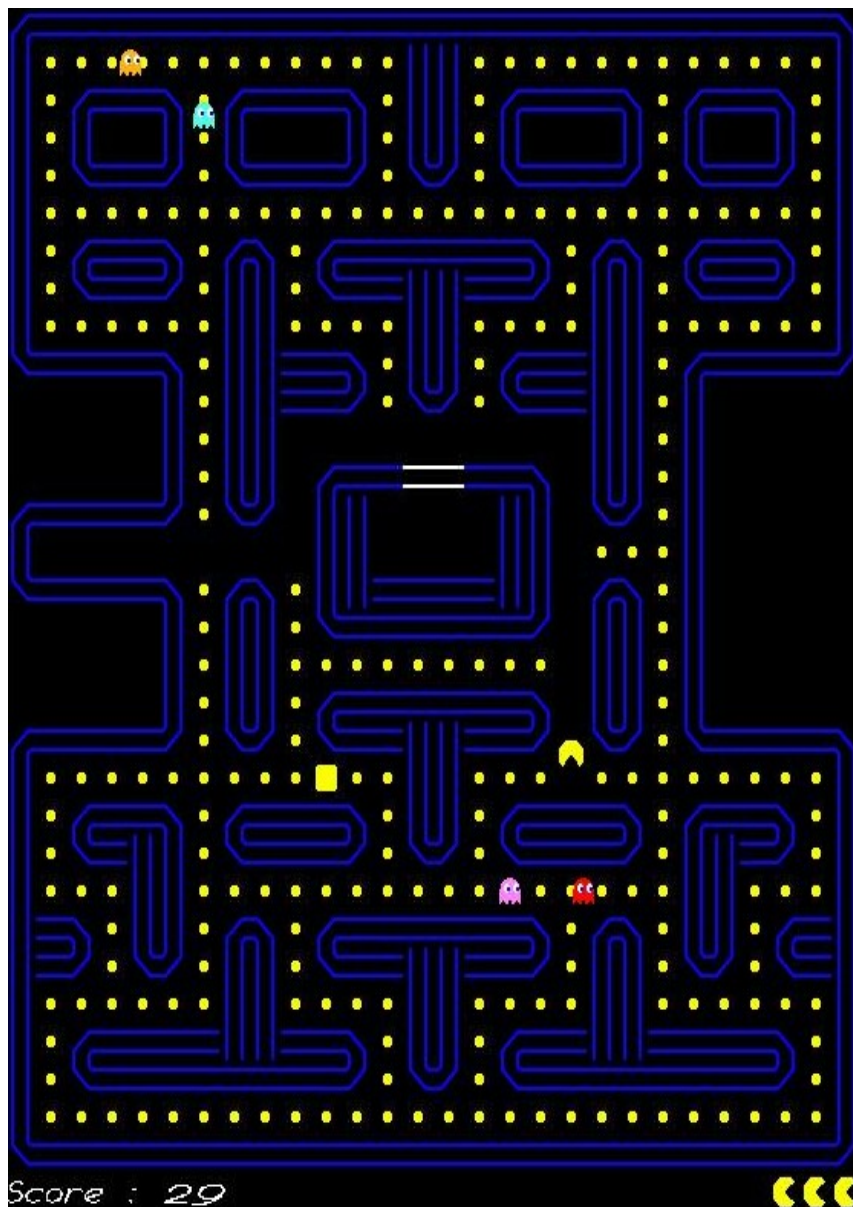


## Projet de C

### Réalisation d'un PacMan



PINCHINAT-LOTH Marin  
STERN Valentin  
SWITALA Yoann

## **I - Idée initiale**

La première intention de notre groupe était de réaliser un Pac-Man où le joueur devait coder le comportement de son Pac-Man et réaliser le meilleur score possible. La première étape consistait donc à recréer ce grand classique du jeu vidéo, et nous nous sommes rendu compte que cette tâche nous demandait déjà beaucoup de travail.

## **II - Choix du support de travail**

Valentin disposant d'un serveur personnel, nous avons choisi d'y héberger notre projet et d'utiliser Git comme gestionnaire de versions. Cependant malgré de nombreuses tentatives il nous fut impossible de récupérer le projet sur l'ordinateur de Marin (sous Windows). Il a donc participé au projet via l'ordinateur de Valentin.

