

Projet de Software Evolution : JPac-Man  
Année Académique 2015-2016  
Groupe 6

{BELLAFKIH HAYAT, M'HUIGUEN ASSIA, NONO TATOU P.G.}

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>JPacMan</b>	<b>3</b>
2.1	Origine . . . . .	3
2.2	Interface graphique . . . . .	3
<b>3</b>	<b>Fonctionnalités à développer</b>	<b>5</b>
<b>4</b>	<b>Fonctionnalisé(s) effectivement développée(s)</b>	<b>6</b>
4.1	Modification du choix d'une direction par les fantômes . . . .	6
4.2	Modification du comportement de base des fantômes . . . .	6
<b>5</b>	<b>Version finale</b>	<b>8</b>

# 1 Introduction

Ce document présente la démarche suivie pour mener à bien le projet JPacMan lié au cours de Software Evolution. Le but du projet a été de développer une nouvelle version de JPacMan afin de mettre en pratique l'enseignement dispensé. Ainsi, la section qui suit donne plus d'informations sur le projet. Après la section sur le projet, la prochaine portera sur un rappel des fonctionnalités à développer. La suivante quant à elle, détaillera les fonctionnalités effectivement implémentées. Et la dernière section portera sur la version finale avec toutes les extensions de la section précédente intégrées.

## 2 JPacMan

### 2.1 Origine

JPacMan est la version Java du jeu arcade Pacman créé en 1980<sup>1</sup>. Il a été développé à l'université de technologie de Delft au Pays-Bas par Arie Van Deursen<sup>2</sup> pour des cours d'ingénierie logiciel. L'objectif étant d'amener les étudiants à se familiariser à des concepts comme Git, Maven et JUnit.

### 2.2 Interface graphique

La figure 1 présente l'écran de jeu initial obtenu lorsque le jeu est lancé. Ce résultat est celui du fork du dépôt <https://github.com/SERG-Delft/jpacman-framework>. Comme on peut le voir, l'interface initiale du jeu est simple. Elle est composée de trois panneaux juxtaposés les uns sur les autres avec en tête un panneau entre autres pour les scores des joueurs, suivi d'un panneau pour le labyrinthe du jeu, et enfin un panneau de commandes.

---

1. <https://fr.wikipedia.org/wiki/Pac-Man>  
2. <http://www.st.ewi.tudelft.nl/~arie/>

