# Reinforcement learning

# Basic Definitions

- Agent: RL Algorithm that learns from trial and error
- Environment: the world the agent acts upon
- Acton (A): possible steps an agent can take.
- State (S): current condition of the environment.
- Observation (o): representation of the environment visible to the agent.
- Reward (R):  an instant return from the environment to appraise the action

# RL: Driving Analogy

Goal: Get to the destination

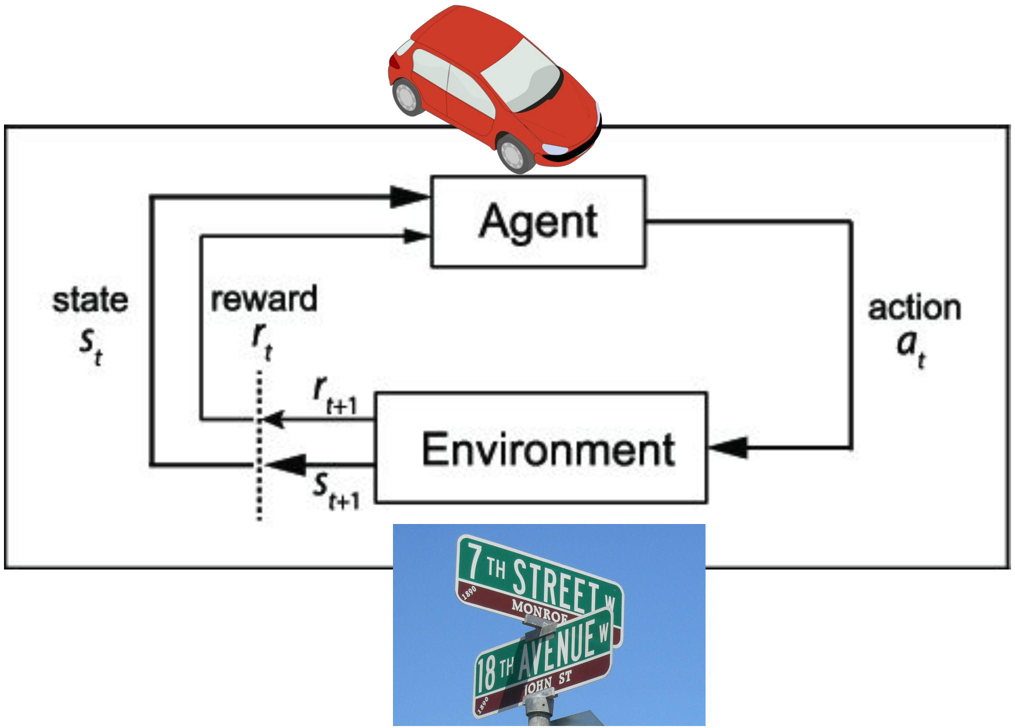
Agent: Driver

Environment: Streets

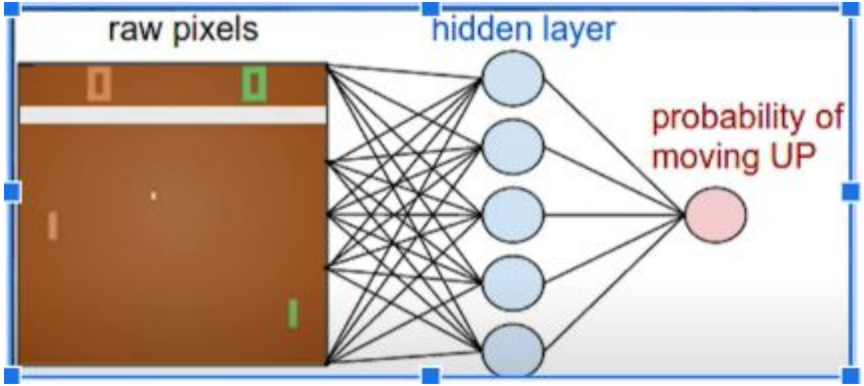
Action: Forward, Left, Right

State: Location of car (S1….S7)

