Generative spaces: assisting the level designer in creating world maps

Abstract

Introduction

Video games are a medium that combines the artifacts from many different fields. Be it music, visual arts or architecture they all have their influence on the overall feel of a given game.

This wide need for resources explains partly why games are so costly to develop. Procedural content generation–the automatic creation of game content–has emerged as an answer to this problem and is currently widely used. For example, *No Man's Sky* is a space exploration game that has used procedural generation to generate a whole universe of over 18 quintillion planets $1.8 \cdot 10^{19}$, suffice to say that you would need many lifetimes to explore them all.

Tile maps are a staple of many genres of games, they are formed by individual tiles of predefined size and assembled into a grid. Traditionally they are built manually: the level designer chooses one tile for each cell of the grid. Given that the grids can be very large, we can easily imagine that this is a time-consuming process. It puts a human limit into the size of the maps that can be used in a game. Being able to generate these maps procedurally would lift a burden off the shoulders of the level designers and allow the generation of much larger worlds.

In this paper we propose a method to assist in the creation of two-dimensional tile-based world maps.

- We develop a graph based model that allows the level designers to specify the high level structure of the world map. Letting them concentrate on the general outline of the world rather than its appearance.
- There is an inherent repetition in tile maps that can be exploited to our advantage to create them. We show that a Markov chain based model allows not only procedural generation but also the reproduction of the

style of a given game. This differs from common rule based methods that have a hard time appropriately capturing patterns not easily expressed.

Describe problem in detail

Describe the general idea

The macro structure of the world is specified with the help of graphs whose nodes embody a spatial concept such as “city”, “village”, “forest”. A connection between nodes signifies they are spatially linked. This graph will form the basic plan of the world to be created, in other words the world space of the game. The creation of this world space is very flexible; it can simply be built manually by the designer or we can easily imagine more automatic ways of generating it with simple rule based approaches or even with graph grammars.

Describe technical details of the idea

Related work

Game companies generally do not publish their procedural content generation methods so it is hard to evaluate what is common in the industry. In academia, researchers have looked into various approaches for automatic level generation. The game *Super Mario Bros.* in particular has received much attention.

Shaker et al. (2012) uses generative grammars combined with an evolutionary algorithm to create *Mario* levels that adapt themselves to the player's experience. In another insightful paper, Sorenson and Pasquier (2010) also uses an evolutionary algorithm to create a generic framework for level creation. Their approach allows the level designer to enforce multiple constraints on the generated levels making it adaptable to multiple genres of games.

Rather than being evolutionary, our method draws from machine learning techniques. The designers do not have to explicitly articulate rules about their designs but simply need to provide the system with examples of what they want to create.

Papers using machine learning methods to generate content are still a minority, bu the trend seems to be on the rise. Our use of Markov chains is directly inspired by the work of Snodgrass and Ontañón (2013) and Dahlskog, Togelius, and Nelson (2014). They both use Markov chains to create *Mario* levels with very convincing results. The difference with our approach lies in the fact that

Mario levels have a linear nature. The maps we focus on are non-linear; the player can move through both dimensions.

Summerville and Mateas (2015) explore non-linear level generation in the context of dungeons for the game *The Legend of Zelda*. Their use of a graph to represent the game space comes from Dormans (2010). Both papers gave us the idea of using a graph to define the high level structure of the world. Our focus is more on world maps rather than dungeons. Dungeons have a clear compartmentalization of individual rooms while world maps have a more organic unfolding of space.

Conclusion and further work

Can we automate the holistic feel of a level? How does one recreate themes in design works successfully? Can machine learning capture the essence of something?

References

Dahlskog, Steve, Julian Togelius, and Mark J. Nelson. 2014. "Linear levels through n-grams." *Proceedings of the 18th International Academic MindTrek Conference on Media Business, Management, Content & Services - Academic-MindTrek '14*, 200–206. doi:10.1145/2676467.2676506.

Dormans, Joris. 2010. "Adventures in level design: generating missions and spaces for action adventure games." *... Workshop on Procedural Content Generation in Games*, 1–8. doi:10.1145/1814256.1814257.

Shaker, Noor, Miguel Nicolau, Georgios N. Yannakakis, Julian Togelius, and Michael O'Neill. 2012. "Evolving levels for Super Mario Bros using grammatical evolution." *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, no. 1: 304–11. doi:10.1109/CIG.2012.6374170.

Snodgrass, Sam, and Santiago Ontañón. 2013. "Generating Maps Using Markov Chains." *Aiide*, 25–28. http://www.aaai.org/ocs/index.php/AIIDE/AIIDE13/paper/viewPDFInterstitial/7447/7632.

Sorenson, Nathan, and Philippe Pasquier. 2010. "Towards a Generic Framework for Automated Video Game Level Creation Applications of Evolutionary Computation" 6024: 131–40 ST–Towards a Generic Framework for Auto. doi:10.1007/978-3-642-12239-2_14.

Summerville, Adam J, and Michael Mateas. 2015. "The Learning of Zelda : Data-Driven Learning of Level Topology," no. Fdg.