Cours-TD d'introduction à l'Intelligence Artificielle

Partie VII

L'apprentissage par renforcement

Simon Gay

Introduction à l’Intelligence Artificielle

- **Menu :**

  - Théorie :
    - Principe de l’apprentissage par renforcement
    - Quelques notions et principes essentiels
    - Le Q-learning
    - Exemple d’apprentissages par renforcement

  - Pratique :
    - Jouons à l’Hexapion !

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Les IA numériques/connexionnistes : un petit rappel du premier cours**

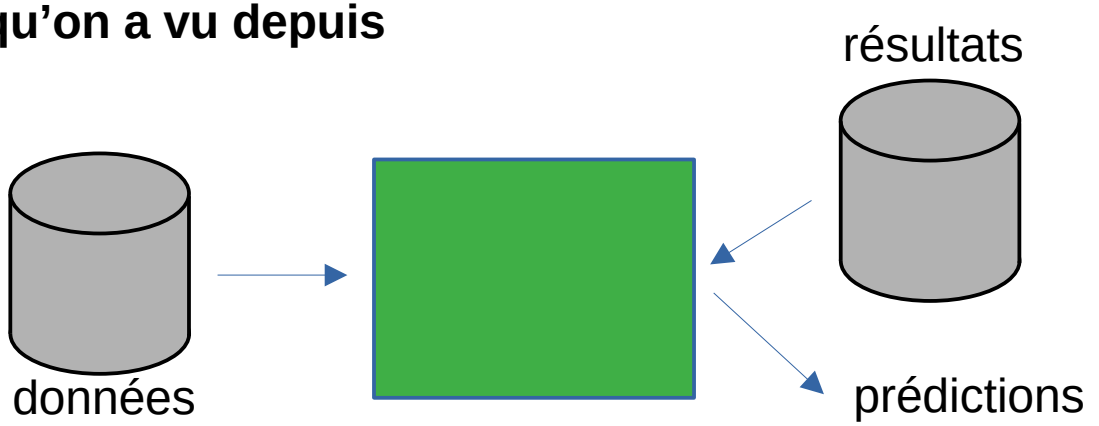
  - **Apprentissage supervisé :** modèle d'apprentissage où on connaît le résultat escompté. L'entraînement se fait sur un jeu d'essais dont les résultats sont connus, puis on exploite le système avec des entrées dont on ne connaît pas le résultat.

  - **Apprentissage non-supervisé :** modèle d'apprentissage qui ne nécessite pas de connaître les résultats du jeux d'essais. Le système doit converger vers une solution optimale (au moins localement)

  - **Apprentissage par renforcement :** le système peut agir sur son environnement et est activement impliqué dans son processus d'apprentissage. Le système effectue des actions et apprend des résultats obtenus.

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Les IA numériques : avec ce qu'on a vu depuis**
  - **Apprentissage supervisé :**

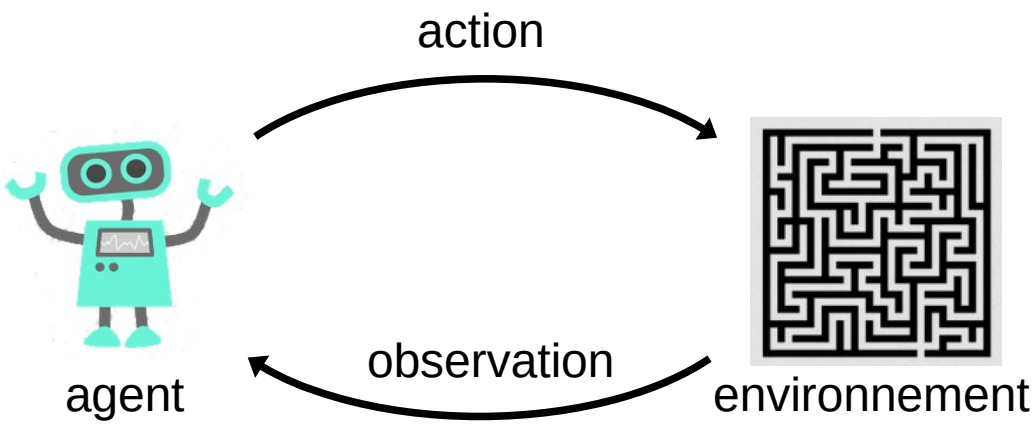


  - **Apprentissage non-supervisé :**

  - **Ces modèles d'IA ne sont au final que des fonctions qui convertissent des données d'entrée en valeurs de sortie**

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Les IA numériques : avec ce qu'on a vu depuis**
  - **Apprentissage par renforcement :**
    - Interaction entre un agent et un environnement
      - L'agent effectue des actions sur l'environnement
      - L'action produit un changement sur cet environnement
      - L'agent apprend de ce changement



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Les IA numériques : avec ce qu'on a vu depuis**

  - **Apprentissage par renforcement :**

    - Premiers travaux sur l'apprentissage artificiel dès les années 50

      - Inspirés des travaux des neurologues et biologistes
        - Règle de Hebbs
        - Modèles de conditionnement animal (Pavlov)

      → idée qu'il faut répéter des actions pour renforcer et favoriser certains comportements.

    - Il faudra attendre la fin des années 80 pour que le concept arrive à maturité.

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

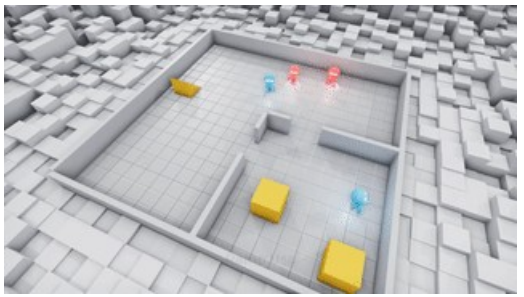
- **Les IA numériques : avec ce qu'on a vu depuis**

  - **Apprentissage par renforcement :**

    - Cette interaction permet à un agent d'évoluer en autonomie dans son environnement (réel ou virtuel)
    - On dépasse ainsi le cadre d'utilisation des IA supervisées et non-supervisées

Learning to Walk via Deep Reinforcement Learning, T. Haarnoja et al., 2019

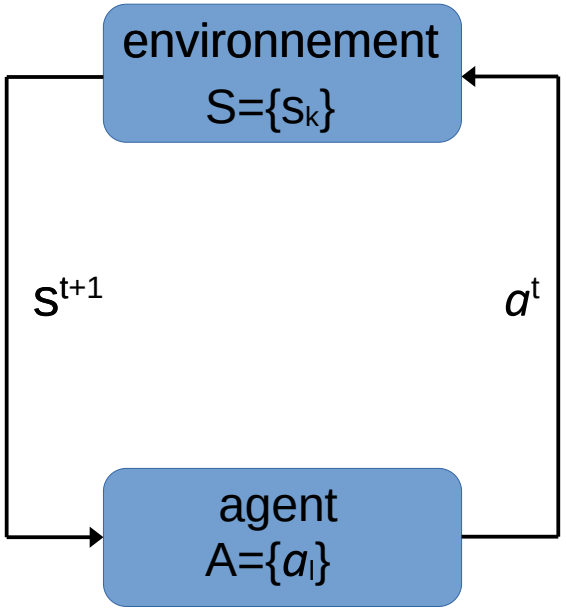
Control of a Quadrotor with Reinforcement Learning, J. Hwangbo et al., 2017



Emergent Tool Use From Multi-Agent Autocurricula, B. Baker et al., 2019

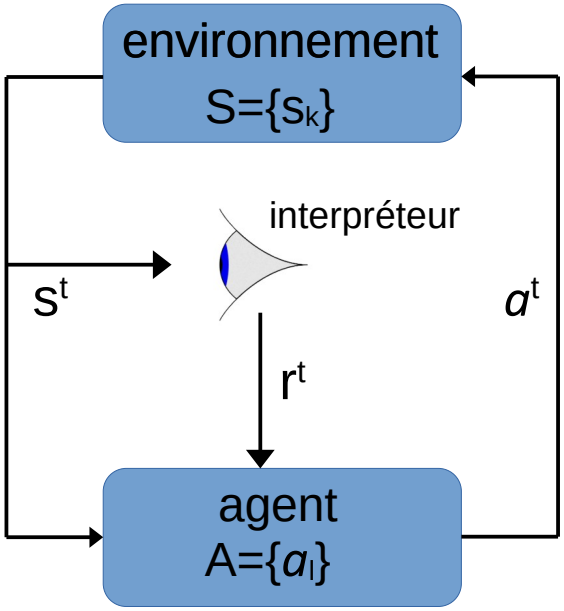
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - L'environnement est perçu sous la forme d'<u>états</u>
    - Un état est une valeur ou un symbole qui caractérise la situation de l'agent. L'ensemble des états forme l'ensemble S (States)
  - À partir de sa perception, l'agent sélectionne une action *a* à effectuer.
    - Les actions permettent à l'agent d'agir sur son environnement. On note en général A l'ensemble des actions possibles.
  - Suite à cette action, l'environnement passe de l'état S^t à l'état S^{t+1}



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - Mais pour l'agent, les actions et les états n'ont aucune signification *a priori*
  - Comment choisir l'action si tout se vaut ?
  - Notion de récompense :
    - Un interpréteur externe va attribuer une récompense numérique en fonction de l'état atteint
  - L'agent va chercher à maximiser la récompense à moyen ou long terme
    → problème général du RL
  - On va chercher à définir une politique (policy)
    $\pi : s \rightarrow a$ pour maximiser le gain total



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

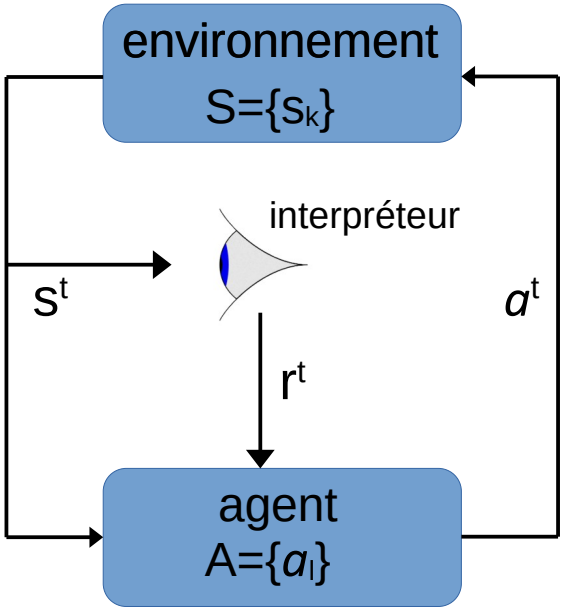
- **Apprentissage par renforcement :** un peu de vocabulaire
  - Deux types de politiques :
  - Déterministe:

    $\pi : S \rightarrow a$

    - À un instant donné et dans un instant donné, l'action sélectionnée sera toujours la même
  - Stochastique :

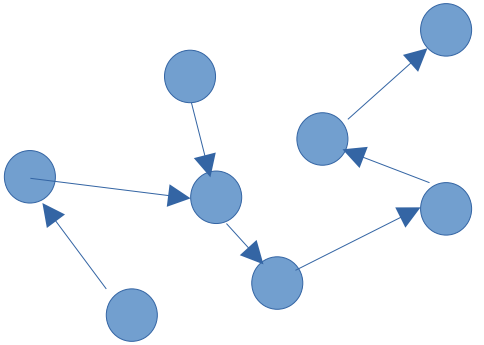
    $\pi : S \times A \rightarrow [0,1]$

    - La politique donne la probabilité de choisir une action *a* dans le contexte d'un état s.
    - Toute action peut être sélectionnée tant que $\pi(s,a)>0$

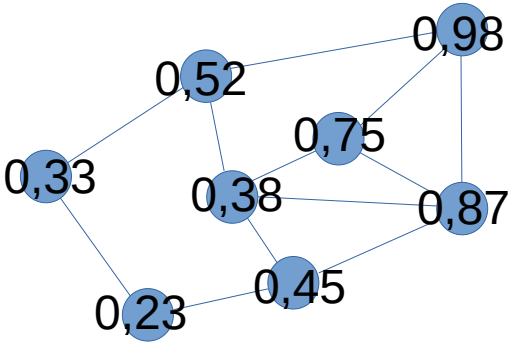


Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** un peu de vocabulaire
  - Deux types d'algorithmes :
  - Si l'apprentissage consiste à apprendre une fonction $\pi : S \rightarrow a$ ou $\pi : S \times A \rightarrow [0,1]$, l'algorithme est dit *Policy-based*