Reward: Time to Goal, Collisions



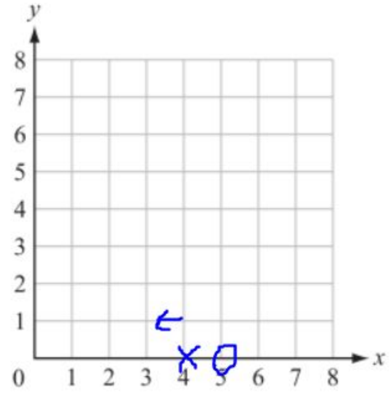
# Return and Policy

- Return (R)
    - The sum of discounted future rewards
    - $V^{\pi}(s) = \mathbb{E}(R_t \mid s_t = s)$
    - $R = \sum_{t=0}^{\infty} \gamma^t r_t.$
- Policy (π): The approach the agent uses to determine the next action based on the current state. Technically, a function from states to a probability distribution over actions.
    - 

# Value vs. Action Value

- Value (V) : expresses the expected value of the policy when agent starts following it from a certain state
    - Expected long term return
- Action Value (Q): similar to value except it takes an extra parameter, the current action.
    - Lets you play with hypothetical of taking a different action the first time
    - $Q(s,a) = r_t + (\gamma * V(s'))$

- Ex: steps away from the goal in value vs action value



# Learning and Planning

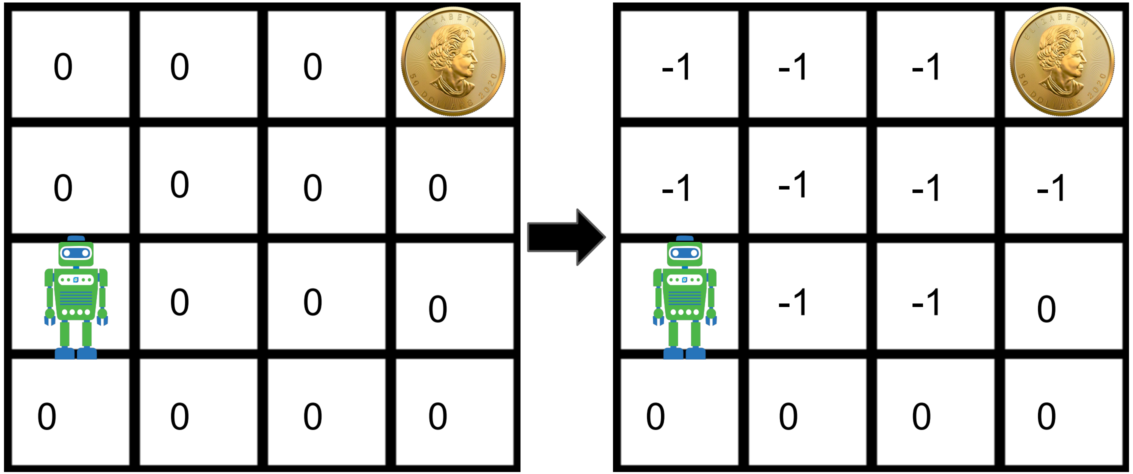
**Learning:**

When the agent has no initial understanding of the environment.

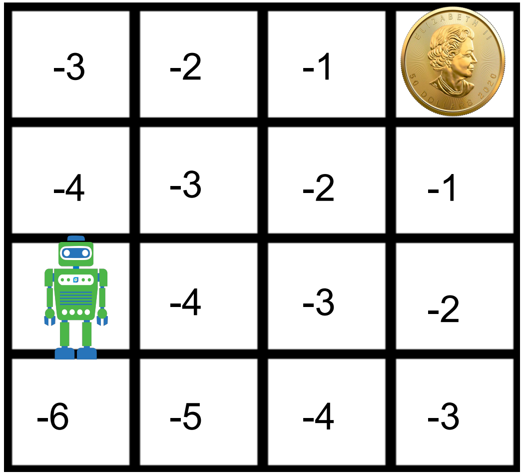
**Planning:**

When the agent has a full understanding of the environment.

