DRL Homework 1

Ludger Masch, Robin Gratz, Linus kleine Kruthaup

April 2022

Task 01

- the set of states S: Finite, every possible game state
- set of actions A: Every legal move of every available piece
- probabilistic state dynamics: No need for uncertainty, should be deterministic. Maybe uncertainty regarding opponents moves.
- the (probabilistic) reward dynamics: No drawback in reward function for time. Positive reward for taking other players pieces, higher rewards for stronger pieces. Max reward for king. Negative reward for losing pieces.
- policy: categorical distribution, either you perform a certain move or you don't

Task 02

- set of states S: Amount of pixels in the game * Color Channel
- set of actions A: Do nothing , fire left, main or right engine
- probabilistic state dynamics: should take into account the velocity of the spaceship when performing an action
- the (probabilistic) reward dynamics: Come to rest (+100), Crash (-100), each leg ground contact (+10), Firing main engine (-0.3 per frame), Solved (+200)
- initial state distribution: Spaceship same starting position. Landing surface randomized apart from the landing pad
- policy: Joint distribution if we assume that multiple actions can be performed at once (as it seems to be in the animation).

Task 03

- The environment dynamics describe the probability of ending up in state s' and receiving the reward r when taking action a in state s. This is a combination of the state transition function p(s'—s,a) and the reward function r(s, a).
  - E.g. in the chess example when performing an action a that exposes the king, there is probably a high chance that you end up in a state s' where the king gets taken by your opponent and you receive a (highly) negative reward r.
  - E.g. in the LunarLander example when performing an action a that throttles the main engine, depending on the velocity, the Spaceship ends up in a state s' and the agent receives a negative reward for that action.
- The true environment dynamics are not known. The state transition function we use in our MDP is always only an approximation of reality. E.g. when considering the BioGas Plant example from the lecture, we can use the data that was gathered across the past to make an educated guess about what will happen when changing a parameter (performing an action a) but we can't know for sure. The same problem holds for the reward function. It can't be known and is defined more or less arbitrarily.