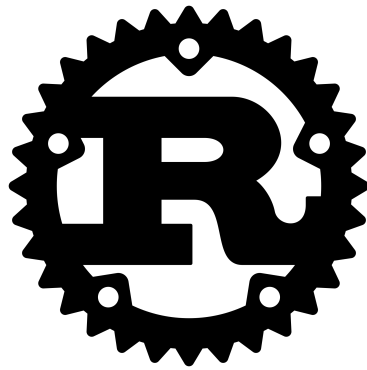


Rapport de Soutenance



Rust.eze

Table des matières

1	Origine et nature du projet	2
2	État de l’art	4
2.1	Modèles mathématiques classiques	4
2.2	Plateformes de simulation multi-agents	4
2.3	Les apports de Rust	5
2.4	Conclusion et perspectives	5
3	Notre équipe	6
4	Répartition des tâches	7
5	Bilan et perspectives des responsables	8
5.1	Guillaume DUFOUR	8
5.2	Lucas BONDARENKO	10
5.3	Amir-Alexandre BARKALLAH	12
5.4	Philippe RASTOUL	14
6	Visuels du projet	17
7	État d’avancement des tâches	19
7.1	Soutenance 1	19
7.2	Soutenance 2	19
7.3	Soutenance 3	19
8	Objet de l’étude	20

1 Origine et nature du projet

L'idée de créer un simulateur interactif d'écosystème naturel est née de la volonté de mieux comprendre, de manière concrète et dynamique, la **complexité du vivant**. Les écosystèmes naturels sont des ensembles d'êtres vivants (faune et flore) en interrelation constante avec leur environnement (climat, ressources, saisonnalité, etc.). Ces interactions sont nombreuses et souvent non linéaires : un léger changement de population chez une espèce peut entraîner des effets en cascade sur l'ensemble de l'écosystème.

Complexité et importance des interactions :

- *Prédation et chaînes alimentaires* : Les relations proies-prédateurs influencent grandement la régulation des populations animales et peuvent modifier la composition végétale lorsqu'une forte pression de prédation (ou au contraire une absence de prédateurs) se fait sentir.
 - *Reproduction et dispersion* : Les espèces se reproduisent selon des cycles et des comportements variés (accouplement, ponte, floraison, etc.), qui peuvent eux-mêmes dépendre de paramètres environnementaux tels que la température ou la disponibilité de la nourriture.
 - *Compétition pour les ressources* : Les plantes se disputent la lumière et les nutriments, tandis que les animaux se concurrencent pour l'eau, la nourriture ou l'espace vital. Cette compétition peut contraindre l'évolution des stratégies de survie et d'adaptation au sein de chaque espèce.
 - *Rôles écosystémiques variés* : Certains organismes, comme les pollinisateurs (abeilles, papillons), jouent un rôle crucial dans la reproduction des plantes. Les décomposeurs (champignons, bactéries) sont essentiels pour recycler la matière organique.
- Les interactions sont donc multiples et interdépendantes.

De plus, **l'environnement physique et climatique** ajoute un niveau de complexité supplémentaire. Les variations de température, d'humidité, la fréquence des précipitations, la survenue d'événements extrêmes (sécheresses, canicules, incendies, inondations, etc.) influencent directement la survie et la dynamique des espèces. Les catastrophes naturelles peuvent perturber drastiquement un écosystème, et la capacité de celui-ci à s'en remettre (résilience) dépend de sa diversité et de la solidité de ses relations internes.

En proposant un **outil de simulation performant**, nous cherchons à observer et analyser ces phénomènes dans un environnement virtuel contrôlé. Cela permet :

- De tester différents scénarios (introduction d'une nouvelle espèce, suppression d'un prédateur, réintroduction d'espèces disparues, etc.).
- D'étudier la résilience d'un écosystème face à des perturbations (catastrophes naturelles, épidémies, etc.).
- D'anticiper les conséquences de variations climatiques ou de pratiques humaines sur la biodiversité.

Les objectifs pédagogiques et scientifiques sont donc multiples : fournir aux étudiants un environnement expérimental virtuel pour mieux comprendre les grands principes de l'écologie, offrir aux chercheurs une plateforme évolutive pour étudier différents mécanismes écologiques, et permettre une sensibilisation du grand public à l'importance de préserver la biodiversité et les équilibres naturels.

Pour garantir la fiabilité et l'**efficacité** de la simulation, nous avons choisi le langage de programmation **Rust**. Celui-ci offre :

- De hautes performances pour gérer un grand nombre d’entités, de calculs et d’interactions en temps réel.
- Une gestion stricte de la mémoire, réduisant les risques d’erreurs courantes (fuites de mémoire, segmentation fault, etc.).
- Un système de typage et de compilation qui encourage des pratiques de développement robustes.

L’infrastructure **open source** et basée sur Linux favorise par ailleurs la **collaboration** et l’interopérabilité avec d’autres systèmes. Ainsi, ce projet s’inscrit à la fois dans une démarche de recherche ouverte et dans une démarche d’innovation technique. En associant pédagogie, recherche et haute performance, il vise à dépasser les modèles statiques ou simplifiés pour offrir une **vision intégrée et évolutive** de la dynamique des écosystèmes.

En résumé, l’**origine et la nature du projet** tiennent à la fois d’une curiosité scientifique profonde envers la complexité du vivant et d’une volonté d’offrir un outil de simulation accessible, performant et extensible. Cette plateforme pourra servir à **identifier, analyser et comprendre** les mécanismes écologiques clé, tout en permettant à des utilisateurs variés (étudiants, enseignants, chercheurs, passionnés de nature) de manipuler différents scénarios de manière interactive et éclairée.

