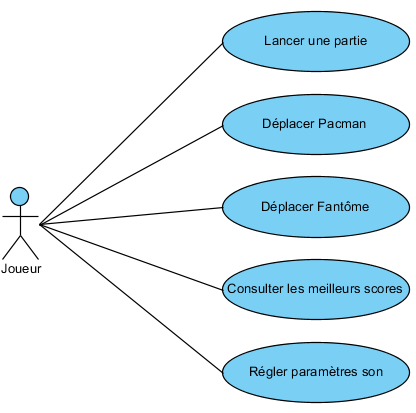
Spécifications : Itération 5

# **Généralités :**

L’objectif est de créer un jeu de type Pacman. Le jeu doit répondre à certains besoins de par sa nature.

Cas d’utilisation :



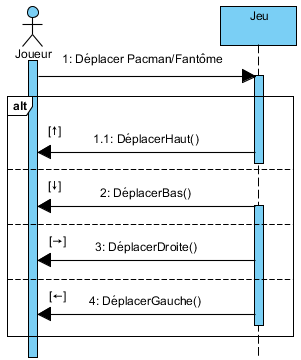
# **Personnage :**

Il s’agit d’un personnage du jeu ayant une apparence particulière (Pacman ou Fantôme). Il est défini par :

* sa position : un point de coordonnées (x,y)
* sa taille en pixel\*pixel
* une vitesse en pixel.s-1

Un Personnage est composé d'un ModèleG.

Un personnage peut : Se déplacer



Description des fonctionnalités :

**Cas : Déplacer le Personnage vers le Haut\***

→ *deplacerHaut()* : soustrait offset\*\* à la coordonnée y du personnage.

**Cas : Déplacer le Personnage vers le Bas\***

→ *deplacerBas()* : additionne offset\*\* à la coordonnée y du personnage.

**Cas : Déplacer le Personnage vers la Gauche\***

→ *deplacerGauche()* : soustrait offset\*\* à la coordonnée x du personnage.

**Cas : Déplacer le Personnage vers la Droite\***

→ *deplacerDroite()* : additionne offset\*\* à la coordonnée x du personnage.

\* Dans tous les cas de déplacements, si le Personnage heurte un mur, il revient à une position adjacente de ce dernier et ne peut jamais le traverser.

\*\* *offset* : variable utilisée comme le « pas » de déplacement des personnages, de type entier.

**Cas : Reset Personnage**

→ *resetPosition()* : remettre à l’état de début de partie les positions de tous les Personnages.

# **Le Pacman :**

Le Pacman est le personnage principal du jeu, il est donc un Personnage. Il est défini par :

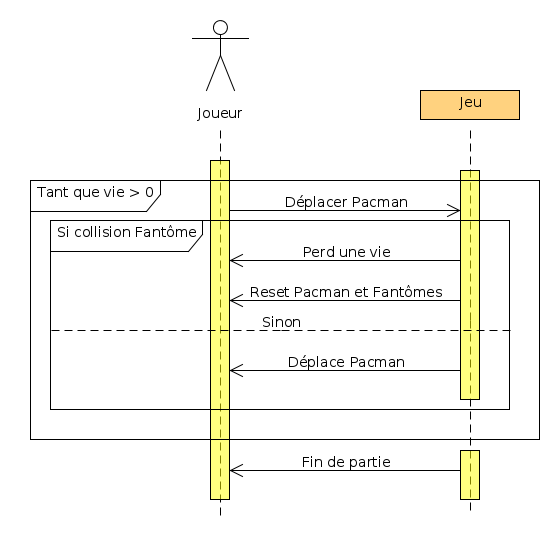
* une image : tableau de pixels
* les variables de Personnage.

Pour les déplacements du Pacman il sera utilisé un clavier (les quatre touches directionnelles), et un écran pour son affichage.

Les collisions possibles avec le Pacman :

Si en se déplaçant il heurte un Fantôme, alors tous les personnages reviennent à leur position initiale, les bonus consommés par le Pacman sont conservés, et ce dernier perd une vie.

Enfin, si le Pacman rencontre un bonus en se déplaçant, ce dernier disparait du labyrinthe pour la partie en cours et le score du joueur augmente.



Description des fonctionnalités :

**Cas : Le Pacman perd une vie (rencontre Fantôme)**

→ *perdVie()* : la vie du Pacman diminue de 1

→ *resetPosition()*

**Cas : Déplacer le Pacman**

Entrée : flèche directionnelle haute → *déplacerHaut()*

Entrée : flèche directionnelle basse → *déplacerBas()*

Entrée : flèche directionnelle droite → *déplacerDroite()*

Entrée : flèche directionnelle gauche → *déplacerGauche()*

**Cas : Fin partie\*\*\***

→ *afficherScore()* : affiche le score du joueur en fin de partie.

