**I3333**

**Pacman**



Carlita Khawand 55028

Yorgo Wakim 54689

**Présenté à**

**Dr. Joseph Costantine**

Juin, 2020

Sommaire

[I. Introduction 1](#_Toc44268204)

[II. La Partie Théorique du Projet 2](#_Toc44268205)

[A. Le labyrinthe 2](#_Toc44268206)

[B. Pacman 3](#_Toc44268207)

[C. Monstres 3](#_Toc44268208)

[D. Pacgommes 4](#_Toc44268209)

[III. La Partie Pratique du Projet 5](#_Toc44268210)

[A. Le Labyrinthe 5](#_Toc44268211)

[B. Pacman 6](#_Toc44268212)

[C. Monstres 7](#_Toc44268213)

[D. Un jeu continue 7](#_Toc44268214)

[IV. Conclusion 7](#_Toc44268215)

[V. Liste des Figures 8](#_Toc44268216)

# Introduction

L’infographie est le domaine de la création d’images numériques assistée par ordinateur. Elle est principalement utilisée dans des applications développées à l’origine sur des technologies d’impression et de dessin traditionnelles. Les modèles graphique 2D peuvent combiner des modèles géométriques, des images digitales, des textes, des fonctions mathématiques et beaucoup plus. Ces composants peuvent être modifiés et manipulés à travers des transformations géométriques bidimensionnelles comme la translation, la rotation et le *scaling*.

En effet, il existe divers méthode d’implémenter cette part de l’informatique, dont Opengl (Open Graphics Library) qui est un ensemble normalisé de fonctions de calcul permettant de développer facilement des applications en 2D ou 3D. En d’autres termes, c’est une bibliothèque qui offre un très grand nombre de possibilités et peut être utilisée sur n’importe quel système d’exploitation qui possède un environnement graphique.

Pour mettre en contexte les différentes notions acquises en infographie (particulièrement sur Opengl), ce projet a traité le jeu d’arcade classique « Pacman » développé par Namco en 1980. Ce jeu consiste à déplacer Pacman, un personnage qui ressemble à un diagramme circulaire à l’intérieur d’un labyrinthe, afin de lui faire manger toutes les pacgommes qui s’y trouvent en évitant d’être touché par des fantômes. On estime que ce jeu traduira efficacement nos compétences en Opengl établies lors de ce cours.

En premier lieu, on parlera de la partie théorique de la conception de « Pacman », puis on discutera la partie pratique et la programmation de ce jeu pour enfin finaliser avec une conclusion.

# La Partie Théorique du Projet

## Le labyrinthe

Comme tous les évènements du jeu se déroulent à l’intérieur du labyrinthe, la solution idéale est d’utiliser une matrice 2D. La largeur du labyrinthe est le nombre de colonnes de la matrice, et sa longueur est le nombre de lignes. Afin de pouvoir dessiner les murs à l’intérieur et aux extrémités, quatre types de murs ont été définis. Le premier est « normal », qui se lie avec le point directement au-dessus et le point directement à sa droite (s’ils sont du type mur aussi). Le second, c’est le mur « horizontal », qui se lie uniquement avec le point mur à sa droite. Le troisième, c’est le mur « vertical », qui se lie uniquement avec le point mur en dessus. Le quatrième et dernier type, c’est le mur qui ne se lie ni en dessus ni à droite. Chacun de ces murs a un code (entier) spécifique, et ils sont enregistrés dans la matrice du labyrinthe.

