Indie developer is wanting to make a procedurally generated 2-D platformer.

1 Introduction

The way in which we use Procedural Content Generators (PCG) to develop content for games is a vital part of the video game design process. With the recent release of popular procedurally generated games, such as Minecraft, Elite: Dangerous and The Binding of Isaac, it is important to choose a procedural content generator that suits your needs, as there are a lot of implementations of PCG to choose from. The work is motivated by the

complexity and variety of all the different approaches to Procedural Content Generation, and aims to provide a comparison of three different papers to make a recommendation for an Indie developer that would like to implement a PCG into their 2-D platformer.

There is always going to be advantages and disadvantages of one PCG over another depending on the context. The three papers that this essay will be reviewing are:

A Rhythm-Based Level Generator for 2-D platformers[1], which is designed modularly into groups called “rhythm groups” which are then pieced together to form a complete level. The objective of this PCG is to ensure every level it generates is playable and each level requires very little hand-authoring, all the designer has to do is modify its parameters.

The next paper is:

Procedural Level Generation Using Occupancy-Regulated Extension
[2] prefers the creative side of PCG,

The last paper is:

Procedural Content Generation Using Patterns as Objectives[3]

2 Your section title here

Write the main body of your essay here. Add more sections if appropriate. You may choose to write about each of your three papers in its own section, or you may choose a different structure. Either way, remember that you are being assessed on technical insight and analysis: it is not enough to merely summarise the contents of the three papers. You must demonstrate the ability to make inferences beyond what is written in the papers, and to draw the three papers together into a single coherent narrative.

3 Recommendation

Your essay must make a clear recommendation, in terms of which of the three techniques you have reviewed is the best according to whichever metric or metrics you feel is most appropriate. You must justify your choice, backing it up with empirical evidence. However remember that an academic essay is not a murder mystery: you should already have briefly discussed your recommendation in the introduction and in other parts of the essay. Do not save it for a grand reveal at the end.

4 Conclusion

Write your conclusion here. The conclusion should do more than summarise the essay, making clear the contribution of the work and highlighting key points, limitations, and outstanding questions. It should not introduce any new content or information.

References

- [1] 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*, pp. 175–182, ACM, 2009.
- [2] P. Mawhorter and M. Mateas, “Procedural level generation using occupancy-regulated extension,” in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pp. 351–358, IEEE, 2010.
- [3] S. Dahlskog and J. Togelius, “Procedural content generation using

patterns as objectives,” in *Applications of Evolutionary Computation*, pp. 325–336, Springer, 2014.