

MINI-PROJET DE PROGRAMMATION EN C++

La colonie de fourmis

2021/2022

Version 1.0

Kaouther Tabia

*Inspiré d'un sujet original d'A. Lebre, C. Porquet
et A. Ninassi*

kaouther.tabia@ensicaen.fr

Mini-projet à réaliser « *de préférence* » en binôme



PRESENTATION DU PROJET

Le but de ce projet est de réaliser une application C++ qui simule l'évolution d'une colonie de fourmis, son expansion et sa recherche de nourriture dans un environnement limité autour de la colonie. Nous souhaitons observer si la colonie de fourmis parvient à optimiser l'accès à des stocks de nourriture en trouvant les chemins les plus courts permettant d'y accéder. Pour cela, vous devez développer une application en C++ mettant en œuvre un maximum des concepts vus pendant le cours de Programmation en C++.

Les critères de notation sont les suivants :

- Respect des consignes sur **le travail à réaliser**, sur 8 points :
- Qualité du code fourni, sur 7 points :
 - Mise en œuvre des concepts vus en cours ;
 - Lisibilité du code : respect des consignes du cours d'ODL ;
 - Optimisation du code, les choix de conception ;
- Rapport, sur 5 points.

Livrables

Vous créerez un dépôt privé sur le serveur GitLab de l'ENSICAEN¹ auquel vous inscrirez votre enseignant. L'arborescence du projet respectera celle préconisée dans le cours ODL. Vous fournirez en particulier :

- Un rapport au format PDF de cinq pages maximum placé dans le dossier *doc*, et dans lequel vous détaillerez :
 - Vos choix de conception (diagramme de classes, etc.) ;
 - Les difficultés rencontrées et comment elles ont été résolues ;
 - Ce qui a été réalisé, et ce qui reste à faire, en indiquant avec précision les parties réalisées par chacun, les aides extérieures dont vous avez bénéficié (sites, personnes, etc.), les informations ou les codes que vous avez intégrés au vôtre ;
 - Les limitations et/ou bugs connus.
- Un fichier README.md présent à la racine, et décrivant succinctement le projet ainsi que les commandes à effectuer pour installer et exécuter le programme ;
- Un fichier concepts.md présent à la racine et qui contient un tableau tel que celui de la page 9, où vous cochez, au fur et à mesure de l'avancement, les différents points du cours mis en œuvre dans votre projet (cf. les sections du cours correspondantes) ;
- Le fichier Makefile (ou CMakeLists.txt si vous utilisez CLion) ;
- Les dossiers *include* et *src* comportant le(s) différent(s) fichier(s) source(s) .cpp et .hpp.

L'originalité de votre travail sera évaluée par rapport aux autres projets rendus. La qualité structurelle et la lisibilité du code, ainsi que les fonctionnalités permettront de classer et de noter votre projet. Enfin, on considérera qu'au moins 70 % des concepts vus en cours devront avoir été mis en œuvre dans ce projet.

¹ Le serveur GitLab de l'ENSICAEN est accessible à l'adresse : <https://gitlab.ecole.ensicaen.fr>

TRAVAIL A REALISER

1. Description des éléments de la colonie

1.1. Types de fourmis

Les fourmis rencontrées dans cette simulation sont une reine, des ouvrières, des éclaireurs, des soldats, ainsi qu'une fourmi ennemie et extérieure à la colonie, appelée « fourmi esclavagiste ».

1.1.1. La reine

La reine est la mère de la colonie et de toutes les fourmis, à l'exception des fourmis esclavagistes. La reine vit au centre de la colonie, où elle habite. La reine peut mourir de faim ou de vieillesse. Lorsque la reine meurt, la simulation se termine. La reine produit deux nouvelles fourmis chaque jour, sauf si elle est agressée par une fourmi esclavagiste, auquel cas aucune nouvelle fourmi ne sera générée le jour suivant l'agression. Une nouvelle fourmi produite par la reine peut être une ouvrière (probabilité : 80 %), un soldat (probabilité : 15 %), ou un éclaireur (probabilité : 5 %). La toute première fourmi à qui elle donne naissance est obligatoirement un éclaireur.