  - Si l'apprentissage consiste à attribuer et modifier des valeurs sur les états, l'algorithme est dit value-based



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** un peu de vocabulaire

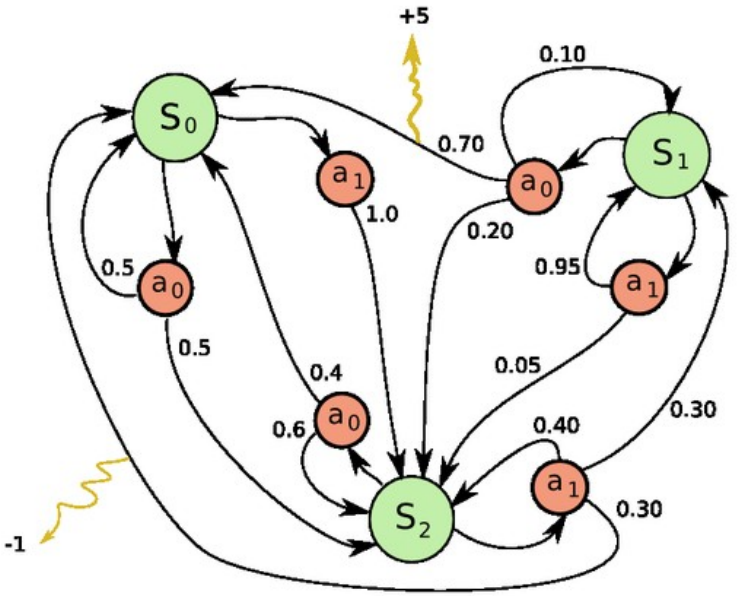
  - Ce type de modèle est appelée Processus de Décision Markovien (MDP : Markov Decision Process)

  - Un MDP est un tuple {S, A, T, R } où :

    - S est l'ensemble des états de l'environnement
    - A est l'ensemble des actions de l'agent
    - T est la fonction donnant, pour chaque action a de l'agent dans un état S, la probabilité de passer dans un état s'

      P (s' | s, a)

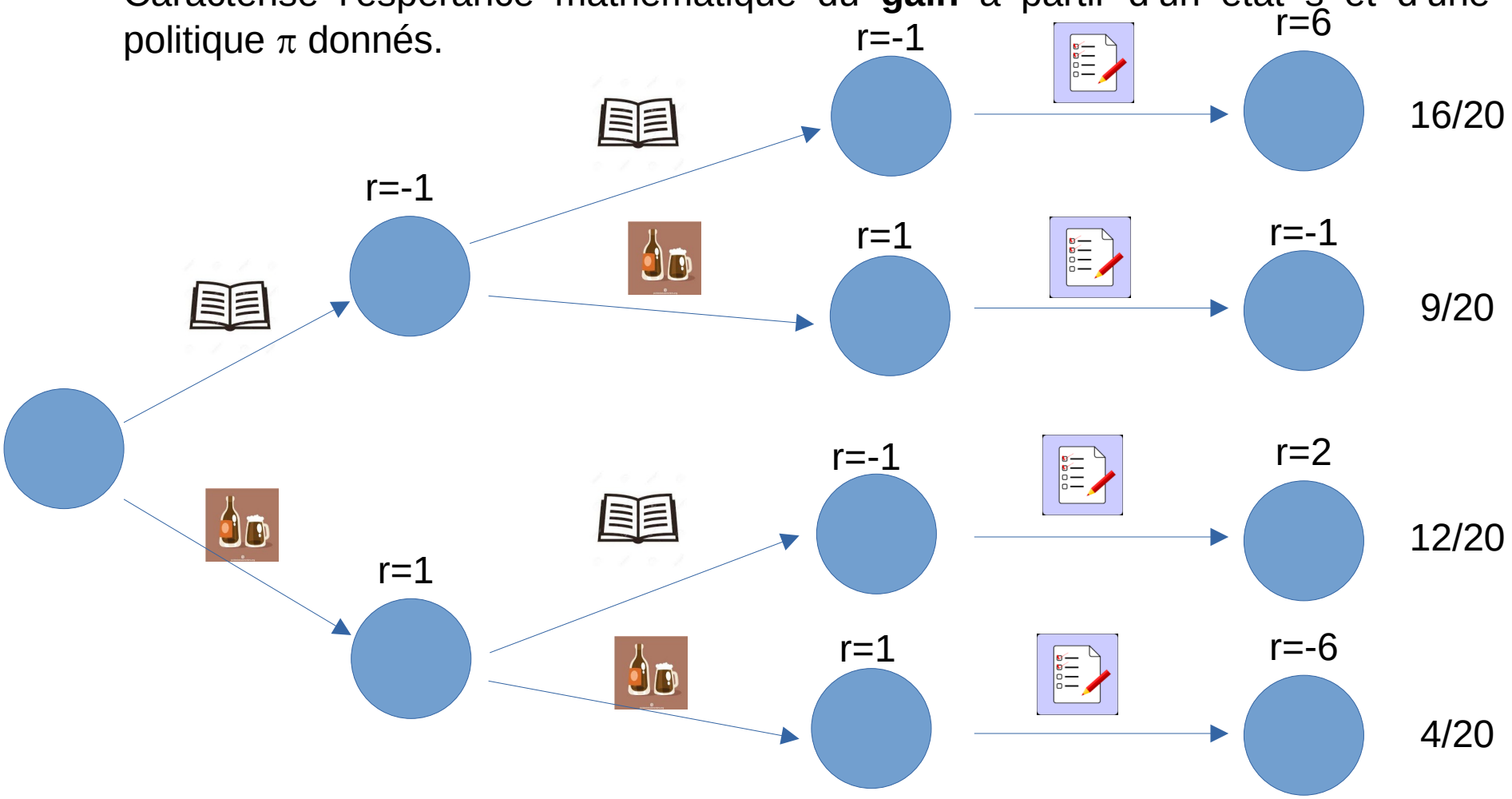
    - R est la fonction qui attribue les récompenses



  - Il existe une version du MDP où l'agent ne connaît pas avec certitude l'état actuel : les Processus de Décision Markovien Partiellement Observables (POMDP)

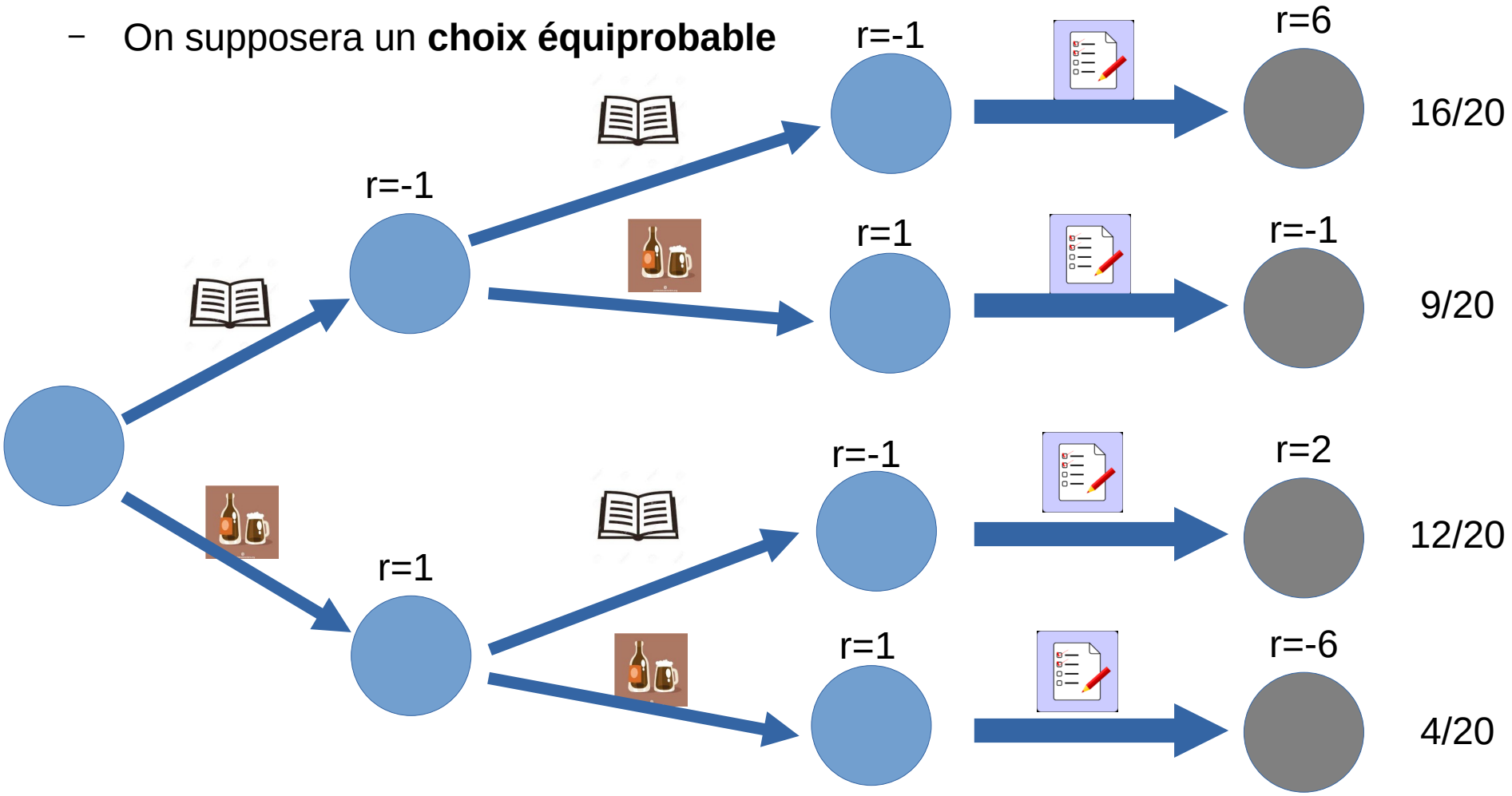
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** La *value function*
  - Caractérise l'espérance mathématique du **gain** à partir d'un état s et d'une politique π donnés.



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

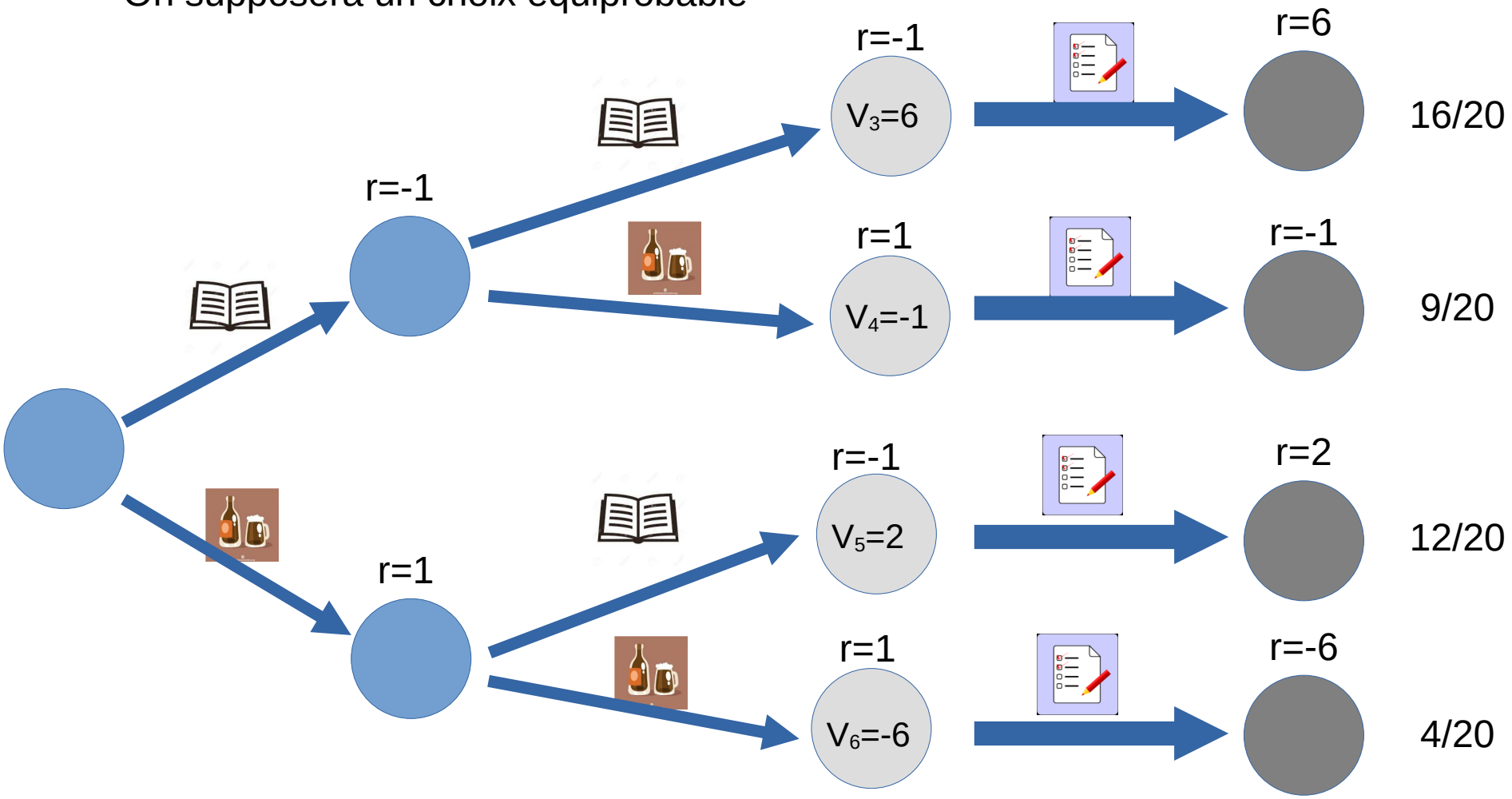
- Value function : Espérance mathématique à partir d'un état s et d'une politique π donnés.
  - On supposera un **choix équiprobable**



