Go For It!

Game: Connect 4 (Hasbro)

Players: 2

Goal: Be the first player to get 4 chips in a row in a 6 × 7 $(r \times c)$ grid, as quickly as possible.

RULES: $I = \{\, if\ your\ turn,\ \ place\ chip\ in\ avail\ able\ slot \rightarrow wait\ for\ other\ player\ to\ play\, \}$

AGENT GOAL: Get the highest score by the end of the game (AI agent is unaware of opponent's score)

$S = \{\, START, PLAYING, P2\_TWO\_IN\_A\_ROW, P1\_TWO\_IN\_A\_ROW,$

$P2\_THREE\_IN\_A\_ROW, P1\_THREE\_IN\_A\_ROW, P1\_WINS, P2\_WINS, TIE\, \}$

$A = \{\, SLOT_1, SLOT_2, SLOT_3, SLOT_4, SLOT_5, SLOT_6, SLOT_7\, \}$

$R = \{\, -1.0, -0.15, -0.1, +0.01, +0.15, +1.0\, \}$

$P1 \equiv AI\ Agent$

$X_k \subset X : X\ is\ any\ set$

NOTE: each state $s \in S$ contains 6 x 7 matrix w/ cell_state of each cell $\{\, EMPTY,\ \ P1\_CHIP,\ \ P2\_CHIP\, \}$

STATE: START

PARAMETERS: $\{\, both\ players\ have\ 0.0\ points\, \}\ \{\, 0\ total\ plays\ have\ been\ executed\, \}$

RULES: I

ACTIONS: $A = \{\, SLOT_1, SLOT_2, SLOT_3, SLOT_4, SLOT_5, SLOT_6, SLOT_7\, \}$

STATE: PLAYING

PARAMETERS: $\{\, 1 +\ chips\ on\ board\, \}$

REWARD: $\{+0.01\}$

RULES: I

ACTIONS: $A_k = A - \{full\ slots\}$

STATE: P2_TWO_IN_ROW

PARAMETERS: { $P2\ gets\ 2\ chips\ in\ a\ row$ }

REWARD: $\{-0.1\}$

RULES: I

ACTIONS: $A_k = A - \{full\ slots\}$

STATE: P1_TWO_IN_ROW

PARAMETERS: { $P1\ gets\ 2\ chips\ in\ a\ row$ }

REWARD: $\{+0.01\}$

RULES: I

ACTIONS: $A_k = A - \{full\ slots\}$

STATE: P2_THREE_IN_ROW

PARAMETERS: { $P2\ gets\ 3\ chips\ in\ a\ row$ }

REWARD: $\{-0.15\}$

RULES: I

ACTIONS: $A_k = A - \{full\ slots\}$

STATE: P1_THREE_IN_ROW

PARAMETERS: { $P1\ gets\ 3\ chips\ in\ a\ row$ }

REWARD: $\{+0.015\}$

RULES: I

ACTIONS: $A_k = A - \{full\ slots\}$

STATE: P1_WINS

PARAMETERS: { $P1\ has\ 4\ chips\ in\ a\ row$ }

REWARD: $\{+1.0\}$

STATE: P2_WINS

PARAMETERS: { $P2\ has\ 4\ chips\ in\ a\ row$ }

REWARD: $\{-1.0\}$

STATE: TIE

PARAMETERS: $\{\ all\ positions\ filled\ without\ 4\ chips\ in\ a\ row\ \}$

REWARD: $\{-1.0\}$

MDP (**Markov Decision Process**) **Comparison**

The Markov decision process (MDP) is ideal for stochastic situations in which the (probabilistic) outcomes are dependent on the decisionmaker's control, as well as randomness from the environment. Markovian processes have future states dependent only on the current state and action, not the sequence of events that led to it. MDP shares many similarities with the approach I took, including states, actions, rewards, and goals. Transition Probabilities P are used in MDP but is something I did not use in my approach. P represents the chances of moving between states given actions. MDP also implements a policy π to evaluate the state and a value function V to determine the policy's effectiveness. MDP also includes components like the Q-value function Q to evaluate actions and a discount factor γ, which is applied in V and Q to balance the importance of immediate versus future rewards. In conclusion, MDP using function approximation (due to the vast number of possible states) is an ideal approach for reinforcement learning (RL) of an AI agent learning to play and win *Connect 4*!

References:

Beysolow II, Taweh. (2017). *Applied Reinforcement Learning with Python: With OpenAI Gym, Tensorflow, and Keras*. Apress L.P.

Lamba, Akshay. (2018, August 2018). *A brief introduction to reinforcement learning*. Medium. https://medium.com/free-code-camp/a-brief-introduction-to-reinforcement-learning-7799af5840db.

Sharma, Abhishek. (n.d.) .*Markov Decision Process*. GeeksforGeeks. https://www.geeksforgeeks.org/markov-decision-process/.

Wikepedia. (n.d.). *Markov decision process*. *https://en.wikipedia.org/wiki/Markov_decision_process#:~:text=In%20mathematics%2C%20a%20Markov%20decision,control%20of%20a%20decision%20maker*.

Wikipedia. (n.d.). *Value function*. https://en.wikipedia.org/wiki/Value_function.