PACMAN

Description du diagramme de classes

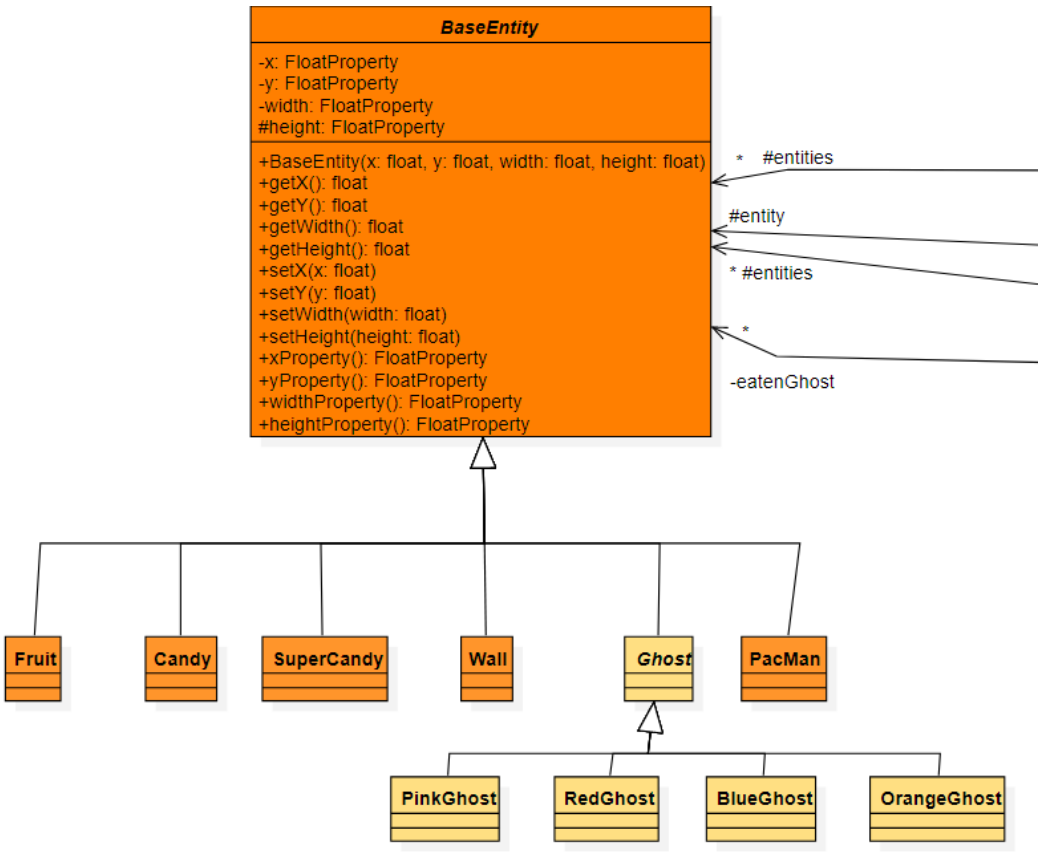
Florent MARQUES - Faustin LEPERON

Table des matières

Les entités (entity) 2
Les collisionneurs (collider) 2
Les observateurs (observer) 3
Les mangeurs (eater) 3
Les déplaceurs (displacer) 4
Les animateurs (animator) 4
Les boucleurs (looper) 5
Les utilitaires (utils) 5
Les chargeurs et sauveurs (loader, saver) 6
Le lanceur du jeu (Launcher) 6
Les managers (SpriteManager, SoundManager) 7
Le score (Score) 7
Le jeu (Game) 8
Le monde (World) 8

