

# INFO-F311 - Project - RL

Luca Palmisano - 000516721

December 2023

# 1 Value Iteration

## 1.1 Limitation

Lorsque le nombre d'agents augmente ou que la taille de la carte croît, cela entraîne une augmentation du nombre d'états. À mesure que le nombre d'états augmente, les exigences en termes de temps de calcul pour l'algorithme de Value Iteration augmentent de manière significative[1]. Cette croissance peut entraîner des problèmes, avec le risque que l'algorithme converge beaucoup plus lentement, voire pas du tout.

## 1.2 Planificateur hors-ligne

Value Iteration n'est pas un algorithme RL pour plusieurs raisons. Tout d'abord, il explore de manière exhaustive l'ensemble des états possibles et des actions réalisables, ce qui peut devenir un inconvénient notable lorsque le nombre d'agents ou la taille de l'environnement augmente (voir 1.1). De plus, contrairement aux algorithmes de RL qui apprennent par essais et erreurs en interagissant directement avec l'environnement, Value Iteration effectue une planification complète même d'interagir avec l'environnement.

## 1.3 Application

Lorsque la *policy* se stabilise, l'algorithme fournit une solution où il commence par récupérer la gemme située sur son trajet, avant de se diriger vers la sortie.

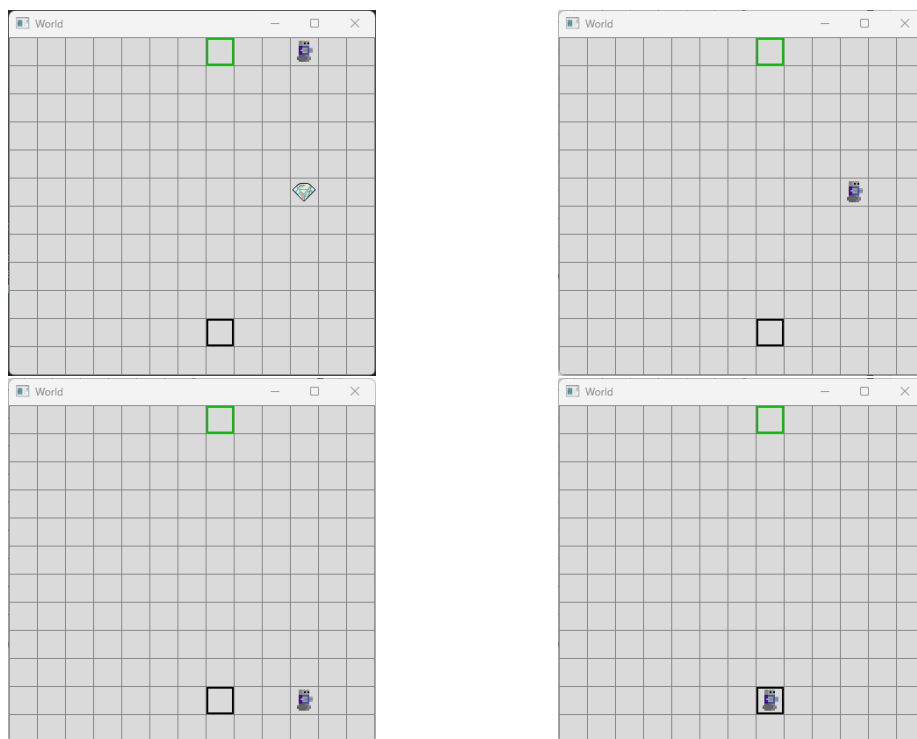


FIGURE 1 – Stratégie

## 2 Expériences avec le Q-learning

*Je n'ai pas effectué cette partie du projet*

## Références

- [1] Eugene A. Feinberg and Jefferson Huang. The value iteration algorithm is not strongly polynomial for discounted dynamic programming, 2013.