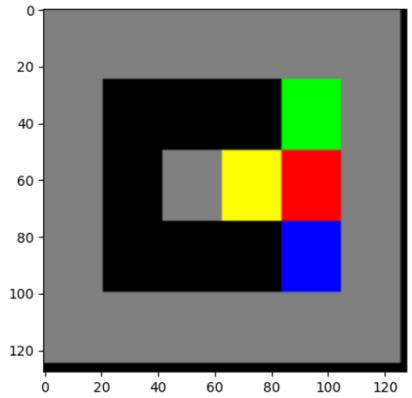
Rapport RLD TME3

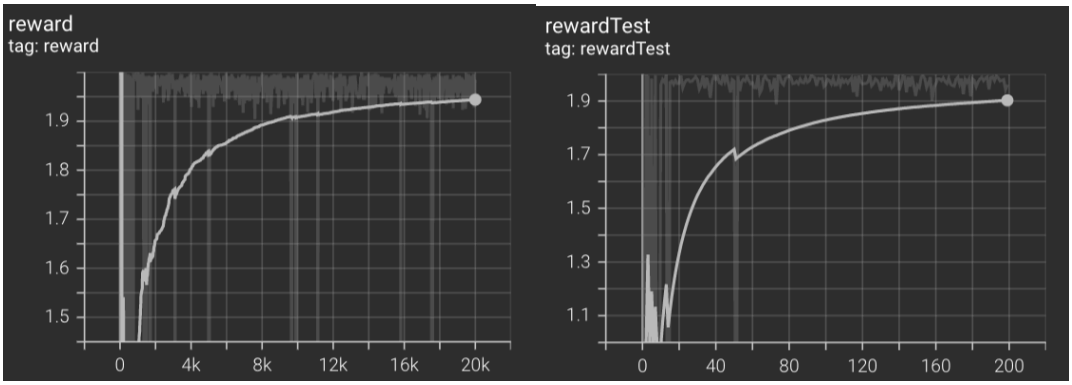
Christopher Heim et Thomas Floquet

Hyperparamètres (sauf indication contraire) :
Epsilon-greedy = 0.5
Decay = 0.999
Discount = 0.99
Learning rate = 0.1
nbModelSamples = 100
nbEpisodes = 20 000

Plan 1 :

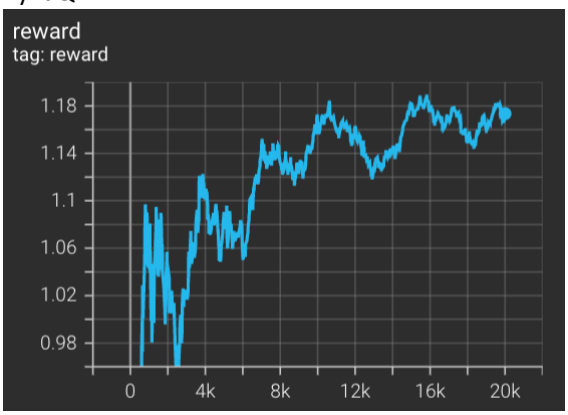


QLearning :



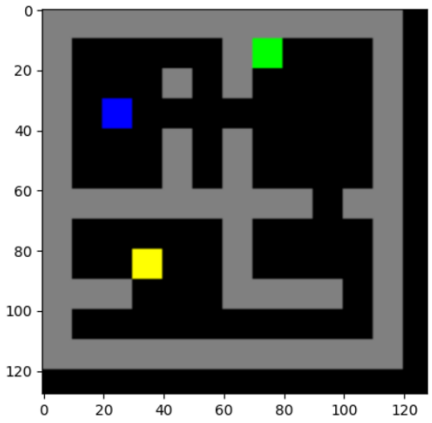
Sarsa :

DynaQ :

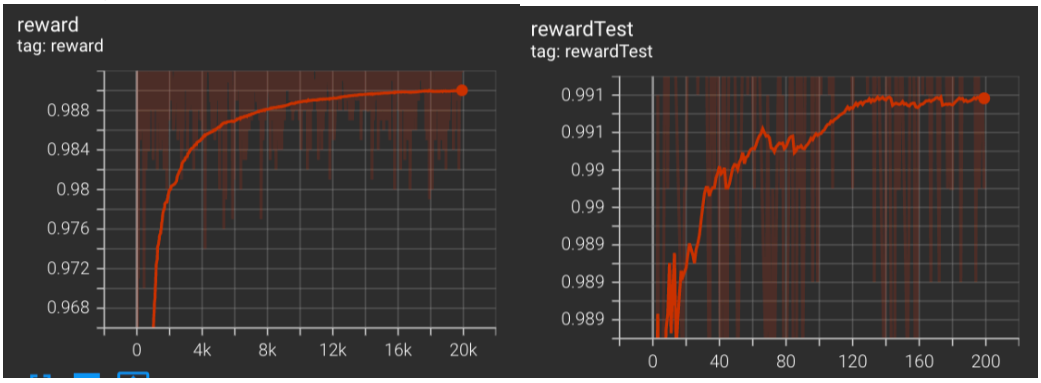


QLearning converge un petit plus vite que Sarsa. Sarsa et QLearning obtiennent un meilleur reward que DynaQ. Peut-être que cela s'explique par le fait que la meilleure stratégie est de toujours aller vers le bas au début, et donc DynaQ reste bloqué plus longtemps (car il apprend le MDP), ce qui diminue le reward
DynaQ a une plus grande variance