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

$V_3 = 1x\ 6$
$V_4 = 1x\ -1$
$V_5 = 1x\ 2$
$V_6 = 1x\ -6$

- Value function : Espérance mathématique à partir d'un état s donné.
  - On supposera un choix équiprobable

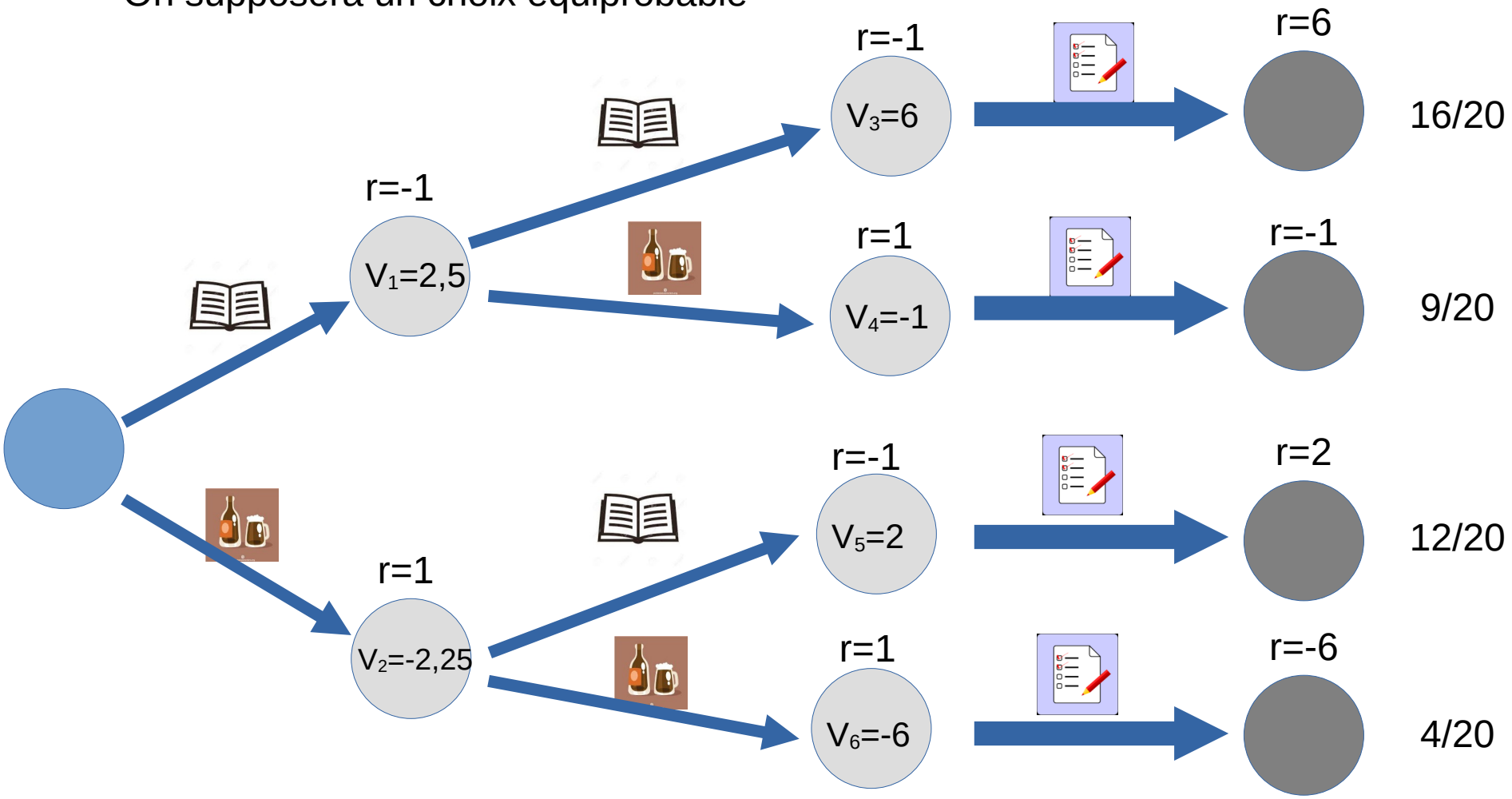


Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

$V_1 = 0{,}5x(-1+V_3) + 0{,}5x(1+V_4)$

$V_2 = 0{,}5x(-1+V_5) + 0{,}5x(1+V_6)$

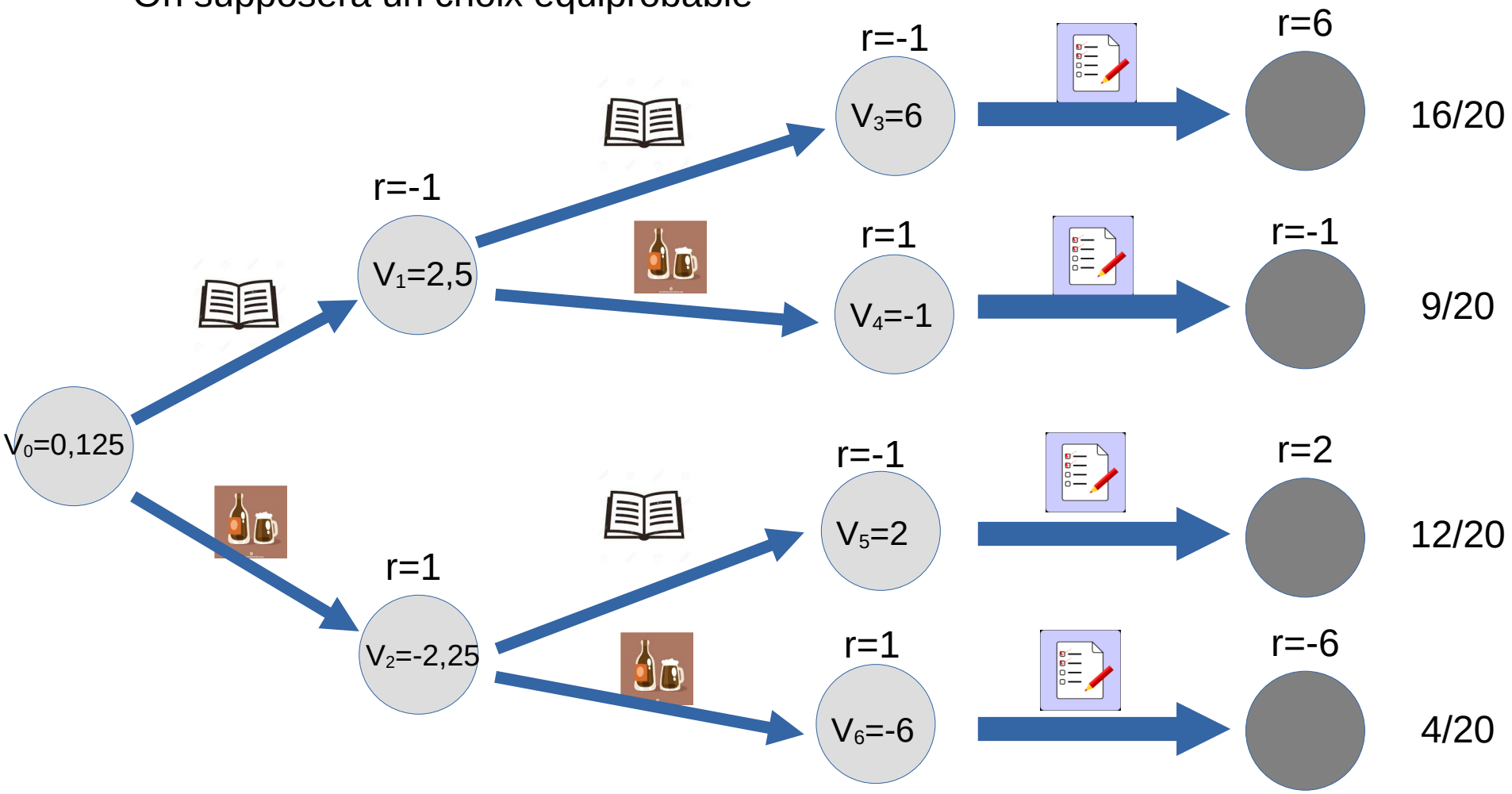
- Value function : Espérance mathématique à partir d'un état s donné.
  - On supposera un choix équiprobable



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

$V_0 = 0{,}5x(-1+V_1) + 0{,}5x(1+V_2)$

- Value function : Espérance mathématique à partir d'un état s donné.
  - On supposera un choix équiprobable



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- Calcul de l'espérance mathématique à partir d'un état s donné :
  - Soit une séquence [s_0, s_1, s_2, s_3] donnée (avec s_3 un état terminal) :
    - $G_{0,3} = r_{0,1} + r_{1,2} + r_{2,3} = r_{0,1} + G_{1,3}$

    - $V(S_2) = E(G_{2,3} \mid s_2, \pi) = \sum_{s_i \mid s_2} P(s_i \mid s_2) \cdot (r_{s_2, s_i})$
    - $V(S_1) = E(G_{1,3} \mid s_1, \pi) = \sum_{s_i \mid s_1} P(s_i \mid s_1) \cdot (r_{s_1, s_i} + \sum_{s_j \mid s_i} P(s_j \mid s_i) \cdot (r_{s_i, s_j}))$

      $= \sum_{s_i \mid s_1} P(s_i \mid s_1) \cdot (r_{s_1, s_i} + V(s_i))$
    - $V(S_0) = E(G_{0,3} \mid s_0, \pi) = \sum_{s_i \mid s_0} P(s_i \mid s_0) \cdot (r_{s_0, s_i} + \sum_{s_j \mid s_i} P(s_j \mid s_i) \cdot (r_{s_i, s_j} + \sum_{s_k \mid s_j} P(s_k \mid s_j) . (r_{s_j, s_k})))$

