Rapport COCOMA Projet MADRL

Angèle Ramaugé, 21309173
Max Maiche, 21312637

Sorbonne Université

2024-2025

1. Environnement

L'environnement que nous avons utilisé est Knights Archers Zombies ou KAZ (lien vers PettingZoo). Il consiste d'une carte fixe où agissent des agents contrôlés (Knights et Archers) et des agents automatiques (Zombies). Les Zombies se déplacent automatiquement de façon aléatoire vers le bas de la carte, nos agents ont pour but de tuer les zombies avant qu'ils arrivent en bas. Si un agent et un zombie se rencontrent, l'agent meurt. Les archers peuvent tirer des flèches devant eux, les chevaliers peuvent balayer une masse devant eux en arc de cercle. La récompense des agents est de 1 lorsqu'ils tuent un zombie (récompense personnelle).

2. Wrappers

Nous avons utilisé différents wrappers pour nous aider lors de l'utilisation de KAZ.

2.1 Environnements Parallèles

Ce Wrapper permet de transformer un environnement en cycle a un environnement parallèle. A chaque tour, les agents choisissent donc leur actions simultanément.

2.2 Black Death

Dans KAZ les agents peuvent mourir pendant un épisode, ils n'ont donc pas d'observables et de récompenses une fois mort. Black Death ajoute des valeurs par défaut (correspondant à un observable vide et une récompense nulle) à tous les agents morts.

3. Expériences

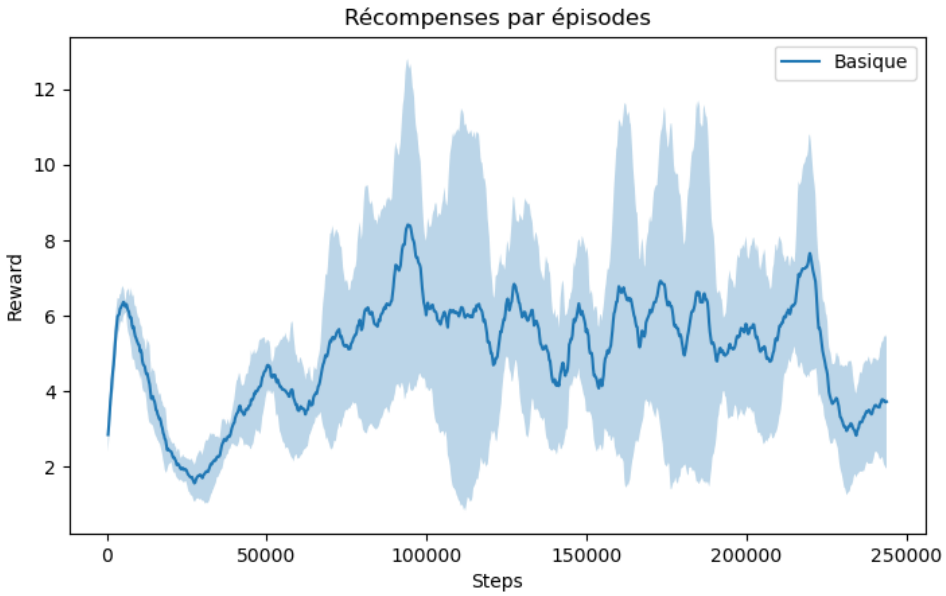
Pour toutes les expériences la métrique est la somme cumulée des récompenses intrinsèques des agents donc avant la modification du signal de récompenses.

3.1 Un DQN unique pour l'environnement

Nous avons dans un premier temps effectué un seul DQN sur l'environnement. Cette première approche nous a permis de prendre en main environnement et de résoudre les problèmes de forme de l'apprentissage en ajoutant les wrappers nécessaires (parallel et blackdeath). Le DQN utilisé est celui de la librairies Stable Baselines3.
Cet apprentissage ne prend pas en compte les différents agents. L'environnement n'est pas statique avec plusieurs agents et le DQN n'est pas efficace.
A la fin de cet apprentissage les archers et les chevaliers continuent de faire des mouvement qui semble aléatoires et les parties se finissent très rapidement.

