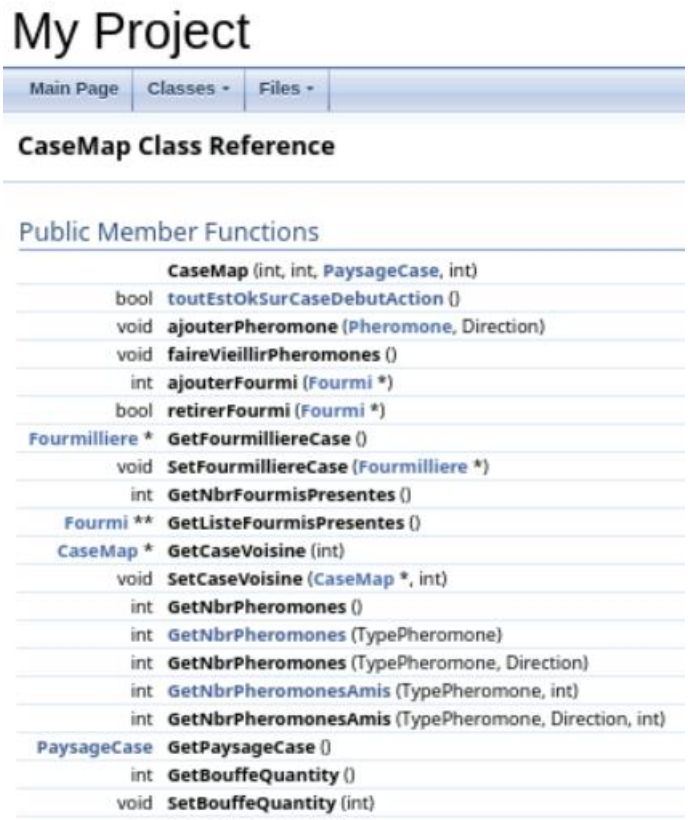
Dossier pour la conception et implémentation

Pour ce projet, nous avons décider d'utiliser Doxygen pour la documentation du code. Voici un exemple des pages générées :



My Project

Main Page | Classes | Files

CaseMap Class Reference

Public Member Functions

	CaseMap (int, int, PaysageCase, int)
bool	**toutEstOkSurCaseDebutAction** ()
void	**ajouterPheromone** (Pheromone, Direction)
void	**faireVieillirPheromones** ()
int	**ajouterFourmi** (Fourmi *)
bool	**retirerFourmi** (Fourmi *)
Fourmilliere *	**GetFourmilliereCase** ()
void	**SetFourmilliereCase** (Fourmilliere *)
int	**GetNbrFourmisPresentes** ()
Fourmi **	**GetListeFourmisPresentes** ()
CaseMap *	**GetCaseVoisine** (int)
void	**SetCaseVoisine** (CaseMap *, int)
int	**GetNbrPheromones** ()
int	**GetNbrPheromones** (TypePheromone)
int	**GetNbrPheromones** (TypePheromone, Direction)
int	**GetNbrPheromonesAmis** (TypePheromone, int)
int	**GetNbrPheromonesAmis** (TypePheromone, Direction, int)
PaysageCase	**GetPaysageCase** ()
int	**GetBouffeQuantity** ()
void	**SetBouffeQuantity** (int)

GetNbrPheromones()

int CaseMap::GetNbrPheromones (TypePheromone typeRecherched)

renvoi le nombre de phéromone du type recherché sur cette case

Parameters

TypePheromone

GetNbrPheromonesAmis()

int CaseMap::GetNbrPheromonesAmis (TypePheromone typeRecherched,
int indiceFourmilliere
)

renvoi le nombre de phéromone du type recherché sur cette case qui appartiennent a la fourmillière

Parameters

TypePheromone
int

toutEstOkSurCaseDebutAction()

bool CaseMap::toutEstOkSurCaseDebutAction ( )

appel uniquement depuis bool Fourmi::toutEstOkDebutAction() donc on part du principe que les tests de cette fonction-là ont été validés

