

# Rapport Projet LINGI1131 Groupe 103

Louis Arys  
Matthieu Meert

27 avril 2018

## Table des matières

<b>1</b>	<b>Préambule</b>	<b>1</b>
<b>2</b>	<b>Main controller :</b>	<b>1</b>
2.1	PacmanTurn : . . . . .	2
2.2	PacmanLive : . . . . .	2
2.3	GhostTurn : . . . . .	3
2.4	GhostLive : . . . . .	3
<b>3</b>	<b>Nos Pacmans :</b>	<b>3</b>
<b>4</b>	<b>Nos Ghosts :</b>	<b>3</b>
<b>5</b>	<b>Le test des Joueurs des autres groupes :</b>	<b>3</b>

## 1 Préambule

Avant de commencer le rapport de ce projet, nous tenions à préciser que celui-ci n'est pas entièrement fait : Pour des raisons personnelles et de travaux dans les autres cours, nous n'avons l'un comme l'autre pu travailler suffisamment sur ce projet. Nous retrouvant à court de temps pour le finaliser.

Ainsi, seulement la partie "turnbyturn" du projet est fonctionnelle.

## 2 Main controller :

Celui-ci marche de la manière suivante :

Premièrement, il initialise les différentes variables qui rentreront en jeu tout le long de la partie : les *Ports* des *Pacmans*, des *Ghosts*, les différentes *Cellules* qui serviront à maintenir des listes dynamiques permettant de connaître la position des *Pacmans* et *Ghosts* tout le long du jeu, ainsi que le nombre de joueurs morts, les points et les bonus qui sont en attentes

de réapparition. De plus, il transforme la Map mise dans le fichier **Input** de manière de la transformer d'une matrice à un ensemble de données utile pour la création de la carte et l'initialisation des différentes variables de partie. Une fois toutes les variables initialisées, le **Main controller** va alors créer une liste avec les *Ports* des différents *Pacmans* et *Ghosts* et la trier pseudo-aléatoirement

Deuxièmement, Une fois la première étape accomplie, le **Main controller** va lancer la fonction **GameManager** qui bouclera sur la liste des joueurs jusqu'à ce que tous les *Pacmans* soient mort. Cette fonction fonctionne en deux parties :

- en premier, elle va mettre à jour les temps de réapparition des points, bonus, joueurs décédés.
- en deuxième, elle va lancer le prochain tour de jeu durant lesquelles les joueurs effectueront les actions les uns après les autres

Comme dit ci-avant, **GameManager** lancera dans un deuxième temps la fonction **GameStartTurn**. Celle-ci prendra chaque joueur séparément, regardera si le joueur est un ghost ou un pacman et lancera la sous fonction appropriée dont nous parlerons dans les sous-sections ci-après.

## 2.1 PacmanTurn :

Cette sous-fonction a pour objectif d'effectuer toutes les actions possibles en fonction de l'état du pacman. Elle va d'abord tester si le pacman est vivant.

Si oui, elle lancera la fonction **PacmanLive** qui contient les différentes actions que *Pacman* peut effectuer de son vivant : à savoir : ramasser un bonus, ramasser un point, ou heurter un *Ghost*.

Si non, elle lancera la fonction **UpdateDead** si le temps de réapparition du *Pacman* est supérieur à 0. Dans le cas ou le temps est égal à 0, ce sera la fonction **PacmanRespawn** qui sera lancée.

## 2.2 PacmanLive :

Cette sous-fonction est la fonction permettant de récupérer le prochain mouvement du *Pacman* et de déterminer quelle sera la conséquence de celui-ci : d'abord, on scanne la position sur laquelle le *Pacman* se retrouvera après son mouvement à l'aide de la fonction **ScanPosition** si c'est un *Ghost* (ou plusieurs) qui se trouve sur la case, on regarde si le mode Hunt est activé ou non et on résoud la situation en conséquence. Cela lancera soit la fonction **PacmanIsKill** ou la fonction **PacmanKillGhost** (la fonction **PacmanIsKillRandom** ou la fonction **PacmanKillGhostPacmanLoop**).

Sinon, on regarde si ce n'est pas un bonus (ce qui lancera la fonction **PacmanGotBonus**) ou un point (ce qui lancera la fonction **GetPoint**).

La fonction vérifie aussi bien que le *Pacman* n'essaye pas de passer par un mur. Dans le cas échéant, une référence au mur sera afficher dans le Browser, et le tour du *Pacman* sera fini sans faire de mouvements.

### 2.3 GhostTurn :

Cette sous-fonction est très semblable à son homologue la sous-fonction **PacmanTurn**. Si le *Ghost* est en vie, elle lancera la fonction **GhostLive**, dans le cas contraire, elle va juste diminuer le temps de réapparition du ghost avec la fonction **UpdateDead**, ou faire réapparaître le *Ghost* (si son temps de réapparition est égal à 0) avec la fonction **GhostRespawn**.

### 2.4 GhostLive :

Cette sous-fonction est une version un peu simplifiée de son homologue la sous-fonction **PacmanLive**. Elle va scanner la prochaine position à l'aide de la fonction **ScanPosition**. Mais par contre, s'il n'y a pas de *Pacmans*, elle ne fera rien de plus, se contentant de se déplacer sans rien faire de plus. Par contre, si elle rencontre un *Pacman* (plusieurs), elle résoudra la situation avec les deux (ou les deux autres) mêmes fonctions que **PacmanLive** (à savoir **PacmanKillGhost** et **PacmanIsKill** (ou **PacmanKillGhostRandom** et **PacmanIsKillGhostLoop**)) mais dans l'ordre inverse que dans celui-ci.

## 3 Nos Pacmans :

Nous avons un pacman nommé Pacman103fred. Son comportement de base est de fuir les fantômes. Si il prend un bonus, il se dirige alors vers les fantômes.

## 4 Nos Ghosts :

Nous avons un fantôme nommé Ghost103vincent. Son comportement de base est d'aller vers les pacman.

## 5 Le test des Joueurs des autres groupes :

Nous les avons intégré mais nous n'avons pas eu le temps de les tester !