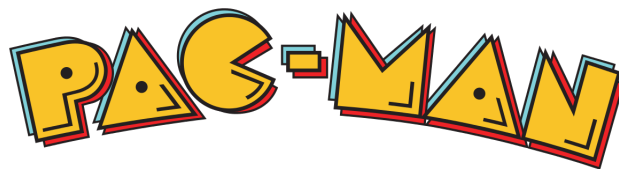


ENSTA PARISTECH

RAPPORT DU PROJET INFORMATIQUE IN104

Réalisation d'un jeu de Pac-Man



Khaoula BELAHSEN
Corentin GIGUET
Promo 2020

Encadrant : M. Stephen CREFF

Juin 2018

Introduction.

Contexte et objectifs.

L'objectif de ce projet était de réaliser un jeu de labyrinthe de type Pac-Man. La réalisation de ce jeu s'inscrit dans le cadre du projet d'IN104 du cursus de 1^{ÈRE} année de l'ENSTA ParisTech.

Pac-Man est un jeu vidéo d'arcade développé en 1980 au Japon. Le but de ce jeu est de faire ramasser au Pac-Man toutes les Pac-Gommes à l'intérieur du labyrinthe tout en évitant d'être touché par des fantômes. Le Pac-Man peut dévorer les fantômes pendant un temps limité après avoir ingéré des Pac-Gommes spéciales, les spéciales Pac-Gommes. Le joueur dispose d'un nombre limité de vies. Il peut augmenter son score en dévorant des bonus sous forme de fruits.

Organisation du travail.

La conception de ce jeu a été décomposée en différentes étapes :

1. Créer le labyrinthe ;
2. Prise en main de Pygame afin de faire des tests graphiques ;
3. Créer les personnages sous comportement aléatoire ;
4. Coder un Pac-Man dirigeable par l'utilisateur ;
5. Implémenter des fantômes autonomes et intelligents ;
6. Intégrer au jeu les éléments de Pac-Man comme les Pac-Gommes, les vies ou les bonus ;

A chaque étape, des tests sont effectués afin de visualiser correctement l'avancement du travail. Les principaux tests ont concerné la création du labyrinthe, l'implémentation du personnage (d'abord pour un comportement directionnel puis fluide et continu), celle des fantômes (un déplacement purement aléatoire puis logique), la superposition des deux éléments, la mise en place des Pac-Gommes... Lorsqu'un test unitaire était concluant, on ajoutait cette composante au programme principal. Au sein des tests unitaires, des tests plus simples, principalement des affichages de variables, permettaient de résoudre les différents problèmes rencontrés.

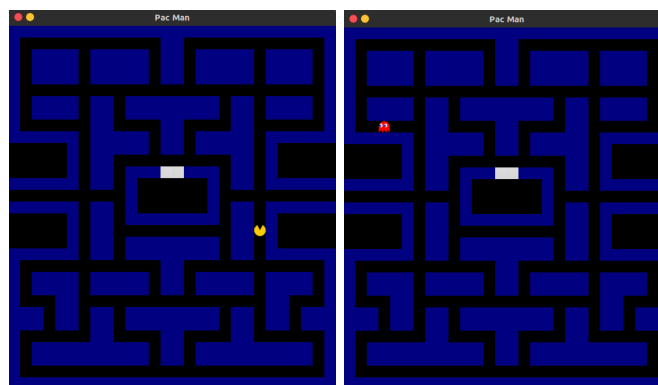


FIGURE 1 – Tests du bon fonctionnement du Pac-Man et d'un fantôme

Certains tests étaient simples à effectuer et à vérifier : la création du labyrinthe, le fonctionnement de Pac-Man... Pour ceux-là, il suffisait de lancer un programme à part, ne contenant que cette portion de code, et de vérifier si elle atteignait nos attentes. Par exemple, il était facile de

voir que le personnage s'arrêtait dans un couloir si une touche directionnelle était pressée, ou bien de remarquer que les fantômes avaient un comportement totalement indésirable. Cela permettait de facilement corriger le problème, ou du moins d'en trouver la source.

