Université Picardie Jules Verne - L3 Informatique 2015/2016

UE Projet

Jeu 2D – PACMAN

Nom : Andujar Stanislas

Suivi par : M Dieudonné

**Table des matières**

[Introduction](#page3) [3](#page3)

[ANALYSE DES OPTIONS DES MENUS](#page4) [4](#page4)

[Menu principal](#page6) [6](#page6)

[ANALYSE ECRAN JEU](#page7) [7](#page7)

[1Le Labyrinthe](#page7) [7](#page7)

[2Le Pac-man](#page8) [8](#page8)

[3 Les Fantômes](#page11) [11](#page11)

[ANALYSE ECRAN MENU PAUSE](#page12) [12](#page12)

[ANALYSE ECRAN FIN PARTIE](#page13) [13](#page13)

[ANALYSE ECRAN AJOUT PSEUDO](#page14) [14](#page14)

[ANALYSE ECRAN TABLEAU SCORE](#page15) [15](#page15)

[Conclusion](#page16) [16](#page16)

[Bibliographie](#page17) [17](#page17)

[ANNEXE](#page18) [18](#page18)

[1Algorithme déplacement du Pac-man :](#page18) [18](#page18)

[2Algorithme test collision :](#page19) [19](#page19)

**Introduction**

j'ai choisit finalement de réaliser un Pacman en 2D. C'est un jeu d'origine Japonaise jouable à l'époque sur une borne d'arcade. Le but pour moi étant d'adapter ce jeu sur un Ordinateur et de proposer au joueurs le même gameplay.

Le langage utilisé pour coder le jeu est JAVA. J'utilise ce langage puisque je l'ai déjà étudié durant mes précédente année de fac, d'autre part pour sa portabilité.

La bibliothèque utilisé est Slick2D. C'est une bibliothèque open-source, spécialisé dans le développement de jeu vidéo 2D, facile d'accès et fondé sur LWJGL(Lightweight Java Game Library) qui est une bibliothèque JAVA de bas niveau utilisant OpenGL(pour les graphiques).

Le développement du jeu sera plus rapide et moins complexe qu'en utilisant directement LWJGL, tout en gardant ses performances.

Un jeu Slick2D est composé de plusieurs états : chaque état est une représentation d'un écran (cf.cahier des charges, III.1 ).

Chaque état est représenté par une classe. Voici donc la liste de états :

* Menu principal;
* Tableau des scores;
* Menu pause;
* Fin de partie;
* Jeu;
* Ajout du pseudo;

Le jeu sera disponible sous tous système d'exploitation possédant la version la plus récente de Java JRE d'installé. Le jeu sera jouable uniquement avec un clavier disposant des touches directionnelles fonctionnelles.

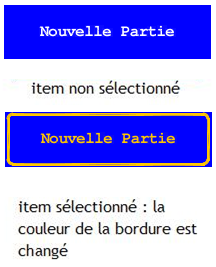
**ANALYSE DES OPTIONS DES MENUS**

Dans tous les menu du jeu les options représente les actions que l'utilisateur peut effectuer.

Dans le menu principal, par exemple, si l'utilisateur sélectionne l'option « Nouvelle Partie » et appuie sur la touché Entré, cela va lui lancer une nouvelle partie.

On observe les options peuvent être regroupés en catégorie d'option. Chaque catégorie d'option possède son propre mode de fonctionnement.

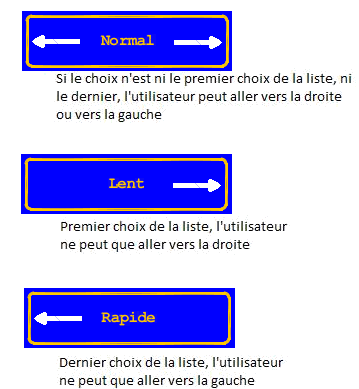
Les schémas suivante représentent les catégories d'options que nous allons utiliser avec leur mode de fonctionnement.



