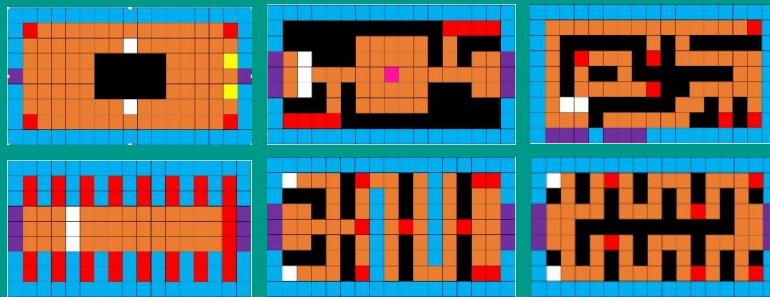


From Design to Unreal

When designing Nebula Knights' modular levels, many maps were exchanged between the team denoting different rooms colour coded to show specific tile properties. Measuring 9 x 17 tiles per room required placing over 150 tiles to see a single room in engine, and with over 50 room designs; building them, if only to visualise the design, became a daunting task. So I wrote a program to do it for us.

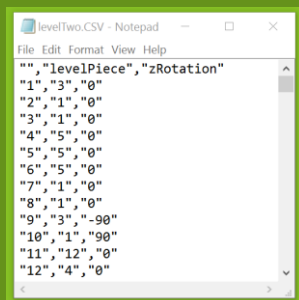


The python program (top right) uses the PyGame library to create a singleton of a room map image, and the csv library to return a .csv file of quoted integer values each corresponding to a tile colour defined as a modular level piece within a key. Spoofing an initial line denoting the column headers results in a file that Unreal Engine can import as a data table to be interpreted and used in blueprints. Unreal Engine can only import data in the form of quoted .csv files shown in the format below, so the python program is essentially an adapter for these incompatible data types (maps) and interpreter (Unreal Engine), so high coupling can be excused.

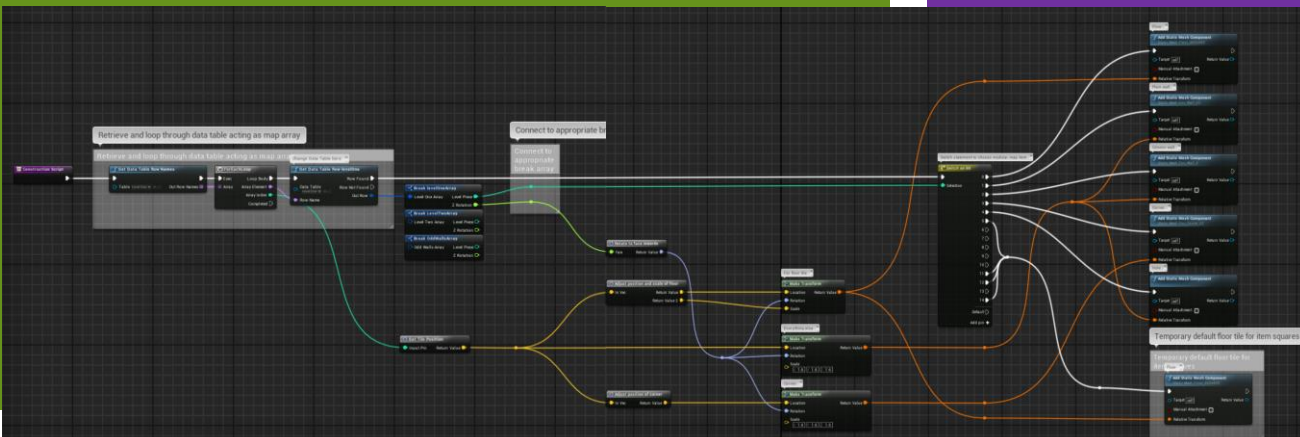
```

1  #
2  # Colour code
3  #
4  # ( 1, 276, 244) = Wall, blue
5  # ( 1, 2, 10) = Room, black
6  # (102, 40, 140) = Door, purple
7  #
8  # (254, 0, 0) = Empty space, red
9  # (255, 255, 255) = Unknown, white
10 # (255, 255, 1) = Team element, yellow
11 #
12 #
13 #
14 #
15 #
16 #
17 #
18 #
19 #
20 #
21 #
22 #
23 #
24 #
25 #
26 #
27 #
28 #
29 #
30 #
31 #
32 #
33 #
34 #
35 #
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #

```



Finally, the blueprint below acts as a factory, looping through whichever data table's rows it is supplied; using the index to calculate the position of the tile, the modular level piece, and any transform needed. In this case, a rotation around the z-axis so that all walls will face inwards.

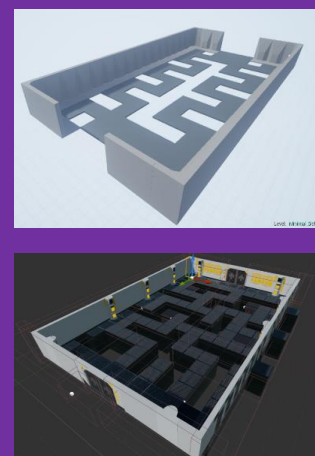


Python Map Reader

```

1  import sys, csv, pygame
2  from pygame.locals import *
3
4  pygame.init()
5
6  #
7  #
8  #
9  #
10 #
11 #
12 #
13 #
14 #
15 #
16 #
17 #
18 #
19 #
20 #
21 #
22 #
23 #
24 #
25 #
26 #
27 #
28 #
29 #
30 #
31 #
32 #
33 #
34 #
35 #
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #

```



The result is a serviceable mock up of the room (top). Whilst it lacks the personal touch of a hand-crafted designers level (below), it can very quickly help to visualise any of the designer's level maps in engine. Even were they crudely drawn on paper, this program could identify the colours of each tile, and parse the data into a quoted .csv file, which the blueprint could build a room from.

Richard
Steele
Nebula Knights -
Bosstick Studios

