Reinforcement Learning Project

BLACKJACK

Hugo DENIS--MARTIN - Brice MABILLE

SUMMARY

1 - INTRODUCTION

Subject Presentation: Blackjack as a learning environment.

Objectives: Study RL approaches and compare their performances.

Originality: Adaptation and experimentation of algorithms on this specific case.

2 - DEFINITION OF THE ENVIRONMENT

Objective of the Game: Achieve a hand value closer to 21 than the dealer's without exceeding 21.

Available Actions:

- Hit: Draw an additional card.
- Stand: Keep the current hand and end the turn

3 - METHODOLOGY

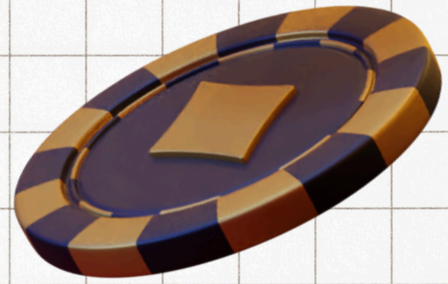
Comparison between 4 different agents:

- Random agent
- Simple agent
- Q-Learning
- SARSA

RANDOM AGENT

Strategy used:
Random action: On each turn, the agent randomly decides between:

- Drawing a card (hit).
- Stopping (stand).



SIMPLE AGENT

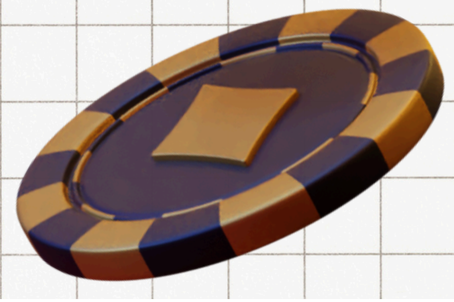
Strategy used:
Fixed threshold: The agent uses a deterministic strategy based on the value of its hand.

- If the hand value is less than 17, the agent draws a card (hit).
- If the hand value is 17 or more, the agent stops (stand).

Q-LEARNING

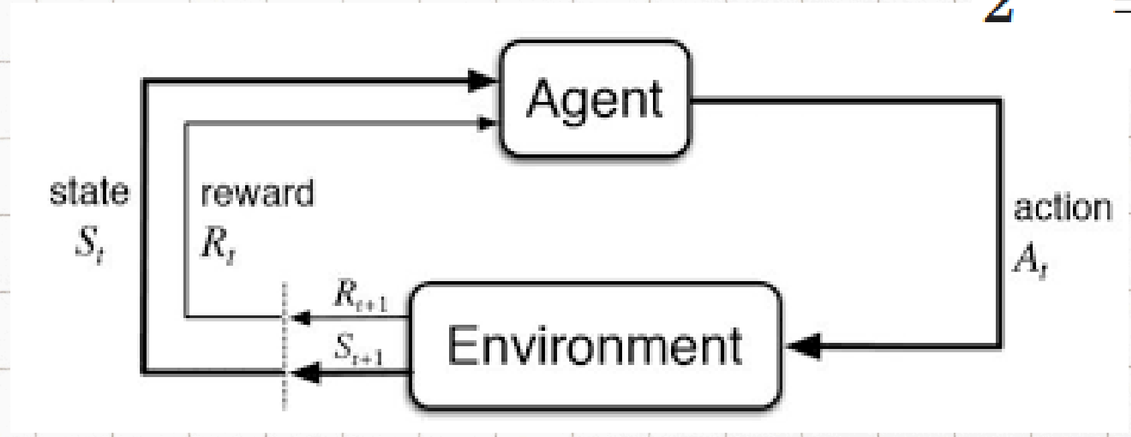
Strategy used:

- The agent learns to play Blackjack by updating a Q-table that associates states (hand values, presence of an Ace, dealer's visible card) with possible actions (hit or stand).
- It uses an epsilon-greedy strategy to balance exploration and exploitation.



