

# IN104 : Projet informatique PACMAN

Emma Lagracie, Laura Le Meur

## 1. Objectifs

L'objectif donné au début de séance était très clair : coder un jeu d'arcade très populaire, le pacman. Un flou autour des moyens à employer a volontairement été fait, ce qui nous a obligées à nous fixer un environnement de travail.

Nous avons décidé de :

- utiliser le langage python, connu et plus simple pour nous deux
- Utiliser l'interface graphique pygame

Après avoir choisi cet environnement de travail, nous avons écrit un cahier des charges :

- Créer un labyrinthe ressemblant au jeu initial pour que pacman puisse s'y déplacer, et y mettre des gommes blanches et des fruits
- gérer le déplacement dudit pacman dans ce labyrinthe
- Comprendre l'interaction clavier écran sous pygame pour contrôler le pacman
- Attribuer un score au pacman qui augmente dès qu'il mange une gomme, un fantôme ou un fruit
- Faire mourir pacman si un fantôme le touche
- faire gagner pacman s'il a mangé toutes les gommes du labyrinthe
- Créer 4 fantômes si possibles ayant des comportements différents (certains plus agressifs que d'autres) qui en temps normal chassent pacman
- Coder un mode bleu pour les fantômes, sous lequel ils sont vulnérables et fuient le pacman

Ainsi, nous avons une idée générale du code à réaliser. Avant de se lancer, nous avons réalisé une timeline du projet pour ne pas prendre de retard :

- Avant le 2 avril : avoir réalisé le labyrinthe (avec un graphisme définitif)
- Avant le 16 avril : avoir codé le déplacement du pacman dans le labyrinthe grâce aux quatre flèches du clavier
- Avant le 23 avril : avoir établi le score du pacman, ainsi que sa capacité à manger des gommes blanches
- Avant les vacances d'avril : réussir à coder le déplacement des fantômes, si possible tous différents

- Avant le 21 mai : réaliser tout ce qu'il reste à faire, c'est-à-dire le game over, le mode bleu des fantômes, leur retour à la case départ s'ils se font manger par pacman...

## 2. Réalisation

### A. Les différentes classes

- La classe Niveau :

Notre labyrinthe étant en fait représenté par des cases (vide, gomme, coin gauche haut, coin droite haut, coin gauche bas, coin droit bas, ligne horizontale, ligne verticale), la classe niveau permet notamment d'avoir en argument un fichier texte à partir duquel on construit le labyrinthe dans un tableau.



- La classe Pacman
- Ses attributs

La classe pacman contient notamment en argument les 4 images de pacman (dans les différents sens).



Pour afficher les différentes directions dans lesquelles pacman peut se tourner quand il se déplace, la classe possède un attribut direction : ainsi, s'il se déplace vers la droite, `pacman.direction` vaudra 'droite' et il s'affichera à l'écran la première image.

Pacman possède également des attributs `x` et `y`, ses coordonnées réelles ainsi que `case_x` et `case_y`, ses coordonnées en termes de cases dans le labyrinthe. Ceci permet dans un premier temps de se déplacer dans le tableau représentant le labyrinthe, puis d'adapter aux coordonnées réelles pour l'affichage.

Enfin, elle possède également un attribut `score` dont le rôle est évident.

## - Fonctions associées

Deux fonctions font évaluer le score de pacman : eatFruit (+score du fruit) et eatGum (+1 si petite gomme, +5 si grosse gomme).

La plus grosse fonction de cette partie est celle du déplacement du pacman. Elle fonctionne grâce à l'attribut direction, qui peut prendre 5 valeurs : 'droite', 'gauche', 'bas', 'haut', 'standard'. Si direction vaut 'standard', alors on continue dans la direction dans laquelle pacman se trouve (cela correspond au fait qu'on a appuyé sur aucune flèche, ou alors qu'il y a un mur dans la direction où on souhaite aller). Sinon, on a un changement de direction du pacman (c'est à dire qu'on a appuyé sur une des flèches et qu'il est possible de se déplacer dans la direction indiquée).

- La classe des Fantômes
- Les attributs

Chaque fantôme possède une image (de couleur différente car chaque fantôme est différent) :



Blinky



Clyde



Inky



Pinky

Globalement, pour les situer dans le labyrinthe, nous avons choisi la même méthode que pour pacman : attributs x, y, case\_x, case\_y.

Ensuite, la classe a pour attribut une image 'standard' (attribut image) pour quand le fantôme est dans son état normal, et une image bleue (attribut imagebleue) qui apparaît à l'écran dès qu'il est vulnérable.

