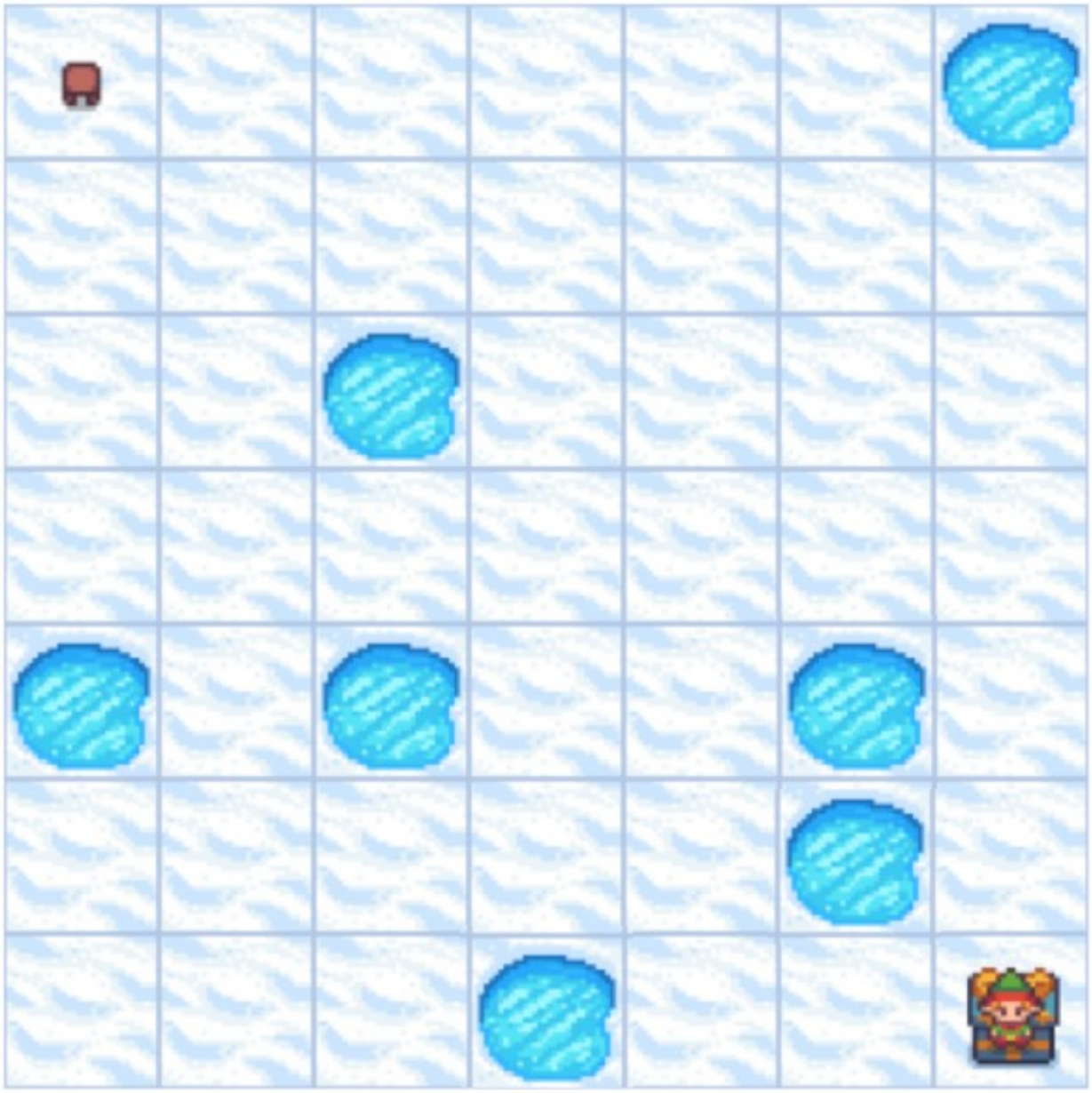
Implémentation de Policy Iteration, Value Iteration et Epsilon-Greedy dans un environnement Frozen Lake

Introduction :

L'objectif de ce TP est d'implémenter et de comparer trois algorithmes fondamentaux en apprentissage par renforcement : Policy Iteration, Value Iteration et l'algorithme Epsilon-Greedy. Ces algorithmes seront appliqués à un environnement Frozen Lake, représenté graphiquement.

Environnement Frozen Lake : Le Frozen Lake est un environnement de grille où un agent (en bas à droite) doit naviguer à travers une surface glacée pour atteindre un objectif (en haut à gauche) tout en évitant des cases dangereuses. La grille est représentée ici graphiquement, et chaque case peut être soit une case gelée (sûr, reward = 0), une case dangereuse (trou, reward = -10), ou l'emplacement de l'objectif (reward = 10). L'agent peut se déplacer vers la gauche, droite, en haut ou en bas. Utiliser le dessin graphique pour définir les valeurs initiales des états et les récompenses associées.

Toute information non précisée sera laissée libre au choix du développeur.se.et précisée dans votre notebook



Rappel de Q-learing

L'algorithme Q-Learning nous permet d'estimer la Q fonction optimale en utilisant uniquement les trajectoires du MDP obtenues en suivant une certaine politique d'exploration.

Q-learning avec une exploration de type ε-greedy effectue la mise à jour suivante au temps t :

1. Dans l'état s_t, on effectue une action a_t telle que a_t est aléatoire avec une probabilité ε et $a_t \in \arg\max_a \hat{Q}_t(s_t,a) $ avec une probabilité $1-\varepsilon$;
2. Observer s_{t+1} et la récompense r_t ;
3. Calculer $\delta_t = r_t + \gamma \, max_a\left(\widehat{Q}_t\left(s_{t+1}, a\right) - \widehat{Q}_t\left(s_t, a_t\right)\right)$;
4. Mettre à jour $\hat{Q}_{t+1}(s, a) = \hat{Q}_t(s, a) + \alpha_t(s,a)\delta_t\mathbb{1}{s=s_t, a=a_t} $.

Tâches à accomplir :

- Policy Iteration (1/4): Implémenter l'algorithme de Policy Iteration pour résoudre le problème du Frozen Lake. Afficher les Q-table et Q-policy obtenues.
- Value Iteration (1/4): Implémenter l'algorithme de Value Iteration pour résoudre le problème du Frozen Lake. Afficher les valeurs d'état obtenue. Fixer un seuil d'arrêt δ en justifiant.
- Q-learning (stratégie de choix : Epsilon-Greedy) (1/4): Implémenter l'algorithme Epsilon-Greedy pour la prise de décision de l'agent dans l'environnement Frozen Lake. Utiliser différentes valeurs de ϵ et afficher les Q-tables obtenues.
- Comparaison des algorithmes (1/4): Comparer les performances des trois algorithmes en termes de convergence, nombre d'itérations nécessaires, et qualité de la politique finale. Vous pouvez représenter la politique de l'agent à travers la grille de jeu. Analyser l'impact du paramètre epsilon dans l'algorithme Epsilon-Greedy sur l'exploration de l'agent.

L'utilisation de la librairie gym est interdite. Seule les librairies numpy et random sont autorisées pour l'impémentation de l'algorithme. Toutes les librairies graphiques (matplot etc...) sont autorisées pour l'interprétation des résultats.