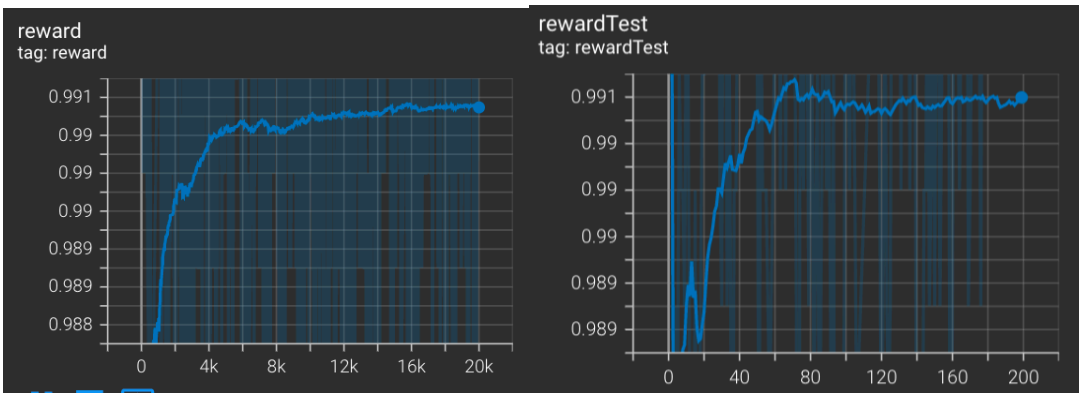
Plan 5 :



QLearning :



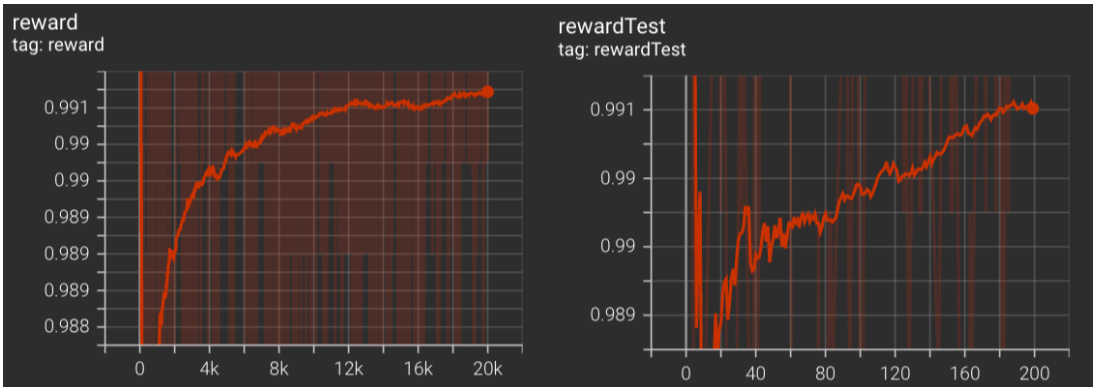
SARSA :
On obtient un légèrement meilleur reward qu'avec QLearning, mais comme avec QLearning, on reste bloqué sur une trajectoire sous-optimale (l'agent ne va pas chercher le carré jaune tout en bas)



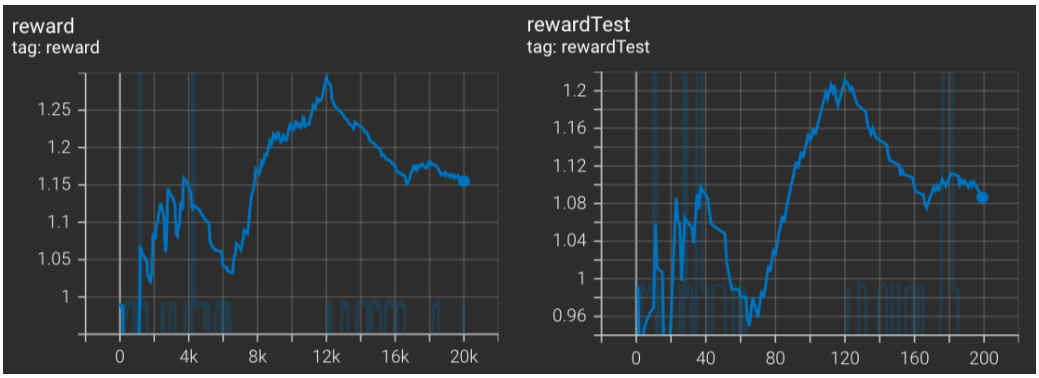
DynaQ :

