An Analysis of the Role of Procedural Content Generation in Game Design

Gillian Smith
Northeastern University
Playable Innovative Technologies Lab
gillian@ccs.neu.edu

There are many purported reasons for using procedural content generation (PCG) in games and a variety of generation techniques can be used to meet these design goals. One of the earliest purposes was data compression: procedurally generating galaxies in *Elite* (Braben and Bell 1984; Boyes 2006) meant that players could explore a larger universe, but it was vital for the designers to retain complete control over the content players would see. *Rogue* and its sister rogue-like games (Toy et al. 1980; Rogue Basin 2012) use PCG to create experiences that are highly replayable, and use techniques that provide relatively little authorial control over generated content. A recent research focus has been on how to build games that are adaptive to player’s preferences and personalities (Hastings et al. 2009; Shaker et al. 2010; Shaker et al. 2011). Broad claims regarding replayability and adaptability have motivated a great deal of research in PCG.

This talk takes a step back from these claims, critiquing the common motivations for creating PCG systems and attempting to more deeply understand the different ways that generated content is integrated into existing games, how PCG stands to influence player experience, and how the design of the PCG systems themselves influence the design of the game. How can we describe the role of PCG in a particular game’s design? Where is the boundary between procedurally generated content and simple randomness? Are there common patterns for how PCG is typically used in games, and are there opportunities for exploring different avenues for PCG’s incorporation into game design? To answer these questions, we examine games that incorporate PCG using Hunicke et al.’s MDA framework (Hunicke et al. 2004) as a lens for analysis.

This analysis is performed on several games that have used PCG to varying extents and for different purposes, including the “endless runner” games *Canabalt* (Saltsman 2009) and *Robot Unicorn Attack* ([adult swim games] 2010), *Rogue* and other rogue-likes such as *Diablo* (Toy et al. 1980; Blizzard North 1997), *Galactic Arms Race* (Hastings, Guha, and Stanley 2009), the *Civilization* series (Meier 1991; Firaxis Games 2005), and Rohrer’s experimental game *Inside a Star-Filled Sky* (Rohrer 2011). These games are analyzed through asking what the games would look like with the procedurally generated content being replaced by small amounts of human-authored content.

The result is a vocabulary for describing the construction of PCG systems and the role of PCG in a game’s design, including a more nuanced view on replayability. The analysis provides us with a greater understanding for why PCG is used in games, how deeply integrated it is into the core gameplay, and how its incorporation influences player experience.
BIBLIOGRAPHY