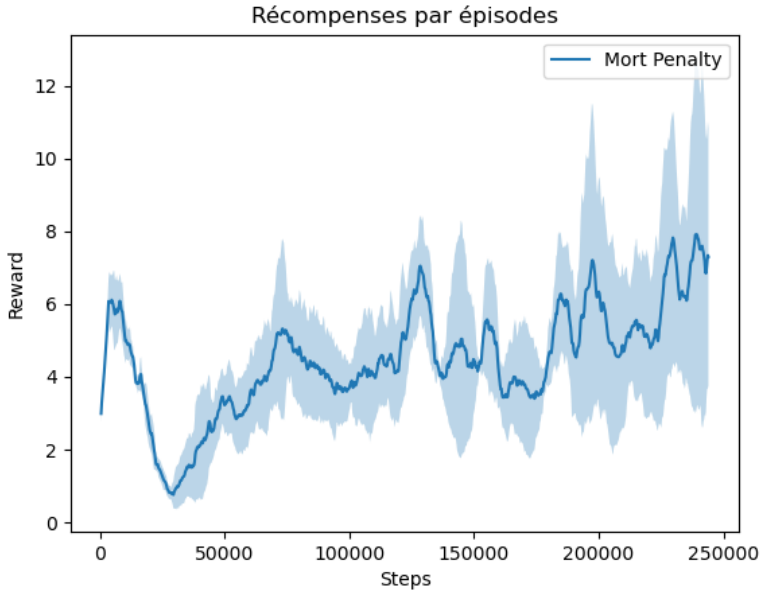
3.2 Un DQN indépendant pour chaque agent

Nous avons ensuite implémenté 1 DQN par agent présent dans l'environnement (avec Stable Baselines3). Cette approche est meilleure car les agents apprennent de façon indépendante dans l'environnement. Cependant ils ne voit que la position des autres agents et ne prennent pas véritablement en compte quelles actions vont faire les autres agents. De plus, les agents ne cherche qu'a optimiser leur récompenses personnelles (nombre de zombies tués).



3.3 Pénalisation de la mort

On a décidé de modifier les récompenses obtenues à chaque tour pour pénaliser le fait de mourir. En effet, jusque là les seules récompenses obtenues sont lorsque l'agent élimine un zombie. Cependant la récompense est rarement obtenue surtout au début et les agents n'évitent donc pas les zombies et sont facilement éliminés. Avec la pénalisation d'un agent quand celui-ci meurt (plus élevée que la récompense d'une élimination), les agents accordent de l'importance à leur vie et jouent plus longtemps, ce qui leur permet d'apprendre à tuer des zombies plus facilement.

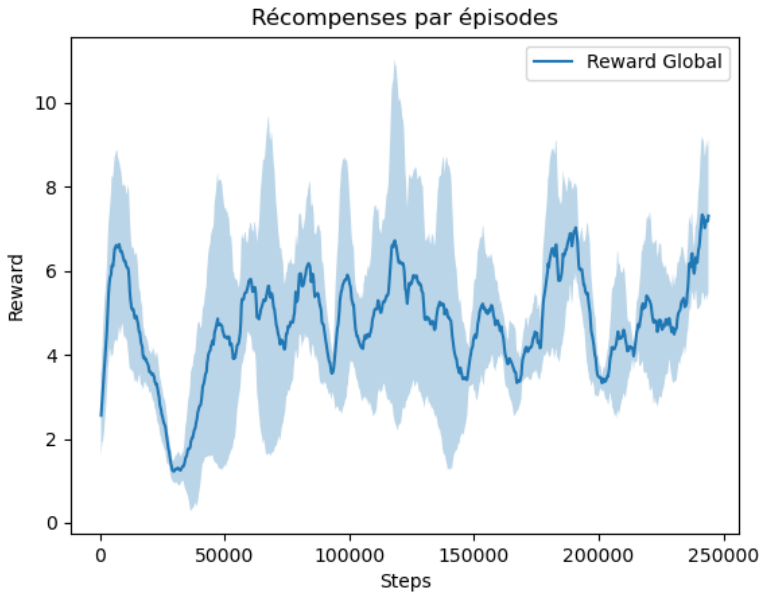


On observe qu'il y a beaucoup moins de variance dans les valeur avec une pénalisation. L'apprentissage est donc plus constant. De plus le temps d'exécution est plus rapide.