$2^{280} = 1.94 * 10^{84}$ possible policies

SARSA

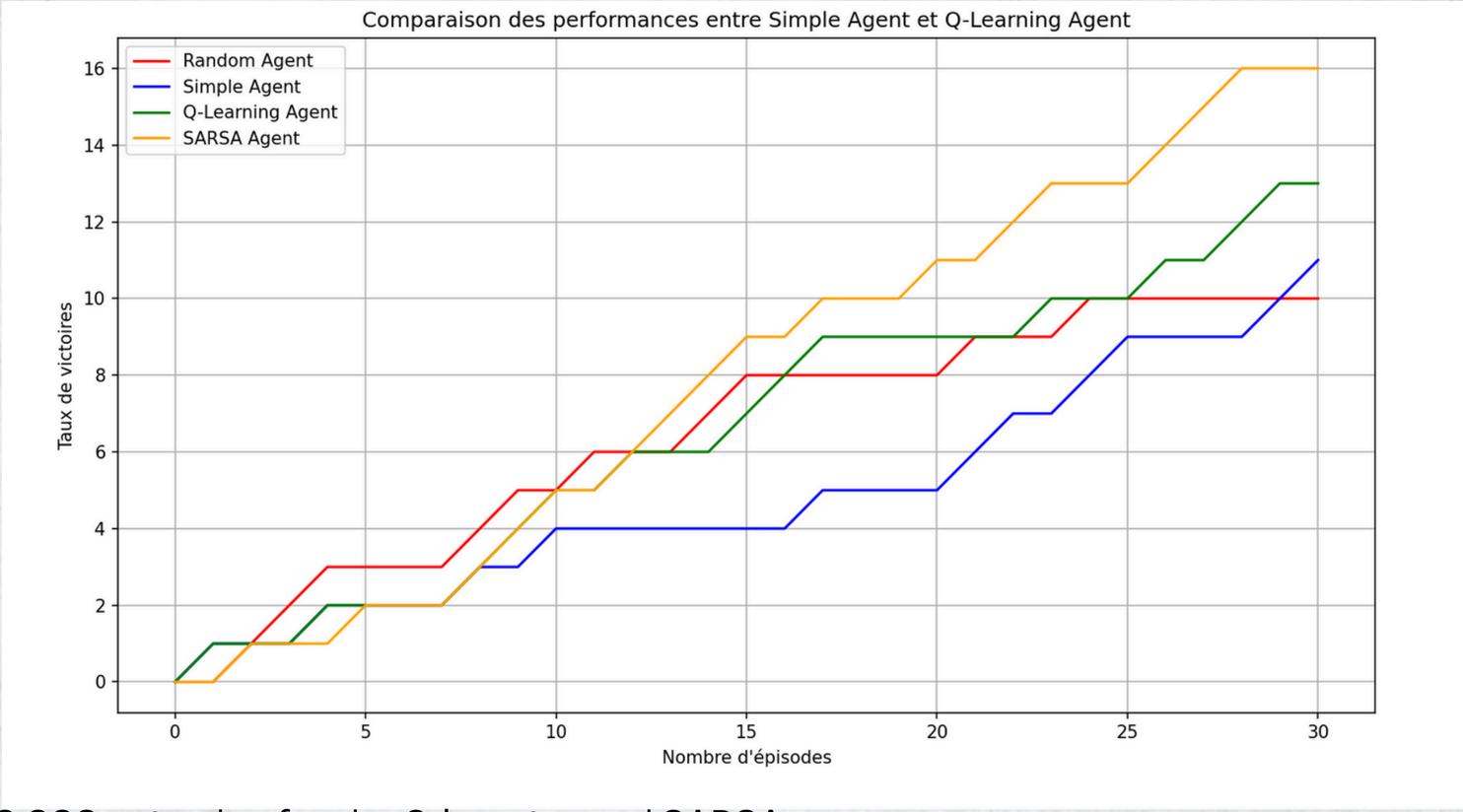


Strategy used:

- The agent uses the SARSA (State-Action-Reward-State-Action) algorithm to learn to play Blackjack.
- Unlike Q-Learning, SARSA takes into account the next chosen action to update the Q-values.

