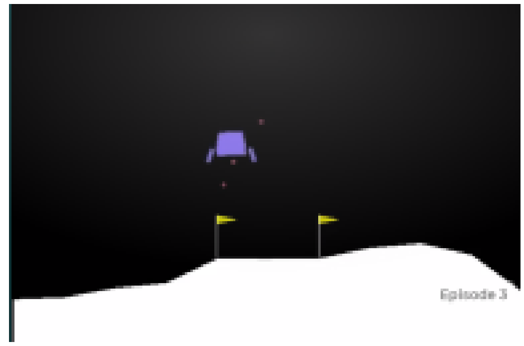

LunarLanderContinuous-v2

Your goal is to land the space-ship between the flags smoothly.

The ship has 3 throttles in it. One throttle points downward and the other 2 points in the left and right direction. With the help of these, you have to control the Ship.

Observation Space: [Position X, Position Y, Velocity X, Velocity Y, Angle, Angular Velocity, Is left leg touching the ground: 0 OR 1, Is right leg touching the ground: 0 OR 1]

Continuous Action Space: Two floats [main engine, left-right engines].

Main engine: -1..0 off, 0..+1 throttle from 50% to 100% power. Engine can't work with less than 50% power.

Left-right: -1.0..-0.5 fire left engine, +0.5..+1.0 fire right engine, -0.5..0.5 off

Please note that there are 2 different Lunar Lander Environments in OpenAIGym. One has discrete action space and the other has continuous action space. **In this part you will solve the continuous one.**

To output discrete action space you will have to quantize the action into a finite number of states. Please use more actions than the discrete case in the openAI gym. Use at least X2 actions than in the discrete case.

Solving the LunarLanderContinuous-v2 means getting an average reward of 200 over 100 consecutive trials.

Link that demonstrate how to use/render the game (with just random actions):

https://colab.research.google.com/drive/1R5BwSTau9zuEj8r4Yh6gB3Nn7NXOmFx?usp=sharing

Your goals in this project are

1) To solve the environment

2) As fast as you can(small number of episodes until solving the problem, i.e. you want a small number of crashes until learning the task) - this is a competition part.

3) Show a comparison of convergence between different methods and variants (e.g. DQN, target network, network architecture, double-DQN, dueling DDQN, experience replay, prioritized experience replay, TD(lambda), discount factor effects, epsilon-greedy, quantizing effects., use terms from the course)