*Schéma 1: fonctionnement de l'option menu basique*

On observe qu'un option basique (le type/catégorie d'option qui est le plus utilisé) à les caractéristique suivante :

* il apparaît une bordure selon si il a le focus ou non du joueur
* il a un label centré avec une couleur
* il a une taille (une largeur et une hauteur)
* et enfin il a des coordonnées (pour le positionnement de l'option à l'écran lors de son affichage)

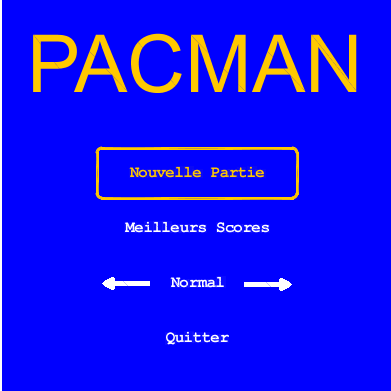


*Schéma 2: Fonctionnement option à choix multiple*

Même chose qu'un option basique avec l'ajout d'une image pour la flèche de droite et la flèche de gauche.

L'utilisateur appuie sur la touche directionnelle droite pour aller vers la droite et la touche directionnelle gauche pour aller vers la gauche.

**Menu principal**



*Schéma 4 : Écran Menu Principal*

**- Nouvelle partie :**

On utilise l'option menu basique.

Si l'option est sélectionné et que l'utilisateur appuie sur entré, on entre dans l'état Jeu.

**- Meilleurs scores :**

On utilise l'option menu basique.

Si l'option est sélectionné et que l'utilisateur appuie sur entré, on entre dans l'état Tableau des scores.

**- Difficulté (lent/normale/rapide) :**

On utilise l'option menu à choix multiple

L'utilisateur appuie sur la touche directionnelle droite ou gauche pour changer la difficulté.

**- Quitter la partie :**

On utilise l'option menu basique.

Si l'option est sélectionné et que l'utilisateur appuie sur entré, le programme s'éteint.

**ANALYSE ECRAN JEU**

* ***Le Labyrinthe***

Le labyrinthe est un tableau à 2 dimensions et chaque case représente une image qui peut être :

* un mur
* un pac-gommes
* un pac-gomme spécial
* du vide

On représente une image avec un Identifiant (0,1,2,..., n) et un chemin d'accès.

On peut ainsi stocker dans le tableau 2D un entier et afficher l'image correspondante (Voir schéma suivante).