4 - RESULTS AND PERFORMANCE

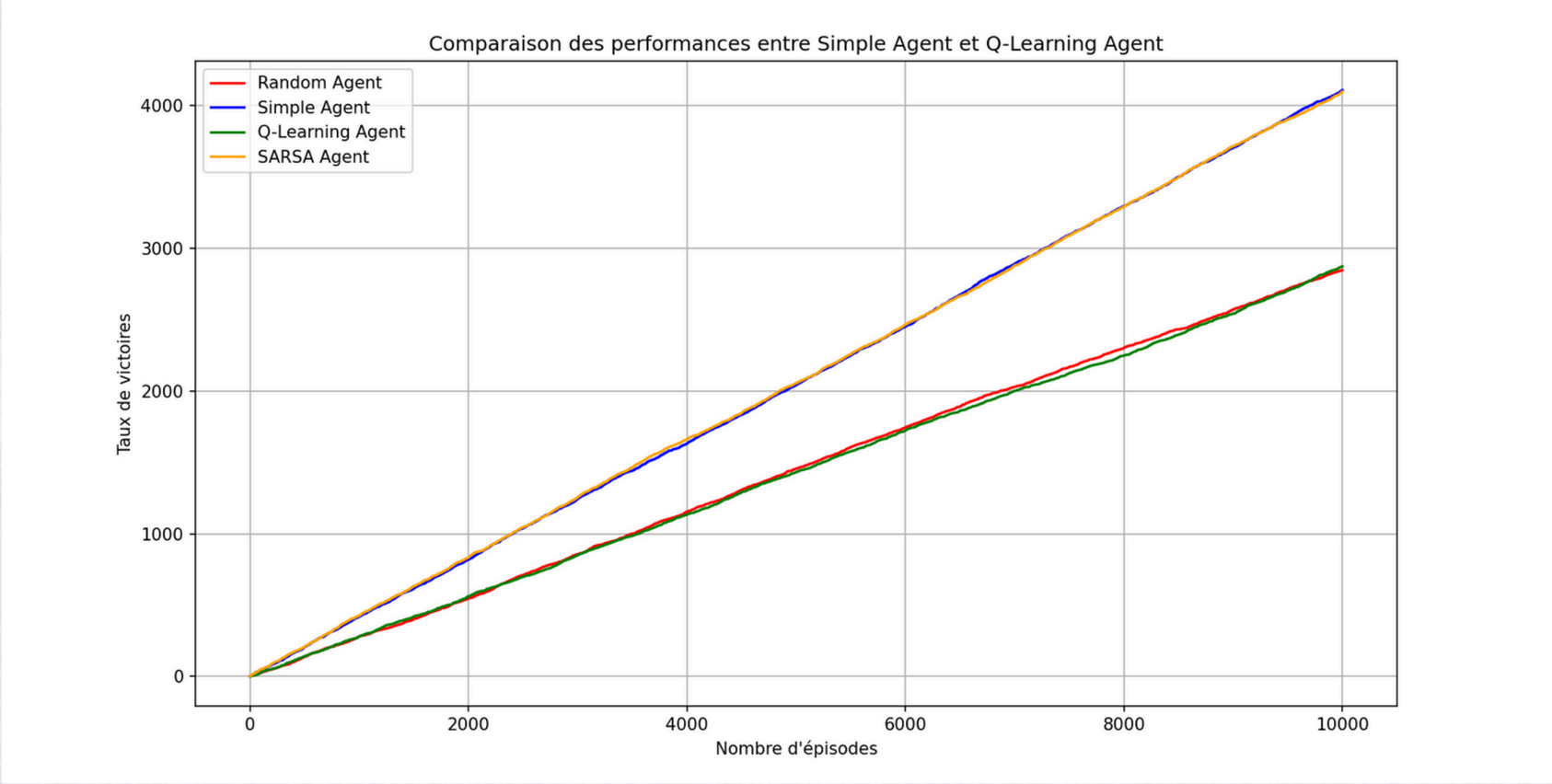
With the code created, comparison.py, we obtain the following performance results:



With a training of 10,000 episodes for the Q-learning and SARSA agents:

4 - RESULTS AND PERFORMANCE

Since we were not satisfied with the previous results, we optimized the agents' parameters and the rewards related to the environment to achieve better performance:



Despite these changes, the element of chance in drawing a card is too significant; the implemented agents learn a loss due to a tie or a hand value > 21. Therefore, they lower their expectations.

6 - CONCLUSION AND PERSPECTIVES

IN THIS BLACKJACK GAME, WE REALIZED THAT REINFORCEMENT LEARNING APPLIED TO BLACKJACK ALLOWS THE AGENT TO REACH THE LEVEL OF THE SIMPLE AGENT BUT DOES NOT SURPASS IT DUE TO THE FLUCTUATING NATURE OF THE GAME'S RESULTS.

TO GO FURTHER, WE COULD DEFINE WHAT ACTION THE AGENT SHOULD CHOOSE BASED ON THE VALUE OF THE CARDS OBTAINED BY THE DEALER.

Soft total \ Dealer hole card	2	3	4	5	6	7	8	9	10	A
13	H	H	H	H	H	H	H	H	H	H
14	H	H	H	H	H	H	H	H	H	H
15	H	H	H	H	H	H	H	H	H	H
16	H	H	H	H	H	H	H	H	H	H
17	H	H	H	H	H	H	H	H	H	H
18	S	S	S	S	S	S	S	H	H	H
19	S	S	S	S	S	S	S	S	S	S
20	S	S	S	S	S	S	S	S	S	S
21	S	S	S	S	S	S	S	S	S	S

7 - REFERENCES

ENVIRONNEMENT_BLACKJACK :

HTTPS://GITHUB.COM/SHEETALBONGALE/BLACKJACK-PYTHON/BLOB/MASTER/BLACKJACK.PY

ARTICLE D'APPRENTISSAGE Q_LEARNING DE LA MEILLEURE STRATÉGIE :

HTTPS://LUCASPAUKER.COM/ARTICLES/REINFORCEMENT-LEARNING-APPLIED-TO-BLACKJACK/

THANKS