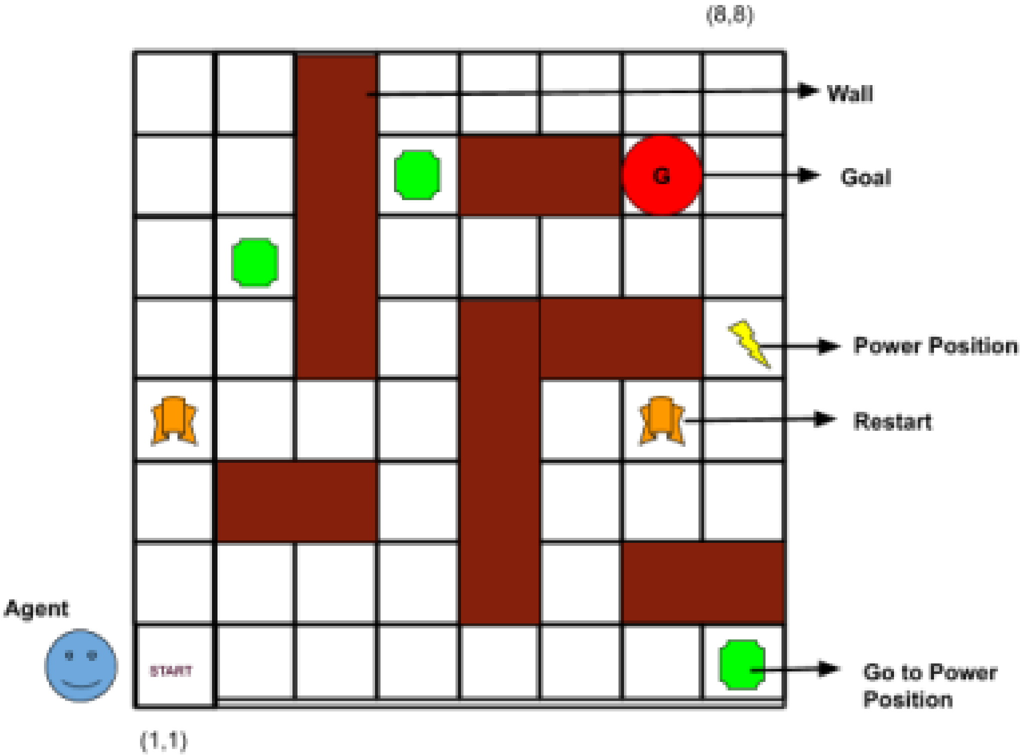
Problem Statement

a. Make an 8x8 grid agent world as shown below.



b. **The Environment: AGENT WORLD**

There are 5 components to take a note of,

1. **Go to Power Position :** On Reaching here, agent will start its next move from **"Power Position"** as shown
2. **Goal :** The location of the goal should be randomly selected from the range (x>5, y>5) eg. (6,6), (8,8), (8,6) etc., every time the code is run, the goal should change
3. **Restart:** The Agent will need to go restart and go to position (1,1) 4. **Brown Blocks :** Walls (Agent isn't allowed to penetrate through these positions)
5. **Start:** Agent will be starting from (1,1) every time the game is started.

c. **Details:**

This assignment is aimed to motivate students to design a reinforcement learning-based model by thinking about policies, goals, knowledge base, environment, and reward function, etc.

You are required to build a reinforcement learning game for the agent and environment as shown above. The position of the goal must be kept in the suggested region and the position of the walls and other entities must be kept unchanged strictly.

You may refer to some searching algorithms to reach the goal to design the policies and further reward functions, the path followed path cost, etc. For example, Simulated Annealing, Local Search, A* Search etc.

d**. To Do:**

1. **Knowledge Base Creation:** How(in what format) the information of the visited nodes is stored in the agent's memory.
   Print the knowledge base with every iteration
2. **Policy Design:** What set of rules are considered to design the agent's behavior/ interaction in the environment.
3. **Reward Function Design:** Design a reward function that will help the agent to reach the goal as early as possible. Note: You can take inference from the Romania Map diagram in the book Peter Norvig-Russel. (A* search Section)
4. **Path Cost:** Print the total Path cost for the path followed by the agent. (Optimal path cost must be there in the end, as with more iterations, the agent will interact more with the environment and thus learn more which leads to more knowledge in the Knowledge base.
5. **Path Followed:** Print the path followed for every iteration and the most optimal path.