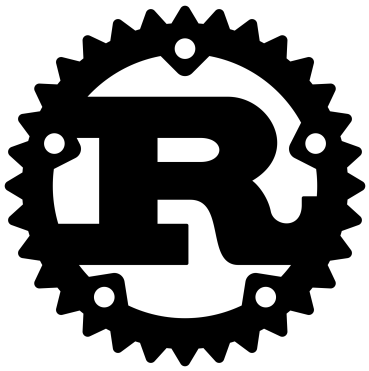
Rapport de Soutenance



Rust.eze

Table des matières

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 mémorielle** :

— **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.

— **Lucas BONDARENKO : Responsable Dynamique des Ressources.**
Lucas est chargé de modéliser la croissance et la répartition des ressources naturelles (nourriture, eau, lumière, etc.). 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 (symbiose, parasitisme, pollinisation). 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 reposant sur une architecture modulaire et une configuration centralisée, assurant ainsi une gestion fine et adaptable de l'ensemble des paramètres essentiels. Dans le fichier `config.rs`, la structure `SimulationConfig` regroupe de manière exhaustive des éléments cruciaux tels que la taille de la grille (`grid_width` et `grid_height`), le nombre initial d'agents répartis en différentes catégories (plantes, DarkGreenPlant, herbivores, carnivores), ainsi que les paramètres de croissance, les seuils de reproduction et les transferts d'énergie. Cette centralisation permet d'obtenir une cohérence dans la simulation et offre la possibilité d'ajuster facilement les variables pour explorer divers scénarios. En adoptant Rust comme langage de programmation, le projet bénéficie d'une sécurité mémoire renforcée et de performances optimisées, des atouts indispensables pour le développement d'un système complexe et évolutif.

Le projet met en œuvre une orchestration minutieuse des interactions entre ses divers modules. La fonction `step` dans le fichier `ecosystem.rs` constitue le cœur de cette coordination, en gérant simultanément la croissance aléatoire des plantes, le déplacement, l'alimentation et la reproduction des herbivores, ainsi que la prédation exercée par les carnivores. Ce mécanisme de mise à jour itératif repose sur des stratégies de clonage et de manipulation fine des vecteurs d'agents, garantissant que chaque action – qu'il s'agisse de mouvement, d'alimentation, de reproduction ou de gestion de la mortalité – soit traitée de manière synchronisée. L'approche adoptée permet ainsi de maintenir un équilibre dynamique au sein de l'écosystème simulé, tout en collectant des données statistiques précises pour faciliter l'analyse et l'optimisation continue du système.

Le projet offre une interface utilisateur interactive développée dans le fichier `main.rs` grâce à la bibliothèque `macroquad`. Cette interface propose une visualisation en temps réel de l'évolution de l'écosystème et permet aux utilisateurs de configurer les paramètres de simulation via un menu intuitif. Les fonctionnalités comprennent la navigation dans l'historique des itérations, le suivi détaillé d'agents spécifiques à l'aide d'indicateurs visuels, ainsi que l'affichage d'indicateurs statistiques pertinents. En combinant un rendu graphique fluide et des contrôles interactifs, l'interface enrichit l'expérience utilisateur tout en facilitant la compréhension des dynamiques complexes de la simulation.

Le projet a été confronté à plusieurs défis techniques majeurs, notamment la synchronisation des processus de simulation et la gestion des conflits de dépendances dans un environnement fortement concurrent. Pour surmonter ces obstacles, une stratégie de modularisation poussée a été mise en place, permettant d'isoler et de tester indépendamment chaque composant. L'utilisation de Rust, avec son système de gestion de la mémoire sans garbage collector, a permis de prévenir les fuites de mémoire et d'optimiser la performance globale. Par ailleurs, des mécanismes de validation et de vérification des états des agents ont été intégrés afin d'assurer la cohérence des interactions, tandis que des tests itératifs ont permis d'identifier et de résoudre les goulets d'étranglement dans la communication entre modules.

Dans une perspective d'évolution continue, le projet envisage de nombreuses améliorations. L'intégration d'algorithmes d'apprentissage automatique pour adapter de manière dynamique les paramètres de la simulation en fonction des conditions écologiques observées représente une avancée prometteuse. Un système de monitoring en temps réel, couplé

à un tableau de bord interactif, pourrait permettre d'identifier instantanément les anomalies et de proposer des ajustements automatiques pour maintenir la stabilité du système. Par ailleurs, la mise en place d'une documentation technique exhaustive et évolutive facilitera l'intégration de nouveaux modules et la formation des utilisateurs, positionnant ainsi le simulateur comme une plateforme de référence pour la recherche en écologie numérique et l'enseignement.

