AI IN THE BUILT ENVIRONMENT
DCP4300

Week 10: Robotics

Reinforcement Learning

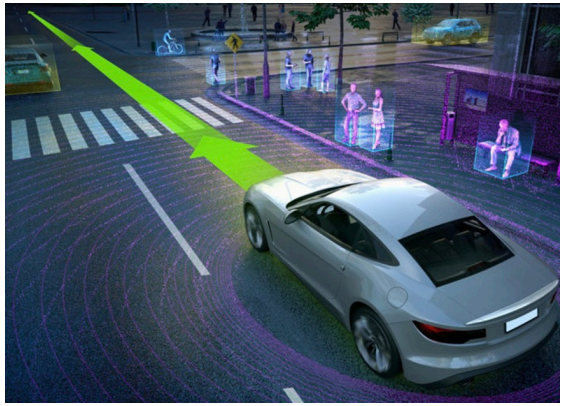
Dr. Chaofeng ‘Charles' Wang

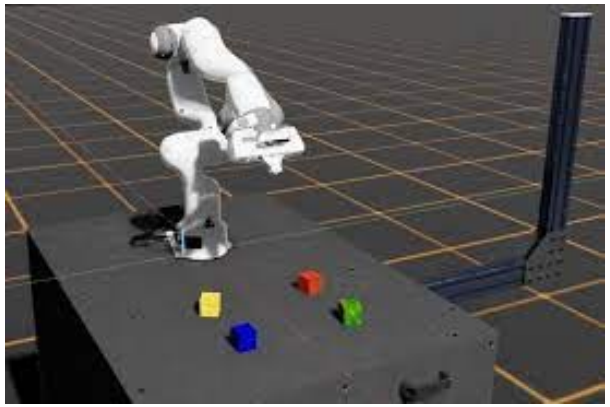
University of Florida
College of Design Construction and Planning

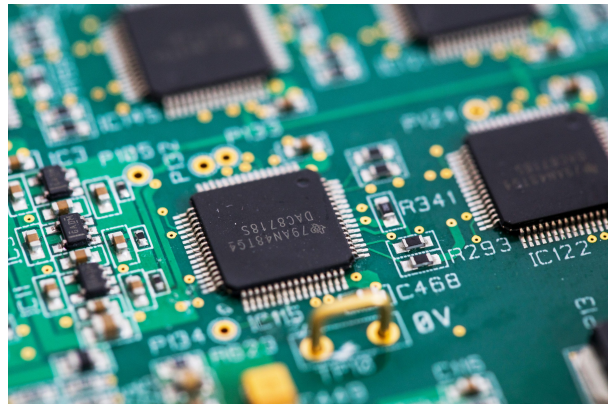
Reinforcement Learning:

Train the Agent to learn to React to an Environment by **trial and error**.

Has broad applications. Some examples:
Autonomous Driving
Gaming AI
Robotics
Design
…







Value-based RL

If we know the optimal action-value function $Q^*(s_t, a_t)$:
The agent can take an action that maximize Q^* :

$$\boldsymbol{a_t = argmax_a} Q^*(s_t, a)$$

$Q^*(s_t, a_t)$ can be calculated by looping over all ***possible future paths***, **for simplest cases**.

A practical method is to approximate it **iteratively**.