3.4 Récompense globale

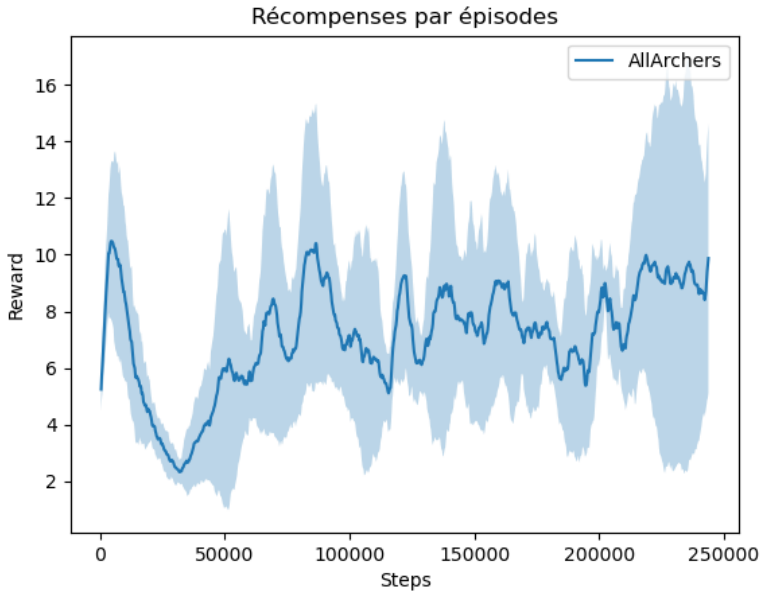
On a essayé de rajouter une récompense commune à tous les agents pour les encourager à coopérer. À la fin de chaque épisode, on ajoute la moyenne des récompenses à chaque

agent. Cette modification n'a pas été très efficace. En effet, la courbe est très similaire à celle avec la pénalisation de la mort. Mais peut être faut il plus de temps d'entraînement pour véritablement voir une meilleure coopération.



3.5 Environnement avec 4 Archers

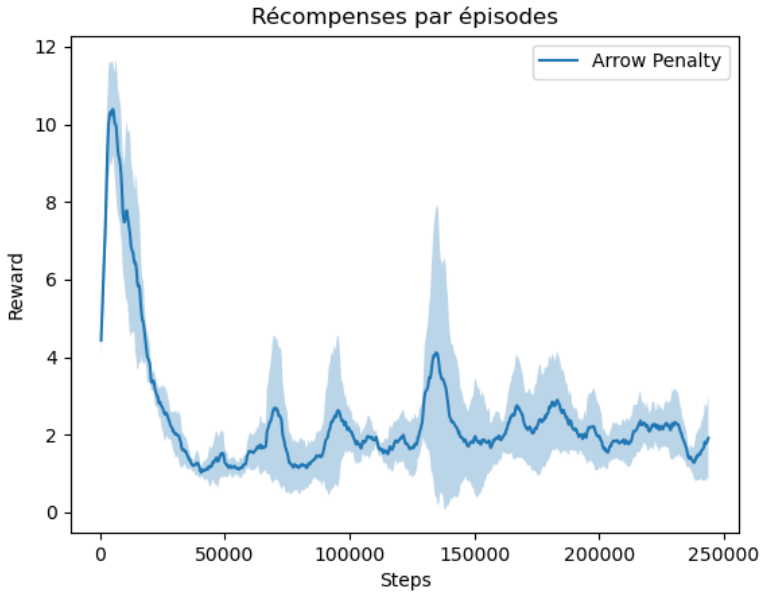
Nous avons remarqué l'inefficacité des chevaliers qui ont plus de mal a apprendre car la seule récompense positive est celle de tuer un zombie. De plus l'apprentissage est aussi très dangereux pour eux vu qu'ils sont au corps à corps. Vous avons donc testé un environnement avec 4 archers au lieu de 2 archer et 2 chevaliers.



