

MODULE : IAD & AI

L'Intelligence Artificielle Distribuée & Agent Intelligent

Master 1 : **S**ciences de **D**onnées et **I**ntelligence **A**rtificielle

2023 - 2024

Plan de Présentation

- Apprentissage par Renforcement :

Définitions , Types d'Apprentissage, Comparaison, RL pour Agent ..

- Exploration & Exploitation.

- La Stratégie : ϵ -Greedy.

- **Valeur d'Etat : $V(s)$.**

- **Valeur d'Action : $Q(s,a)$.**

- L'Algorithme : Q-Learning.

- L'Algorithme : Deep Q-Learning.

- L'Algorithme : Policy Gradient.

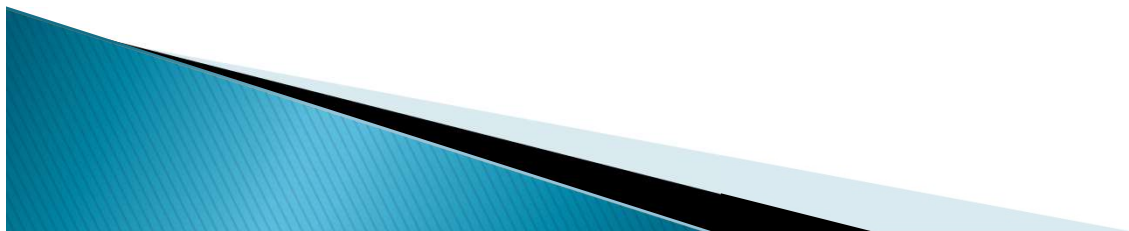
Fonction de Valeur

- Dans l'apprentissage par renforcement, une **fonction de valeur** (value function) est une fonction qui estime la **valeur d'un état** ou **d'une action** dans un environnement donné.
- Cette fonction est cruciale car elle guide les décisions de l'agent dans l'environnement.
- **But :** Faire de la prédiction sur les bonnes actions de l'agent, pour maximiser le nombre des récompenses.



Valeur d'Etat $V(s)$

- La valeur d'état estime simplement l'importance d'être dans un état particulier s de l'environnement, indépendamment de la manière dont l'agent est arrivé à cet état ou des actions qu'il choisira ensuite.
- Elle représente la récompense cumulative attendue que l'agent peut s'attendre à recevoir s'il se trouve dans cet état.
- La valeur d'état est une mesure essentielle de la qualité de l'état lui-même, et elle ne dépend pas de la stratégie spécifique suivie par l'agent.



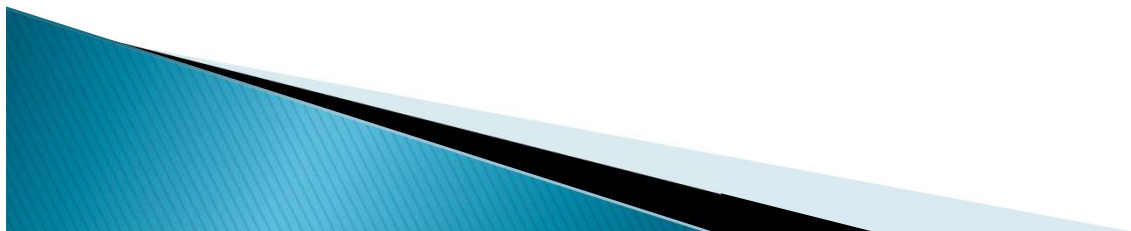
Valeur d'Action $Q(s,a)$

- La valeur d'action estime l'importance de prendre une action particulière a dans un état donné s , en suivant une politique spécifique pour choisir les actions ultérieures.
- Elle dépend donc de la politique adoptée par l'agent.
- Constat : Deux politiques différentes peuvent conduire à des valeurs d'action différentes pour la même paire état-action.
- Par conséquent, la valeur d'action est principalement liée à la stratégie adoptée par l'agent pour choisir ses actions dans l'environnement.



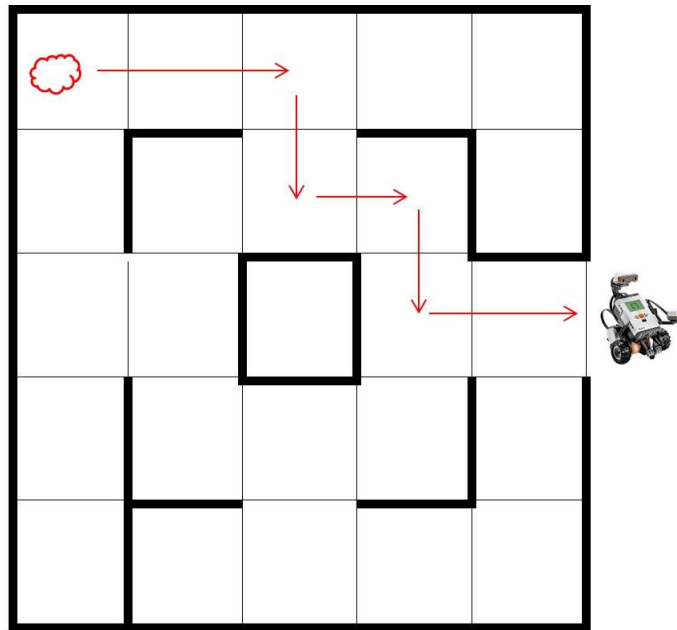
$V(s)$ vs $Q(s,a)$

- La valeur d'état est une mesure générale de l'importance des états de l'environnement, tandis que la valeur d'action est spécifique à une action donnée dans un état donné, et elle dépend de la politique suivie par l'agent.
- La valeur d'action est essentiellement la valeur associée au résultat de la fonction Q , qui estime l'importance ou l'utilité de prendre une action spécifique dans un état donné.
- La valeur d'état fournit une vue globale de l'utilité des états, tandis que la valeur d'action fournit une vue détaillée de l'utilité des actions dans des contextes spécifiques.



Exemple

- Un agent robotique qui se déplace dans un labyrinthe pour atteindre une sortie. Supposons que le labyrinthe soit représenté par une grille d'états, où chaque case représente un état possible pour le robot. Les états peuvent être des cases vides, des murs, ou la sortie du labyrinthe.