      $= \sum_{s_i \mid s_0} P(s_i \mid s_0) \cdot (r_{s_0, s_i} + V(s_i))$

$$V^{\pi}(s) = \sum_{s_i} P(s_i \mid s, \pi) \cdot (r_{s, s_i} + V^{\pi}(s_i))$$

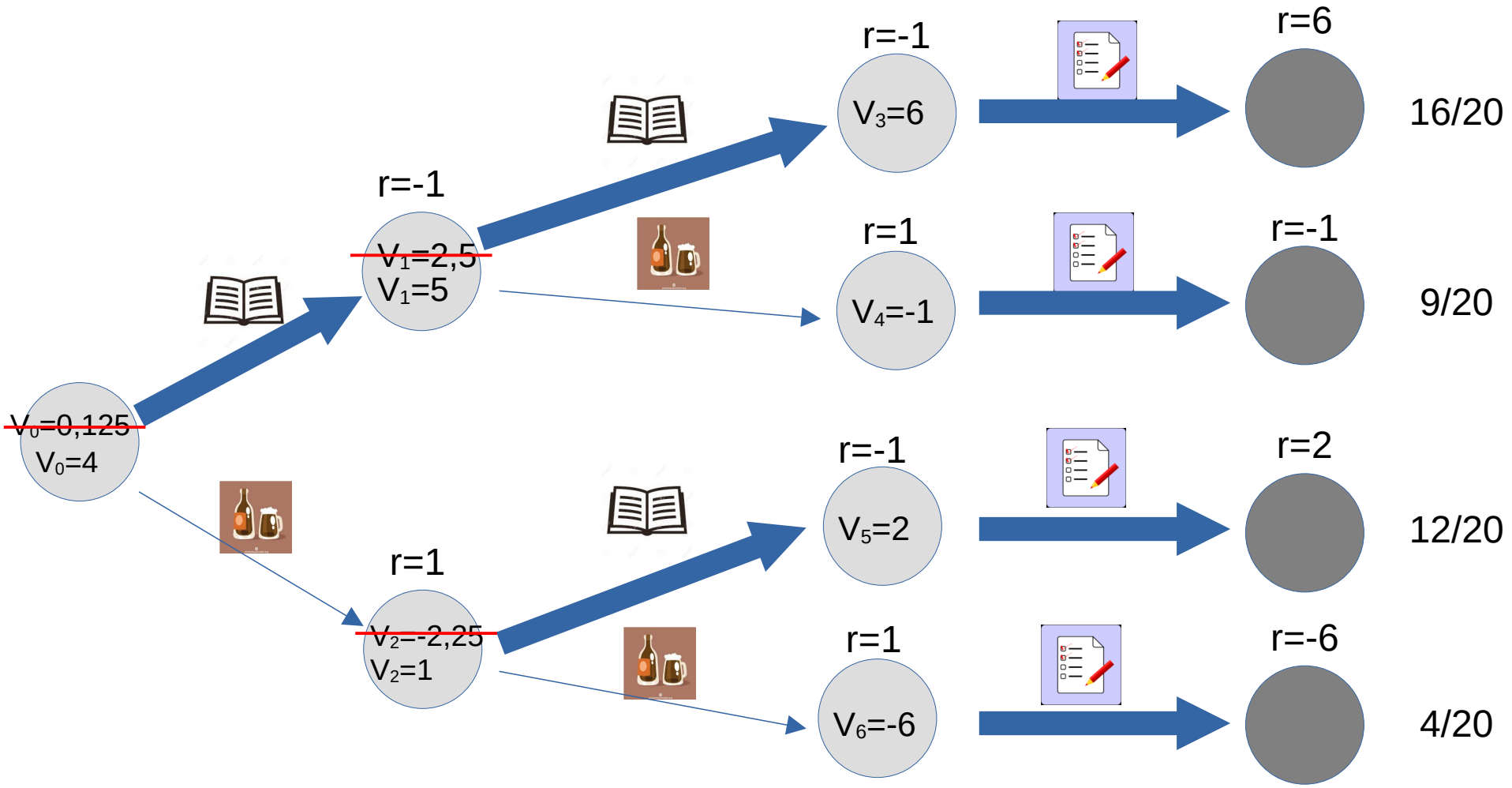
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- Espérance mathématique à partir d'un état s donné.
  - Comment mettre à jour les poids essais après essais ?
    - Avec un *learning rate* !
    - $V(s_i) \leftarrow (1 - \alpha) . V(s_i) + \alpha . [r_{i, i+1} + V(s_{i+1})]$
    - Fonction de mise à jour de la Value function :

$$\mathbf{V(s_i) \leftarrow V(s_i) + \alpha . [r_{i, i+1} + V(s_{i+1}) - V(s_i)]}$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- On se concentre maintenant sur l'action qui mène au 'meilleur' état



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- Calcul de l'espérance mathématique à partir d'un état S donné : cas choix du maximum

  - Soit une séquence [s_0, s_1, s_2, s_3] donnée (avec s_3 un état terminal) :
    - On choisi toujours l'état suivant avec la meilleure valeur $V^*(S) = E(G|s, \pi^*)$
      - Dans l'état s_2, s_3 était l'état avec la plus forte value function

$$G_{2,3} = \max_{s_i|s_2}(G_{2,i}) = \max_{s_i|s_2}(r_{s_2,s_i}) \qquad V^*(s_2) = \max_{s_i|s_2}(r_{s_2,s_i}) = r_{s_2,s_3}$$

      - Dans l'état s_1, s_2 était l'état avec la plus forte value function

$$G_{1,3} = \max_{s_i,s_j|s_1}(G_{1,j}) = \max_{s_i|s_1}(r_{s_1,s_i} + \max_{s_j|s_i}(G_{i,j})) \qquad V^*(S_1) = \max_{S_i|S_1}(r_{S_1,S_i} + V^*(S_i))$$

      - Dans l'état s_0, s_1 était l'état avec la plus forte value function

$$G_{0,3} = \max_{s_i,s_j,s_k|s_0}(G_{0,k}) = \max_{s_i|s_0}(r_{s_0,s_i} + \max_{s_j,s_k|s_i}(G_{i,k})) = \max_{s_i|s_0}(r_{s_0,s_i} + \max_{s_j|s_i}(r_{s_i,s_j} + \max_{s_k|s_j}(G_{j,k})))$$

$$V^*(S_0) = \max_{S_i|S_0}(r_{S_0,S_i} + V^*(S_i))$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- Calcul de l'espérance mathématique à partir d'un état s donné : cas choix du maximum

  - Si on choisi la valeur maximale parmi les états joignables possibles, la value function fait 'remonter' récursivement les valeurs maximales

$$V^*(s) = \max_{s'|s} \left(r_{s,s'} + V^*(s') \right)$$

    - Dans un environnement déterministe (P(s' | s,a) = 1 , pour tout (s,a,s'))

$$V^*(s) = \sum_a P(s'|s,a) \cdot \max_{s'|s} \left(r_{s,s'} + V^*(s') \right)$$

    - Dans un environnement non déterministe

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

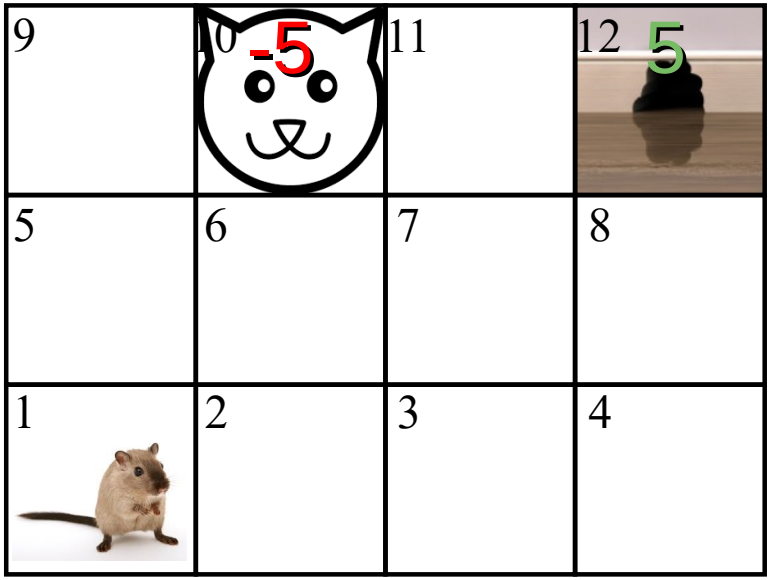
- La façon de choisir ses action a donc une influence sur les valeurs des états
  - Un choix équiprobable fait tendre les valeurs vers une espérance mathématique
    - Problème : cette espérance ne montre pas forcément les chemins les plus efficaces.
  - Choisir à chaque fois la meilleure possibilité fait converger les valeurs vers la plus grande récompense que l'on peut obtenir, en renforçant le 'meilleur chemin' découvert.
    - Problème : les valeurs vont renforcer le premier chemin 'positif', au détriments d'autres chemins plus efficaces non encore découverts.
  - Cet équilibre entre **exploration** et **exploitation** est une des questions fondamentales de l'apprentissage par renforcement.

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- Une approche parmi les plus utilisées : le ε-greedy
  - On définit une valeur ε dans [0, 1]
    - Ce seuil définit la probabilité de choisir une action au hasard (exploration) ou la meilleure action (exploitation)
    - Algorithme :
      - Tirer un nombre *rand* entre 0 et 1
      - Si $rand<\varepsilon$
        - Choisir une action aléatoirement
      - Sinon
        - Choisir l'action qui mène à la valeur la plus élevée
  - Ɛ peut évoluer au cours du temps : plus d'exploration au début ($Ɛ^{t0} = 1$), de plus en plus d'exploitation par la suite

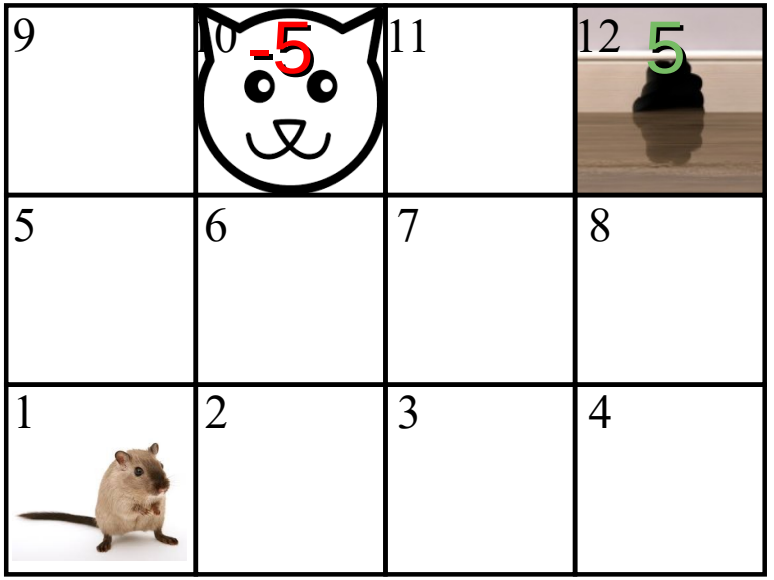
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - Soit l'environnement suivant
    - Chaque état est identifié par un numéro
    - On peut passer d'un état à un état adjacent (A = {haut, bas, droite, gauche})
    - Certains états offrent une récompense positive ou négative



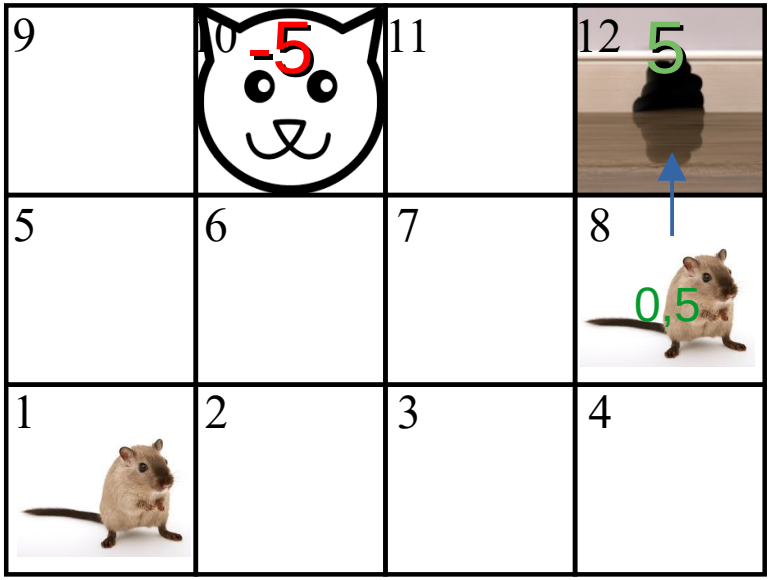
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On définit le gain G comme la somme des récompenses sur une séquence d'états
    - Soit la séquence {1, 2, 3, 4, 8, 12}
      → Le gain vaut G = 0 + 0 + 0 + 0 + 5 = 5
    - On va chercher à optimiser le gain.