2 État de l'art

La modélisation des écosystèmes est un domaine ancien, remontant aux premiers travaux en **écologie théorique** et en **biomathématiques**, et elle évolue constamment au fil des découvertes scientifiques et des avancées en informatique. Au cœur de ce champ se trouvent les **interactions entre les organismes vivants** et leur environnement, dont l'étude requiert souvent la mise en place de modèles de complexité variable.

2.1 Modèles mathématiques classiques

Les premiers outils de modélisation de la dynamique des populations se sont basés sur des **équations différentielles**. Les équations de Lotka-Volterra, par exemple, décrivent l'évolution des populations proies et prédateurs dans un cadre simplifié. Ces modèles ont permis d'établir d'importants principes en écologie, notamment :

- La mise en évidence de *cycles prédateurs-proies* (alternance d'abondance et de raréfaction).
- L'influence réciproque des populations sur leur taux de croissance ou de décroissance.

Cependant, **les limites** de ces modèles résident dans leur simplification parfois *trop* extrême des relations biologiques et de la **variabilité environnementale**. Par exemple, ils ne prennent généralement pas en compte :

- Les changements saisonniers ou climatiques.
- Les variations du comportement individuel (migration, apprentissage, stratégies d'adaptation).
- Les phénomènes stochastiques (perturbations aléatoires, épidémies, catastrophes naturelles).

Pourtant, ils constituent un **socle théorique** solide pour comprendre les mécanismes fondamentaux de la dynamique des populations et servent encore de base à de nombreux travaux.

2.2 Plateformes de simulation multi-agents

Avec l'essor de l'informatique, les approches **multi-agents** se sont développées. Des plateformes telles que **NetLogo**, **GAMA** ou **MASON** permettent de simuler de manière **individualisée** le comportement de chaque entité (un animal, une plante, un humain, etc.). Chaque agent est doté de règles (décision, déplacement, reproduction, consommation de ressources) et évolue dans un environnement commun.

Avantages :

- *Granularité fine* : On peut modéliser des comportements individuels complexes, des déplacements spatialisés, des interactions locales.
- *Visualisation dynamique* : L'évolution de la simulation se suit en temps réel, facilitant la compréhension des phénomènes émergents.
- *Flexibilité* : On peut introduire ou retirer des règles, tester divers scénarios et observer l'impact global de changements locaux.

Inconvénients :

- *Performance limitée* : Lorsque le nombre d'agents et le niveau de détail des interactions augmentent, les calculs peuvent devenir très lourds et ralentir la simulation.

- *Complexité de la conception* : Définir des comportements réalistes et gérer les multiples interactions peut rapidement conduire à des modèles difficiles à maintenir ou à valider.

2.3 Les apports de Rust

Le langage **Rust** est récent mais se distingue déjà par sa **philosophie** et sa **gestion mémoirelle** :

- **Performance** : Rust rivalise avec le C/C++ en termes de rapidité d'exécution, ce qui est crucial pour simuler des écosystèmes complexes et potentiellement massifs (grand nombre d'agents, multiples itérations).
- **Sécurité mémoire** : Grâce au *borrowing checker* et à l'absence de garbage collector, Rust prévient une large classe d'erreurs (fuites, corruptions mémoire), assurant une **fiabilité** accrue et des optimisations plus sûres.
- **Parallélisation** : Le compilateur de Rust et son modèle de concurrence (basé sur l'emprunt et la propriété de données) facilitent l'écriture de code parallèle sans risquer d'accéder à des données partagées de manière non sécurisée.
- **Écosystème grandissant** : La communauté Rust propose de nombreuses bibliothèques (*crates*) pour la simulation, la visualisation ou le calcul scientifique, permettant un développement plus rapide et modulaire.

Ces propriétés font de Rust un choix particulièrement adapté pour **dépasser les limites** des plateformes de simulation multi-agents traditionnelles et gérer des **scénarios de grande envergure** sans sacrifier la stabilité.

2.4 Conclusion et perspectives

En associant :

- **Les fondements théoriques** des modèles mathématiques (comme Lotka-Volterra),
- **La granularité des approches multi-agents** (pour les interactions complexes),
- **La puissance et la sécurité** offertes par Rust,

ce projet entend **proposer une solution innovante** pour la **simulation écologique**. Les ambitions principales sont :

- Modéliser des écosystèmes **riches et réalistes**, avec de multiples espèces, ressources et facteurs environnementaux.
- Gérer des **charges de calcul** potentiellement très importantes grâce à la puissance de Rust.
- Offrir une **plateforme flexible et extensible**, qui pourra être enrichie de nouvelles règles, de nouveaux modules de visualisation, ou encore de nouvelles perspectives de recherche (évolution des comportements, intelligence artificielle, etc.).

Ce travail se situe donc au carrefour de l'**écologie**, de la **biologie**, de l'**informatique** et du **développement logiciel**, ouvrant la voie à des recherches interdisciplinaires. En visant à combler les lacunes des approches existantes, ce simulateur a pour objectif de devenir un **outil de référence** pour qui souhaite comprendre et expérimenter les dynamiques écosystémiques de manière à la fois *rigoureuse*, *flexible* et *performante*.

3 Notre équipe

- **Guillaume DUFOUR : Chef de projet et Responsable Architecture et Coordination.**

En tant que chef de projet, Guillaume veille à la bonne répartition des tâches, au respect du calendrier de développement et à la coordination globale de l'équipe. Il définit également l'architecture logicielle du simulateur et s'assure de sa cohérence d'ensemble. Son rôle inclut la planification des réunions et la supervision des décisions techniques majeures. Il est aussi le développeur du site web.

