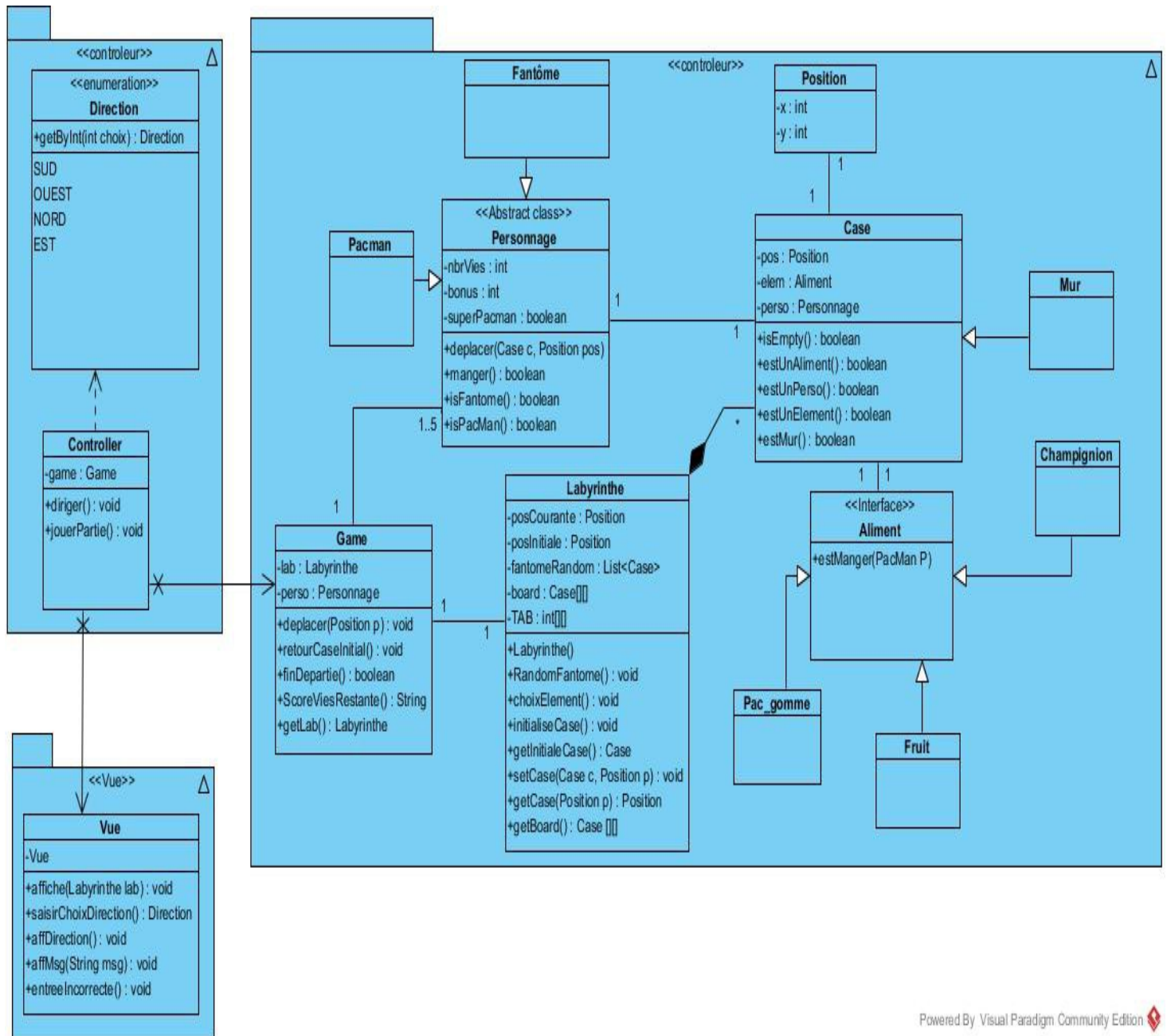
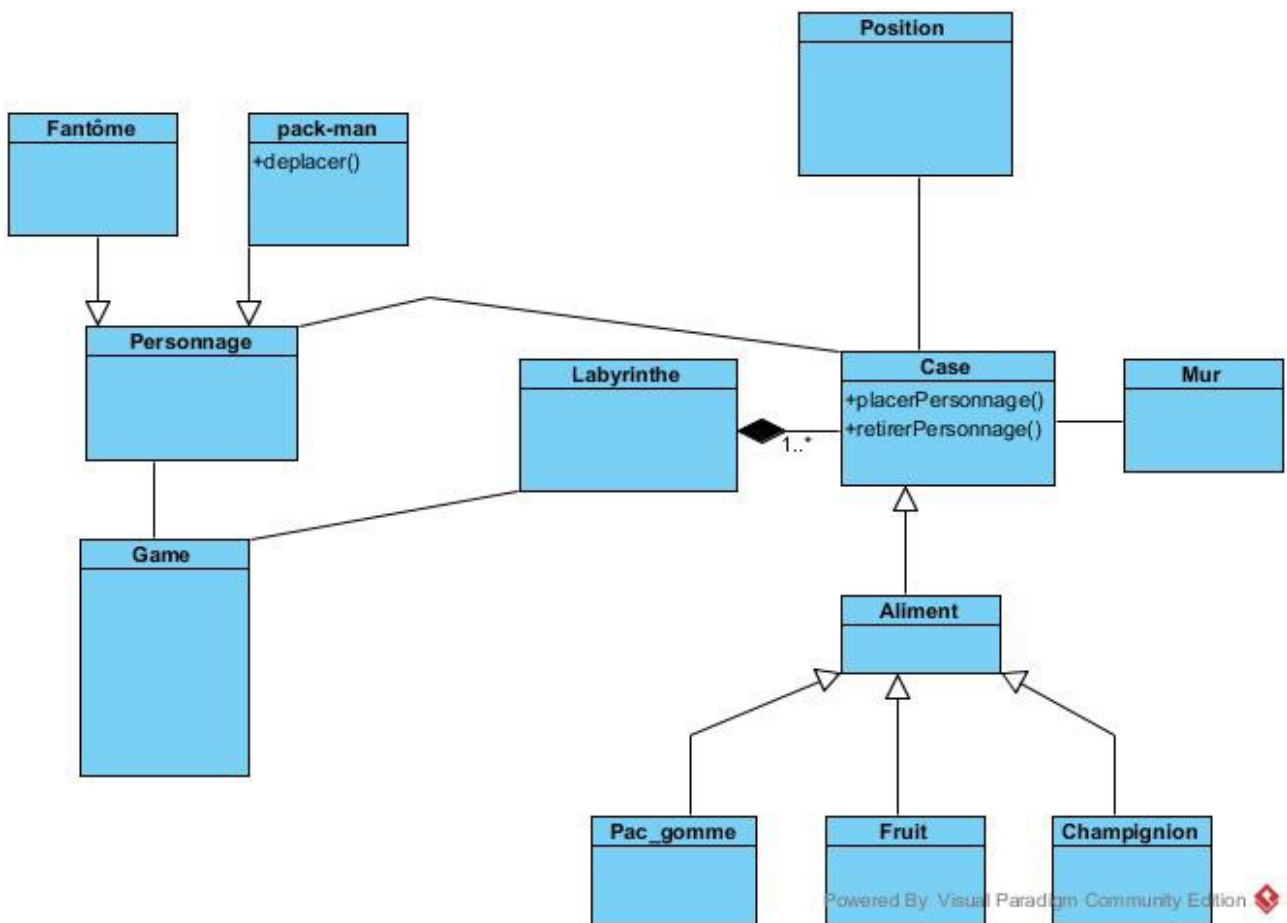