Exemple : Valeur d'Etat $V(s)$

- Imaginons que l'agent estime la valeur de chaque case du labyrinthe, en fonction de la distance à la sortie. Plus une case est proche de la sortie, plus elle aura une valeur élevée.
- Supposons que l'agent estime que la valeur de la case1 (s_1) est 5, la valeur de la case2 (s_2) est 10, et ainsi de suite.
- Ces valeurs représentent l'importance de chaque état indépendamment des actions que l'agent prend pour y arriver.



Exemple : Valeur d'Action $Q(s,a)$

- Considérons une situation où l'agent est dans la **case1 ($s1$)**. Il doit choisir entre trois actions : aller vers **le haut**, vers **la droite** ou vers **le bas**.
- La valeur d'action $Q(s1, droite)$ dépendra de la politique de l'agent pour choisir entre ces actions. Si l'agent suit une politique qui favorise la sortie **la plus rapide (exploitation)**, alors $Q(s1, droite)$ pourrait avoir une valeur élevée, car aller vers la droite rapproche l'agent de la sortie.
- Cependant, si l'agent suit une politique qui **favorise l'exploration**, alors $Q(s1, bas)$ pourrait avoir une valeur élevée car l'agent pourrait découvrir une nouvelle zone du labyrinthe en allant vers le bas.



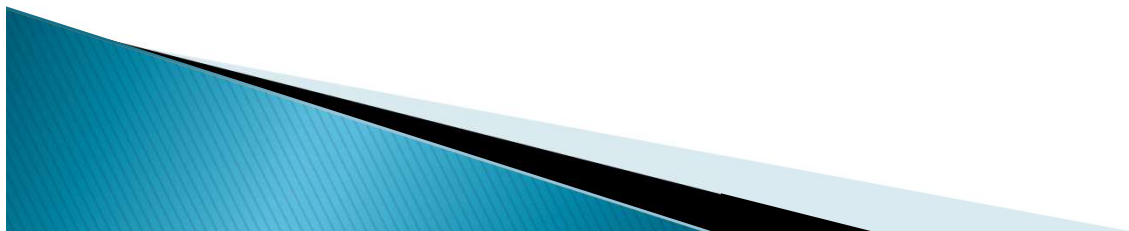
Récapitulatif : Fonction de Valeur

- **Fonction de valeur d'état (State Value Function)** : Cette fonction estime la valeur d'être dans un état particulier de l'environnement.
 - Elle est généralement notée $V(s)$, où s représente l'état.
 - Elle indique à l'agent à quel point il est bénéfique d'être dans un certain état donné.
- **Fonction de valeur d'action (Action Value Function)** : Cette fonction estime la valeur d'entreprendre une action spécifique dans un état donné.
 - Elle est généralement notée $Q(s,a)$, où s est l'état et a est l'action.
 - Elle indique à l'agent à quel point il est bénéfique de prendre une action spécifique lorsqu'il se trouve dans un état donné.

