



## 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 17 janvier 8h** sur l'implémentation des fonctions des API demandées,
- un rendu le **vendredi 19 janvier 8h** sur l'IA que vous aurez développée,
- une soutenance le **vendredi 19 janvier matin** au cours de laquelle vous présenterez votre travail.




La SAÉ se terminera par un tournoi le **vendredi 19 janvier** après midi où vos IA s'affronteront.




# 2 Présentation du sujet

Pacman IUT'O est un jeu librement inspiré du jeu d'arcade Pacman dans lequel chaque joueur contrôle un pacman chargé de récupérer des objets et un fantôme chargé d'attaquer les autres pacmans. Ce jeu avait initialement conçu par Lionel Martin pour le concours des 24h des IUT Informatique organisé à Orléans en 2013. L'objectif de cette SAE est de concevoir une IA qui gère les déplacements du pacman et du fantôme d'une équipe.

Le plateau de jeu est un labyrinthe comportant des couloirs et des murs infranchissables (sauf exception). Le plateau est un tore c'est-à-dire que lorsqu'un pacman ou un fantôme sort du plateau par la droite, il réapparaît sur la première colonne à gauche de la même ligne.

Sur le plateau peuvent apparaître différents objets qui rapportent des points et ont éventuellement un pouvoir.

- La vitamine  qui rapporte **3 points**.
- Le glouton  qui rapporte **50 points**. Cet objet permet au pacman qui l'obtient d'attaquer les fantômes adverses pendant **20 tours** de jeu.
- L'immobilité  qui rapporte **50 points** et permet d'immobiliser tous les fantômes qui se trouvent à moins de 5 cases du pacman qui l'obtient. Ce pouvoir dure **10 tours**.

- Le passemuraille  qui rapporte **50 points** et qui permet au pacman qui l'obtient de passer au travers des murs pendant **20 tours** de jeu.
- Les cerises  qui rapporte **100 points** mais n'a pas de pouvoir particulier.
- Le téléporteur  qui rapporte **50 points** et permet au pacman qui l'obtient de se téléporter sur une case vide du plateau choisie de manière aléatoire.

Lors d'un tour de jeu, chaque joueur choisit une direction (Nord, Sud, Est ou Ouest) pour son pacman et son fantôme. Le serveur choisit un ordre aléatoire des joueurs à chaque tour.

Pour chaque joueur,

1. il déplace d'abord le pacman. Si l'ordre donné par le joueur est incorrect ou n'est pas possible, le joueur perd un point. Si c'est le quatrième faux mouvement du pacman celui-ci sera téléporté aléatoirement sur la case vide du plateau. Si le pacman peut se déplacer, si la case d'arrivée contient un objet, celui sera attribué au pacman avec le pouvoir correspondant. Si un ou plusieurs fantômes se trouvent sur la case d'arrivée le joueur perd 20 points par fantôme. Les points perdus seront donnés aux joueurs à qui appartiennent les fantômes.
2. Ensuite le serveur déplace le fantôme du joueur. Comme pour le pacman si l'ordre est incorrect ou le déplacement est impossible, le joueur est pénalisé et le fantôme est téléporté au bout de quatre faux mouvements. Si le déplacement est possible, le joueur à qui appartient le fantôme prend 20 points à chaque pacman qui se trouve sur la case d'arrivée.

Lors de la création de la partie, la durée (en nombre de tours) de celle-ci est fixée. Au début de la partie, les pacmans et les fantômes sont placés aléatoirement sur le plateau. A la fin de la partie les joueurs sont classés en fonction du nombre de points qu'ils ont accumulés. Attention, les joueurs peuvent se retrouver avec des points négatifs.

### 3 Principes du serveur de jeu

Voici la structure du répertoire `SAE_pacman_iuto` 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 et les fichiers de tests,
- un répertoire `images` contenant les images utilisées par la partie graphique,
- un répertoire `cartes` qui contient différentes cartes utilisées par le serveur de jeu,

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 du plateau,
- attendre l'enregistrement de 4 joueurs,
- lancer la partie.

Au cours de la partie, le serveur

- envoie à chaque joueur, l'état du plateau et des joueurs,
- 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 `client_joueur.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.