8111111781111111111781111117

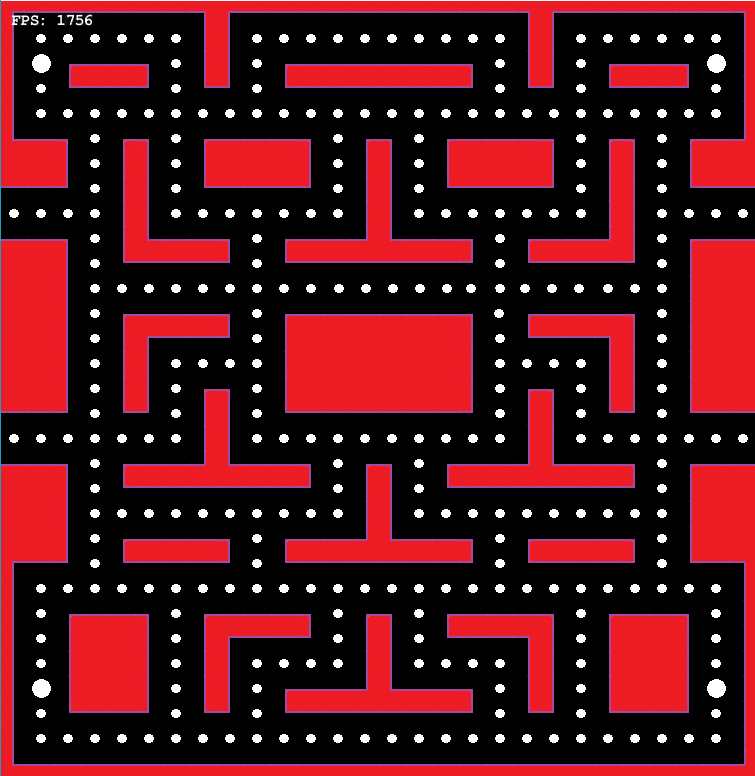
3P0000P43P00000000P43P0000P4

30B22C0430B222222C0430B22C04

30E11D0ED0E111111D0ED0E11D04

3P0P00P00P00P00P00P00P00P0P4

52C0BC0B222C0BC0B222C0BC0B26

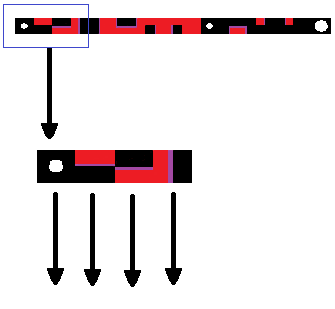


9930430499930430499930430499

11D0430E111D0430E111D0430E11

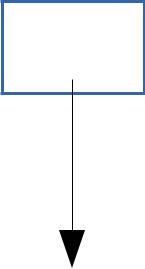
P00P43P00P00P43P00P00P43P00P

22C04522C0B226522C0B22630B22



9930E111D0E111111D0E111D0499





0 1 2 3

3P00P430499304

522C0430499304

111D0ED0E11D04

0000P00P0000P4

22222222222226

Schéma 5 : création du labyrinthe à partir d'un fichier texte et d'une texture

A noté que j'ai en réalité besoin de deux tableaux d'entier.

En effet, j'ai le tableau d'entier vue précédemment pour l'affichage du labyrinthe. Les valeurs des entiers vont changer au cours du temps : lorsque le Pac-man va manger les pac-gommes ceux-ci ne doivent plus apparaître (les images des pac-gommes doivent être remplacé par du vide).

Ensuite il nous faut un deuxième tableau d'entier qui fera la même taille que le tableau d'affichage et qui permettra de connaître d'autres renseignements pour rendre les algorithmes de déplacement des Fantômes et du Pac-man plus simple.

Le déplacement des Fantômes et du Pac-man se fait d'intersection en intersection. Il y a aussi le tunnel à prendre en compte.

Le tableau qui va aider la réalisation des algorithmes est une copie du tableau d'affichage de départ et contrairement à celui-ci, il ne sera pas changé au cours du temps afin de garder les informations concernant les intersections, les tunnels, etc.

* ***Le Pac-man***

Le Pac-man est représenté par une image qui a des coordonnées X et Y.

Son déplacement est effectué par le joueur : pour se déplacer il faut donc simplement récupérer la touche directionnelle appuyé par le joueur et changer ses coordonnées X et Y.

Toutefois, le Pac-man ne peut pas traverser les murs, il faut donc faire ce qu'on appelle une gestion des collisions avec les murs du labyrinthe.

Les algorithmes utilisés pour le déplacement du Pac-man sont décrit en annexe.

Explication sur l'algorithme de déplacement (Annexe page.... ) :

La première étape va permettre de savoir si le Pac-man est sur une intersection. En effet, il n'est pas nécessaire de chercher à changer la direction du Pac-man s'il n'est pas sur une intersection, car la seule possibilité de changements de direction est d'aller à l'opposé de sa direction courante :

– s'il n'est pas sur une intersection et que sa direction courante est GAUCHE, le Pac-man ne peut que continuer vers la GAUCHE ou alors aller vers la DROITE (donc pas besoin de tester si l'utilisateur souhaite aller vers le haut ou le bas : ce ne sera pas possible d'y aller).

– s'il n'est pas sur une intersection et que sa direction courante est HAUT, le Pac-man ne peut que continuer vers le HAUT ou alors aller vers le BAS (donc pas besoin de tester si l'utilisateur souhaite aller vers la droite ou la gauche : ce ne sera pas possible d'y aller).

–même chose pour les deux autres directions.

Maintenant, pour des raisons visuelles, l'image du Pac-man est plus grande que l'image d'une case du labyrinthe : chaque case est représenté par une image de taille 27\*25 et l'image du Pac-man à une taille de 47\*44 (voir schéma suivante).





27\*25 47\*44

Pourquoi ? =>  +  +  = 







+ =

(Couloir type du labyrinthe)

*Schéma 6 : raison visuelle pour laquelle l'image du Pac-man à une taille plus grande que la taille d'une case du labyrinthe*

Comment savoir sur quelle case se situe le Pac-man (pour savoir si cette case est une intersection ou non) ?

Normalement, la case sur laquelle est positionné le Pac-man est calculé par rapport au centre du Pac-man.

Le problème est que visuellement si le Pac-man change de direction à l'intersection avec ce calcul, celui-ci va s'afficher à moitié dans les murs, et cela fausse également les calcules des collisions.