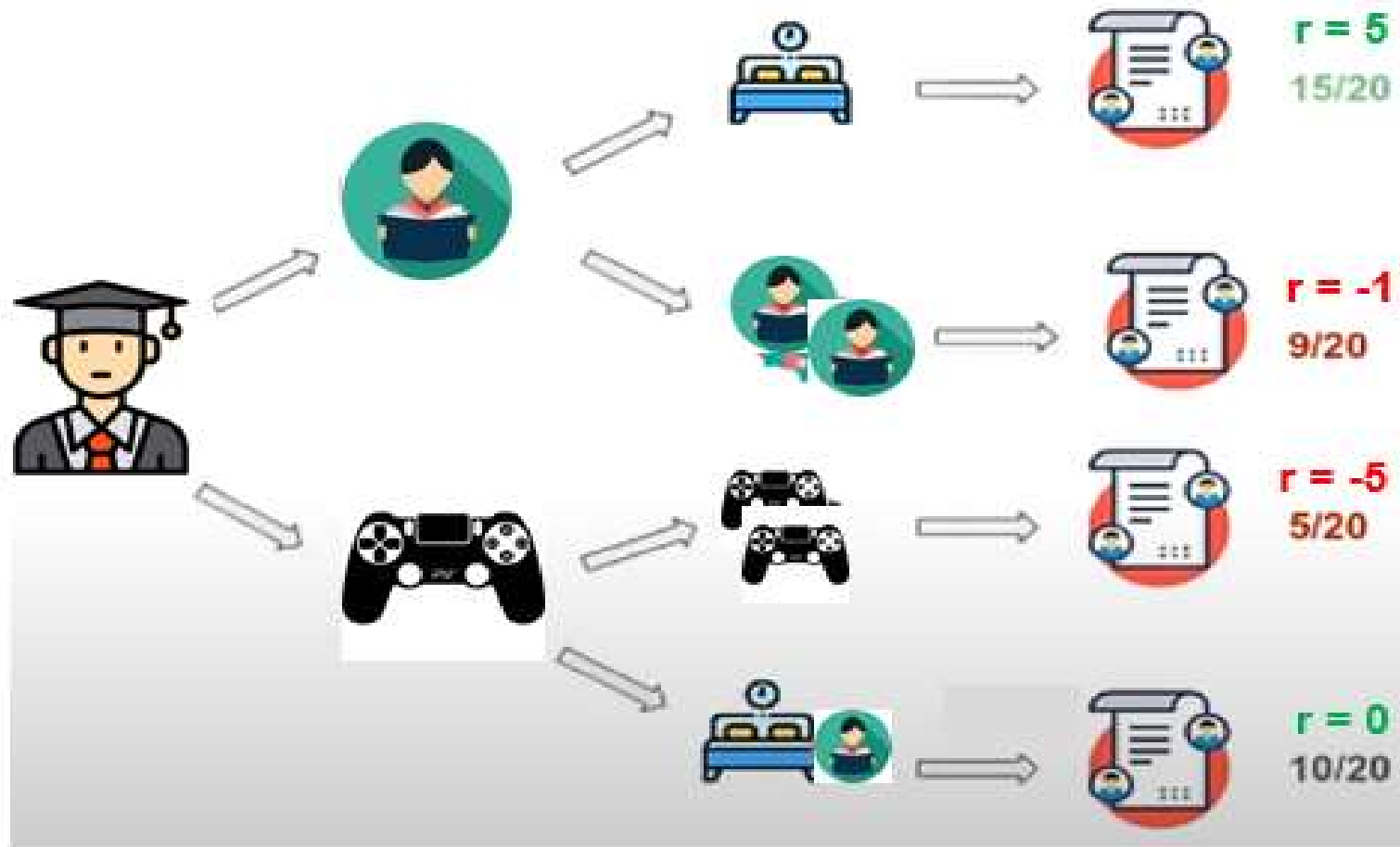


Utilité : Fonction de Valeur

- ▶ Ces fonctions de valeur sont utilisées par les algorithmes d'apprentissage par renforcement pour guider les actions de l'agent.
- ▶ **Exemple** : dans l'apprentissage par renforcement par programmation dynamique ou par des méthodes basées sur des réseaux de neurones comme les réseaux de neurones profonds (Q-Learning, Deep Q-Networks), ces fonctions sont itérativement mises à jour **pour s'approcher de la véritable valeur des états ou des actions**.
- ▶ Cela permet à l'agent d'apprendre à prendre des décisions optimales dans son environnement pour maximiser une récompense cumulative.



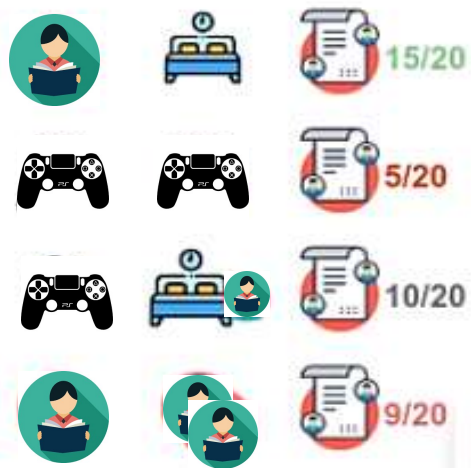
Exemple Illustratif : Valeur d'Etat



Il est à noter que les récompenses r arrivent après la fin de chaque épisode.

Autrement dit : après l'état final .


Les Différentes Expériences



Décomposer
Chaque Transition



(s: , s': , r: 0)



(s: , s': , r: 5)
15/20

(s: , s': , r: 0)

(s: , s': , r: -5)
5/20

(s: , s': , r: 0)

(s: , s': , r: 0)
10/20

(s: , s': , r: 0)

(s: , s': , r: -1)
9/20

Fonction de Valeur

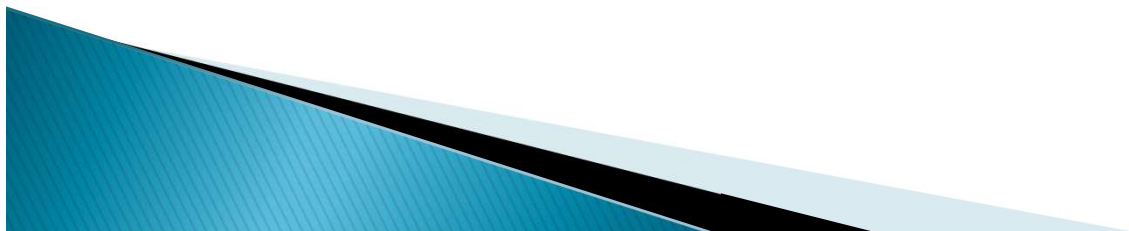
$$V(s) = V(s) + lr^*(V(s') - V(s))$$

► Avec :

$V(s)$: Valeur de l'état actuel.

$V(s')$: Valeur de l'état prochain.

lr : Valeur d'agrégation (Taux d'Apprentissage).



États & Transitions

$V(\text{diplôme}) = 0$	$V(\text{diplôme}) = 0$
$V(\text{jeu}) = 0$	15/20
$V(\text{jeu}) = 0$	$V(\text{diplôme}) = 0$
	10/20
$V(\text{lecture}) = 0$	$V(\text{diplôme}) = 0$
	9/20
$V(\text{sommeil}) = 0$	$V(\text{diplôme}) = 0$
	5/20
$V(\text{lecture}) = 0$	

Toutes les valeurs
sont initialisées à 0

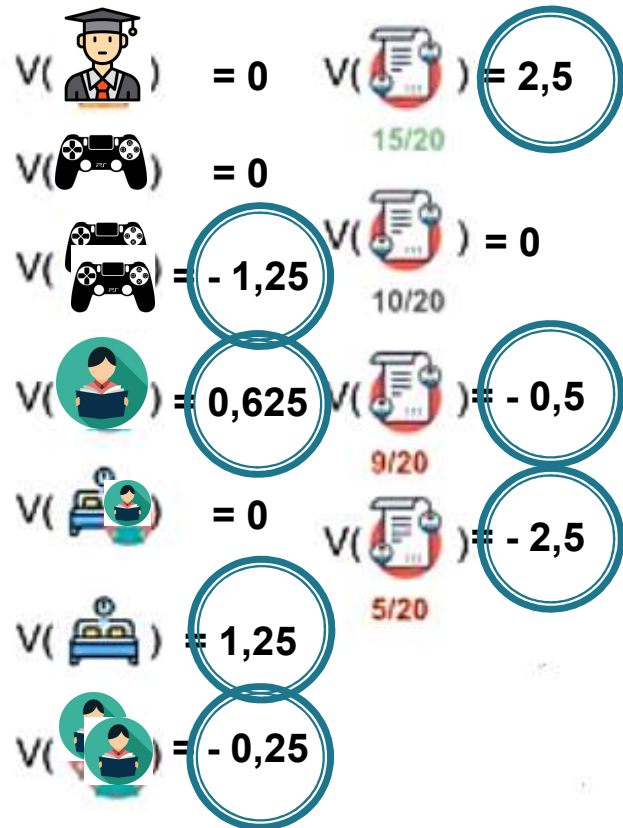


$(s: \text{lecture}, s': \text{sommeil}, r: 0)$
$(s: \text{sommeil}, s': \text{diplôme}, r: 5)$
15/20
$(s: \text{jeu}, s': \text{jeu}, r: 0)$
$(s: \text{jeu}, s': \text{diplôme}, r: -5)$
5/20
$(s: \text{jeu}, s': \text{sommeil}, r: 0)$
$(s: \text{sommeil}, s': \text{diplôme}, r: 0)$
10/20
$(s: \text{lecture}, s': \text{lecture}, r: 0)$
$(s: \text{lecture}, s': \text{diplôme}, r: -1)$
9/20

L'idée-Clé : Faire propager la récompense Finale.

