Method

The method I have chosen is to create a starting room and branch rooms out from that one by randomly placing new rooms, making sure they are next to an already existing room. I will keep information on each room stored in a 2D array, such as how many neighbors the room has, its position on the grid, the type of room it is, and where the doors are. I will also keep a list of occupied rooms so that I can create systems to easily search through what rooms are taken and which are available.

Positives

This method will allow me to control the branching within the dungeon design by creating an algorithm that searches for and creates rooms with a certain amount of neighbors, this could be automated during generation to change between clusters of rooms and branches. I could also implement the ability to control how many rooms are created which could be automated to create scaling difficulty through dungeon size.

It also allows for easy minimap creating as the players position will always be stored as coordinates within the dungeon, each room being one coordinate so there will always be an easy way to show where the player is, similar to binding of isaac.

This system could also be modified to allow for an open procedurally generated dungeon by removing the walls from the rooms and creating an edge detection algorithm to only place walls around the edge of the world you could create big procedurally generated environments with only slight tweaks and additions.

Negatives

One big negative is this method does not allow for the custom creation of rooms, meaning all rooms will have to be handmade and stored in a game object array. This could detract from the random feel of the dungeon layouts as you can see the same rooms repeated. Another negative is it would be hard to implement a system to randomize the connections between the rooms, as with the system currently all adjacent rooms are connected.