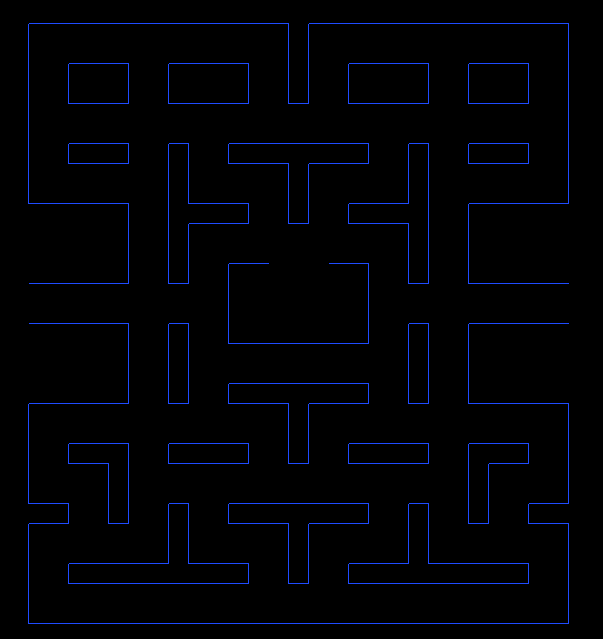


Figure 1 - Le labyrinthe

## Pacman

Le caractère Pacman est dessiné à l’aide d’un ensemble de triangles dont on calcul les sommets. Afin de laisser une ouverture, quelques sommets restent non calculés. Un petit point noir est dessiné par rapport au centre du cercle (le sommet commun de tous les triangles) afin d’avoir l’œil de Pacman.

Figure 3 - Pacman Fermé

Figure 2 - Pacman Ouvert

## Monstres

Les monstres, sont dessinés tout d’abord sous forme de cercle. Puis trois petites ellipses sont ajoutées pour les pieds et pour les yeux il y a quatre ellipses en total un blanc et un noir pour chaque œil.

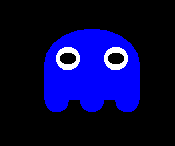


Figure 4 – Monstre

## Pacgommes

Les pacgommes (la nourriture de Pacman) sont de petits points blancs ajoutés chacun à sa position dans la matrice du labyrinthe.

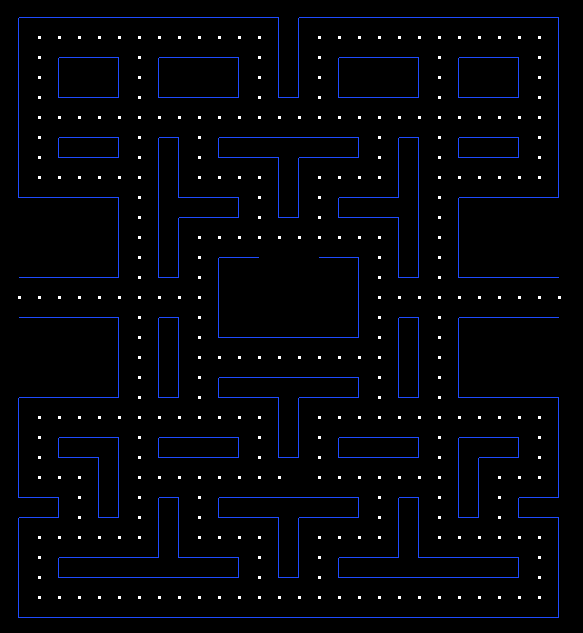


Figure 5 - Pacgommes

# La Partie Pratique du Projet

## Le Labyrinthe

Dans l’initialisation du labyrinthe, une matrice est définie. Les valeurs stockées dans cette matrice varient selon quel caractère se trouve à la position i, j. Ces valeurs sont enregistrées comme variables globales :

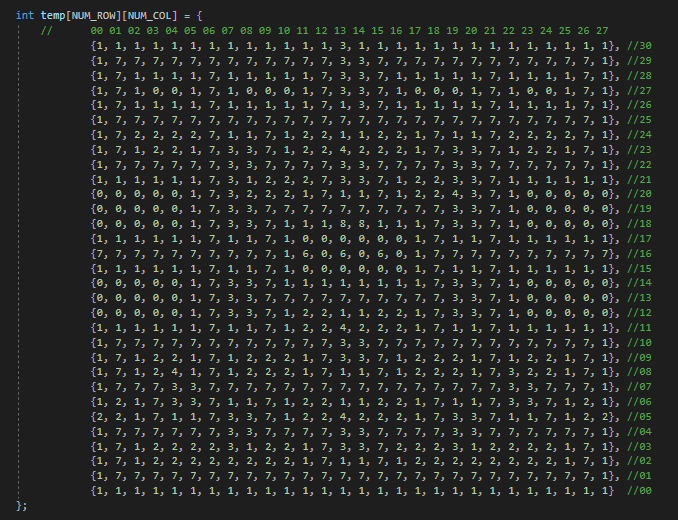
* Numéro d’une case vide : 0
* Numéro d’un mur ordinaire : 1
* Numéro d’un mur horizontal : 2
* Numéro d’un mur vertical : 3
* Numéro d’un mur point : 4
* Numéro de Pacman : 5
* Numéro du Monstre : 6
* Numéro des pacgommes : 7
* Numéro du portail : 8

Figure 6 - Matrice

Lors de l’affichage sur l’écran, un parcours de la matrice est effectué afin de vérifier sur chaque point quel caractère doit être affiché pour appeler la fonction d’affichage qui lui correspond.

