

AntAlgorithm

Cahier des charges fonctionnel

Rédacteurs :

Emmanuel LION
Océane MONGES

Date : 8 janv 2024 - Version V01

Visa :

SOMMAIRE

SOMMAIRE.....	1
TABLES DES MODIFICATIONS.....	2
1. INTRODUCTION.....	2
1.1. Contexte.....	2
1.2. Objectifs du projet.....	2
2. SPECIFICATIONS TECHNIQUES.....	2
2.1. Technologies utilisées.....	2
2.2. Contraintes techniques.....	3
3. FONCTIONNALITÉS PRINCIPALES.....	3
3.1. Environnement de simulation.....	3
3.2. Système de fourmis.....	3
3.2.1. Principes.....	3
3.2.2. Algorithme de colonies de fourmis.....	4
3.3. Système de phéromones.....	5
3.4. Contrôles de Simulation.....	5
4. INTERFACE UTILISATEUR.....	5
4.1. Éléments d'Interface.....	5
4.2. Représentation Graphique.....	6
5. DESCRIPTION DU LIVRABLE.....	6
6. PLANNING ET JALONS.....	6
6.1. Dates Clés.....	6

TABLES DES MODIFICATIONS

Révision	Date	Page ou §	Objet de la révision
V1	08 janv 2024	Toutes	

1. INTRODUCTION

Ce document présente le cahier des charges fonctionnel pour le projet "AntAlgorithm". L'objectif est de développer un système multi-agents en JavaScript, HTML et CSS, suivant une architecture MVC, capable de rechercher les chemins les plus courts pour atteindre des objectifs donnés, en s'inspirant de l'algorithme de colonies de fourmis.

1.1. Contexte

Dans le cadre d'un projet académique, une équipe de deux personnes développe un système multi-agents simulant une colonie de fourmis, en s'appuyant sur l'algorithme d'optimisation par colonie de fourmis (Ant Colony Optimization, ACO). Cet algorithme, inspiré du comportement naturel des fourmis, permet de résoudre des problèmes complexes d'optimisation en exploitant des mécanismes tels que le dépôt et l'évaporation de phéromones pour identifier des chemins optimaux.

Le projet vise à modéliser ces interactions pour aboutir à une solution efficace et collaborative, reflétant les dynamiques observées dans les colonies de fourmis réelles.

1.2. Objectifs du projet

- Développer un système multi-agents permettant à des agents (fourmis) de trouver et d'apporter des ressources (nourriture) à leur point de départ (colonie).
- Implémenter une simulation où les agents interagissent avec un environnement représenté par une grille composée de différents types de cellules.
- Utiliser l'algorithme de colonies de fourmis pour optimiser les trajets des agents en se basant sur des traces de phéromones.

2. SPECIFICATIONS TECHNIQUES

2.1. Technologies utilisées

Le projet repose sur une combinaison de technologies modernes pour le développement web. Pour le frontend, HTML5 sera utilisé pour structurer la page, avec un accent particulier sur l'élément <canvas> pour la représentation graphique de l'environnement et des agents. CSS3 permettra de styliser l'interface,

assurant une présentation visuelle claire et intuitive. JavaScript Vanilla sera employé pour la logique et les interactions, exploitant ses capacités en programmation orientée objet pour modéliser les agents et leur environnement. Le développement suivra une architecture MVC (Modèle-Vue-Contrôleur), garantissant une séparation claire entre la gestion des données, la logique applicative et l'interface utilisateur. Un serveur local sera utilisé pour le développement, permettant de tester les fonctionnalités et de profiter des avantages des modules JavaScript modernes.

2.2. Contraintes techniques

Les contraintes suivantes devront être respectées:

- Pas de frameworks JavaScript autorisés
- Compatible avec les navigateurs modernes
- Responsive design non requis

3. FONCTIONNALITÉS PRINCIPALES

3.1. Environnement de simulation

Le projet comprend une grille interactive composée de cellules, représentant l'environnement dans lequel évoluent les agents. Les cellules se divisent en plusieurs types :

Point de départ	Une seule cellule initiale d'où les agents commencent leur exploration.
Obstacles	Des cellules bloquant le déplacement des agents, en nombre variable selon la configuration.
Objectifs	Au moins une cellule contenant des ressources à collecter.
Cellules libres	Cellules vides en nombre variable, pouvant stocker des informations sur les phéromones.

3.2. Système de fourmis

3.2.1. Principes

Les agents, assimilés à des fourmis, possèdent les caractéristiques suivantes :

Capacité de transport	Chaque fourmi peut transporter jusqu'à 0.5 unité de ressource à la fois.
Vision limitée	Elles ne perçoivent que les cellules adjacentes sans inclure les diagonales.
Mémoire du trajet	Elles mémorisent leur chemin actuel jusqu'à leur retour à la colonie.

