
RANDOMIZED ROOM FORMAT

OFFICIAL DOCUMENTATION

Basic Concepts

Before getting into the randomized room format, there are some basic concepts you will need to know first.

Tiles and Collision

There are 256 tiles in each tilemap, each represented by one byte. Each of the 256 tiles are mapped to a single collision mode (also one byte), which determines how entities interact with that tile. Here is a list of collision modes:

- **Mode 00:** None. Entities do not collide with these tiles, making them act like empty space. Useful for empty space and background decorations.
- **Mode B0:** Solid. These tiles act like solid walls and will prevent entities from passing through them in any direction.
- **Mode F0:** Not touching. This mode is used when an entity is completely outside of a tilemap and acts equivalently to mode 00. Because it is used when an entity is outside of any tilemap, it should not be assigned to any tiles.
- **Mode FF:** Undefined. This collision mode should be assigned to tiles that should not be mapped in any tilemap. Upon touching this collision mode (or any other unknown collision mode) the game will throw an error and crash.

Collision modes are assigned in a collision map (`.c1mp`) file, which is exactly 256 bytes long. The file is indexed using a tile ID, giving tile 00 the first collision mode located within the file, and tile FF the last collision mode located within the file.

Tilemaps and Tilemap Colliders

A tilemap represents a grid of tiles, with each tile having a position within the tilemap (origin is the top left). Tilemaps require graphics data in the form of a texture to function properly, and this texture must contain exactly 256 tiles (16 tiles wide and 16 tiles tall). By default, the tilemap is filled with tile 00, and as such it is a good idea to have tile 00 be empty space (but is not required).

Tilemaps do not have any collision on their own, but they can be given collision by creating a tilemap collider that links it with a collision map (mentioned in the previous section). This tilemap collider can then be used to check if entities are touching the tiles in a tilemap. Tilemap colliders can also be stored within a tilemap, if required.

Rooms

Rooms make up the overall world map, and as such are a very important part of the game. Rooms can either be initialized or uninitialized, which determines what data they store:

- **Uninitialized rooms (room definitions):** These rooms do not have an internal layout associated with them, and only store data about the way the room connects to other rooms, as well as the section that it is within.
- **Initialized rooms (full rooms):** These rooms have an internal layout associated with them, in addition to the same data as uninitialized rooms.

Rooms can be initialized using a map generator and a randomized room template file.

Randomized Room Templates

Randomized room templates are files that guide the map generator towards generating the internal layout of a room by providing randomization instructions. In order to make randomized room templates, you will need a hex editor.

Header

The header is the section of the randomized room template before the room data itself begins. However, the file must also begin with a pre-header, which determines the size of the room. The pre-header looks like this:

```
'RRM'[3] 0x00[9] width[2] height[2]
```

- The width and height are little-endian.
- There can be as many **FF** bytes for padding within the header as you want.

Pools

Pools are sets of tiles identified with a 16-bit integer, and tiles are randomly picked from pools in order to generate the room layout. A pool containing only one tile is called a constant pool, due to the fact that it will never be randomized.

A pool is defined in the header section of a randomized room template. The structure of a pool definition is below:

```
'P'[1] id[2] length[1] tiles[length]
```

- Pools must be aligned in 8-byte groups. Pad with **FF** if necessary. A trailing **FF** is not required.
- No two pools may have the same identifier.

Pool Instances

- Instances are values used to group pools together in order to create more complex randomized structures.
- The room data is made of pool instances.
- All pool instances which use the same instance number will have the same index into the pools selected. For example, two instances of the same pool that have the same instance number will select the same tile.

- The instance number **FFFF** will always generate random selections, regardless of how many pools use it. Use it when you don't care about the instance number.

A pool instance definition is as follows:

poolID[2] instanceNo[2]

Room Data

The actual room data begins with a short header:

'RMD' [3] 0x00[13]

After that, width × height pool instance definitions are chained until the entire size of the room has been filled.