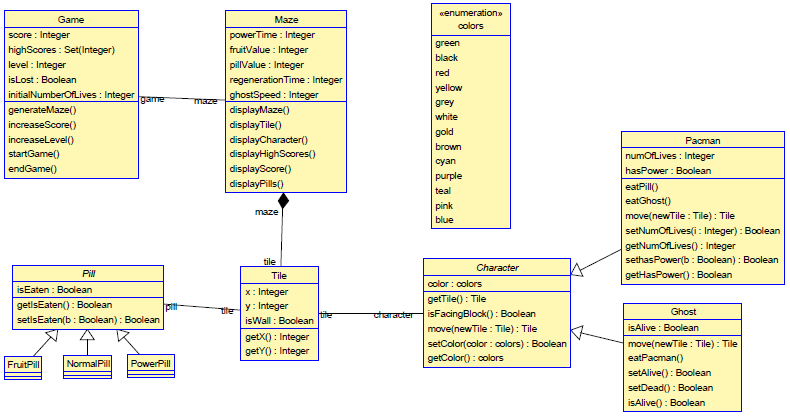
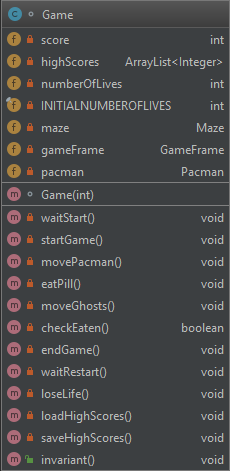
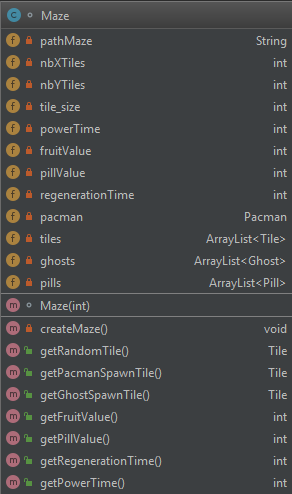
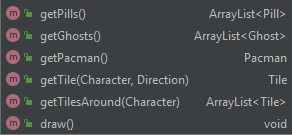
# Modélisation initialement prévue



# Différences par rapport à la prévision

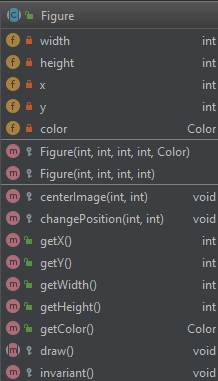
Les méthodes eatPacman(), eatGhost() ou encore eatPills() qui étaient censées être dans les classes Pacman et Ghost sont finalement dans la classe Game, puisque ces méthodes nécessitent des données de tout le plateau, or les classes Pacman et Ghost n’ont pas accès aux autres cases du plateau. C’est donc dans la classe Game que tout est géré.

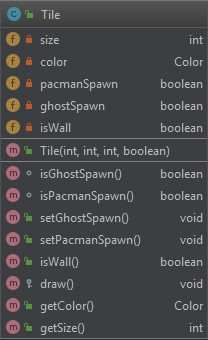
Des méthodes initialement prévues n’ont finalement pas été implémentées, comme increaseLevel(), puisqu’il suffit d’incrémenter l’attribut level. Enfin, nous avons rajouté quelques méthodes comme waitStart() et waitRestart(), qui communiquent avec la fenêtre GameFrame.



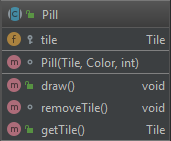
Nous avons rajouté, en attribut de la classe Maze, Pacman et les Fantômes, puisque différentes méthodes des classes Maze ou Game nécessitent ces objets. Par rapport à la modélisation prévue, nous avons rajouté getTilesAround(), permettant d’obtenir les cases disponibles utour d’un personnage. Cette méthode est notamment utilisée pour le déplacement des fantômes. Nous avons également rajouté getTile(), qui renvoie la Tile située à la direction indiquée en paramètre par rapport au personnage également donné en paramètre.

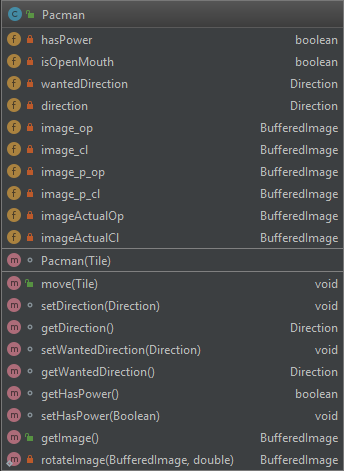
Toute les méthodes display () sont regroupées dans la méthode draw(), qui dessine tous les éléments du plateau.

Afin de dessiner les éléments du jeu dans une fenêtre, nous avons créer une classe Figure, dont héritent les classes Character, Pill et Tile. Une figure est un élément qui peut être affiché à l’écran. Chaque figure a des coordonnées x et y indiquant leur position sur l’écran, et des dimensions.

Nous avons rajouté des attributs pacmanSpawn et ghostSpawn, indiquant si la Tile est le point d’apparition de pacman ou des fantômes. Si ceux-ci ne dont pas définis, alors les personnages apparaissent aléatoirement sur le plateau.

La taille des cases est également un attribut de Tile.

L’attribut isEaten initialement prévu n’a pas été implémenté. Pour savoir si la Pill a été mangée, on vérifie si la Tile associée à la Pill est existante ou null. Lorsque Pacman mange une Pill, alors la Tile associée à cette Pill devient null.

Nous avons rajouté beaucoup d’attributs et méthodes dans la classe Pacman. L’attribut isOpenMouth indique si Pacman a la bouche ouverte ou fermée, puisqu’il ouvre/ferme celle-ci de manière continue pendant la partie. En fonction de cette ouverture, l’image à afficher est différente. Les images de Pacman dans ses différents états sont aussi des attributs rajoutés, comme image\_cl ou encore image\_op.

Le mouvement de Pacman est différent par rapport à ce qui avait été prévu. Pacman se déplace sur la case se situant dans la direction actuelle. Quand l’utilisateur appuie sur une des fléches, la « wantedDirection » de Pacman change, et dès qu’il a l’occasion d’aller dans cette direction, il s’y rend et sa « direction » est changée. Si aucun changement de direction n’est demandé, ou si la direction voulue n’est pas accessible, il continue dans sa direction actuelle.