- **Lucas BONDARENKO : Responsable Dynamique des Ressources.**

Lucas est chargé de modéliser la croissance et la répartition de la flore. Il travaille sur les algorithmes de régulation et les mécanismes qui gèrent l'interaction entre les plantes, l'environnement et les ressources disponibles. Son expertise permet de rendre la simulation réaliste quant aux limitations écologiques et à la compétition entre espèces végétales.

- **Amir-Alexandre BARKALLAH : Responsable Faune et Interactions Biologiques.**

Amir-Alexandre conçoit et implémente les comportements animaux (déplacement, reproduction, prédation, etc.) ainsi que les interactions biologiques. Il s'appuie sur des modèles biologiques et écologiques existants afin de reproduire au mieux les dynamiques de populations animales dans l'écosystème. Il collabore étroitement avec Lucas pour synchroniser les cycles animaux et végétaux.

- **Philippe RASTOUL : Responsable Interface, Visualisation et Optimisation.**

Philippe se concentre sur la conception de l'interface (graphique ou ligne de commande) permettant aux utilisateurs de configurer et d'observer la simulation en temps réel. Parallèlement, il optimise le code pour exploiter au mieux les ressources matérielles disponibles (calcul parallèle, utilisation de bibliothèques Rust performantes). Il veille également à la qualité de l'expérience utilisateur, en facilitant l'interaction avec le simulateur.

4 Répartition des tâches

Tâche	Responsable
Analyse des besoins	Guillaume DUFOUR
Conception de l'architecture	Guillaume DUFOUR
Implémentation des plantes	Lucas BONDARENKO
Implémentation des animaux	Amir-Alexandre BARKALLAH
Conception de l'interface	Philippe RASTOUL
Tests unitaires	Amir-Alexandre BARKALLAH
Optimisation et parallélisation	Philippe RASTOUL
Documentation	Lucas BONDARENKO

TABLE 1 – Répartition des tâches

5 Bilan et perspectives des responsables

5.1 Guillaume DUFOUR

Le projet propose une solution de simulation d'écosystèmes fondée sur une architecture modulaire et une configuration centralisée permettant une gestion fine et adaptative des paramètres essentiels. Dans le fichier `config.rs`, la structure `SimulationConfig` regroupe désormais un ensemble étendu d'éléments critiques, tels que la taille de la grille (`grid_width`, `grid_height`), le nombre initial d'agents répartis en catégories distinctes (plantes claires et foncées, herbivores, carnivores, omnivores), ainsi que les paramètres de croissance végétale, les seuils de reproduction spécifiques et les dynamiques d'énergie (`herbivore_energy_gain`, `carnivore_energy_gain`, `omnivore_energy_gain_plants`, etc.). Cette centralisation garantit une cohérence accrue dans les simulations et facilite l'ajustement rapide des variables pour explorer une grande diversité de scénarios. L'utilisation du langage Rust continue d'offrir au projet des avantages majeurs en termes de sécurité mémoire et de performances optimisées, essentiels à l'évolution d'un système complexe et évolutif.

Le projet implémente désormais une orchestration encore plus raffinée des interactions entre ses différents modules. La fonction `step`, située dans `ecosystem.rs`, constitue toujours le cœur de cette coordination, mais intègre dorénavant des éléments supplémentaires tels que la gestion dynamique d'environnements comme les lacs et les arbres (via `maybe_spawn_water` et `maybe_spawn_tree`). Ce processus gère la croissance aléatoire des plantes, le déplacement, l'alimentation et la reproduction des herbivores, carnivores et omnivores, ainsi que les interactions spécifiques liées à l'apparition d'eau et d'arbres. L'approche modulaire adoptée, utilisant des stratégies de clonage maîtrisées et une manipulation optimisée des vecteurs d'agents, permet de préserver un équilibre dynamique tout en fournissant des statistiques précises, notamment les consommations différenciées des omnivores et carnivores, facilitant l'analyse et l'amélioration continue du système.

L'interface utilisateur interactive, développée dans `main.rs` à l'aide de la bibliothèque `macroquad`, présente des améliorations significatives. Elle propose toujours une visualisation temps réel de l'écosystème, enrichie par une expérience utilisateur plus fluide grâce à l'affichage en plein écran et à une gestion intuitive des interactions. Le menu de configuration permet désormais de spécifier précisément les chances d'apparition de lacs et d'arbres, ainsi que les populations initiales différenciées de plantes claires et foncées. Parmi les fonctionnalités avancées figurent la navigation à travers l'historique des itérations, le suivi détaillé et interactif des agents sélectionnés avec des indicateurs visuels distinctifs, ainsi que l'affichage enrichi de statistiques globales et individuelles des agents. Cette interface renforce non seulement l'engagement utilisateur, mais améliore également la compréhension intuitive des dynamiques écologiques complexes.

Durant son développement récent, le projet a affronté plusieurs défis techniques nouveaux, notamment la gestion complexe des interactions environnementales introduites par l'ajout d'agents spéciaux tels que les lacs et les arbres. Afin de répondre à ces défis, une stratégie approfondie de modularisation et d'isolation fonctionnelle a été adoptée, facilitant le test et l'optimisation des nouveaux composants indépendamment du cœur existant. L'emploi de Rust et de ses fonctionnalités avancées telles que la gestion explicite des durées de vie et l'absence de garbage collector ont permis de maintenir des performances élevées et d'éviter les problèmes classiques de fuite de mémoire. Des mécanismes supplémentaires de validation et de suivi détaillé des états internes (par exemple via `TrackingInfo`) as-

surent désormais la cohérence complète des simulations et facilitent la résolution rapide des éventuels goulets d'étranglement.