Etant donné qu'un joueur doit indiquer à chaque tour de jeu uniquement la direction de son pacman et de son fantôme, il est très facile de faire un joueur complètement aléatoire (voir le fichier `client_joueur.py`).

## 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 `python3 serveur.py` dans la première fenêtre. Dans la seconde, vous allez lancer le client d'affichage puis quatre joueurs

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

Enfin dans la fenêtre du serveur vous pouvez appuyer sur la touche **Entrée** pour lancer la partie.

## 4 Objectifs de la SAÉ

Comme indiqué précédemment, l'objectif de la SAÉ est d'aller au-delà du joueur aléatoire et de construire une IA qui prenne des décisions en fonction de l'état du plateau à un instant donné. Pour cela, il va falloir décoder ce que le serveur envoie car toutes les informations sont fournies sous forme de chaînes de caractères. Il faudra aussi stocker ces informations dans des structures de données adéquates. Ensuite il faudra écrire des fonctions qui vous permettent de connaître l'état du jeu et prendre des décisions.

Ne perdez pas de vue qu'au-delà de l'efficacité de votre IA, la SAÉ sert à évaluer les apprentissages critiques rappelés section 1.

### 1er rendu : Implémentation des API

Pour la première partie (décodage et stockage de l'information), il vous est demandé d'implémenter les API suivantes :

- `joueur.py` qui gère les informations liées à un joueur donné (son nom, son nombre de points, ses objets etc.).
- `case.py` qui gère une case du plateau avec les objets et joueurs qui sont dessus.
- `plateau.py` qui gère un tableau en deux dimensions de cases.

Notez bien que certaines fonctions de ces API ne sont pas strictement indispensables à la création de l'IA mais peuvent malgré tout vous aider à comprendre la structure de données.

Pour implémenter ces API vous pouvez choisir la représentation que vous souhaitez (mais les dictionnaires sont fortement conseillés). Des jeux de tests vous sont fournis afin que vous puissiez vérifier que vos fonctions sont correctes. La spécification des fonctions est indiquées dans les fichiers Python. Cette partie sera notée en fonction des résultats des tests.

Des détails sur le codage des informations vous sont donnés section 5

### Affichage graphique

Le script `affichage_test.py` fourni avec le sujet vous permet d'avoir un retour visuel du contenu de vos structures. Pour configurer ce script, il faut modifier les lignes tout à la fin du script où plusieurs plans de plateau sont proposés.

## 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 du plateau. 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 du plateau et les autres cases de ce plateau (voir TP12). Voici un certain nombre d'informations dont vous pourriez avoir besoin pour votre IA.



### Quelques idées de fonctions d'observation

En fonction de la position de mon pacman sur le plateau :

- Quelle direction prendre pour atteindre l'objet le plus proche ?
- Quelle direction prendre pour m'éloigner du fantôme le plus proche ?

En fonction de la position de mon fantôme sur le plateau, quelle direction prendre pour m'approcher d'un pacman adverse.

Evidemment ce ne sont que des exemples mais vous pouvez imaginer toute sorte de stratégies qui auront besoin d'autres informations.

Pour cette partie, votre rendu sera composé

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. 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.

## 5 Codage des informations

Comme indiqué précédemment, le serveur de jeu envoie à chaque tour de jeu l'état du plateau et des joueurs sous la forme d'une chaîne de caractères. Cette section indique le codage utilisé pour décrire ces informations.

Dans le jeu, une couleur est attribuée à chaque joueur. Ces couleurs sont représentées par des lettres de l'alphabet en partant de 'A'. Les lettres majuscules représentent les pacmans et les lettres minuscules représentent les fantômes. Les objets sont représentés par les caractères suivants :

- ' ' aucun objet
- '.' une vitamine
- '\$' un glouton
- '@' une immobilité
- '~' un passe-muraille
- '&' une fraise
- '!' une téléportation

Les directions sont exprimées par un caractère : 'E' pour Est, 'S' pour Sud, 'O' pour ouest, 'N' pour Nord.

