



Programmation Java (Processing) - Projet - Pac-Man

Maxime MARIA

2025-2026

0 Consignes

- Projet à réaliser en seul.
- Vous devez rendre le projet avant le **dimanche 04/01/2026, 23h59** sur l'espace de dépôt dédié sur la plate-forme pédagogique. Tout retard entraînera une pénalité égale à $2^{\text{nombre_de_jours_de_retard}}$ (attention, ça va vite). Aucune excuse ne sera acceptée (« j'ai perdu mes fichiers », « mon ordinateur est tombé en panne », ...). Donc, sauvegardez régulièrement et sur plusieurs supports votre travail (encore mieux, utilisez Git)! Vous rendrez votre programme dans un dossier compressé `NOM.PRENOM.zip` (en remplaçant bien sûr `NOM` et `PRENOM` par votre nom de famille et votre prénom). Si vous ne respectez pas cette consigne, vous perdrez déjà un point.
- Vous ajouterez un rapport (format PDF) expliquant votre programme, les difficultés rencontrées, les solutions choisies... (Ce rapport est important dans la notation!). Celui-ci devra être nommé tel que `NOM.PRENOM_rapport.pdf`.

Attention au plagiat ! Nous avons à disposition des outils de détection de plagiat très performants ! Si nous détectons un quelconque plagiat, les conséquences peuvent être dramatiques pour la poursuite de vos études ! Donc évitez de prendre du code à un autre groupe ou de copier-coller du code d'internet...

Si votre projet est plus volumineux que la taille limite du dépôt, vous devrez utiliser le site <https://filesender.renater.fr/> pour uploader votre projet. Sur l'espace de dépôt, vous mettrez un fichier texte `NOM.PRENOM.txt` dans lequel vous copierez le lien de téléchargement. Attention, prévoyez un délai d'expiration du fichier d'au moins deux semaines ! (Encore une fois, aucune excuse ne sera acceptée).

1 Introduction

L'objectif de ce projet est de développer un jeu de type « Pac-Man » avec Processing. Voici un exemple de jeu : <https://freepacman.org/>.

La suite de ce document donne les spécifications nécessaires au développement. Comme vous le verrez, certains points sont précis et devront donc être respectés scrupuleusement, d'autres vous laissent totalement libres, et d'autres sont un peu ambigus... Et oui ! C'est souvent le problème quand on développe un programme pour un client. « Welcome to the real world ! ».

2 Règles du jeu

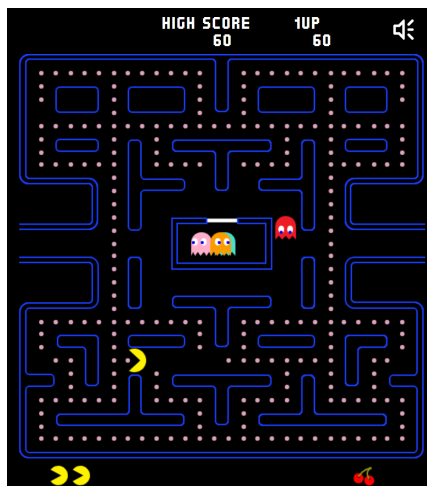
Vous incarnez un personnage (Pac-Man) dont le but est de manger toutes les billes (Pac-Gommes) à l'intérieur d'un labyrinthe, sans se faire toucher par l'un des quatre fantômes déambulant dans le labyrinthe. L'objectif est d'obtenir le plus de points possible.

Plateau de jeu

Le plateau de jeu est constitué d'éléments statiques :

- les murs du labyrinthe ;
- les Pac-Gommes classiques ;
- 4 Super-Pac-Gommes ;
- des bonus (représentés par des fruits dans le jeu initial),

et d'éléments dynamiques :



- Pac-Man, guidé par le joueur ;
- 4 fantômes se déplaçant dans le labyrinthe.

Début de partie

Au début d'une partie, le joueur a 2 vies et 0 point. La position initiale de Pac-Man et des différents éléments est fixée. Les quatre fantômes doivent être dans un espace réservé, duquel ils sortiront au fur et à mesure.

Déroulement d'une partie

Les déplacements de Pac-Man s'effectuent le long des murs du labyrinthe sans pouvoir les traverser. Le joueur contrôle sa direction. Aux bords du labyrinthe se trouvent des passages permettant à Pac-Man de passer d'un bord à l'autre (haut ↔ bas et/ou gauche ↔ droite).

Les fantômes déambulent dans le labyrinthe en essayant de toucher Pac-Man.

