



SA1.02 COMPARAISON D'APPROCHES ALGORITHMIQUES

1 Compétences ciblées

Cette SAÉ vise à travailler sur la **compétence 2** *Appréhender et construire des algorithmes*.

Les apprentissages critiques visés sont les suivants :

- **AC12.01** Analyser un problème avec méthode (découpage en éléments algorithmiques simples, structures de données...)
- **AC12.02** Comparer des algorithmes pour des problèmes classiques (tris simples, recherche...)

Cette SAÉ se fera en groupe de 3 ou 4. Ces groupes vous sont imposés. Il y aura plusieurs rendus à faire :

- un rendu le **mercredi 14 janvier 9h** sur l'implémentation des fonctions d'une API ainsi que des fonctions de copie
- un rendu le **vendredi 17 janvier 11h** concernant l'IA que vous aurez développée,
- un devoir sur table le **vendredi 17 janvier matin** sur les apprentissages critiques de cette SAÉ.






La SAÉ se terminera par un tournoi le **vendredi 17 janvier** après midi où vos IA s'affronteront.

2 Présentation du jeu

SerpIUT'O est un jeu inspiré des jeux Snakes et Boxes. Il consiste à essayer de faire grandir son serpent en accumulant des boîtes qui ont des valeurs. Le serpent peut se raccourcir s'il contient 2 boîtes consécutives de même valeur. Celles-ci vont fusionner au bout d'un certain temps en une boîte ayant le double de valeur. Pour qu'un serpent puisse manger une boîte, sa tête doit avoir une valeur supérieure ou égale à la boîte avalée. Un serpent peut croquer une des boîtes d'un adversaire. Dans ce cas, il gagne en points la somme de toutes les boîtes de la queue du serpent mordu. Si un serpent essaie de manger une boîte trop grosse pour lui ou s'il se cogne dans un mur, la valeur de toutes les boîtes qui le composent est divisée par 2 (et les boîtes qui valaient 1 disparaissent). Un serpent complètement disparu réapparaît sur une case vide de l'arène de jeu.



A chaque tour de jeu, le joueur gagne en points la valeur de la tête de son serpent. Un certain nombre de bonus permettent aux serpents soit d'avoir des pouvoirs supplémentaires soit avoir des points en plus. Le fait d'avaler un bonus coûte quelques points.

- Les boîtes standard qui rapportent 1 ou 2 points
- L'addition  coûte 1 point. Elle permet d'allonger le serpent d'une boîte de la même valeur que sa queue.
- La multiplication  coûte 2 points. Elle permet de multiplier par 2 la boîte de queue du serpent.
- Le mange-mur  coûte 3 points. Il permet de manger les murs internes de l'arène en perdant des points. Ce bonus est actif pendant 10 tours.
- La surpuissance  coûte 4 points. Elle permet de manger des boîtes de valeur supérieure à la tête du serpent. Ce bonus est actif pendant 10 tours.
- La protection  coûte 5 points. Elle protège le serpent de toute morsure pendant la durée de vie de ce bonus qui est de 10 tours. Les serpents qui s'attaquent à lui subissent les pénalités d'un serpent qui mord dans une boîte trop grosse pour lui.

Notez qu'un serpent peut cumuler plusieurs bonus.

Au début de la partie, les serpents de chaque joueur sont composés d'une seule boîte de valeur 1. Ils sont disposés de manière aléatoire sur l'arène. Quelques boîtes de valeur 1 ou 2 sont aussi disposées de manière aléatoire dans l'arène.

A chaque tour de jeu, chaque joueur doit décider la direction vers laquelle diriger son serpent. Il a le choix entre Nord, Est, Sud, Ouest. L'ordre d'exécution des décisions des joueurs est aléatoire à chaque tour.

Dans les trois cas suivants :

- l'ordre donné par le joueur est incorrect ou
- l'ordre donné fait sortir le joueur de l'arène ou
- la case d'arrivée contient une boîte de valeur supérieur à sa tête

le serpent reste sur place, le joueur prend un malus de 10 points. De plus, toutes les boîtes de son serpent sont divisées par deux, (les boîtes valant 1 disparaissent). Comme indiqué précédemment, un serpent disparaissant complètement ré-apparaît sur une case aléatoire de l'arène avec juste une tête valant 1.