1.1.2. Les ouvrières

Durant les quinze premiers jours de leur vie, les ouvrières sont appelées « mineures » et se consacrent au maintien et à l'expansion de la colonie. Elles deviennent ensuite « majeures » et partent à la recherche de nourriture. Une fois qu'elles ont trouvé de la nourriture sur une case, elles la rapportent dans la colonie, laissant derrière elles une traînée de phéromones. D'autres ouvrières pourront alors suivre ces traces de phéromones, transportant de la nourriture jusqu'à épuisement du stock. Si une ouvrière « majeure » meurt en transportant de la nourriture, elle la laisse tomber sur la case où elle meurt. Une ouvrière « majeure » peut transporter une unité de nourriture à chaque voyage.

1.1.3. Les éclaireurs

Les éclaireurs sont des ouvrières spécialisées qui ne restent « mineures » que les deux premiers jours de leur vie. Elles explorent alors de nouvelles cases autour de la colonie en se déplaçant au hasard. Les soldats et les ouvrières ne peuvent pas pénétrer dans un territoire inexploré à moins qu'un éclaireur ne l'ait préalablement exploré.

1.1.4. Les soldats

Le travail des fourmis soldats est de protéger la reine, ainsi que la zone couverte par la colonie. Chaque soldat peut se déplacer au hasard de case en case dans la zone connue jusqu'à rencontrer une fourmi ennemie esclavagiste qu'elle va alors essayer de tuer². Enfin, tous les cent jours, tous les soldats ressentent le besoin de repasser par le centre de la colonie.

² Dans le cas où plusieurs fourmis esclavagistes se trouveraient sur la même case, la fourmi soldat ne pourrait essayer d'en tuer qu'une à chaque tour

1.1.5. Les fourmis esclavagistes

Les fourmis esclavagistes sont l'ennemi de la colonie. Elles ont comme but de voler les larves de la colonie puis de les emmener en esclavage dans leur propre colonie située hors du terrain de simulation. Les fourmis entrent par les quatre coins du terrain.

1.2. Règles concernant les fourmis

À l'exception de la reine qui vit dix ans, toutes les fourmis ont une durée de vie d'un an.

1.2.1. Alimentation

La reine doit, pour survivre, consommer chaque jour 1 % d'une unité de nourriture. Les autres fourmis de la colonie se contentent, quant à elles, de 0.1 % d'une unité de nourriture. Cette nourriture est issue du stock géré par la colonie en son sein. Nous considérerons que les fourmis prélèvent leur part de nourriture sur le stock de la colonie sans qu'elles ne soient obligées d'y retourner chaque jour.

Quant aux fourmis esclavagistes, elles peuvent survivre jusqu'à 10 jours sans nourriture et se contentent de 0.1 % d'une unité de nourriture, que cette dernière soit prélevée sur un stock de nourriture du terrain, ou encore dérobée dans le stock de la colonie si la fourmi se trouve dans cette dernière.

1.2.2. Règles de déplacement

Toute fourmi possède une « mémoire » de l'emplacement de leur colonie et du chemin suivi depuis cette dernière.

Les déplacements peuvent s'effectuer sur l'une des huit cases voisines de la case sur laquelle se trouve toute fourmi (on parlera de 8-voisinage).

1.2.3. Cas du déplacement des ouvrières

Dans le cas des ouvrières « majeures », le déplacement est de plus lié aux phéromones que sont capables de produire les fourmis.

Tout au long de leur vie, les ouvrières « majeures » réalisent les opérations suivantes : chercher la nourriture dans une direction donnée, prendre la nourriture et faire demi-tour, se rediriger vers le centre de la colonie en déposant de la phéromone sur chacune des cases du chemin de retour, déposer la nourriture dans la colonie et reconstituer sa réserve de phéromones, puis se remettre en recherche.