L'objectif est de trouver le point de l'image du Pac-man qui servira de référence pour situer sur quelle case se trouve le Pac-man.

Le problème et la solution sont présentés à la schéma suivante.

Problème :

Le centre du Pac-man est sur l'intersection



Solution :

On prend un autre point de référence que le centre



N.B : le point de référence change selon la

*Schéma 7 : changement de direction aux intersections*

Enfin, la dernière étape avant de changer les coordonnées du Pac-man pour le déplacer, consiste à effectuer les différents événements qui doivent se produire selon sur quelle type de case se situe le Pac-man :

– Case vide : il n'y a rien de particulier à faire ;

– Case avec un pac-gomme : augmente les points du joueur (+10 points) et changement visuelle de la case (le pac-gomme est mangé, donc la case devient une case vide) ;

– Case avec un pac-gomme spécial : augmente les points du joueur (+50 points), le Pac-man rentre en mode Super Pac-man (il a alors la possibilité de manger les fantômes pendant quelques secondes ) et la case devient une case vide ;

– Case tunnel : Le Pac-man traverse le tunnel : il se retrouver téléporter de l'autre côté du labyrinthe (côté droit si le Pac-man se situe sur le côté gauche et inversement).

Toutes les vérifications/action effectuer, on déplace le Pac-man selon sa vitesse.

* ***Les Fantômes***

Les fantômes, au nombre de 4, sont dirigés par l'ordinateur et non par le joueur. Autrement dit, ils doivent se déplacer tout seul.

Ils possède en réalité 3 modes de déplacement :

– Le mode le plus fréquent : mode chase (« chasse » en français). Dans ce mode, les fantômes doivent chasser le Pac-man pour essayer de le manger ;

– Mode « scatter » (« éparpillement » en français) : Chacun des 4 fantômes se dirige vers un des 4 coins du labyrinthe ;

– Mode « frightened » (« effrayé » en français) : Les fantômes sont effrayés lorsque le Pac-man devient un Super Pac-man : ils prennent la première direction qui s'offre à eux lorsqu'ils sont sur une intersection.

L'algorithme de déplacement des fantômes est globalement le même que celui du Pac-man : ils se déplacent d'intersection en intersection et l'algorithme de gestion des collisions utilisé est le même.

La différence se situe au niveau des changements de directions : lorsqu'un fantôme arrive à une intersection, il doit choisir quelle direction prendre (il peut bien-sûr garder la même direction si cela est possible).

3 algorithmes sont utilisés pour ce choix de direction : 1 algorithme pour chaque mode (chase, scatter ou frightened).

A noter que chaque fantôme agit différemment lorsqu'il est en mode chase et scatter, mais en réalité la seule chose qui change est la case ciblé par le fantôme.

La direction choisit est toujours celle qui permet d'aller à la case ciblé le plus rapidement possible : on regarde donc selon les directions que le fantôme peut prendre quelle est celle où le fantôme aura le moins de distance à parcourir.

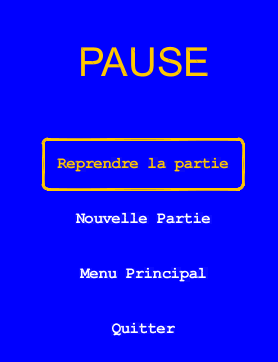
Le plus important au final est la gestion des modes.

L'algorithme de gestion des modes est relativement simple à comprendre : chaque mode à une durée et chaque fantôme à un timer (« chronomètre ») interne. Lorsque cette durée est dépassé, le fantôme remet à zéro son timer interne et change de mode.

Le mode par défaut du fantôme (celui qui est activé au lancement du jeu) est le mode chase. Au bout d'un certain temps de jeu le mode scatter n'est plus utilisé : seule les modes chase et frightened continuent.

**ANALYSE ECRAN MENU PAUSE**

L' écran menu pause est accédé en appuyant la touche « échappe » pendant une partie de jeu. Cette écran utilise le même fonctionnement pour les options de menu qui est déjà expliqué dans l'analyse des options du menu (cf.première partie du rapport). La partie de jeu qui est mis en pause est conservée dans son état précédent quand le joueur appuie sur le touche « échappe » pour signaler le changement de l’état de jeu à passer à l'écran menu pause.