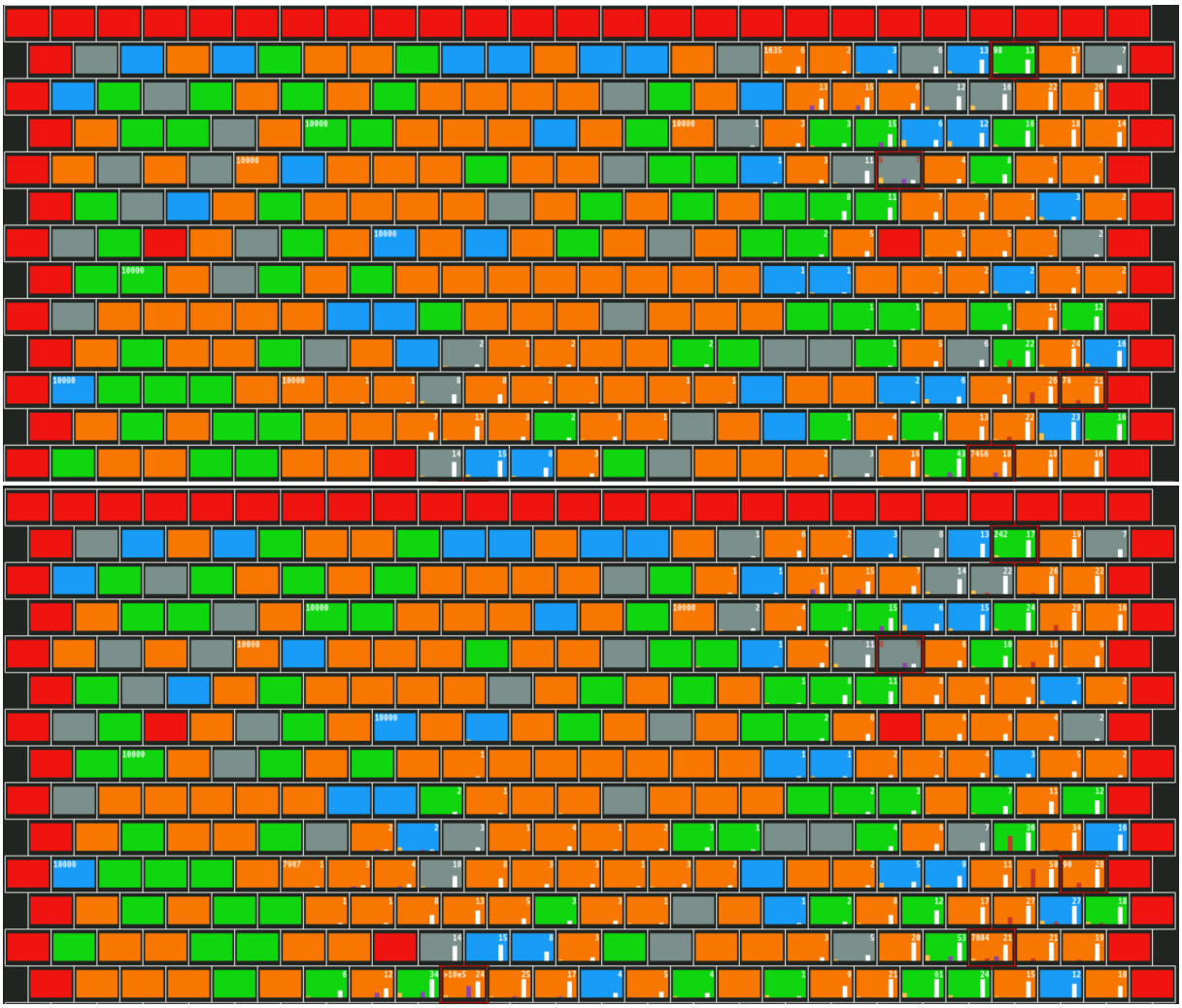
The documentation for this class was generated from the following files:

- caseMap.hpp
- caseMap.cpp

Afin de réaliser la simulation multi agent que nous avions choisi, c'est-à-dire la simulation de plusieurs fourmilières, nous avons choisi les 3 patrons suivants :

- Patron Singleton : Nous avons une classe Manager avec une instance unique qui génère le passage des tours, le contrôle de la simulation et l'affichage de la carte.
- Patron MCV : Le Manager agit en que Controler et View. La CaseMap, les Fourmis et les Fourmillères sont les modèles.
- Patron Etat : Puisque les fourmis se retrouvent dans plusieurs états, il est pertinent d'utiliser le patron état.

Voici la carte affiché par le logiciel à 2 tours différents.



On peut voir les différentes Case de la carte. Chaque type de case a une valeur de "terrain" différentes. Cette valeur représente la difficulté d'une fourmi pour traverser cette case :

- Les cases rouges sont impassables.
- Les bleus sont de l'eau, de valeur 8.
- Les grise des rocher, de valeur 4.
- Les vertes de l'herbe, de valeur 2.
- Les orange un terrain plat, de valeur 1.

En haut a droite des cases, on peut voir la quantité de nourriture présente sur cette case.

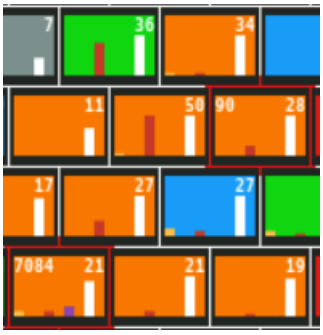
Les carte dont la bordure est rouge sont des cases sur lesquels il y a une fourmilière. Le numéro en haut a droite sur ces cases est la quantité de nourriture présent dans la fourmilière. Si se numéro atteint zéro, il devient rouge, indiquant que la fourmilière est morte de faim.

En haut a gauche de la case est le nombre de phéromones présents sur la case.

Sur chaque case nous pouvons voir 4 barres :

- La première barre en jaune indique le nombre de fourmi sur la case. Les fourmis sont stockées dans la case dans une liste. Si la liste est pleine et que l'on ne peut plus ajouter de fourmi sur cette sur case, cette barre est rouge.

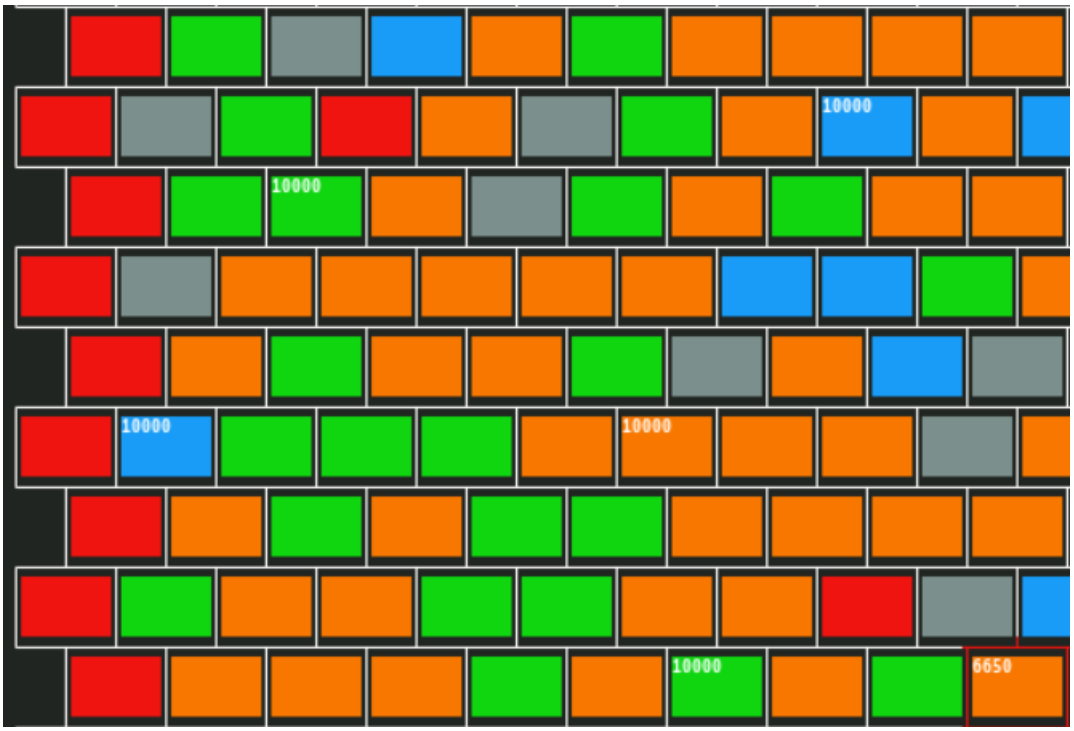
- La deuxième barre en rouge représente les phéromones d'alerte, déposé par une fourmi quand elle rencontre une fourmi d'une autre fourmilière. Par exemple, on peut voir ici beaucoup de phéromone d'alerte en raison de la proximité de deux fourmilières :