\*\*\* La fin de la partie arrive lorsque le Pacman rencontre un Fantôme et qu’il n’a plus de vie en réserve.

Fonctionnalité des powers :

→ *powerUp()*: active le power afin que le Pacman puisse manger les Fantômes

→ *powerDown()* : désactive le power

# **Les fantômes :**

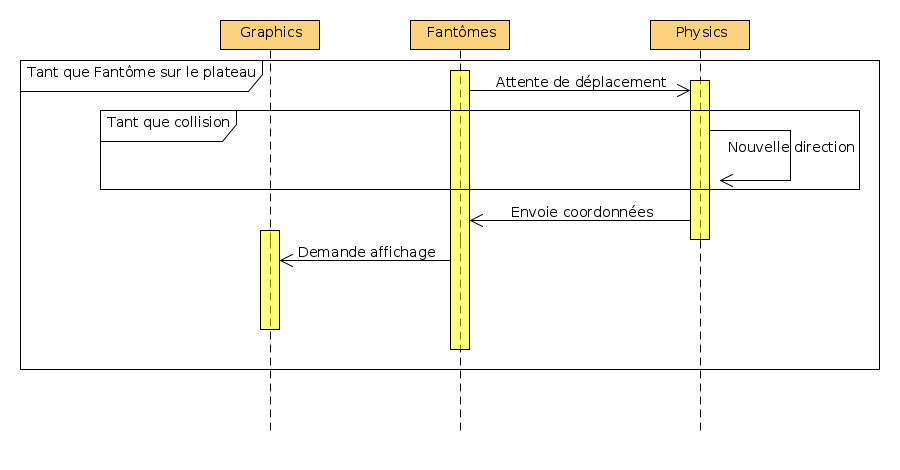
Les Fantômes sont les prédateurs naturels du Pacman. Ils ont pour seul but d’attraper le Pacman. Ils sont représentés sur le plateau de jeu par des images de petits fantômes.

Les fantômes peuvent entrer en collision avec le Pacman. Lorsqu’un Fantôme touche le Pacman il lui fait perdre une vie ou perdre la partie si le Pacman n’a plus de vie en réserve.

Il est possible de choisir le Fantôme que l’on veut contrôler dans le cas d’un test à 2 joueurs en choisissant un chiffre compris entre 1 et 4 en cours de partie.

Ils bénéficient des fonctions de déplacement d’un Personnage, ils partent de leur position initiale un peu après ce dernier et ont tous des comportements différents.

Les Fantômes peuvent : Se déplacer



Description des fonctionnalités :

**Cas : Attente de déplacement**

* **Avec un algorithme d’intelligence artificielle (à venir)**

→ *déplacerHaut()* ou *déplacerBas()* ou *déplacerDroite()* ou *déplacerGauche()*

* **Avec les touches directionnelles**

Entrée : touche « Z » → *déplacerHaut()*

Entrée : touche « S » → *déplacerBas()*

Entrée : touche « D » → *déplacerDroite()*

Entrée : touche « Q » → *déplacerGauche()*

**Cas : Nouvelle direction**

Reprendre le Cas : Attente de déplacement tant qu’il y a une collision entre le Fantôme et un Mur du labyrinthe.

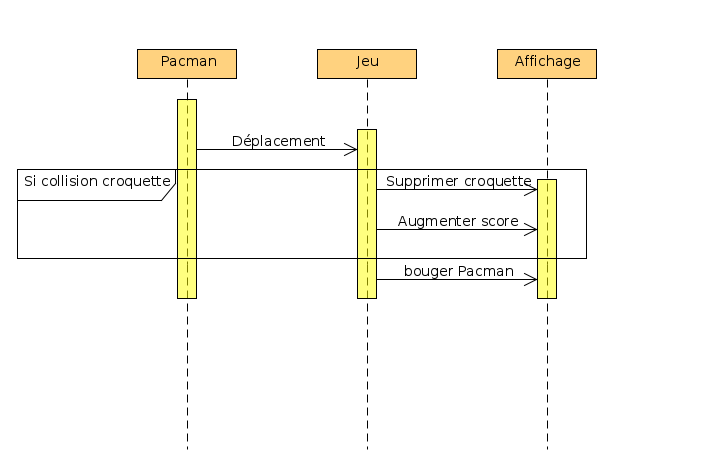
**Cas : Envoie coordonnées & Demande affichage**

Ces deux cas sont compris dans le déplacement des Personnages.