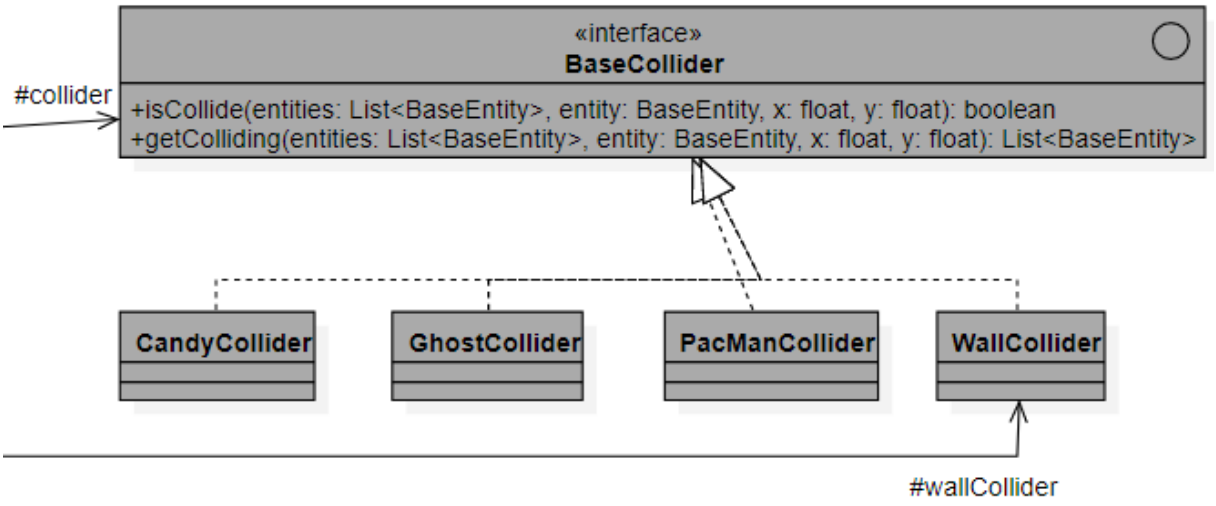
Les entités (entity)

Toutes les entités du jeu héritent de la classe « BaseEntity » qui possèdent les attributs et méthodes nécessaires pour la position sur une carte tel que sa position x et y ainsi que sa largeur et sa hauteur.



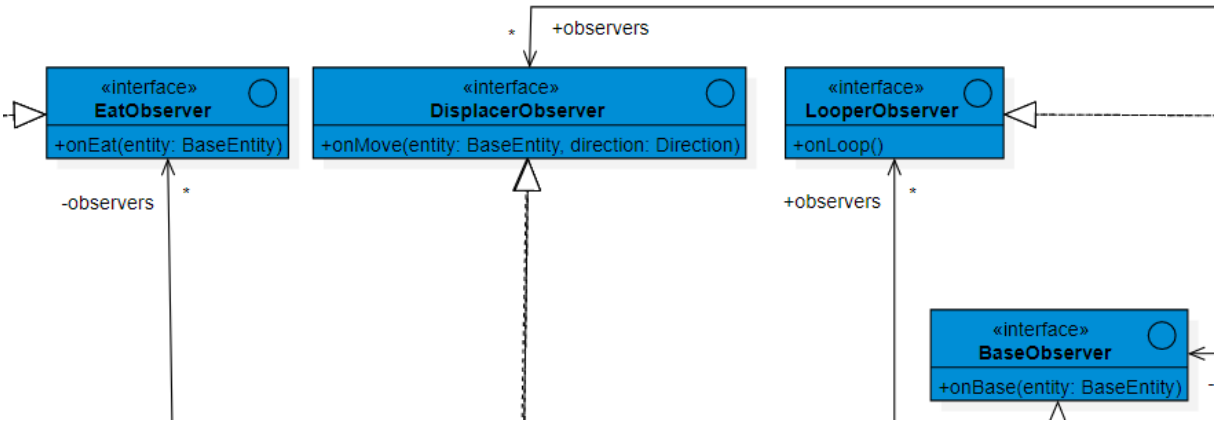
Les collisionneurs (collider)

Ils permettent de gérer les collisions entre les entités. Un collisionneur concret doit implémenter l'interface « BaseCollider » pour pouvoir définir les méthodes « isCollide » et « getColliding ». Le collisionneur permet de définir comment deux entités rentrent en collision.