➤ En revenant en arrière sur les états précédents.

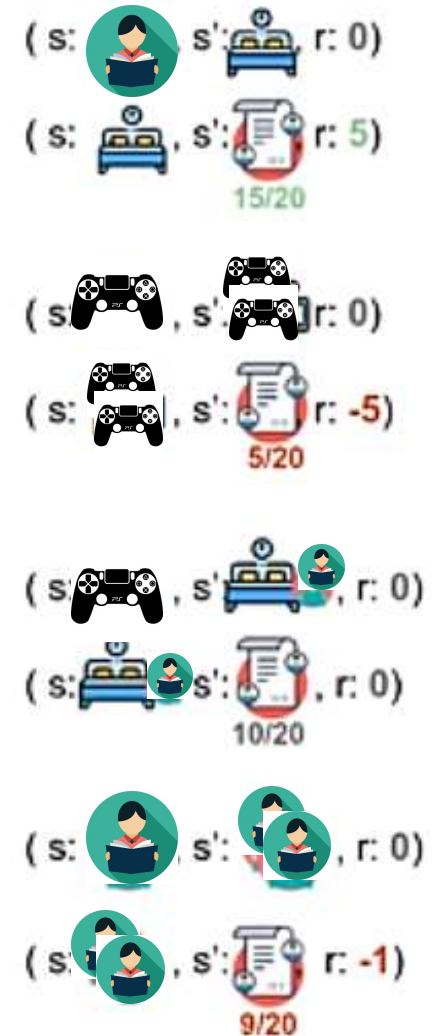
États & Transitions



Avec une 1^{ère}
Mise à jour



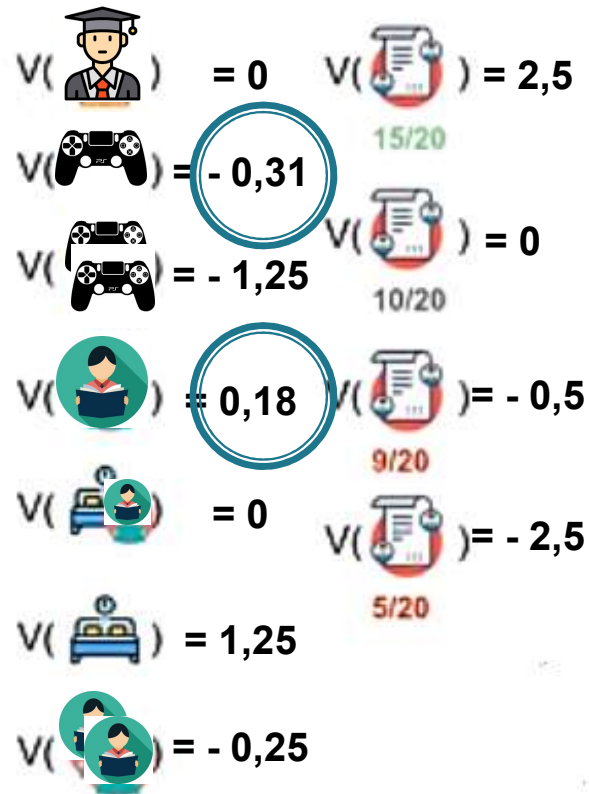
$lr = 0,5$



➤ En utilisant la formule précédente :

