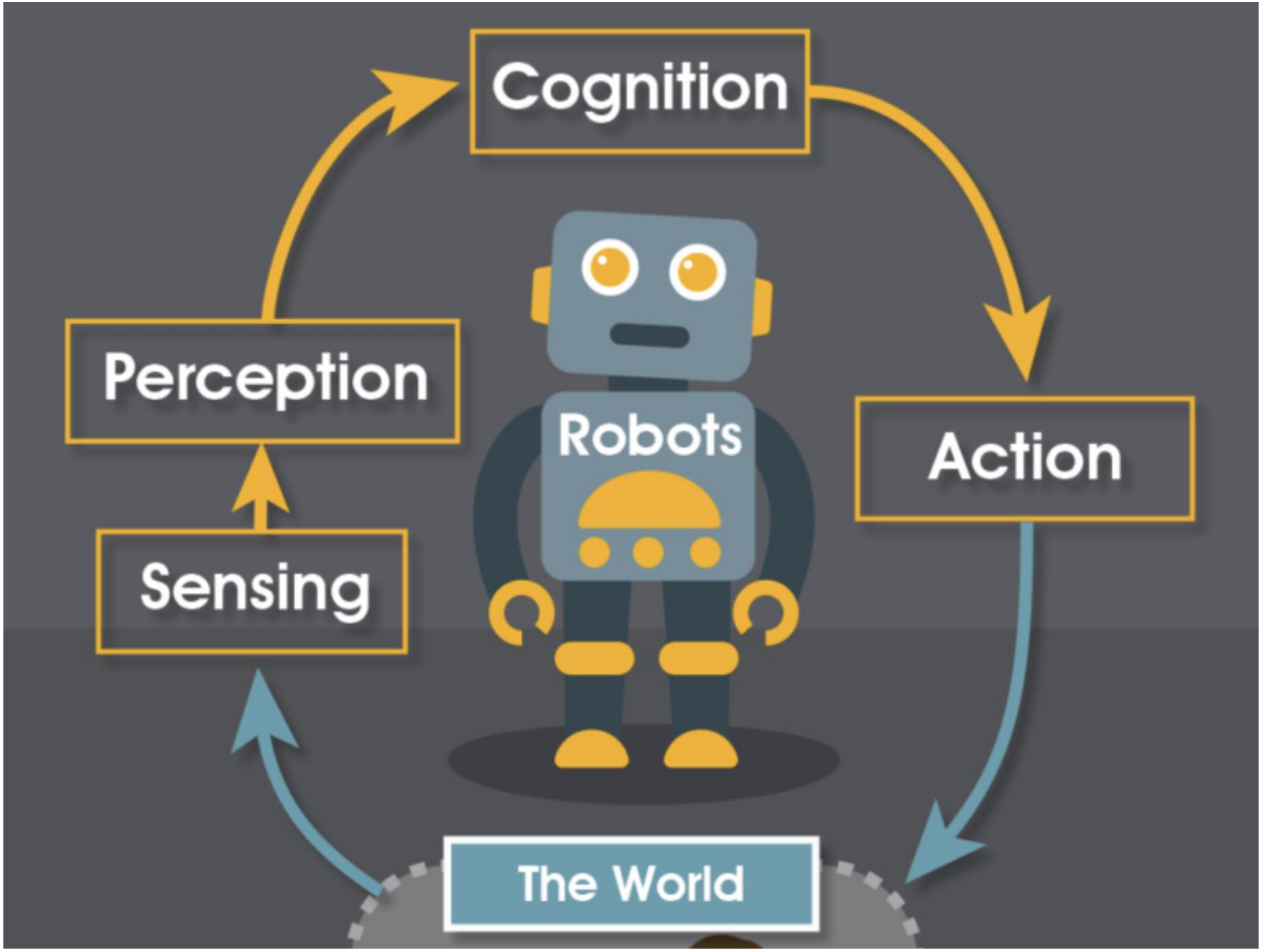
REINFORCEMENT LEARNING

AM – Rana Muhammad Saad
Lab: Autonomous & Decision Support

What makes a robot a robot and not just another machine?

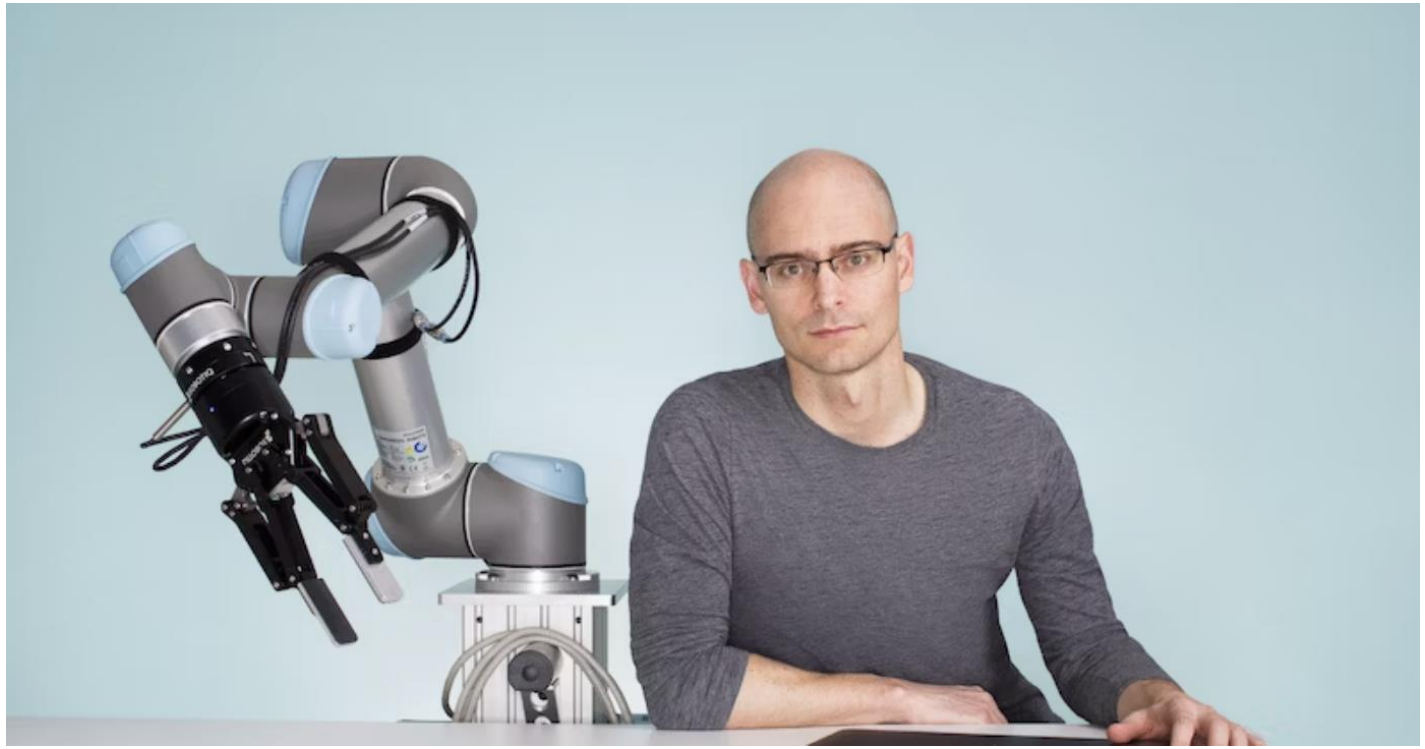
- **Perception** → Perceive the environment using sensors
- **Cognition** → Make plans using algorithms implemented in computer programs
- **Action** → Perform actions enabled by actuators

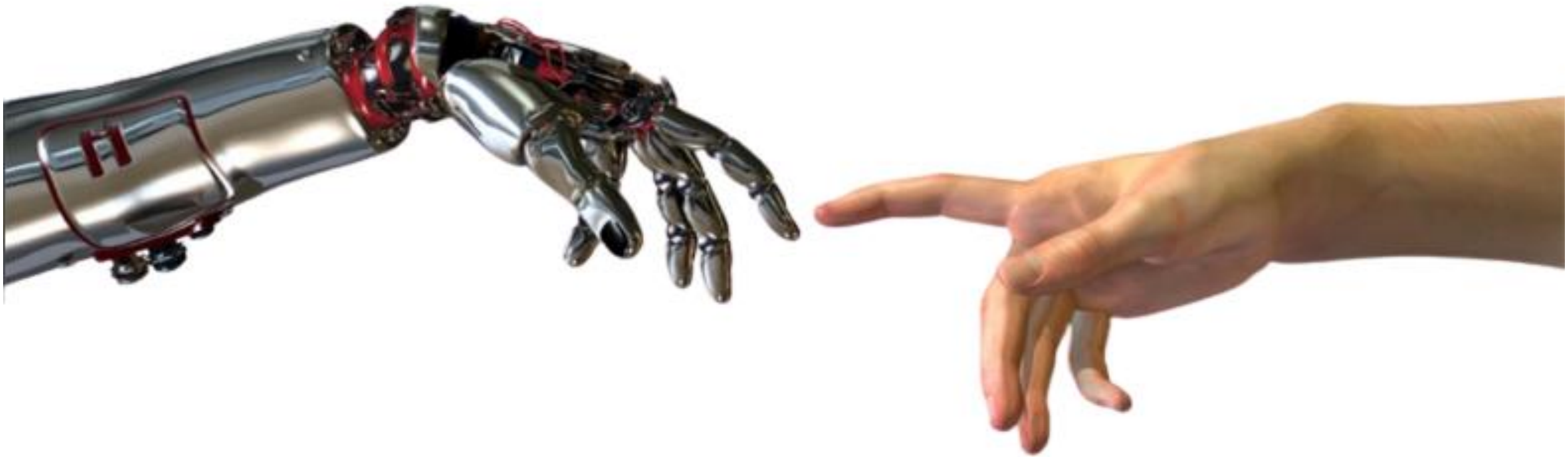

Cognition
Perception
Robots
Action
Sensing
The World


Intelligence

"At this stage, we have enough capable hardware, but we need software intelligence"

--Pieter Abbeel (2017)





Hardware Demo

Petoi Bittle

Unitree Go 1

Which Intelligence

1. Supervised Learning.

2. Unsupervised Learning.

But do we have data for training?