Q-learning Algorithm

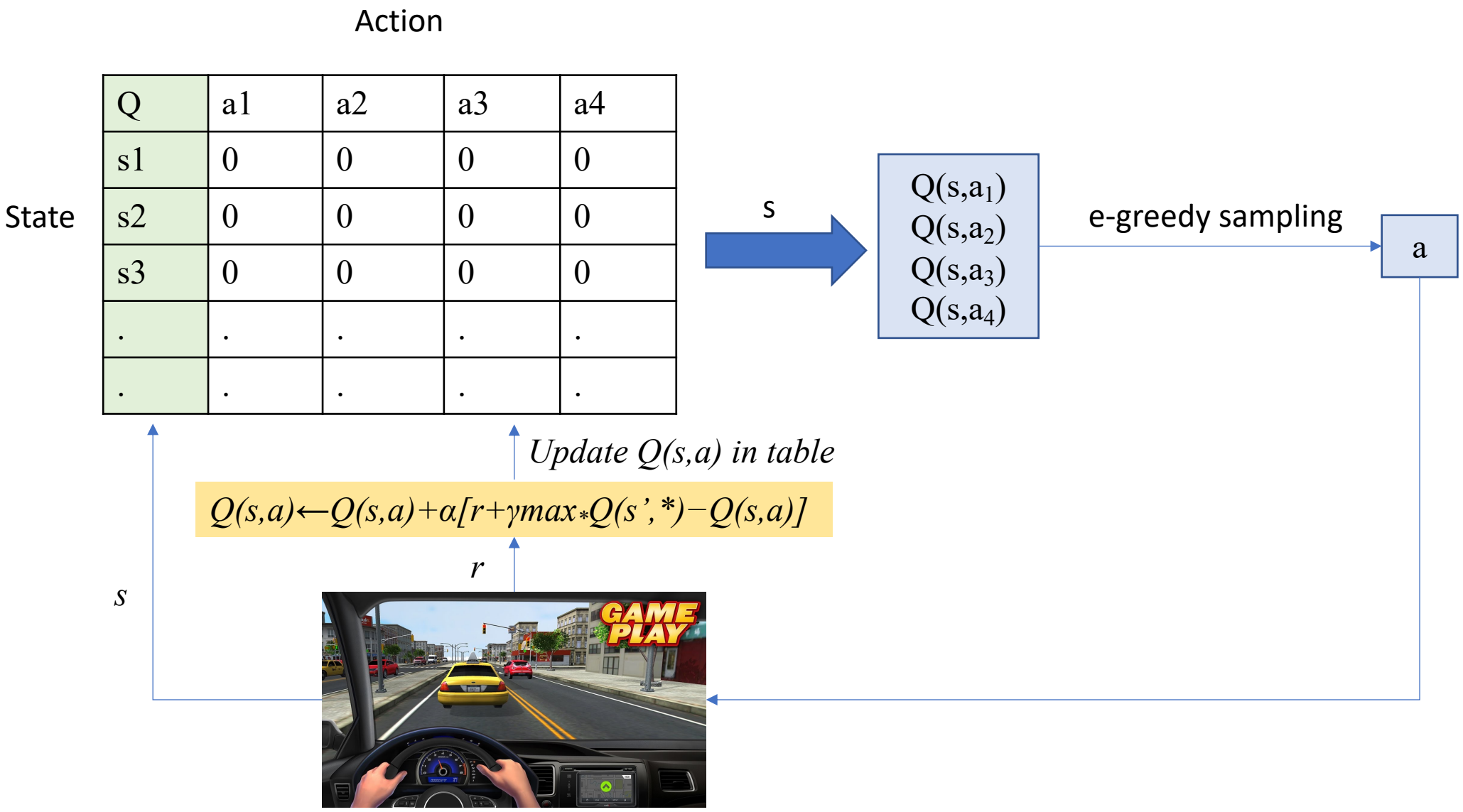
Action

State

Q	a_1	a_2	a_3	a_4
s_1	$Q(s_1, a_1)$	$Q(s_1, a_2)$	$Q(s_1, a_3)$	$Q(s_1, a_4)$
s_2	$Q(s_2, a_1)$	$Q(s_2, a_2)$	$Q(s_2, a_3)$	$Q(s_2, a_4)$
s_3	$Q(s_3, a_1)$	$Q(s_3, a_2)$	$Q(s_3, a_3)$	$Q(s_3, a_4)$
.	.	.	.	.
.	.	.	.	.

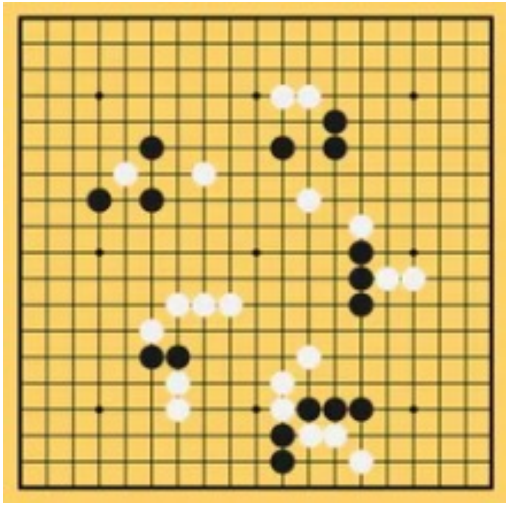
Our objective is to learn the values $Q(s_t, a_t)$ which represents the policy

Q-learning Algorithm



In Q-learning, we use a table to store and calculate the optimal Q values.
This is ok for simple cases.

