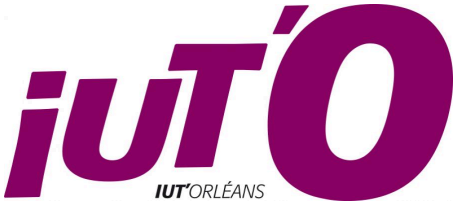
Audor Etienne Monet Bastien Laurin Alexandre



RAPPORT SAE PYTHON

1. Stratégie de l'ia :

La stratégie de notre IA se base essentiellement sur la fonction *objet_voisinage*, qui nous indique de nombreuses informations sur les objets ou serpents présents à proximité.
Notre IA a pour but principal de manger les boîtes afin d'avoir la plus grosse tête le plus tôt possible. Elle mange aussi, dans un premier temps, les serpents qui se trouvent juste à côté d'elle et ramasse (si la distance est rentable) les bonus de protection et de surpuissance.

À l'aide de ces informations (distance, valeur, numéro du serpent, chemin le plus court), nous utilisons la propriété des dictionnaires Python, qui se trient par défaut, pour obtenir en premier les objets les plus proches. Par la suite, nous vérifions que la valeur de la tête de notre serpent est assez élevée pour se déplacer sur une case. Nous évaluons aussi la distance par rapport à la durée de disparition des objets alentour.

Avec tous ces paramètres, nous demandons à notre joueur d'atteindre les cases les plus proches contenant des bonus, des boîtes ou des serpents plus petits que sa tête. Ensuite, si notre serpent a une valeur de tête supérieure à 16, nous nous attaquons davantage aux joueurs adverses, car atteindre une tête de 32 est très long, et manger les autres joueurs rapporte plus de points tout en freinant le score des autres groupes.

2. Implémentation:

La fonction *direction_possible* nous permet d'évaluer les directions possibles afin que notre IA ne se déplace ni hors du terrain ni dans un mur. La fonction est en O(1) et est parfaitement fonctionnelle.

Dans un second temps, nous avons réalisé la fonction *objet_voisinage* à l'aide du principe d'inondation vu en cours. Nous créons alors un calque de l'arène pour estimer les

déplacements possibles à une distance maximale de notre tête, puis nous gardons en mémoire les positions des cases importantes à proximité.

Le calque nous permet ensuite d'obtenir le plus court chemin sous forme d'une liste des positions à atteindre pour accéder aux différentes cases importantes. Cette liste est ensuite convertie en une succession de déplacements nord-sud-est-ouest. Le tout est stocké dans un dictionnaire.