**<u>Learning</u>**




**<u>Planning</u>**

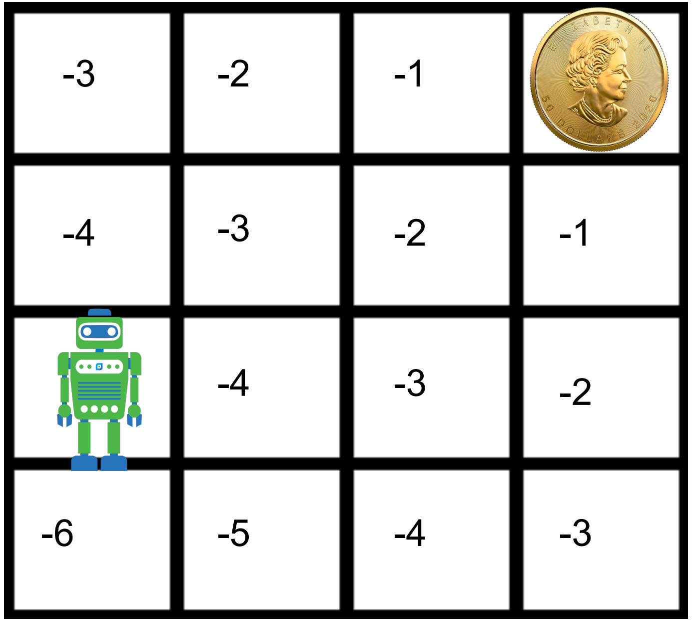


# Reward shaping

**Credit Assignment Problem:**
Deducing where to assign rewards so that the agent will be encouraged to take actions that will maximise its reward total.

**Exploration and Exploitation:**
Balancing the agents decision to explore the environment against exploiting its current knowledge.

**Sparse Reward Setting:**
Delaying the returned rewards from the agents policy and action.



# Prediction and controls

**Prediction:**

Evaluating the the future reward of a given policy.

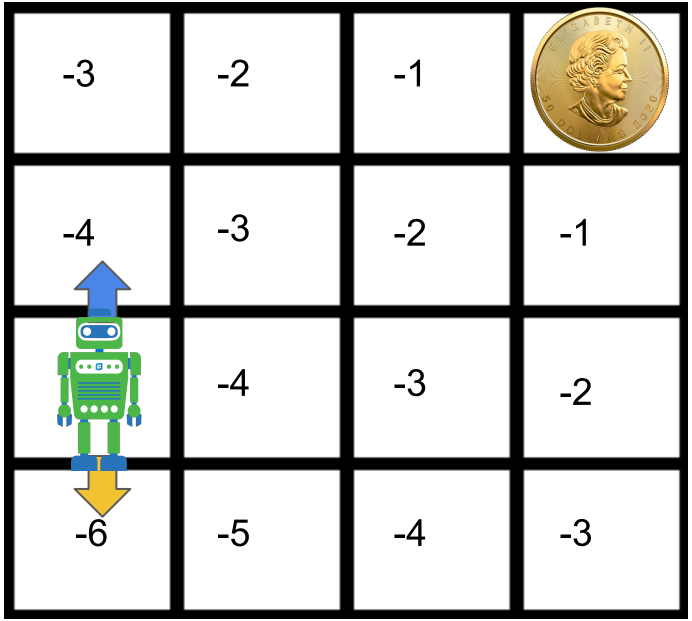
Reward(up policy) = -4

Reward(down policy) = -6

**Control:**

Identifying the most profitable policy given a value function.

Policy(current state) = up policy



# Algorithms/Methods

- Dynamic Programming
- Model-Free: Monte-Carlo/Temporal Difference
- Q-Learning: SARSA

- Dynamic Programming (DP):
    - Christopher J. C. H. Watkins, Learning from Delayed Rewards, Ph.D. Thesis, Cambridge University, 1989. [Thesis]
- Monte Carlo:
    - Andrew Barto, Michael Duff, Monte Carlo Inversion and Reinforcement Learning, NIPS, 1994. [Paper]
    - Satinder P. Singh, Richard S. Sutton, Reinforcement Learning with Replacing Eligibility Traces, Machine Learning, 1996. [Paper]
- Temporal-Difference:
    - Richard S. Sutton, Learning to predict by the methods of temporal differences. Machine Learning 3: 9-44, 1988. [Paper]
- Q-Learning (Off-policy TD algorithm):
    - Chris Watkins, Learning from Delayed Rewards, Cambridge, 1989. [Thesis]
- Sarsa (On-policy TD algorithm):
    - G.A. Rummery, M. Niranjan, On-line Q-learning using connectionist systems, Technical Report, Cambridge Univ., 1994. [Report]
    - Richard S. Sutton, Generalization in Reinforcement Learning: Successful examples using sparse coding, NIPS, 1996. [Paper]