Le code est en grande partie en orienté objet et se sépare en 7 blocs dont 4 sont des classes.
Les classes sont Merchant ,MerchantBehaviour ,MerchantGridAI et World.
De manière générale les villes et produit sont des entiers.
World correspond a la classe des mondes dans lesquels les marchands évoluent.
Les éléments de cette classe ont pour attributs une grille de production(qui peut être donnée a la création) et une grille de quantités de produits ainsi qu'un nombre de villes et un nombre de types de produits et aussi une fonction d'évaluation des valeurs des produits qui prends en paramètre la quantité d'un produit dans une ville et renvoie la valeur du produit dans cette ville.
Un monde a les méthodes suivantes :
prendreproduit qui permet de décrémenter la quantité de produit d'une ville de 1, donnerproduit qui permet d'incrémenter la quantité de produit d'une ville de 1, evaluercoutproduit qui utilise la fonction d'évaluation pour déduire le coût d'un produit dans une ville et passeruntour qui ajoute les productions au quantités de produits.
A la création fourchette corresponds a la fourchette de valeurs aléatoires possibles pour les productions d'un monde si l'on ne passe pas directement la grille de production.

Merchant corresponds a la classe des marchands qui achètent et vendent des produits.
Les éléments de cette classes ont pour attributs une ville (peut être None) , une quantité d'argent et un comportement ainsi qu'un trajet planifié (un élément de la classe MerchantBehaviour ou None).
Les marchands ont les méthodes suivantes : effectuertransaction qui prends en paramètre un chemin(un chemin est un triplet ville1,ville2,produit) et un monde et qui si le marchand est en ville 1 tente d'acheter un de produit produit en ville 1 puis déplace le marchand en ville 2 et tente de vendre le produit en ville 2(s'il l'a acheté en ville 1) cette fonction renvoie le bénéfice sur le trajet (négatif si c'est une perte) et faire un trajet coûte 1
associercomportement permet d'associer un comportement au marchands.
associerville permet d'associer une ville a un marchand.
choisirtransaction demande au comportement du marchand de choisir un trajet
passeruntour : le marchand effectue le trajet et indique a son comportement le bénéfice.

Puis il y a les classes de comportement(j'utilise un patron stratégie) il y a MerchantBehaviour qui donne la structure des classes comportement des marchands et qui doit être hérité par les stratégies concrètes.
MerchantGridAI est une de ces stratégie concrètes.
MerchantBehaviour indique que les ais doivent avoir une méthode choisirchemin avec comme paramètre la position et une méthode echecoureussite qui prends en paramètre un chemin et un bénéfice.
Les éléments de MerchantGridAI ont comme attributs une grille de poids 3d de dimensions nbvilles départ,nbvilles arrivée,nbproduits et une fonction d'ajustement.(a la création ils ont besoin d'un monde en paramètre pour structurer leur grille)
Pour choisir un trajet il regarde dans sa grille de départ la ville de départ et prends au hasard une ville d'arrivée et un produit et a plus de chance de prendre les paires avec de plus haut coefficients(il a N fois plus de chance de faire un trajet a coefficient N qu'un trajet a coefficient 1).
la méthode echecoureussite utilise la fonction d'évaluation (attribut du comportement) et le bénéfice ainsi que la valeur dans la grille pour déterminer la nouvelle valeur de la grille pour remplacer l'ancienne.
(Essentiellement par défaut si un trajet a un bénéfice positif l'on augmente la valeur sinon on la réduit)

Ensuite il y a les blocs non objet, le main, le gestionnaire de tours et les tests.
Le main crée un ou des marchands et leurs comportements ainsi qu'un monde et fait appel au gestionnaire de tours qui demande a ces marchands de planifier puis faire des trajets(dans un ordre aléatoire) et demande au monde de produire
faireuntourgenerantmarchand demande la suppression des marchands sans le sou et

crée un marchand avec un comportement en plus de faire un tour.
Tests est censé tester divers choses pour voir si tout marche bien.