5.2 Lucas BONDARENKO

L'écosimulateur développé vise à modéliser un environnement où différents types d'agents interagissent entre eux. Parmi ces agents, les plantes jouent un rôle fondamental en tant que source de nourriture pour les herbivores. L'implémentation des plantes a nécessité plusieurs itérations afin d'assurer leur bon fonctionnement tout en respectant les contraintes de la simulation, notamment en évitant la superposition des entités et en permettant une dynamique d'évolution naturelle du paysage végétal.

Cette partie détaille le fonctionnement des plantes dans l'écosystème, l'implémentation de leurs mécanismes de croissance et de remplacement, ainsi que les défis rencontrés au cours du développement.

Pour commencer, l'environnement comprend deux types de plantes (noms pas encore choisis) :

Plantes normales (Plant)

Plantes vert foncé (DarkGreenPlant)

Ces plantes sont générées initialement avec une quantité définie par l'utilisateur lors de l'écran de configuration. Elles peuvent se propager à chaque étape de simulation avec un taux de croissance défini, mais ne peuvent pas dépasser une densité maximale d'une plante par case.

Le cycle de vie des plantes est géré dans la fonction step() de l'écosystème. À chaque itération :

- Chaque plante tente de se reproduire avec une probabilité plant_growth_rate.
- Elle choisit une case adjacente de manière aléatoire.
- Si la case est libre, une nouvelle plante du même type y pousse.
- Si la case est occupée par une plante de l'autre type, la nouvelle plante a une chance de remplacer l'ancienne pour permettre une concurrence végétale dynamique.

Ainsi, les plantes évoluent constamment en s'étendant sur l'espace disponible et en se disputant les territoires occupés par d'autres plantes.

Les plantes servent de source de nourriture pour les herbivores. Lorsqu'un herbivore se déplace sur une case contenant une plante, la plante est immédiatement consommée et retirée du jeu.

Ce mécanisme assure un équilibre entre la croissance des plantes et la consommation par les herbivores, maintenant ainsi une dynamique écologique stable.

Lors de l'implémentation nous avons rencontrer divers problèmes que nous allons détailler ci dessous.

1. Superposition des Plantes

Problème : Dans les premières versions, plusieurs plantes pouvaient apparaître sur la même case, ce qui donnait un comportement irréaliste où une même position pouvait contenir plusieurs couches de végétation.

Solution : Une vérification a été ajoutée dans la fonction step() pour s'assurer qu'une seule plante est présente par case. Avant d'ajouter une nouvelle plante, on vérifie que la case est bien libre.

2. Impossibilité pour les Plantes de Remplacer une Autre Plante

Problème : Initialement, une plante ne pouvait pousser que sur une case vide, ce qui bloquait l'expansion de certains types de plantes si la grille était déjà couverte.

Solution : Une logique a été ajoutée pour que les plantes puissent remplacer d'autres plantes d'un type différent. Lorsque la plante tente de pousser, on vérifie si une autre plante est déjà là. Si oui, on décide de la remplacer.

3. Problème de Paramétrage des Plantes Initiales

Problème : L'utilisateur ne pouvait initialement pas définir séparément les quantités initiales de Plant et DarkGreenPlant, ce qui limitait la personnalisation de la simulation.

Solution : Une nouvelle entrée a été ajoutée à l'écran de configuration, permettant de fixer une valeur spécifique pour chaque type de plante. Le SimulationConfig a été mis à jour afin d'y remédier ainsi que l'ajout de code dans le menu de configuration et lors de la création de l'écosystème.

5.3 Amir-Alexandre BARKALLAH

Le projet propose une approche innovante pour simuler les interactions biologiques et les comportements animaliers dans un écosystème virtuel. S'appuyant sur des modèles écologiques éprouvés et sur une architecture logicielle robuste écrite en Rust, la solution intègre de manière cohérente plusieurs modules répartis dans les fichiers `config.rs`, `ecosystem.rs` et `main.rs`. Cette répartition permet non seulement de définir précisément les paramètres de la simulation, mais aussi d'orchestrer dynamiquement l'évolution des agents et de fournir une interface interactive pour une visualisation en temps réel.

La simulation intègre un ensemble d'algorithmes dédiés à la reproduction, au déplacement, à la prédation et à des interactions complexes telles que la symbiose. Dans le fichier `config.rs`, les paramètres clés – tels que `herbivore_reproduction_rate`, `carnivore_reproduction_rate`, `herbivore_energy_gain`, et `carnivore_energy_gain` – sont définis de manière précise dans la structure `SimulationConfig`. Ces valeurs servent de référence pour la dynamique des agents, permettant de calibrer finement la gestion énergétique et les seuils de reproduction. Dans le fichier `ecosystem.rs`, la fonction `step` orchestre l'évolution de l'écosystème en traitant successivement les déplacements, la consommation de ressources et la reproduction des herbivores et carnivores. La fonction `random_adjacent_aux` génère des mouvements aléatoires qui simulent le comportement naturel des agents, tandis que des contrôles sur les gains et pertes d'énergie garantissent que chaque itération reflète des interactions réalistes entre les espèces. Cette approche itérative et modulaire permet d'atteindre une synchronisation fine des cycles biologiques et environnementaux, assurant ainsi une simulation fidèle aux dynamiques naturelles.

