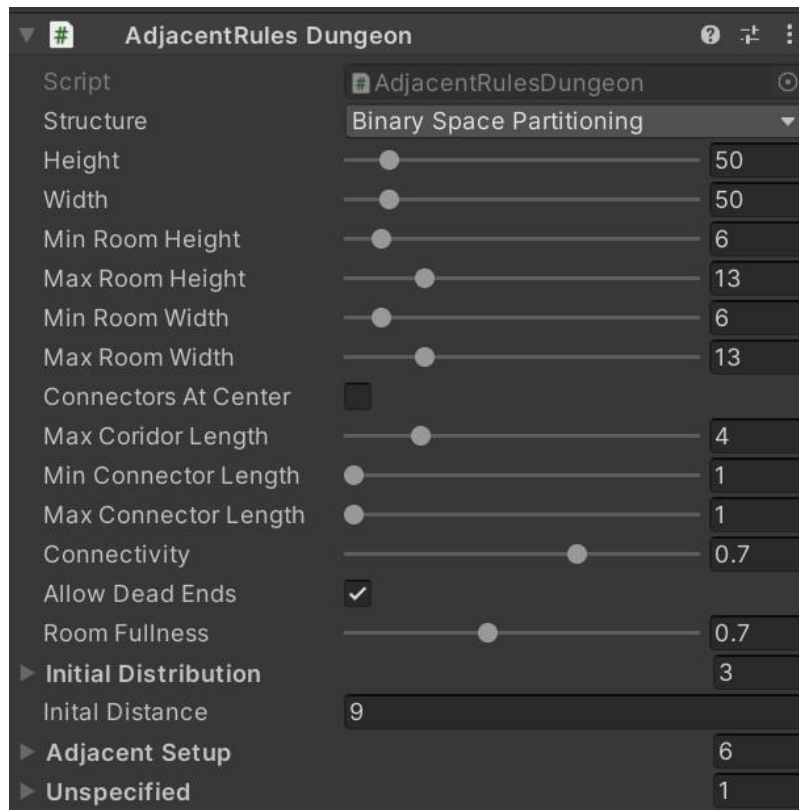


# AdjacentRules Dungeon



This layout generator is very similar to Random Dungeon, but uses different way to assign tags:

1. Randomly distributes **InitialDistribution**, according to given count for each tag, and also maintains **Initial Distance** between their centers.
2. After step 1 is done, it starts walking in all directions from each initial room, assigning tags according to given chances.
3. If some rooms were not given tag - selects from Unspecified list according to given relative chance.

## Example

Let's set up a dungeon, where we initially spawn 3 rooms: Start, End and Weapons, and want to fill the dungeon with good(Health, Loot) and bad(Monsters, Traps) rooms, and make it that way, that when going out of bad room you will likely get to good room, and when going out of good room you will likely get to bad room. We setup, that from Start, End, Weapons, Loot, Health we have chance to assign