Lorsque Pac-Man passe sur une Pac-Gomme, celle-ci disparaît et le score du joueur est augmenté de 10 points. S'il s'agit d'une Super-Pac-Gomme, le score du joueur est augmenté de 50 points et les fantômes sont temporairement ralentis. Durant ce laps de temps, Pac-Man est capable de manger les fantômes : l'ambiance générale du jeu (les couleurs, *etc.*) doit changer de manière à ce que le joueur puisse se rendre compte facilement du changement. Lorsqu'un fantôme est mangé, il retourne immédiatement à sa position initiale et le score du joueur est augmenté de 200 points. Si plusieurs fantômes sont mangés pendant le même laps de temps, les points gagnés sont doublés (200, 400, 800, 1 600).

Lorsque Pac-Man passe sur un bonus, celui-ci disparaît et le score est augmenté de 500 points.

Si le score du joueur dépasse 10 000 points, il gagne une vie.

Si Pac-Man est touché par un fantôme, le joueur perd une vie. Pac-Man et les fantômes sont replacés immédiatement à leur position initiale.

N.B. : Les scores présentés ici sont à titre indicatif, il faudra sûrement les adapter en fonction de vos choix.

Fin de partie

La partie se termine lorsque toutes les Pac-Gommes ont été mangées ou lorsque le nombre de vies atteint 0.

3 Spécifications

Téléchargez le dossier `Architecture_projet` depuis la plate-forme. Il contient un ensemble de fichiers définissant la base de l'architecture de votre projet. Vous êtes **obligés** d'utiliser ces classes. Bien entendu, vous pouvez ajouter des attributs/méthodes aux classes existantes, modifier des types/valeurs, ainsi qu'ajouter de nouvelles classes !

Ce dossier contient :

- `pacman.pde` : programme principal qui contient les fonctions `setup()`, `draw()`, `keyPressed()` ... ;
- `game.pde` : classe gérant le jeu dans son ensemble ;
- `board.pde` : classe gérant le plateau de jeu ;
- `hero.pde` : classe gérant le héros (PacMan dans la version originale) ;
- `menu.pde` : classe gérant le menu ;
- `constants.pde` : constantes utiles dans le programme ;

- un dossier `data` : contient les données utiles pour le jeu (images, scores...);
- un dossier `levels` : contient les fichiers décrivant les niveaux.

Plateau de jeu

Le plateau de jeu est contenu dans la classe `Board`.

Il est représenté par un tableau 2D. Chaque case contient un unique élément dont le type est représenté grâce au type `enum TypeCell`.

Dans un premier temps, définissez le plateau « en dur », *i.e.* directement dans le code. Ensuite, le plateau devra être décrit par un fichier `.txt`. Un exemple vous est donné dans le fichier `levels/level1.txt`. La première ligne contient le titre du niveau. Les lignes suivantes contiennent la description du tableau avec :

- `x` : mur;
- `V` : vide;
- `o` : Pac-Gomme;
- `O` : Super-Pac-Gomme;
- `P` : Pac-Man.

Enfin, vous devrez avoir la possibilité de sauvegarder une partie en cours (via le menu).

Fenêtre de jeu

La fenêtre de jeu est constituée d'au moins deux éléments :

- le plateau de jeu;
- les informations relatives à la partie en cours (nombre de vies restantes, score actuel, ...)

Menu

Lorsque le joueur appuie sur `Echap`, le jeu est mis en pause et un menu « pop-up » s'affiche. À partir du menu, l'utilisateur doit pouvoir :

- recommencer la partie en cours;
- sauvegarder la partie en cours;
- charger une partie sauvegardée;
- consulter les meilleurs scores
- quitter le jeu.

Déplacement des personnages

Les déplacements de Pac-Man s'effectuent le long des murs du labyrinthe sans pouvoir les traverser. Le joueur peut contrôler sa direction à l'aide des touches Z-Q-S-D (Haut-Gauche-Bas-Droite) ainsi qu'avec les flèches. Vous devrez donc déterminer à tout moment si le déplacement demandé par l'utilisateur est possible (*i.e.* s'il n'y a pas de mur sur le passage). Ces changements de direction ne sont possibles qu'aux croisements du labyrinthe. À terme, vous pourrez mettre en place un système de « cache » permettant de garder en mémoire une demande de changement de direction tant que celle-ci n'est pas possible.

Les fantômes sortent de leur emplacement initial à tour de rôle avec un pas de temps fixé. Ils déambulent ensuite de manière aléatoire dans le labyrinthe, à vitesse constante. Ces changements aléatoires de direction ne sont possibles qu'aux croisements du labyrinthe. Une gestion particulière devra être mise en place pour sortir les fantômes de leur emplacement initial.