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

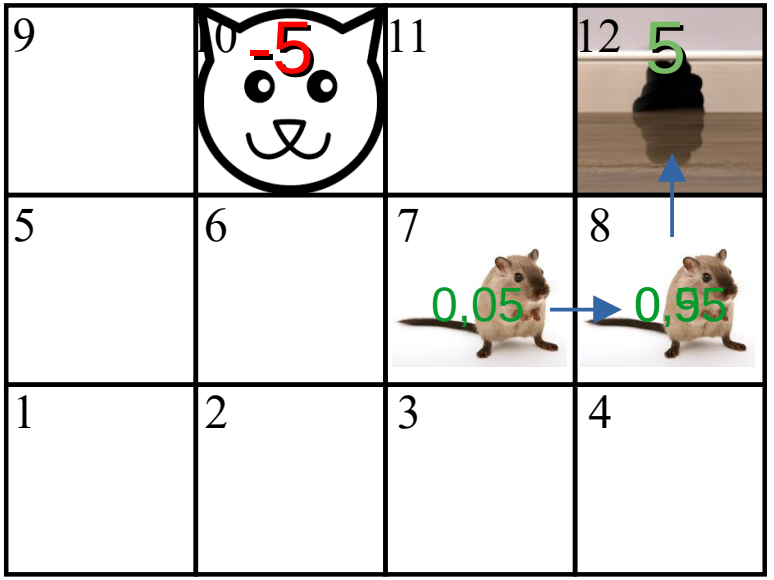
- **Apprentissage par renforcement :**
  - On effectue un apprentissage (décision aléatoire à chaque étape)
  - À un moment, l'agent se trouve dans l'état 8 et se dirige vers l'état 12
    - L'agent reçoit une récompense de 5
    - La value function V_8 est mise à jour : $V_8 \leftarrow 0 + \alpha \, . \, (5 + 0 - 0) = 0,5$



$\alpha = 0,1$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - Plus tard, l'agent se trouve dans l'état 7 et se dirige vers l'état 8
    - La value function V_7 est mise à jour : $V_7 \leftarrow 0 + \alpha \, . \, (0 + 0{,}5 - 0) = 0{,}05$
  - Puis il se dirige vers l'état 12
    - La value function V_8 est mise à jour : $V_8 \leftarrow 0{,}5 + \alpha \, . \, (5 + 0 - 0{,}5) = 0{,}95$



$\alpha = 0{,}1$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - Progressivement, les valeurs de récompense vont se 'diffuser' dans les états au travers des value functions
  - Chaque value function donne l'espérance de gain en considérant des décisions aléatoires par la suite depuis un état S.
  - À noter : on peut enregistrer la séquence pour propager les valeurs en une fois !

Résultats après 5000 actions

$V^{rand}(s) = E(G \mid s, \pi_{rand})$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On réduit ensuite progressivement ε pour évoluer vers une exploitation
  - Les values functions vont alors évoluer pour refléter l'espérance de gain en choisissant toujours la meilleure action possible

9 -4,71	10  	11 0,7	12 
5 -3,36	6 -2,15	7 -0,04	8 2,97
1 -1,97	2 -1,45	3 -0,05	4 1,07

Résultats après 5000 actions

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On réduit ensuite progressivement ε pour évoluer vers une exploitation
  - Les values functions vont alors évoluer pour refléter l'espérance de gain en choisissant toujours la meilleure action possible

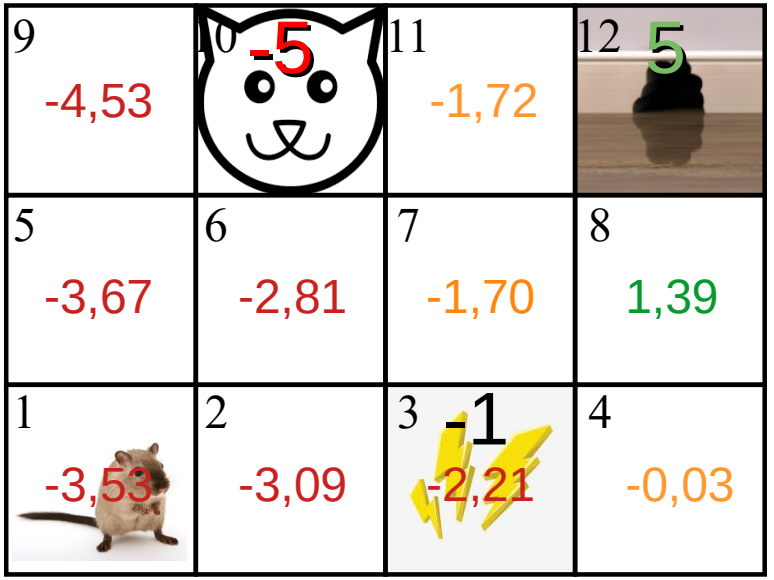
9 -4,04	10 -5 	11 2,13	12 5 
5 2,13	6 3,29	7 4,20	8 5,0
1 4,50	2 5,0	3 5,0	4 5,0

Résultats après 5000 actions supplémentaires

$V^*(s) = E(G \mid s, \pi^*)$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

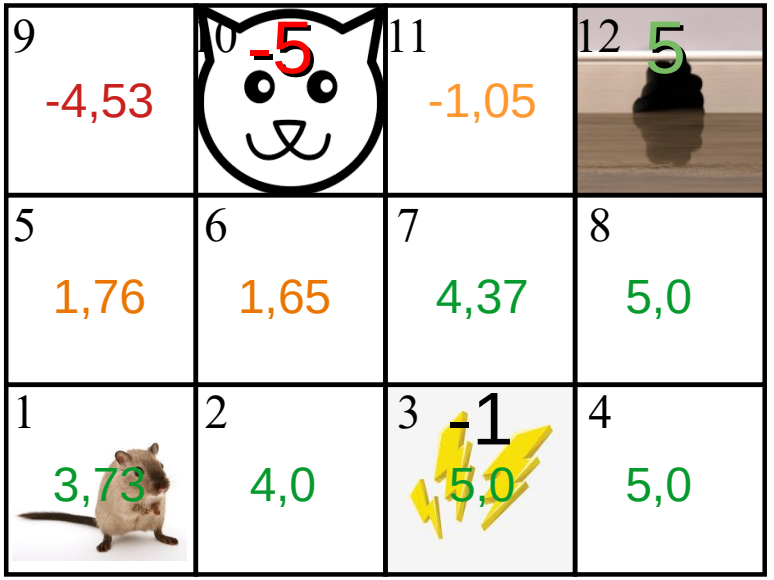
- **Apprentissage par renforcement :**
  - On ajoute une récompense négative sur l'état 3
    - Ce n'est pas un état terminal : l'agent peut continuer à agir
  - On recommence l'exploration



Résultats après 5000 actions

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On réduit ensuite progressivement ε pour évoluer vers une exploitation
  - Les values functions vont alors évoluer pour refléter l'espérance de gain en choisissant toujours la meilleure action possible
  - Un problème apparaît !



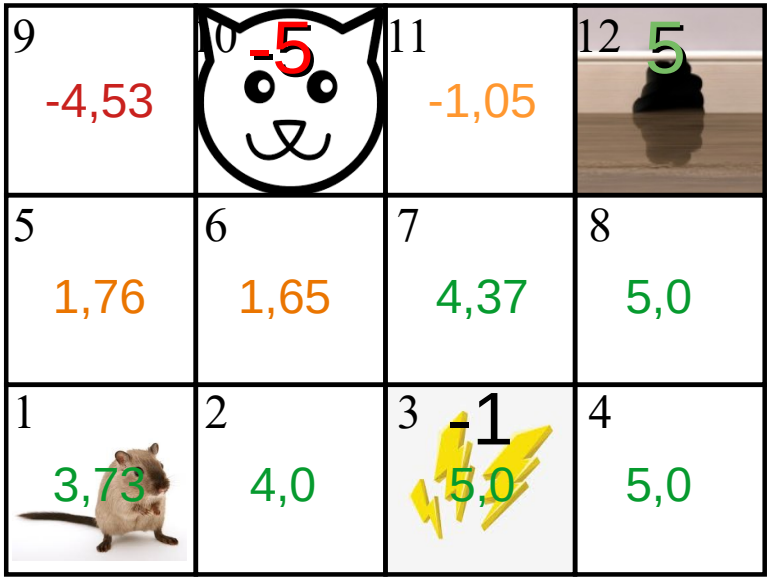
Résultats après 5000 actions supplémentaires

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - Un problème apparaît !

→ Le 'meilleur' chemin n'est pas optimal : on passe par l'état 3

- Le chemin donne toujours un gain positif (+4), mais la présence du chat a suffisamment réduit les value functions autours pour que ce chemin ne soit pas utilisé en exploitation
- Il faudrait pouvoir augmenter en 2 l'importance du reward de l'état 3



Résultats après 5000 actions supplémentaires

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On va modifier le calcul du gain pour diminuer l'importance des récompenses futures par rapport aux plus proches.
    - On introduit le *facteur d'actualisation* γ (dans [0,1[ )
    - On définit le gain comme : $G = \gamma^0 . r_0 + \gamma^1 . r_1 + \gamma^2 . r_2 + \gamma^3 . r_3 + \ldots + \gamma^n . r_n + \ldots$
      - Si $\gamma = 0,9$, $0,9^2 = 0,89$, $0,9^3 = 0,729$, $0,9^4 = 6561$, $0,9^5 = 0,5905$ …
      - À noter : comme γ^n diminue très vite, on peut considérer une limite au-delà de laquelle $\gamma^n \approx 0$ (horizon)
    - Généralement, on utilise un facteur γ entre 0,9 et 0,99

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**

  - On peut redéfinir la mise à jour de la value function :
    - En suivant le même principe récursif que précédemment

      $V(s) = E (G \mid s)$

      $= E (r + \gamma^1 . r_1 + \gamma^2 . r_2 + \gamma^3 . r_3 + \dots \mid s)$

      $= E (r + \gamma . (r_1 + \gamma^1 . r_2 + \gamma^2 . r_3 + \dots) \mid s)$

      $= E (r + \gamma . E(G \mid s_1) \mid s)$

      $= E (r + \gamma . V(S_1) \mid s)$

Richard Bellman
1920-1984

    - Et pour la valeur max

$$V^*(s) = \max_{s'|s} (r_{s,s'} + \gamma . V^*(s'))$$

$$V^*(s) = \sum_a P(s'|s,a) . \max_{s'|s} (r_{s,s'} + \gamma . V^*(s'))$$

(version non-déterministe)

    - Equation de Bellman

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

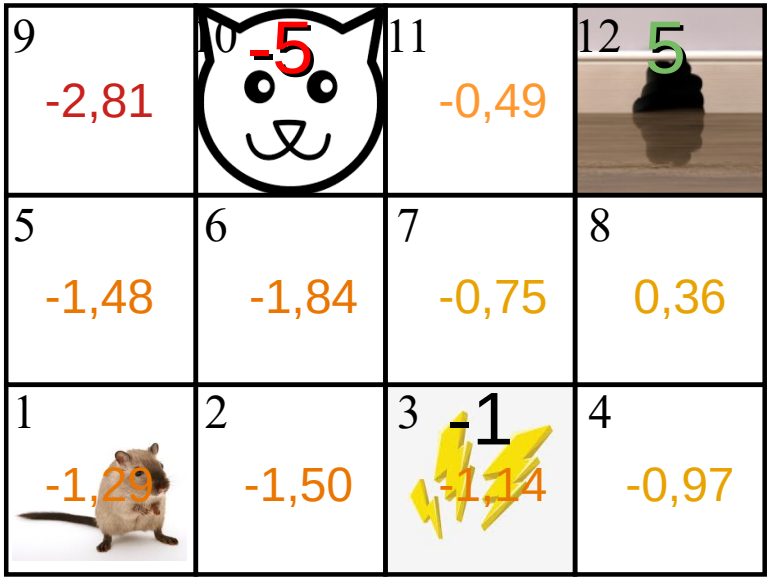
- **Apprentissage par renforcement :**
  - Mise à jour des valeurs à chaque transition s_i vers s_{i+1} :

$$V(s_i) \leftarrow (1 - \alpha)\, V(s_i) + \alpha \,.\, [\, r_{i,i+1} + \gamma \,.\, V(s_{i+1}) \,]$$

$$V(s_i) \leftarrow V(s_i) + \alpha \,.\, [\, r_{i,i+1} + \gamma \,.\, V(s_{i+1}) - V(s_i) \,]$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