Les fourmis adoptent plusieurs comportements :

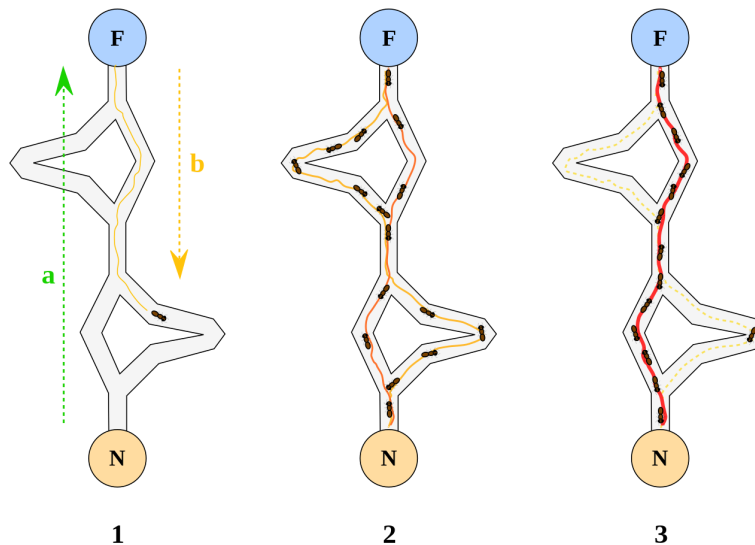
Exploration autonome pour rechercher des ressources.
--

Retour à la colonie lorsqu'elles trouvent de la nourriture.

Dépôt de phéromones pour marquer leur trajet et guider les autres agents.

3.2.2. Algorithme de colonies de fourmis

Ce projet est inspiré de l'algorithme de colonies de fourmis qui est une méthode d'optimisation elle-même inspirée par le comportement des fourmis réelles lorsqu'elles cherchent des chemins optimaux entre leur colonie et une source de nourriture. Cet algorithme simule le comportement des fourmis en utilisant des agents virtuels que nous pouvons appeler "fourmis". Ces dernières se déplacent à travers un espace de solutions potentielles et lorsque l'une d'entre elles en trouve une, elle laisse derrière elle une trace chimique appelée "phéromone". Ces phéromones servent de communication indirecte entre les fourmis, indiquant la qualité des solutions trouvées. Ainsi, les fourmis explorent différentes solutions tout en favorisant les chemins contenant davantage de phéromones. Cela permet un équilibre entre l'exploration de nouvelles possibilités et l'exploitation des solutions connues. Les phéromones s'évaporent avec le temps, simulant le phénomène naturel où les traces chimiques laissées par les fourmis disparaissent progressivement.



https://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis#/media/Fichier:Aco_branches.svg

Dans le cadre de ce projet, nous pouvons résumer l'algorithme de colonies de fourmis par 3 équations. La première permet de calculer la probabilité associée aux différents déplacements possibles par une fourmi à une itération donnée t .

$$p_{ij}^k(t) = \begin{cases} \frac{\gamma + \tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in J_i^k} (\gamma + \tau_{il}(t)^\alpha \cdot \eta_{il}^\beta)} & \text{si } j \in J_i^k \\ 0 & \text{si } j \notin J_i^k \end{cases}$$

où J_i^k est la liste des déplacements possibles pour une fourmi k lorsqu'elle se trouve à une position i (dans notre cas, si elles existent, les cellules au-dessus, en dessous et sur les côtés de la position actuelle de la fourmi), η_{ij} la visibilité, qui est égale à l'inverse de la distance de deux positions i et j ($1/d_{ij}$) et $\tau_{ij}(t)$ l'intensité de la piste à une itération donnée t . Les deux principaux paramètres contrôlant l'algorithme sont α et β , qui contrôlent l'importance relative de l'intensité et de la visibilité d'une arête. Enfin, pour permettre aux fourmis de continuer à explorer des pistes non découvertes, on attribue une probabilité non nulle d'exploration des positions « inconnues », contrôlée par le paramètre γ .

De par la contrainte liée au voisinage visible par une fourmi (1 case autour d'elle), le paramètre de visibilité n'est pas exploité (ou alors est égal à 1/1 pour toutes les cases). Nous pourrions l'utiliser si nous ajoutons un système de difficulté associée au déplacement par rapport à la nature des tuiles. Par exemple, une fourmi se déplace plus facilement sur la terre que dans l'herbe.

La seconde équation permet, une fois la fourmi retournée à son point de départ pour déposer la nourriture, de calculer la quantité de phéromone $\Delta\tau_{ij}^k$ à déposer sur l'ensemble des cellules qui compose le trajet emprunté par la fourmi entre la nourriture et son point de départ. Cette quantité vient s'additionner à la quantité déjà présente sur la cellule.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases}$$

où $T^k(t)$ est le trajet emprunté entre la nourriture et la colonie (son point de départ) par la fourmi k à l'itération t , $L^k(t)$ la longueur du trajet et Q un paramètre de réglage.