Fin de partie

Lorsque le jeu est terminé, si le score obtenu fait partie des cinq meilleurs scores, il doit être sauvegardé dans un fichier donné. Pensez à demander le nom du joueur !

Direction artistique

Vous êtes complètement libre sur le choix du thème de votre jeu. Vous n'êtes pas obligés de respecter le thème original de Pac-Man, laissez libre court à votre imagination ! Vous pouvez (devez) renommer les classes si vous changez de thème. Si votre Pac-Man est une chèvre (par exemple), il faut appeler la classe `pacgoat` et non `pacman`.

4 Extensions possibles

Une fois l'ensemble des fonctionnalités implémentées, testées et validées, vous pouvez améliorer votre programme en proposant par exemple (attention, certaines extensions sont beaucoup plus complexes que d'autres) :

- prise en compte du temps pour finir le niveau dans le calcul du score final ;
- comportement intelligent des fantômes (vous pouvez vous inspirer du jeu initial, *cf.* Annexe A) ;
- génération aléatoire de labyrinthe ;
- animation des personnages (par exemple, Pac-Man ouvre et ferme la bouche quand il avance) ;
- ajout d'effets visuels (brouillard, « traînée » de particules suivant les déplacements...) ;
- gestion des exceptions
- ajout de son...

5 Conseils

1. Avancez par étapes ! N'essayez pas de tout faire d'un coup ! Voici un exemple d'organisation :
 - (a) Coder la plateau de jeu, tester son affichage
 - (b) Ajouter les déplacements du personnage (case par case, par à-coups)
 - (c) Faire en sorte qu'il ramasse les Pac-Gommes
 - (d) Modifier le déplacement pour qu'il soit plus « lisse »
 - (e) Calculer le score
 - (f) Afficher le score
 - (g) Gérer les vies
 - (h) Ajouter les fantômes
 - (i) Ajouter le menu
 - (j) ...
2. Gardez chaque version (étape) du programme afin de pouvoir revenir facilement à une version fonctionnelle (Git est parfait pour cela).
3. Respectez les bonnes pratiques de programmation données en début d'année !
4. Ne passez pas les trois quarts du projet à modéliser de beaux objets pour remplacer Pac-Man ou les fantômes. Vous aurez une meilleure note si le programme fonctionne bien mais que les personnages sont de simples disques, qu'avec de belles images et un programme tout buggué...

A Comportement des fantômes

Dans le jeu initial, les 4 fantômes sont différents, ils ont un nom, une couleur et un comportement distincts :

1. Shadow (ombre) - Blinky (rouge) : suit Pac-Man en permanence ;
2. Speedy (rapide) - Pinky (rose) : vise l'endroit où se dirige Pac-Man ;
3. Bashful (timide) - Inky (bleu) : de temps en temps, part dans la direction opposées de Pac-Man ;
4. Pokey (bébête ?) - Clyde (orange) : de temps en temps, change de direction.

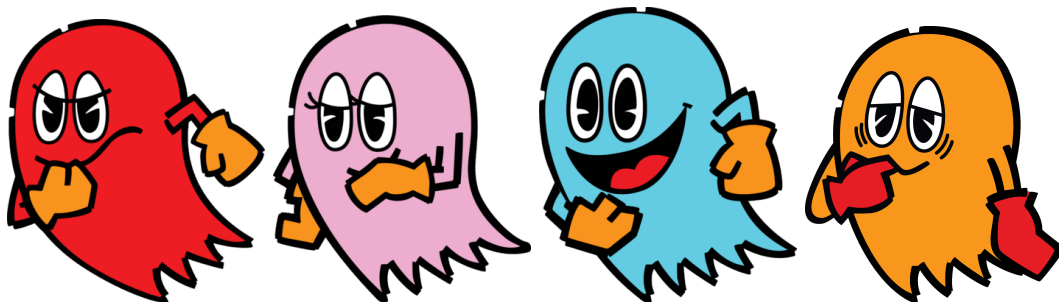


FIGURE 1 – Les quatre fantômes de Pac-Man

Lorsque Pac-Man a mangé une Super-Pac-Gomme, les fantômes fuient : ils sont plus lents et changent de direction aléatoirement. Pour gérer le comportement des fantômes, il sera nécessaire de considérer le plateau comme un graphe ! Inutile de changer sa représentation mais il faudra sûrement adapter les algorithmes de parcours de graphes (Dijkstra, A*) à cette structure de données particulière (ici une grille homogène).