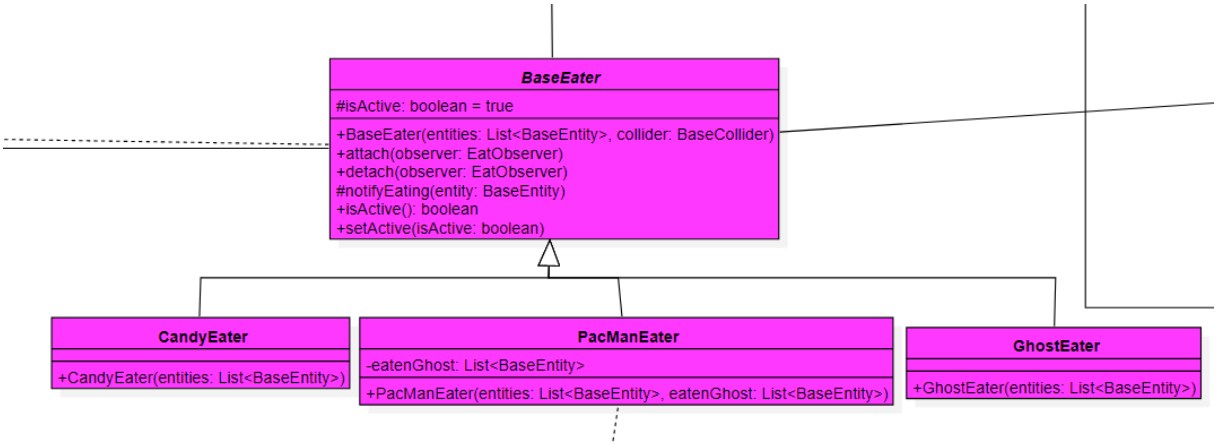
Les observateurs (observer)

Ce sont les interfaces nécessaires pour pouvoir implémenter le patron de conception observateur. Dans notre cas, ils existent 4 types d'événements possibles : « Manger », « Déplacer », « Boucler », « Point de départ ». Les classes qui implémentent ces interfaces seront notifiés par les classes implémentant le patron de conception observateur.



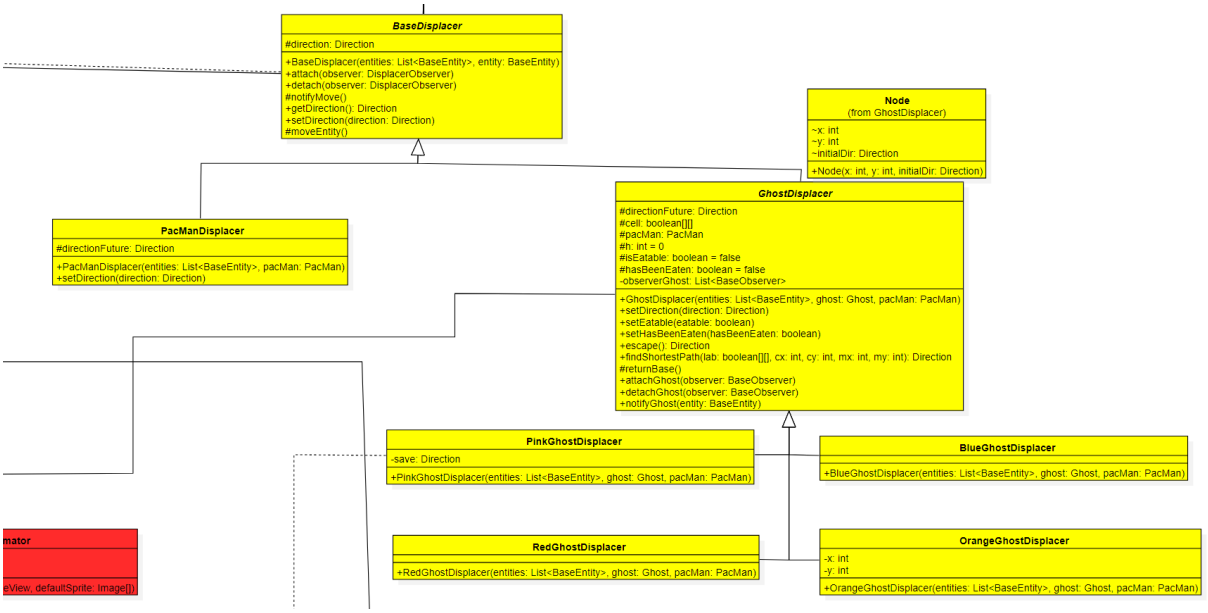
Les mangeurs (eater)