Le fantôme doit pouvoir se mettre dans deux états différents, un standard et un vulnérable, ceci est rendu possible grâce à l'attribut decomppte\_bleu, qui est à 0 pour l'état standard et positif en état vulnérable, associé au mode bleu.

Enfin, comme les fantômes se déplacent en fonction de pacman, un attribut perso leur est associé, qui permet d'accéder aux attributs du pacman.

## - Les déplacements des Fantômes

Les fantômes se déplacent vers pacman de manière plus ou moins intelligente. Ici sont explicitées grosso modo les stratégies de chaque fantôme.

Pinky a le déplacement le plus basique. Il se déplace toujours de la même manière, sans prendre en compte la position du pacman. Ainsi, il ira toujours à droite d'abord si c'est possible, sinon en haut, sinon en bas, sinon à gauche.

Inky quand à lui, se déplace de manière aléatoire. Les différentes directions ont 25 % de chance de se réaliser.

Blinky a une stratégie un peu plus élaborée. En effet, il calcule la distance en ordonnée et en abscisse de pacman et se déplace dans la direction où cette distance est la plus grande. Il aura donc un comportement de suiveur (toujours après le pacman).

Quand à Clyde, c'est le fantôme le plus redoutable. En effet à chaque tour de boucle, on calcule le plus court chemin à pacman et il emprunte la direction de ce plus court chemin. Ainsi, il est capable, à la différence de Blinky, de devancer pacman.

## B. Le programme principal

Le programme principal appelé `pacman.py` utilise les classes créées dans le fichier classe dans une boucle qui continue tant que l'utilisateur n'a pas perdu ou ne choisit pas de quitter le jeu. Ceci s'effectue à l'aide d'une variable 'continuer' qui porte la valeur 1 en standard et 0 si le jeu doit s'arrêter.

Après l'affichage de l'accueil, le jeu pacman en tant que tel se lance, à nouveau avec une boucle tant que 'continuer\_jeu' est à 1.

## C. Les tests

A chaque nouvelle fonction écrite, nous testions son bon fonctionnement en faisant tourner le programme et en mettant en commentaire les fonctions annexes à la partie testée. Nous avons donc tout d'abord testé l'affichage du labyrinthe, les déplacements de pacman puis les déplacements des fantômes et enfin le mode de fuite des fantômes. Nous affichions aussi des variables intermédiaires pour tester le bon fonctionnement de certaines de nos boucles ou fonctions.

## 3. Déroulement du projet, organisation et difficultés

### A. Répartition du travail

Au début, Emma s'est beaucoup occupée des interfaces graphiques (créer les cases du labyrinthe sur paint, le graphisme du pacman et des fantômes ainsi que l'écran d'accueil).

Laura s'est quant à elle concentrée dès le début sur le déplacement du pacman dans le labyrinthe ainsi que sa capacité à manger des gommes blanches, puis de la classe des fantômes

Le déplacement des fantômes a été réparti entre nous deux. Laura s'est occupée de Blinky et Clyde, tandis qu'Emma s'occupait d'Inky et de Pinky.

Enfin, Emma a pris en charge le « mode bleu » des fantômes ainsi que la victoire de pacman, tandis que Laura codait le game over et la fuite des fantômes en « mode bleu ».

## B. Difficultés et leur résolution

La première difficulté a été de respecter la timeline, en effet, la création du labyrinthe et coder les premiers déplacements de pacman ont été plus longs que prévu, puisque nous avons changé totalement notre choix d’affichage du plateau au bout de deux semaines pour plus de praticité, de fluidité et d’esthétisme.

Ensuite, certains déplacements de fantômes ont été complexes, notamment le déplacement de Clyde, qui parcourt le plus court chemin à pacman. Nous avons choisi après réflexion de réaliser un parcours en largeur du labyrinthe pour cet algorithme de plus court chemin, avec succès.

## ***Conclusion***

Nous retirons principalement de ce projet l’apprentissage de la gestion d’un travail sur le long terme et en groupe. Il faut en effet bien déterminer en amont les exigences, se fixer un planning et une répartition des tâches, ainsi que constamment communiquer sur les avancées du travail de son partenaire afin d’éviter de perdre du temps et de l’efficacité.

Nous avons aussi compris qu’avec l’accord de ses collègues, il ne faut pas hésiter à remettre en question le cahier des charges et les choix initiaux, comme nous l’avons fait pour le labyrinthe, dont nous avons changé le format et l’affichage après avoir réalisé qu’ils n’étaient pas pertinents.