# **Les Croquettes, Fruits et Powers :**

Les croquettes sont la nourriture de base de Pacman. Celles-ci sont représentées par des billes jaunes.

Les croquettes restent immobiles sur le plateau de jeu. Lorsque le Pacman les rencontre, elles disparaissent ce qui a pour effet d’augmenter le score du joueur du Pacman.



Description des fonctionnalités :

**Cas : Déplacement**

Voir *Cas : Déplacer le Pacman* dans la section *Le Pacman*

**Cas : Supprimer croquette**

→ *supprimerCroquette()*: supprime la Croquette du jeu lorsque le Pacman passe dessus, lorsqu’il la « mange ».

**Cas : Augmenter Score**

→ *augmenterScore(x)*: augmente le score du joueur de « x » lorsque le Pacman mange une Croquette.

**Cas : Bouger Pacman**

Affiche le Pacman à la place de la Croquette.

Croquettes spéciales :

* **Les Fruits :**

→ *augmenterScore(x)*: augmente le score d’une valeur plus élevée que les Croquettes basiques.

* **Les Powers :**

→ *augmenterScore(x)*

→ *activerPouvoir(x)*: Pacman. → *powerUp()*

# **Le Labyrinthe :**

Le labyrinthe représente la zone de jeu. C’est une image composée de murs et de couloirs afin de créer un labyrinthe dans lequel le Pacman et les Fantômes pourront évoluer.

Le labyrinthe est une image contenant un Background avec des murs dessinés dessus afin de former le labyrinthe.

Celui-ci délimitera de façons visuelle, et par la suite physique la zone de jeu dans laquelle le Pacman devra serpenter afin d’éviter les Fantômes et de manger les Croquettes.

# **Le moteur Graphique**

Le moteur graphique se chargera de gérer l’affichage de tous les éléments précédents. Il s’occupera en majeure partie de la sélection de scènes (niveaux, menu…) et de leur affichage correct.

Les formules permettant un affichage correct lors de l’utilisation du jeu sont :

* Pour le Pacman :

*LargeurPacman = k1 \* Largeur*

*HauteurPacman = k1 \* Hauteur*

*Explications des variables et de la formule :*

**Largeur** et **Hauteur** sont respectivement la largeur et la hauteur du terrain de jeu.

**k** est un pourcentage.

Dans notre cas *k1 = 0.03* et *k2 = 0.062*

Ces valeurs ont été choisies afin que le Pacman soit le plus rond possible sans qu’il ne fasse de collisions avec les murs lors du redimensionnement de la fenêtre de jeu.

* Pour le Labyrinthe :

*LargeurLabyrinthe = Largeur*

*HauteurLabyrinthe = Hauteur*

*Explications des variables:*

**Largeur** et **Hauteur** sont respectivement la largeur et la hauteur du terrain de jeu.

→ *show()*: initialise la taille et la position des différents composants de la scène en fonction de la taille de la fenêtre puis les ajoute à la scène.

# **Le moteur physique**

Le moteur physique permettra aussi de contrôler la vitesse à laquelle se déplaceront le Pacman et les Fantômes par ces quelques formules :

→ *resizeEvent()* : à chaque fois que la taille de la fenêtre est modifiée, cette méthode redimensionne les composants de la fenêtre ainsi que la vitesse des Personnages.

Le moteur graphique s’occupera aussi de la collision entre les Fantômes et Pacman, mais aussi entre Pacman et ses croquettes.

# **Joueur**

Le Joueur pourra déplacer le Pacman à l'aide d'un clavier, le score du joueur augmentera à chaque fois que le Pacman mangera un Bonus.

Le joueur pourra également déplacer les 4 Fantômes un par un à l’aide du clavier.

# **ModeleG**

Un modèle graphique est composé d'une image et d'une position.

# **Entrées / Sorties**

*→ keyPressEvent()*: s’occupe de l’entrée clavier pour le traduire en fonctions compréhensibles pour le programme. Elle sera utilisée pour traduire les touches :

* « Q, Z, D, S » pour le déplacement du Fantôme choisi
* « 🡨, 🡩, 🡪, 🡫 » pour le déplacement du Pacman
* « 1, 2, 3, 4 » pour le choix du Fantôme à déplacer.

→ *showEvent()*: au lancement de l’application cette méthode est utilisée, elle appelle la méthode show() du moteur graphique et initialise la scène.