

---

## TP 6 : Pac-Man

---

### 1 Présentation du jeu

Pac-Man est un jeu vidéo, dit d'arcade, qui consiste à déplacer un personnage (Pac-Man) à l'intérieur d'un labyrinthe, afin de lui faire manger des gommes qui s'y trouvent, en évitant d'être touché par un fantôme. Le labyrinthe est constitué d'une grille dont certaines cases sont des murs. Au début du jeu, Pac-Man dispose d'un certain nombre de vies.<sup>1</sup>

#### a. Règles de base du jeu

- Aucun des objets du jeu (Pac-Man, fruits, fantômes) ne peut se trouver sur une case occupée par un mur.
- Pac-Man et les fantômes peuvent se déplacer, alors que les fruits sont immobiles. Les fantômes se déplacent de façon aléatoire.
- Plusieurs fantômes peuvent se trouver sur la même case.
- Au début du jeu, Pac-Man dispose d'un certain nombre de vies. Quand Pac-Man et un fantôme se retrouvent sur la même case, Pac-Man perd une vie.
- Quand Pac-Man et un fruit se retrouvent sur la même case, Pac-Man mange le fruit, le fruit disparaît et le joueur gagne des points correspondant à la valeur du fruit.
- Quand Pac-Man a mangé tous les fruits, il passe au niveau supérieur, conserve ses vies et ses points, et le plateau de jeu est réinitialisé.
- Quand Pac-Man n'a plus de vies ou qu'il a terminé le dernier niveau, le jeu est terminé.

#### b. Règles complémentaires

- Certains fruits, appelés super-fruits, permettent, quand Pac-Man les mange, de lui donner temporairement le pouvoir de manger les fantômes. La durée de ce pouvoir dépend du niveau dans lequel il se trouve. Pendant le temps que dure ce pouvoir les éléments du jeu (fantômes et Pac-Man) changent d'aspect.
- Quand un fantôme est mangé, il se déplace vers la zone centrale, où il peut se régénérer et redevenir "normal".
- À chaque niveau correspond une configuration différente du labyrinthe, ainsi que de nouveaux types de fruits, une accélération du jeu, ...
- Les fantômes ont chacun des caractéristiques différentes : couleurs, formes, comportements, ...

#### c. Points

Différents types de fruits peuvent apparaître aux différents niveaux du jeu. Quand Pac-Man mange un fruit ou un fantôme, il gagne des points :

- Fruit simple : 10 points
- Fruit bonus : selon le type du fruit<sup>1</sup>.

---

<sup>1</sup>cf <https://fr.wikipedia.org/wiki/Pac-Man>

## 2 Développement du jeu

Les classes implémentant ce jeu se décomposent en trois parties, regroupées dans trois *packages* :

- a. *data* pour l'accès aux données du jeu
- b. *logic* pour la mise en œuvre des règles du jeu
- c. *view* pour l'affichage du jeu

Ces trois parties sont organisées en couches, chaque couche s'appuyant sur la couche inférieure (*c* s'appuie sur *b*, qui s'appuie sur *a*), et uniquement sur celle-ci. À chaque niveau, les services offerts par la couche *i* à la couche *i + 1* sont définis par une interface : la couche *i* implémente cette interface et la couche *i + 1* utilise cette interface.

### a. Accès aux données du jeu (*data*)

Les données initiales du jeu sont stockées dans des fichiers (un fichier par niveau de jeu), et chargées au lancement de chaque niveau de jeu. Pour chaque niveau, on y trouve :

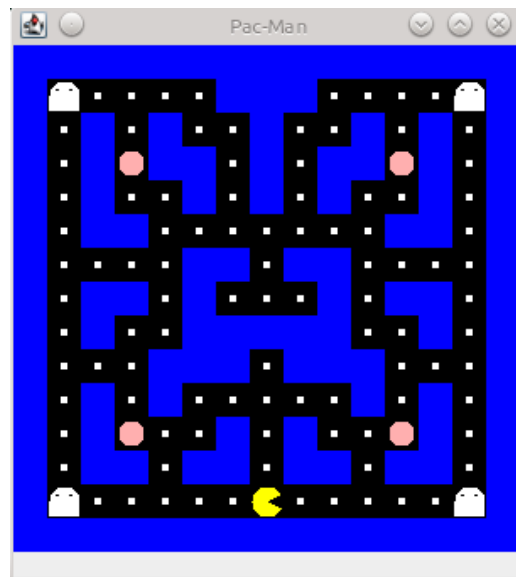
- plateau de jeu (emplacement des murs et des fruits)
- valeurs des fruits selon leur type
- emplacements des éléments du jeu (pacman et fantômes)
- vitesse du jeu
- ...

### b. Mise en œuvre des règles du jeu (*logic*)

Dans cette partie sont définies et implémentées les classes permettant de décrire et mettre en oeuvre les règles du jeu. Elles permettent de contrôler les règles du jeu, ainsi que les déplacements des éléments du jeu (fantômes). Elles permettent également de compter les points. Cette partie repose sur le package *data* pour l'accès aux données (chargement des données initiales).

### c. Affichage du jeu (*view*)

Le jeu est représenté par une interface graphique. Les mouvements de Pac-Man sont commandés par le joueur, via les touches de direction de son clavier (UP, DOWN, RIGHT, LEFT). Chaque élément du jeu (Pac-Man, fantômes, fruits, super-fruits) a une apparence qui dépend de son type (vous pouvez vous inspirer de l'image ci-dessus). Les classes de ce package utilisent celles du package *logic* – par exemple pour connaître l'état des différents éléments du jeu.



### 3 Déroulement du projet

Le projet se déroulera en 3 étapes:

1. Vous devez vous mettre d'accord sur les services offerts par les couches 1 et 2 aux couches 2 et 3, en réfléchissant bien au rôle de chacune de ces couches, et en définissant le plus précisément possible les contrats de ces couches sous forme d'interfaces bien documentées. De plus, vous devrez écrire une implémentation de chacune de ces deux interfaces fournissant des résultats par défaut à chacune des opérations (*mocks*).
2. Vous devez vous répartir les couches à développer et chacun des membres de l'équipe doit développer sa partie, en respectant bien les contrats définis dans la 1<sup>ère</sup> étape. Chaque classe doit être bien testée.
3. Vous devez intégrer les trois parties et bien tester l'application après l'intégration de ces trois couches.

### 4 À rendre

À chaque étape, vous devrez rendre une archive ZIP (à l'exclusion de tout autre format d'archive), contenant :

- Pour la 1<sup>ère</sup> étape, la documentation html générée par l'outil javadoc à partir des deux interfaces des couches 1 et 2 : cette documentation doit contenir – entre autres – un fichier `index.html` qui doit se trouver à la racine du fichier ZIP.
- Pour la 2<sup>ème</sup> étape, chacun doit fournir les fichiers sources du package qu'il a développé (classes Java bien documentées), et, si besoin, les fichiers de données du jeu à charger au démarrage
- Pour la 3<sup>ème</sup> étape :
  - les trois packages développés (dans les répertoires correspondants), avec les fichiers sources et les classes compilées (fichiers `.class`). Ces trois packages peuvent se trouver ou non dans une archive *jar*.
  - les fichiers contenant les données du jeu à charger au démarrage.
  - un fichier README donnant la commande à exécuter pour lancer le programme (assurez vous que, après avoir dézippé l'archive, le programme s'exécute correctement si on lance cette commande depuis le répertoire où se trouve le fichier README).