$$V(s) = V(s) + lr * (V(s') - V(s))$$

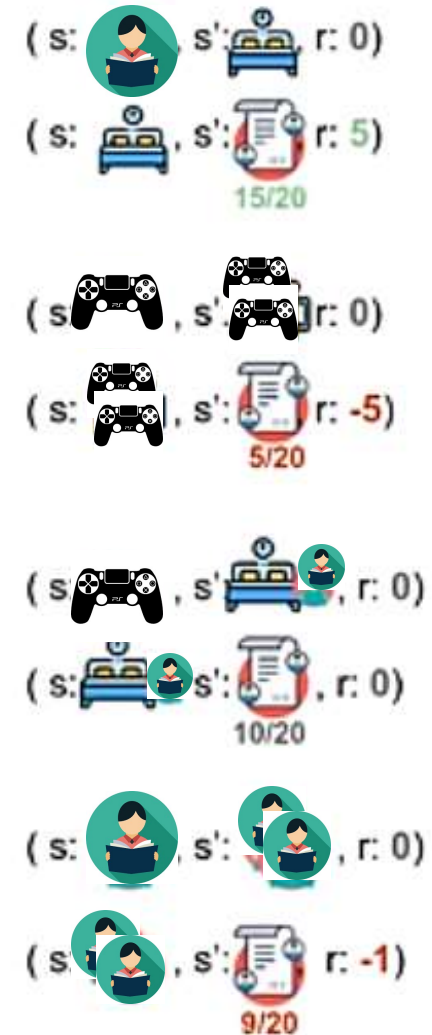
États & Transitions



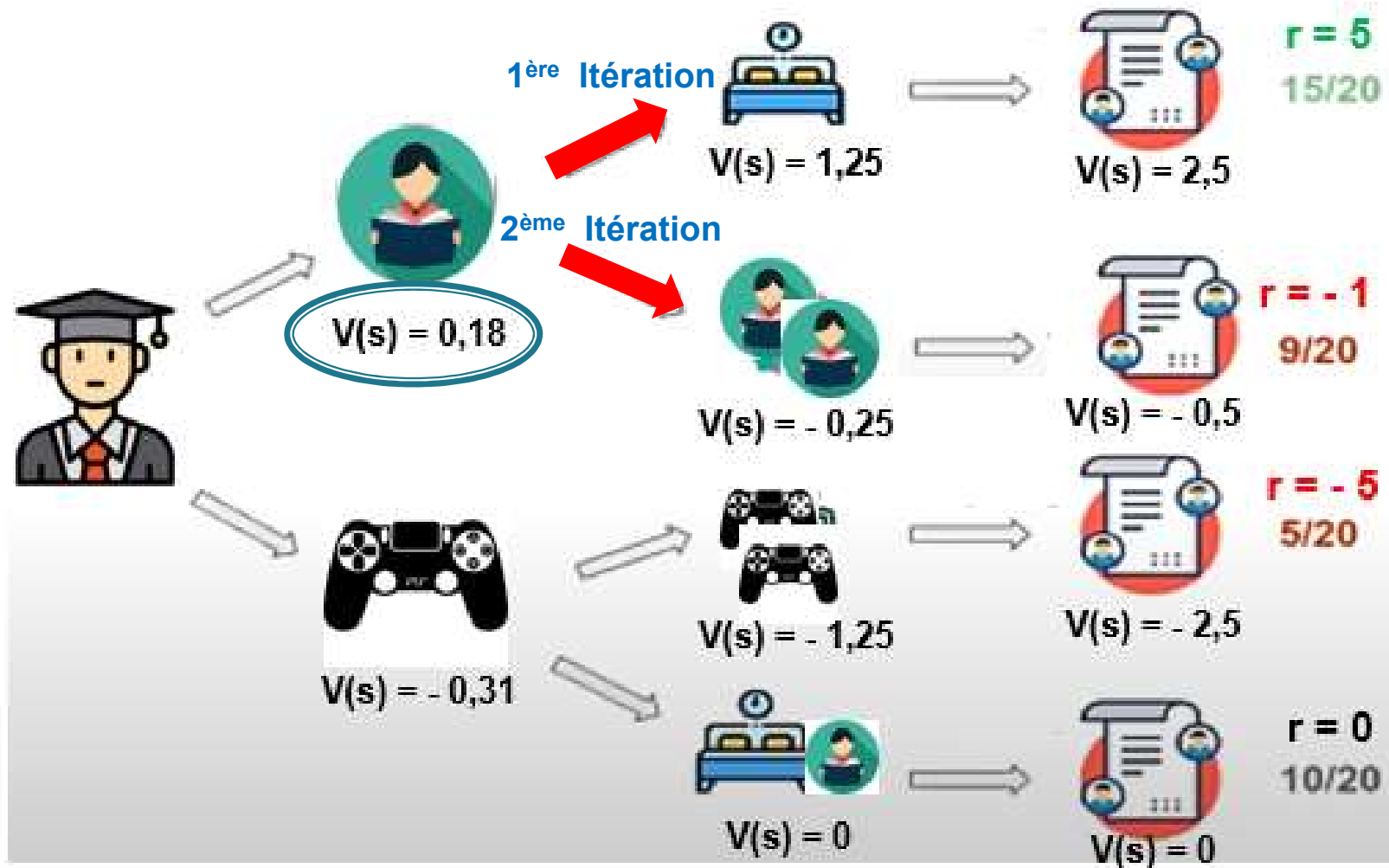
Avec une 2^{ème}
Mise à jour



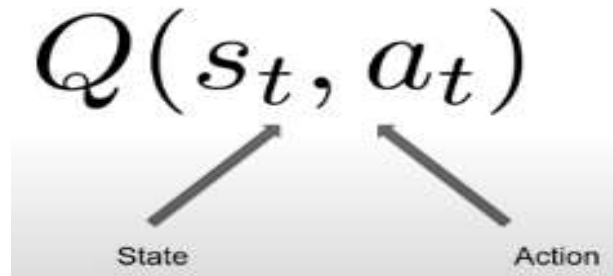
$$V(s) = V(s) + \alpha r + \gamma V(s') - V(s)$$



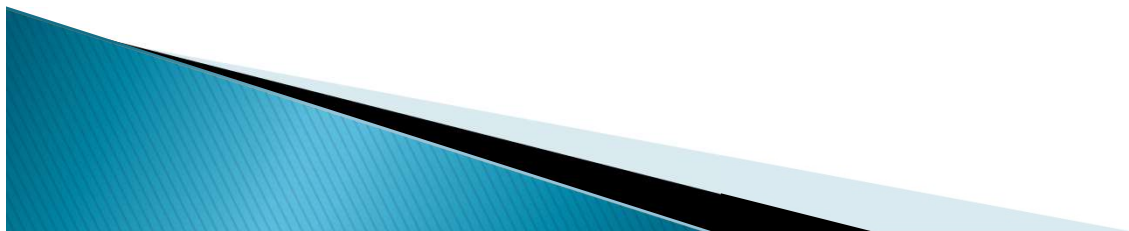
Résultat Final



Q-Fonction



- ▶ Elle ne prend pas uniquement l'état où on se trouve, mais elle prend également l'action qu'on veut effectuer.
- ▶ C-à-d : si je suis dans l'état s , à l'instant t , et que je prend l'action a : Quel est le nombre de récompenses que je pourrai avoir dans le futur ?



Q-Fonction

- ▶ On peut revoir cette espérance à travers la formule suivante :

$$Q(s_t, a_t)^\pi = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t, a_t]$$

Avec :

st : l'état à l'instant *t*.

at : l'action à l'instant *t*.