Par rapport au coordonnés, le point initial (0,0) est le point de référence sur l’écran. Le labyrinthe possède les coordonnés de ce point et de la, les coordonnes de tous les points de la matrice peuvent être calculés suivant i et j.

Soit A(x, y) un point de la matrice, Calculons les coordonnes de A sur l’écran :

x = startingPoint.x + STEP\_SIZE \* j.

y = startingPoint.y + STEP\_SIZE \* i.

Avec, STEP\_SIZE c’est la taille fixe entre tous les objets de la matrice.

## Pacman

Pacman a deux fonctionnalités principales. Il a besoin de se déplacer afin de manger les pacgommes et échapper aux monstres et à chaque changement de direction, il doit changer son orientation.

Tout d’abord, à son état initial, Pacman se dirige vers la droite. Dans la fonction initPacman, on sauvegarde les sommets des triangles déjà calculés par rapport à l’origine (0,0) avec le nombre de ces sommets et ceci dans ses deux états (ouvert et fermé), et comme sa direction initial est la droite, il sera initialement orienté vers la droite. De ce fait, lors d’un changement de direction, on n’a plus besoin d’utiliser une matrice de rotation complexe ou de faire une certaine série de transformation. En effet, les points étant sauvegardés ayant comme centre l’origine, on n’a qu’à faire une simple rotation (inverse dans le cas présent) par rapport à l’origine.

Evidemment, dessiner Pacman à l’origine à chaque fois rend le jeu dénudé de sens ; pour cela, on effectue une simple translation de vecteur OX (X étant le centre désiré de Pacman) avant de l’afficher. C’est pourquoi la fonction drawPacman prend à chaque fois son nouveau centre et effectue une translation sur les sommets initiaux (déjà sauvegardés) puis le dessine dans sa nouvelle position.

## Monstres

Les monstres fonctionnent comme Pacman en ce qui concerne leur déplacement. Une direction aléatoire est générée et puis la fonction drawMonsters dessine chaque monstre à sa position par rapport aux coordonnés calculées.

Dans le cas où un monstre rencontre Pacman, l’utilisateur perd la partie et a la possibilité de rejouer. Ce cas est traduit par Pacman, et au moins un monstre, ayant les mêmes coordonnées dans la matrice.

## Un jeu continue

Un jeu comme celui-ci est souvent abordé avec l’approche multithreading mais cela, comme appris après quelque recherche, ne fonctionne pas très bien avec Opengl. On a donc opté pour une autre approche qui consiste à redessiner l’interface après un certain temps avec l’aide de la fonction glutTimerFunc ().

# Conclusion

En Conclusion, et comme on a pu le constaté lors de ce projet et tout au long de cette matière, Opengl est une matière première très utile dans le cas de la conception de n’importe quelle application graphique avec C++, et à ceci s’y applique aisément les jeux tel que Pacman.

En effet Cette librairie offre tous les outils dont on a eu besoin pour coder ce jeu et subvenir à n’importe quel problème qui a surgit lors de l’implémentation, que ce soit la création et représentation des divers éléments du jeu tant que pour l’interaction entre l’utilisateur et l’application.

# Liste des Figures

[Figure 1 - Le labyrinthe 2](file:///C:\Users\Charbel\Desktop\rapportPacmanProjet.docx#_Toc44268182)

[Figure 3 - Pacman Fermé 3](file:///C:\Users\Charbel\Desktop\rapportPacmanProjet.docx#_Toc44268183)

[Figure 2 - Pacman Ouvert 3](file:///C:\Users\Charbel\Desktop\rapportPacmanProjet.docx#_Toc44268184)

[Figure 4 – Monstre 3](#_Toc44268185)

[Figure 5 - Pacgommes 4](file:///C:\Users\Charbel\Desktop\rapportPacmanProjet.docx#_Toc44268186)

[Figure 6 - Matrice 5](file:///C:\Users\Charbel\Desktop\rapportPacmanProjet.docx#_Toc44268187)