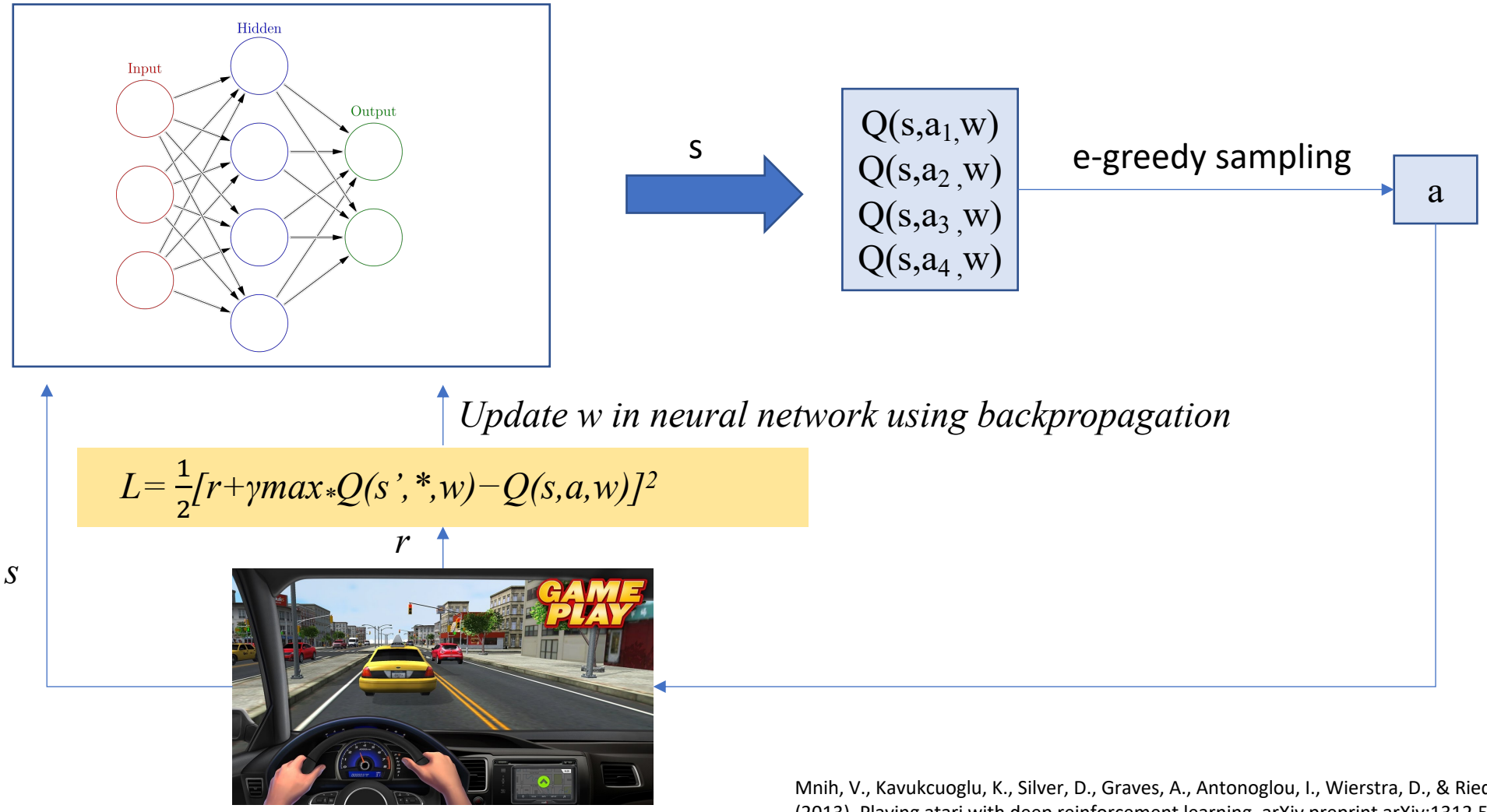
But for more complicated cases, this is not doable because the dimension of the table can the calculation needed are too large.



Instead of using a table, we can use a neural network to approximate the real $Q^*(s_t, a_t)$.

