

# Présentation du projet : La fourmi de Langton

*Calixte Lamotte*



Dans le cas de l'image : L'Autoroute (spirale infinie) est atteinte à partir des 7000 pas.

Nom de votre projet	La Fourmi de Langton (améliorée)
Membre de l'équipe n°1 (prénom/nom)	Calixte Lamotte

## 1 / PRÉSENTATION GÉNÉRALE - [LIEN VIDEO](#)

Le projet "*Fourmi de Langton Améliorée*" est une implémentation interactive et enrichie du célèbre automate cellulaire inventé par Christopher Langton en 1986. Cette simulation, développée par *Calixte Lamotte*, transforme le concept mathématique original en une expérience utilisateur immersive et pédagogique.

La Fourmi de Langton est un automate cellulaire bidimensionnel qui suit des règles simples mais génère des comportements complexes :

- **Règle 1** : Si la fourmi est sur une case blanche, elle tourne à droite, change la case en noir, et avance
- **Règle 2** : Si la fourmi est sur une case noire, elle tourne à gauche, change la case en blanc, et avance

Malgré ces règles élémentaires, la fourmi produit des motifs fascinants qui évoluent de manière chaotique avant de finalement converger vers un comportement cyclique appelé "autoroute", comportement pour lequel aucun contre-exemple n'a été trouvé.

## 2 / ORGANISATION DU TRAVAIL

Ce projet a été entièrement développé en autonomie par *Calixte Lamotte* dans le cadre de ses études en **NSI** (Numérique et Sciences Informatiques). L'environnement de développement *Visual Studio Code* a été utilisé pour l'écriture et le débogage du code ; La gestion de versions a été assurée par *GitHub*, permettant un suivi rigoureux des modifications, et la sauvegarde sécurisée du code.

## 3 / ÉTAPES DU PROJET

### CAHIER DE BORD - FOURMI DE LANGTON AMÉLIORÉE

---

#### *SÉANCES SCOLAIRES – 11h*

---

#### **Mercredi 7 mai 2024 - (2h)**

**Objectif** : Initialisation du projet et mise en place des fondations

- Recherche documentaire sur la Fourmi de Langton
- Étude du concept mathématique original
- Analyse des règles de base

- Consultation de vidéos explicatives

### **Architecture et planification**

- Création du repository GitHub
- Définition de la structure du projet
- Rédaction des spécifications techniques initiales
- Choix des technologies (Pygame + NumPy)

### **Implémentation de base**

- Création de la classe `LangtonAnt` avec les méthodes fondamentales
- Implémentation de l'algorithme de base (step, movement)
- Tests unitaires de la logique de déplacement

### **Mercredi 14 mai 2024 - (1h)**

**Objectif :** Interface graphique de base

### **Configuration de Pygame**

- Installation et configuration de l'environnement Pygame
- Création de la fenêtre principale et boucle de jeu
- Définition des constantes de base (couleurs, dimensions)

### **Rendu initial**

- Implémentation du rendu de la grille
- Affichage de la fourmi et des cellules actives
- Test de performance avec grille 200x200

### **Premiers contrôles**

- Ajout des contrôles clavier de base (SPACE pour pause, R pour reset)

### **Mercredi 21 mai 2024 - (2h)**

**Objectif :** Système de navigation et interface utilisateur

### **Implémentation du zoom et pan**

- Développement du système de coordonnées écran/grille
- Fonction `zoom\_at\_point()` pour zoom centré sur le curseur
- Gestion des événements de molette de souris
- Tests de performance avec différents niveaux de zoom

### **Interface utilisateur avancée**

- Création du système de boutons avec `draw\_button()`
- Implémentation des boutons EXIT, INFO, MENU
- Système de détection de survol avec effets visuels
- Gestion des états d'interface (menu ouvert/fermé)

### **Lundi 26 mai 2024 - (2h)**

**Objectif :** Recherche et travail sur le son et intégration des MP3

#### **Recherche audio**

- Recherche de sons appropriés pour les transitions
- Sélection de fichiers MP3 : sons de cloche pour différencier blanc→noir et noir→blanc
- Étude de pygame.mixer pour l'intégration audio

#### **Implémentation du système audio**

- Configuration de `pygame.mixer.init()`
- Chargement des fichiers audio (start.mp3, 1-bell.mp3, 2-bell.mp3)
- Intégration dans les fonctions `step()` et `modify\_cell()`
- Tests de synchronisation audio/visuel

#### **Système de sentier (trail)**

- Développement du suivi de trajectoire de la fourmi
- Implémentation de l'affichage du chemin parcouru
- Bouton TRAIL avec activation/désactivation

### **Mercredi 28 mai 2024 - (2h)**

**Objectif :** Travail personnel sur les fonctionnalités avancées

#### **Planification des fonctionnalités**

- Analyse des besoins utilisateur
- Conception du système de vitesse variable
- Planification du mode modification de pixels

#### **Système de vitesse fractionnaire**

- Développement du champ de saisie de vitesse
- Tests avec différentes valeurs (0.1, 0.5, 2.0, etc.)