Start

Tag

Start

Adjacent Chances

2

Monsters

1

Traps

1

+

-

Monsters

Tag

Monsters

Adjacent Chances

2

Health

2

Loot

1

+

-

Traps

Tag

Traps

Adjacent Chances

2

Health

2

Loot

1

+

-

Health

Tag

Health

Adjacent Chances

2

Monsters

2

Traps

1

+

-

Loot

Tag

Loot

Adjacent Chances

2

Monsters

2

Traps

1

+

-

End

Tag

End

Adjacent Chances

2

Monsters

2

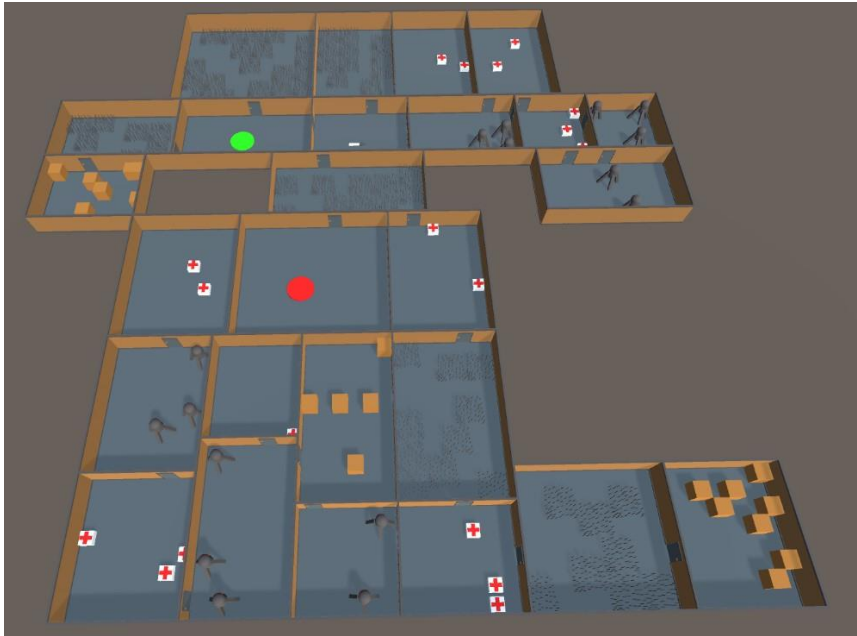
Traps

1

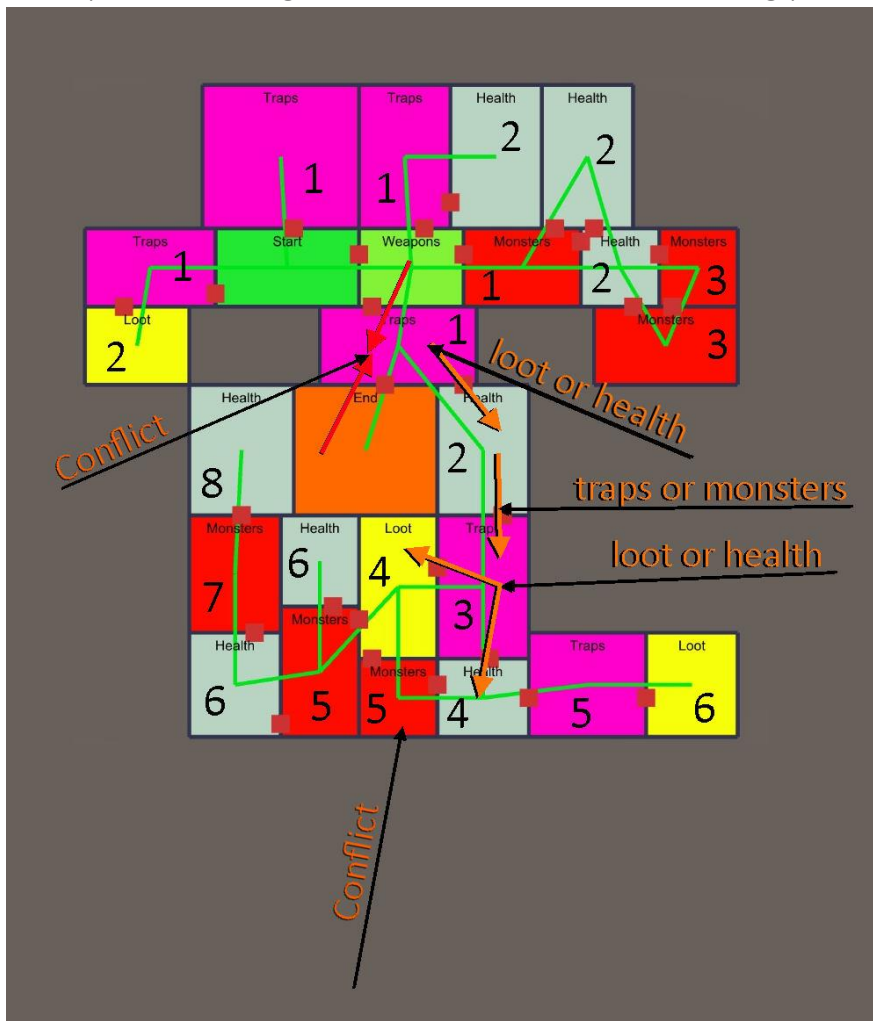
+

-

[illegible]



Example of how algorithm works is on the following picture:



Number shows order of room visiting(initial rooms have no number)  
Conflicts are resolved by first path to get there, therefore order of tags in  
InitialDistribution can affect algorithm.