## 5.1 Le plateau

Pour vous indiquer l'état du plateau, le serveur de jeu enverra un texte de la forme de celui qui est à la gauche de la figure 1. La représentation graphique de ce plateau est donnée sur la partie droite de la figure 1.

```
10;10
#####
..##.....
# #####!###
#.      ##
#.#### ###
.##...@..
#.#.#.###
#.#.#.###
#&....~ ##
#####
2
A;1;1
B;1;8
2
a;8;2
b;3;5
```

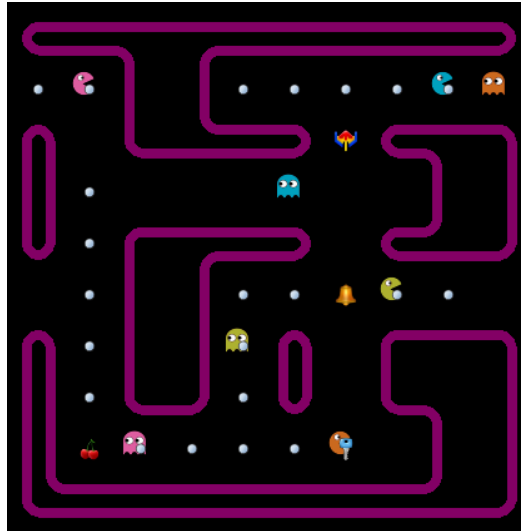


FIGURE 1 – Le codage d'un plateau et sa représentation graphique

La première ligne du texte indique le nombre de lignes et le nombre de colonnes du plateau. Les lignes suivantes représentent le plateau avec la convention suivante :

- # indique que cette case est un mur,
- un des caractères . \$ @ & ! indique un objet.
- un espace indique que cette case est vide.

Pour compléter les informations, la liste des pacmans et leur position ainsi que la liste des fantomes et leur position sont fournies après la description du plateau. Ainsi le 2, indique qu'il y a deux pacmans. Le premier a la couleur A et se situe en position (1,1). Le second est de couleur B et se situe sur la case (1,8). Pour les fantômes le codage est le même.

Dans l'API, les couleurs des joueurs sont représentées par la lettre majuscule correspondant à cette couleur et les fantômes seront représentés par la lettre minuscule de la couleur du joueur.

## 5.2 Les joueurs et leurs caractéristiques

Dans le codage du plateau, seule les positions des joueurs et des objets sont indiquées. Afin de donner une information complète, le serveur de jeu vous enverra les informations des joueurs sous la forme d'un texte comme ci-dessous :

```
A;253;0;1;1;8;2;0;5;0;Ghostbuster
B;145;1;1;8;3;3;0;0;Greedy
```

Chaque ligne décrit un joueur. Les informations sont séparées par des ; et sont données dans l'ordre suivant :

- couleur
- nombre de points
- nombre de faux mouvement en cours
- ligne et colonne du pacman
- ligne et colonne du fantôme
- nombre de tours restant pour le bonus glouton
- nombre de tours restant pour le bonus immobilité
- nombre de tours restant pour le bonus passe-muraille
- Nom du joueur

### 5.3 Les caractéristiques générales du jeu

Ces informations sont données afin de permettre aux IA de connaître des informations qui peuvent être importantes dans la prise de décision. Elles sont données sous la forme d'une ligne dont les informations sont séparées par des ; dans l'ordre suivant :

`duree_ecoul;duree_tot`

`duree_ecoul` est la durée écoulée depuis le début de la partie, `duree_totale` est la durée totale de la partie (la partie s'arrête quand `duree_act` est égale à `duree_totale`).

## 6 Rappel des rendus



### Important

Pour chaque rendu, vous rendrez une archive `.zip` de votre répertoire `SAE_pacman_iuto` sans changer son nom.

La SAÉ va se dérouler en trois grandes étapes :

1. **17 janvier 9h** : Rendu des API joueur, case et plateau
2. **19 janvier 8h** : Rendu du répertoire implementation avec une IA fonctionnelle (même si elle est presque complètement aléatoire). Ce répertoire devra contenir votre rapport contenant les informations indiquées section 4
3. **19 janvier** : Soutenance (15 min) au cours de laquelle vous présenterez votre travail et votre organisation
4. **19 janvier après midi** : Tournoi des IA !