Programmer attempting to steal artists job

Tobias Elinder

LTH
tobiaselinder@gmail.com

Automatically generating environments is a very appealing concept. Instead of manually creating environments you can create rules which in turn create the content. This project has used this idea to build whole cities, including interiors, automatically (procedurally).

What are the actual problems that procedural generation solves? Well for one - it saves a lot of manual labor. You also don't have to store things in computer memory. Imagine creating a huge world manually, for example in a game. A huge map would occupy a lot of space on your hard drive, and if you make the world big enough it wouldn't fit at all! Anyone with experience playing computer games should be familiar with the GTA series, which boasts beautiful and intricate environments, but they would be even more interesting if the maps and cities could be made much bigger, this is where procedural generation shines. With procedural generation it's even possible to have infinite worlds, only generating areas when close, like a storyteller reacting to a person's choices by improvising new content on the fly. Procedural generation is also more dynamic than manual generation in that it's possible to tune parameters of the generator for very different results very quickly, without having to rebuild a massive world by hand.

This project has looked at how whole cities can be created with procedural algorithms, from the toilet in a small apartment to whole neighborhoods and structure of road networks. You'd think this was a well explored research area but it isn't. There have been a couple of projects working on procedural city structure and a few working on building interiors before, but few if any has ever tried to do everything simultaneously, which seems like a logical end goal. The vision for this project was a complete city without shallow facades, meaning that a window on a building far away should also lead in to a room when getting close enough.

What makes procedural cities/buildings so hard is the need for coherency balanced against unpredictability. Generating a mountain is easy, no coherency gives chaotic results, and nature is chaotic! Make an office with similar chaotic techniques and you'll end up with very unsafe work environments with cliffs in the middle of the office floor and windows behind bookshelves. Make the rules too strict and you'll end up with a sterile and predictable set of rectangular rooms where everything is always in the same location. To avoid these two extremes you have to find a middle ground where the environment is structured and coherent, but unpredictable. To do this I created a top-down method which generates roads first, then plots from road structures, then buildings from plot structures, the apartment and rooms from building structures, and so on. There is a lot going on beneath the surface, but the general algorithms themselves are actually quite simple. For a peek at the algorithms in action, see this demo.