Map Generation

Objectives: To have a map generation engine that can fill maps with rooms in a random way while avoiding unnatural or undesirable formations, or otherwise being unable to fulfill room count or density criteria for a map.

Wish list:

1. Configurability of maps through parameters
2. Automatically show error messages or prohibit actions that would cause edge cases that break the generator by overlapping conflicting constraints, such as requiring more rooms of a certain size than a map has space for
3. Simplicity as well as customizability
4. Able to evenly distribute rooms or focus rooms in a certain area or areas
5. Able to control room density within certain bounds of a map on a per-room basis or by type of room

Bad cases:

-Running out of space to fill in necessary rooms such as the door and exit

-uninteresting room layouts such as too many of a certain map too close together or too numerous

-too many rooms with content in a floor, or too few rooms with content in a floor

Implementation Ideas:

-Iterate through grid on map to make a list of squares that mark valid areas for placing a given room. Add their coordinates too a list, use rng to select one coordinate, place the room at that location

-Ability to ensure specific rooms to appear in a map in addition to normal generation:

-Have variables that can be changed to control:

Wish list 4: have the iteration for checking for room spaces act within defined bounds of x and y.

-Rooms should be pre-designed for the sake of well-crafted engagements that emphasize difference in how the player must approach them, not just in superficial configuration variations. Enemy number should be proportional to room size, and enemy type should be in consideration to room shape and obstacles. This also lends the levels to be designed more like a series of trpg maps neighboring each other, and be less random and uninspiring than Disgaea’s item world. This ALSO prevents randomly designed levels that don’t make sense, such as a c-shaped dining room with three full tables and uneven numbers of chairs on each side of each table.

-The entire level will start out as a grid map that’s empty. The map size starts at the largest possible room size and iterates through the map from left to right, and top to bottom. It starts with an area of grid squares that would fit the map, and randomly chooses whether or not to spawn that room there based on the room’s spawn chance and its chance of spawning next to certain other types of rooms. If it spawns the room, it checks to see if the maximum number of rooms of that type are allowed. If that number isn’t reached, it moves to the next empty area of sufficient size and rolls for the chance to spawn the room again until the limit for that type of room is reached. It then moves on to the next room of the same size and tries again with the next stretch of empty space. When it runs out of rooms of that same size it goes on to the next smallest set of rooms and tries to find spaces that fit those.

-It searches for spaces to put rooms in by finding an area of grid squares that is large enough to fit the room. It will iterate to the right by one grid each time it doesn’t fill in a room, and if it does fill in the room it starts from the next empty square after the space the room was filled in. When it reaches the end of a row, it moves on to the next row of grid squares down. It iterates one grid square at a time, using that square as the origin to check the area from.

-Room density will be controlled by a set of percentage values. A map will have percentages of its map reserved for certain types of rooms, measured in tiles taken up by that room type. Having 50% of the map dedicated to 5x5 maps will mean that the room generator will iterate through and fill the map with rooms of size 5x5 until half of the existing grid squares on the map are taken up by 5x5 tiles, in which case it will stop filling in 5x5 tiles. This will be an additional condition to stopping filling rooms, aside from lack of available spaces to place them and maximum number of the room type allowed in a map(which is also set by hand). The percentage of the total available map space will be taken after filling the map with “reserved” rooms.

-“reserved” rooms will be rooms considered mandatory to be included in a map in certain quantities. These rooms will be added first before other rooms of their size class. If no space on the map is allotted to rooms of their size class, they will be added before rooms of a size class before theirs. The map generator will fill them in as normal, except that if it iterates to the end of the map without filling it in then it will look at the last fillable position and fill that in with 100% probability

-