En termes de perspectives d'évolution, le projet envisage plusieurs pistes ambitieuses d'amélioration. L'intégration d'algorithmes d'apprentissage automatique pour adapter dynamiquement les paramètres de la simulation selon les évolutions observées représente une avancée notable envisagée à court terme. Un système de monitoring et un tableau de bord interactif en temps réel pourraient être développés afin d'identifier immédiatement toute anomalie et proposer automatiquement des ajustements pour garantir la stabilité continue du système. Enfin, l'élaboration d'une documentation technique approfondie et évolutive favorisera une meilleure intégration de futurs modules et facilitera grandement la prise en main par de nouveaux utilisateurs, positionnant le simulateur comme une référence incontournable pour la recherche en écologie numérique et l'enseignement académique.

5.2 Lucas BONDARENKO

L'écosimulateur développé vise à modéliser un environnement dynamique où différents types d'agents interagissent entre eux. Parmi ces agents, les plantes occupent une place centrale, constituant la base alimentaire des herbivores et omnivores. L'implémentation des plantes a évolué significativement pour intégrer deux types distincts clairement définis dans la structure `AgentType` : les plantes claires (`LightPlant`) et les plantes foncées (`DarkPlant`). Cette distinction enrichit la dynamique de concurrence végétale et permet une évolution naturelle plus réaliste du paysage végétal.

Cette partie détaille le fonctionnement des plantes dans l'écosystème, notamment leurs mécanismes améliorés de croissance, de remplacement et d'interaction avec les nouveaux éléments environnementaux comme les lacs et les arbres.

Lors du lancement d'une simulation, le nombre initial des plantes claires et foncées peut être précisément défini par l'utilisateur depuis l'écran de configuration, grâce aux champs spécifiques `initial_light_plants` et `initial_dark_plants` dans la structure `SimulationConfig`. Chaque plante possède un taux de croissance spécifique (`plant_growth_rate`) qui détermine sa capacité à se reproduire à chaque étape de la simulation, tout en respectant la règle stricte d'une seule plante par case.

Le cycle de vie des plantes, géré dans la fonction `step()` du fichier `ecosystem.rs`, comprend désormais des mécanismes enrichis :

- Chaque plante tente de se reproduire en choisissant aléatoirement une position libre sur la grille.
- Si la position est déjà occupée par une plante du type opposé, un mécanisme dynamique permet à la nouvelle plante de remplacer l'ancienne, favorisant ainsi une compétition végétale réaliste.
- L'apparition de lacs et d'arbres influence significativement la répartition végétale : les lacs favorisent exclusivement la croissance des plantes claires, tandis que les arbres privilégient celle des plantes foncées, complexifiant davantage l'interaction végétale.

Ce mécanisme assure une évolution constante des plantes à travers une compétition dynamique pour l'espace disponible.

Les plantes continuent d'être consommées par les herbivores et les omnivores lorsqu'ils se déplacent sur leur case, maintenant ainsi l'équilibre dynamique de l'écosystème.

Plusieurs défis techniques ont été relevés pendant le développement :

1. **Superposition des plantes :** *Problème* : Plusieurs plantes pouvaient initialement apparaître simultanément sur une même case, créant une situation irréaliste. *Solution* : Un contrôle strict a été intégré avant l'ajout d'une nouvelle plante dans la fonction `step()`, garantissant une unique plante par case.
2. **Impossibilité de remplacement végétal :** *Problème* : Les plantes ne pouvaient pas remplacer celles d'un type différent, limitant la dynamique écologique. *Solution* : L'implémentation actuelle permet explicitement à une plante de remplacer une plante du type opposé, enrichissant les interactions végétales.
3. **Paramétrage initial limité :** *Problème* : Initialement, les utilisateurs ne pouvaient pas spécifier individuellement les quantités initiales de chaque type de plante. *Solution* : Le fichier de configuration et l'interface utilisateur ont été enrichis de champs dédiés (`initial_light_plants`, `initial_dark_plants`), facilitant ainsi une personnalisation plus fine de la simulation.

Ces améliorations garantissent une dynamique végétale réaliste et permettent une meilleure adaptation des scénarios écologiques simulés.

5.3 Amir-Alexandre BARKALLAH

Le projet propose une approche innovante et complète pour simuler les interactions biologiques et les comportements animaliers au sein d'un écosystème virtuel dynamique. En s'appuyant sur des modèles écologiques robustes et une architecture logicielle modulaire implémentée en Rust, la solution articule précisément plusieurs modules essentiels répartis dans les fichiers `config.rs`, `ecosystem.rs` et `main.rs`. Cette organisation modulaire permet non seulement de configurer finement l'ensemble des paramètres critiques de la simulation, mais aussi d'orchestrer dynamiquement les interactions complexes entre agents et leur environnement, tout en offrant une interface utilisateur interactive pour une visualisation approfondie en temps réel.