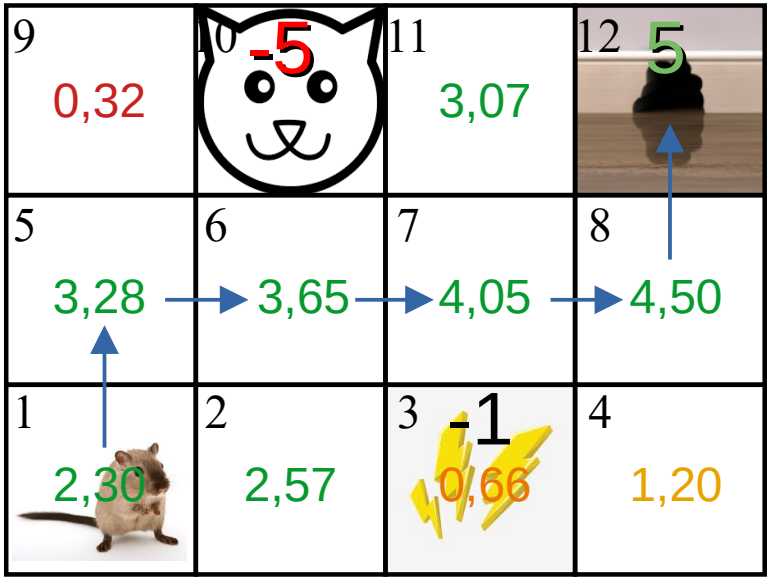
- **Apprentissage par renforcement :**
  - On réessaie avec la nouvelle fonction de mise à jour et un facteur d'actualisation $\gamma = 0{,}9$ sur 5000 actions
    - $V(S_i) \leftarrow V(S_i) + \alpha . [r_{i, i+1} + \gamma . V(S_{i+1}) - V(S_i)]$



Résultats après 5000 actions supplémentaires

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**
  - On réessaie avec la nouvelle fonction de mise à jour et un facteur d'actualisation $\gamma = 0,9$ sur 5000 actions
    - $V(S_i) \leftarrow V(S_i) + \alpha . [r_{i, i+1} + \gamma . V(S_{i+1}) - V(S_i)]$
  - Puis on on évolue progressivement vers une exploitation sur 5000 autres actions



Résultats après 5000 actions supplémentaires

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**

  - Comment appliquer ce principe si on ne sait pas comment passer d'un état vers un autre ?

    → dans un processus de décision markovien, une même action peut conduire à des états différents (avec une certaine probabilité)

    → Un modèle qui n'a pas besoin de connaître les transitions est appelé *'model-free'*

  - En 1981, Stevo Bozinovski propose une approche avec une matrice liant les actions et les états avec des poids : $W = [w(a_i,s_j)]$

  - En 1989, Chris Watkins propose une nouvelle approche, appelée Q-learning (Q pour 'Quality')

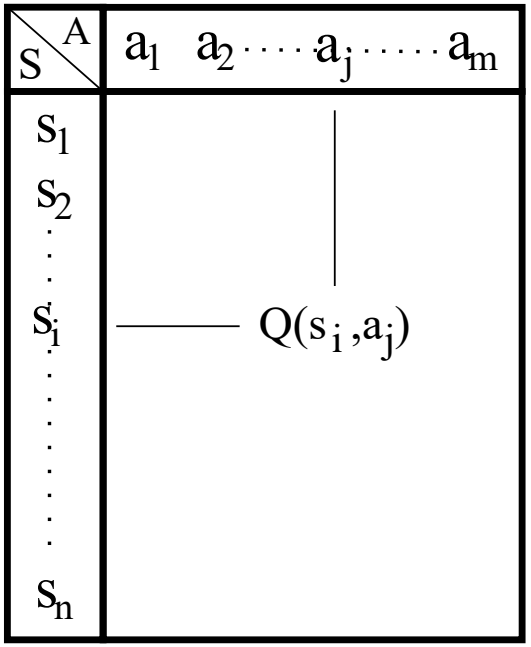
    - Principe (Value function VS Q-learning) :
      - La Value Function donne l'espérance d'après l'état courant :

        $V : S \rightarrow R$
      - Le Q-learning donne l'espérance d'après l'état et les actions possibles :

        $Q : S \times A \rightarrow R$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning
  - En pratique :
    - On définit une table, appelée Q-table



    - On définit l'espérance de gain pour chaque état courant et action possible

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning

  - Calcul des Q-values

    - On définit le gain : $G = \gamma^0 . r_0 + \gamma^1 . r_1 + \gamma^2 . r_2 + \gamma^3 . r_3 + \gamma^4 . r_4 \ldots$

    - On définit ensuite l'espérance du gain, en supposant le choix de la meilleure valeur à chaque action.
      - Rappel : équation de Bellman $V^*(s) = \max_{s'|s}(r_{s,s'} + \gamma . V^*(s'))$

    - Cette fois, la récompense va dépendre de la décision a : $r_{s,s'} = r(s,a)$

$$Q^*(s,a) = r_{s,s'} + \gamma . \max_{a' \in A}(Q^*(s',a'))$$

Maximum sur une ligne de la Q-table

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning

  - Mise à jour des Q-values à chaque transition (s,a,s',r) :

$$Q(s,a) \Leftarrow (1-\alpha).Q(s,a) + \alpha.(r_{s,s'} + \gamma.\max_{a' \in A}(Q(s',a')))$$

$$Q(s,a) \Leftarrow Q(s,a) + \alpha.(r_{s,s'} + \gamma.\max_{a' \in A}(Q(s',a')) - Q(s,a))$$

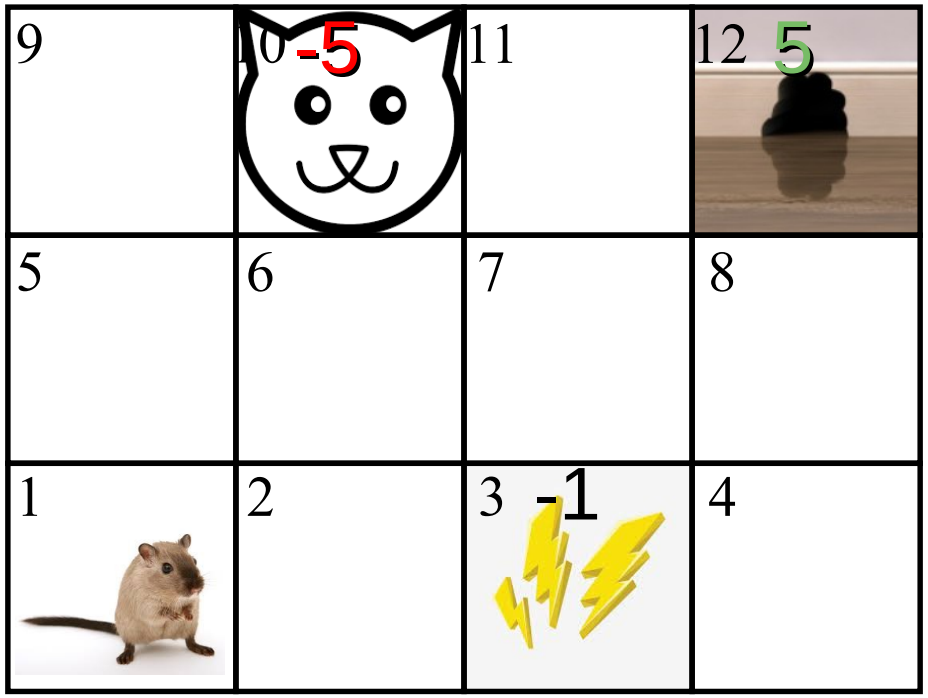
Maximum sur la ligne courante de la Q-table

  - À noter : dans un environnement non déterministe, il faut prendre en compte la probabilité P(s' | s,a) :

$$Q^*(s,a) = \sum_{s'} P(s'|s,a) \cdot (r_{s,s'} + \gamma.\max_{a' \in A}(Q^*(s',a')))$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning
  - Reprenons notre exemple :
    (ε-greedy, $\alpha = 0{,}1$, $\gamma = 0{,}9$)

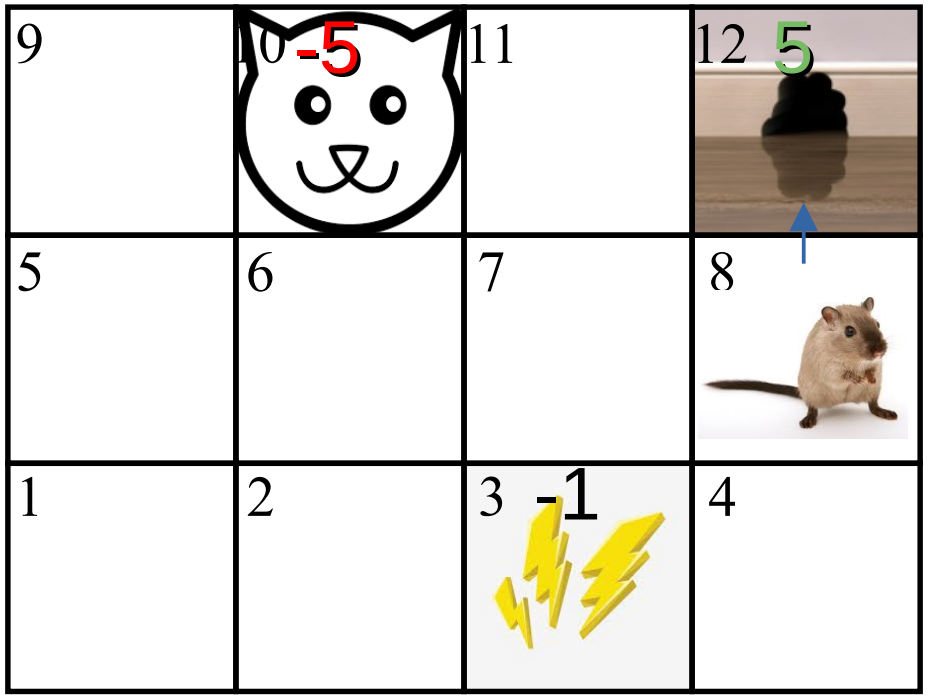


Q-table

	→	↑	←	↓
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning
  - À un moment, l'agent est dans l'état 8 et effectue l'action 2 (haut) → r = 5
  - On met à jour Q(8,2) :
    Q(8,2)← 0 + 0,1x(5 + 0,9x0 – 0) = 0,5

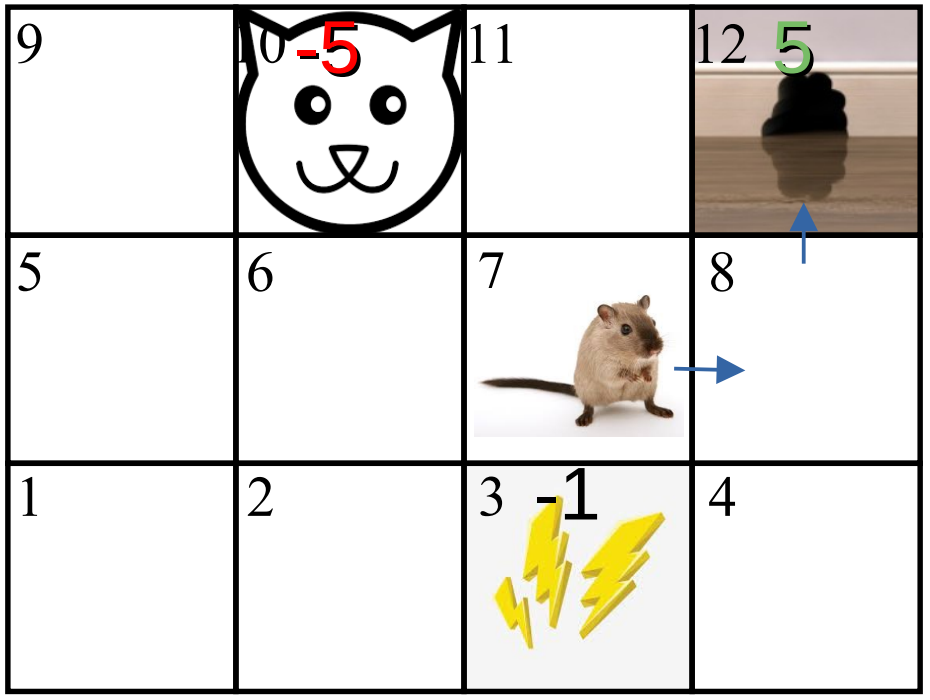


	→	↑	←	↓
1				
2				
3				
4				
5				
6				
7				
8		0,5		
9				
10				
11				
12				