Ils permettent de déterminer si une entité est mangée. Ils utilisent un collisionneur et implémente le patron de conception observateur pour notifier qu'une entité peut être mangée aux observateurs. Les mangeurs sont des « observateurs de déplacements », c'est-à-dire qu'ils implémentent la méthode de l'interface « DisplacerObserver » ce qui permet de définir le bout de code à faire lorsqu'un déplacement a été effectué.



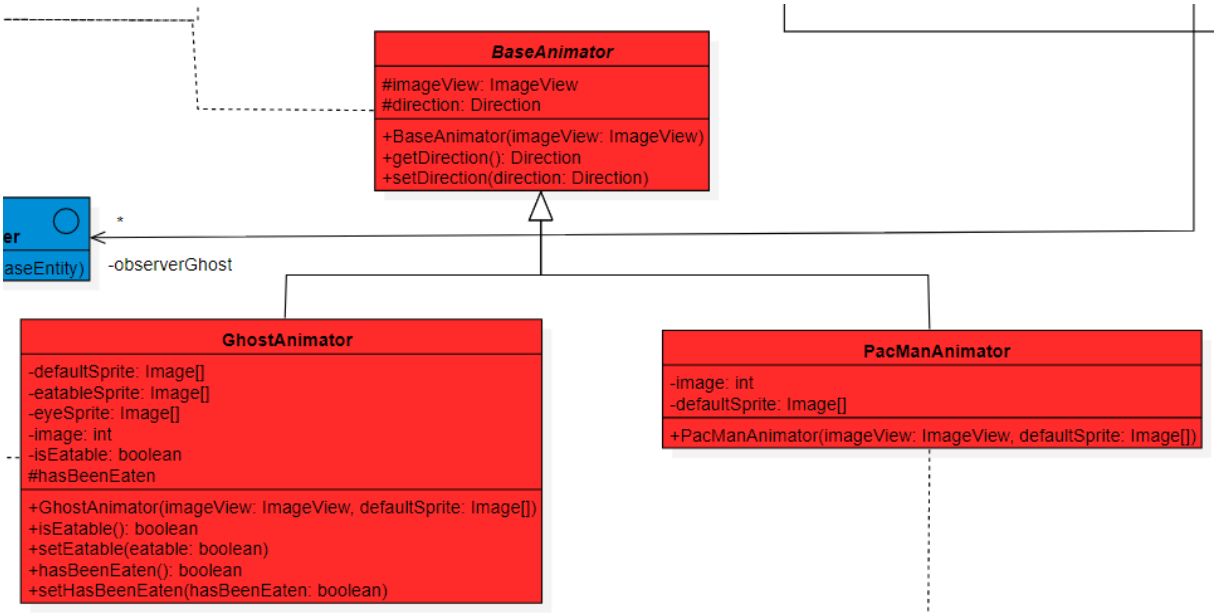
Les déplaceurs (displacer)

Ils servent à déplacer les entités si cela est possible. Pour cela, les déplaceurs utilisent le collisionneur de mur par exemple pour savoir si l'entité ne va pas rentrer dans un mur. Les déplaceurs ont aussi tous leurs manières de déplacer une entité. Par exemple les déplaceurs de fantômes n'agissent pas de la même façon car ils implémentent 4 intelligence artificielle différentes. Les déplaceurs implémentent le patron de conception observateur pour notifier le changement de position d'une entité.