*Schéma 8 : L'écran Menu Pause*

Il y a 4 options dans l'écran menu pause :

1. **« Reprendre la partie »**

Si le joueur sélectionne cette option, l'écran menu pause change l'état de jeu à l'écran de jeu où le joueur peut continuer à jouer le jeu qui était mis en pause.

1. **« Nouvelle partie »**

Le joueur peut recommencer une partie en sélectionnant cette option. Le menu pause change l'état de jeu à l'écran de jeu et l'écran de jeu réinitialise son état puisque le joueur peut jouer une nouvelle partie.

1. **« Menu Principal »**

Le joueur peut se diriger vers le menu principal en sélectionnant cette option. Le menu pause change l'état de jeu à l'écran menu principal. La partie courante est considéré terminée et le joueur ne peut plus revenir pour continuer la partie qu'il était en train de jouer.

1. **« Quitter »**

Cette option permet au joueur de quitter le programme tout simplement.

**ANALYSE ECRAN FIN PARTIE**

L'écran fin partie est accédé par l’écran de jeu quand la partie est terminé(c'est-à-dire lorsque le joueur n'a plus aucune vie) et le joueur n'a pas eu un meilleur score pour entrer son pseudo dans le tableau des 5 meilleurs scores. Cette écran utilise le même fonctionnement pour les options de menu qui est déjà expliqué dans l'analyse des options du menu (cf.première partie du rapport). Cette écran affiche aussi le nombre de points(score) que le joueur a gagné.



*Schéma 9 : L'écran Fin Partie*

Il y a 3 options dans l'écran fin partie :

1. **« Nouvelle partie »**

Le joueur a l'option de recommencer la partie de jeu en sélectionnant cette option.

1. **« Menu Principal »**

Le joueur peut se diriger vers le menu principal en sélectionnant cette option. La partie courante est alors considérée terminée et le joueur ne peut plus revenir pour continuer la partie qu'il était en train de jouer.

1. **« Quitter »**

Cette option permet au joueur de quitter le programme tout simplement.

**ANALYSE ECRAN AJOUT PSEUDO**

L'écran ajout pseudo est accédé par l’écran de jeu quand la partie termine(c'est-à-dire le joueur n'a plus aucune vie) et le joueur a eu un meilleur score pour entrer son pseudo dans le tableau des 5 meilleurs scores. Cette écran utilise le même fonctionnement pour les options de menu qui est déjà expliqué dans l'analyse des options du menu (cf.première partie du rapport). En plus, cette écran utilise un « TextField » pour enregistrer les caractères entrés au clavier par le joueur. Cette écran affiche le nombre de points(score) que le joueur a gagné et annonce que le joueur a eu un meilleur score. Le joueur doit entrer son pseudo(limité à 3 caractères comme sur la borne d'arcade) en utilisant le clavier. Après que le joueur a entré son pseudo, il doit appuyer sur la touche « entrée » pour continuer à l'écran tableau de scores qui affiche son classement parmi les 5 meilleurs scores. Son pseudo et son score sont classés et sauvegardés sur un fichier texte « highScore.txt » afin de permettre au programme d'afficher les meilleurs scores a chaque lancement du jeu.