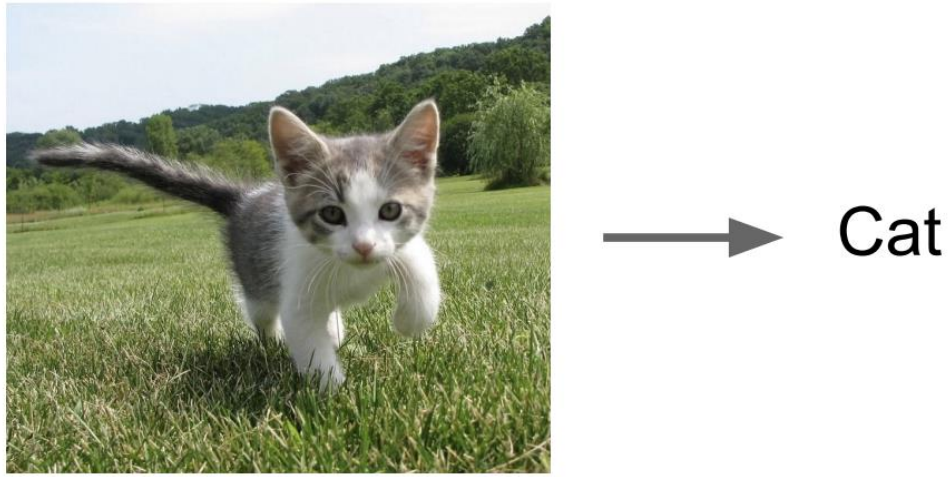
3. Reinforcement Learning

Supervised Learning

Data: (x, y) x is data, y is label

Goal: Learn a function to map x → y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

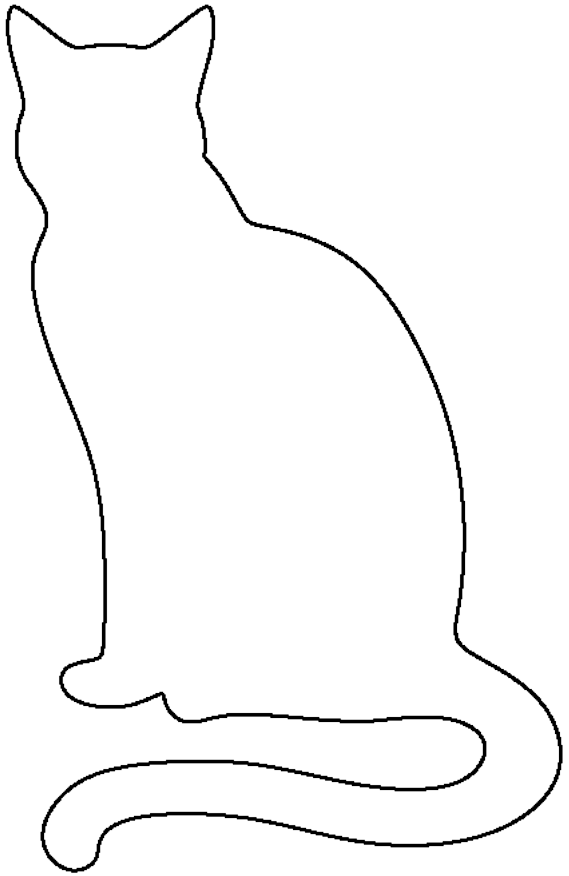


Unsupervised Learning

Data: x Just data, no labels!

Goal: Learn some underlying hidden structure of the data

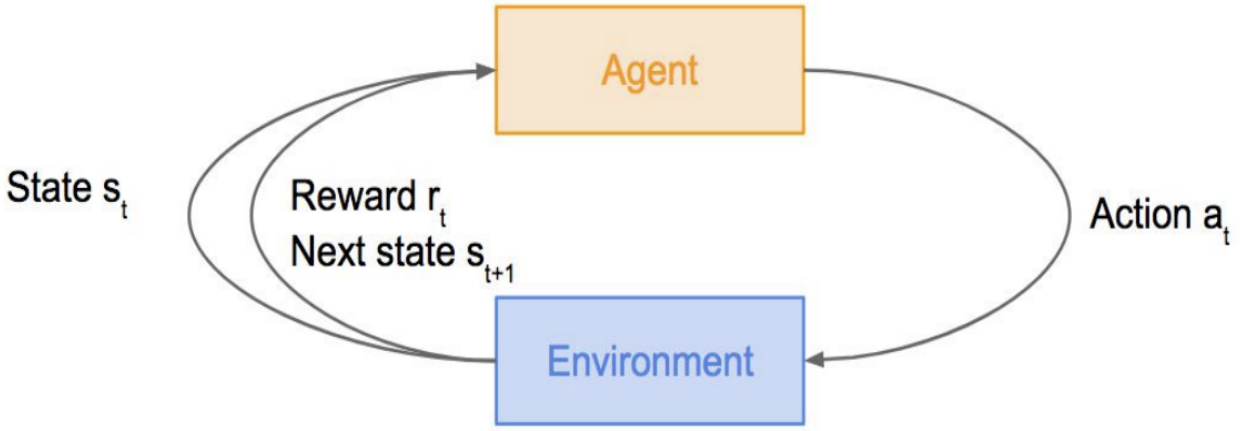
Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Reinforcement Learning

Problems involving an agent interacting with an environment, which provides numeric reward

Goal: Learn how to take actions in order to maximize reward



TODAY'S FOCUS →

Reinforcement Learning

Data: state-action pairs

Goal: Maximize future rewards over many time steps

Apple example:

Eat this thing because it will keep you alive.

Reinforcement Learning (RL): Key Concepts

AGENT

Agent: takes actions.

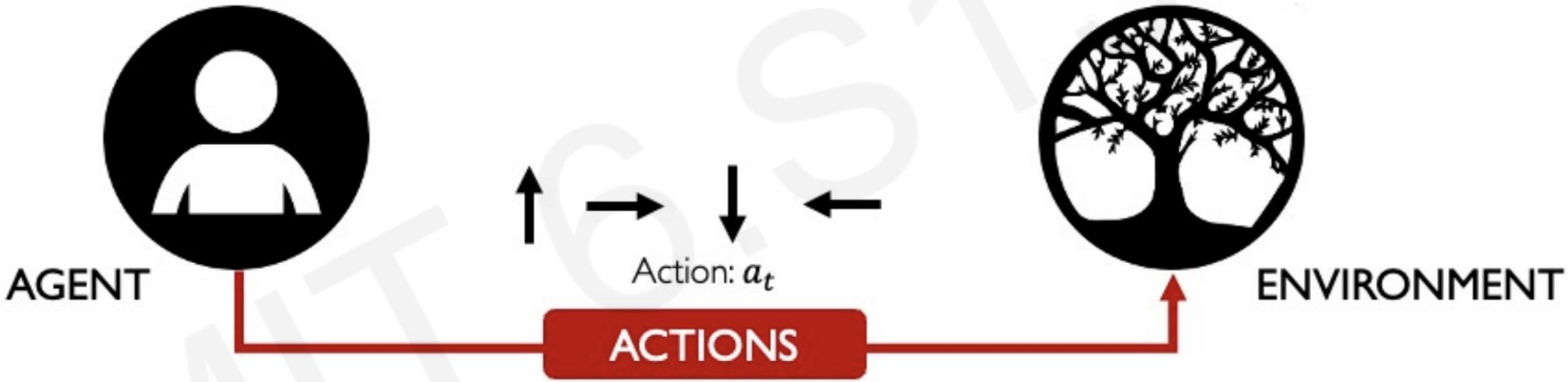
Reinforcement Learning (RL): Key Concepts

AGENT

ENVIRONMENT

Environment: the world in which the agent exists and operates.

Reinforcement Learning (RL): Key Concepts



Action: a move the agent can make in the environment.

Action space A: the set of possible actions an agent can make in the environment

Reinforcement Learning (RL): Key Concepts

Observations: of the environment after taking actions.

Reinforcement Learning (RL): Key Concepts

State: a situation which the agent perceives.

Reinforcement Learning (RL): Key Concepts

Reward: feedback that measures the success or failure of the agent's action.

Reinforcement Learning (RL): Key Concepts

Total Reward
(Return)

$$R_t = \sum_{i=t}^{\infty} r_i$$

Reinforcement Learning (RL): Key Concepts

Total Reward (Return)

$$R_t = \sum_{i=t}^{\infty} r_i = r_t + r_{t+1} \ldots + r_{t+n} + \cdots$$

Action Space and Observation Space Examples

1. Board Games (e.g., Chess):

Action Space: In chess, the action space consists of all legal moves that a player can make on their turn. This includes moving specific pieces to different positions on the board or capturing opponent pieces.

Observation Space: The observation space typically includes the current state of the chessboard, which consists of the positions and types of all pieces on the board.

Action Space and Observation Space Examples

2. Autonomous Vehicles:

Action Space: For an autonomous car, the action space may involve commands like accelerate, brake, turn left, turn right, and maintain current speed.

Observation Space: The observation space includes data from various sensors such as cameras, lidar, radar, GPS, and vehicle speed, which provide information about the vehicle's surroundings and internal state.

Action Space and Observation Space Examples

3. Robot Manipulation:

Action Space: In a robotic arm tasked with picking and placing objects, the action space includes joint angles or velocities, which control the movement of the arm's joints.

Observation Space: The observation space could include data from cameras, depth sensors, and touch sensors to provide information about the location of objects, the arm's position, and the environment.

Action Space and Observation Space Examples

4. Video Games (e.g., Atari Games):

Action Space: In a video game like Pong, the action space might include commands like move paddle up, move paddle down, or do nothing.

Observation Space: The observation space consists of the pixels on the game screen, which serve as the raw input for the agent.

