Levels

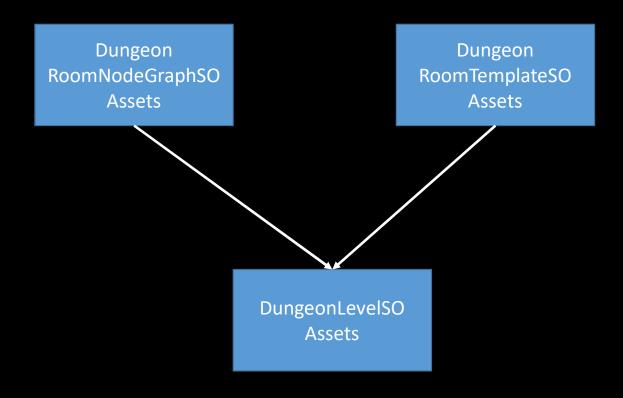
Dungeon Builder Concepts Levels

We've created multiple room node graph scriptable object assets for each level.

We also have multiple dungeon room tilemap template prefabs and corresponding room template scriptable object assets .

What we need now is a way of linking together which room node graphs and room templates should be used for a particular level.

Dungeon Builder Concepts Levels



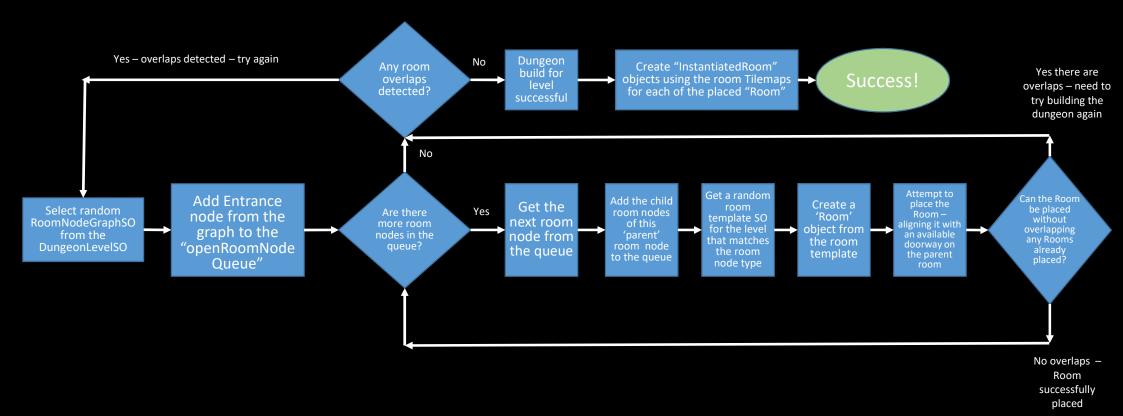
Dungeon Builder Concepts Levels

The DungeonLevelSO asset will have a list of all the room node graph scriptable objects for a level and also a list of all the room template scriptable objects that should be used for that level.

When the Dungeon Builder algorithm wants to build a dungeon level it will use DungeonLevelSO to pick a random room node graph and pick random room templates for each room node type in the graph.

Dungeon Building Algorithm

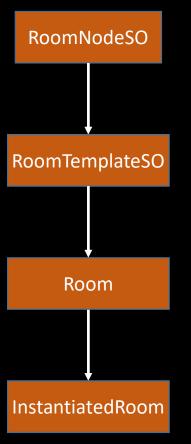
Dungeon Building Algorithm



Dungeon Room Classes Used By The Dungeon Builder

The Room Node Graphs contain RoomNodeSO scriptable objects that are linked together to define the level.

A 'Room' object is created when the Dungeon Builder tries to place a room without overlaps. This room 'placement' happens by looking at the bounding boxes of all the 'Room' objects when they are placed. At this point the actual tilemap rooms aren't created.



The Dungeon Builder selects a random RoomTemplateSO that matches the RoomNodeSO (RoomNodeTypeSO).

The InstantiatedRoom component will be added to the dungeon room tilemap prefabs. After the dungeon has been successfully built with no overlaps using the 'Room' objects, the Dungeon Builder then instantiates all the room tilemap prefabs (including the InstantiatedRoom components). The InstantiatedRoom component will hold a reference to the parent 'Room' and references to the tilemaps in the instantiated room prefab. The InstantiatedRoom component will contain methods to block off any unused doorways on the instantiated tilemap.