
Le Jeu de Pac-Man

Le projet

Pac-Man est un personnage qui se déplace dans un labyrinthe. Il doit y manger des *pac-gommes* tout en évitant d'être mangé par des fantômes¹. Le but de ce projet est de mettre en œuvre un programme permettant à un utilisateur de jouer à *Pac-Man*.

Les règles

Notre *Pac-Man* suit des règles bien spécifiques que voici :

Le labyrinthe occupe une aire de jeu rectangulaire divisées en cases formant soit les murs, soit les emplacements sur lesquels se meuvent les personnages du jeu. *Pac-Man* est gouverné par l'utilisateur pour se déplacer dans le labyrinthe. Les fantômes, eux, sont gouvernés par le logiciel. Au départ, ils se trouvent dans une zone particulière du labyrinthe appelée la salle centrale.

Dans la situation normale, lorsque *Pac-Man* rencontre un fantôme (c'est-à-dire lorsque *Pac-Man* et le fantôme se trouvent sur une même case), le fantôme le mange. Cependant, durant certaines périodes, *Pac-Man* ne craint plus les fantômes mais au contraire est capable de les manger. Durant ces périodes, il devient *Super Pac-Man* et les fantômes le craignent.

Lorsqu'un fantôme mange *Pac-Man*, celui-ci perd une vie. Lorsque *Pac-Man* mange un fantôme, celui-ci est renvoyé dans la salle centrale et *Pac-Man* gagne 20 points.

De plus, sur les cases du labyrinthe où se déplacent *Pac-Man* et les fantômes, peuvent se trouver des aliments, soit des *pac-gommes*, soit des fruits, soit des champignons. Lorsque *Pac-Man* se déplace sur une telle case, il mange ce qui s'y trouve. Les fantômes y sont, par contre, insensibles.

Lorsque *Pac-Man* mange une *pac-gomme*, il gagne un point. Lorsqu'il mange un fruit, il devient *Super Pac-Man*. Manger un champignon permet de mémoriser la situation du jeu ce qui permettra d'y revenir dans certaines circonstances.

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

Itération 1

Vous réaliserez une application **console** qui permettra à l'utilisateur de jouer à une version très simple de *Pac-Man*.

En console, l'interactivité et la dynamique de l'application seront forcément simplifiées. Par exemple, les fantômes resteront immobiles. Le but de cette première itération est essentiellement de vous pousser à réaliser une analyse élaborée du **modèle** de l'application.

Analyse

Il est primordial que vous ne vous lanciez pas immédiatement dans le codage de votre application mais que vous consacriez le temps nécessaire une analyse préalable du **modèle**. En particulier votre analyse doit refléter l'énoncé général des règles du domaine et donner lieu à l'existence d'un nombre minimal de classes correspondant aux différents concepts du domaine (labyrinthe, personnages mobiles, aliments...).

Fonctionnalités

Pour cette première itération, la forme du labyrinthe est fixe. Cette forme ainsi que les positions de *Pac-Man* et des aliments sont définis sous la forme d'un tableau (bidimensionnel) d'entiers. Un fichier texte (`labyrinthe.txt`) vous est fourni. Vous pouvez copier/coller son contenu dans le code source de votre programme.

En rapport avec la section précédente de cet énoncé, il devrait être clair que le tableau d'entiers fourni n'est qu'une représentation primaire de l'état initial du labyrinthe et qu'en aucun cas il ne peut servir à représenter votre labyrinthe.

Au départ de l'application, 4 fantômes sont placés au hasard dans le labyrinthe (pas sur la case où se trouve *Pac-Man*). Ensuite, le labyrinthe et son contenu (*Pac-Man*, les fantômes et les aliments) sont dessinés à la console. D'autres informations sur l'état du jeu (voir la suite) sont affichés à chaque affichage du labyrinthe.

Seul *Pac-Man* peut bouger. Pour le déplacer, l'utilisateur enfonce, par exemple, l'une des touches **H**(aut), **B**(as), **G**(auche) et **D**(roite) (ou tout autre choix à votre meilleure convenance) suivie de la touche *Enter*. Le labyrinthe est alors re-dessiné avec le déplacement pris en compte. Si lors d'un déplacement, *Pac-Man* est passé sur une case contenant un aliment, l'aliment est mangé et disparaît de la case.

De plus, lorsque *Pac-Man* mange une *pac-gomme*, il gagne donc un point. Lorsqu'il mange un fruit, il devient *Super Pac-Man*. Pour cette itération, par contre, manger un champignon ne change rien.

Si, lors de ses déplacements, *Pac-Man* passe sur la même case qu'un fantôme (rappelons que pour cette itération les fantômes sont immobiles), *Pac-Man* perd une vie. Lorsque *Pac-Man* a perdu toutes ses vies, le jeu se termine. Pour cette itération, *Pac-man* possède, au départ, 2 vies.

Lorsque *Pac-Man* est devenu *Super Pac-Man*, il le reste jusqu'à la fin du jeu. Dans cet état, s'il rencontre un fantôme, il le mange et, pour cette itération, le fantôme disparaît du jeu. Le jeu se termine aussi si tous les fantômes ont disparu.

A chaque étape (chaque affichage), en plus de réafficher le labyrinthe, vous afficherez les informations suivantes concernant l'état du jeu:

- Le nombre de vie restant à *Pac-Man*.
- Le nombre de points de *Pac-Man*.
- Le nombre de fantômes dans le labyrinthe.
- Le nombre de pac-gommes restantes dans le labyrinthe.

Architecture

Cette application sera bien sûr correctement **architecturée** selon un modèle MVC (de «type» Modèle-Business / Vue-Présentation / Contrôleur-Logique applicative).

- Une couche de Présentation sera développée (la Vue). Dans cette itération, celle-ci sera implémentée sous forme de vue en console.
- Une couche Modèle exigera l'élaboration d'un **diagramme de classes du domaine** intégrant tous les concepts-métier liés à ce stade du jeu.
- Une couche liée à la logique applicative (le CTRL) proposera toutes les fonctionnalités utiles à la réalisation des fonctionnalités souhaitées.

Nous exigeons par ailleurs l'utilisation du design pattern **Observer / Observable**.

Rapport d'analyse–Conception–Développement

Comme chacune des futures itérations, celle-ci sera constituée d'un **rapport d'analyse** exposant les diagrammes *UML* nécessaires à la compréhension du développement réalisé.

Pour votre travail, vous utiliserez le logiciel de version *GIT*, plus particulièrement l'outil de repository *Bitbucket*. Nous créerons, pour chaque équipe, un repository et nous gérerons les accès à ceux-ci.

Agenda et échéances

Le rapport d'analyse ainsi que le code source de votre projet devront être déposés dans votre repository pour **le 9 janvier au plus tard**.