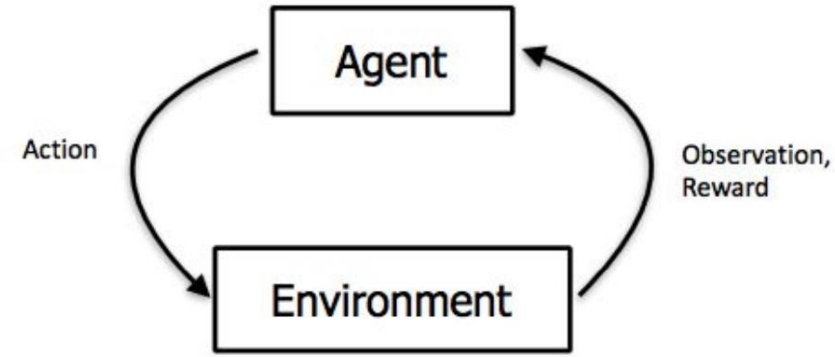
Action Space and Observation Space Examples

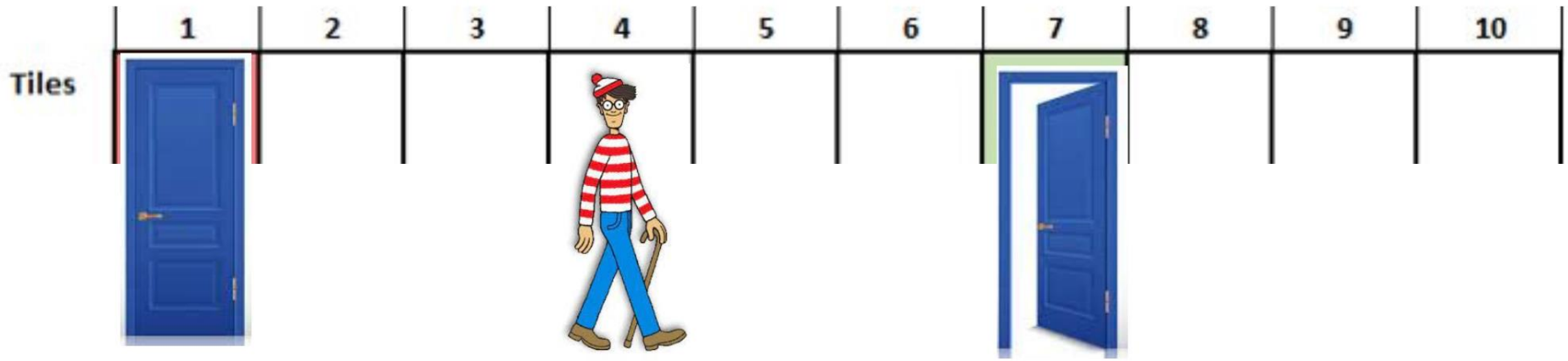
5. Stock Trading:

Action Space: In a financial trading environment, the action space might include decisions to buy, sell, or hold a particular stock.

Observation Space: The observation space typically comprises historical price data, trading volumes, and other financial indicators for the selected stocks.