Lors de la recherche de nourriture, les ouvrières se déplacent de case en case, en fonction du niveau de phéromones présent sur les cases voisines, et elles auront tendance à choisir celle qui possède le niveau le plus élevé. De plus, elles mémorisent les cases traversées à l'aller afin de faciliter leur retour dans la colonie. Enfin, les ouvrières auront comme contrainte supplémentaire de chercher à continuer dans la même direction (en s'éloignant le plus possible de la colonie) si un chemin « phéromonné » n'aboutit à rien (cas d'un chemin menant un stock récemment vidé par exemple).

Lors du retour vers le centre de la colonie, les ouvrières déposent une dose de phéromones, de plus en plus petite (mais non nulle), sur les cases sur lesquelles elles passent.

On considérera qu'à mesure que le temps passe, les niveaux de phéromones dans chaque case vont se dégrader par évaporation et diffusion sur les cases voisines.

1.3. Terrain de simulation

L'espace sur lequel se déroule la simulation est une matrice rectangulaire composée d'un nombre impair de cases sur sa largeur et sa hauteur (voir l'exemple de terrain sur la Figure 1 **Erreur ! Source du renvoi introuvable.**).


















































	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6							
7							

Figure 1. Exemple de terrain 7×7 comprenant la colonie au centre, des obstacles et de la nourriture. Les cases explorées apparaissent sur fond ondulé clair.

1.3.1. Les types de cases

Une case peut être : une case vierge ; une case contenant des unités de nourriture ; une partie de la colonie ; un obstacle devant être contourné.

La capacité maximale d'une case en nombre d'individus, hormis si la case contient un obstacle, est de 12 fourmis à chaque tour. Toutefois, les cases de la colonie elle-même peuvent accueillir jusqu'à 100 individus à la fois.

Pour la constitution des blocs de cases d'un même type (nourriture, colonie et obstacle), le voisinage entre les cases d'un même blocs sera du 4-voisinage, c'est à dire que deux cases seront considérées comme « voisines », si elles possèdent au moins 1 côté en commun.

1.3.2. La colonie

La colonie occupe au départ la case centrale de la grille. L'expansion de la colonie sur les cases voisines est liée au nombre d'individus de la colonie et non pas au nombre d'individus qui peuvent se trouver sur une case à un instant donné. On considérera qu'une case représente 100 individus.

1.3.3. Les obstacles

Les obstacles représentent 30 % de la surface totale et sont placés aléatoirement. 50 % de ces obstacles sont des obstacles d'une case, 30 % des obstacles de deux cases voisines, 10 % des obstacles de trois cases voisines, 5 % des obstacles de quatre cases voisines, 4 % des obstacles de cinq cases voisines et 1 % des obstacles de six cases voisines.

1.3.4. La nourriture

Des stocks de 10 unités de nourriture recouvrant chacun une case et isolés les uns des autres sont placés aléatoirement sur 0.02 % de la surface totale du terrain.

De plus, deux stocks de 100 000 unités et recouvrant uniformément cinq cases voisines pour chacun d'entre eux, forment des blocs compacts placés à proximité de la bordure du terrain.

1.3.5. Combat entre soldat et esclavagiste

Si une fourmi esclavagiste est tuée par un soldat en retournant dans sa colonie après avoir capturé deux larves, les deux larves sont réintégrées à la colonie afin de devenir deux fourmis le jour suivant (il y aura donc quatre fourmis produites le jour suivant).

2. Implémentation

2.1. Initialisation

Le terrain est de dimension 211×201. Il comporte la colonie et la reine en son centre. La quantité de nourriture disponible initialement dans la colonie est de 2 unités de nourriture.

Le nombre de tours total de la simulation correspond à la durée de vie de la reine. Un tour correspondra à 1 h. Vous utiliserez un générateur aléatoire pour placer les obstacles, les petites quantités de nourriture, ainsi que les gros stocks de nourriture.

Concernant le niveau de phéromones sur une case, on considérera que celui-ci sature à 1000. De même, on choisira un taux d'évaporation de 5 % et un taux de diffusion de 0.2 %.

La réserve de phéromones de chaque ouvrière « majeure » est fixée à 500.

On supposera qu'une fourmi esclavagiste entre aléatoirement tous les cinq à dix jours sur le terrain.