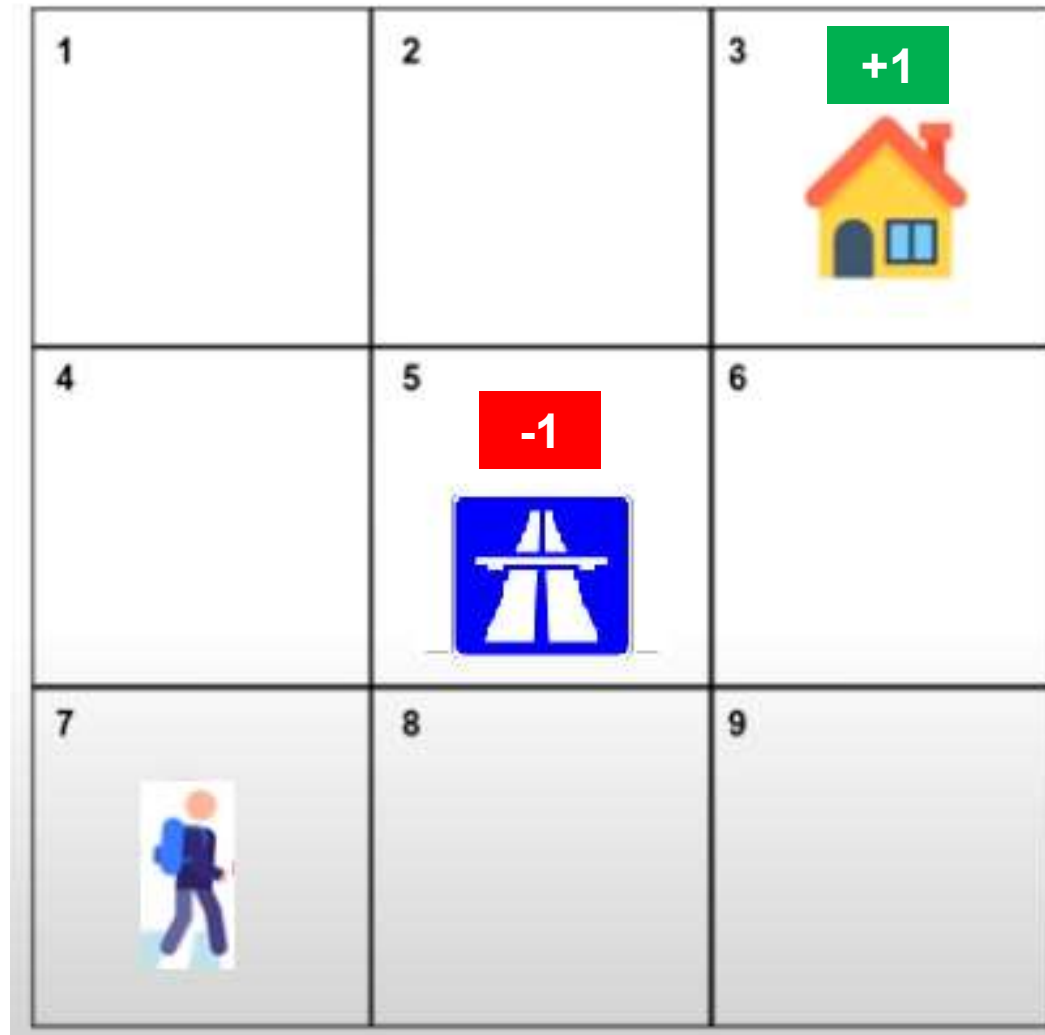
Rt : la récompense à l'instant *t*.

γ: valeur arbitraire; l'importance de la prochaine récompense.
(Favoriser les récompense les plus proches).

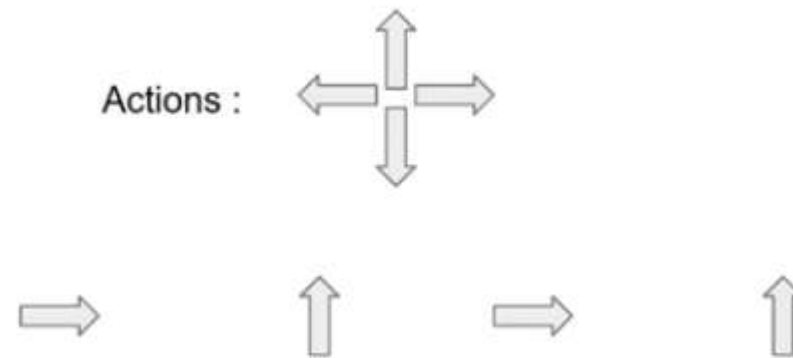
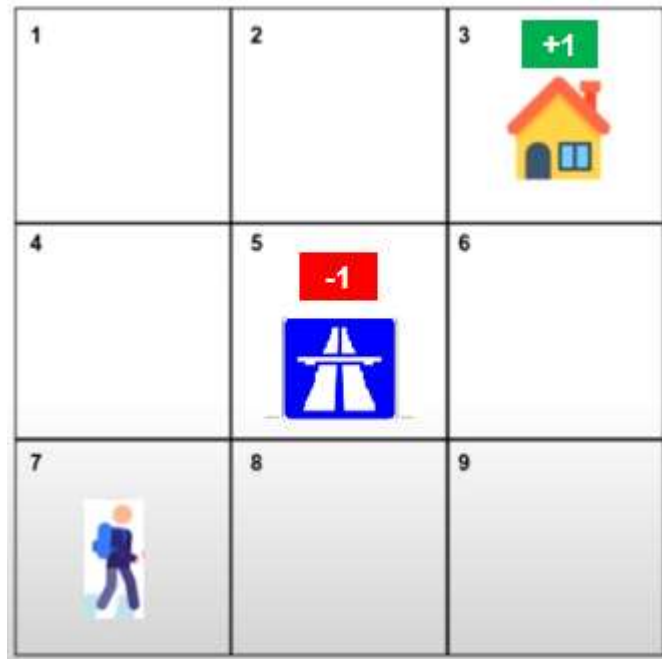
π : la politique choisie.