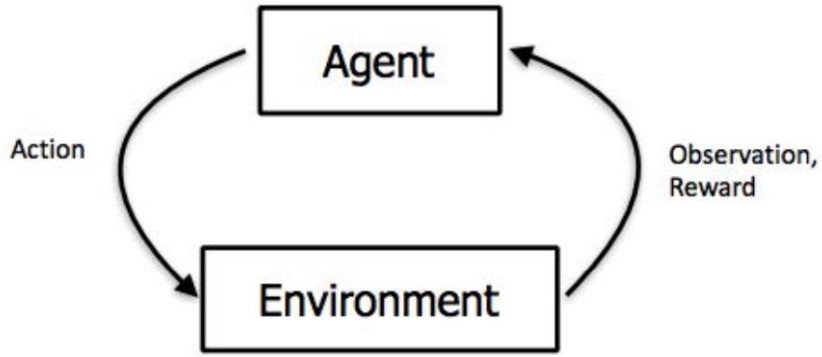
Reinforcement Learning: Delayed Rewards

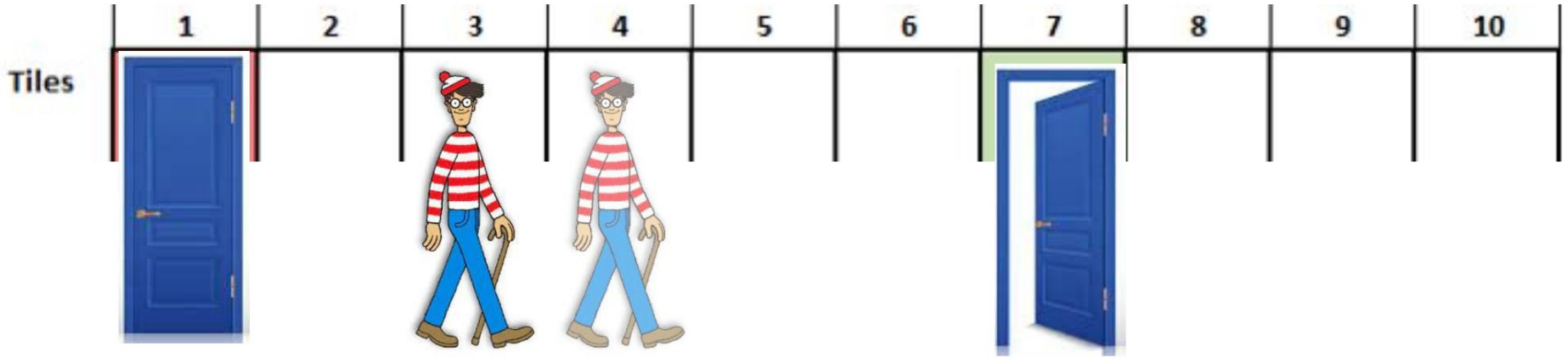




https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

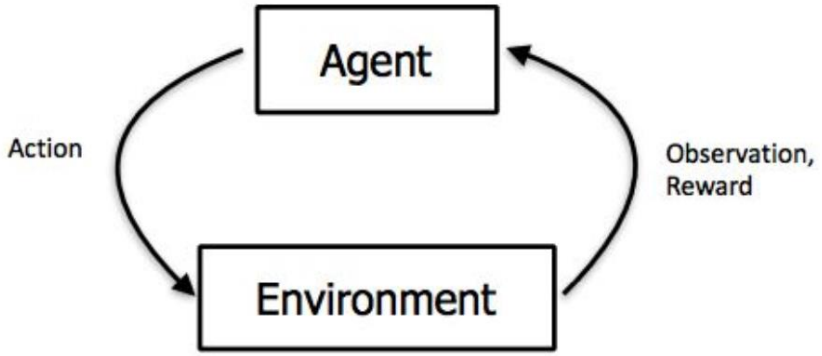
Reinforcement Learning: Delayed Rewards

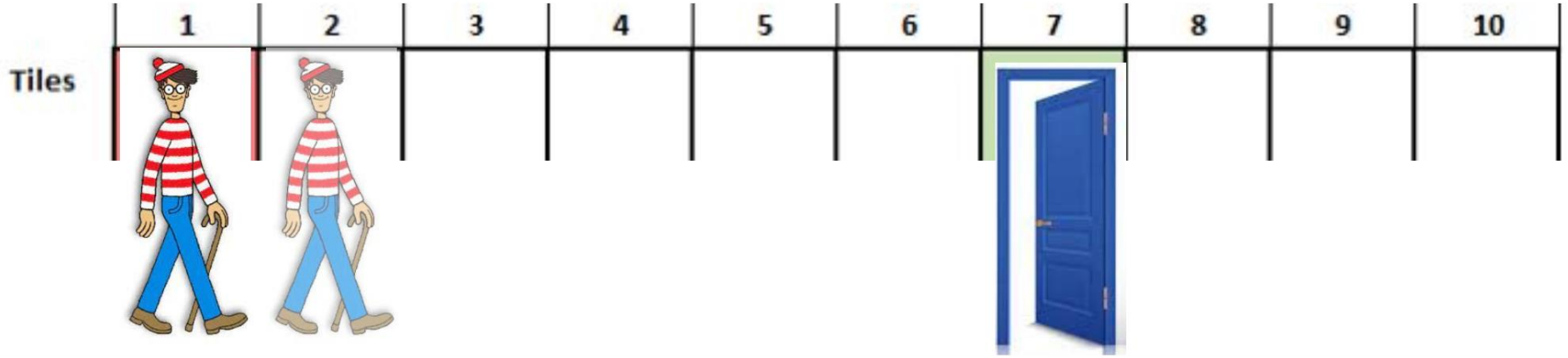


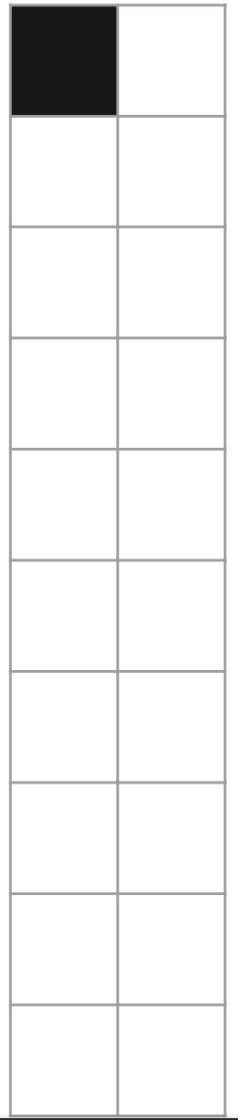


https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

Reinforcement Learning: Delayed Rewards

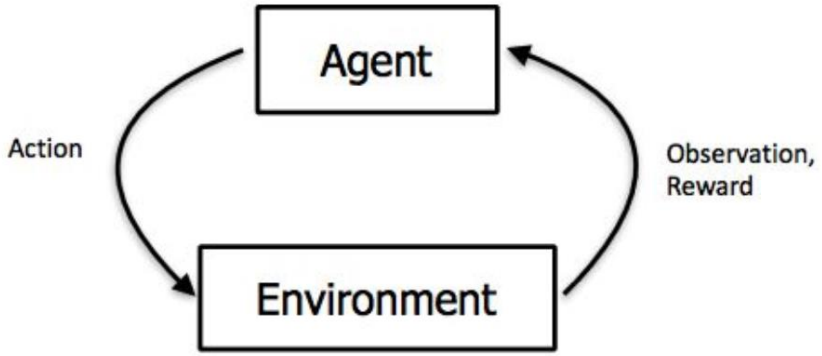


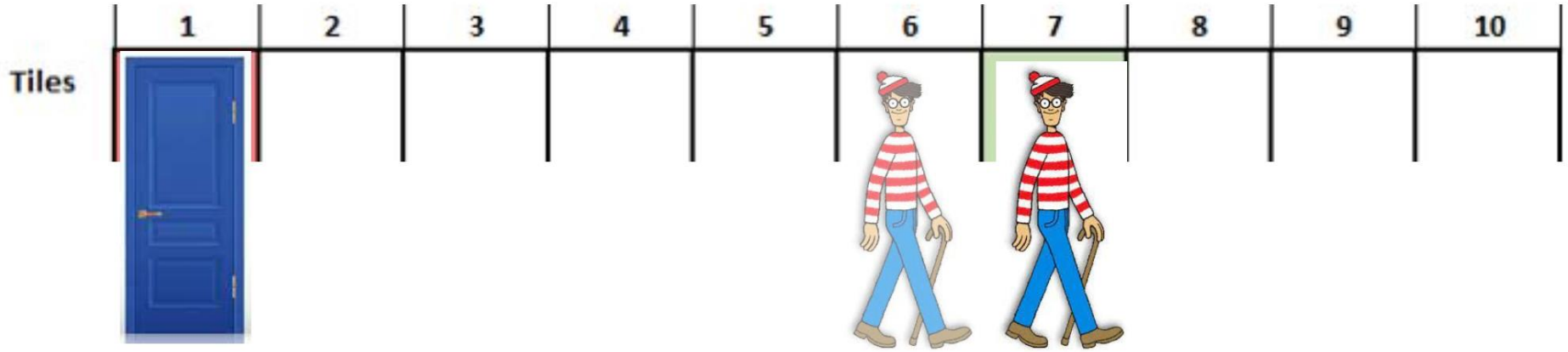




https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

Reinforcement Learning: Delayed Rewards





Policy

	L	R
1		
2	-1	
3		
4		
5		
6		1
7		
8		
9		
10		

https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

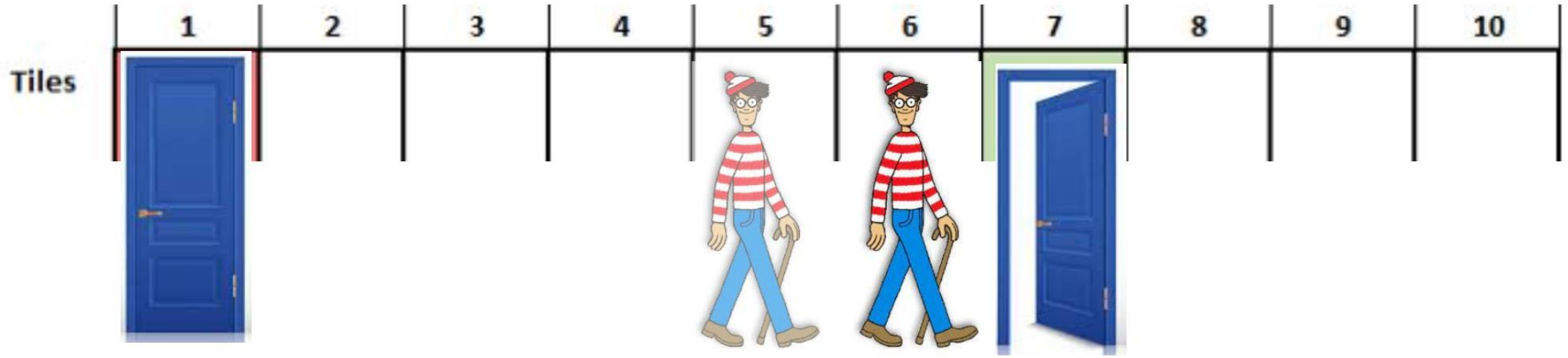
Q Learning: Delayed Rewards

6 L

5 R 0 0.1 6 R

$$Q(s,a) = r + \gamma(max(Q(s',a'))$$

the bellman equation for discounted future rewards



Policy

	L	R
1		
2	-1	
3		
4		
5		
6		1
7		
8		
9		
10		

https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

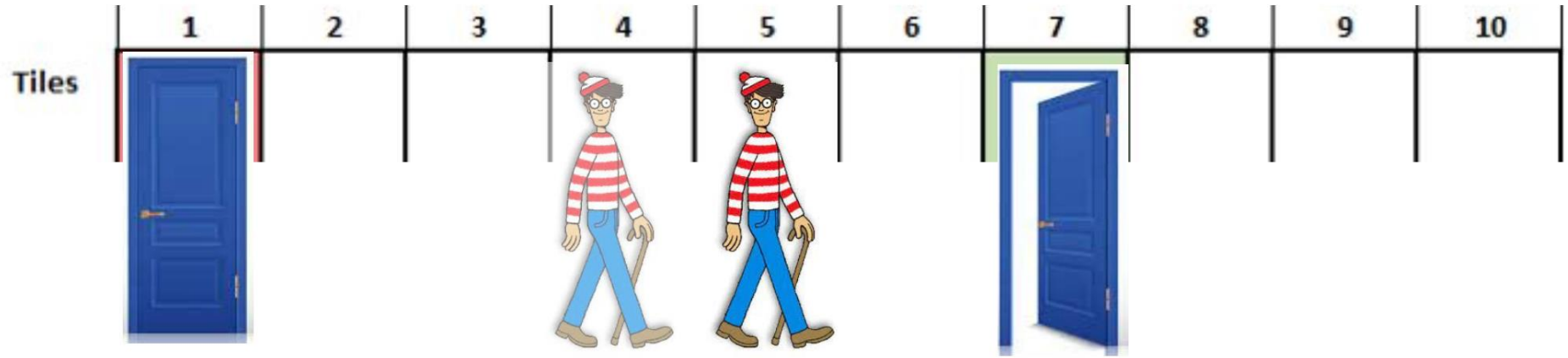
Q Learning: Delayed Rewards

6 L

5 R 0 0.1 6 R

$$Q(s,a) = r + \gamma(max(Q(s',a'))$$

the bellman equation for discounted future rewards



Policy

	L	R
1		
2	-1	
3		
4		
5		
6		1
7		
8		
9		
10		

https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

Q Learning: Delayed Rewards

6 L

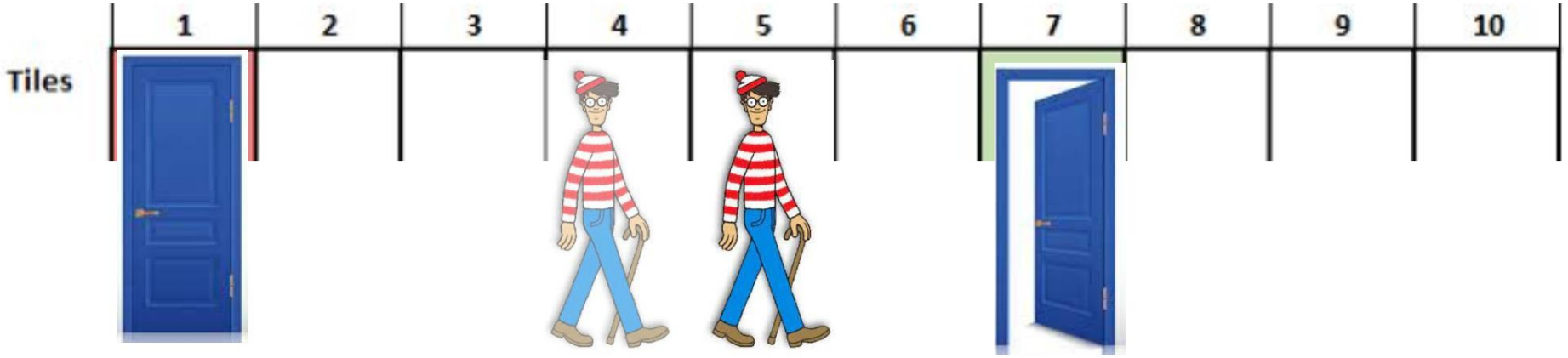
5 R 0 0.1 6 R

$$Q(s,a) = r + \gamma(max(Q(s',a'))$$

the bellman equation for discounted future rewards

After 1000 episodes, Q table some what looks like this

	Action	
State	Left	Right
0	0	0
1	-100	65.61
2	59.049	72.9
3	65.61	81
4	72.9	90
5	81	100
6	0	0
7	100	81
8	90	72.9
9	81	0



https://itnext.io/reinforcement-learning-with-q-tables-5f11168862c8

Role of Discount Factor

Reward on reaching target = 10 points

$\gamma = 0.8$

steps taken = 5

Actual Reward = $0.8^5 \times 10 =$ **3.27 points**

Reward on reaching target = 10 points

$\gamma = 0.8$

steps taken = 3

Actual Reward = $0.8^2 \times 10 =$ **5.12 points**

Role of Discount Factor

Reward on reaching target = 10 points

$\gamma = 0.1$

steps taken = 5

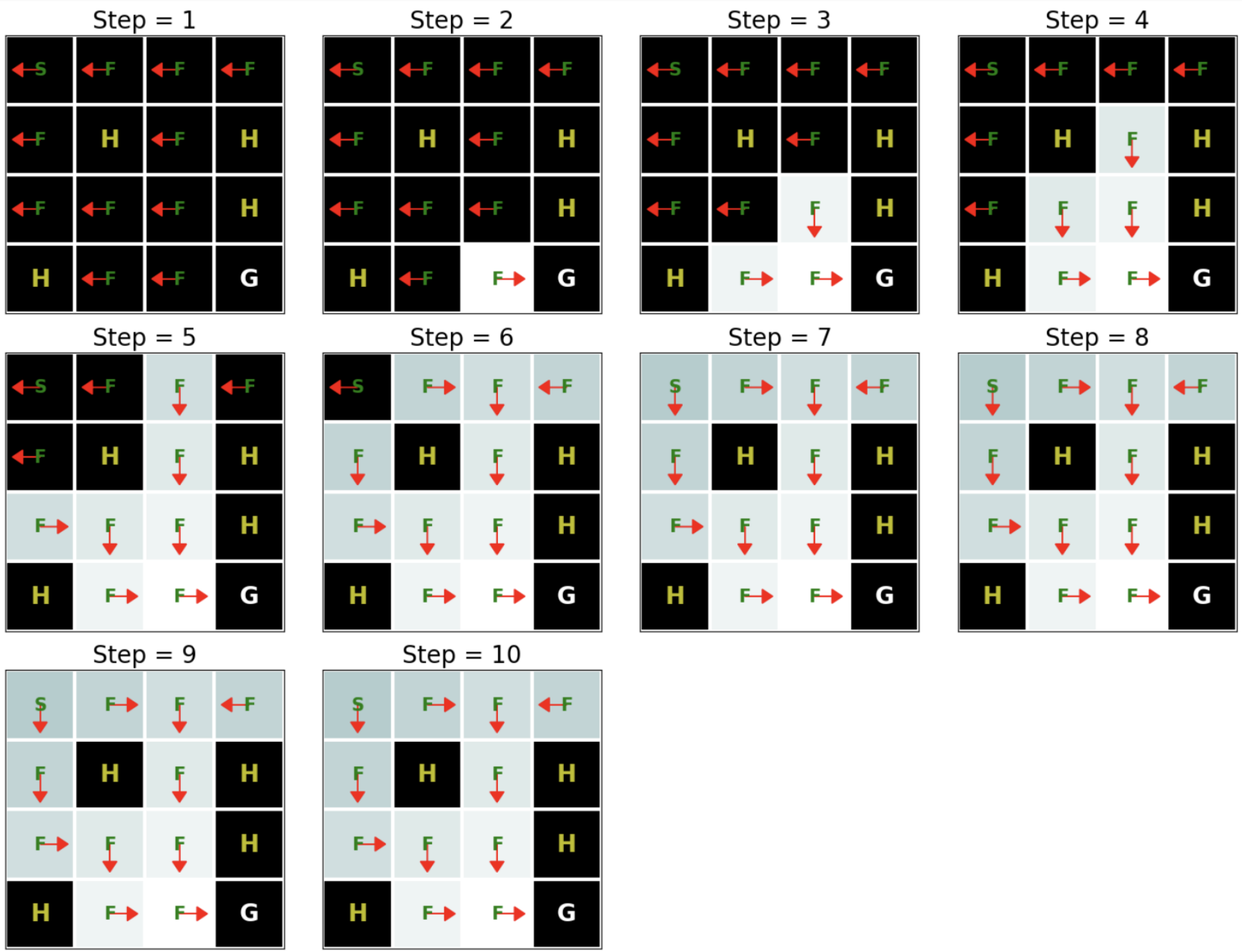
Actual Reward = 0.15 x 10 = **0.0001 points**

