EPITECH

TAXI DRIVER

KEYNOTE

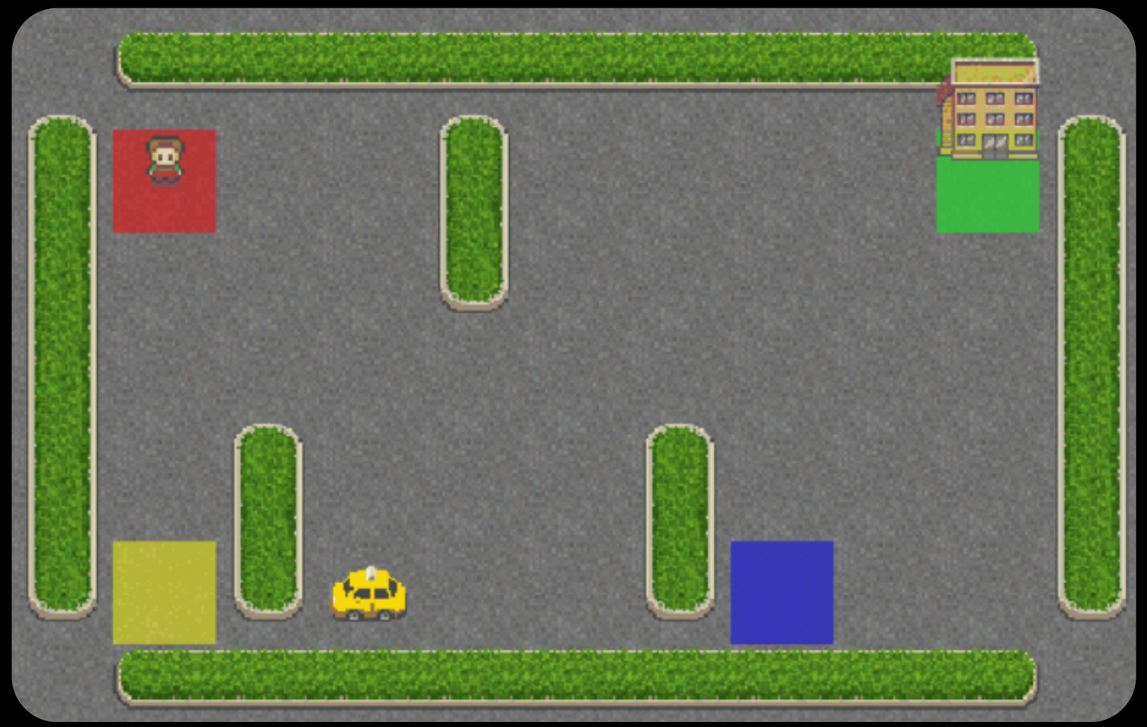
ANAIS, NICOLAS, SULIAN, AMINE

EPITECH

PROJET

- Résoudre l’exercice via 3 type algorithmes d’apprentissage par renforcement
- Fournir une analyse complète des performances

TAXI-V3



Objectif :

1. Récupérer un client
2. Le déposer à sa destination

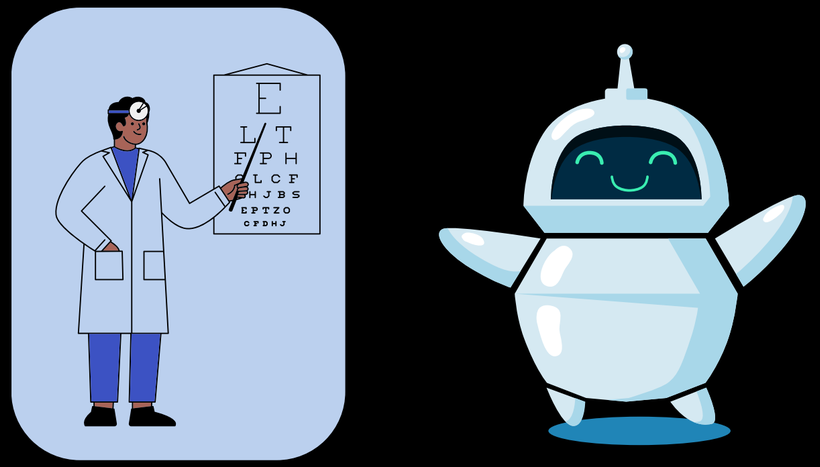
Observation : 500 états

Action : 6

- Déplacements (haut, bas, gauche, droite)
- Prendre le passager
- Déposer le passager