## Class Logiciel :



## Class Domaine :



## Pacman justification :

### Explication choix du Modèle :

Dans notre programme nous avons créé deux classes :

- 1) **Personnage** :(Fantome et Pacman) qui héritent de Personnage.
- 2) **Aliment** : (Fruit, Champignon, Gomme ) qui implémentent Aliment.

Afin de faciliter la construction logique de la Case, nous avons eu besoin de deux constructeurs qui prennent en paramètre Personnage ou Aliment.

Ceci, au lieu de créer 6 constructeurs qui auraient pris en paramètre tous les personnages et les aliments de notre jeux.

Donc avec l'implémentation de ces deux classes nous avons factorisé notre code .

### a) classe Personnage :

La classe **Personnage** est une classe abstraite qui possède des méthodes abstraites et concrètes , plus des attributs. On a défini comme attributs *bonus* et *nbrVies* dans cette classe car la méthode *manger* est définie dans cette classe donc ces deux attributs sont modifiés à chaque appel de cette méthode.

La méthode *deplacer* permet de changer l'attribut Personnage, dans la case de destination et via différentes sous méthodes qui vont changer la valeur des différents attributs pour le bon fonctionnements du jeu.

On ne veut pas que le Personnage connaisse le Labyrinthe et que les méthodes qui gèrent les déplacements soient dans la classe personnage.

### b) class Aliment :

La classe **Aliment** est une interface car on possède que des signatures de méthodes abstraites.

### c) class position :

La classe **Position** définit le positionnement dans le labyrinthe.

### d) class Labyrinthe :

Cette classe possède un tableau de case afin de définir tous les éléments et personnages, elle possède aussi deux attributs position courante et position initiale .

Un constructeur fait appel à deux méthodes, ce qui permet de gérer l'initialisation des cases du tableau .

Une de méthode gère l'emplacement des fantômes, l'autre permet d'initialiser les éléments et l'initialisation du pacman, elle permet aussi d'initialiser la position initiale et position courante.

Une autre méthode permet de gérer le changement des attributs après le passage du pacman.

### d) class Game :

Cette classe possède deux attributs.

Labyrinthe : c'est là où pacman se déplace.

Personnage : on recupère sa première position initiale afin de pouvoir le déplacer dans le labyrinthe.

Une méthode *deplacer* permet de renvoyer le pacman à sa case initiale lorsqu'il perd une vie et un appel à la méthode *initialise* de labyrinthe remet les attribut a « null » derrière le pacman.

