

# Simulateur de Fourmilière

Rappel de l'objectif : **Développer un logiciel permettant de créer une Simulation de Fourmilière.**

## MainViewModel et classes de base

MainViewModel : contient tous les paramètres de la simulation avec la possibilité de charger et de sauvegarder une partie.

Statistique : contient le nombre de Fourmi présentes sur le Terrain, le nombre de tours passés depuis le début du Jeu, le nombre de phéromones nourriture ou maison sur une Case sélectionnée.

Terrain : contient un tableau 2 dimensions de taille x sur y.

FabriqueGénéral : permet la génération d'une Fourmilière, de Fourmi de Nourriture.

## Classes relatives à une Case

Case : contient un nombre de phéromones "maison", un nombre de phéromones "nourriture" et un tableau de Fourmi.

Case Normal : est une Case vide.

Case Fourmilière : contient une Fourmilière.

Case Nourriture : contient une Nourriture.

## Classes relatives à la Fourmi

Fourmilière : contient un nombre de Nourriture rapportées par les Fourmi et génère des Fourmi sur sa Case spontanément en fonction du nombre de Nourriture dans celle-ci.

Fourmi : a un nombre de points de vie défini et peut se déplacer de Case en Case en utilisant la Stratégie RechercheNourriture jusqu'à ce qu'elle tombe sur une Case. Nourriture. Une fois la Nourriture ramassée, elle utilise la Stratégie RetourMaison pour apporter la Nourriture à la Fourmilière. Lorsque sa vie atteint 0, elle meurt.

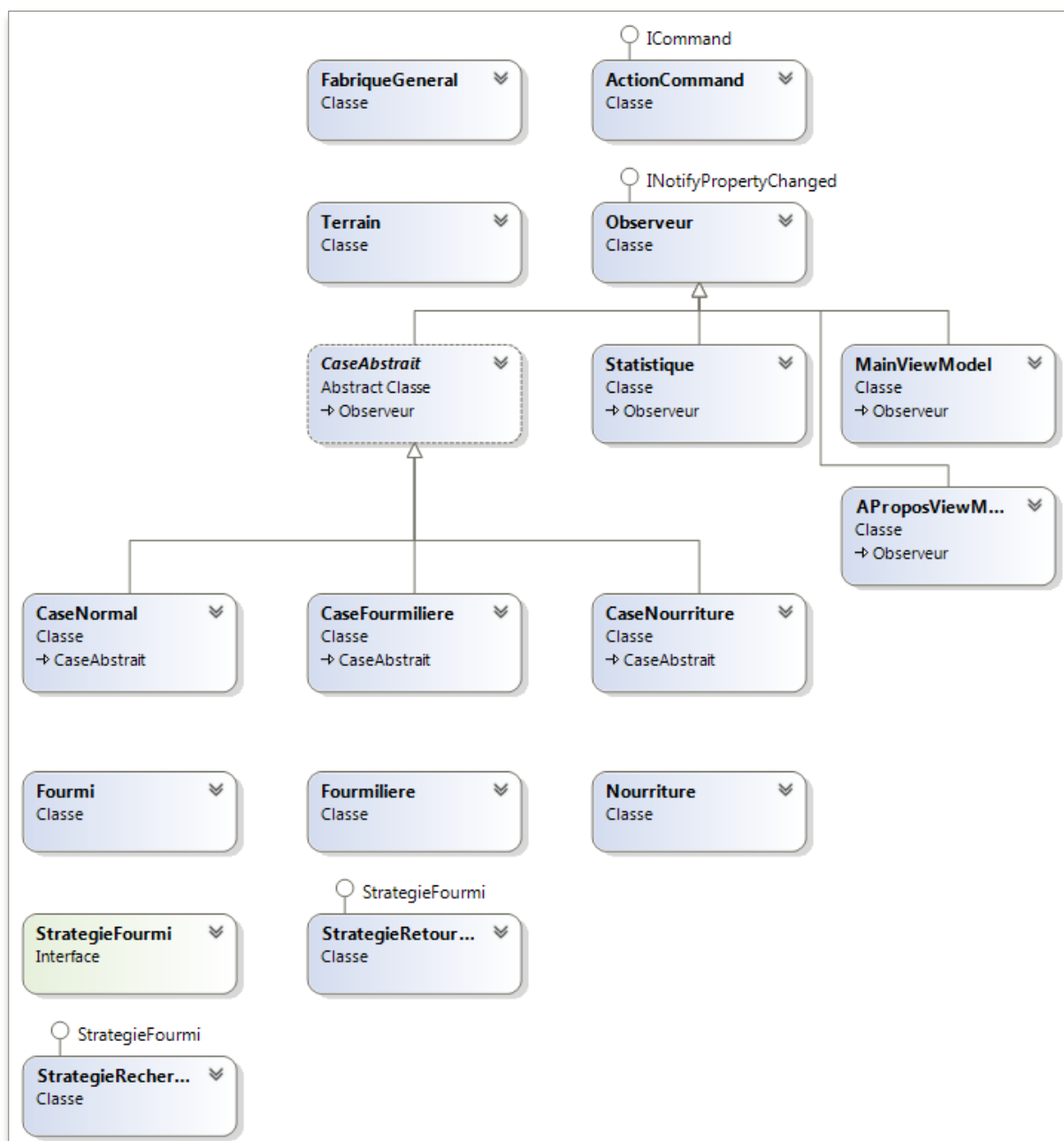
Nourriture : a un poids défini et en perd lorsqu'elle est mangée par une Fourmi.

# Classes de Stratégie

Stratégie Fourmi : interface pour les Stratégie RechercheNourriture et Stratégie RetourMaison.

Stratégie RechercheNourriture : permet l'ajout d'une phéromone maison lorsqu'une Fourmi utilise cette Stratégie.

Stratégie RetourMaison : permet l'ajout d'une phéromone nourriture lorsqu'une Fourmi utilise cette Stratégie.



# Diagramme de classes

## Patterns utilisés

### Observateur

Il envoie un signal aux différentes classes qui l'utilisent pour les notifier d'un changement sur celles-ci. Utilisé par Case, Statistique et MainViewModel.

### Stratégie

Il permet de permuter dynamiquement les méthodes utilisées par les Fourmi pendant leurs excursions entre la Fourmilière et la Nourriture.

Il est utilisé par StratégieFourmi (interface de StratégieRechercheNourriture et StratégieRetourMaison).

### Commande

Il communique l'action à effectuer lorsque l'utilisateur clique sur un bouton du menu. Utilisé par MainViewModel.

### État

Il crée et gère dynamiquement les comportements de l'application. Utilisé par Case qui est l'état et Terrain le contexte.

### Décorateur

Il englobe la Fourmilière et la Nourriture dans ses Case respectives. Utilisé par CaseNormal, CaseFourmilière et CaseNourriture.