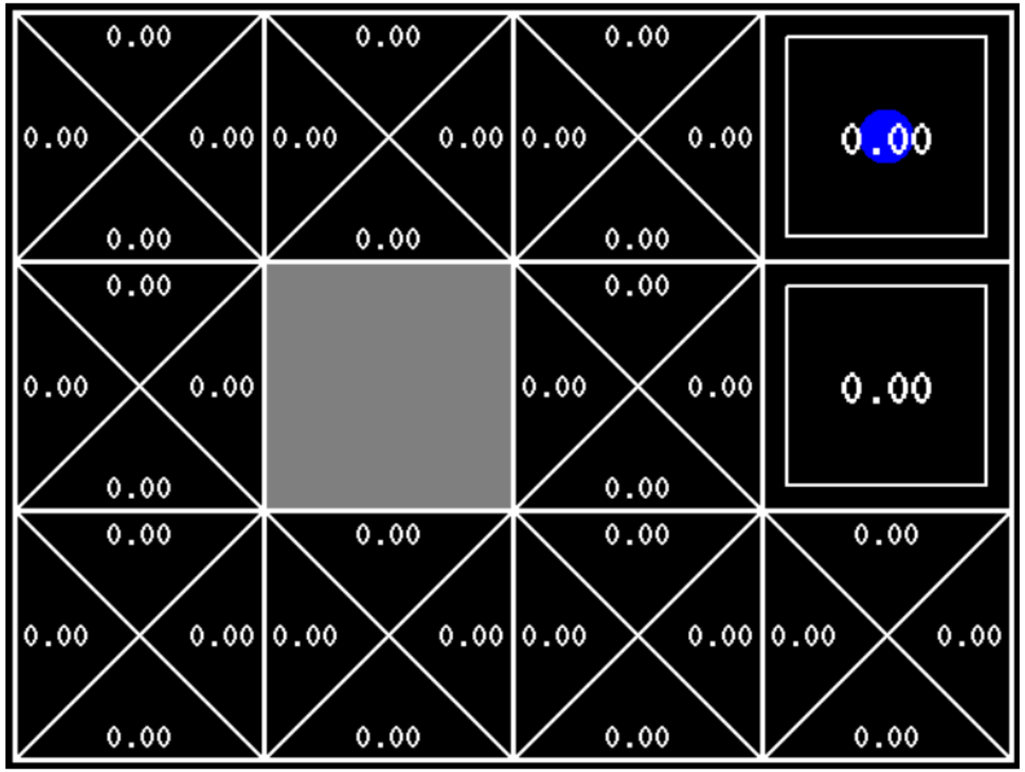
Reward on reaching target = 10 points

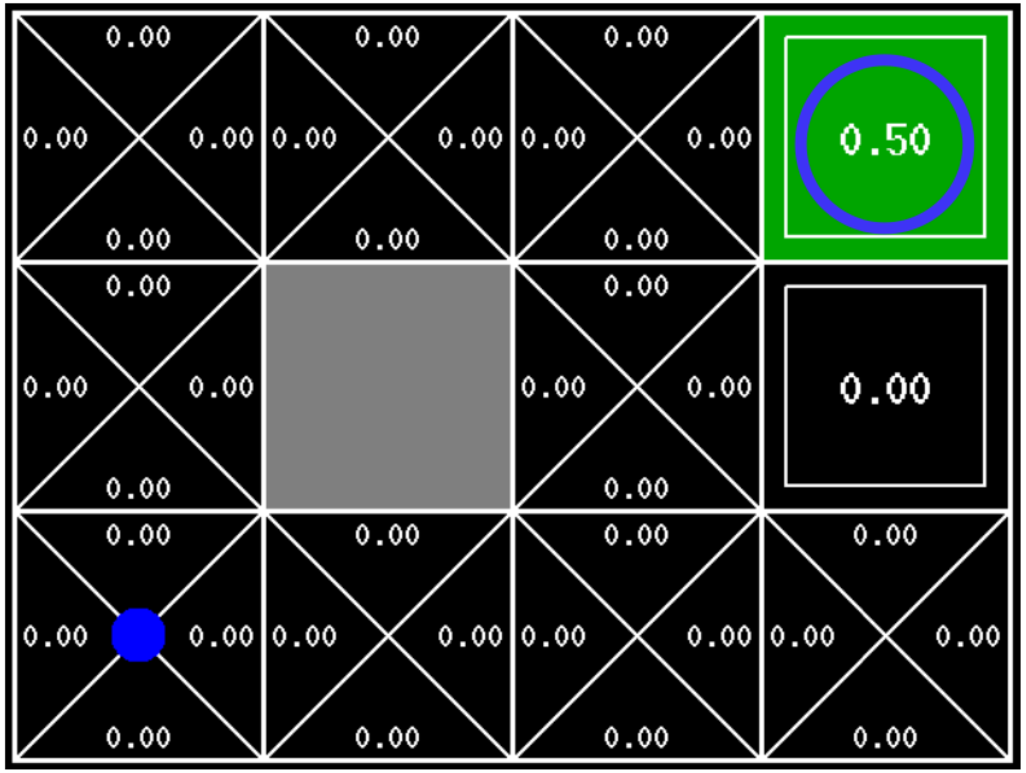
$\gamma = 0.1$

steps taken = 3

Actual Reward = 0.13 x 10 = **0.01 points**


Step = 1
Step = 2
Step = 3
Step = 4
Step = 5
Step = 6
Step = 7
Step = 8
Step = 9
Step = 10

