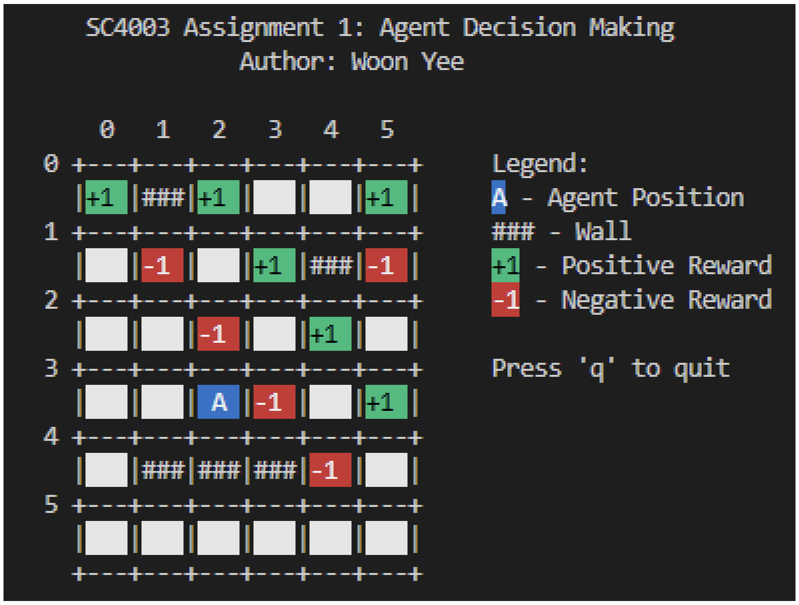
NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

Semester 2 AY 24/25

SC4003 Intelligent Agent Assignment 1

Completed by:
Ng Woon Yee (U2120622C)

Initial Maze Environment:



```
     SC4003 Assignment 1: Agent Decision Making
                Author: Woon Yee

     0   1   2   3   4   5
 0 +---+---+---+---+---+---+    Legend:
   |+1 |###|+1 | ⬜ | ⬜ |+1 |    A - Agent Position
 1 +---+---+---+---+---+---+    ### - Wall
   | ⬜ |-1 | ⬜ |+1 |###|-1 |    +1 - Positive Reward
 2 +---+---+---+---+---+---+    -1 - Negative Reward
   | ⬜ | ⬜ |-1 | ⬜ |+1 | ⬜ |
 3 +---+---+---+---+---+---+    Press 'q' to quit
   | ⬜ | ⬜ | A |-1 | ⬜ |+1 |
 4 +---+---+---+---+---+---+
   | ⬜ |###|###|###|-1 | ⬜ |
 5 +---+---+---+---+---+---+
   | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ |
   +---+---+---+---+---+---+
```

Part 1:

Value iteration:

- Descriptions of implemented solutions (5 marks)

The utility of a state is given by

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U(s')$$

Which means the utility of a state is the immediate reward for that state plus the expected discounted utility of the next state, assuming that the agent chooses the optimal action.

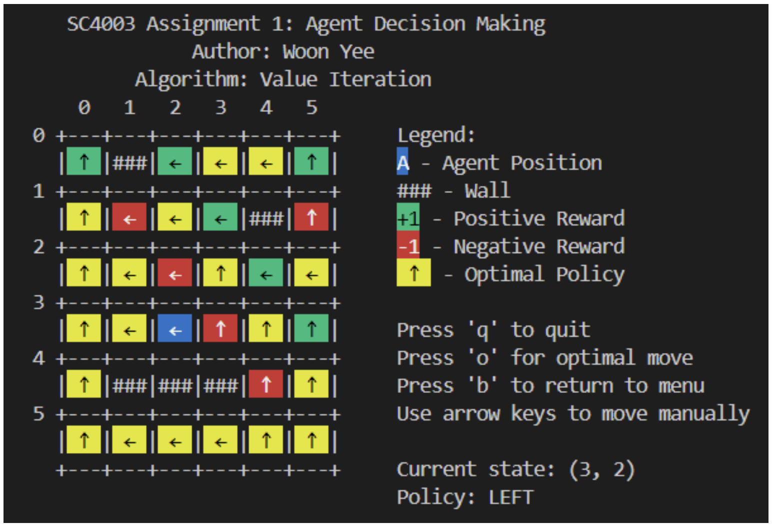
This is called the Bellman equation, and it is the basis of value iteration algorithm for solving Markov Decision Processes. If there are n possible states, that means there are n Bellman equations for the iteration. However, the problem is the Bellman equations are non-linear due to the max operator, and hence these systems cannot be solved by pure linear algebra techniques. And thus, iterative approaches. Initially, we start the utilities with 0. We let $U_i(s)$ be the utility value for state s at the i-th iteration. The iteration step is as follows:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a)\, U_i(s')$$

The convergence criterion is based on Bellman error bound:

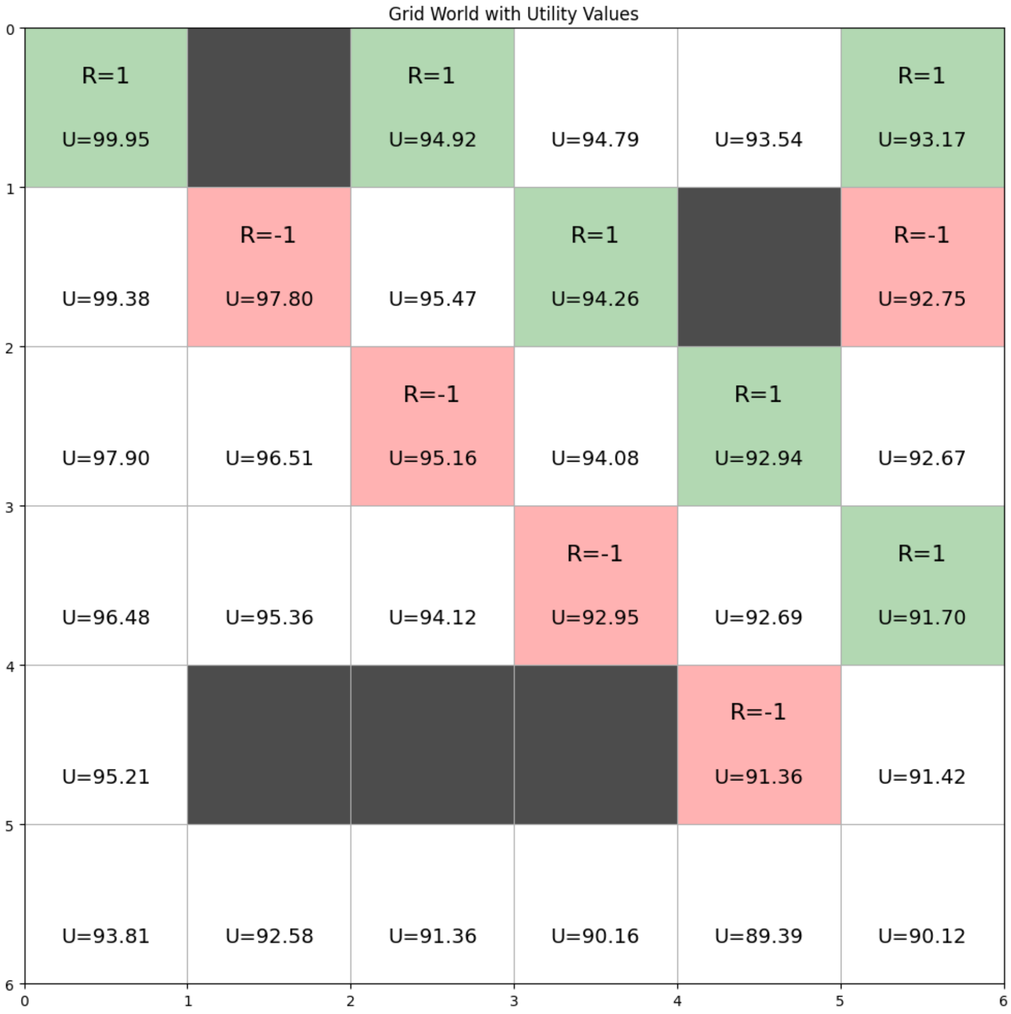
$$\delta \ < \ \epsilon(1-\gamma)/\gamma$$

- Plot of optimal policy (10 marks)



```
     SC4003 Assignment 1: Agent Decision Making
                 Author: Woon Yee
             Algorithm: Value Iteration
   0   1   2   3   4   5
0 +---+---+---+---+---+---+     Legend:
  | ↑ |###| ← | ← | ← | ↑ |     A - Agent Position
1 +---+---+---+---+---+---+     ### - Wall
  | ↑ | ← | ← | ← |###| ↑ |     +1 - Positive Reward
2 +---+---+---+---+---+---+     -1 - Negative Reward
  | ↑ | ← | ← | ↑ | ← | ← |     ↑ - Optimal Policy
3 +---+---+---+---+---+---+
  | ↑ | ← | ← | ↑ | ↑ | ↑ |     Press 'q' to quit
4 +---+---+---+---+---+---+     Press 'o' for optimal move
  | ↑ |###|###|###| ↑ | ↑ |     Press 'b' to return to menu
5 +---+---+---+---+---+---+     Use arrow keys to move manually
  | ↑ | ← | ← | ← | ↑ | ↑ |
  +---+---+---+---+---+---+     Current state: (3, 2)
                                Policy: LEFT
```