The method is called Deep Q Network (DQN)

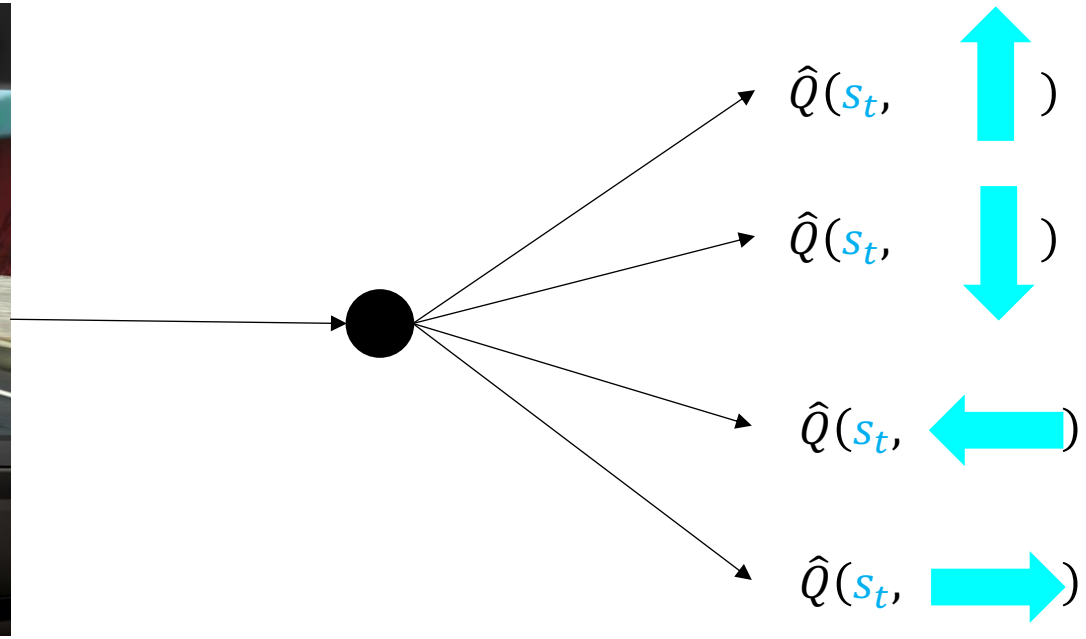
Deep Q Network (DQN)



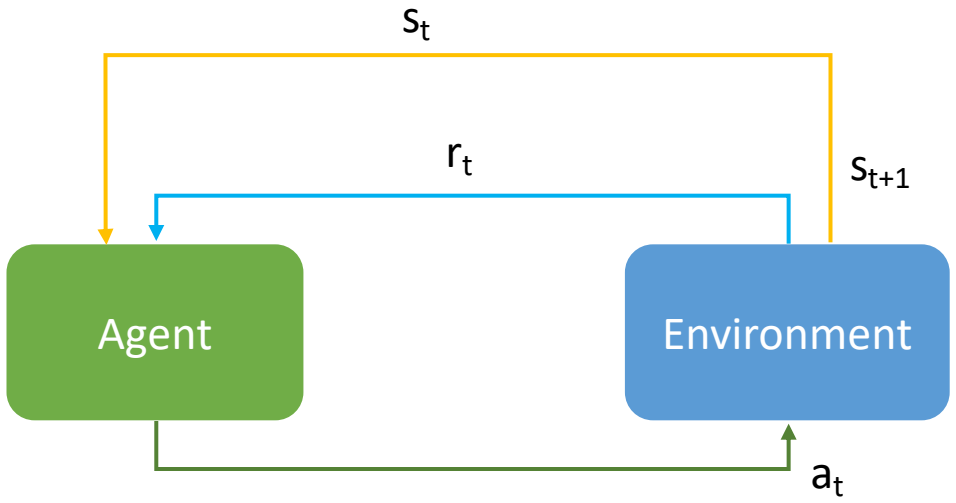
Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602. https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

Deep Q Network (DQN)