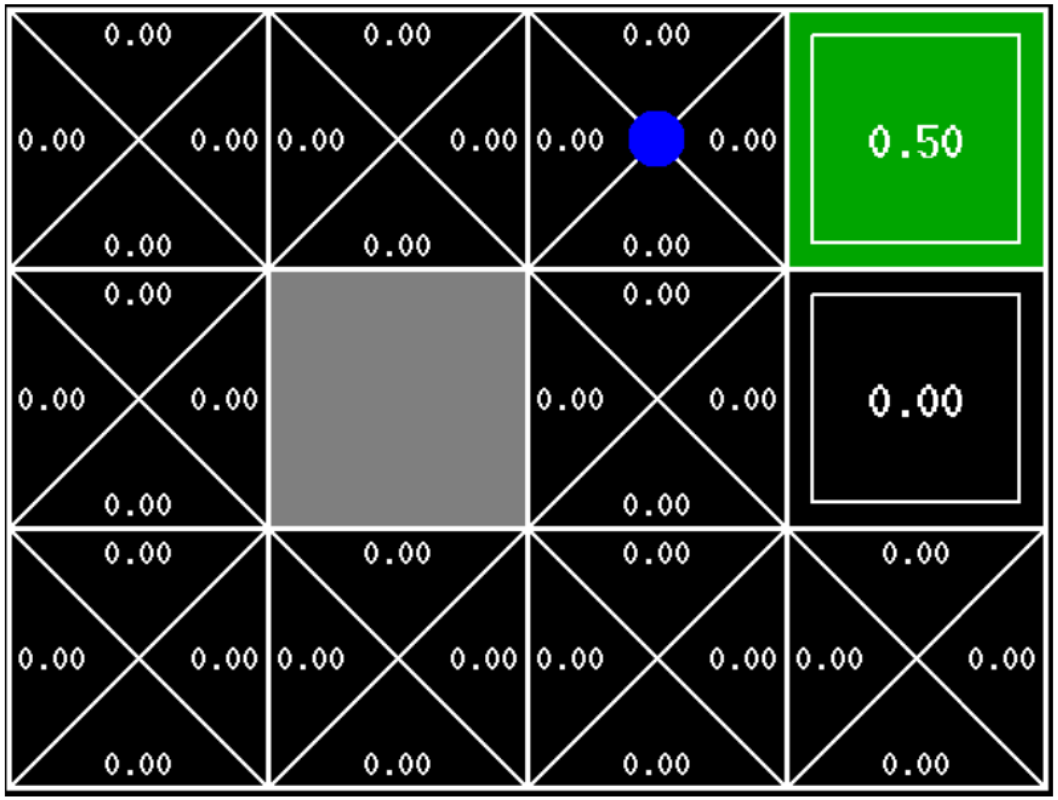


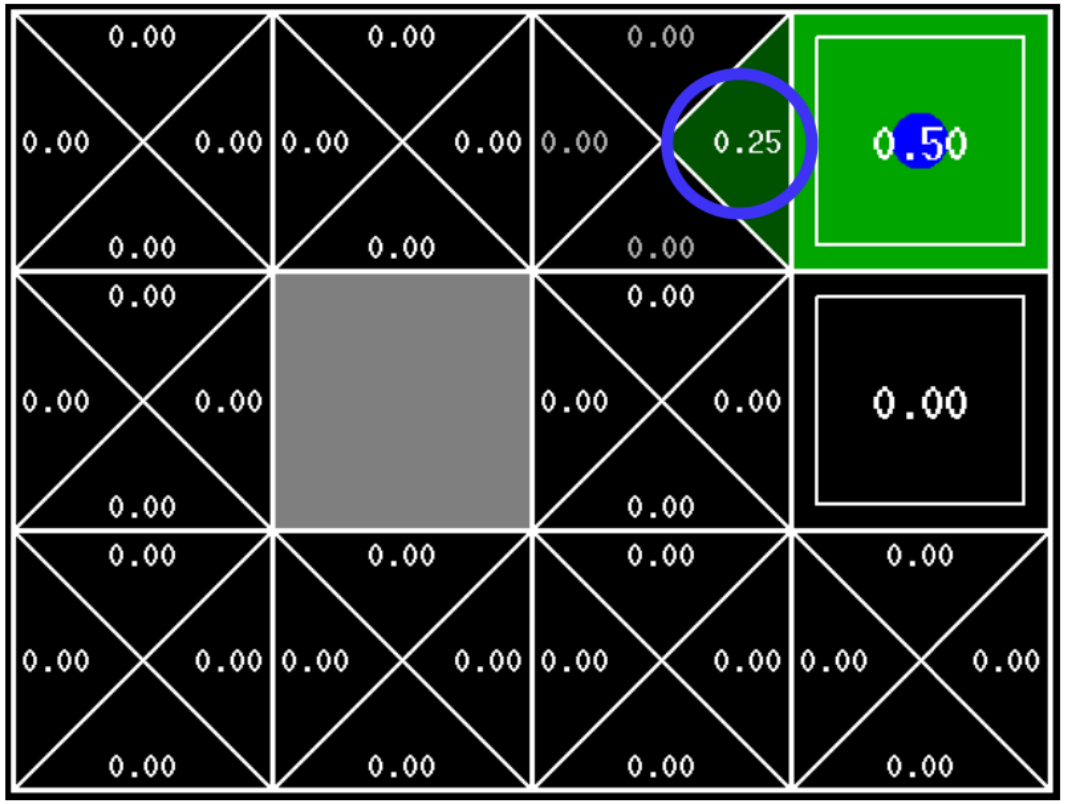


Say γ = 1
and α = 0.5

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

0 0.5 1 1 0 0



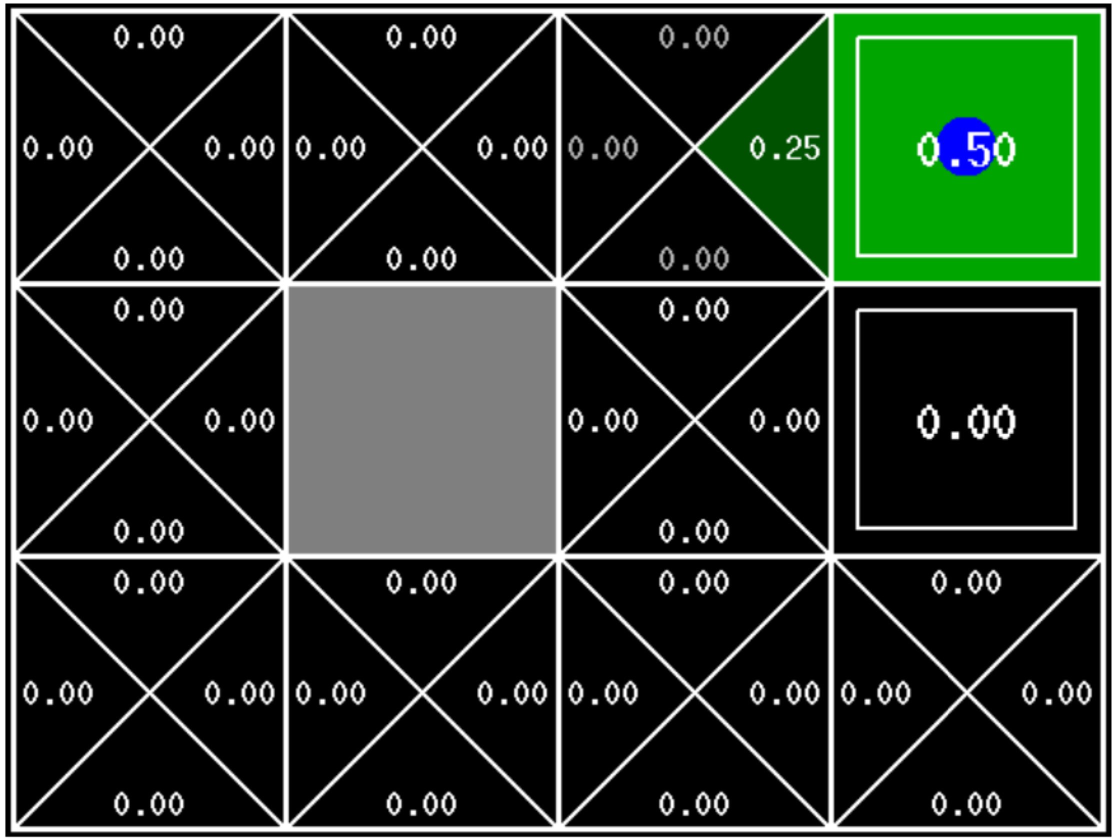


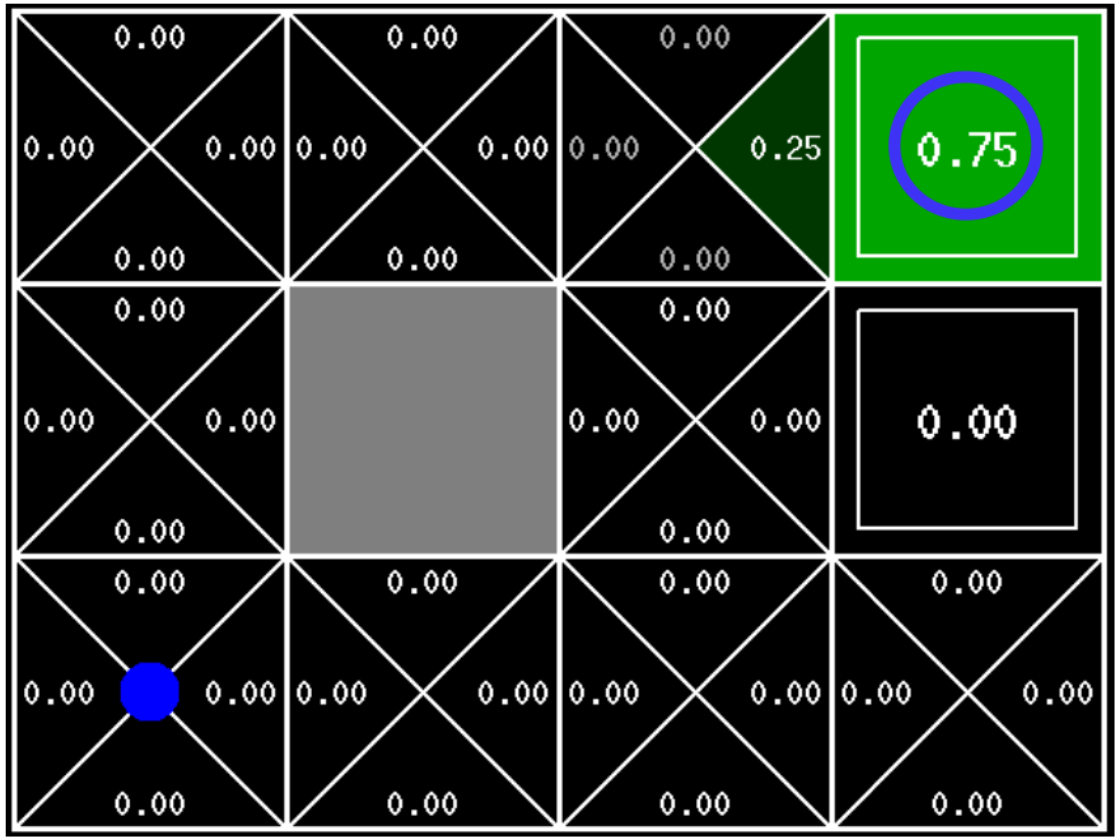
$\gamma = 1$

$\alpha = 0.5$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

0 0.5 0 1 0.5 0

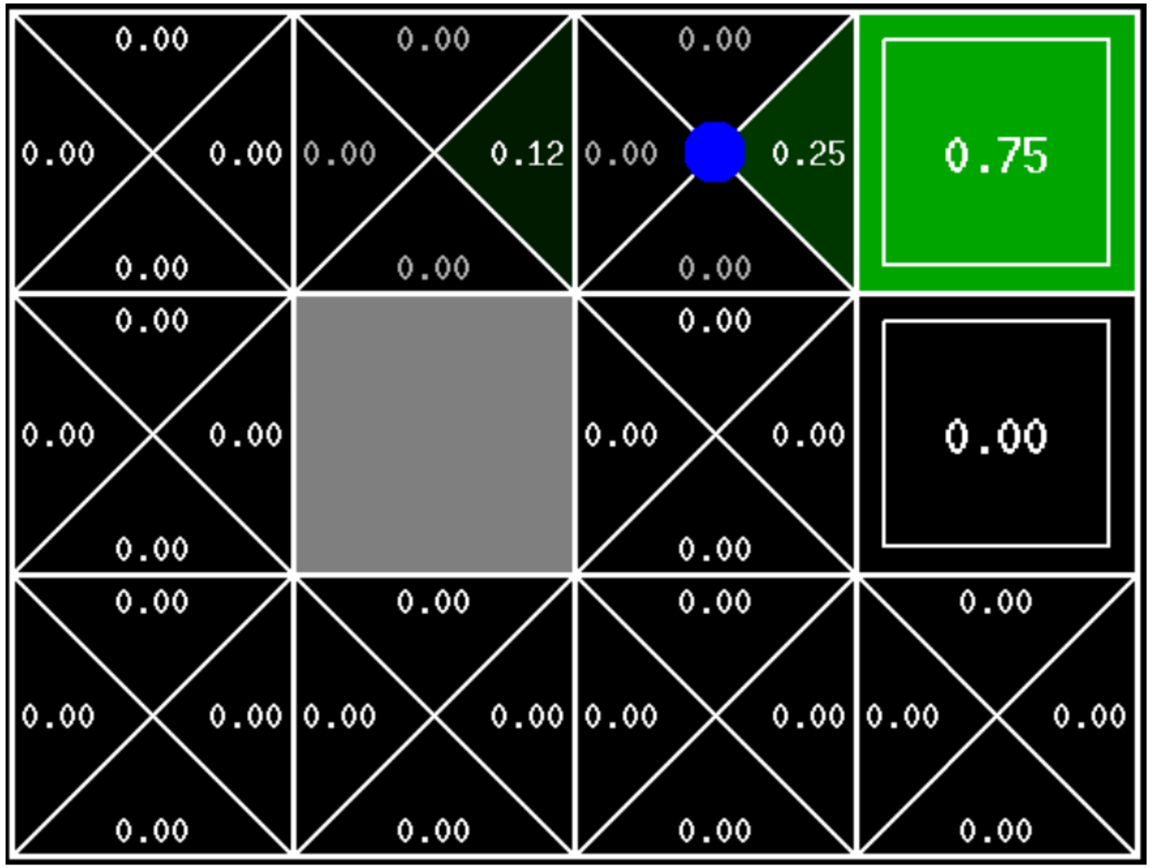


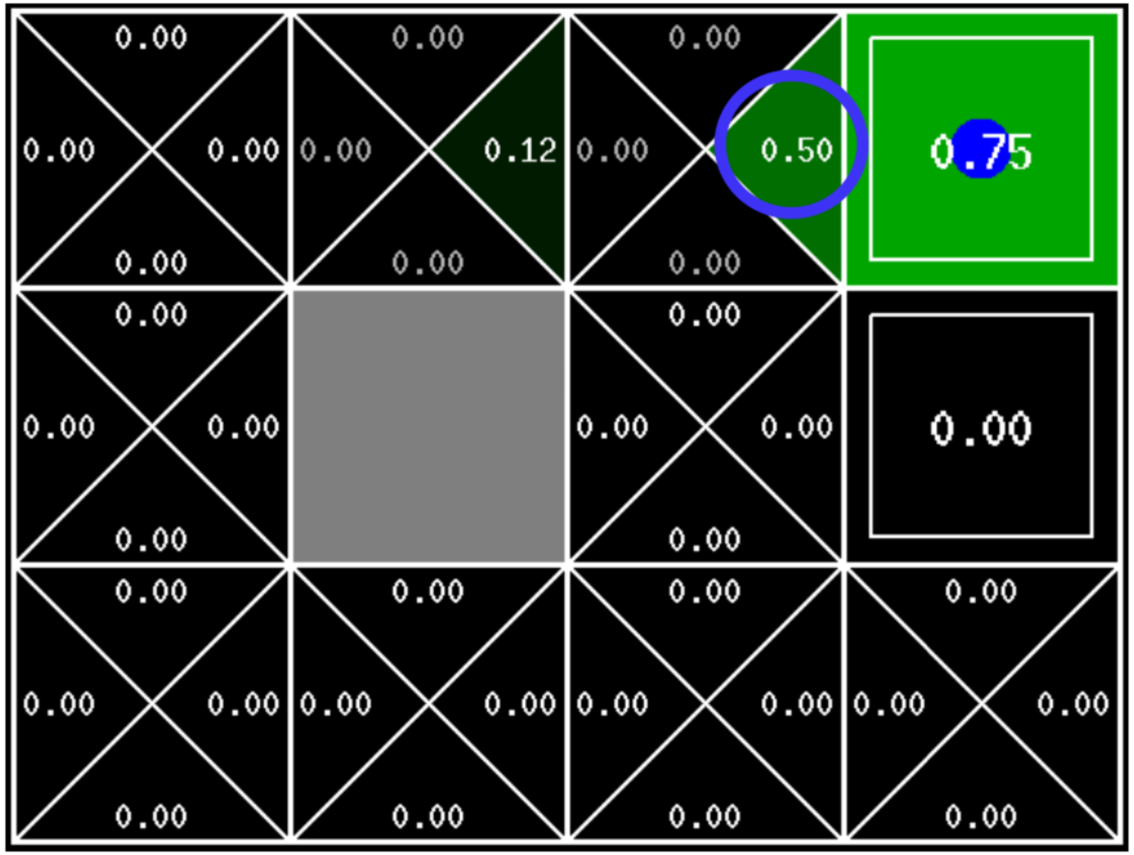


$\gamma = 1$
$\alpha = 0.5$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

0.5 0.5 1 1 0 0.5

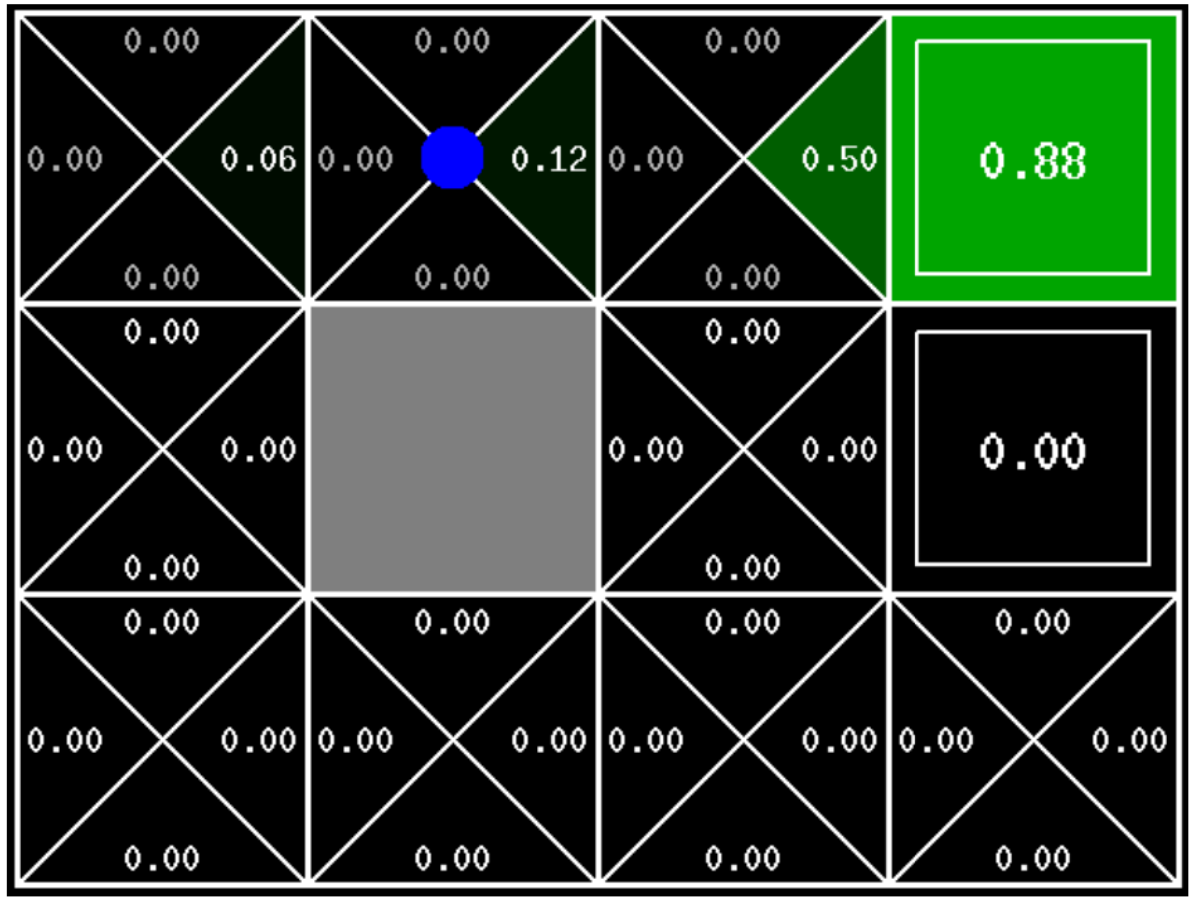


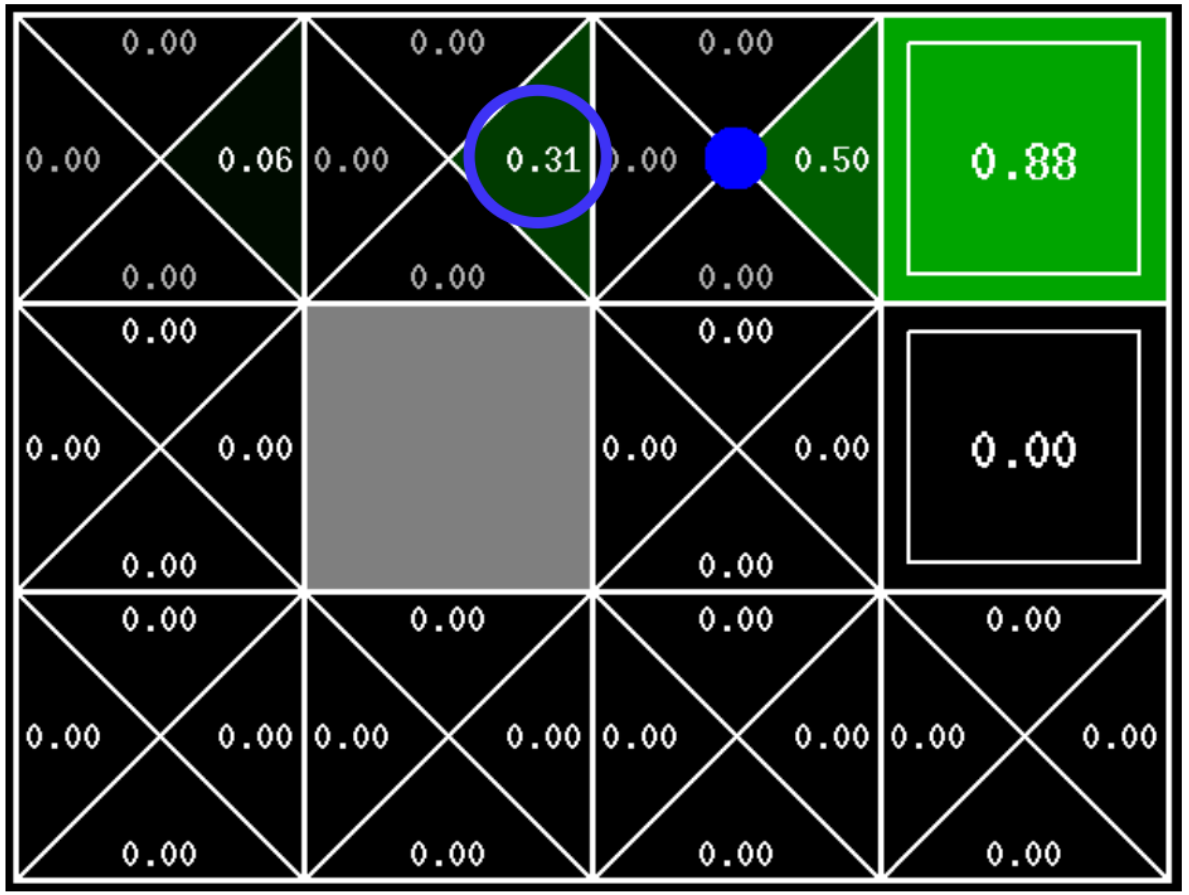


γ = 1
α = 0.5

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

0.25 0.5 0 1 0.75 0.25



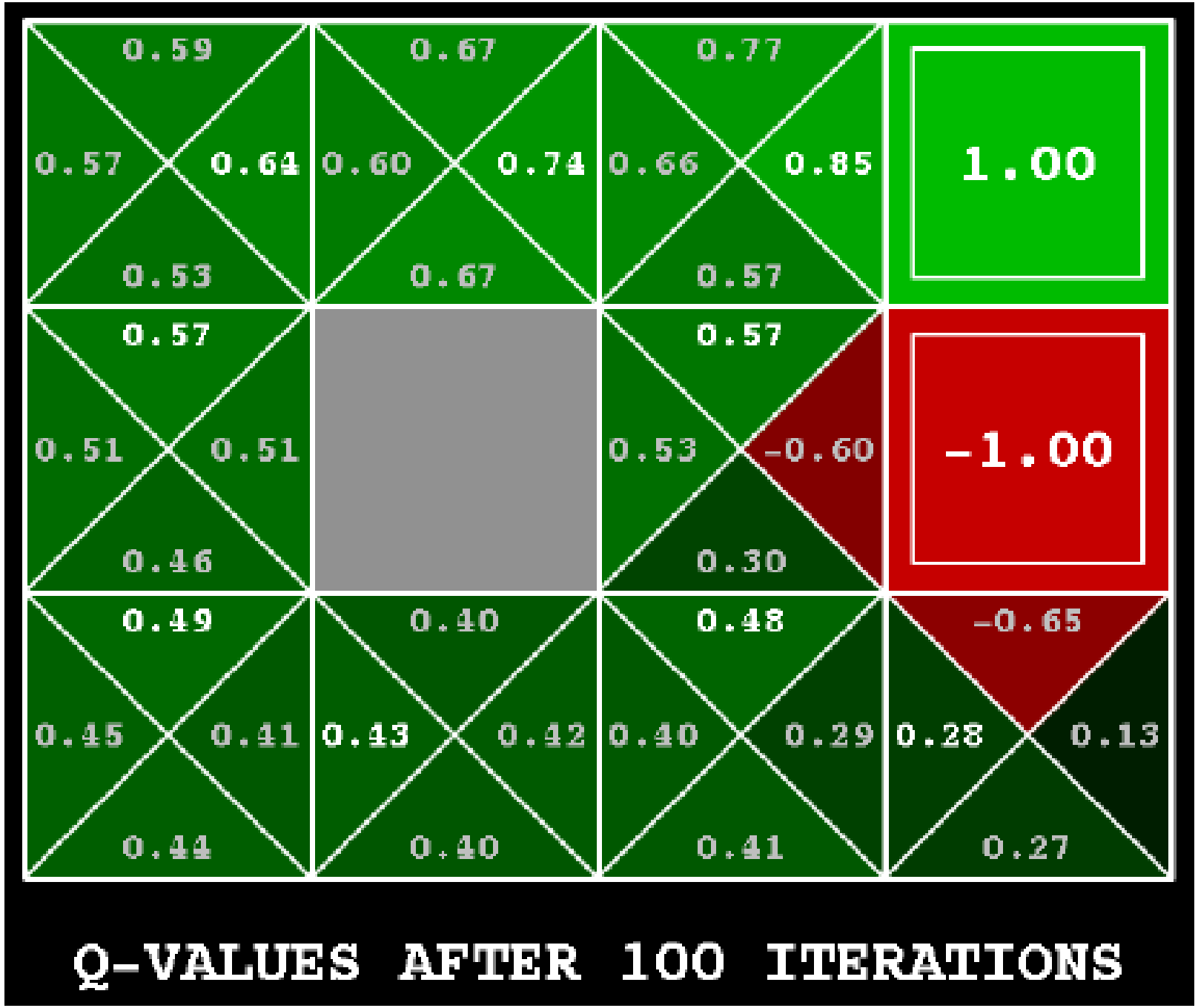


$\gamma = 1$

$\alpha = 0.5$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

0.12 0.5 0 1 0.50 0.12


0.59
0.57
0.64
0.53
0.67
0.60
0.74
0.67
0.77
0.66
0.85
0.57
1.00
0.57
0.51
0.51
0.46
0.57
0.53
-0.60
0.30
-1.00
0.49
0.45
0.41
0.44
0.40
0.43
0.42
0.40
0.48
0.40
0.29
0.41
-0.65
0.28
0.13
0.27
Q-VALUES AFTER 100 ITERATIONS


R
G
Y
B

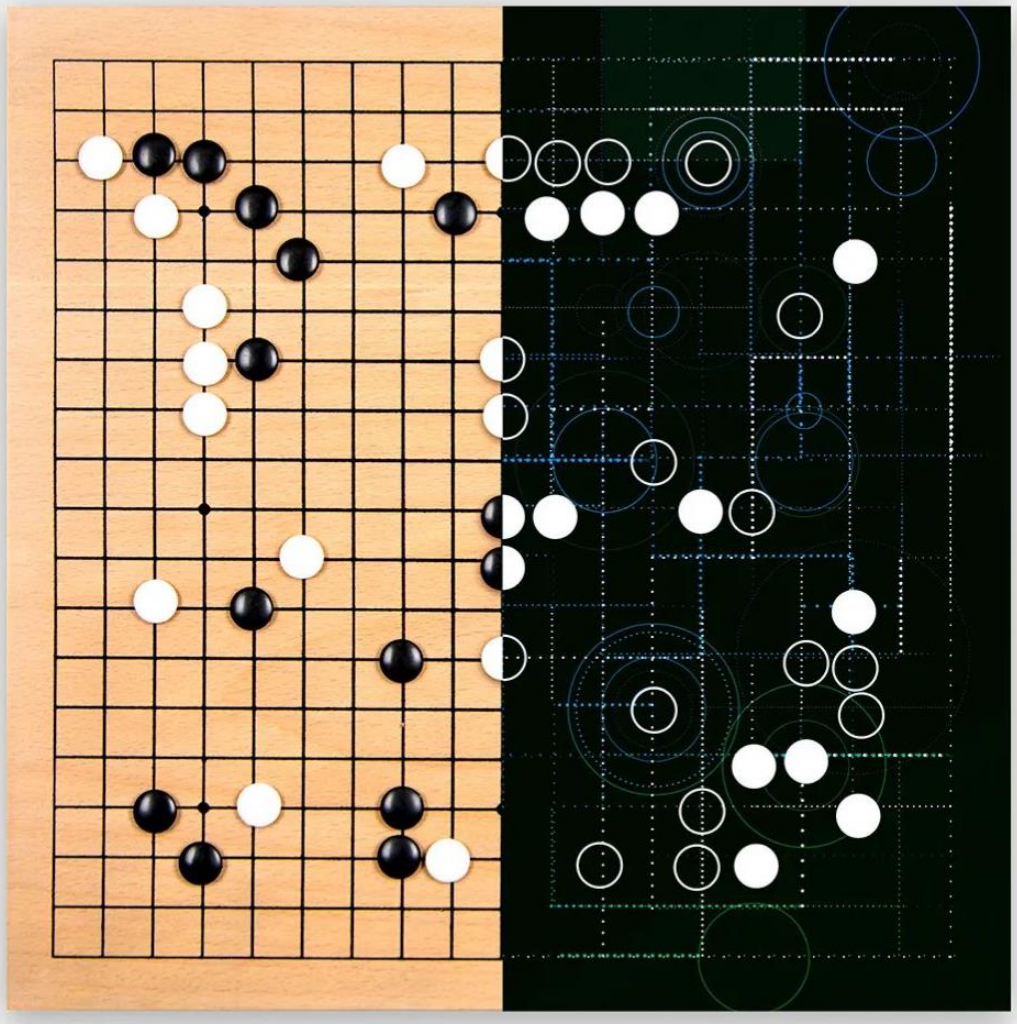
In a Huge Breakthrough, Google's AI Beats a Top Player at the Game of Go