Comme vous le voyez un agent apprend beaucoup plus que les autres. Cela est du a la limite du nombre de flèches dans l'environnement. Si deux archers font l'action de tirer une flèche au même moment et qu'il ne reste qu'une flèche disponible alors ce sera toujours le premier archer (l'ordre ne change pas tout le long de l'apprentissage) qui va tirer sa flèche. Il y aura donc plus de chances de toucher un zombie et donc d'avoir une récompenses. Il y a donc une forme de compétition entre les archers.

3.6 Pénalisation du nombre de flèches tirées

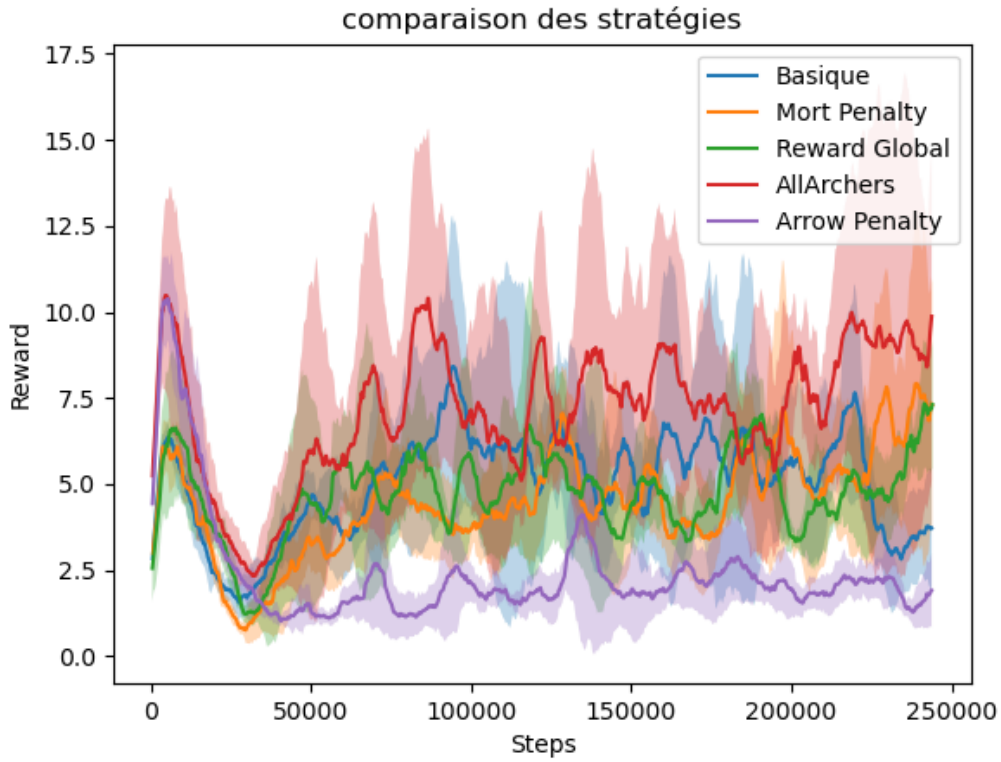
Pour essayer de remédier a cette compétition nous avons testé de pénaliser l'envoi d'une flèche pour éviter le monopole des flèches par un agent.



On observe que plutôt que d'aider cette modification empêche les archers d'apprendre correctement. Leur performance est donc moins efficace que sans la pénalisation. Il faudrait donc une autre façon d'éviter la compétition pour les ressources partagées.

4. Conclusion

Voici la comparaison du niveau de récompenses (nombre de zombies) pour chaque approche.



On peut voir que même avec des stratégies pour améliorer l'apprentissage, l'application de ces approche n'est pas très efficace. En effet, l'approche avec seulement des archers est légèrement meilleure et l'approche pénalisant les flèches est moins performante. Cependant les autres approches malgré l'intention d'augmenter les performances ne sont pas très efficaces. On peut expliquer cela par la complexité de l'environnement qui donne des récompenses en différé de l'action, l'aléatoire des mouvements des zombies qui est mal considéré dans l'apprentissage, mais surtout par l'environnement non-statique quand un agent apprend. De plus un environnement aussi complexe demande un apprentissage plus long que 250 000 épisodes, mais non machines étant trop lente pour lancer des phases d'apprentissage plus longues.