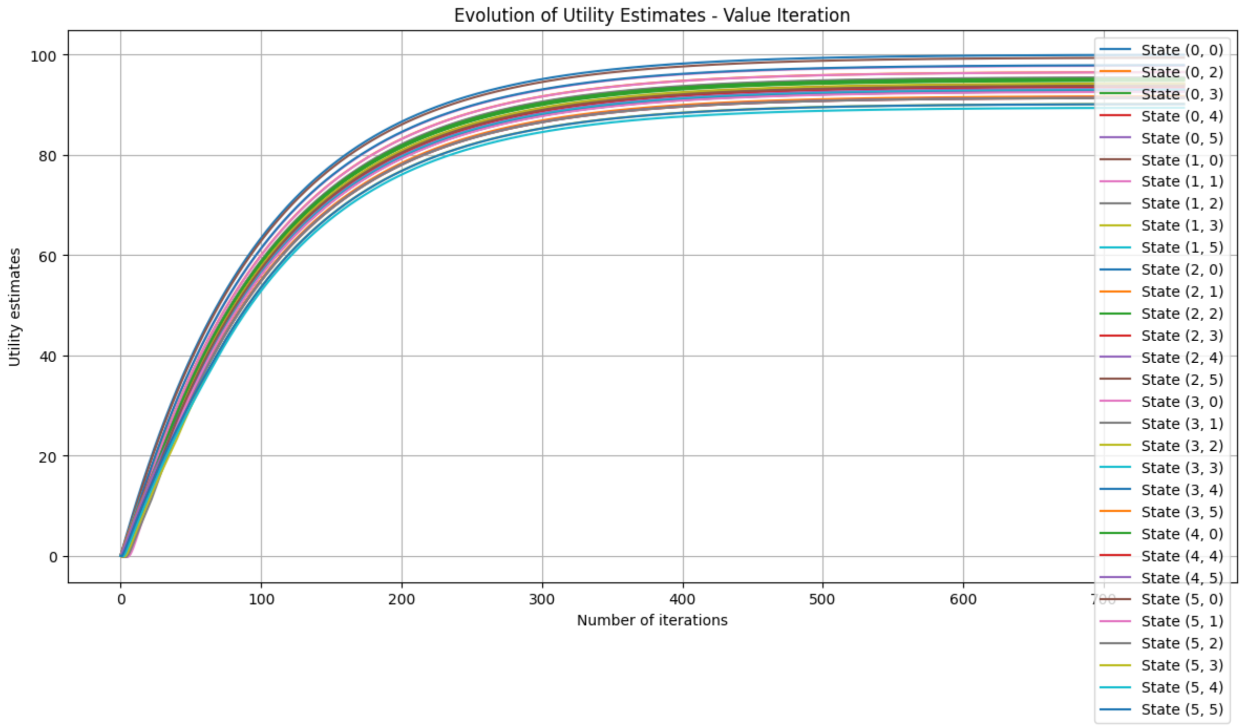
- Utilities of all states (10 marks)



Grid World with Utility Values

	0	1	2	3	4	5
0	R=1 U=99.95		R=1 U=94.92	U=94.79	U=93.54	R=1 U=93.17
1	U=99.38	R=-1 U=97.80	U=95.47	R=1 U=94.26		R=-1 U=92.75
2	U=97.90	U=96.51	R=-1 U=95.16	U=94.08	R=1 U=92.94	U=92.67
3	U=96.48	U=95.36	U=94.12	R=-1 U=92.95	U=92.69	R=1 U=91.70
4	U=95.21				R=-1 U=91.36	U=91.42
5	U=93.81	U=92.58	U=91.36	U=90.16	U=89.39	U=90.12

- Plot of utility estimates as a function of the number of iterations (10 marks)



Policy iteration:

- Descriptions of implemented solutions (5 marks)

In value iteration, we observed that it is possible to get an optimal policy even when the utility function estimate is accurate. If one action is clearly better than all other choices, then exact magnitude of the utility function need not to be precise. Hence, policy iteration algorithm is proposed, altering between two steps:

- Policy Evaluation: Given a policy π_i, calculate the utility of each state if $\pi_i$were to be executed. This is simpler than the standard Bellman Equation as it removes the max operator and restore the linearity of the equations.
- Policy Improvement: Calculate a new policy π_{i+1}, using one step look ahead based on U_i