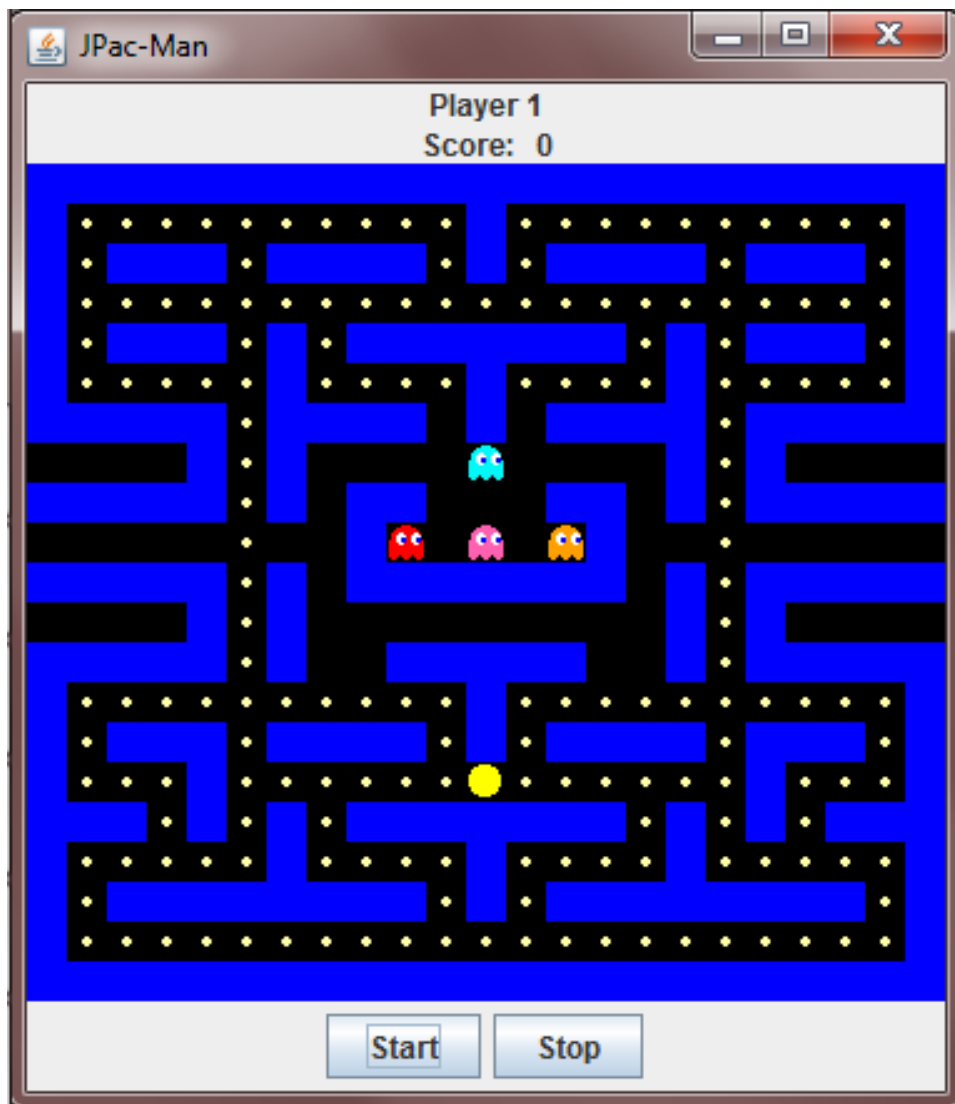


FIGURE 1 – Écran de jeu initial.

Comme le montre la figure 2, de haut en bas, le premier panneau affiche également d'autres messages que le score des joueurs. Par exemple, en cas de jeu perdu, le jeu informe le joueur en faisant précéder son score par le message "*You died.*".

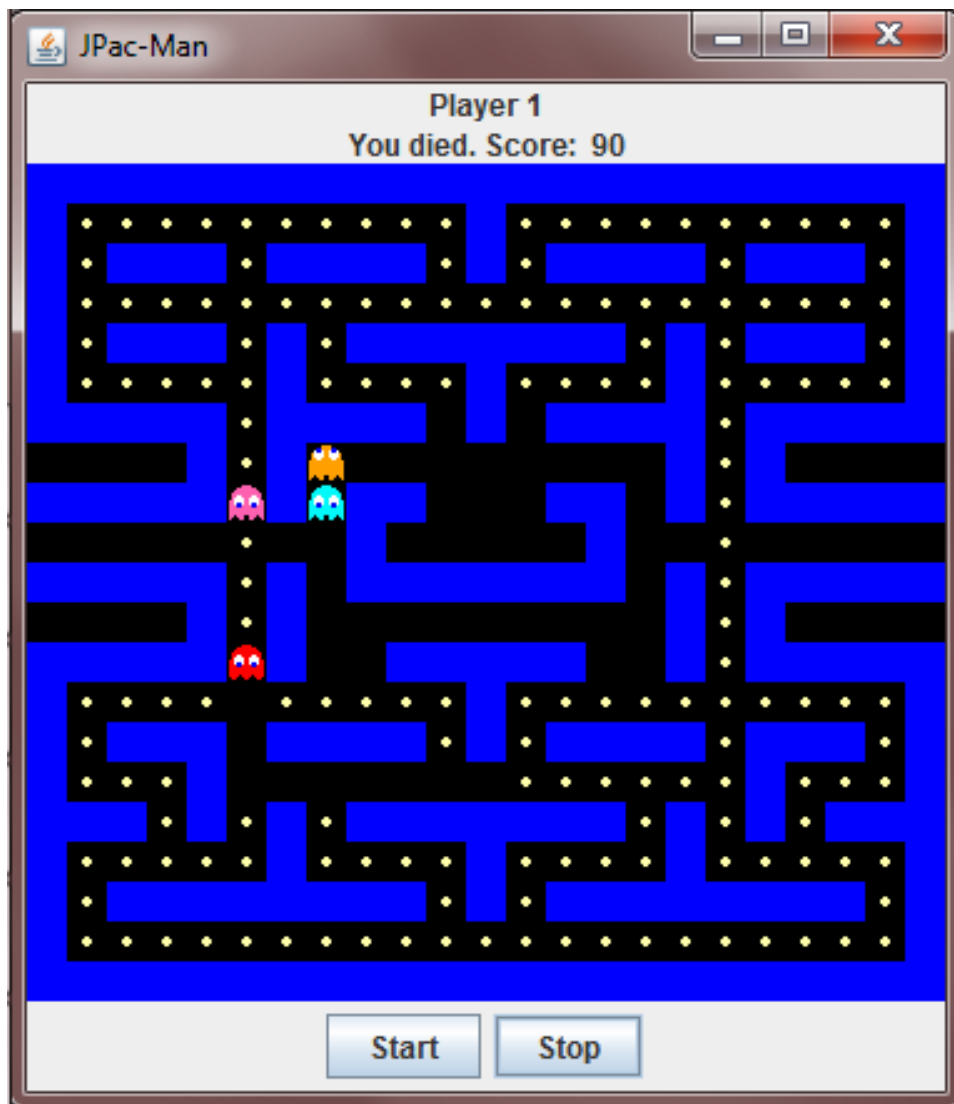


FIGURE 2 – Écran de jeu en cas de jeu perdu.