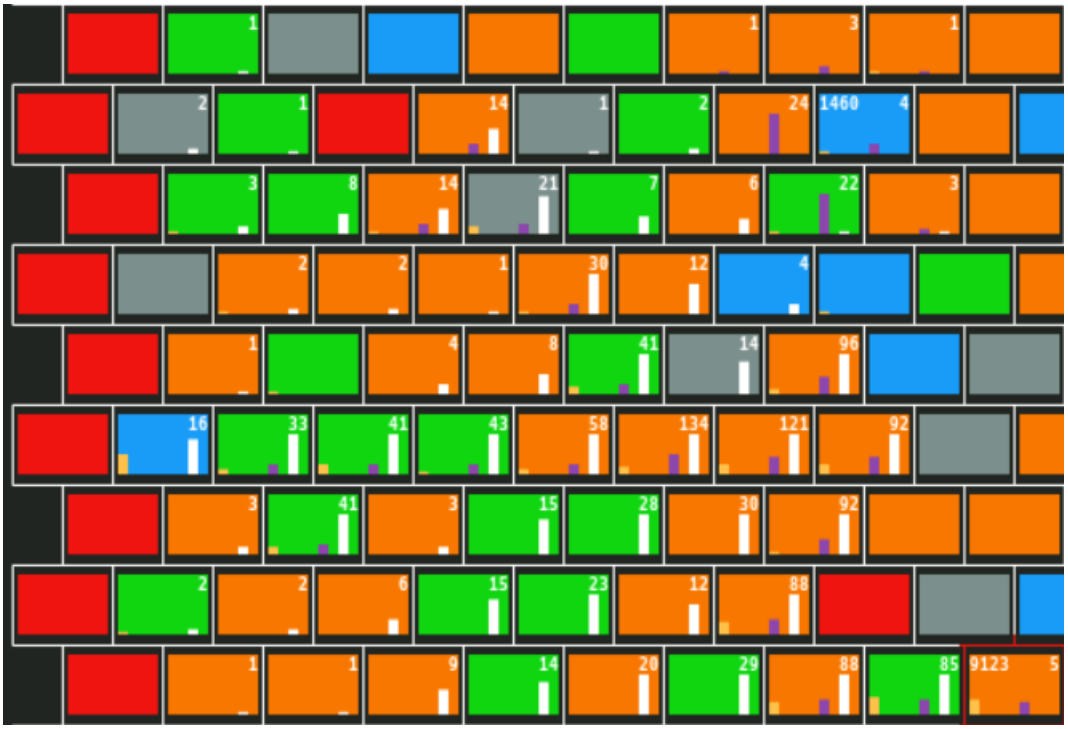


- La troisième barre en violet représente les phéromones qui indique un chemin vers de la nourriture. C'est phéromone sont déposées quand la fourmi rentre à la fourmilière avec de la nourriture.