Meilleur reward et nécessite moins d'itérations pour converger. Ne reste pas bloqué sur une trajectoire sous-optimale car va chercher le carré jaune en bas de la carte grâce à l'échantillonnage des *nbModelSamples* couples (état, action)
Variance importante

DynaQ avec nbModelSamples = 5 :
Itérations beaucoup plus rapides qu'avec nbModelSamples = 100, mais résultats moins bons (l'agent ne va jamais chercher la récompense en bas, reste bloqué sur trajectoire sous-optimale)
Plus nbModelSamples est petit, plus on se rapproche des résultats de QLearning car on ne met à jour que très peu de couples (état, action)

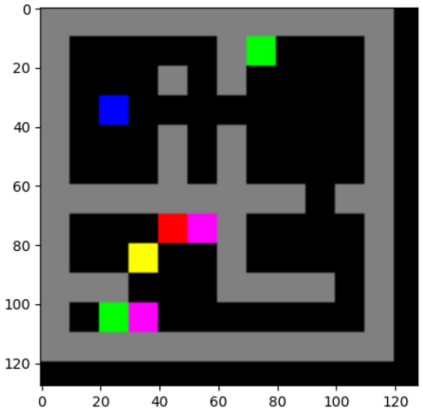


DynaQ avec nbModelSamples = 200 :

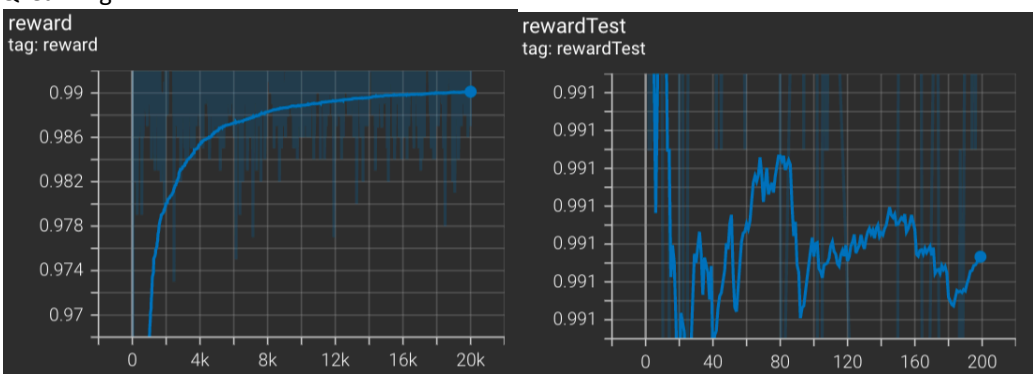


Résultats assez similaires au cas où nbModelSamples = 100, mais les itérations sont beaucoup plus longues car on met à jour deux fois plus de couples à chaque fois

Plan 6 :



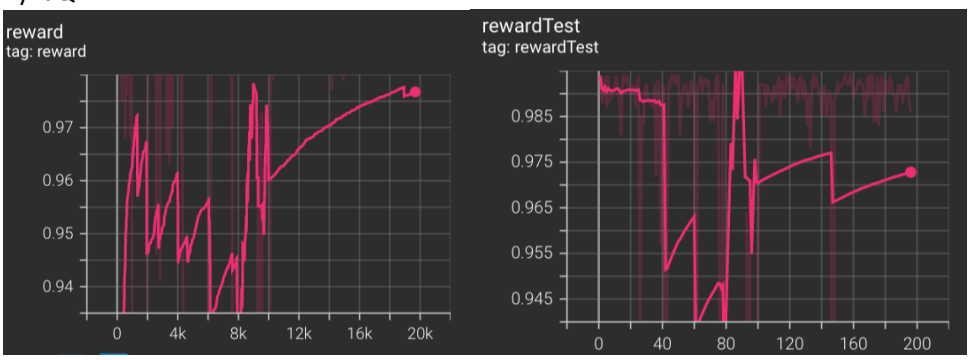
QLearning :



Sarsa :

Résultats très similaires à QLearning
L'agent ne va pas chercher la récompense jaune en bas, on reste bloqué sur une trajectoire sous-optimale

DynaQ :



Assez instable : est déjà parvenu à aller chercher la récompense en bas puis à revenir (stratégie qui rapport le plus de points), mais globalement reste souvent bloqué sur trajectoire sous-optimale
Peut-être qu'avec davantage d'itérations on pourrait converger vers la solution optimale