La simulation s'appuie sur un ensemble complet d'algorithmes qui intègrent des mécanismes sophistiqués de reproduction, de déplacement, de prédation, et désormais, des interactions environnementales spécifiques comme l'apparition aléatoire de lacs et d'arbres. Dans le fichier `config.rs`, la structure `SimulationConfig` définit avec précision les paramètres essentiels tels que les seuils énergétiques de reproduction et de consommation spécifiques aux herbivores, carnivores et omnivores, ainsi que les taux de croissance végétale et les durées de vie des éléments environnementaux. La gestion énergétique détaillée inclut des valeurs clairement identifiées telles que `herbivore_energy_gain`, `carnivore_energy_gain`, `omnivore_energy_gain_plants`, et `omnivore_energy_gain_herbivores`, facilitant un calibrage précis et réaliste des dynamiques écologiques.

La fonction centrale `step`, située dans `ecosystem.rs`, représente le cœur opérationnel de la simulation, orchestrant minutieusement l'évolution des différents agents. À chaque itération, elle traite simultanément les déplacements aléatoires, la consommation des ressources, les mécanismes de reproduction ainsi que les interactions spécifiques introduites par les lacs et les arbres. La fonction auxiliaire `random_adjacent_aux` génère des déplacements naturels aléatoires réalistes pour chaque agent, tandis que les contrôles rigoureux des gains et pertes énergétiques assurent une cohérence écologique et permettent la gestion de dynamiques démographiques complexes. Cette architecture modulaire et itérative garantit une synchronisation précise des cycles biologiques et environnementaux, reflétant fidèlement les dynamiques naturelles observées dans les écosystèmes réels.

L'intégration poussée de ces mécanismes biologiques au sein d'une interface utilisateur interactive, développée dans le fichier `main.rs` grâce à la bibliothèque `macroquad`, améliore significativement l'expérience utilisateur. Cette interface propose une visualisation claire et fluide en temps réel, facilitant la compréhension intuitive des dynamiques écologiques complexes. Les utilisateurs bénéficient de fonctionnalités avancées telles que la navigation détaillée à travers l'historique complet des itérations, le suivi interactif des agents spécifiques, ainsi que l'affichage d'indicateurs statistiques détaillés permettant une analyse approfondie des comportements émergents. L'interface constitue ainsi un outil indispensable pour identifier rapidement les déséquilibres écologiques et optimiser les paramètres de simulation.

Le développement récent de cette modélisation a mis en lumière plusieurs défis techniques importants, notamment liés à la gestion simultanée des nombreuses interactions entre agents et éléments environnementaux dans des zones fortement peuplées. Ces défis ont entraîné initialement des comportements imprévus et des oscillations dans les populations. Pour résoudre ces problèmes, le projet a adopté une approche itérative d'ajustement minutieux des paramètres énergétiques et reproductifs, accompagnée d'une refonte partielle de certaines parties de la logique interne de la fonction `step`, améliorant ainsi la

stabilité des cycles biologiques. Une étroite collaboration entre les équipes responsables de la dynamique des ressources et celles dédiées à la modélisation animale a permis de garantir une cohérence et une synchronisation optimale, minimisant ainsi les conflits de dépendances internes et assurant une évolution harmonieuse de l'écosystème.

Pour renforcer davantage la précision et la complexité de la simulation, plusieurs axes d'amélioration ambitieux sont envisagés. L'intégration prochaine d'algorithmes avancés d'intelligence artificielle, notamment par apprentissage par renforcement, permettra aux agents d'adopter des comportements adaptatifs en réponse aux changements environnementaux observés. Par ailleurs, l'ajout futur de modèles plus complexes de migration saisonnière, de dispersion génétique et de communication inter-agents offrira des dimensions supplémentaires pour simuler des stratégies collectives avancées telles que la coopération ou la compétition sophistiquée. Enfin, la mise en place prévue d'un tableau de bord interactif pour le suivi en temps réel, combinée à une documentation technique exhaustive et régulièrement mise à jour, positionnera définitivement ce simulateur comme un outil essentiel dans les domaines de la recherche en écologie théorique et de l'enseignement universitaire.

5.4 Philippe RASTOUL

Pour commencer à créer l'interface graphique du simulateur, il était essentiel de sélectionner une bibliothèque adaptée permettant la réalisation d'applications interactives intuitives. La bibliothèque choisie, **macroquad**, est une bibliothèque Rust spécialisée dans la création de graphiques 2D. Cette dernière se distingue par sa simplicité, sa rapidité d'exécution et sa facilité d'utilisation. Grâce à ses abstractions claires et efficaces, **macroquad** permet aux développeurs de se concentrer directement sur la logique applicative sans devoir gérer les détails complexes propres aux moteurs graphiques de bas niveau. Cette caractéristique facilite grandement la gestion des interactions utilisateurs, telles que les clics de souris, les entrées clavier, ainsi que les contrôles graphiques en temps réel.

Une fois la bibliothèque intégrée, l'étape suivante a consisté à implémenter la configuration initiale de la fenêtre d'affichage. La fonction `clear_background()` a été utilisée pour rafraîchir le fond d'écran avec une couleur prédéfinie, tandis que la fonction `draw_text()` permet d'afficher clairement les paramètres et les caractéristiques clés de la simulation à des coordonnées précises de l'écran. L'utilisation d'une police de taille 20 a été choisie après plusieurs essais, car elle offrait un bon compromis entre lisibilité et compacité, permettant une présentation ordonnée et facile à parcourir des divers paramètres disponibles dans une seule colonne bien structurée. La gestion temporelle des images affichées est quant à elle assurée par la fonction `next_frame().await`, qui synchronise efficacement le rafraîchissement de l'affichage.

Au lancement de l'application, un menu de configuration est automatiquement présenté à l'utilisateur. Ce menu permet de définir en détail les paramètres spécifiques qui régissent le comportement des plantes, des herbivores, des carnivores et des omnivores au sein de la grille de simulation. L'interface utilisateur propose des champs ajustables directement au clavier, chacun correspondant à un aspect précis de la dynamique de l'écosystème.