LES TYPES D'APPRENTISSAGE EN IA

Apprentissage supervisé et non supervisé



Apprentissage par renforcement

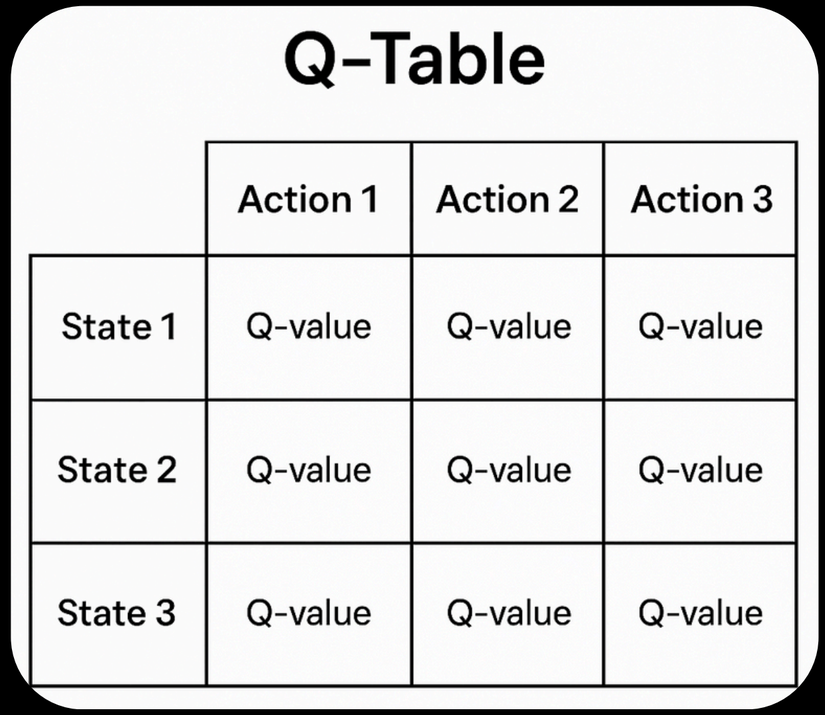
EPITECH

APPRENTISSAGE PAR RENFORCEMENT

NOTIONS IMPORTANTES

Epsilon

compromis entre exploration et exploitation



EPITECH

Q-LEARNING

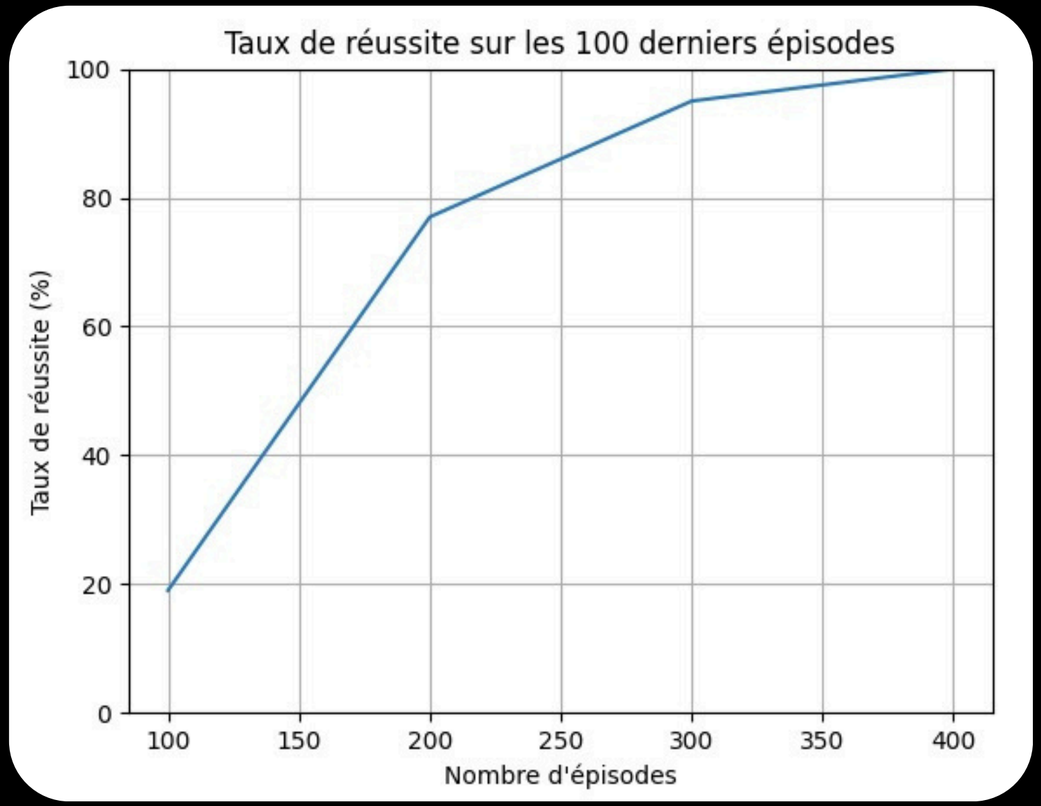
PRINCIPE

À chaque étape :

1. **L'agent est dans un état.**
2. **Il choisit une action (exploration / exploitation)**
3. **Il reçoit une récompense et passe dans un nouvel état.**
4. **Il met à jour la Q-Table en fonction de la récompense et de l'état obtenu.**
5. **Il réduit l'epsilon pour favoriser l'exploitation**