D'une façon générale, l'ensemble des paramètres devra pouvoir être facilement modifiable.

2.2. Simulateur

À chaque tour, le programme met à jour le fichier evolution.txt qui sera utilisé par le logiciel Gnuplot³ afin de visualiser l'évolution d'un certain nombre de paramètres, tels que le nombre total d'individus de la colonie, le nombre d'individus de chaque type, le nombre d'unités de

³ Gnuplot : <https://sourceforge.net/projects/gnuplot/>

nourriture dans le stock de la colonie, etc. (vous essaieriez de suivre le maximum de paramètres).

Il doit aussi être possible de suivre la simulation en réalisant une interface utilisateur textuelle, ou bien mieux, graphique. Ainsi, il sera particulièrement intéressant de suivre l'évolution des fourmis, de la nourriture et des niveaux de phéromones de chaque case de manière à tracer les chemins les plus empruntés au fil du temps.

L'implémentation de l'interface graphique peut être réalisée en langage C++ avec la bibliothèque SFML (Simple and Fast and Multimedia Library) disponible à l'adresse : <https://www.sfml-dev.org>.

CALENDRIER

La version définitive de votre projet est à rendre pour le **dimanche 19/12/2021 à 23h59** au plus tard. Le projet sera récupéré directement sur votre dépôt GitLab et vous n'avez donc pas d'action particulière à effectuer pour le rendre, mais vous devez vous assurer que les fichiers requis sont bien présents dans la branche master. Les groupes dont le projet ne pourra pas être récupéré correctement seront sanctionnés.

Concept	Mis en œuvre
Surcharge des fonctions	<input type="checkbox"/>
Entrées / Sorties	<input type="checkbox"/>
Références	<input type="checkbox"/>
Constance	<input type="checkbox"/>
Opérateur de résolution de portée	<input type="checkbox"/>
Espaces de noms	<input type="checkbox"/>
Fichiers d'en-têtes et espaces de noms	<input type="checkbox"/>
Portée	<input type="checkbox"/>
Gestion mémoire : Allocation	<input type="checkbox"/>
Gestion mémoire : Libération	<input type="checkbox"/>
Classes et objets	<input type="checkbox"/>
Membres & partage	<input type="checkbox"/>
Accès aux Membres & Portée	<input type="checkbox"/>
Membres & Protection	<input type="checkbox"/>
Fonction membre : accès & pointeur this	<input type="checkbox"/>
Membres amis : friend	<input type="checkbox"/>
Constructeur	<input type="checkbox"/>
Destructeur	<input type="checkbox"/>
Surcharge d'opérateurs	<input type="checkbox"/>
Conversion utilisateur	<input type="checkbox"/>
Fonctions amies versus fonctions membres	<input type="checkbox"/>
Fonctions inline	<input type="checkbox"/>
Constructeur, initialisation, affectation	<input type="checkbox"/>
Constructeur, initialisation, affectation et allocation dynamique	<input type="checkbox"/>
Classes dérivées – Héritage simple	<input type="checkbox"/>
Héritage et constructeur	<input type="checkbox"/>
Conversion classes de base <-> classes dérivées	<input type="checkbox"/>
Dérivation et protection	<input type="checkbox"/>
Vers le polymorphisme	<input type="checkbox"/>
Liaison retardée, fonctions virtuelles	<input type="checkbox"/>
Classes abstraites et fonctions virtuelles pures	<input type="checkbox"/>
Type des objets et conversion explicite	<input type="checkbox"/>
Héritage multiple	<input type="checkbox"/>
Généricité : Patrons de fonction	<input type="checkbox"/>
Généricité : Patrons de classe	<input type="checkbox"/>
Flots, Entrées /Sorties, Fichiers	<input type="checkbox"/>
La Standard Template Library (STL)	<input type="checkbox"/>
Programmation défensive – Assertions	<input type="checkbox"/>
Programmation défensive – Exceptions	<input type="checkbox"/>

Tableau 1 Concepts à mettre en œuvre (voir les sections du cours correspondantes pour plus d'informations)



Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04