- La quatrième barre en blanc est le chemin vers la fourmilière déposé par les fourmis lorsqu'elles explorent.

Dans les deux images qui suivent nous pouvons voir les chemins créer par les fourmis après avoir trouvées de la nourriture ainsi que les fourmis qui se déplacent dessus. En haut à droite nous pouvons également voir plusieurs fourmi qui se sont perdu car elles ne retrouvent pas de chemin vers la fourmilière, comme on peut le voir par le très grand nombre de phéromone de nourriture, ce qui indique qu'elles sont en train de tourner en rond. Cela arrive quand une fourmi explore pendant trop longtemps et que sont chemin de retour disparait en raison du vieillissement des phéromones.





Fonctionnement de la simulation

Premièrement, on génère la carte en plaçant de la nourriture sur la carte et plusieurs fourmilières de façon aléatoire grâce a Mersenne Twister.

La carte est hexagonale, avec chaque case représentée par une CaseMap. La CaseMap connait :

- les case voisines,
- le nombre de fourmi présent sur la case,
- le nombre de phéromone sur la case,
- le type de terrain,
- la présence ou non d'une fourmilière,
- la quantité de nourriture sur la case.

Chaque phéromone est un objet avec un type et un âge. Chaque phéromone a une durée de vie limité, représente par son âge atteignant 0 grâce à la méthode vieillirPheromone().

Une fois la carte générée, le Manager démarre les tours. Chaque tour a la structure suivante.

- Régénère la nourriture. Si on le désire, on peut choisir de faire apparaitre une quantité de nourriture à chaque tour, par défaut, cette quantité est 0.
- Vieilli les phéromones. On baisse l'âge de tous les phéromones de 1.
- Fait agir les fourmilières :

En effet, le Manager ne garde pas la liste des fourmis mais des fourmilières, qui elles savent les fourmis qui leur appartiennent.

Une fourmilière peut notamment :

- Savoir si une fourmi lui appartient,
- Savoir les fourmis présentes dans la fourmilières,
- Nourrir les fourmis présentes dans la fourmilières
- Récupérer la liste des fourmis qui lui sont loyales,
- Récupérer la quantité de nourriture présentes dans la fourmilière,

Lors de l'action d'une fourmilière, 2 action principale sont réalisées :

- Si possible, la reine pond de nouvelle fourmi. Cela dépend de 2 paramètres : le nombre de fourmis loyales a la fourmilière et la quantité de nourriture disponible. Pour savoir si la ponte est possible, la fourmilière prend en compte :

Food/tour * Nbfourmi + Food/tourReine + Food/ponte + Reserve.

Food/tour : la quantité de nourriture consommé par tour pour une fourmi.

Nbfourmi : le nombre de fourmi loyale.

Food/tourReine : la quantité de nourriture consommé par tour pour la reine de la fourmilière, qui représente la consommation passive de la fourmilière pour éviter qu'une fourmilière puisse survivre sans ouvrière.

Food/ponte : la quantité de nourriture consommé pour réaliser une ponte.

Cela est ensuite comparer à la quantité de nourriture présente dans la fourmilière pour déterminer si la reine peut pondre ou non.

De plus, la fourmilière mesure au Ces quantités sont variable en fonction de l'état de la fourmilière qui sont :

- Fait agir toute ses fourmis loyales.

Une fourmi connait :

- La quantité de nourriture qu'elle a, qui diminue à chaque tour,
- La quantité de nourriture qu'elle transporte,
- Ses points d'actions, qui sont régénéré à chaque tour,
- Son âge,
- L'id de sa fourmilière
- Un pointeur vers la case ou elle se trouve.
- Le nombre de tour depuis la dernière alerte

La vie d'une fourmi serait la suivante :

A sa naissance elle est dans une fourmilière.