s_t



Create A Reinforcement Learning Workflow



OpenAI Gym
DeepMind-control
Atari
Your own

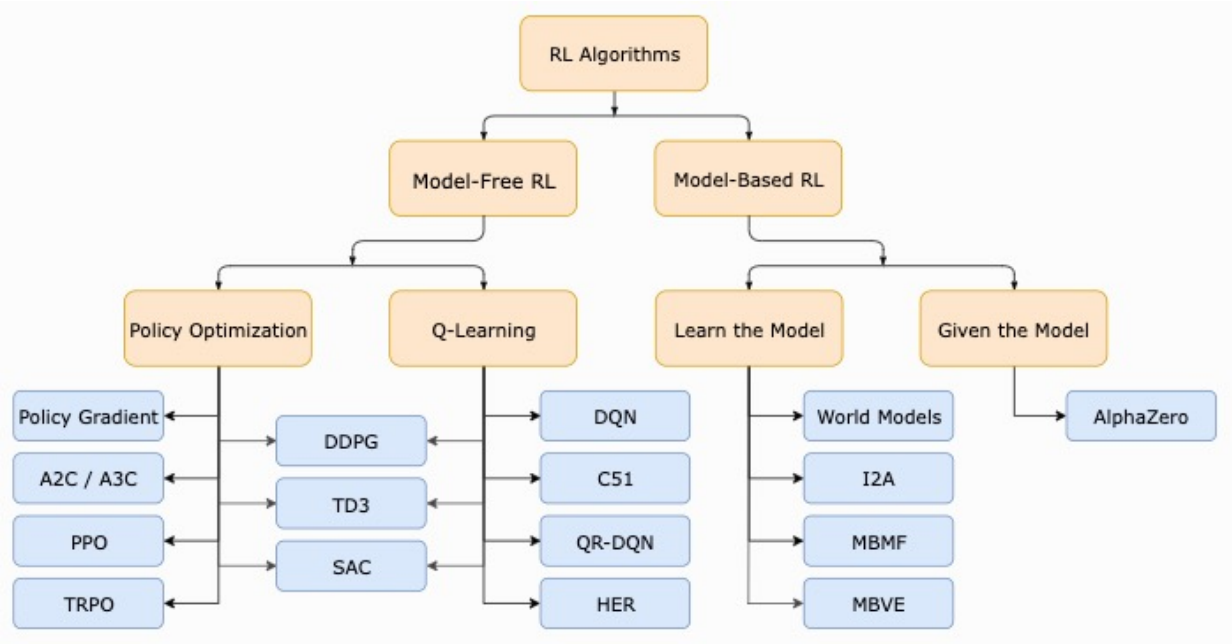
Algorithms:
ARS
A2C
DDPG
DQN
HER
PPO
QR-DQN
SAC
TD3
TQC
TRPO
Maskable PPO
…

Some robotics simulation environments:
OpenAI Gym:
https://www.gymlibrary.ml/pages/third_party_environments/#other-environments
A Review of Physics Simulators for Robotic Applications
https://ieeexplore.ieee.org/document/9386154
A Survey on Simulation Environments for Reinforcement Learning
https://ieeexplore.ieee.org/document/9494694

More RL

OpenAI Spinning Up

https://spinningup.openai.com/en/latest/



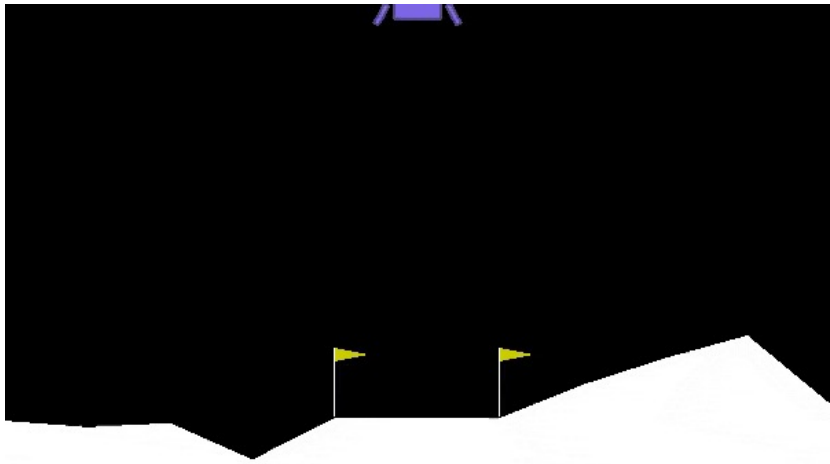
DeepMind x UCL Reinforcement Learning Lecture Series

https://deepmind.com/learning-resources/reinforcement-learning-series-2021

Collection of implementations of RL
https://stable-baselines3.readthedocs.io/en/master/

Tensorflow Agents also has implementations
https://www.tensorflow.org/agents

Exercise: RL with Stable-baseline3



Lunar Lander example: https://stable-baselines3.readthedocs.io/en/master/guide/examples.html

Create your own environments: https://stable-baselines3.readthedocs.io/en/master/guide/custom_env.html

Exercise: Build your own environment and RL workflow (TensorFlow)

Tensorflow Agents: Environment: https://www.tensorflow.org/agents/tutorials/2_environments_tutorial

Tensorflow Agents: DQN https://www.tensorflow.org/agents/tutorials/1_dqn_tutorial

Workflow:
https://colab.research.google.com/drive/1wkklY2cj_-qvA7AJvRUnTYWuE_GNJvvt?usp=sharing