

map2.py

```
1  # A simple program to draw a map of rooms and connections using Tkinter
2  # This program uses a JSON file to load room data and visualize the connections
   between them.
3  # The JSON file should contain room names and their connections in a specific
   format.
4  # The program uses Tkinter to create a graphical window and draw rectangles for
   rooms and lines for connections.
5  # The program also includes a recursive function to place rooms and draw
   connections based on the loaded data.
6
7  import tkinter as tk
8  import json
9
10 # Load room data uncomment as required
11 #datafile = "star_wars.json"
12 #datafile = "game_data.json"
13 #datafile = "game_data.json3"
14
15
16 with open(datafile) as f:
17     data = json.load(f)
18
19 rooms = data["rooms"]
20
21 directions = ["North", "South", "East", "West", "Up", "Down", "Out"]
22
23 # Define direction-based position offsets
24 direction_offsets = {
25     "North": (0, -1),    #up
26     "South": (0, 1),     #down
27     "East": (1, 0),      #right
28     "West": (-1, 0),     #left
29     "Up": (1, -1),       #up-right
30     "Down": (-1, 1),     #down-left
31     "Out": (-1, -1)
32 }
33
34 #width and height of window
35 w = 1200
36 h = 800
37
38 # Tkinter window setup
39 root = tk.Tk()
40 root.title("Room Map Viewer")
41
42 # we need a canvas widget to draw on
43 canvas = tk.Canvas(root, width=w, height=h, bg="white")
44 canvas.pack()
45
46
47 x0, y0 = 500, 600 # Center starting coordinates
48
```

```
49 room_coords = {} # Store coordinates of each room to draw connections
50 sizex = 40 # size of box to draw
51 sizey = 40 # size of box to draw
52
53 # Create a Visited set . A set is unchangeable!
54 visited = set()
55
56
57 #draw the room as a rectangle shape
58 def draw_room(name, x, y, colour):
59     room_coords[name] = (x, y)
60     canvas.create_rectangle(x - sizex, y - sizey, x + sizex, y + sizey,
61 fill=colour)
62     canvas.create_text(x, y-30, text=name, font=("Arial", 8), justify="center")
63
64     ## print items in the room
65     items = rooms[name].get("Item", [])
66     if items:
67         item_text = items
68         canvas.create_text(x, y + 10, text=item_text, font=("Arial", 6),
69 fill="blue")
70
71     #print enemies in the room
72     enemies = rooms[name].get("Enemy", [])
73     if enemies:
74         enemy_text = enemies
75         canvas.create_text(x, y + 20, text=enemy_text, font=("Arial", 7),
76 fill="red")
77
78 def draw_connection(from_room, to_room, direction, label=True):
79
80     # only draw a connection if we've drawn it
81     if to_room not in room_coords: #room_coords are only added when we draw it (
82 see draw_room() function above )
83         return
84
85     x1, y1 = room_coords[from_room]
86     x2, y2 = room_coords[to_room]
87     canvas.create_line(x1, y1, x2, y2, arrow=tk.LAST)
88
89     # draw direction in middle of line if required
90     if label:
91         mid_x, mid_y = (x1 + x2) // 2, (y1 + y2) // 2
92         canvas.create_text(mid_x, mid_y - 10, text=direction, font=("Arial", 6),
93 fill="grey")
94
95 # RECURSIVE function to place room and connections
96 def place_room(name, x, y, colour="lightblue"):
97
98     #if you've been here before then go back
99     if name in visited:
100         return
101
102     #add room to visited set and draw
```

```
98     visited.add(name)
99     draw_room(name, x, y, colour)
100
101     # loop through all exits and draw those
102     for direction, target in rooms[name].items():
103
104         # Only draw directions
105         if direction not in directions:
106             continue
107
108         # find where to draw the room - if unknown then draw on top
109         dx, dy = direction_offsets.get(direction, (0, 0))
110         new_x = x + dx * 110
111         new_y = y + dy * 110
112
113         #if not already drawn - then draw it
114         if target not in room_coords:
115             place_room(target, new_x, new_y)
116
117         #now connect together
118         draw_connection(name, target, direction, label=False)
119
120 # Start drawing from the starting room
121 start_room = data.get("start", "No start room")
122
123 place_room(start_room, x0, y0, colour="limegreen")
124
125 root.mainloop()
126
```