Q-LEARNING

RESULTATS



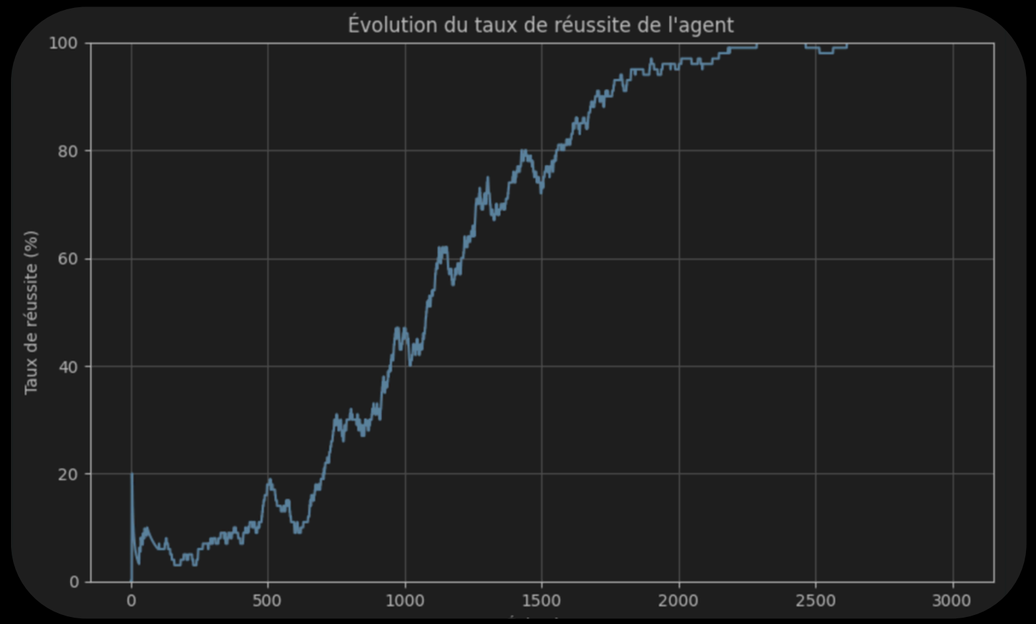
DEEP Q-LEARNING

PRINCIPE

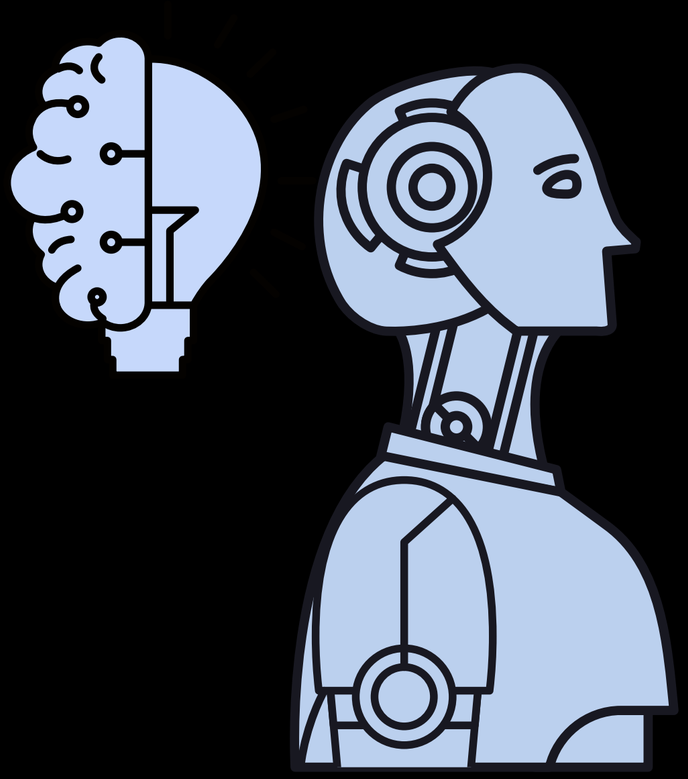
q-table → reseau de neurones (memory)

