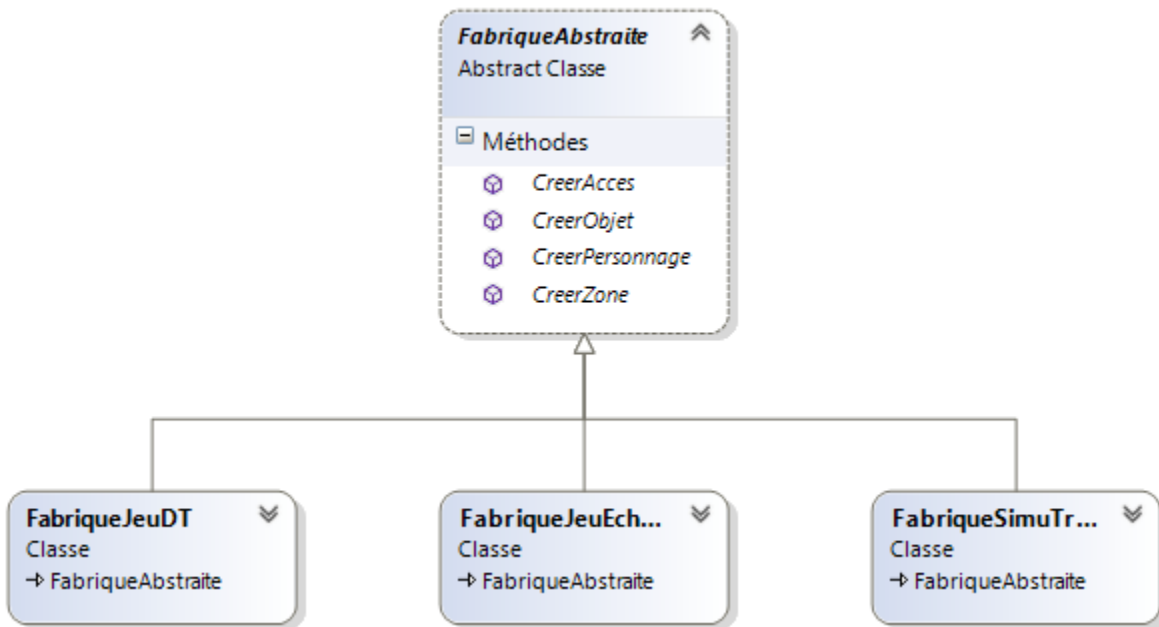


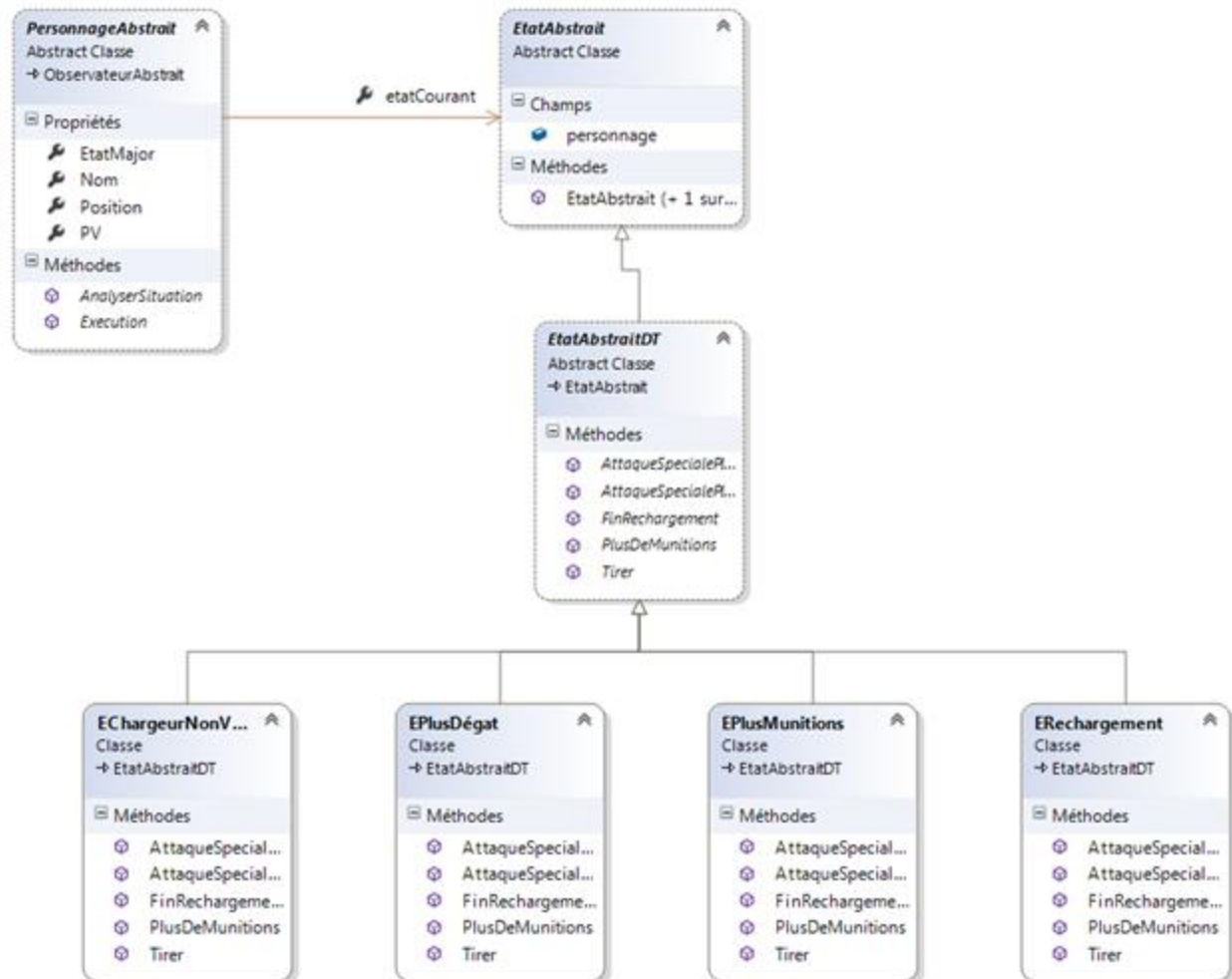
## Descriptif des patterns utilisés

Fabrique/Fabrique abstraite



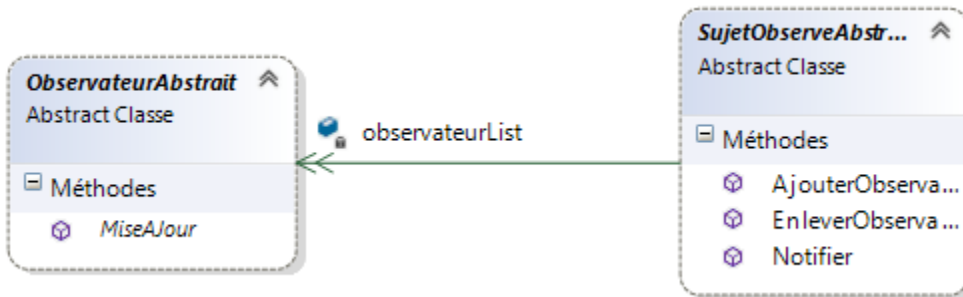
La fabrique est utilisée pour créer les Accès, les Zones, les Personnages et les Objets pour chaque jeu. Chaque jeu implémente la Fabrique Abstraite et gère sa propre fabrique. Les fabriques seront utilisées lors de l'instanciation des différents jeux par le chargement des fichiers XML.

## État



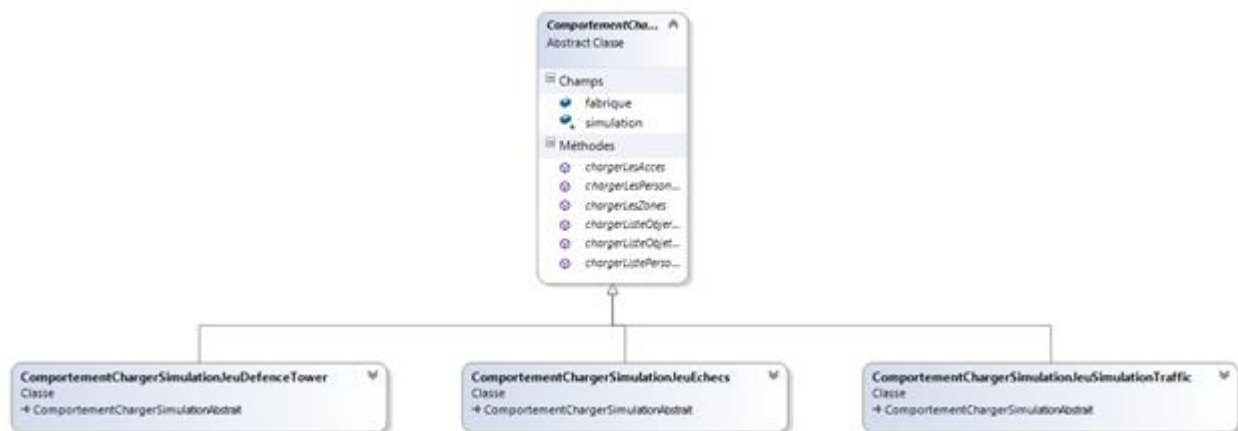
Le pattern état est utilisé dans le jeu de Tower Defense pour affecter des tirs spéciaux au château. En fonction de l'état du tir, soit le château fait des dégâts normaux soit des dégâts doubles. Si le château n'a plus de munition, il rentre dans l'état "Rechargement" et ces tirs ne font plus de dégât car il doit recharger. Une fois les munitions recharger, il retourne dans l'état ChargeurNonVide et peut continuer de tirer.

## Observateur



Utilisé dans la simulation de trafic routier, pour la synchronisation des feux de signalisation. Deux feux s'observent l'un et l'autre, quand l'un passe au rouge, l'autre passe au vert et vice versa.

## Stratégie



Utilisé pour gérer les comportements des Personnages (déplacement, attaque...) ainsi que pour le chargement des fichiers XML (un comportement par scénario).



Par exemple, le comportement de déplacement des pièces d'échecs utilise ce pattern. Un comportement général définit la base des déplacements et chaque comportement de pièce hérite du comportement général pour définir les déplacements possible des pièces. L'ensemble des pièces sauf cas particulier (pion) appelle les méthodes du comportement des pièces d'échecs général car elles sont basées sur le même algorithme. L'algorithme prend en compte les déplacements infinis (tour, fou et la reine), pour cela il faut que le comportement de la pièce en question indique qu'elle peut avoir un déplacement infini.

## Singleton

Utilisé pour les fichiers de configuration de l'application.