Les paramètres configurables incluent notamment :

Plantes claires initiales : Nombre initial de plantes claires présentes dans la simulation.

Plantes foncées initiales : Nombre initial de plantes foncées dans la simulation.

Herbivores initiaux : Quantité initiale d'herbivores introduite dans l'écosystème.

Carnivores initiaux : Nombre initial de carnivores disponibles au début de la simulation.

Omnivores initiaux : Nombre initial d'omnivores introduits dans la grille.

Taux de croissance des plantes : Paramètre flottant définissant le taux de reproduction des plantes à chaque itération.

Gain d'énergie des herbivores : Énergie gagnée par un herbivore lorsqu'il consomme une plante.

Perte d'énergie des herbivores : Quantité d'énergie perdue par les herbivores à chaque déplacement ou action.

Énergie initiale des herbivores : Niveau initial d'énergie dont disposent les herbivores au démarrage de la simulation.

Seuil de reproduction des herbivores : Niveau d'énergie requis pour permettre aux herbivores de se reproduire.

Gain d'énergie des carnivores : Énergie gagnée par un carnivore lorsqu'il consomme un herbivore.

Perte d'énergie des carnivores : Énergie perdue par les carnivores à chaque étape.

Énergie initiale des carnivores : Niveau initial d'énergie des carnivores à leur création.

Seuil de reproduction des carnivores : Niveau d'énergie nécessaire pour permettre la reproduction des carnivores.

Gain d'énergie des omnivores : Énergie acquise par un omnivore selon le type de nourriture (plante ou herbivore).

Perte d'énergie des omnivores : Énergie consommée par les omnivores à chaque déplacement.

Énergie initiale des omnivores : Niveau d'énergie initial attribué aux omnivores.

Seuil de reproduction des omnivores : Niveau d'énergie requis pour déclencher la reproduction des omnivores.

Les utilisateurs peuvent aisément modifier ces champs à l'aide de commandes simples et intuitives :

- Utiliser les touches Flèche haut et Flèche bas pour naviguer entre les champs.
- Taper directement des chiffres (y compris décimaux) pour ajuster les valeurs.
- Appuyer sur Retour arrière pour corriger les entrées erronées.
- Appuyer sur Entrée pour valider les paramètres et lancer la simulation avec la configuration choisie.

Afin de rendre le menu de configuration encore plus visuellement cohérent, une petite touche de décoration y a été apportée. Chaque paramètre est désormais coloré de la même manière que les éléments sur la grille de biodiversité, facilitant ainsi la lecture des paramètres.

Couleurs :

- **Vert** : Plantes claires
- **Vert foncé** : Plantes sombres
- **Rose** : Herbivores
- **Rouge** : Carnivores
- **Orange** : Omnivores
- **Bleu** : Points d'eau
- **Marron** : Arbre

Une fois lancée, la simulation propose une exploration détaillée grâce à une fonction avancée de navigation dans l'historique des itérations, permettant de revenir en arrière ou d'avancer dans le temps pour examiner précisément l'évolution de l'écosystème.

Contrôles de navigation dans l'historique :

Flèche gauche (\leftarrow) : Retour à l'étape précédente.

Flèche droite (\rightarrow) : Avance à l'étape suivante ou crée une nouvelle itération si nécessaire.