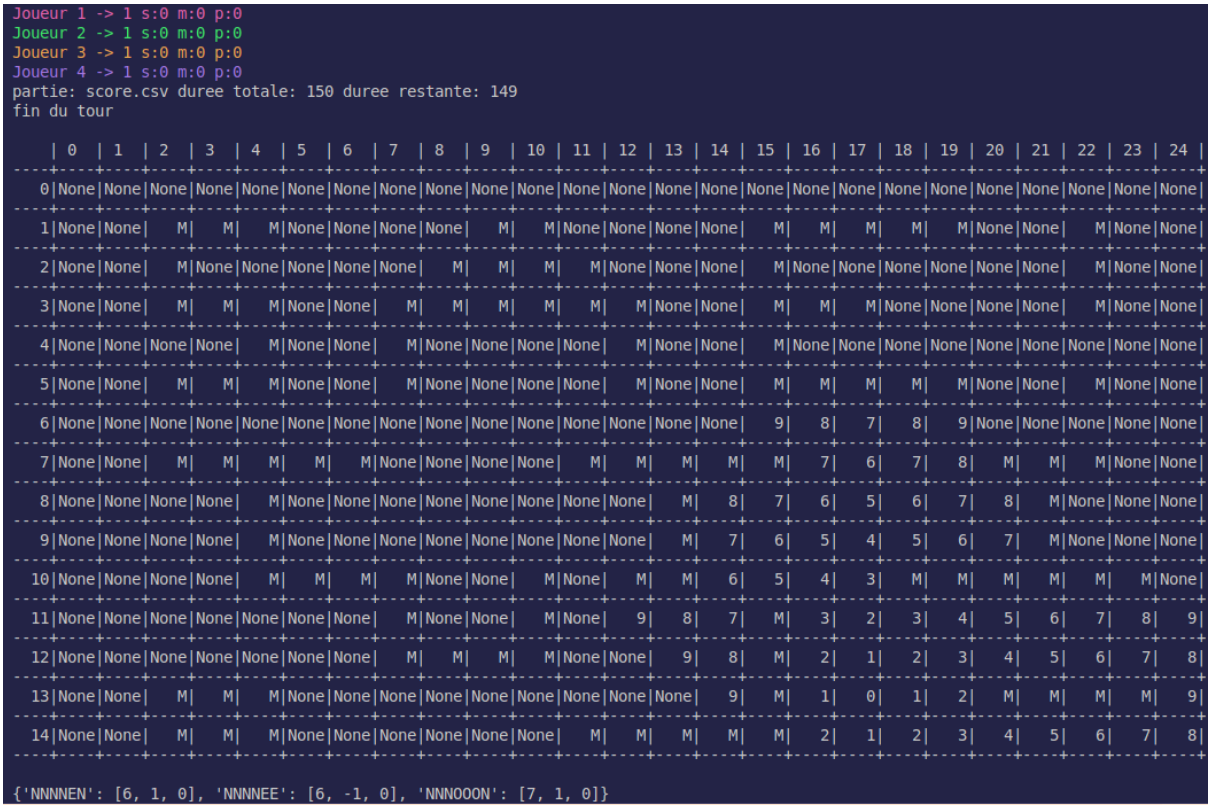


figure 1

Sur la figure 1 ci-dessus, vous pouvez observer le calque de la fonction *objet_voisinage*. La tête de notre serpent est représentée par la valeur 0, et les M représentent les murs. Vous pouvez ainsi observer l'inondation se former grâce aux chiffres de 1 à 9, qui correspondent au nombre de déplacements nécessaires pour atteindre chaque case.

Enfin, tout en bas de la figure 1, vous pouvez également remarquer le retour final de *objet_voisinage*, que nous pouvons interpréter ainsi : il y a 3 objets à une distance de moins de 9 cases de notre tête, deux boîtes de valeur 1 à 6 et 7 cases de notre serpent, et un bonus additionnel à 6 cases de là.

La fonction a une complexité finale en O(N2)×dist_max de l'inondation, qui est défini à 9 dans notre IA.

Ce que nous n’avons pas (encore) réussi à faire :

Nous n’avons malheureusement pas réussi à effectuer une refactorisation de la fonction *mon_IA*, afin de réduire la longueur du script et de gagner en lisibilité.

3. Compte rendu personnel

Bastien

Durant cette SAE, j’ai tout d’abord aidé mon groupe en leur expliquant l’utilisation d’outils de travail d’équipe comme GitHub et Live Share. Nous avons souvent utilisé GitHub, ce qui nous a permis de maintenir une bonne coordination.

J’ai participé au remplissage du fichier *serpent.py* et à la création des deux fonctions *distance_possible* et *objet_voisinage*, étant le seul de mon groupe à posséder les connaissances acquises lors du TP12.

Enfin, j’ai contribué à la conception de l’IA en corrigeant des erreurs de programmation et en participant à la réalisation de quelques fonctions, comme *is_protection*, *get_val_temps* et *get_case_from_chemin*.

Etienne

Tout d’abord, je tiens à dire que c’est l’une des premières fois que je participe à un projet de groupe de cette ampleur. Cette SAE m’a permis de créer une bonne cohésion de groupe (du moins avec ceux qui étaient vraiment impliqués dans le projet). De plus, cela m’a permis de m’exercer à l’utilisation de GitHub, mais aussi sur le langage Python. Pour ma part, je me suis chargé de remplir la plupart des fonctions de *serpent.py*. J’ai notamment aidé à la conception de la fonction *objet_voisinage* (qui était pour moi le vrai défi de cette SAE). Par la suite, j’ai élaboré une stratégie, puis je l’ai réalisée avec l’aide de Bastien. Nous avons, pour finir, continué à améliorer notre IA grâce aux multiples tests, en la faisant s’affronter avec différents niveaux d’IA que nous avons créés uniquement pour les tests.

Pour conclure, même si le projet est, selon moi, abouti, je suis déçu du comportement de l’un des membres de mon groupe, qui a passé tout son temps sur son téléphone plutôt qu’à nous aider.

Alexandre

Durant cette SAE, j’ai contribué au remplissage de l’API *serpent.py* en m’occupant de créer les fonctions de type “set”. J’ai également apporté mon aide à ceux qui en avaient besoin.

Lors de cette SAE, j’ai utilisé GitHub pour la première fois, ce qui m’a permis de mieux communiquer avec mon groupe. J’ai proposé différentes stratégies (bonnes ou mauvaises) à mon groupe, qui a choisi celles qu’il jugeait pertinentes. Certaines stratégies ont malheureusement dû être abandonnées par manque de temps ou en raison d’une trop grande complexité de programmation.

Ce projet reste une belle expérience malgré sa longueur, car il est rare de travailler sur un projet qui dure une semaine entière, sans interruption.