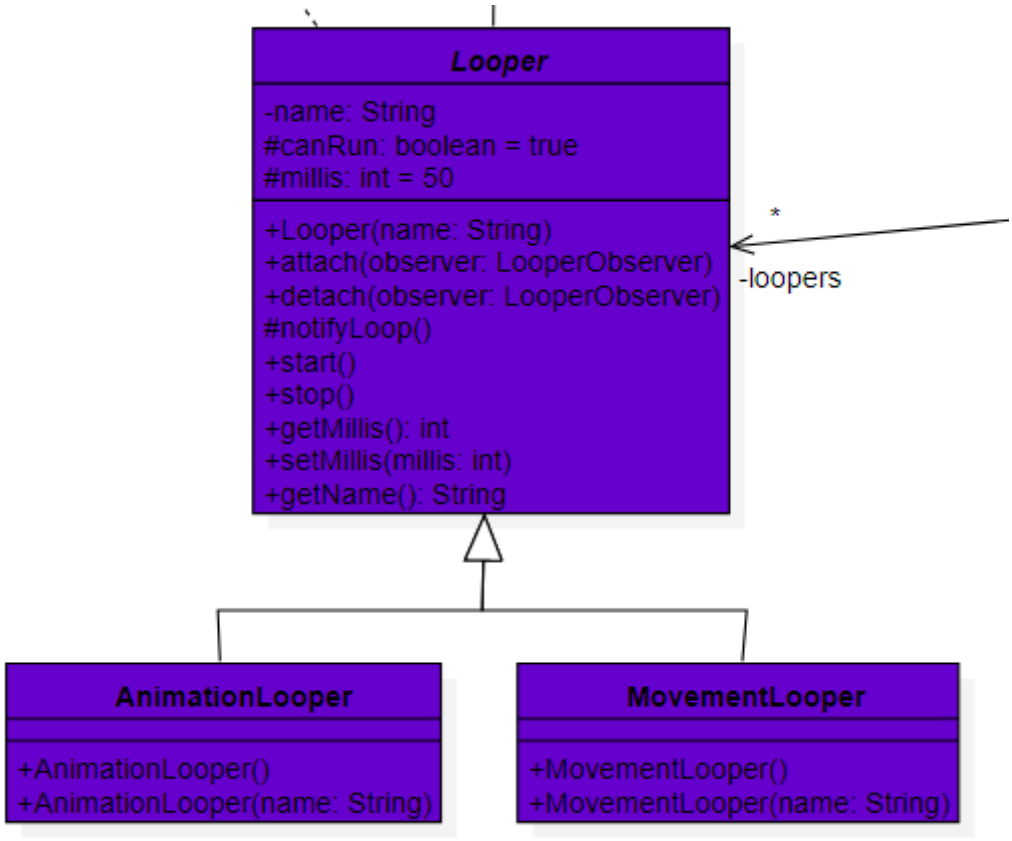
Les animateurs (animator)

Ils permettent d'animer l'entité en changeant son apparence en fonction de sa direction ou de son état. Par exemple, si Pac-Man se déplace vers le bas, l'animateur va changer l'apparence de l'entité pour qu'elle corresponde à la direction de Pac-Man.



Les boucleurs (looper)

Les boucleurs permettent de donner la cadence du jeu comme la vitesse des déplacements ou des animations. La classe mère « Looper » est abstraite et implémente le patron de conception observateur, cela permet aux observateurs d'être notifié qu'un tour de boucle a été effectué et donc que de nouvelles actions doivent être effectuées.

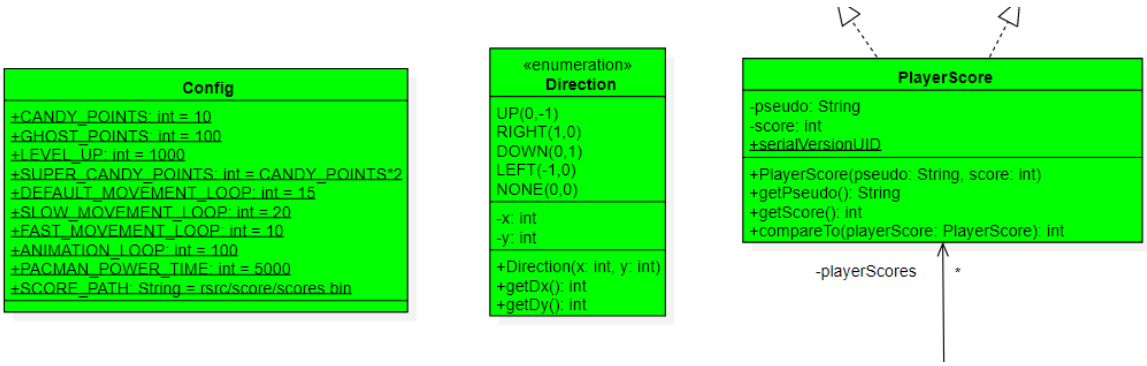


Les utilitaires (utils)

La classe « Config » permet de définir des paramètres pour toute l'application.