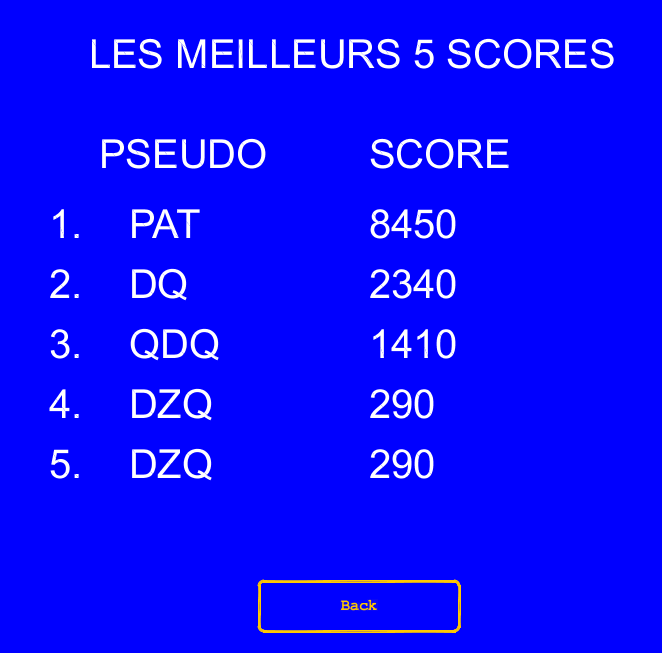


*Schéma 10 : L'écran Ajout Pseudo*

**ANALYSE ECRAN TABLEAU SCORE**

L'écran tableau score est accédé soit par l'option « Meilleur Score » dans l’écran menu principal ou soit après que le joueur a entré son pseudo dans l'écran ajout pseudo. Cette écran utilise le même fonctionnement pour les options de menu qui est déjà expliqué dans l'analyse des options du menu (cf.première partie du rapport). Les 5 meilleurs scores et les pseudos correspondant sont affichés dans cette écran selon le classement. Ces informations sont obtenus en lisant un fichier texte « highScore.txt » qui sauvegarde les 5 meilleurs scores et les pseudos correspondan

t. Le joueur peut appuyer sur l'option « Back » qui signale de changer l'état de jeu à l'écran menu principal.



*Schéma 11 : L'écran Tableau Score*

**Conclusion**

Grâce a ce projet, j'ai pu voir toutes les étapes de la conception d'un logiciel, du cahier des charges à la mise en production (ici création du fichier exécutable après avoir terminé la réalisation du jeu).

Cela nous a permis de mieux comprendre l'importance de l'analyse et de la mise en place et du respect de normes afin de posséder un logiciel qui puisse être facilement améliorable.

Toutefois, nous n'avons pas pu réaliser toutes les caractéristiques du jeu Pacman que nous voulions implémenter.

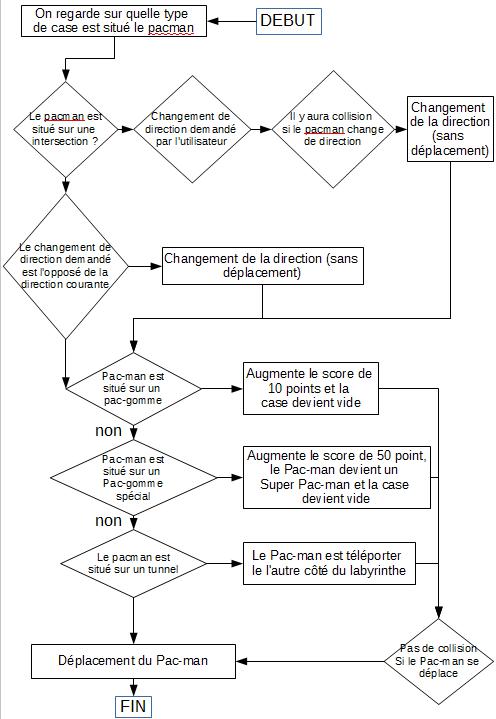
Il est donc possible d'améliorer le jeu en ajoutant des fonctionnalités comme : La gestion de bonus pour permettre au joueur de gagner plus de points.

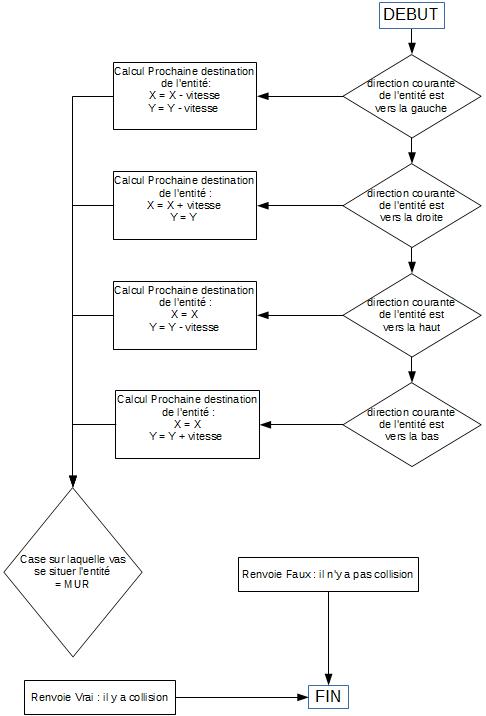
La gestion du son.

L'ajout d'un ou plusieurs labyrinthe.

**ANNEXE**

* ***Algorithme déplacement du Pac-man :***

******

* ***Algorithme test collision ***