Exemple Illustratif : Q-Fonction



Valeur Q-Fonction

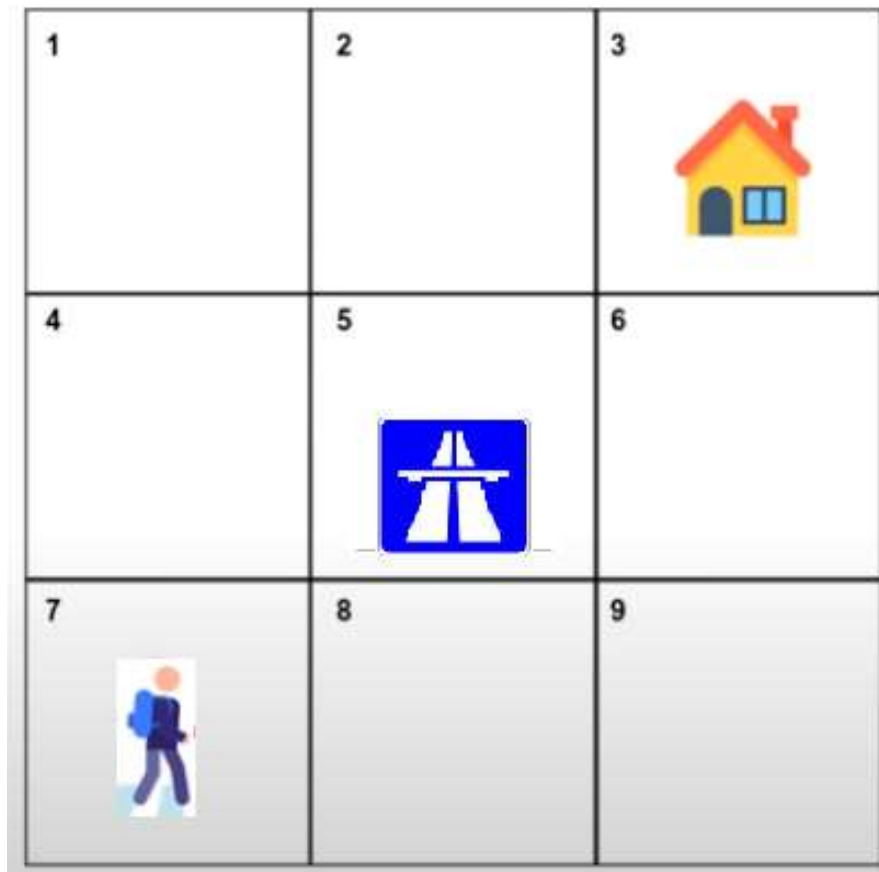


Supposons que le parcours soit :

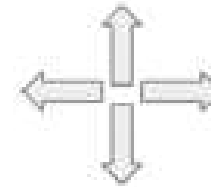
s8, s5, s6, s3

- **L'idée** : est de toujours tirer les récompenses existantes, et de les faire propager vers les états précédents, tout en appliquant plusieurs itérations, jusqu'à la fin de tous les épisodes.

Valeur Q-Fonction



Actions :



$$R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4}$$
$$0 + (-1) + (0) + 1$$

$$R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4}$$
$$0 + (-0.9) + 0 + 0.72$$

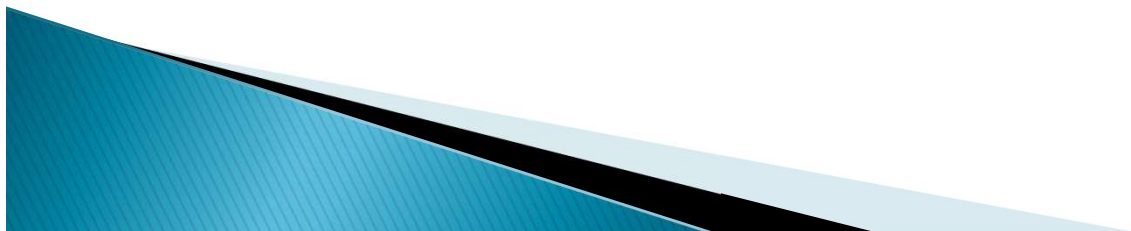
- ▶ En choisissant la valeur arbitraire $\gamma = 0,9$.
- ▶ Dont le but est de favoriser les récompenses les plus proches.

Fonction de Bellman

- ▶ On peut reformuler cette espérance à travers une fonction récursive :

$$Q(s_t, a_t)^\pi = r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})^\pi$$

- ▶ On essaye de choisir l'action a qui maximise notre Q-Fonction, et selon une politique π (exemple : ϵ -Greedy).
- ▶ Il s'agit d'une formule équivalente à la précédente, juste qu'elle est définie d'une manière récursive.



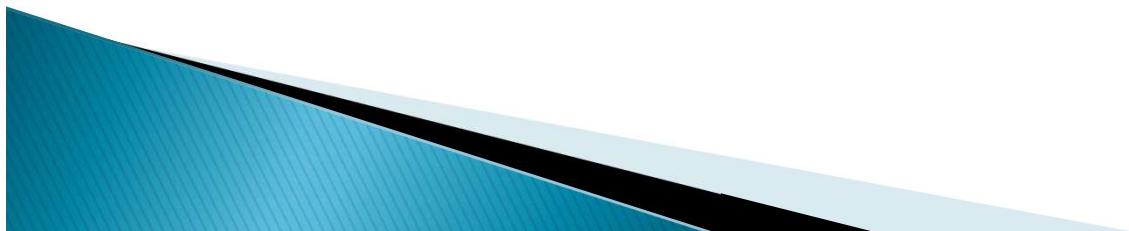
Formule Finale de Mise à Jour

- ▶ La formule finale de mise à jour de la table Q, sera définie comme suit :

$$Q(s_t, a_t)_{new} = Q(s_t, a_t)_{old} + \alpha[r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)_{old}]$$

De cette manière et selon l'exemple illustratif précédent :

- ▶ **L'idée-Clé :** Les deux récompenses **(+1)** et **(-1)** vont se propager au fur et à mesure, selon les différentes itérations faites.
- ▶ Jusqu'à l'épuisement de tous les scénarios possibles.



