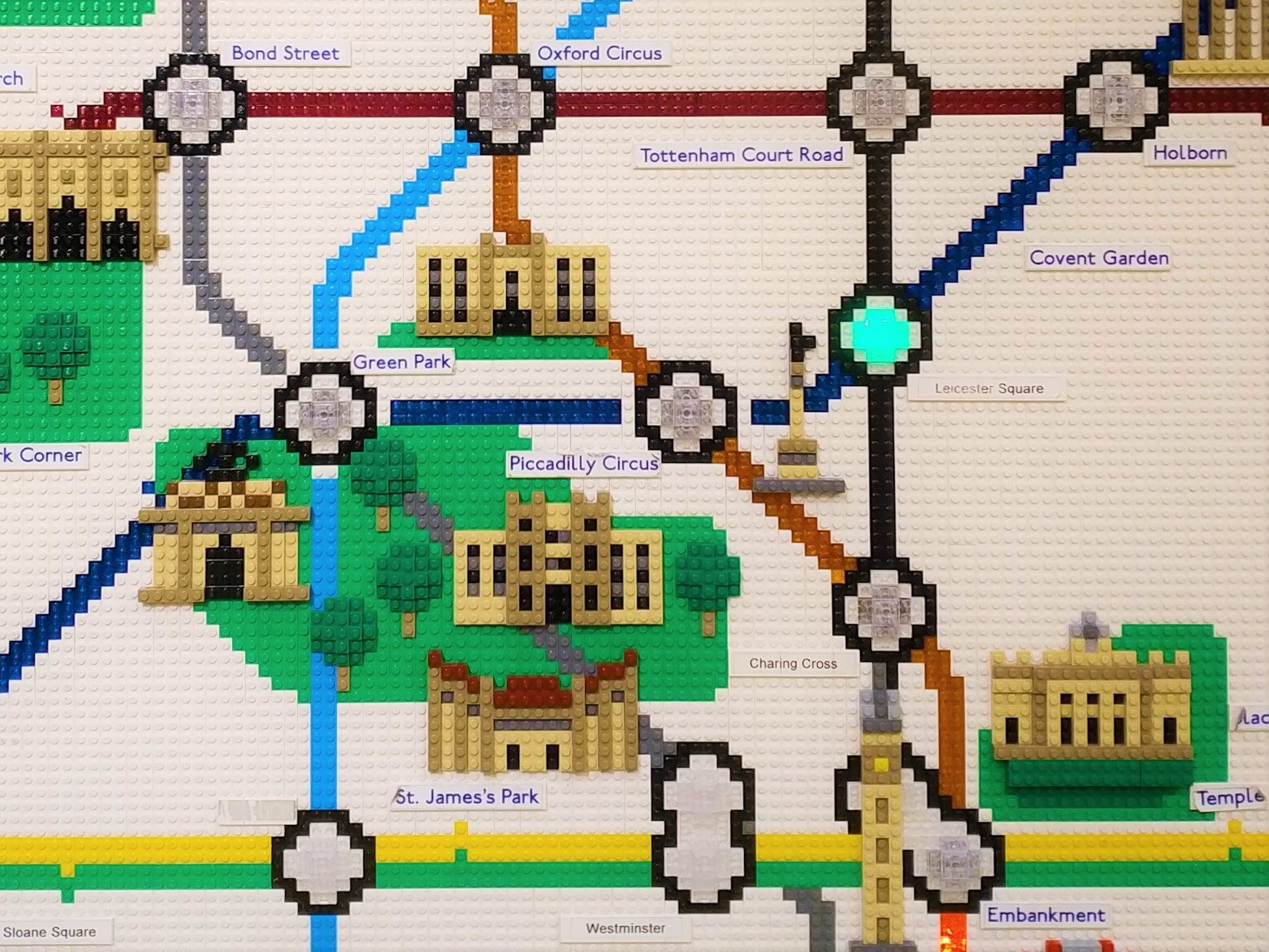
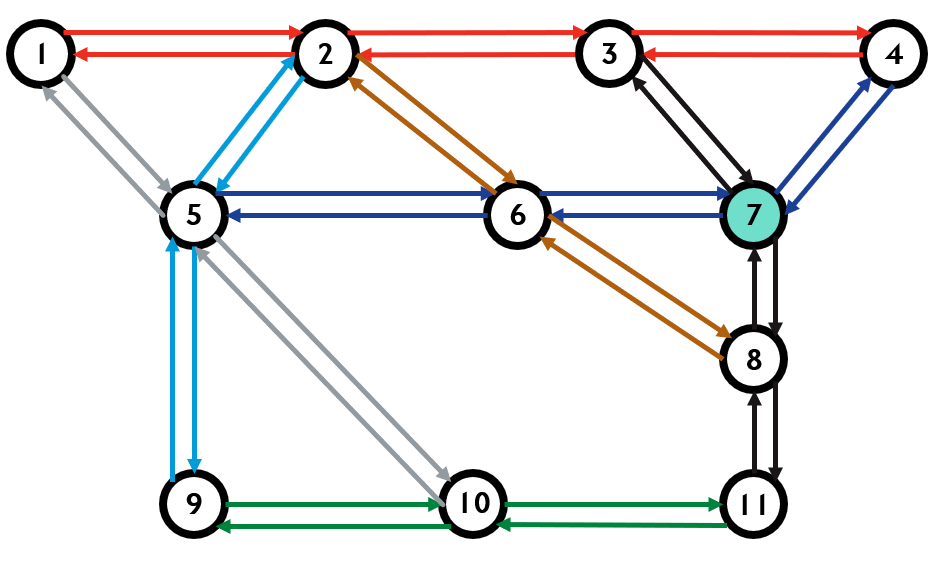
**Colour key:**