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning
  - Plus tard : (7 → 8), puis (8 ↑ 12)

    Q(7,1) = 0 + 0,1x(0 + 0,9x0,5 -0) = 0,045



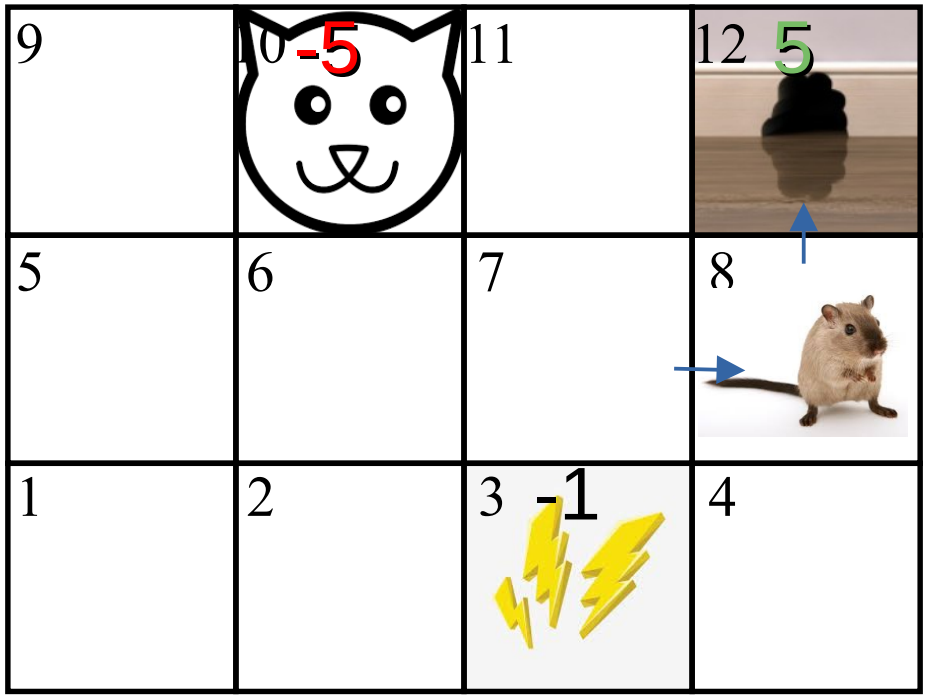
	→	↑	←	↓
1				
2				
3				
4				
5				
6				
7	0,045			
8		0,5		
9				
10				
11				
12				

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** le Q-learning
  - Plus tard : (7 → 8), puis (8 ↑ 12)

    Q(7,1) = 0 + 0,1x(0 + 0,9x0,5 -0) = 0,045

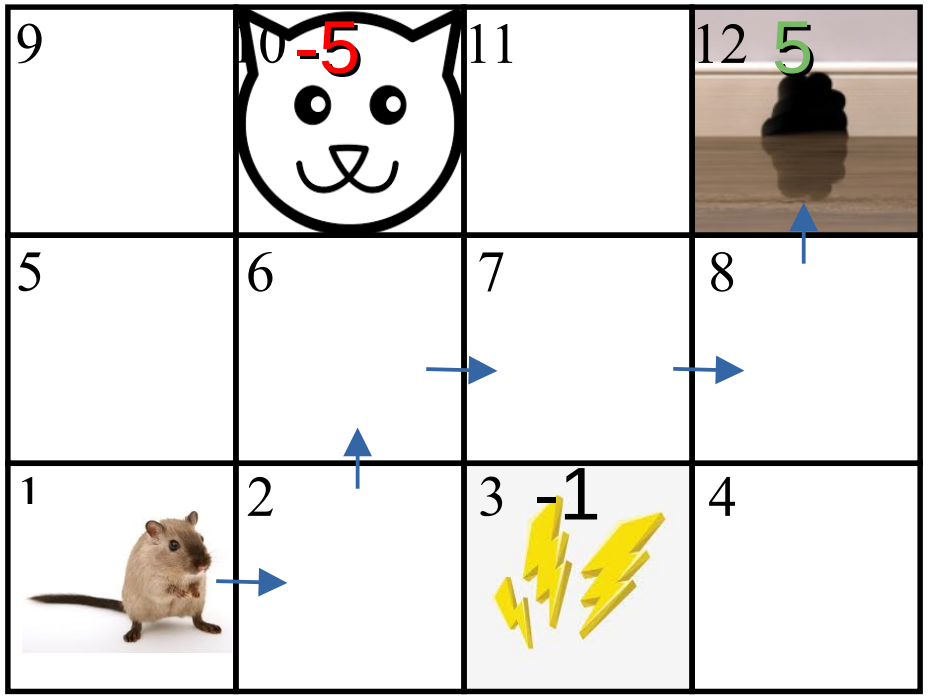
    Q(8,2) = 0,5+0,1x(5+0,9x0-0,5) = 0,95



	→	↑	←	↓
1				
2				
3				
4				
5				
6				
7	0,045			
8		0,95		
9				
10				
11				
12				

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

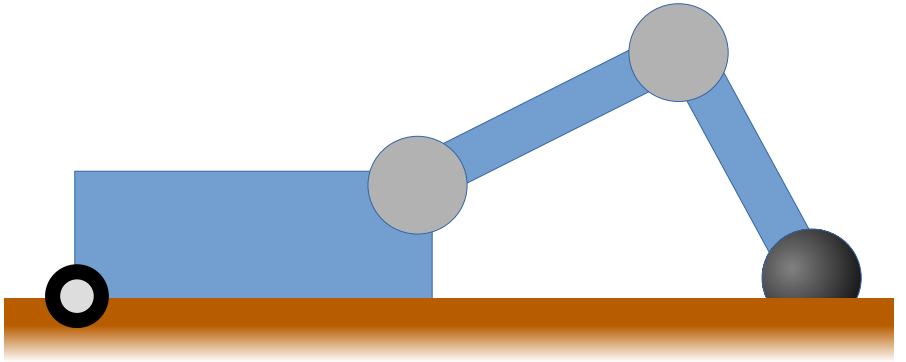
- **Apprentissage par renforcement :** le Q-learning
  - Apprentissage sur 10 000 actions



	→	↑	←	↓
1	3,28	3,16	2,84	2,83
2	2,62	3,65	2,91	3,27
3	4,05	3,97	3,13	2,54
4	4,0	4,50	2,55	3,98
5	3,64	2,69	3,16	2,80
6	4,05	-5	3,24	3,25
7	4,5	4,35	3,57	2,58
8	4,49	5	4,02	4,0
9	-4,99	2,50	2,55	3,10
10	0	0	0	0
11	4,94	3,81	-4,83	3,5
12	0	0	0	0

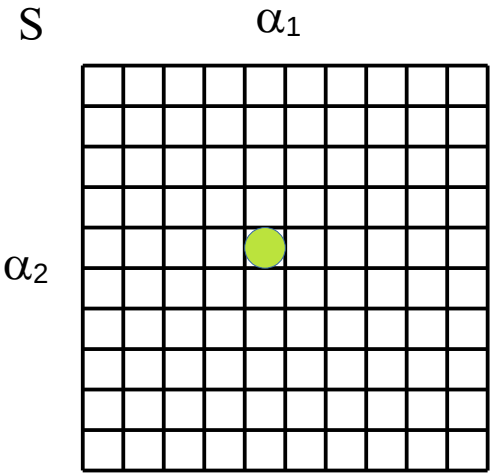
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** exemples d'application
  - Faire marcher un robot :
    - Deux angles α_1 et α_2
    - Ensemble d'états S discrétisant l'espace { (α_1 , α_2) }
    - 4 actions possibles : incrémenter ou décrémenter α_1 et α_2 (équivalent à un déplacement sur S)
    - Récompense définie par la mesure du déplacement (positif ou négatif)



$$A = \{ \alpha_1\text{++}, \alpha_1\text{--} , \alpha_2\text{++} , \alpha_2\text{--} \}$$

$$r_{s,\,a} = p^{t+1} - p^{t}$$



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** exemples d'application
  - Faire marcher un robot :
    - Le robot s'adapte au type de sol !

The Crawler, A Class Room Demonstrator for Reinforcement Learning, M. Tokic, W. Ertel and J Fessler, 2009.

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** exemples d'application
  - Quelques autres applications pratiques :
    - Contrôle de robots ou de drones
    - Gestion des ressources
    - Chimie (prédiction des réactions)
    - Planification des tâches
    - trading

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** en résumé

  - **Reinforcement learning** : pour les systèmes dont on connaît les transitions entre états
    - Un ensemble d'états S, un ensemble d'actions A
    - Une politique d'action $\pi(S) \rightarrow a$
    - Une value function pour chaque état

      $V(s_i) \leftarrow V(s_i) + \alpha . [r_{i, i+1} + \gamma . V(s_{i+1}) - V(s_i)]$

  - **Q-learning** : plus 'universel', mais beaucoup plus de valeurs à définir
    - Un ensemble d'états S, un ensemble d'actions A
    - Une politique d'action $\pi(S) \rightarrow a$
    - Une Q-table (card(A) x card(S) valeurs)
    - Une valeur de qualité pour chaque couple (s^t, a^{t+1})

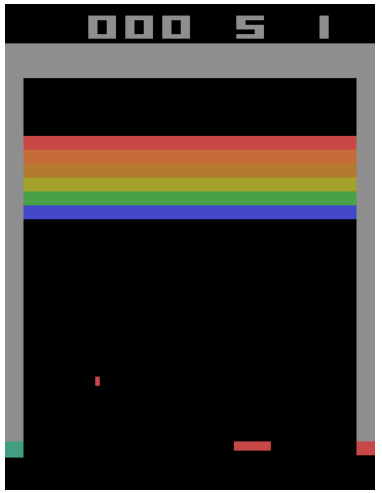
      $$Q(s,a) \Leftarrow Q(s,a) + \alpha.(r_{s,s'} + \gamma . \max_{a' \in A}(Q(s',a')) - Q(s,a))$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :** en résumé

  - Une faiblesse : le nombre d'états à gérer
    - Plus les états sont nombreux, plus il faut d'essais
    - De nombreuses méthodes existent pour rassembler et fusionner des états similaires, mais ce n'est pas toujours suffisant.

    - Exemple : une IA par renforcement peut-elle jouer à un jeu vidéo ?

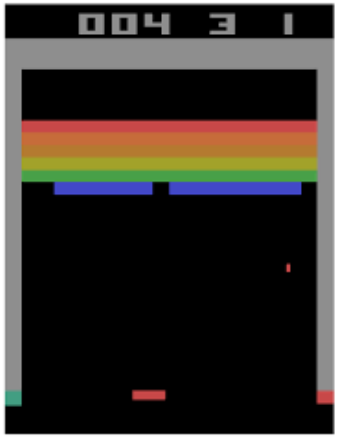


Breakout (Atari 2600)

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**

  - Breakout :
    - États : configuration des pixels (2^{108}), position de la balle, position de la raquette → card(S) > 10^{32}
    - Est c'est un jeu simple !

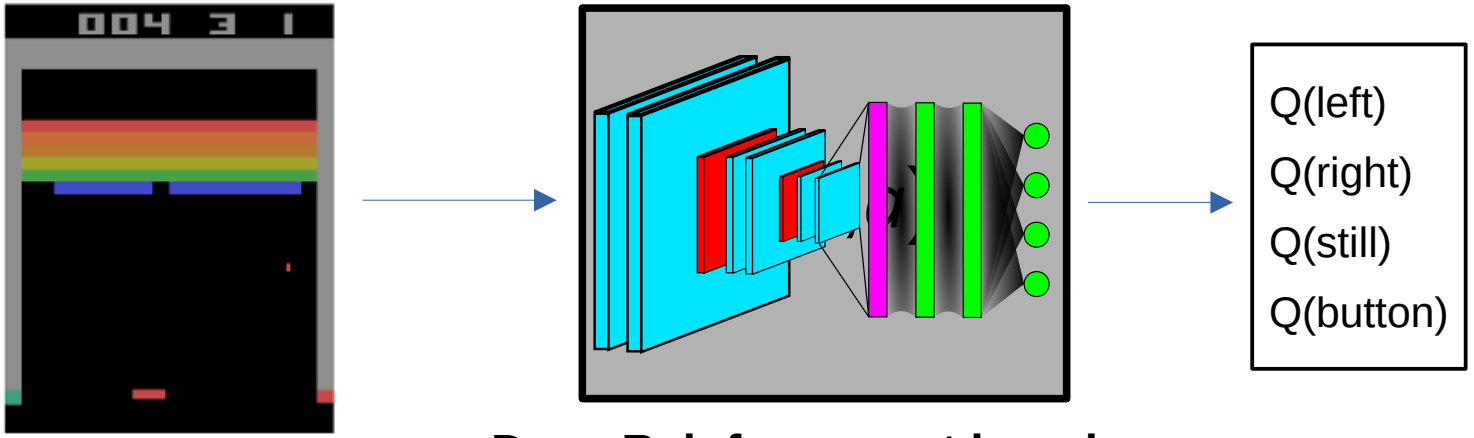


    - Même avec des simplification, le nombre d'état reste prohibitif !