En ce qui concerne le comportement personnalisé des fantômes, il est beaucoup plus dur de l'observer. Avec un seul fantôme, on peut sans problème remarquer qu'il parvient à se diriger vers le personnage. En revanche, lorsque les comportements sont plus particuliers, il est plus ardu de trancher si le test correspond ou non à nos attentes.

Deux versions du projet sont données, elles diffèrent de par leur complexité :

Version basique :

- Même comportement des fantômes ;
- Pas de bonus ;
- Déplacement aléatoire des fantômes ;
- Interface graphique minimale ;
- Ajout des vies ;
- Départ des personnages aléatoire sur le terrain ;

Version poussée :

- Comportements différents pour les fantômes ;
- Rajouter les éléments bonus ;
- Mode chasse chez les fantômes ;
- Alternance entre les modes chasse et aléatoire ;
- Interface graphique poussée : mise en place du son et des images ;
- Départ des fantômes d'une zone protégée ;
- Retour à une case départ de Pac-Man en cas de mort ;

Que doit faire le jeu ?

FP1 : Créer une zone de jeu sous forme de labyrinthe.

- C1 : Faire un tableau $n*m$;
- C2 : Avoir une enceinte fermée ;
- C3 : Intégrer les éléments du jeu (Pac-Man, Pac-Gommes...) ;
- C4 : Zone protégée pour les fantômes ;

FP2 : Créer des personnages.

- C1 : Un Pac-Man ;
- C2 : Un nombre donné de fantômes ;
- C3 : Pac-Man est contrôlé par l'utilisateur ;
- C4 : Les fantômes sont dirigés par l'IA ;

FP3 : Intégrer les Pac-Gommes.

- C1 : Mettre une Pac-Gomme par case accessible ;
- C2 : Chaque Pac-Gomme délivre 10 points ;

FP4 : Intégrer les super Pac-Gommes.

- C1 : Avoir un nombre limité de super Pac-Gommes ;
- C2 : Activation si Pac-Man passe dessus ;
- C3 : Les fantômes deviennent inactifs pendant un temps T déterminé ;

FP5 : Intégrer les bonus.

- C1 : Il existe 5 types de bonus ;
- C2 : Rapportent un nombre de points spécifiques ;
- C3 : Rajouter des vies ;

FP6 : Comportement des fantômes.

- C1 : Départ d'une zone isolée ;
- C2 : Alternent une période de chasse et de Déplacement aléatoire ;
- C3 : Tentent de retourner dans leur zone s'ils sont attaqués par une super Pac-Gomme ;
- C4 : Chacun d'entre eux a un comportement différent ;

FP7 : Implémenter une interface graphique.

- C1 : Offre une notice de jeu ;
- C2 : Intégrer les couleurs du jeu ;
- C3 : Intégrer des sons ;
- C4 : Intégrer un compteur de points ;

FP 8 : Gérer les vies de Pac-Man.

C1 : Partir de trois vies ;

C2 : A chaque contact avec un fantôme, une vie est perdue ;

C3 : Tous les 1000 points, une vie supplémentaire est offerte ;

Architecture du programme.

Le programme est composé de différents sous-programmes :

- *game.py*
- les fichiers classes
- *constantes.py*

Les programmes les plus importants sont les fichiers *classes* : ils recensent les différentes classes utilisées, qui elles sont au nombre de cinq :

Niveau : permet de concevoir le niveau ainsi que le labyrinthe à partir d'un fichier texte modélisant le-dit labyrinthe sous forme de caractères ;

Perso : classe correspondant au personnage. Elle recense son fonctionnement et sa réaction aux instructions données par l'utilisateur ;

Ghost : correspond aux fantômes. Ses méthodes dictent aux fantômes comment se comporter à chaque intersection, coude et lorsque leur comportement est en mode chasse ou aléatoire ;

Gums : permet de mettre en place les Pac-Gommes ;

Bonus : permet d'intégrer les éléments bonus du jeu ;

Le fichier *constantes.py* quant à lui regroupe les constantes du jeu ainsi que les liens vers les différentes images du jeu.