La fourmi est capable de choisir le chemin le plus court à emprunter pour revenir à sa colonie en tenant compte uniquement des cellules qu'elle a explorées durant son trajet actuel. Une fois arrivée à la colonie, la mémorisation des cellules empruntées durant son dernier trajet est réinitialisée.

Enfin, la troisième équation permet à une fréquence donnée de simuler le phénomène d'évaporation des phéromones : $\rho\tau_{ij}(t)$ avec ρ un paramètre de réglage.

3.3. Système de phéromones

Le système repose sur l'implémentation des équations de l'algorithme ACO (Ant Colony Optimization) :

Calcul des probabilités de déplacement	Calcul des probabilités de déplacement pour guider les agents en fonction de la concentration de phéromones.
Évaporation des phéromones	Évaporation des phéromones, permettant de simuler leur dissipation naturelle.
Visualisation graphique	Visualisation graphique des phéromones sous forme de gradients, indiquant les zones les plus fréquentées.

3.4. Contrôles de Simulation

Les utilisateurs disposent de plusieurs fonctionnalités pour interagir avec la simulation :

- Démarrage et mise en pause de la simulation.
- Réinitialisation complète pour relancer une session.
- Affichage d'informations en temps réel sur l'état des agents, des ressources, et des phéromones.

4. INTERFACE UTILISATEUR

4.1. Éléments d'Interface

L'interface comprend plusieurs composants interactifs :

Grille de simulation	Une grille de simulation pour visualiser l'environnement et les agents.
Contrôles de simulation	Des contrôles de simulation pour démarrer, mettre en pause, ou réinitialiser la simulation.
Indicateurs visuels	Des Indicateurs visuels pour suivre : <ul style="list-style-type: none"> - Le niveau de phéromones sur les cellules. - La position des fourmis en temps réel. - L'état des ressources présentes sur les cellules objectives.

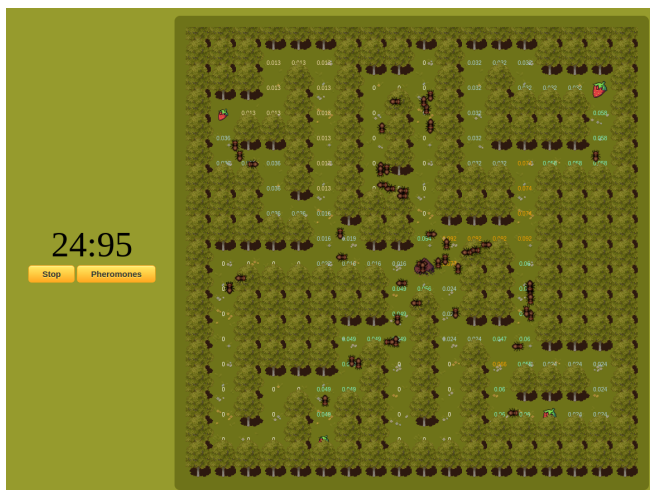
4.2. Représentation Graphique

L'aspect visuel du projet met en avant une distinction claire des éléments :

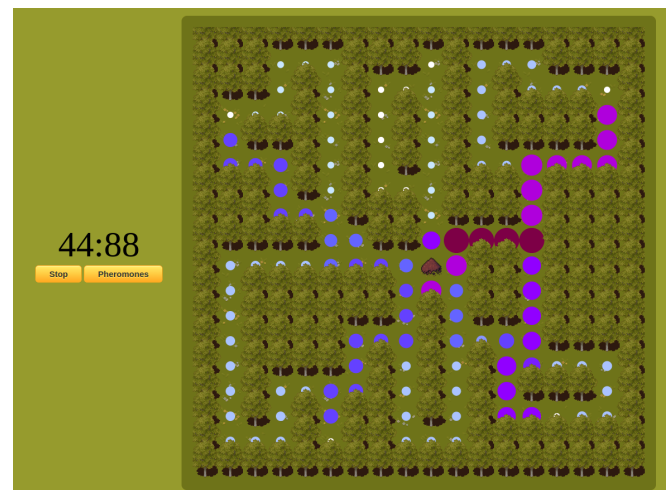
Les types de cellules sont représentés par des couleurs ou des symboles distincts.
Les phéromones sont visualisées à travers un gradient de couleur, indiquant leur intensité.
Les déplacements des fourmis sont animés de manière fluide pour assurer une simulation réaliste et agréable.

5. DESCRIPTION DU LIVRABLE

Le livrable final inclura une simulation interactive d'un système multi-agents, avec les caractéristiques suivantes :



Vue avec pheromones textuels



Vue avec pheromones visuels

6. PLANNING ET JALONS

6.1. Dates Clés

Date de fin : 02/02/2024

Durée de la présentation : 15 minutes (10 + 5)