Billaud Pierre / Couturieux Victor / STMN P5

Intelligence artificielle : Halite III

Compte rendu

Introduction

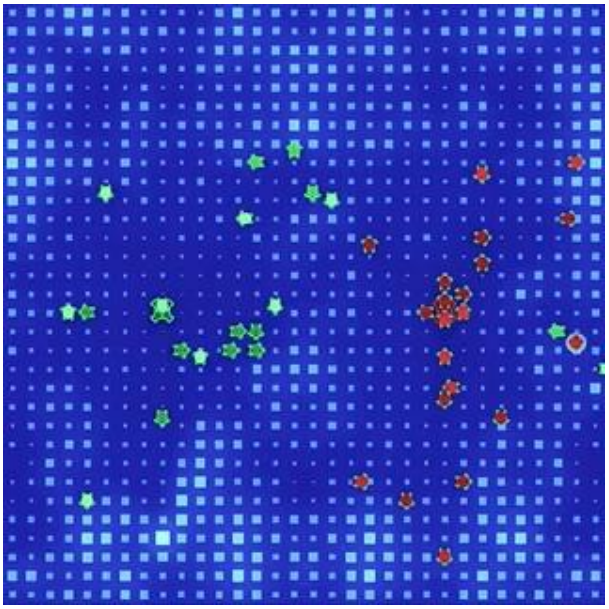
Ce compte rendu a pour objectif principal de présenter et expliquer sous différents angles le « bot » que nous avons réalisé dans le cadre du cours « Intelligence Artificielle » tutoré par M.Buendia. À ce titre, nous allons aborder principalement tout ce qui a trait à l'intelligence artificielle sur le côté technique, et sur sa conception de manière générale. Nous allons nous appesantir sur la manière dont nous avons organisé le code pour créer cette IA. Nous reviendrons également sur les difficultés que l'on a rencontrées et sur les points principaux à améliorer. Pour finir, nous exprimerons notre ressenti respectif sur le travail réalisé.

Readme

call make.bat pour compiler

halite.exe --replay-directory replays/ -vvv --width 32 --height 32 "MyBot.exe" "MyBot.exe" pour executer le bot

Sur certains ordinateurs, il est possible que le dossier « saves » ne se créé pas automatiquement lors de l'exécution du bot. Auquel cas, veuillez créer ce dossier pour pouvoir utiliser l'algorithme génétique pleinement.



I_Fonctionnalités

Toutes les fonctionnalités basiques d'Halite ont été implémentées à ce jour :

- Déplacements des bateaux dans les quatre directions

- Récolte

- Création de base de collecte

- Création de bateau

Remarque : On décrit ici les fonctionnalités de base d'Halite III, et non pas les comportements et les logiques stratégiques.

Source : https://2018.halite.io/learn-programming-challenge/game-overview

II_Architecture globale

Cette IA est globalement divisée en deux parties. Une partie dédiée aux mécaniques de jeu et aux comportements que le bot va avoir en fonction des données du jeu (de la répartition des halites). Une autre partie dédiée à l'algorithme génétique qui va, en fonction des résultats des parties jouées, faire muter des paramètres et sélectionner peu à peu les paramètres dont le taux de réussite est élevé.

- Bot – Fonctionnement / caractéristiques

Partie I : on récupère les données de la carte et on détermine la zone de 9 cases qui est la plus intéressante (notée en fonction de trois paramètres : du nombre d'halite / nombre d'ennemis sur les cases / distance du chantier naval). En faisant varier ses paramètres, les bateaux vont être plus ou moins peureux, vont privilégier plus ou moins la prudence.

Partie II : Une fois la zone déterminée, on récapitule l'état de tous nos bateaux : soit ils sont pleins (et retourne à la base), soit ils sont en exploration, soit ils sont en collecte, soit ils sont nouveaux. Ce statut sert ensuite à déterminer leur prochaine action.

(Partie II.5 : C'est ici que l'on envisage (en fonction de nos halites) si on créé un bateau ou non sur le chantier naval.)

Partie III : On choisit en détail l'action que chaque bateau va réaliser en fonction de son statut (on va indiquer une direction, le transformer en base etc..).

Partie IV : Une fois toutes les actions choisies, on anticipe les éventuelles futures collisions des bateaux et on modifie en conséquence les actions puis on renvoie celles-ci. On regarde également si aucun bateau ne gêne la création d'un nouveau bateau.

- Algorithme génétique

L'algorithme génétique va principalement agir sur cinq paramètres de l'IA (5 paramètres que la documentation d'Halite III recommande de faire varier dans le contexte d'un algorithme génétique) :

- Maximum distance the ship "sees" when looking for halite-rich cells (1 to 32)
- Maximum halite at which a ship moves off a tile (0 to 1,000)
- Maximum halite at which a ship returns to a shipyard or dropoff (0 to 1,000)
- Number of ships to produce in a game (1 to 100)
- The point in the game when the bot spawns no more ships (turn number 1 to 500)

Source : **https://2018.halite.io/learn-programming-challenge/tutorials/ml-ga**

Dans son fonctionnement, à chaque partie jouée cet algorithme va enregistrer les cinq paramètres donnés et va leur associer un résultat (qui correspond à un taux d'halite récoltés, à un nombre de bateaux créés etc.). À chaque nouvelle partie, il aura donc une base de données plus fournie.