As recently as this month, top AI experts outside Google questioned whether such a victory could be achieved anytime soon.

Machines have topped the best humans at most games held up as measures of human intellect, including chess, Scrabble, Othello, even *Jeopardy!*. But with Go---a 2,500-year-old game that's exponentially more complex than chess

March 2016 victory of AlphaGo against Lee Sedol, a legendary professional player of the game of Go, and in May 2017 against Ke Jie, the world champion

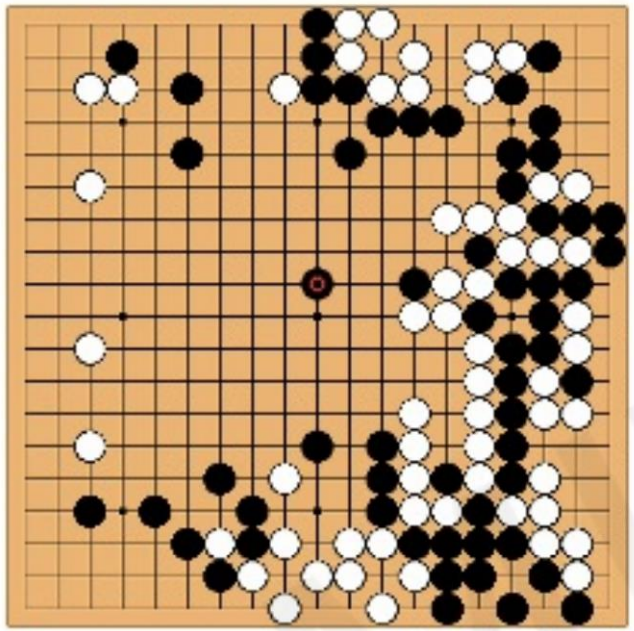
DeepMind was bought by Google for over $500 million in 2014.



https://www.wired.com/2016/01/in-a-huge-breakthrough-googles-ai-beats-a-top-player-at-the-game-of-go/

The Game of Go

Aim: Get more board territory than your opponent.



Board Size $n \times n$	Positions 3^{n^2}	% Legal	Legal Positions
1×1	3	33.33%	1
2×2	81	70.37%	57
3×3	19,683	64.40%	12,675
4×4	43,046,721	56.49%	24,318,165
5×5	847,288,609,443	48.90%	414,295,148,741
9×9	$4.434264882 \times 10^{38}$	23.44%	$1.039191487 91 \times 10^{38}$
13×13	$4.300233593 \times 10^{80}$	8.66%	$3.72497923077 \times 10^{79}$
19×19	$1.740896506 \times 10^{172}$	1.20%	$2.08168199382 \times 10^{170}$

Greater number of legal board positions than atoms in the universe.

https://youtu.be/W_gxLKSsSIE?list=PL5nBAYUyJTrM48dViibyi68urttMlUv7e&t=26

https://www.youtube.com/watch?v=xAXvfVTgqr0

Limitations of Learning by Interaction

- The agent should have the chance to try (and fail) MANY times
- This is impossible when safety is a concern: we cannot afford to fail
- This is also quite impossible in general in real life where each interaction takes time

One of the Solution: RL Simulators

1. OpenAI Gym: An open-source toolkit that provides a wide range of environments for developing and testing reinforcement learning algorithms.

2. Unity ML-Agents: A platform that allows you to train agents in realistic 3D environments, making it ideal for training agents for robotics and game-related applications.

3. MuJoCo: A physics engine for simulating complex and dynamic environments, often used in conjunction with reinforcement learning frameworks for robotics research.

4. CARLA: A simulator designed for autonomous driving research, offering realistic urban environments and sensor data to train self-driving car algorithms.

5. Gazebo: A versatile robot simulation software that is widely used in robotics research and development, offering physics-based simulations for various robot platforms and sensors.

THANK YOU