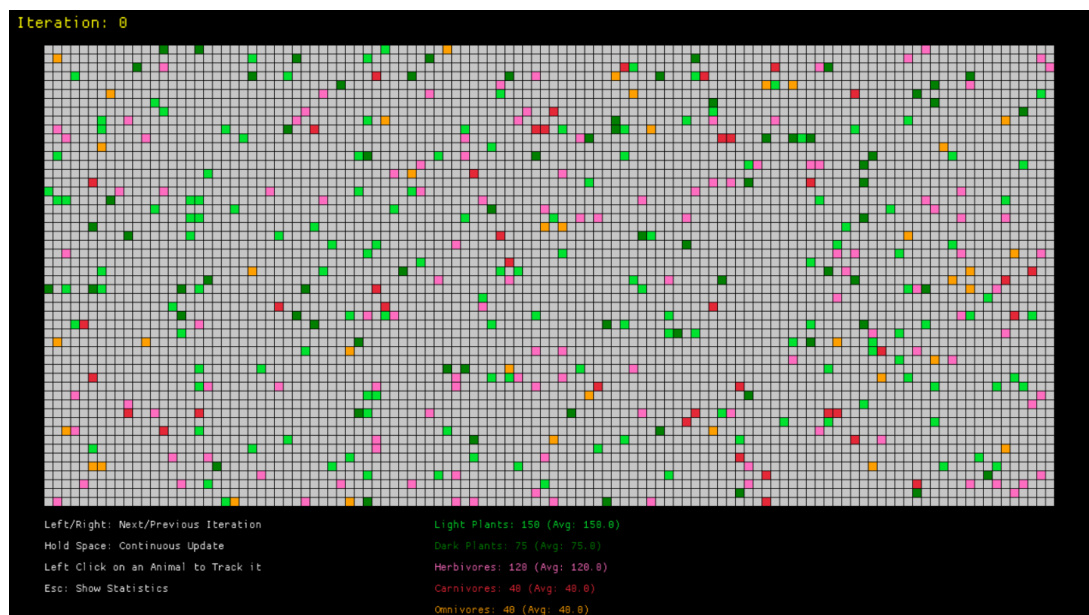
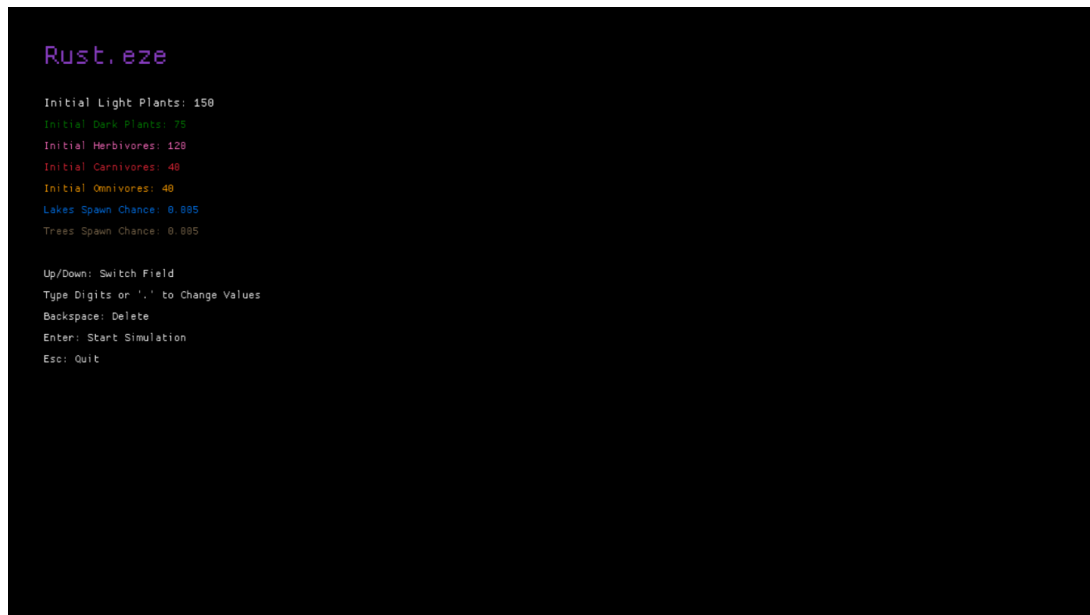
Barre d'espace : Avancement continu de la simulation avec création automatique d'historique.

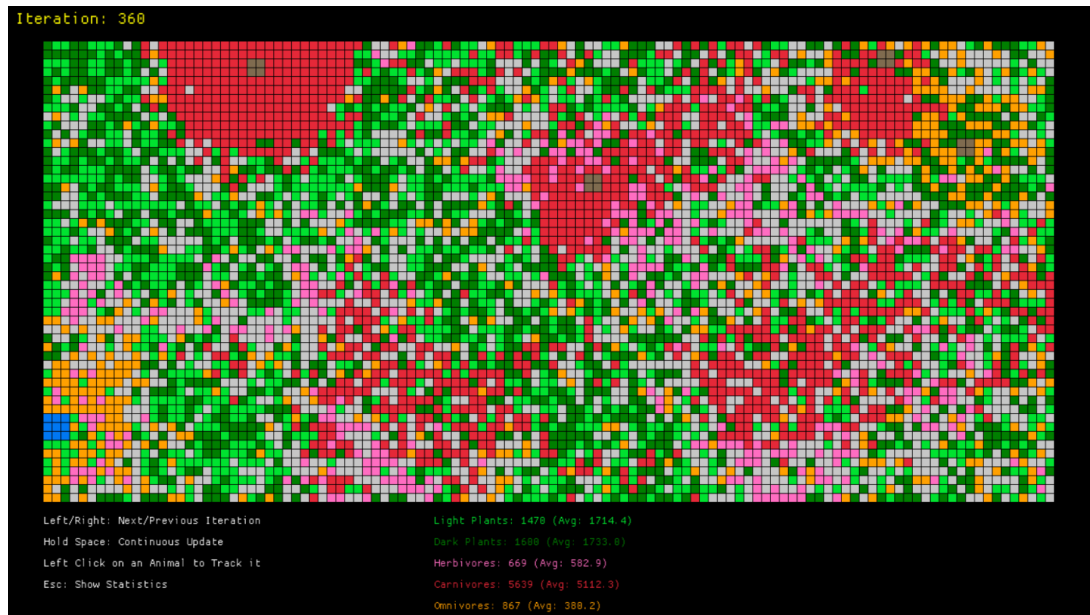
Échap : Retour au menu de configuration.

Pendant la simulation, l'utilisateur peut interagir directement avec la grille en cliquant sur les cellules, permettant ainsi le suivi détaillé et en temps réel d'un agent sélectionné. Les informations sur la naissance, la position, l'énergie et le statut (vivant ou décédé) de l'agent suivi sont affichées clairement à l'écran, enrichissant ainsi l'expérience utilisateur.

Enfin, bien que les fondements interactifs soient opérationnels, des améliorations futures incluent notamment l'ajout d'un bouton de retour rapide au menu de configuration, ainsi qu'une amélioration esthétique significative par l'intégration de couleurs distinctives pour les agents, rendant ainsi l'application plus intuitive et visuellement agréable.