Elle sort de la fourmilière. La quantité de nourriture qu'elle emporte avec elle dépend la l'état de la fourmilière : Famine, Ok, Parfait. Dans les états Ok et Parfait, la fourmilière garde une portion de ses fourmi à l'intérieur ainsi que plusieurs tours de réserve pour chaque fourmi. S'il y a plus de fourmi que la portion de finit que la quantité de nourriture dans la fourmilière est suffisante, alors la fourmi est envoyé à l'extérieur avec autant de nourriture que possible quand l'état est Parfait, ou bien avec seulement assez de nourriture pour une vingtaine de tour dans l'état Ok. Si l'état est Famine, alors toutes les fourmis quittent la fourmilière sans nourriture.

A chaque tour, si la fourmi est dans l'état de recherche, l'état initial, elle regarde plusieurs choses :

- S'il y a une autre fourmi sur la case
- S'il y a de la nourriture sur la case
- S'il y a une phéromone de nourriture sur la case.
- Combien de nourriture il lui reste.

S'il y a une autre fourmi sur la case, elle regarde si elle est de la même fourmilière, si c'est le cas, la fourmi qui a le plus de nourriture en donne a l'autre, de tel sorte à ce que les deux fourmis aient autant de nourriture. Si elles ne sont pas de la même fourmilière, alors la fourmi passe dans l'état alerte. Elle Communiquera avec toutes les fourmis sur sa case, s'il lui reste des points d'action et qu'elle n'a pas rencontrer d'ennemi. Chaque communication avec une autre fourmi coutera 1 point d'action.

S'il y a de la nourriture sur la case, elle en prend autant que possible. Si elle atteint sa capacité maximale, elle passe dans l'état RentrerFoumilièreavecFood, sinon elle continue de chercher.

S'il y a une phéromone de nourriture, elle passe dans l'état RécupererFood.

Quand la fourmi part, en fonction de l'état de la fourmilière, la fourmi a un niveau de nourriture minimal en dessous duquel elle doit rentrer. Par exemple, dans l'état Famine, ce niveau est 0, elle ne rentre pas. Sinon, c'est quand il lui reste une dizaine de tour de nourriture, auquel cas elle passe dans l'état RentrerFoumilière.

A chaque tour, la fourmi regénère un nombre de point d'action, ce n'est point sont consommés lors du déplacement ou de la communication. Tant qu'il lui reste des points d'action, elle peut agir, toutefois, certaine action ont un cout en point élevé et il est possible que la fourmi se retrouve en déficit de points, qui seront compensés au tour suivant, parfois il est même possible qu'elle ne puisse pas agir pendant plusieurs tours.

Lors de son déplacement, en fonction de l'état dans lequel est, la fourmi peut soit suivre des phéromones, soit choisir une case au hasard.

Chaque déplacement a un cout lié aux valeurs de terrain précédemment mentionné. Plus cette valeur est haute plus le déplacement courtera de Point d'action.

Sur chaque case il y a une liste de phéromones, chaque phéromone ayant un type et une direction. Lorsque la fourmi suit une phéromone, elle parcourt la liste des phéromones en cherchant le type de phéromones correspondant. Elle compte le nombre de phéromone pointant dans chaque direction et va dans la direction du plus grand nombre de phéromone.

Dans les états RentrerFoumilièreavecFood et Alerte, elle rentre à la fourmilière en suivant les phéromones de retour. A chaque déplacement, elle déposera une phéromone correspondant à son état : un phéromone de nourriture ou un phéromone d'alerte.

Dans l'état RécupererFood, elle suit les phéromones de nourriture et pose des phéromones de retour.

Dans l'état RentrerFoumilière, elle suit juste les phéromones de retour.

Dans l'état Recherche, elle choisit une direction au hasard en fonction de plusieurs paramètres :

- Le nombre de phéromone de retour sur la case, indiquant que des fourmis sont passées par là.
- La difficulté du terrain de la case.
- Une composante aléatoire.

La fourmi continuera ainsi jusqu'à sa mort, soit de vieillesse, soit de faim.