Feuille de route
Projet d'IA appliquée à µRTS

PAINCHAULT Emmanuel
TRABELSI Ayoub
RASSAT Marian

Introduction

L'utilisation d'IA dans le domaine du jeu vidéo n'est pas récente, et se trouve être dans la plupart d'entre eux un des aspects essentiels afin de fournir une expérience qualitative aux joueurs.

Elle peut aussi permettre de s'adapter au niveau du joueur en proposant plusieurs niveaux de difficultés. Une des façons de faire autrefois était d'ajuster des mécaniques telles que les points de vie ou de dégâts, aujourd'hui il est devenu accessible de pouvoir influencer la stratégie qu'adopte l'IA durant son exécution.

L'utilisation des réseaux de neurones a permis dans le domaine de l'IA une certaine révolution en permettant de basculer une grande partie de la conception d'une IA vers son entraînement. C'est là une avancé majeure qui a notamment permis la conception d'AlphaGo dans le début des années 2010 permettant de rivaliser face aux humains au jeu de go, des échecs ainsi qu'au shōgi.

La complexité de la réalisation d'une IA pouvant effectuer cette tâche résidait dans le nombre de combinaisons possibles d'une partie de ces jeux. Cela à donc était rendu possible grâce à l'utilisation de réseaux de neurones ainsi que d'un entraînement poussé afin de la perfectionner.

1 Spécifications fonctionnelles

L'objectif de ce projet est, dans un premier temps, de se familiariser avec l'algorithme dit PPO, afin d'être capables d'expliquer au mieux possible le fonctionnement de celui-ci de manière générale, mais également au sein de µRTS, le jeu dont il est question ici.

Par ailleurs, en accomplissant cette tâche, il nous sera impératif de se familiariser avec les environnements "gym" permettant l'entraînement d'une IA à un jeu ou à une tâche. Ainsi il sera question de programmer une implémentation de l'algorithme étudié au sein d'un environnement de recherche récent (plusieurs papiers de recherche [1]utilisant cet environnement ont déjà pu être publiés entre 2019 et 2021).

Enfin, et c'est le réel objectif final du projet, nous devrons obtenir des résultats non-triviaux de la part de notre IA, c'est-à-dire une IA qui aura une réelle stratégie au long d'une partie, et qui pourrait battre un humain débutant ou intermédiaire au jeu.

2 Contraintes techniques

µRTS est un jeu développé en Java. Ainsi, celui-ci sera essentiel au bon fonctionnement du projet. Il sera donc installé sur tous les environnements sous lesquels nous travaillerons.

De plus, l'environnement Gym qui servira d'espace d'entraînement à notre IA est programmé en Python. Nous l'utiliserons donc également. Ce dernier a aussi pour deuxième avantage que beaucoup de modules déjà existants permettent la création de modèles d'IA, ce qui sera certainement un atout majeur dans l'avancement et le bon déroulement du projet.

Le code sera déposé sur Git, et chacun pourra l'exécuter de la manière dont il l'entend. Puisque Python et Java sont tous deux cross-platform, il ne sera pas nécessaire d'avoir un environnement commun, partagé ou même similaire pour travailler.

3 Principales étapes envisagées

Plusieurs étapes à réaliser peuvent déjà se dégager de tout cela :

- Se renseigner sur les algorithmes PPO (le principe général, la manière de fonctionner dans le cas des RTS ou plus généralement des jeux, etc.).
- Apprendre à utiliser Gym pour entraîner une IA et en sortir des résultats utilisables (statistiques, model fitness, "Trueskill").
- Utiliser l'implémentation déjà existante de l'algorithme PPO afin d'obtenir une IA répondant aux critères définis : capable d'avoir une stratégie, de battre un humain, et de jouer sur des maps inconnues une fois entraîné.
- Implémenter notre propre solution utilisant les algorithmes PPO dans l'environnement Gym et obtenir des résultats convaincant si le temps le permet.
- Modifier et améliorer le modèle (hyperparamètres etc.) afin d'obtenir des résultats non-triviaux, relevant d'un apprentissage indéniable, et si possible obtenir au final une IA capable de jouer très bien contre des humains, sur des maps de tailles et de configurations inconnues, ou encore avec des stratégies variées.

4 Répartition des tâches

Au début du projet, le temps de chacun sera alloué à la compréhension approfondie des détails de celui-ci, à savoir :

- Les algorithmes PPO, leur fonctionnement ainsi que la façon de les implémenter.
- Comment "wrapper" µRTS afin de pouvoir simuler un joueur, et récupérer certaines informations essentielles pour l'entraînement et l'évaluation d'une IA.

Ces deux étapes pouvant très bien nécessiter plus d'une personne en raison du grand nombre d'informations à aller chercher impliquera qu'une personne pourra être amenée à changer de tâche suivant l'avancement de l'une ou l'autre. Il est ainsi difficile de dire qui fera quoi personnellement, mais le tout sera détaillé dans les rapports sur l'avancement du projet.

Enfin, chaque personne pourra s'occuper de l'une de ces tâches :

- Utiliser Gym-µRTS afin d'entraîner une IA implémentant PPO, en comprenant le processus qui se déroule en amont pour obtenir un comportement non trivial.
- Concevoir une IA implémentant notre propre version de PPO, pouvant utiliser le wrapper Gym-µRTS et qui réussira à obtenir un comportement non trivial.

5 Sources

[1] GitHub MicroRTS-Py
Publication sur l'apprentissage renforcé dans les RTS

6 Bibliographie

[1] The Combinatorial Multi-Armed Bandit Problem and Its Application to Real-Time Strategy Games (Santiago Ontañón 2013)

[2] Proximal Policy Optimization Algorithms
(John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov 2017)