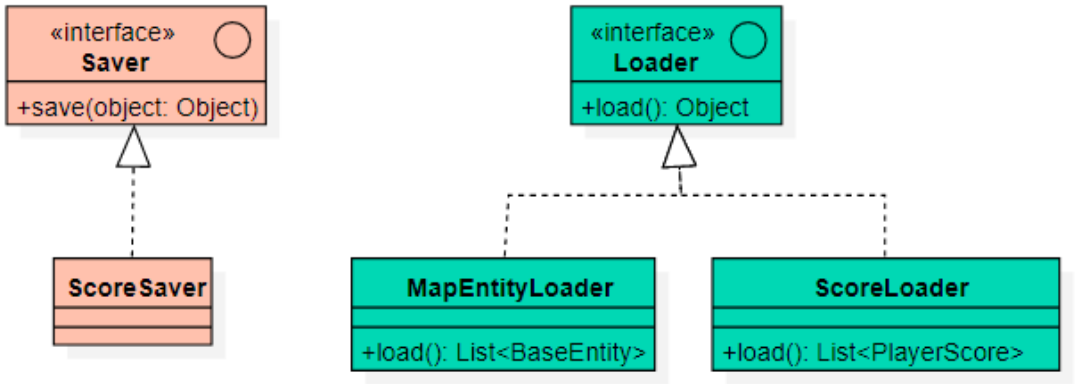
L'énumération « Direction » permet d'orienter les entités selon un repère orthonormé.

La classe PlayerScore permet de sérialiser les informations du joueur pour pouvoir les charger et les sauvegarder dans et à partir d'un fichier.



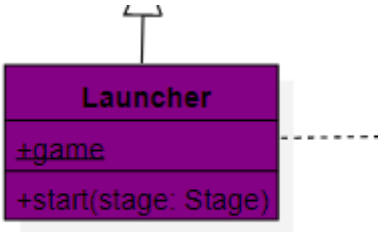
Les chargeurs et sauveurs (loader, saver)

Les deux interfaces permettent de définir comment charger un objet à partir d'un fichier et comment le sauvegarder dans un fichier.



Le lanceur du jeu (Launcher)

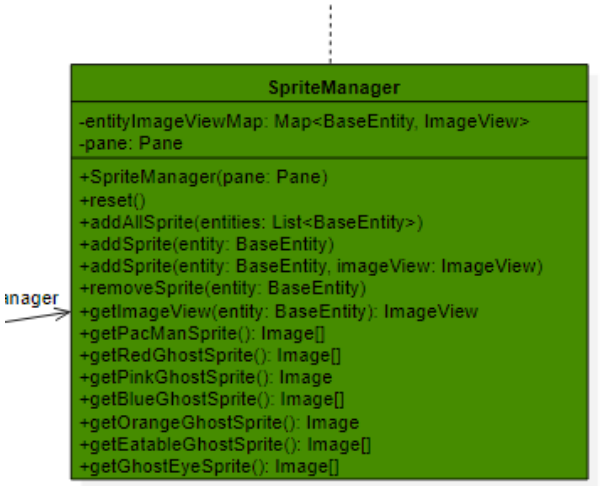
Cette classe permet de lancer le jeu et de paramétrer la fenêtre.

