

Procedural Dungeon Generation

Student: Olivier Legat, **Supervisor:** Andy Day
School of Computing Sciences, August 2012



Aim and Objectives

This study is focused around video games that make full use of Procedural Content Generation (PCG) techniques to create randomised environments. The aims of the project are:

- Generate dungeon-like environments
- Procedurally model the environment into a 3D mesh.
- Provide means of interacting with the environment.



Figure 1. Procedural Modelling. The DunGen software uses procedural modelling and pre-modelled assets to create a virtual 3D environment

Methods Used

In order to generate a perfect maze, algorithms that generate a spanning tree for graph were used. These include Kruskal's and Prim's algorithm.

The 3D environment is modelled using cuboids and planes. A bounding volume hierarchy is used to interact with the environment.

Software

The software includes an independent program called *DunGen.exe* that generates and renders the environment.

A launcher developed in Java was also developed to help parameterise the software.

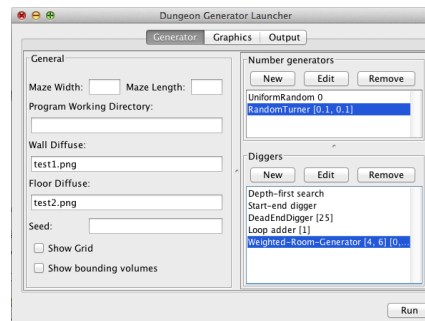


Figure 2. The Launcher. The software used to define the algorithm's behaviour.

Maze post-processing

By applying some post-processing effects such as blocking dead-ends and adding loops to the maze we obtain an entirely different kind of environment

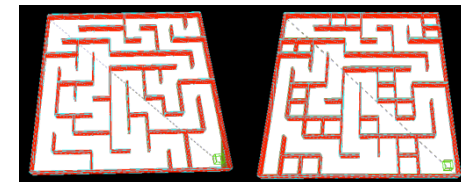


Figure 3. Maze sparseness. This post-processor blocks off dead-ends to reduce the density of mazes [J. Buck 2008]

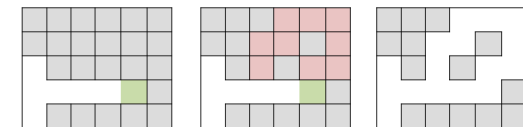


Figure 4. Loop-adder. The post-processor technique used to add loops to the maze [J. Buck 2008]

Conclusion

- The software is able generate a 3D dungeon. It can also render and interact efficiently with it.
- Unfortunately the aesthetic of the environment is tampered by repetition.
- Game-play elements cannot be tackled in a direct manner by a generic generator.