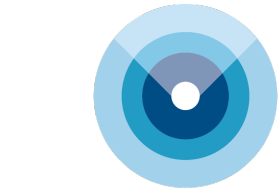

Université Côte d'Azur

Master Informatique, parcours Intelligence Artificielle

Project report

Apprentissage par Renforcement sur Mario Bross

Authors :
DE SEROUX Colin & HADDOU Amine

Supervised by :
MARTINET Jean

Due Date :
21 octobre 2024

Apprentissage par Renforcement

CouscousSamba

21/10/2024

Table des matières

1 Introduction

L'objectif de ce projet est de développer un modèle d'apprentissage par renforcement appliqué au contexte des jeux vidéo. Nous avons choisi d'utiliser le célèbre jeu *Super Mario Bros* comme environnement d'entraînement pour notre modèle. L'idée est de faire en sorte que le modèle puisse apprendre à terminer un niveau personnalisé en franchissant des obstacles tout au long du parcours.

2 Description de l'environnement

Le modèle doit parcourir une carte personnalisée comprenant des obstacles tels que des trous, des blocs qui obstruent le passage et d'autres qui empêchent les sauts. Le modèle dispose de deux actions possibles : avancer ou sauter. L'action "sauter" permet au modèle de se déplacer de deux cases en avant, en sautant au-dessus de la case adjacente.

3 Algorithmes d'apprentissage

Pour entraîner notre modèle, nous avons décidé d'explorer deux types d'algorithmes d'apprentissage par renforcement : *Q-Learning* et *Monte Carlo Tree Search* (MCTS). Dans ce rapport, nous détaillerons dans un premier temps la mise en œuvre du *Q-Learning*, tandis que la partie concernant *MCTS* sera traitée ultérieurement.

3.1 Q-Learning

Le *Q-Learning* est un algorithme d'apprentissage par renforcement hors-ligne qui vise à apprendre une politique optimale pour maximiser le cumul des récompenses au fil du temps. Dans notre cas, nous avons paramétré le modèle de la manière suivante :

- **Epsilon-greedy strategy :** Nous avons initialisé ϵ à 1 avec un taux de décroissance de 0.001. À chaque épisode, ϵ est mis à jour selon la formule suivante :

  $$\epsilon = (\text{episodes} - \text{episode}) \times \text{decay}$$

  Cela permet de s'assurer que l'exploration est dominante au début de l'entraînement, tandis que l'exploitation devient progressivement plus importante en fin d'apprentissage.

- **Fonction de récompense :** Un trou sur le parcours fait perdre 100 points (-100), tandis que terminer un niveau rapporte 100 points (états finaux). Il est également possible de collecter des pièces qui rapportent 50 points. À chaque pas, une pénalité de -1 est appliquée pour inciter le modèle à optimiser ses mouvements et terminer le niveau plus rapidement (300 épisodes pour un modèle finissant le jeu de manière poussive).

- **Paramètres d'apprentissage :** Nous avons choisi un taux d'apprentissage de 0.1 et un facteur de discount (γ) de 0.99 pour mettre l'accent sur les récompenses futures.

4 Résultats et Conclusion

L'implémentation de notre modèle d'apprentissage par renforcement à l'aide de l'algorithme *Q-Learning* a été couronnée de succès. Le modèle a réussi à apprendre efficacement à éviter les obstacles et à collecter les pièces tout en atteignant l'état final, qui est la fin du niveau. L'utilisation de la stratégie ϵ-greedy a permis un équilibre adéquat entre exploration et exploitation au cours de l'entraînement, tandis que la fonction de récompense et les paramètres d'apprentissage ont favorisé une convergence rapide vers une politique optimale.