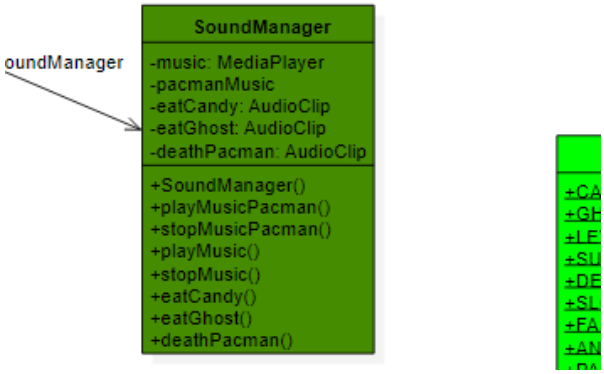


Les managers (SpriteManager, SoundManager)

Le manager de sprite, dit « SpriteManager », permet de gérer les sprites présents sur la carte donc sur la vue.

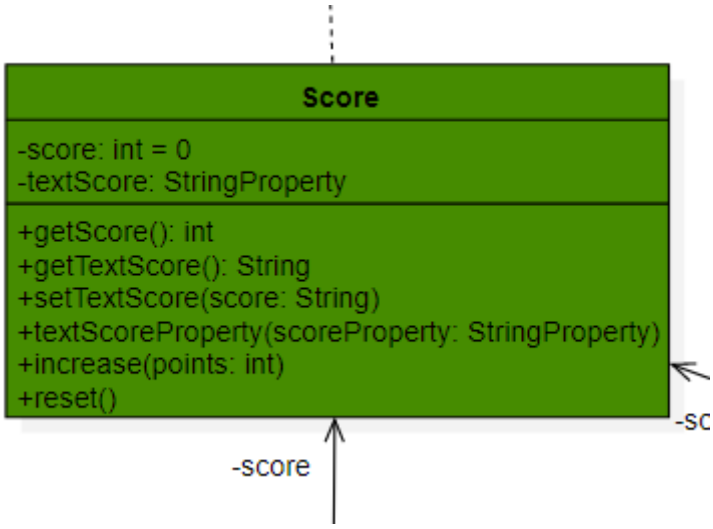
Le manager de son, dit « SoundManager », permet de gérer les sons du jeu.





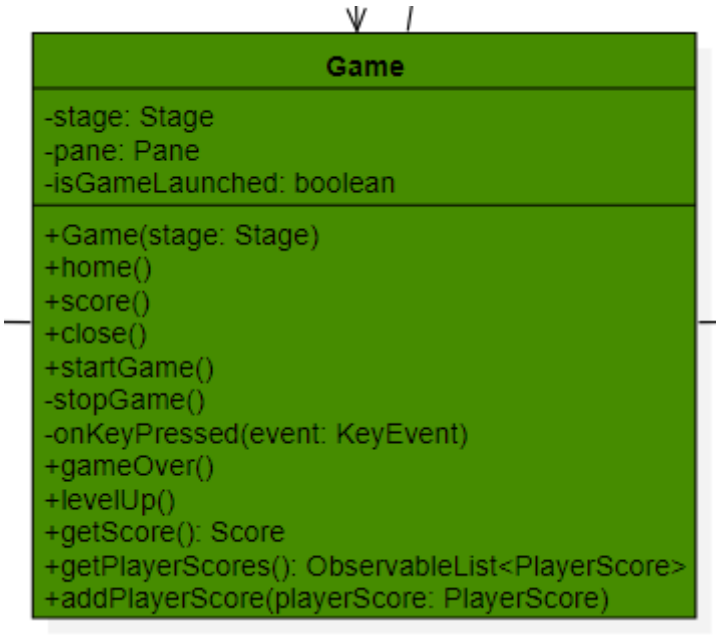
Le score (Score)

Cette classe permet de gérer le score d'une partie. La classe « Score » possède une propriété de type « StringProperty » pour pouvoir afficher le score en chaine de caractères.



Le jeu (Game)

Cette classe représente le jeu complet, elle permet de naviguer entre les différentes vues du jeu et les différentes phases du jeu (accueil, jeu, game over, score).



Le monde (World)

Cette classe représente le monde de Pac-Man avec toutes les entités associées mais contient aussi tous les classes en rapport avec les entités durant la partie tel que les animateurs ou encore les déplaceurs.