Enfin, le fichier *game.py* est le cœur du jeu : c'est dans ce fichier que sont traitées les interactions de l'utilisateur avec la console, que sont conçues et actualisées les images, et que les événements sont analysés afin de décider des actions des personnages.

Détails des éléments du jeu.

Labyrinthe.

Afin de parvenir à correctement concevoir ce jeu, nous nous sommes librement inspiré d'un exemple offert dans une formation *Pygame* sur le site *OpenClassrooms* : un jeu de labyrinthe *Donkey Kong*. De manière plus technique, nous avons modélisé le terrain par un tableau 28x31 selon le modèle de jeu classique. Les labyrinthes de Donkey Kong et Pac-Man n'ayant que peu de différence, il a été simple et rapide de coder celui de notre version du jeu.

Les éléments du terrain ont été représentés dans nos programmes de la manière suivante :

0 : correspond à une voie libre, où les personnages peuvent circuler ;

m : correspond à un mur ;

x : correspond à une entrée de la zone réservée aux fantômes, eux seuls peuvent y accéder lorsqu'ils sortent au départ du jeu et lorsqu'ils sont attaqués par Pac-Man ;

r ou **l** : correspondent aux cases à droite et à gauche du terrain, elles sont reliées entre elles et permettent au Pac-Man de traverser le terrain. Elles sont inaccessibles aux fantômes ;

c : correspond à un coude dans un couloir ;

i : correspond à une intersection ;

Nous avons également introduit d'autres notations pour simplifier la mise en place du labyrinthe, mais elles ne sont pas significatives.

Pac-Man.

Le personnage de Pac-Man a été le plus simple à coder pour nous : il suit les instructions des touches pressées par l'utilisateur.

Au début, le personnage était dirigé en *touche par touche* : il suivait les instructions des touches pressées par l'utilisateur une par une. Dans ce mode, si l'utilisateur cesse de presser des touches, le personnage s'arrête dans sa course et reste immobile. Ce comportement est loin du comportement du réel Pac-Man. En effet, le déplacement doit être continu sur une direction tant qu'aucune autre (direction cohérente) n'est demandée. Cela a été difficile à implémenter. Dans une première version de ce mode, le personnage continuait bien dans une même direction mais s'arrêtait dès pression d'une touche, même si ce déplacement était impossible. Mais grâce à l'ajout d'un attribut supplémentaire à la classe *Perso*, nous avons pu résoudre ce problème ; et nous avons pu arriver à une version où si une direction impossible à prendre était commandée, elle était considérée à la prochaine intersection ou au prochain coude seulement.

Notre version du jeu comprend également un nombre fixé de vies associées au personnage. Il en possède trois au départ, puis en perd une à chaque fois qu'il rencontre un fantôme. Une fois les trois vies perdues, le jeu s'arrête et le joueur a perdu. Aussi, à chaque vie perdue, le personnage retourne à sa case départ. Notre version comprend également des vies supplémentaires qui peuvent être obtenues tous les 1000 points.

Pac-Gommes.

Notre version de Pac-Man comprend également des éléments dits Pac-Gommes (boules jaunes que Pac-Man doit manger) que nous avons mis sur chaque case du labyrinthe. Chacune PAC-

gomme permet de gagner un certain nombre de points, que le Pac-Man acquière en en ingérant. On considère que Pac-Man a mangé le Pac-Gomme si les deux se trouvent sur la même case. Et une fois toutes les Pac-Gommes consommées, le jeu s'arrête et le joueur a gagné. Afin de concevoir un élément Pac-Gomme, nous avons utilisé une matrice sœur de celle servant au labyrinthe, chaque case contenant une certaine valeur selon qu'elle est occupée par une Pac-Gomme ou non. et à chaque déplacement, cette matrice, ainsi que le nombre de points sont mis à jour.

Fantômes.