Si la case d'arrivée contient une boîte de valeur inférieure ou égale à la valeur de sa tête, le serpent avale cette boîte qui ira se placer dans l'ordre décroissant des boîtes. De plus, si la boîte avalée

appartient à un serpent, le joueur gagne la somme des valeurs des boîtes de la fin du serpent mangé. Le serpent mangé lui perd les points gagnés par l'agresseur et perd toutes les boîtes de sa queue. Comme expliqué précédemment, les bonus peuvent changer les règles qui viennent d'être énoncées.

Tous les serpents qui ont réussi leur déplacement, gagnent le nombre de points correspondant à la valeur de leur tête. Une fois les déplacements et leurs effets effectués, les serpents ayant conservé deux boîtes de même valeur pendant 8 tours de jeu verront ces boîtes fusionner en une boîte d'une valeur double. Le serpent gagne alors un nombre de points égal à la nouvelle boîte.

A la fin du tour, de nouvelles boîtes et de nouveaux bonus apparaissent. Par contre toutes les boîtes et bonus non consommés pendant 10 tours disparaissent.

Il faut noter qu'un joueur peut avoir un nombre de points négatif. On peut aussi remarquer qu'un serpent ne disparaît jamais de la partie puisqu'il renaît dès qu'il a perdu toutes ses boîtes.

Lors de la création de la partie, la durée (en nombre de tours) de celle-ci est fixée. La durée de vie des objets est fixée à 10 tours et le temps de fusion à 8. Pour le tournoi les parties se dérouleront avec 4 joueurs.

3 Votre mission

Avec le sujet vous est fournie une implémentation du jeu qui peut se lancer sous plusieurs formes (voir section suivante). Votre mission va consister à implémenter une IA permettant de prendre des décisions à chaque tour de jeu afin de gagner une partie. Avant de faire cette implémentation, vous aurez l'API serpent et des fonctions de recopie à implémenter.

L'IA de votre joueur est une fonction se trouvant dans le module `IA.py`. Il s'agit de la fonction `mon_IA()` qui prend en paramètre votre numéro de joueur et l'état de la partie en cours. La fonction doit retourner une des lettres N,O,S ou E qui indique dans quelle direction vous souhaitez diriger votre serpent. Cette fonction sera exécutée à chaque tour de jeu. La version fournie avec le sujet retourne une direction de manière complètement aléatoire ce qui n'est pas très malin. Pour obtenir un comportement plus intelligent vous aurez à implémenter différentes fonctions qui vous permettront d'évaluer l'intérêt ou le danger de prendre telle ou telle direction. Une partie des fonctions à implémenter vous sont imposées mais vous pourrez implémenter toutes celles que vous souhaitez. Par contre dans le cadre de la SAE il faudra pour chaque fonction implémentée, la documenter avec une docstring évaluer sa complexité.

Pour tester votre IA, vous avez la possibilité

- soit de lancer l'application en mode texte (pas d'environnement client/serveur) via la commande `jeu_mode_texte.py`. L'avantage de ce mode est qu'il est simple à lancer et permet donc plus facilement de faire des tests.
- soit de lancer l'application en mode client/serveur dans les mêmes conditions que le tournoi. L'intérêt de ce mode est de tester votre IA en mode réel. Des explications pour lancer ce mode vous sont fournies plus loin.

4 Les différents codes fournis

Voici la structure du répertoire `SAE_serpiuto` contenu dans l'archive

- le sujet en pdf,

- le fichier `EQUIPE` que vous devrez remplir pour donner un nom à votre groupe et indiquer les membres de votre projet,
- tous les fichiers sources,
- un répertoire `img` contenant les images utilisées par la partie graphique,
- un répertoire `plan` qui contient différentes cartes utilisées par le serveur de jeu.

Vos implémentations devront se faire dans le fichier `IA.py`. Si vous avez besoin pour structurer votre code de créer de nouveaux fichiers Python vous pouvez le faire.

