Mark Damon Hughes

Game Design: Article 07: Roguelike Dungeon Generation

Index | Articles | Links

By Mark Damon Hughes <*kamikaze@kuoi.asui.uidaho.edu*>

The original Rogue algorithm is pretty nifty. Any time you need a random dungeon, give this a try:

1. Divide the map into a grid (Rogue uses 3x3, but any size will work).
2. Give each grid a flag indicating if it's "connected" or not, and an array of which grid numbers it's connected to.
3. Pick a random room to start with, and mark it "connected".
4. While there are unconnected neighbor rooms, connect to one of them, make that the current room, mark it "connected", and repeat.
5. While there are unconnected rooms, try to connect them to a random connected neighbor (if a room has no connected neighbors yet, just keep cycling, you'll fill out to it eventually).
6. All rooms are now connected at least once.
7. Make 0 or more random connections to taste; I find rnd(grid_width) random connections looks good.
8. Draw the rooms onto the map, and draw a corridor from the center of each room to the center of each connected room, changing wall blocks into corridors. If your rooms fill most or all of the space of the grid, your corridors will very short - just holes in the wall.
9. Scan the map for corridor squares with 2 bordering walls, 1-2 bordering rooms, and 0-1 bordering corridor, and change those to doors.
10. Place your stairs up in the first room you chose, and your stairs down in the last room chosen in step 5. This will almost always be a LONG way away.
11. All done!

Rogue also has "gone rooms", which just put a corridor space instead of the room, and draws L-shaped corridors instead of straight lines, but those are just flavor.

This algorithm also has the virtues of being extremely fast (even on MUCH bigger grid arrays than 3x3), and guaranteed to succeed.