6 Visuels du projet





Simulation Statistics				
Iteration Count: 360				
Light Plants				
Births:	56269	Deaths:	55788	
Dark Plants				
Births:	59385	Deaths:	57821	
Herbivores				
Births:	21814	Deaths:	114312	Consumptions: 56197
Carnivores				
Births:	182298	Deaths:	76451	Consumptions: 182418
Omnivores				
Births:	1876	Deaths:	1286	Consumptions (Plants): 53979 Consumptions (Herbivores): 8159
Lakes				
Appearances:	1	Disappearances:	0	
Trees				
Appearances:	4	Disappearances:	0	
Press Esc Again to Quit				

7 État d'avancement des tâches

7.1 Soutenance 1

- Analyse des besoins : 100% Terminé
- Conception de l'architecture : 40% En cours
- Implémentation des plantes : 10% Débuté
- Implémentation des animaux : 0% Non débuté
- Tests unitaires : 0% Non débuté
- Documentation : 0% Non débuté

7.2 Soutenance 2

- Analyse des besoins : 100% Terminé
- Conception de l'architecture : 90% En cours
- Implémentation des plantes : 80% En cours
- Implémentation des animaux : 90% En cours
- Tests unitaires : 60% En cours
- Documentation : 60% En cours

7.3 Soutenance 3

- Analyse des besoins : 100% Terminé
- Conception de l'architecture : 100% Terminé
- Implémentation des plantes : 100% Terminé
- Implémentation des animaux : 100% Terminé
- Tests unitaires : 100% Terminé
- Documentation : 100% Terminé

8 Objet de l'étude

L'objet de cette étude est de **concevoir et développer un simulateur d'écosystème naturel** capable de reproduire fidèlement et de manière flexible les dynamiques complexes régissant la vie de multiples espèces animales et végétales dans un environnement virtuel interactif. L'objectif est de **fournir un outil complet de recherche, d'enseignement et d'apprentissage**, suffisamment riche fonctionnellement pour permettre de tester des hypothèses écologiques détaillées et d'illustrer de façon ludique et visuelle les concepts fondamentaux de la biologie et de l'écologie.

Pour atteindre cet objectif ambitieux, le projet s'appuie sur **des fondations théoriques solides** intégrant des modèles mathématiques robustes et des approches de simulation multi-agents, associées à **une architecture logicielle performante et sécurisée** basée sur le langage Rust. Cette approche duale présente de multiples avantages :

- **Réalisme des interactions** : Les mécanismes biologiques intégrés incluent la prédation, la compétition, la reproduction différenciée, et prennent en compte des éléments environnementaux avancés comme l'apparition de ressources, de lacs et d'arbres, qui influencent directement les dynamiques observées. Ce réalisme permet aux scénarios simulés d'être proches des dynamiques réelles observées en milieu naturel.
- **Évolutivité et modularité** : La structure modulaire facilite l'ajout ou la suppression d'espèces, la configuration précise de paramètres écologiques variés (taux de croissance, seuils énergétiques, taux de reproduction), ainsi que la simulation de scénarios spécifiques tels que l'introduction d'espèces invasives ou les effets des changements climatiques. Cette flexibilité rend le simulateur utile à une vaste gamme d'applications pédagogiques et de recherche.
- **Performance et fiabilité** : L'utilisation de Rust assure une gestion optimale des ressources système, permettant de mener efficacement des simulations complexes à grande échelle, impliquant un nombre élevé d'agents simultanés. La sécurité mémoire intrinsèque à Rust garantit en outre une grande robustesse et une excellente durabilité du logiciel.

Du **point de vue pédagogique**, le simulateur propose une **interface graphique claire et interactive** qui permet aux utilisateurs, notamment les étudiants, d'explorer intuitivement les effets immédiats des modifications apportées aux paramètres écologiques (ressources disponibles, pressions de prédation, variations de populations initiales). En visualisant en temps réel l'évolution des espèces et la modification des équilibres écologiques, les étudiants peuvent facilement saisir et comprendre les mécanismes écologiques complexes grâce à une expérimentation directe et intuitive.

Du **point de vue de la recherche**, le simulateur constitue un **laboratoire virtuel avancé** où les chercheurs peuvent tester des scénarios écologiques complexes et variés (multiples interactions, espèces concurrentes, phénomènes aléatoires). Grâce aux données statistiques détaillées générées par le simulateur, il devient possible d'analyser finement l'impact de chaque paramètre étudié, ce qui permet d'orienter efficacement les futures recherches sur le terrain et d'affiner les modèles écologiques existants, avec un gain de temps et de ressources conséquent.

En conciliant **rigueur scientifique, performance logicielle et ergonomie**, ce projet se positionne comme un outil **transdisciplinaire** puissant et adaptable. Son ambition est d'être largement **accessible** grâce à une conception open-source compatible avec les

systèmes Linux, tout en restant **facilement appropriable** par un public varié (chercheurs, enseignants, étudiants et grand public intéressé par l'écologie). L'aspect **éducatif** est renforcé par des modules didactiques intégrés à l'interface graphique, tandis que la **dimension scientifique** est valorisée par la possibilité de simuler des scénarios extrêmement détaillés et réalistes.

En définitive, la **réalisation de ce simulateur** répond à un double enjeu : **approfondir la compréhension des écosystèmes naturels** et **proposer un outil performant, innovant et collaboratif** pour éclairer les enjeux actuels de conservation de la biodiversité et de gestion durable des ressources naturelles.

Bilan global : À ce stade du développement, des progrès significatifs ont été accomplis dans la conception et l'intégration des principaux modules de simulation. Toutefois, certains retards persistent, principalement liés à la complexité intrinsèque des interactions écologiques modélisées et à la nécessité d'itérations multiples pour assurer la cohérence globale du système. Les prochaines étapes consisteront en la finalisation, le test approfondi et l'intégration complète des modules restants, en vue d'une première version fonctionnelle et complète de l'écosimulateur.