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement :**

  - Breakout :
    - On a donc : une image représentant l'environnement
    - On cherche à obtenir une estimation de la quality value de chaque action (vecteur numérique) : { Q(left), Q(right), Q(still), Q(button) } : une fonction *f(a,s) → [0,1]*

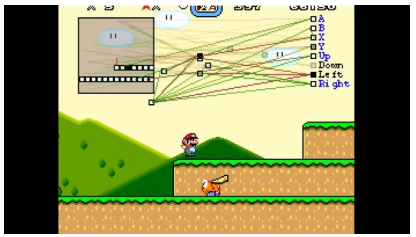


Deep Reinforcement learning

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**

  - On est moins limité par la complexité de l'environnement !
  - En 2013, DeepMind propose une architecture capable de jouer à Breakout
  - Depuis, d'autres applications (dont des jeux vidéos) ont pu bénéficier du Deep RL

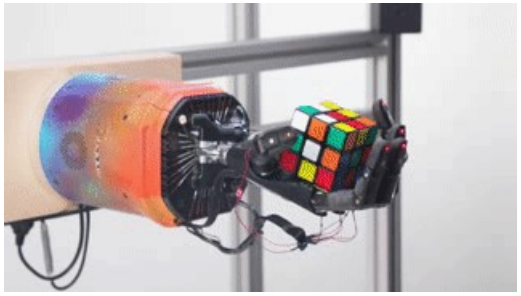




  - En 2015, AlphaGo (Google) devient la première IA à battre le champion mondial au jeu de Go

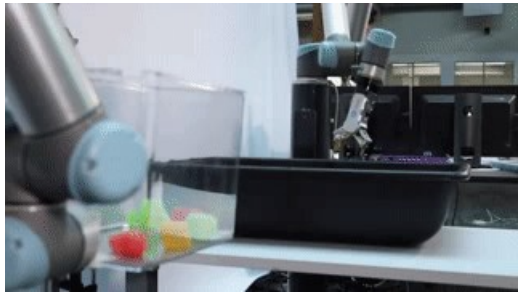
Tutoriel Keras pour jouer à Breakout :
https://keras.io/examples/rl/deep_q_network_breakout/

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Applications en robotique :



Solving Rubik's Cube with a Robot Hand, OpenAI, I. Akkaya et al., 2018



TossingBot: Learning to Throw Arbitrary Objects with Residual Physics, A. Zeng et al., 2019

Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots, Z. Li et al., 2021

  - Conduite automatique (futur des véhicules autonomes?)
  - Résoudre des problèmes avec une très grande complexité algorithmique

Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning, F. Fuchs et al., 2021

AlphaGo (Google), 2015

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**

  - Spécificités du Deep RL :
    - Très peu d'exemples par rapport au Deep Learning : un test par cycle de décision
    - Interaction 'séquentielle' avec l'environnement : les actions influent sur les états futurs
      - Le réseau peut découvrir des régularités ou des biais statistiques liées à une partie de l'environnement (ce qui peut réduire les performances dans les autres cas).
    - À contrario, les spécificités d'une partie de l'environnement peuvent faire émerger des propriétés qui supplantent l'apprentissage dans les parties visitées précédemment.

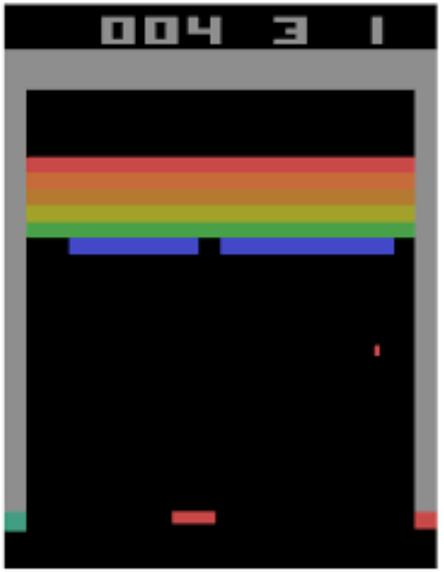
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**

  - Problèmes liées au jeu de données :
    - On va enregistrer des 'épisodes' pendant l'interaction agent-environnement
    - Épisodes sous la forme de tuples {s, a, r, s'}
      - État s, action a, récompense reçue r, nouvel état s'.
      - Ces épisodes vont constituer le dataset
    - C'est le principe de l'*Experience Replay*
    - À chaque apprentissage, on créé un batch aléatoires contenant des épisodes récents et anciens (mélangés)
      - Le dataset augmente en taille avec le temps
      - On évite les biais statistiques et les 'connaissances' passées sont maintenues
    - À noter : souvent, c'est ce module qui sélectionne l'action et interagit avec l'environnement

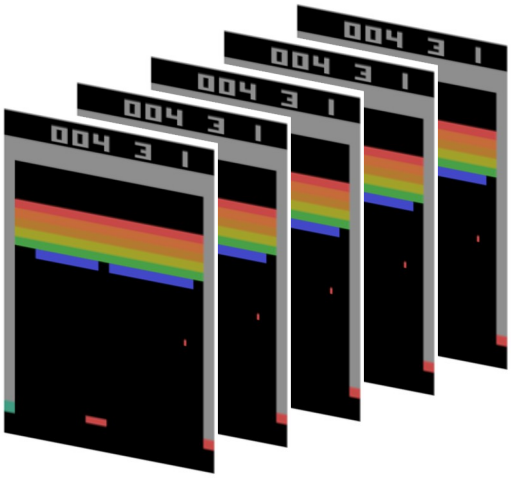
Introduction à l’Intelligence Artificielle : L’apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Spécificités du Deep RL :
    - Environnement dynamique : une seule image n’est parfois pas suffisante pour prendre une décision (exemple : direction du mouvement de la balle)



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Spécificités du Deep RL : problème de l'environnement dynamique
    - Pour palier à ce problème, on empile plusieurs frames consécutives
      - On peut empiler les images (en augmentant la profondeur)
        → convolution 2D
      - Ou considérer le temps comme une dimension supplémentaire
        → convolution 3D



Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Apprentissage du réseau :
    - Rappels : mise à jour des Q-values et fonction de coût d'un réseau

$$Q(s,a) \Leftarrow Q(s,a) + \alpha . (r_{s,s'} + \gamma . \max_{a' \in A}(Q(s',a')) - Q(s,a))$$

Résultat attendu — Résultat de la Q-table

$$E(y) = (r - y)^2$$

Résultat attendu — Résultat du réseau

$$L(\theta) = [(r_{s,s'} + \gamma . \max_{a' \in A}(Q(s',a',\theta))) - Q(s,a,\theta)]^2$$

- À noter : le réseau sert à la fois à obtenir la valeur de sortie Q(s,a,θ) à calculer la valeur cible avec max(Q(s',a',θ))
- Mise à jour des poids avec une descente de gradient :

$$w \Leftarrow w + \alpha . [(r_{s,s'} + \gamma . \max_{a' \in A}(Q(s',a',\theta))) - Q(s,a,\theta)] . \frac{\partial Q(s,a,\theta)}{\partial w}$$

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**

  - Quelques améliorations possibles :
    - Utiliser un second réseau pour la Q-value prédite (max(Q(s',a'))
      - Ce réseau secondaire n'est pas entraîné : on recopie les poids du premier réseau à intervalle régulier

        → stabilité des valeurs cibles pour entraîner le premier réseau sur la durée d'un intervalle.

    - Double Deep Q Network (Double DQN) :
      - On utilise un second réseau pour prédire la meilleure action dans un état s :

$$Q(s,a,\theta) = r + \gamma . Q(s', \underset{a'}{argmax}\, Q(s',a',\theta'), \theta)$$

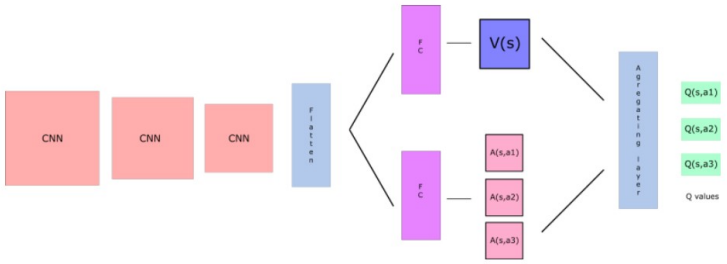
      - On évite de sur-évaluer la Q-value cible

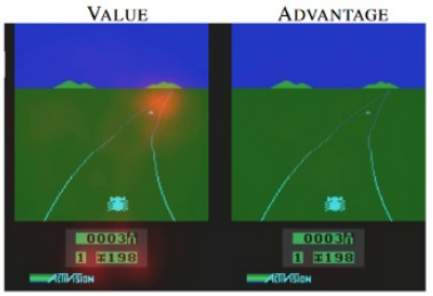
Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

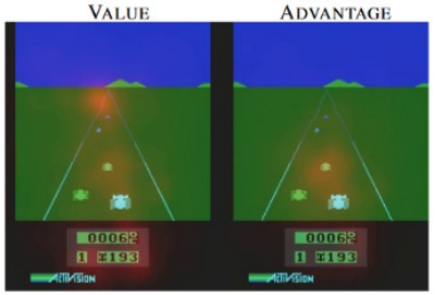
- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Quelques améliorations possibles :

    - Dueling DQN :
      - On sépare la Q-value en deux : la V-value et l'avantage de l'action dans un état s :

$$Q(s,a) = V(s) + A(s,a)$$



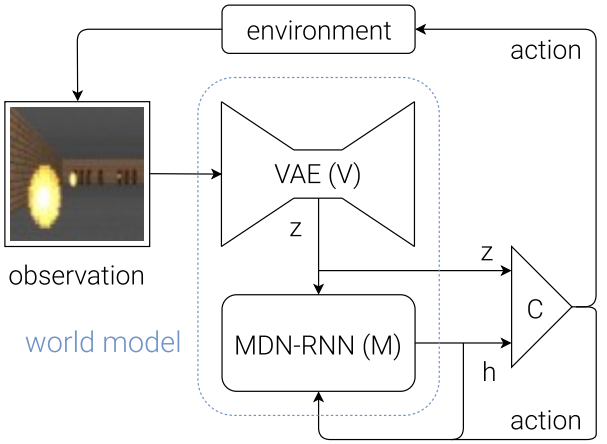




      - Simplifie considérablement le traitement des situations où l'action n'a pas d'incidence sur les états suivants

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Deep Reinforcement Learning**
  - Prédiction des états futurs MDN-RNN (Mixture Density Network) :
    - Un réseau encodeur-décodeur génère un espace latent pour représenter l'environnement
    - Une mémoire apprend à prédire le vecteur latent futur à partir du vecteur présent et de l'action de l'agent



  - On peut ensuite ré-injecter ces prédictions comme entrées !

https://worldmodels.github.io/

Introduction à l'Intelligence Artificielle : L'apprentissage par renforcement

- **Apprentissage par renforcement : Conclusion**

  - Le reinforcement learning permet de résoudre des problèmes 'ancrés dans le réel'

  - Contrairement à une IA de type deep learning, une IA de type RL peut être autonome
    - Mais les deux modèles ne sont pas incompatibles :

      Les modèles de deep reinforcement learning permettent de compenser les limites du reinforcement learning (complexité de l'environnement).

    - Le deep RL est un domaine très récent, mais déjà très prometteur pour des applications robotiques, mais pas seulement...