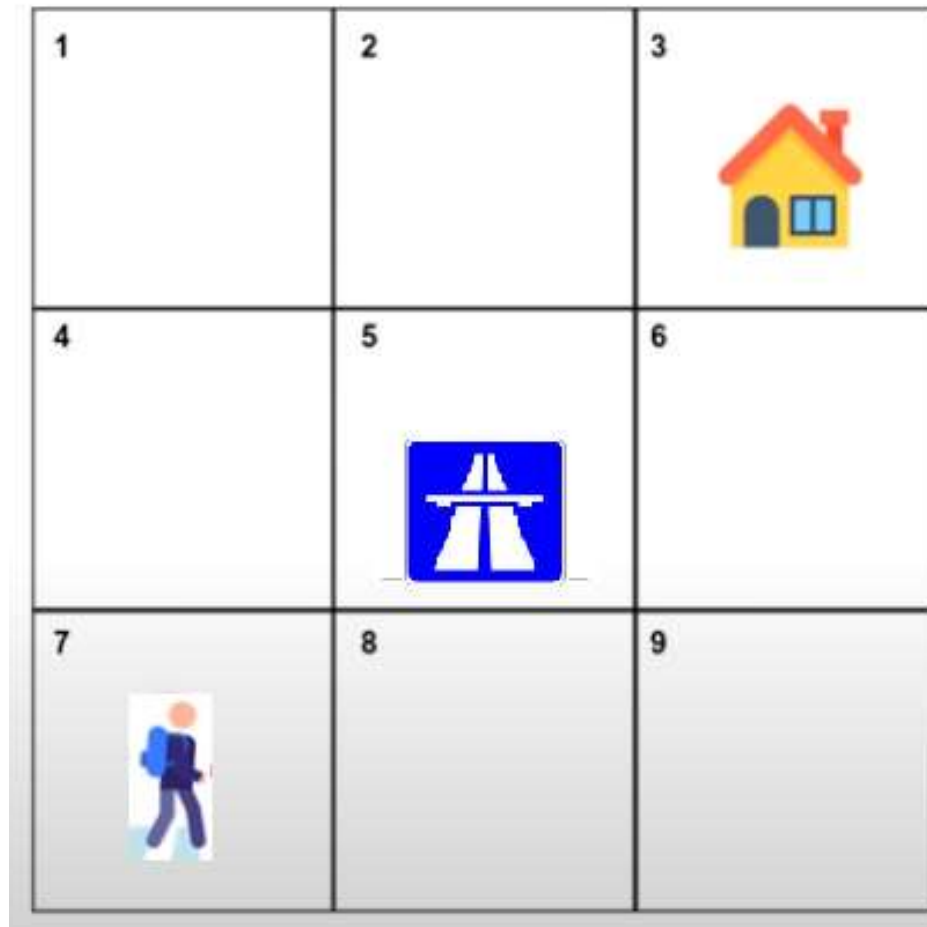
Application de l'Algorithme Q-Learning

- ▶ En appliquant la précédente formule dans un algorithme d'apprentissage par renforcement, à savoir l'algorithme de Q-Learning :

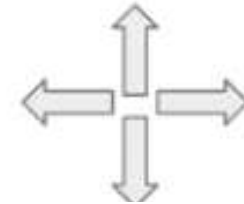
1. **Initialisation de la table Q.**
2. **Choix de l'action :** Nbre Aléatoire, ϵ , Exploitation ou Exploration, Fonction-Q.
3. **Exécution de l'action et observation de la récompense du prochain état.**
4. **Mise à jour de la table Q.**
5. **Répéter les étapes 2 et 4.**
6. **Convergence.**



Valeur Function-Q : Initialisation



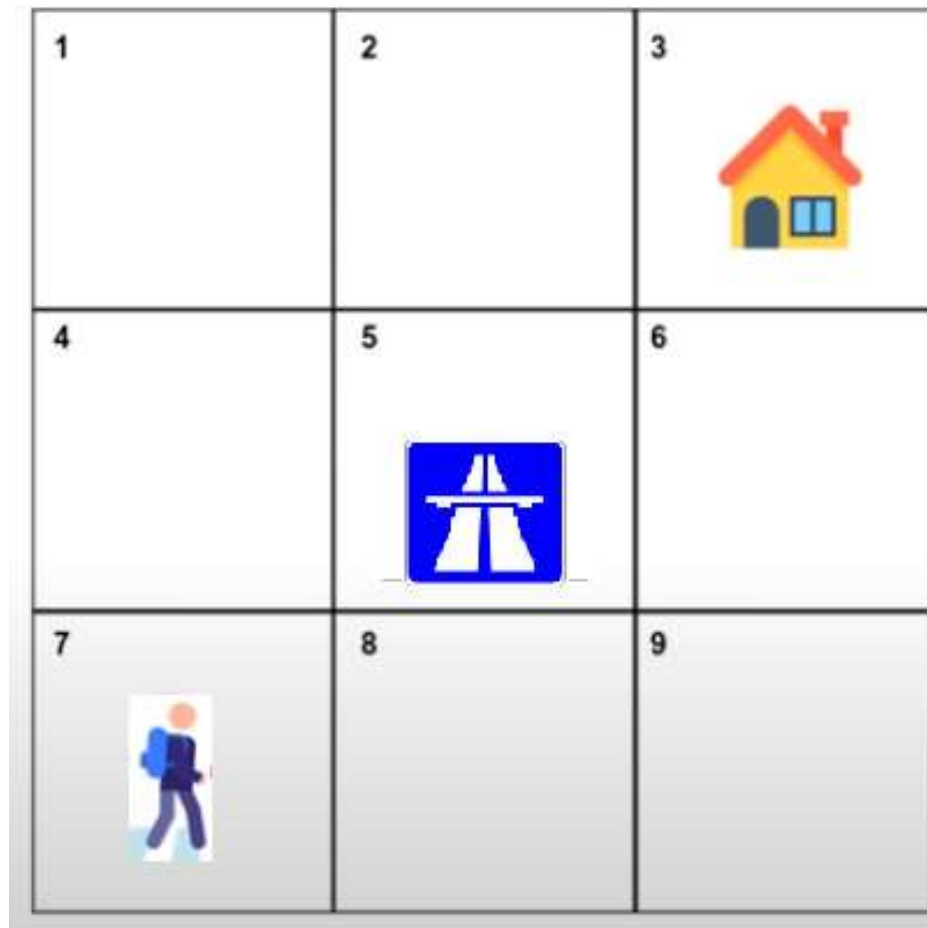
Actions :



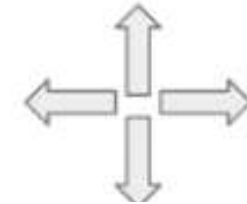
Q-Table:

1	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
2	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
3	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
4	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
5	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
6	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
7	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
8	↑ 0.00	↓ 0.00	← 0.00	→ 0.00
9	↑ 0.00	↓ 0.00	← 0.00	→ 0.00

Valeur Function-Q : Mise à Jour



Actions :



Q-Table:

1	↑ 0,00	↓ 0,21	← 0,00	→ 0,83
2	↑ 0,00	↓ -0,56	← 0,11	→ 0,92
3	↑ 0,00	↓ 0,00	← 0,00	→ 0,00
4	↑ 0,26	↓ 0,29	← 0,00	→ -0,83
5	↑ 0,75	↓ 0,14	← 0,27	→ 0,72
6	↑ 0,92	↓ 0,38	← -0,65	→ 0,00
7	↑ 0,29	↓ 0,00	← 0,00	→ 0,33
8	↑ -0,88	↓ 0,00	← 0,32	→ 0,42
9	↑ 0,56	↓ 0,00	← 0,30	→ 0,00

The background of the slide is a light gray gradient. It features a complex, multi-colored circuit board pattern in red, blue, and green. A prominent silhouette of a human brain is formed by a dense network of blue circuit lines on the right side of the slide. The text "Merci pour votre attention .." is centered in a bold, blue, italicized serif font.

***Merci pour votre
attention ..***