

Cozy Caves

A D&D Dungeon Generator

Dungeon Generation Module

Written By
Carlo Cigaral

Reviewed By
Gavin Lim
Abdulrahman Asfari
Gideon Wilkins

Technical Outline - Dungeon Builder

Algorithm Brief

Binary Space Partitioning (BSP) is the chosen procedural map generation algorithm and works by recursively dividing a given bounding space into smaller regions until a termination criteria is met. This generates a hierarchical binary tree structure which enables easy manipulation of individual regions of space for a variety of post processing procedures to create complex and intricate layouts.

BSP is being used as opposed to other algorithms as the balance between map complexity and implementation simplicity lends itself well to create a robust system focused on ease of integration with other smaller systems. This limits the computational stress focused onto the module and enhances performance scalability, giving much more freedom and customizability for the user.

Module Structure

The Dungeon Generation module makes use of the Builder design pattern as it places an emphasis on the separation of concerns between the construction of a map and its representation that is being returned to the modules using the builder. This module provides the following public setter methods for the respective fields for any given map:

setSize()

Sets the height and width of the map.

setMinGap()

Sets the minimum distance between any two partitions placed through BSP

setMaxDepth()

Sets the maximum recursion depth of BSP

setTotalCoverage()

Sets the desired total percentage floor coverage of the map

setSeed()

Sets the seed used for pseudo random number generation throughout the map construction process

setPreset()

An optional setter which utilises preset values to automatically set all parameters.

Dungeon Building Process

1. When the 'build()' method is called each field is checked to see if they have all been correctly set prior to generating a map.
2. The BSP algorithm is then run to generate the partitions which define the layout of the map. A temporary region object is created to represent the region of space created by BSP, each of these define where individual rooms may be placed.

3. Random selection is performed on the collection of regions to decide which will be kept and which are discarded, this continues until the desired floor coverage is met.
4. For each region that is kept, random room dimensions and position are calculated within the region and a Room object is created and pushed to a new list.
5. This list is passed onto the Hallway generation module to add hallway connections between rooms and the list is returned now containing hallways.
6. The final list of room and hallway objects are then passed on for rendering.

Further discussion on the behaviour of the algorithm and how this is affected by each of the different fields of the builder can be found in the README file of the Dungeon Generation module.