Le jeu du tournoi repose sur un serveur de jeu qui effectue les actions suivantes :

- créer un jeu à partir d'une carte qui définit les murs et les couloirs de l'arène,
- attendre l'enregistrement de 4 joueurs,
- lancer la partie.

Au cours de la partie, le serveur

- envoie à chaque joueur, l'état du jeu,
- attend les ordres envoyés par chaque joueur,
- applique les ordres de chaque joueur en tirant au sort l'ordre de passage.

Chaque joueur est un programme client qui va se connecter au serveur. Un exemple de programme client vous est fourni dans le fichier `IA.py`. Toute la partie connexion et communication avec le serveur vous est fournie, vous *n'aurez qu'à* implémenter la fonction `mon_IA` pour déterminer les ordres à envoyer au serveur.

Lancement d'une partie

Pour lancer une partie, il faut ouvrir au moins deux fenêtres dans le répertoire du projet. Dans la première, vous lancez le serveur avec la commande `serveur.py`. Dans la seconde, vous allez lancer le client d'affichage puis quatre joueurs

```
1 python3 affichage.py&
2 python3 IA.py --equipe joueur1&
3 python3 IA.py --equipe joueur2&
4 python3 IA.py --equipe joueur3&
5 python3 IA.py --equipe joueur4&
```

Enfin dans la fenêtre du serveur vous pouvez appuyer sur les touches **S** puis **Entrée** pour lancer la partie.

5 Objectifs de la SAE

Comme indiqué précédemment, l'objectif de la SAE est d'aller au-delà du joueur aléatoire et de construire une IA qui prenne des décisions en fonction de l'état de la partie à un instant donné.

1er rendu : Implémentation de l'API `serpent.py` et recopie d'une arène

L'objectif de ce premier rendu est de vous faire explorer le code existant afin de vous l'approprier. Les spécifications de l'API `serpent.py` sont données directement dans le fichier. Les fonctions de recopie à implémenter se trouvent à la fin des fichiers `case.py`, `serpent.py`, `arene.py` et `partie.py`. L'avantage d'être capable de recopier une partie est de pouvoir tester l'évolution du jeu si tel joueur joue dans telle direction.

Pour cette partie, je devez rendre dans une archive `.zip` du répertoire `SAE_serpiuto` contenant les fichiers indiqués complétés ainsi que le fichier `EQUIPE`

2eme rendu : Définition de l'IA et son implémentation

Afin d'implémenter votre IA vous allez devoir établir une stratégie qui va reposer sur une observation de l'état de l'arène. Il va donc falloir définir un certain nombre de fonctions qui vont donner des informations vous permettant de prendre des décisions. Une bonne partie de ces fonctions vont reposer sur la connaissance des distances entre un point de l'arène et les autres cases de cette arène (voir TP12).

Pour cette partie, votre rendu est une archive `.zip` du répertoire `SAE_serpiuto` contenant les fichiers fournis avec le sujet complétés par

1. un rapport décrivant
 - la stratégie de votre IA,
 - les principaux algorithmes que vous aurez implémentés. Pour chaque algorithme, il faudra donner une approximation de sa complexité,
 - un état de ce que vous avez réussi à faire et de ce qui ne marche pas.
2. l'implémentation de votre IA et des fonctions d'observation dans le fichier `IA.py`. Chaque fonction devra être documentée par une docstring.

Une partie de la notation portera sur la clarté de vos implémentations et de la documentation.



Conseil

Commencez par mettre en place une stratégie simple puis enrichissez la au fur et à mesure. Faites bien des sauvegardes à chaque fois que vous avez une version qui fonctionne.

6 Rappel des rendus

La SAE va se dérouler en trois grandes étapes :

1. **14 janvier 9h** : Rendu de l'API serpent et des fonctions de copie de l'arène
2. **19 janvier 8h** : Rendu du répertoire implementation avec une IA fonctionnelle. Ce répertoire devra contenir votre rapport contenant les informations indiquées section 5
3. **19 janvier** : Devoir sur table où vous serez interrogés sur les apprentissages critiques de cette SAE
4. **19 janvier après midi** : Tournoi des IA !