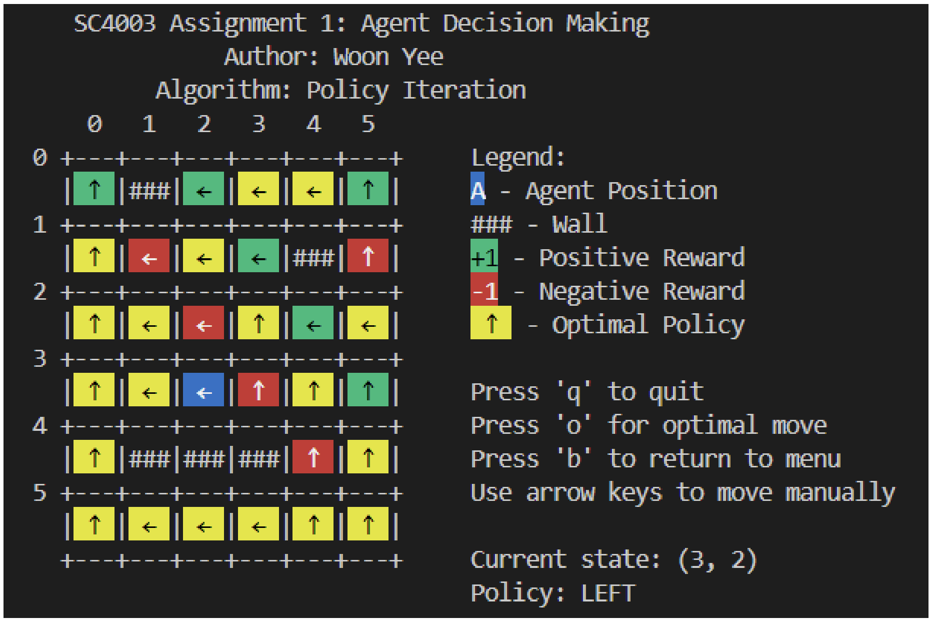
The utility function in policy iteration is given by

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s\prime} P(s'|s, \pi_i(s))\, U_i(s')$$

And the policy improvement step is as follows:

$$\pi^{i+1}(s) \ = \ \underset{a \in A(s)}{\text{argmax}} \sum_{s\prime} P(s'|s,a)U \quad (s')$$

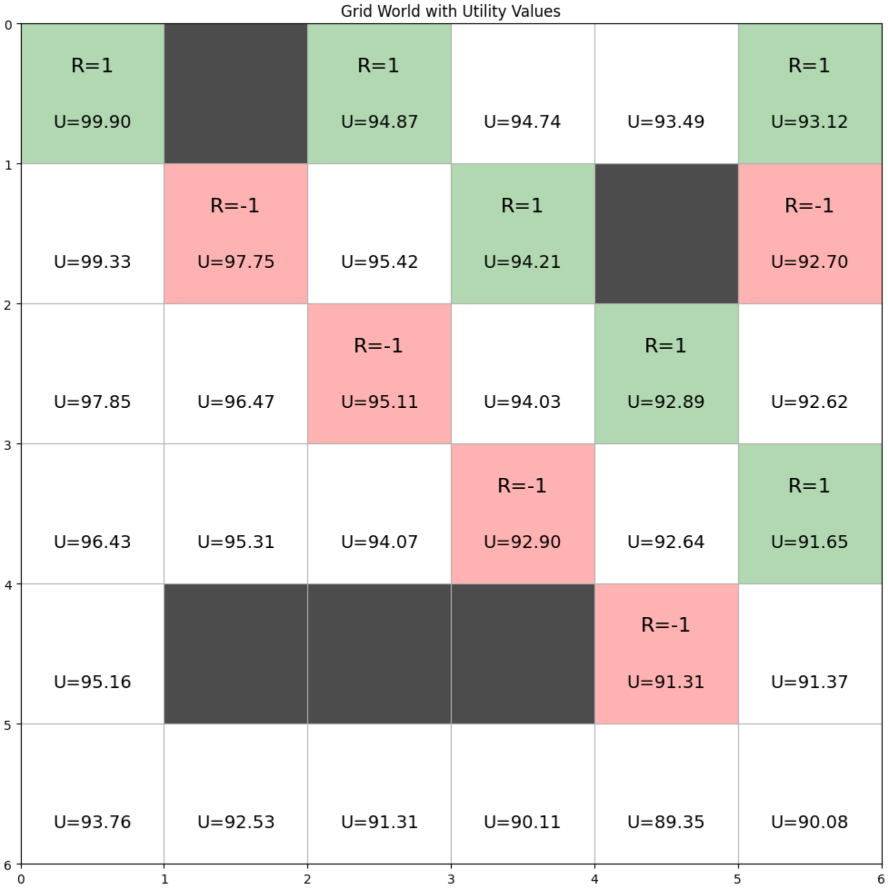
- Plot of optimal policy (10 marks)



```
    SC4003 Assignment 1: Agent Decision Making
              Author: Woon Yee
         Algorithm: Policy Iteration
      0   1   2   3   4   5
  0 +---+---+---+---+---+---+       Legend:
    | ↑ |###| ← | ← | ← | ↑ |       A - Agent Position
  1 +---+---+---+---+---+---+       ### - Wall
    | ↑ | ← | ← | ← |###| ↑ |       +1 - Positive Reward
  2 +---+---+---+---+---+---+       -1 - Negative Reward
    | ↑ | ← | ← | ↑ | ← | ← |       ↑ - Optimal Policy
  3 +---+---+---+---+---+---+
    | ↑ | ← | ← | ↑ | ↑ | ↑ |       Press 'q' to quit
  4 +---+---+---+---+---+---+       Press 'o' for optimal move
    | ↑ |###|###|###| ↑ | ↑ |       Press 'b' to return to menu
  5 +---+---+---+---+---+---+       Use arrow keys to move manually
    | ↑ | ← | ← | ← | ↑ | ↑ |
    +---+---+---+---+---+---+       Current state: (3, 2)
                                    Policy: LEFT
```