Au tout début de chaque partie il applique une sélection par tournoi, un accouplement et une mutation qui déterminent les cinq variables ci-dessus, qui vont ensuite être enregistrées à leur tour comme données d'entrée et qui vont ensuite donner un résultat associé.

III_Comportements / logiques stratégiques

Comme notre bot évalue les zones à récolter en fonction d'une note répartie en trois points (nombre d'halite / nombre d'ennemis sur les cases / distance du chantier naval), en faisant varier l'importance de chaque point, le comportement du bot sera plus ou moins craintif etc.

Globalement, l'IA que l'on a réalisée reste assez opportuniste car même si on ne cible qu'une seule zone, à partir d'un moment de la partie, la zone peut être amenée à changer à chaque tour.

L'une des faiblesses de notre bot est donc sa versatilité, ce qui peut lui faire perdre beaucoup de halite en déplacements.

IV_Difficultés rencontrées

Les difficultés principales ont été de comprendre le jeu en lui-même et de réfléchir à un mode de fonctionnement qui était pertinent pour ce type de jeu. En d'autres mots, élaborer le bot en lui-même et sa logique de jeu a été la partie la plus difficile. Beaucoup d'effets de bord apparaissent rapidement (des petites logiques auxquelles on n'a pas pensé). Avoir un rendu à peu près cohérent est difficile à mettre en place.

V_Points à améliorer

Nous avons conscience que l'algorithme génétique implémenté en tant que tel n'apporte qu'une efficacité relative au bot car les données recueillies sont issues d'un seul et même type de combat : des combats contre la même IA.

Par ailleurs, nous avons conscience que l'algorithme génétique mériterait à avoir plus de paramètres à faire varier, à avoir des données plus précises à évaluer et plus de fonctions qui aident à faire converger les résultats. Actuellement les données ne convergent pas assez vite, notamment en raison du fait que les parties se font une par une. Par ailleurs les résultats des parties n'ont pas assez de données à évaluer ce qui limite l'efficacité de l'algorithme génétique.

D'un autre côté, le bot mériterait à avoir des ajustements sur de nombreuses variables (le nombre de bateaux maximum qu'il peut créer, l'évaluation des zones de récoltes etc.). Cela demande de nombreux tests à chaque nouvelle itération. S'ils avaient été disponibles, bénéficier de tous les outils du site d'Halite III aurait été intéressant (et sans doute un gain de temps).

VI_Avis / ressentis

Pierre

Beaucoup de frustrations en début de projet, liées à la compréhension du sujet et au fait de devoir s'immerger sérieusement dans la documentation / codes pour comprendre les bases d'Halite III. On a réalisé une première version du bot qui fonctionnait très mal (moins de 5% de récolte sur la map au total) et on a décidé de changer de nombreux points ensuite (quasiment la moitié). Ça a pris du temps et le moral en a pris un coup. Cela étant, la seconde partie du projet s'est beaucoup mieux déroulée et au final (ironiquement) je pense qu'on est arrivé au moment du projet où en fait les choses commencent **à être intéressantes**.

C'est-à-dire que l'on a bien compris comment améliorer le bot, le tester (faire varier des paramètres etc.) et comment le réfléchir efficacement. On a la base pour faire un algorithme

génétique plus performant et on a la base d'un bot consistant. Au final, on a réussi à faire une base satisfaisante qui peut bien progresser si on avait plus de temps.

Victor

Sujet très intéressant dans l'ensemble. Afin de mieux comprendre son fonctionnement, il était utile de s'inspirer d'autres projets déjà existants. En les reproduisant, nous ne pouvions que mieux appréhender son fonctionnement. Cela constitue la phase de recherche et apprentissage du sujet. Après avoir pris en main le concept d'Halite et en croisant avec nos compétences, nous avons donc été en mesure de proposer une IA personnelle en liant une mise en œuvre rapide et pertinente.

Cependant, cette expérience reste quelque peu frustrante. En effet, son élaboration pouvant encore être étudiée et améliorée, nous avons dû restreindre notre travail afin de tenir les délais de soumission du projet.

Répartition du travail

On s'est réparti le travail à 50% chacun. On a fait le choix de ne pas indiquer de nom sur les fichiers (qui a fait quoi) car ça n'aurait pas été révélateur du travail de chacun : nous avons apporté notre point de vue sur presque chaque fonctionnalité du point de vue de la conception et du travail d'écriture du code.