#### **Mode modification interactif**

- Développement du mode `MODIFY\_MODE`
- Système de modification continue avec glisser-déposer

### **Lundi 2 juin 2024 - (2h)**

**Objectif :** Recherches pour dernières fonctionnalités et peaufinage

**Objectif :** Fonctionnalités d'import/export

#### **Conception du système de sauvegarde**

- Recherche sur les formats de compression (zlib, base64)
- Conception de la structure des données exportées

- Planification de l'interface utilisateur pour import/export

### **Implémentation export**

- Développement de ``export_design()`` avec compression zlib
- Système d'encodage base64 pour le partage
- Interface de champ de texte avec copie automatique

### **Implémentation import**

- Développement de ``import_design()`` avec validation
- Gestion des erreurs et vérification de format
- Support du collage Ctrl+V

### **Recherche UX/UI**

- Planification des dernières améliorations visuelles

### **Fonctionnalités finales**

- Implémentation du bouton multiplicateur de vitesse  $\times 6$
- Développement du compteur de pas centralisé
- Ajout du mode information avec image et lien vidéo
- Intégration du titre et textes informatifs

### **Peaufinage et tests finaux**

- Tests complets de toutes les fonctionnalités
- Optimisation finale des performances

---

## ***TRAVAIL PERSONNEL – 25h***

---

*En plus des tâches listés ci-dessous, la majorité des 25h ont été utilisées afin de d'améliorer et compléter les taches scolaires incomplètes ou bugées listés plus haut.*

### **Expérimentation algorithmique**

- Tests de différentes tailles de grille
- Observation des patterns émergents

### **Refactoring du code**

- Réorganisation des constantes pour plus de clarté
- Amélioration de la lisibilité du code
- Ajout de commentaires détaillés

### **Système de gestion d'événements**

- Refonte complète de ``handle_events()``

- Séparation logique des différents types d'interactions
- Amélioration de la responsivité de l'interface

### **Tests et validation**

- Tests de stress avec simulations longues
- Validation de la mémoire et des performances

## **4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ**

### **Fonctionnalités principales**

- Interface utilisateur intuitive
- **Navigation fluide** : Zoom et panoramique avec la souris pour explorer la grille en détail
- **Contrôles interactifs** : Boutons pour pause/lecture, réinitialisation, et accès aux options
- **Affichage en temps réel** : Compteur de pas et informations sur l'état de la simulation
- **Outils de simulation avancés**
- **Contrôle de vitesse** : Ajustement précis de la vitesse de simulation (support des valeurs fractionnaires)
- **Multiplicateur de vitesse** : Bouton ×6 pour accélération temporaire
- **Mode modification** : Édition manuelle des cellules avec mise en surbrillance
- **Traçage du sentier** : Visualisation du chemin parcouru par la fourmi

### **Fonctionnalités d'import/export**

- **Sauvegarde de designs** : Export des configurations sous forme de code compressé
- **Partage et restauration** : Import de designs via copier-coller
- **Compression intelligente** : Utilisation de *zlib* et *base64* pour optimiser l'espace

### **Expérience multimédia**

- **Effets sonores** : Sons distincts pour les transitions blanc→noir et noir→blanc
- **Design moderne** : Couleurs cohérentes
- **Mode information** : Écran dédié avec lien vers ressource éducative

### **Valeur pédagogique**

*Ce projet illustre plusieurs concepts informatiques et mathématiques :*

- **Automates cellulaires** : Démonstration de l'émergence de complexité à partir de règles simples
- **Programmation orientée objet** : Architecture claire avec séparation des responsabilités
- **Interfaces utilisateur** : Conception d'une expérience utilisateur intuitive
- **Algorithmes de compression** : Implémentation pratique de techniques d'optimisation des données
- **Programmation temps réel** : Gestion des événements et boucles de jeu

### Points forts du projet

- **Accessibilité** : Contrôles intuitifs
- **Flexibilité** : Possibilité de modifier la simulation en temps réel
- **Performance** : Optimisations pour gérer des grilles de 200×200 cellules à 60 FPS
- **Extensibilité** : Architecture modulaire permettant l'ajout de nouvelles fonctionnalités
- **Documentation** : Code bien commenté et structure claire

### Public cible

- **Étudiants** : Découverte des automates cellulaires et de leurs propriétés
- **Enseignants** : Outil de démonstration pour cours de mathématiques/informatique
- **Développeurs** : Exemple d'implémentation d'interface graphique interactive
- **Curieux** : Exploration ludique d'un phénomène mathématique fascinant

## 5 / OUVERTURE

Ce projet de Fourmi de Langton Améliorée a permis de développer des compétences en programmation, gestion d'interface utilisateur et optimisation de performances, tout en révélant des axes d'amélioration prometteurs pour l'avenir. Si le projet était à refaire, l'architecture pourrait bénéficier de l'implémentation d'un système de plugins pour faciliter l'ajout de nouvelles variantes d'automates cellulaires. Les fonctionnalités envisagées à moyen terme incluent la prise en charge de règles personnalisables (automates de Langton généralisés), un mode multi-fourmis avec collision, et une interface web pour favoriser l'accessibilité universelle. L'aspect inclusion pourrait être renforcé par l'ajout de raccourcis clavier complets, d'options de contraste élevé, et de descriptions audio pour les utilisateurs malvoyants. Cette expérience a

particulièrement développé ma capacité à concevoir des interfaces intuitives et à gérer la complexité croissante d'un projet logiciel, compétences directement transférables vers des projets de plus grande envergure.