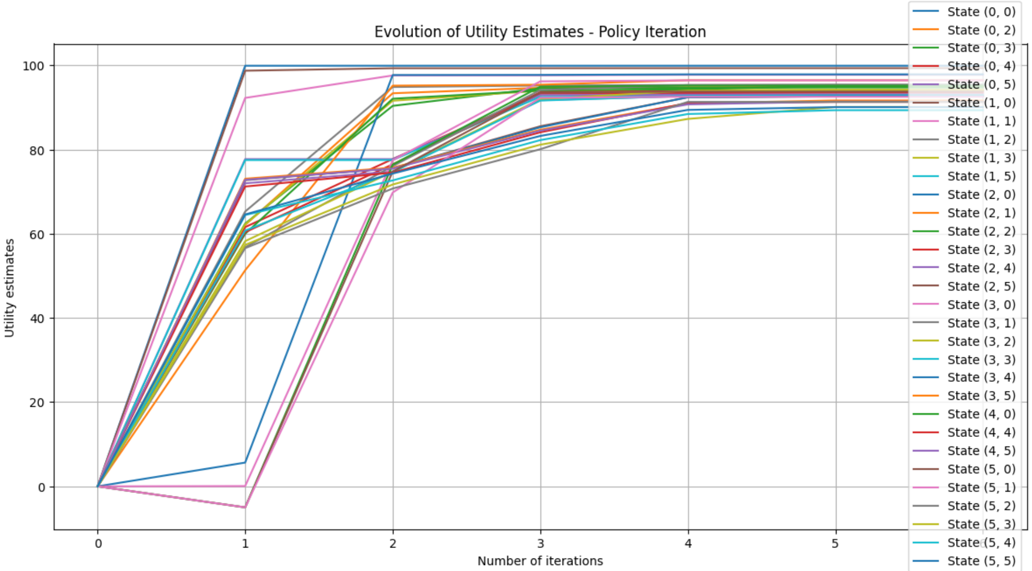
- Utilities of all states (10 marks)

Grid World with Utility Values



R=1 U=99.90		R=1 U=94.87	U=94.74	U=93.49	R=1 U=93.12
U=99.33	R=-1 U=97.75	U=95.42	R=1 U=94.21		R=-1 U=92.70
U=97.85	U=96.47	R=-1 U=95.11	U=94.03	R=1 U=92.89	U=92.62
U=96.43	U=95.31	U=94.07	R=-1 U=92.90	U=92.64	R=1 U=91.65
U=95.16				R=-1 U=91.31	U=91.37
U=93.76	U=92.53	U=91.31	U=90.11	U=89.35	U=90.08

- Plot of utility estimates as a function of the number of iterations (10 marks)

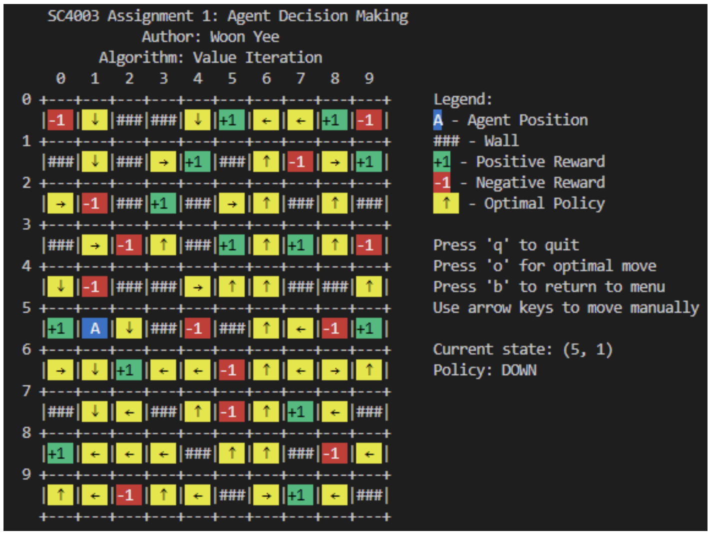


<u>Source Code (Include Part 1 and 2) – 15 Marks:</u>
https://github.com/woonyee28/SC4003-Intelligent-Agent-Assignment-1

Part 2:

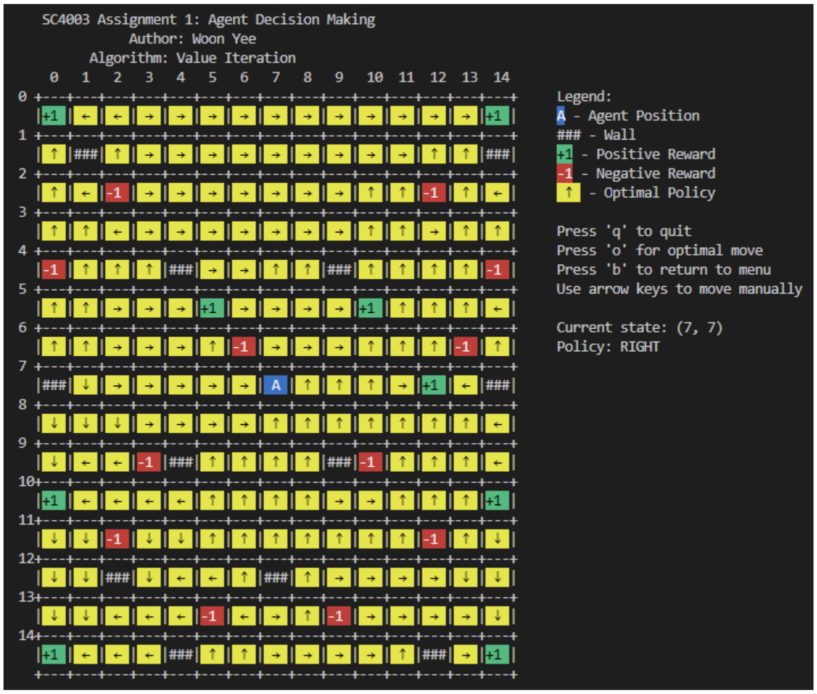
1. Design a more complicated maze environment of your own and re-run the algorithms designed for Part 1 on it.

10x10:



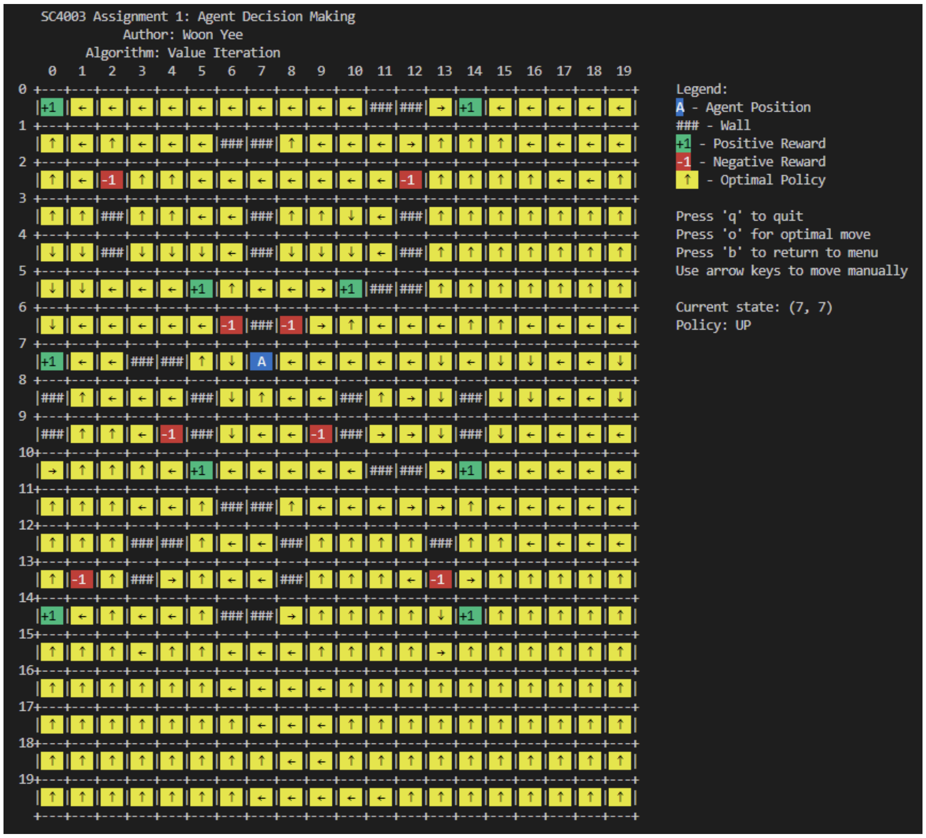
```
     SC4003 Assignment 1: Agent Decision Making
               Author: Woon Yee
          Algorithm: Value Iteration
     0   1   2   3   4   5   6   7   8   9
 0 +---+---+---+---+---+---+---+---+---+---+     Legend:
   |-1 | ↓ |###|###| ↓ |+1 | ← | ← |+1 |-1 |     A - Agent Position
 1 +---+---+---+---+---+---+---+---+---+---+     ### - Wall
   |###| ↓ |###| → |+1 |###| ↑ |-1 | → |+1 |     +1 - Positive Reward
 2 +---+---+---+---+---+---+---+---+---+---+     -1 - Negative Reward
   | → |-1 |###|+1 |###| → | ↑ |###| ↑ |###|     ↑ - Optimal Policy
 3 +---+---+---+---+---+---+---+---+---+---+
   |###| → |-1 | ↑ |###|+1 | ↑ |+1 | ↑ |-1 |     Press 'q' to quit
 4 +---+---+---+---+---+---+---+---+---+---+     Press 'o' for optimal move
   | ↓ |-1 |###|###| → | ↑ | ↑ |###|###| ↑ |     Press 'b' to return to menu
 5 +---+---+---+---+---+---+---+---+---+---+     Use arrow keys to move manually
   |+1 | A | ↓ |###|-1 |###| ↑ | ← |-1 |+1 |
 6 +---+---+---+---+---+---+---+---+---+---+     Current state: (5, 1)
   | → | ↓ |+1 | ← | ← |-1 | ↑ | ← | → | ↑ |     Policy: DOWN
 7 +---+---+---+---+---+---+---+---+---+---+
   |###| ↓ | ← |###| ↑ |-1 | ↑ |+1 | ← |###|
 8 +---+---+---+---+---+---+---+---+---+---+
   |+1 | ← | ← | ← |###| ↑ | ↑ |###|-1 | ← |
 9 +---+---+---+---+---+---+---+---+---+---+
   | ↑ | ← |-1 | ↑ | ← |###| → |+1 | ← |###|
   +---+---+---+---+---+---+---+---+---+---+
```