Les fantômes représentent les ennemis du Pac-Man. Ils ont été le gros challenge de notre travail sur ce projet vu qu'il a fallu les coder en agents intelligents. Dans un premier temps, nous avons calqué leur comportement sur celui de l'agent Pac-Man mais sans Interface Clavier. Les fantômes arrivaient donc à se mouvoir correctement dans le labyrinthe, en évitant les murs, mais c'était fait de façon aléatoire, et surtout sans reconnaître le Pac-Man à aucun moment. A chaque tour, selon les possibilités, une direction au hasard était choisie. Chose qui était fortement gênante dans un couloir, puisque le fantôme avait une chance sur deux de faire demi-tour.

On a donc introduit deux nouvelles notations pour les cases :

- i** pour les intersections ;
- c** pour les coudes ;

Avec ce changement, on est arrivé au comportement suivant :

Tant qu'un fantôme est dans un couloir, il continue dans la direction qu'il avait au tour précédent. S'il arrive sur un coude, il teste s'il peut aller dans une direction orthogonale, puis dans l'autre, puis se déplace. Enfin, à une intersection, il regarde dans quelles directions il peut aller, et ce selon la direction dont il vient : il ne peut pas faire demi-tour. Il se déplace ensuite dans l'une des directions possibles de manière aléatoire.

La phase de l'implémentation du comportement des fantômes a représenté l'étape longue et difficile de notre projet. Pendant un bon bout de temps un comportement apparemment incompréhensible vis-à-vis du code apparaissait et on ne voyait pas d'où ça venait. Néanmoins, en procédant à un débogage efficace, la cause a été trouvée et corrigée : la direction prise en mémoire par le fantôme était une variable globale du script père et une des méthodes de la classe ghost la modifiait sur place dans la méthode, sans que ce changement n'affecte la valeur de la variable globale.

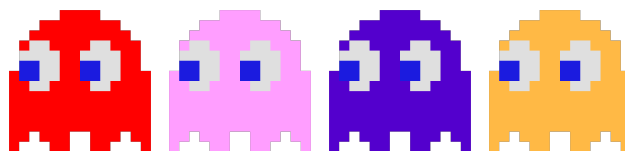


FIGURE 2 – Fantômes du jeu

Nos fantômes, désormais autonomes, doivent passer au niveau supérieur : l'attaque. Et pour cela, ils doivent reconnaître le Pac-Man, se figurer par où ils doivent passer pour l'atteindre, et le pourchasser pour le cerner. De nombreux algorithmes de résolution de labyrinthe existent déjà (A* ou de manière récursive), mais ils demandaient une certaine charge de travail pour les adapter proprement au jeu.

Nous avons donc préféré utiliser une méthode alternative, efficace au premier abord. Le comportement des fantômes avec cette méthode est le suivant : En mode chasse, chaque fantôme étudie la position du personnage (cela ne se fait qu'aux intersections, vu que c'est le seul endroit où ils peuvent changer de direction). Si le personnage se trouve par exemple dans le quart en haut à droite du fantôme, ce dernier pourrait le rejoindre en passant soit par la droite, soit par le haut. En comparant ceci avec les possibilités qu'offre l'intersection, on choisit la direction à prendre :

- unique s'il n'y a qu'une direction commune ;
- au hasard s'il y en a deux ou aucune ;

Cette méthode n'est pas optimale, non seulement parce qu'elle est très approximative, mais également parce qu'une partie des décisions est laissée au hasard. Une chose étonnante est que cette part laissée au hasard rend notre jeu des fois plus difficile que le jeu original. Cette méthode donne néanmoins de meilleurs résultats que le simple comportement aléatoire auquel on était arrivés au début, et elle repose bien sur des agents dotés d'une certaine part d'intelligence.

Dans le jeu initial, il y a quatre fantômes, avec quatre noms différents, quatre couleurs différentes, mais surtout quatre comportements différents. Les comportements les régissant sont les suivants :

- Blinky (rouge)** : Chasse directement le Pac-Man en le suivant comme son ombre ;
- Pinky (rose)** : Tente d'attraper Pac-Man par embuscade, il vise les positions futures du personnage ;
- Inky (bleu)** : Part de temps en temps dans la direction opposée de Pac-Man ;
- Clyde (orange)** : Prend parfois une direction totalement aléatoire ;

Comparaison avec le jeu original.