prédiction des valeurs Q

RESULTATS



PRINCIPE DE MONTE CARLO



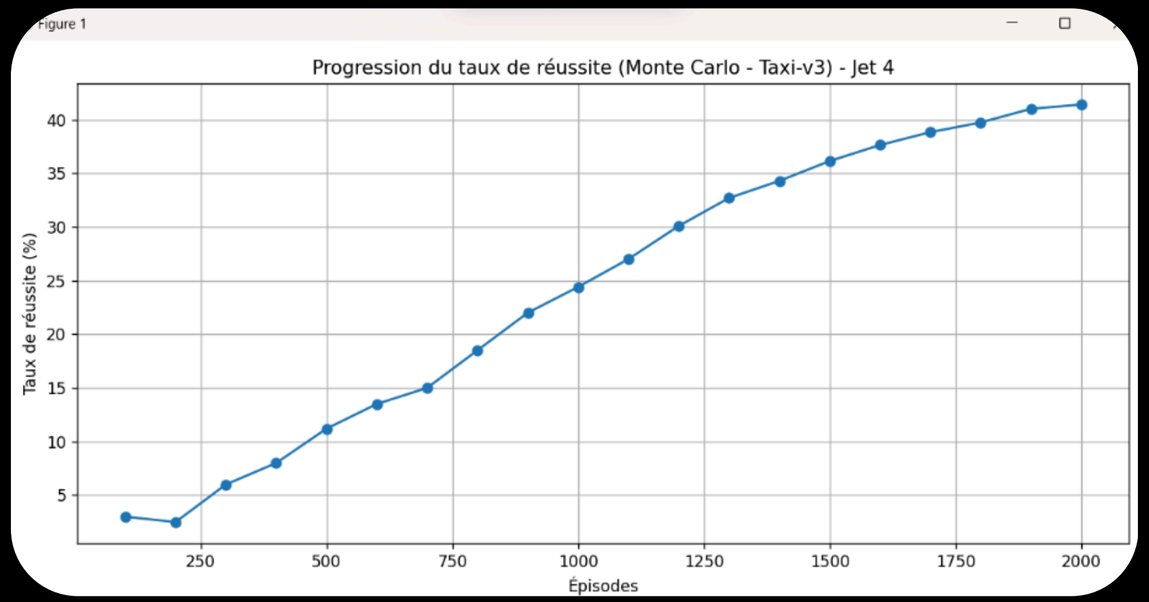
Apprentissage à partir d’épisodes complets