Le but du groupe, Groupe 6, va donc consister non seulement à enrichir cette interface mais surtout à étendre le fonctionnement de base du jeu selon les fonctionnalités assignées.

### 3 Fonctionnalités à développer

Le nombre de fonctionnalités distinctes à développer est de 3. Une fonctionnalité par membre du groupe. Ces extensions et les membres à charge des celles-ci sont :

- Multi-joueur sans Pacman par M'HUIGUEN Assia ;
- IA pour fantôme par NONO TATOU ;
- IA pour Pacman par BELLAFKIH Hayat.

Cependant, dans la version finale du jeu, seule une des trois fonctionnalités y a été intégrée.

## 4 Fonctionnalié(s) effectivement développée(s)

*IA pour les fantômes* est la seule fonctionnalité attendue présente dans cette version de JPacman. Pour cette fonctionnalité, il a été demandé d'une part de modifier le choix d'une direction par les fantômes selon un ordre donné et d'autre part, de modifier le comportement de base des fantômes.

### 4.1 Modification du choix d'une direction par les fantômes

Si un fantôme a le choix entre plusieurs directions, alors il choisira une direction suivant un ordre antihoraire. Autrement dit le haut sera prioritaire sur la gauche qui sera prioritaire sur le bas qui sera prioritaire sur la droite. Pour ce faire, la classe énumérative `Direction` a été modifiée en rangeant les différentes énumérations suivant l'ordre imposé.

### 4.2 Modification du comportement de base des fantômes

D'après l'énoncé, le comportement de base des fantôme est plutôt instable. Par comportement, il faut entendre déplacement. Il a donc été demandé que tous les fantômes aient deux modes déplacement qui pourront alterner suivant certaines contraintes temporelles. Ces modes de déplacement sont : **dispersion** et **poursuite**.

Pour y arriver, il a été recommandé d'utiliser le *Strategy Design Pattern*. C'est ainsi que l'interface `MoveStrategy` a été créée. Cette dernière contient l'unique méthode `nextMove()`.

#### Dispersion

En mode dispersion, chaque fantôme retourne chez soi. Les positions des maisons dans le labyrinthe sont :

- le coin supérieur droit pour Blinky(*fantôme rouge*) ;
- le coin supérieur gauche pour Pinky(*fantôme rose*) ;

- le coin inférieur droit pour Inky(*fantôme rose*);
- le coin inférieur gauche pour Clyde(*fantôme rose*).

Le fichier texte *ressources/board.txt* correspondant au labyrinthe du jeu a du être modifié par l'ajout du caractère 'H' aux positions correspondantes aux maisons des fantômes. Ensuite, dans la classe **Ghost**, la variable **home** a été créée de sorte que chaque fantôme puisse connaître l'emplacement de sa maison. L'accès à cette variable est rendu possible grâce aux méthodes **setHome(Square)** et **getHome()**. Dans la classe **MapParser**, la méthode **addSquare** a été modifiée de façon à stocker dans une liste les différentes positions des maisons de fantôme à la lecture du caractère 'H'. Et enfin, la méthode **assignHouseToGhost** de la classe **MapParser** se charge de répartir les maisons entre les différents fantômes.

Pour le déplacement en mode *dispersion*, la classe **DispersionMove** a été créée. Cette classe implémente l'interface **MoveStrategy** et a comme héritières les classes **BlinkyDispersion**, **PinkyDispersion**, **InkyDispersion** et **ClydeDispersion**. C'est dans chacune de ces dernières qu'a été défini le circuit à suivre par le fantôme une fois chez lui. Afin de savoir si un fantôme est arrivé à la position correspondant à sa maison, la variable **inMyCorner** a été ajoutée à la classe **Ghost**. Celle-ci est accessible grâce aux méthodes **inMyCorner(boolean)** et **isInMyCorner()**.