XL - green

HL - blue

**Lego Finding Home**





**Basic:**

1. Domain and Task:
   1. Navigation episodic task
   2. Find the shortest path to home (Lego Store @ Leicester Square [7]) from a random selected station at each episode
   3. TD Q-learning

1. State transition function

|  |  |
| --- | --- |
| State (s) | Next State (s’) |
| 1 | 1,2,5 |
| 2 | 1,2,3,5,6 |
| 3 | 2,4,7 |
| 4 | 3,4,7 |
| 5 | 1,2,6,9,10 |
| 6 | 2,5,7,8 |
| 7 | 3,4,6,7,8 |
| 8 | 6,7,11 |
| 9 | 5,9,10 |
| 10 | 5,9,10,11 |
| 11 | 8,10,11 |

1. Reward function
   1. The use of reward signal to formalise the idea of a goal is one of the foundations of reinforcement learning
   2. The reward signal is the way to communicate to the agent regarding what to achieve
   3. In general, the goal of agent is to maximum the cumulative rewards receive in the long run. The maximisation of the expected return, which is the sum of the rewards followed by each action:
   4. The expected return lives in two forms: State-Value Function (v), and Action-Value Function (q)
   5. In the context of Q-Learning, the reward is the immediate return by taking an action plus the maximum of the reward by taking the next action.
   6. In our case, we gives -1 reward for taking each step, and 100 reward by reaching the destination. Thus the agent is incentivised to take the shortest route. We also give -2 reward for the disconnected route to prevent agent takes that route.

1. Policy
   1. Policy is a mapping from states to action
   2. On-policy vs off-policy
   3. Off-policy TD control, Q-learning (policy evaluation)
   4. Greedy learning (policy improvement)

1. R-matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Action | | | | | | | | | | | |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| State | 1 | -1 | -1 | -2 | -2 | -1 | -2 | -2 | -2 | -2 | -2 | -2 |
| 2 | -1 | -1 | -1 | -2 | -1 | -1 | -2 | -2 | -2 | -2 | -2 |
| 3 | -2 | -1 | -1 | -1 | -2 | -2 | 100 | -2 | -2 | -2 | -2 |
| 4 | -2 | -2 | -1 | -1 | -2 | -2 | 100 | -2 | -2 | -2 | -2 |
| 5 | -1 | -1 | -2 | -2 | -1 | -1 | -2 | -2 | -1 | -1 | -2 |
| 6 | -2 | -1 | -2 | -2 | -1 | -1 | 100 | -1 | -2 | -2 | -2 |
| 7 | -2 | -2 | -1 | -1 | -2 | -1 | 100 | -1 | -2 | -2 | -2 |
| 8 | -2 | -2 | -2 | -2 | -2 | -1 | 100 | -1 | -2 | -2 | -1 |
| 9 | -2 | -2 | -2 | -2 | -1 | -2 | -2 | -2 | -1 | -1 | -2 |
| 10 | -2 | -2 | -2 | -2 | -1 | -2 | -2 | -2 | -1 | -1 | -1 |
| 11 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -1 | -2 | -1 | -1 |

1. Parameters for Q-learning
   1. Discount rate (uncertainty): set 0.8 for illustration
   2. Learning rate (how quick to converge), why we need learning rate? set 1 for illustration (0-1)
   3. Exploration factor

1. Q-matrix update
   1. Initial Q-matrix to 0
   2. Start from state 2 for this episode

Step 1:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **11** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Step 2

Q(2,3) = Q(2,3) + 1\*[R(2,3)+0.8\*max((3,4),(3,7))-Q(2,3)]=0+1\*[-1+0.8\*max(0,100)-0]=78 (not sure if it’s correct, need to review)

Q(2,6) is the same as above

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 78 | 0 | 0 | 78 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **11** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Step 3

Q(3,7)=99+0.8\*0

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 78 | 0 | 0 | 78 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **11** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Step 4

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 78 | 0 | 0 | 78 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **10** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **11** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Some ideas for advance section:

1. Random start and end state
2. Increase the size of the map with function approximation
3. Reflect the difference in speed of tube
4. Give penalty for change the line
5. Compare MC and Q-learning
6. Compare SARSA and Q-learning
7. Try GLIE (Greedy in the limit of infinite exploration, decay ε)