Mise à jour Q-table avec la moyenne des retours

Pas besoin de connaître les transitions

Stratégie ε-greedy pour exploration/exploitation

PERFORMANCE ET RÉSULTATS



Courbe d’apprentisage plus lente que q-Learning

Mise A jour uniquement A la fin de l’episode

Taux de Réussite progesse, mais moins stable

EPITECH

COMPARAISON MONTE CARLO VS Q-LEARNING

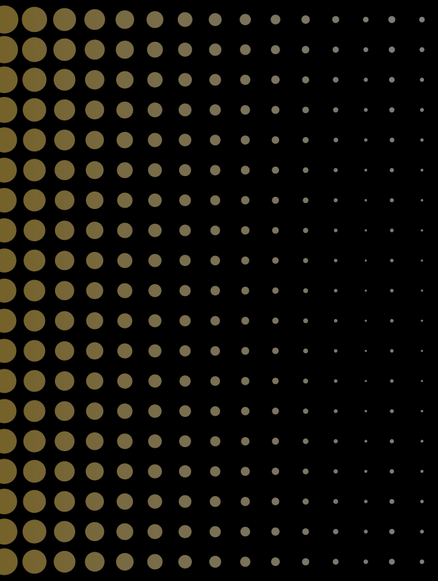
Monte Carlo : apprentissage plus précis mais plus lent

Les deux utilisent ε-greedy

Critère	Q-Learning	Monte Carlo
Moment de mise à jour	Mise à jour à chaque étape (temps réel)	Mise à jour à la fin de chaque épisode (apprentissage par retour d'expérience)
Méthode de calcul	Basé sur une approximation récursive : $Q(s,a) \leftarrow Q(s,a) + \alpha\,[R + \gamma \max(Q(s', a')) - Q(s,a)]$	Basé sur la moyenne des retours (G) : récompense cumulée depuis l'étape jusqu'à la fin
Stabilité de l'apprentissage	Moins stable à court terme (car mise à jour fréquente)	Plus stable à long terme (moyenne des retours)
Convergence	Plus rapide : stratégie optimale apprise en moins d'épisodes	Plus lente, mais plus précise si nombre d'épisodes élevé
Utilisation de First-Visit	Non applicable	Oui : seule la première apparition d'une paire (état, action) est prise en compte par épisode
Indicateurs mesurés	Nombre moyen d'étapes / Récompense moyenne	Nombre moyen d'étapes / Taux de réussite (succès à 20 points)
Courbe de performance	Courbe plus stable et ascendante rapidement	Courbe plus lente et irrégulière, due au retard d'apprentissage
Adapté à Taxi-v3 ?	✅ Oui : apprentissage rapide, environnement dynamique	❌ Moins adapté : apprentissage lent, interactions fréquentes requièrent mise à jour rapide

Q-Learning : mise à jour à chaque étape → rapide

Q-learning = TD Learning | Monte Carlo = Episode-based



DES QUESTIONS ?

PRÉVENTION

La différence est dans quand on met à jour la Q-table?
En First-Visit, on ne met à jour une paire (état, action) que la première fois qu'elle apparaît dans un épisode.
En Every-Visit, on la met à jour à chaque fois qu'elle apparaît, même plusieurs fois dans le même épisode.

Parce que First-Visit réduit la variance des mises à jour, ce qui rend l'apprentissage plus stable. Il est aussi plus simple à implémenter, surtout pour débuter. Every-Visit met à jour plusieurs fois par épisode, ce qui peut introduire du bruit dans les estimations.

Pourquoi le taux de réussite n'augmente plus après 40-45 % ? Pourquoi peut-il redescendre ?
Le taux de réussite plafonne ou baisse après 40 % car Monte Carlo First-Visit met à jour la Q-table uniquement à la fin des épisodes, ce qui rend l'apprentissage lent, peu réactif, et sensible aux mauvaises estimations initiales. De plus, si l'exploration diminue trop tôt, l'agent peut se bloquer dans une stratégie sous-optimale.