Un jeu étant un projet complexe à réaliser, certains choix de simplification ont dû être faits afin de le rendre réalisable à notre niveau, ou du moins afin de pouvoir obtenir une version du projet terminée et opérationnelle.

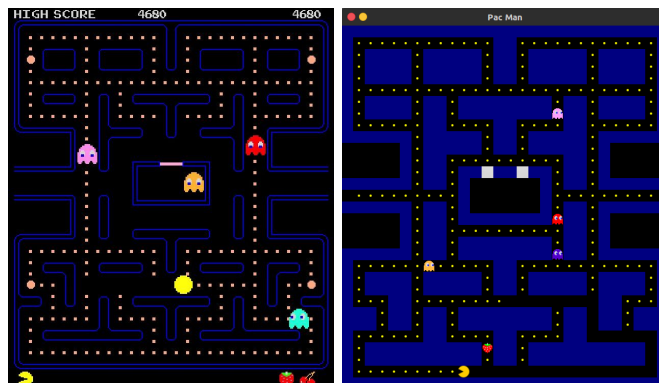


FIGURE 3 – Jeu original et notre version

Les principales différences pour la partie graphique concernent l’affichage des vies et des points, qui a lieu dans la console du terminal dans notre version, et à l’écran pour le jeu. Pour une question de pratique lors de l’implémentation, nous avons décalé la sortie du domaine des fantômes sous les intersections alors qu’elle sont au milieu de la porte de leur refuge dans le jeu initial, car ceci a facilité le codage de leur comportement suivant les éléments de matrice de notre labyrinthe.

Une autre grosse différence et simplification est liée au comportement des fantômes. Les nôtres ont un comportement beaucoup plus naïf que les fantômes du jeu original, comme il a été précisé plus haut. Également, comme les déplacements sont actualisés à chaque tour de boucle, on n’a pas pu intégrer une différence de vitesse entre les fantômes et le personnage. Enfin, nous n’avons pas inclus de super Pac-Gommes dans notre version, ces éléments permettent au Pac-Man de pourchasser les fantômes et d’engloutir ses ennemis éternels dans le jeu original. Pour cette raison, nous n’avons pas introduit le mode fuite des fantômes qui n’ont donc pas eu la possibilité de retourner dans leur zone de protection.

Aussi, parfois notre programme a du mal à remarquer que le personnage et les fantômes sont sur la même case, et donc des vies qui devraient être perdues ne le sont pas.

Plus subtilement, le décompte des points n’est pas tout à fait le même non plus. Comme les fantômes sont plus rapides que dans le jeu original, les points délivrés par les bonus dans notre jeu sont plus importants, surtout que ils sont moins variés. L’apparition de ces bonus est approximative par manque de littérature précise sur le sujet. Enfin, pour combler la difficulté supérieure de notre version à celle de la jeu original, nous avons décidé d’ajouter une fonctionnalité, qui est qu’une nouvelle vie est attribuée tous les 1000 points et non pas tous les 10000 points.

Cependant, quelques détails qui caractérisent le jeu ont pu être introduits afin de le rendre plus réaliste. Par exemple, les Pac-Gommes ont été ajoutées, ainsi que les bonus, au nombre de quatre dans notre jeu. Ça permet de rendre le jeu plus attrayant et plus intéressant qu’un simple labyrinthe. Nous avons également fait partir les fantômes de leur zone protégée et le Pac-Man d’une case départ pour mieux coller au jeu original. Nous avons aussi rajouté un laps de temps de pause à la mort de Pac-Man pour pas que l’utilisateur ne soit pris au dépourvu face à l’attaque

des fantômes lors du nouveau départ. Tous les personnages retournent également à leur point de départ lors de la mort du personnage. Les sons du jeu (le thème principal et le jingle lors de la mort du Pac-Man) ont aussi été introduits grâce aux fonctionnalités de *Pygame*.