## Poursuite

En mode poursuite, l'unique but des fantôme est de tuer Pacman. Pour ce faire, ils doivent se trouver la même case que Pacman. Cependant, chaque fantôme a dans ce mode un comportement qui lui est spécifique. C'est ainsi que :

- Blinky se rendra, vitesse maximale, sur la case où se trouve Pacman ;
- Pinky, maître des embuscades, essaiera de deviner 4 mouvements plus tard où se trouvera Pacman et, ira se place sur cette case ;
- Inky, l'imprévisible, dont le comportement est la résultante de ceux de Blinky et Pacman ;
- Clyde se comportera comme Blinky si Pacman se trouve à 8 cases de lui. Sinon, il se rendra dans le coin inférieur gauche du labyrinthe.

Pour le déplacement en mode *poursuite*, la classe **PursuitMove** a été créée. Cette classe implémente l'interface **MoveStrategy** et a comme héritières les classes **BlinkyPursuit**, **PinkyPursuit**, **InkyPursuit** et **ClydePursuit**.

## Alternance de modes

Le changement de stratégie de déplacement est faite grâce à la méthode `changeMove` ajoutée à la classe `NPC` et utilisée dans la classe interne privée `NPCMoveTask` de la classe `Level`.

## 5 Version finale

La version finale du jeu JPac-Man sur le plan fonctionnel est uniquement étendue à la fonctionnalité présentée à la section précédente. Et afin de mieux le vérifier sans commencer à parcourir le code, l'interface graphique a également été refaite.

La figure 3 montre la premier écran qui apparaît lorsqu'on démarre le jeu. Cet écran permet au joueur de choisir un mode de jeu entre le mode *Solo* et le mode *I.A.*. Néanmoins, seul le mode *Solo* est disponible. Le mode *Solo* est le mode où Pacman est contrôlé par un joueur humain. L'objectif dans ce mode est de manger tous les "Pellets" du labyrinthe sans se faire tuer par un fantôme.



FIGURE 3 – Panneau de choix du mode jeu.

La figure 4 montre l'interface auquel s'attendre lorsque le mode *Solo* a été choisi. Comme on peut le voir, une barre de menu a été ajoutée. Au centre



sur le côté gauche se trouve un panneau permettant d'avoir des informations sur le fantôme. Une barre d'état a également ajoutée. Dans cette barre on retrouve des informations sur le temps de jeu écoulé ainsi que le score du ou des joueur(s).

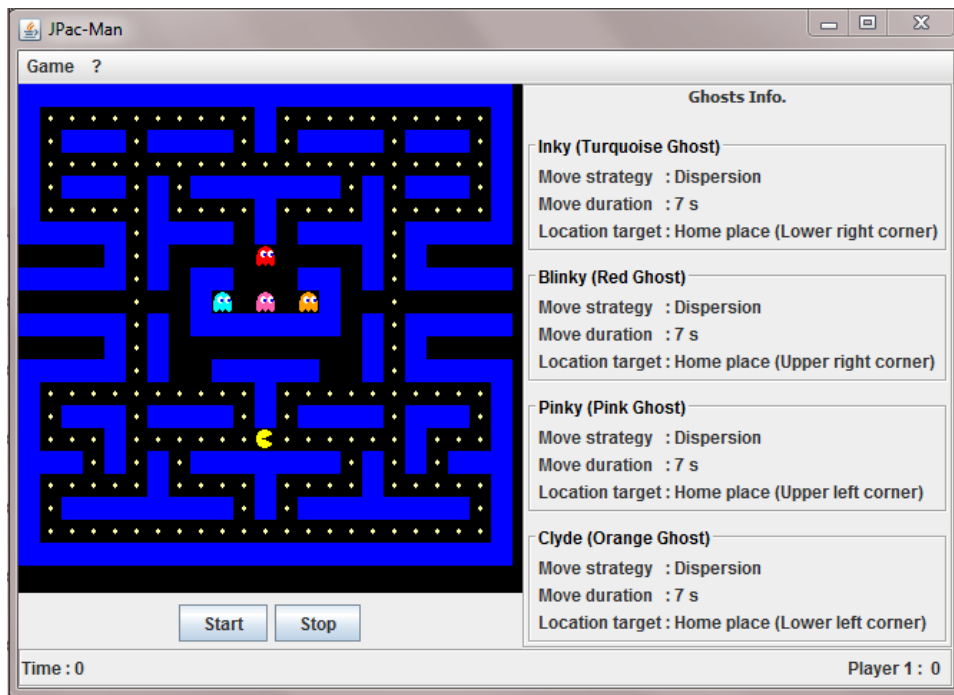


FIGURE 4 – Panneau de choix du mode jeu.