15x15:



```
    SC4003 Assignment 1: Agent Decision Making
               Author: Woon Yee
          Algorithm: Value Iteration
    0   1   2   3   4   5   6   7   8   9  10  11  12  13  14
0 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Legend:
  |+1 | ← | ← | → | → | → | → | → | → | → | → | → | → | → |+1 |    A - Agent Position
1 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    ### - Wall
  | ↑ |###| ↑ | → | → | → | → | → | → | → | → | → | ↑ | ↑ |###|    +1 - Positive Reward
2 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    -1 - Negative Reward
  | ↑ | ← |-1 | → | → | → | → | → | → | → | ↑ | ↑ |-1 | ↑ | ← |    ↑  - Optimal Policy
3 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ← | → | → | → | → | → | → | → | ↑ | ↑ | → | ↑ | ↑ |    Press 'q' to quit
4 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Press 'o' for optimal move
  |-1 | ↑ | ↑ | ↑ |###| → | → | ↑ | ↑ |###| ↑ | ↑ | ↑ | ↑ |-1 |    Press 'b' to return to menu
5 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Use arrow keys to move manually
  | ↑ | ↑ | → | → | → |+1 | → | → | → | → |+1 | ↑ | ↑ | ↑ | ← |
6 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Current state: (7, 7)
  | ↑ | ↑ | → | → | → | ↑ |-1 | → | → | → | ↑ | ↑ | ↑ |-1 | ↑ |    Policy: RIGHT
7 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |###| ↓ | → | → | → | → | → | A | ↑ | ↑ | ↑ | → |+1 | ← |###|
8 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↓ | ↓ | ↓ | → | → | → | → | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ← |
9 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↓ | ← | ← |-1 |###| ↑ | ↑ | ↑ | ↑ |###|-1 | ↑ | ↑ | ↑ | ← |
10+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |+1 | ← | ← | ← | ← | ↑ | ↑ | ↑ | ↑ | → | → | ↑ | ↑ | ↑ |+1 |
11+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↓ | ↓ |-1 | ↓ | ↓ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |-1 | ↑ | ↓ |
12+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↓ | ↓ |###| ↓ | ← | ← | ↑ |###| ↑ | → | → | → | → | ↓ | ↓ |
13+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↓ | ↓ | ← | ← | ← |-1 | ← | → | ↑ |-1 | → | → | → | → | ↓ |
14+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |+1 | ← | ← | ← |###| ↑ | ↑ | → | → | → | → | ↑ |###| → |+1 |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

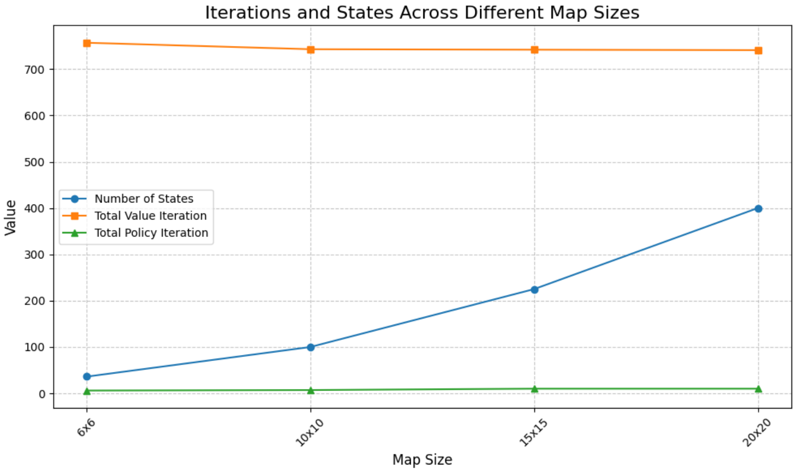
20x20:



```
    SC4003 Assignment 1: Agent Decision Making
               Author: Woon Yee
          Algorithm: Value Iteration
    0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
0 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Legend:
  |+1 | ← | ← | ← | ← | ← | ← | ← | ← | ← | ← |###|###| → |+1 | ← | ← | ← | ← | ← |    A - Agent Position
1 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    ### - Wall
  | ↑ | ← | ↑ | ← | ← | ← |###|###| ↑ | ← | ← | ← | → | ↑ | ↑ | ↑ | ← | ← | ← | ← |    +1 - Positive Reward
2 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    -1 - Negative Reward
  | ↑ | ← |-1 | ↑ | ↑ | ← | ← | ← | ← | ← | ← | ← |-1 | ↑ | ↑ | ↑ | ↑ | ← | ← | ↑ |    ↑  - Optimal Policy
3 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ |###| ↑ | ↑ | ← | ← |###| ↑ | ↑ | ↓ | ← |###| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |    Press 'q' to quit
4 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Press 'o' for optimal move
  | ↓ | ↓ |###| ↓ | ↓ | ↓ | ← |###| ↓ | ↓ | ↓ | ← |###| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |    Press 'b' to return to menu
5 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Use arrow keys to move manually
  | ↓ | ↓ | ← | ← | ← |+1 | ↑ | ← | ← | → |+1 |###|###| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
6 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+    Current state: (7, 7)
  | ↓ | ← | ← | ← | ← | ← |-1 |###|-1 | → | ↑ | ← | ← | ← | ↑ | ↑ | ← | ← | ← | ← |    Policy: UP
7 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |+1 | ← | ← |###|###| ↑ | ↓ | A | ← | ← | ← | ← | ← | ↓ | ← | ↓ | ↓ | ← | ← | ↓ |
8 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |###| ↑ | ← | ← | ← |###| ↓ | ↑ | ← | ← |###| ↑ | → | ↓ |###| ↓ | ↓ | ← | ← | ↓ |
9 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |###| ↑ | ↑ | ← |-1 |###| ↓ | ← | ← |-1 |###| → | → | ↓ |###| ↓ | ← | ← | ← | ← |
10+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | → | ↑ | ↑ | ↑ | ← |+1 | ← | ← | ← | ← | ← |###|###| → |+1 | ← | ← | ← | ← | ← |
11+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ← | ← | ↑ |###|###| ↑ | ← | ← | ← | → | → | ↑ | ← | ← | ← | ← | ← |
12+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ |###|###| ↑ | ← | ← |###| ↑ | ↑ | ↑ | ↑ |###| ↑ | ↑ | ← | ← | ← | ← |
13+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ |-1 | ↑ |###| → | ↑ | ← | ← |###| ↑ | ↑ | ↑ | ← |-1 | → | ↑ | ↑ | ↑ | ↑ | ↑ |
14+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |+1 | ← | ↑ | ← | ← | ↑ |###|###| → | ↑ | ↑ | ↑ | ↑ | ↓ |+1 | ↑ | ↑ | ↑ | ↑ | ↑ |
15+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ← | ↑ | ↑ | ← | ← | ← | ↑ | ↑ | ↑ | ↑ | → | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
16+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ← | ← | ← | ← | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
17+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ← | ← | ← | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
18+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ← | ← | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
19+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ← | ← | ← | ← | ← | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

2. How does the number of states and the complexity of the environment affect convergence?

Maps	Number of States	Total Value Iteration before convergence	Total Policy Iteration before convergence
6x6	36	757	6
10x10	100	743	7
15x15	225	742	10
20x20	400	741	10



Observation:

- For both Value Iteration and Policy Iteration, the amount of iteration required before convergence remains quite stable even as the map size increases. Policy Iteration constantly outperform Value Iteration in terms of computationally efficiency. I also observed that value iteration reaches the optimal policy way before the convergence, and the rest of the computations are wasteful, and that's why Policy Iteration is preferable in this context.

3. How complex can you make the environment and still be able to learn the right policy?

Observation:

- I have managed to output the map up to the size 20x20. Beyond that the algorithm takes very long to run, especially for Value Iteration. This is due to their respective complexities. Value iteration runs in O(S^2 * A) whereas policy iteration runs in O(S^2) while computing the values. S is the number of states and A is the number of possible actions.