## **III - Avancées du projet**

La première étape du projet a consisté à créer un niveau de test vide dans lequel on pouvait déplacer un point par le biais des touches directionnelles. Ceci fait il a fallu ensuite rendre ce déplacement continu, ce qui a constitué notre premier vrai défi. La solution a consisté à utiliser un mouvement pixel par pixel de l'objet à déplacer en l'empêchant de pouvoir modifier son déplacement tant que son précédent mouvement n'est pas terminé (tant qu'il n'a pas atteint la case suivante). Ainsi, chaque case mesure 20x20 pixels.

L'étape suivante a été la création de niveaux. La solution la plus simple que nous avons retenue est de créer ces niveaux dans des fichiers texte sous forme de 0 et de 1, chaque chiffre définissant une case (0 pour une case vide, 1 pour un mur). La taille du niveau à créer est spécifiée au début du fichier. Par la suite nous avons ajouté les points (symbolisés par .. des points) qui lorsqu'ils sont « mangés » incrémentent un compteur de score, et les positions de départ des fantômes (B, C , I et P pour Blinky, Clyde, Inky et Pinky).

Cependant, avant de s'occuper des fantômes nous avons cherché à texturer le Pac-Man, les murs et les points. Nous avons donc créé pixel par pixel les images dont nous avons besoin à l'aide du logiciel Gimp.

À ce stade le jeu était jouable, mais n'avait pas encore de fin et se fermait dès le premier contact avec un fantôme.

Une fois le jeu texturé et les fantômes créées, nous avons introduit les « super-points », plus gros, qui dans un premier temps donnent plus de score que les points normaux. Ces « super-points » sont destinés à permettre au Pac-Man de manger les fantômes.

Nous avons enchaîné en accordant des vies au joueur, le faisant réapparaître à son point de départ s'il rencontre un fantôme. Nous avons aussi ajouté l'affichage du score, dont le décompte s'est cependant révélé imparfait. Manger un fantôme et manger un Super-point accorde bien du score bonus, mais le score issu des points normaux était défaillait du fait d'un problème d'actualisation du niveau.

Il est ensuite venu le temps de coder l'intelligence artificielle des fantômes. Au premier, Clyde nous voulions donner un comportement aléatoire. En le testant, nous avons observé qu'il essayait parfois de se diriger dans des murs et restait donc immobile. Après avoir corrigé cette erreur Clyde se déplaçait bien aléatoirement mais en changeant frénétiquement de direction alors qu'il n'était censé le faire qu'aux intersections. Séduits par ce comportement amusant, nous avons pensé à le conserver. Cependant, ainsi codé Clyde avançait peu dans le niveau, et nous avons donc décidé de le faire aller tout droit jusqu'à entrer en collision avec un mur.

Pour coder l'intelligence artificielle des deux autres fantômes nous avons eu besoin d'élaborer un algorithme de recherche du plus court chemin. Malheureusement, malgré le fait que nous ayons compris comment un tel algorithme pouvait fonctionner, nous avons été incapable de l'écrire par manque de structures de données à notre disposition.

Après cet échec nous avons attribué la même intelligence artificielle à tous les fantômes : avancer tout droit jusqu'à rencontrer un mur, puis tourner aléatoirement. Par la même occasion nous avons doté chaque fantôme d'un « code d'initialisation » lui permettant de sortir de sa maison sans problèmes.

Ceci fait, nous avons corrigé le problème de comptage du score et ajouté la possibilité de choisir sa carte depuis la console. Dans le même élan, nous avons résolu des problèmes d'affichage de murs, ajouté un Easter Egg (niveau 42).

Les dernières heures de travail furent consacrées à l'affichage d'un menu de sélection de niveau ainsi qu'à empêcher les fantômes mangés de sortir de leur base afin d'empêcher la technique du « spawn-kill » qui gonflerait artificiellement les scores.

#### **IV – Options que nous n'avons pas pu ajouter**

Voici la liste des fonctionnalités auxquelles nous avons pensé mais que nous n'avons pas pu ajouter par manque de temps ou par manque de compétences :

- Un algorithme de recherche du plus court chemin à utiliser dans l'intelligence artificielle des fantômes.
- Une intelligence artificielle tournant aléatoirement à chaque intersection.
- Une génération aléatoire de niveau.
- Une sauvegarde du meilleur score.

#### **V – Résumé rapide**

Le jeu tel que nous le présentons est un Pac-Man fonctionnel disposant de plusieurs niveaux de tailles différentes à choisir via un menu de démarrage. Le joueur joue grâce aux flèches directionnelles. Il dispose de trois vies, peut voir son score évoluer en direct et peut manger temporairement les fantômes (ce qui augmente son score) grâce à des « Super-points ».