L'intégration de la modélisation animalière au reste du système se retrouve également dans le fichier `main.rs`. Ici, l'interface utilisateur développée avec la bibliothèque `macroquad` permet de visualiser en temps réel l'évolution des populations d'agents. L'interface offre des fonctionnalités interactives, telles que le suivi d'agents spécifiques et la navigation dans l'historique des itérations, facilitant ainsi l'observation des comportements émergents. Cette visualisation en temps réel, combinée aux mises à jour régulières de la simulation, permet d'identifier rapidement les déséquilibres ou les anomalies dans le comportement des agents, et offre un outil précieux pour le diagnostic et l'optimisation des paramètres de simulation.

Le développement de cette modélisation a mis en évidence plusieurs défis techniques majeurs. La synchronisation des cycles de reproduction et de prédation, ainsi que la gestion simultanée des interactions dans des zones fortement peuplées, ont entraîné des comportements imprévus et des oscillations dans la dynamique des populations. Pour pallier ces problèmes, le projet a adopté une démarche itérative d'ajustement des paramètres, en affinant notamment les valeurs de `herbivore_energy_gain` et `carnivore_energy_gain`, ainsi que les taux de reproduction. Une refonte partielle de certaines portions du code dans la fonction `step` a permis de renforcer les contrôles sur la mortalité et de stabiliser les cycles de reproduction. Par ailleurs, une étroite collaboration entre les responsables de la dynamique des ressources et de la modélisation animalière a permis d'assurer une cohérence globale, en synchronisant les cycles biologiques avec les variations environnementales. Cette synergie technique a été essentielle pour résoudre les conflits de dépendance et garantir une évolution harmonieuse de l'écosystème simulé.

En vue de renforcer encore la fidélité et la complexité de la simulation, plusieurs axes d'amélioration sont envisagés. Parmi ceux-ci, l'intégration d'algorithmes d'intelligence artificielle, notamment des techniques de renforcement, permettra aux agents d'apprendre

et d'ajuster dynamiquement leurs stratégies de déplacement, de prédation et de reproduction. L'ajout de modèles de migration saisonnière et de dispersion génétique offrirait une dimension supplémentaire à la simulation, en introduisant des variations spatiales et temporelles plus réalistes. De plus, l'implémentation d'un système de communication entre agents pourrait simuler des comportements d'intelligence collective, ouvrant ainsi la voie à l'émergence de stratégies de coopération ou de compétition avancées. Enfin, la mise en place d'un tableau de bord interactif pour le monitoring en temps réel, combiné à une documentation technique exhaustive et évolutive, renforcerait l'utilité du simulateur comme outil de recherche en écologie théorique et comme support pédagogique. Ces perspectives d'amélioration témoignent de l'engagement du projet à évoluer constamment pour répondre aux défis complexes de la simulation d'écosystèmes.

5.4 Philippe RASTOUL

Pour commencer à créer l'interface graphique, il fallait disposer d'une librairie permettant de créer une application comportant des boutons et une plateforme avec laquelle l'utilisateur peut interagir. Macroquad est une bibliothèque Rust dédiée à la création de graphiques 2D, conçue pour être simple, rapide et efficace, tout en permettant de développer des applications graphiques en 2D de manière intuitive. Macroquad est basée sur des abstractions relativement simples, ce qui permet aux développeurs de se concentrer rapidement sur la logique de l'application sans se soucier des détails de bas niveau des moteurs graphiques. Macroquad permet également de gérer facilement les entrées utilisateur, comme les clics de souris ou les touches du clavier.

Une fois la bibliothèque téléchargée, la configuration des options d'interaction dans le simulateur peut être mise en place. D'abord, la création de la fenêtre s'effectue en utilisant la fonction `clear_background()` qui nettoie le fond d'écran avec la couleur correspondante, puis la fonction `draw_text()` qui permet d'afficher les caractéristiques de la simulation de base à des coordonnées précises. Par ailleurs, nous avons choisi d'afficher les mots dans le menu avec une police de taille 20, car c'est celle qui offrait la meilleure expérience, étant suffisamment grande pour être lue tout en permettant de créer la liste des caractéristiques en une seule colonne. Ensuite, il y a `next_frame().await` qui attend la fin de l'affichage de l'image actuelle et se prépare pour l'image suivante.

Au début de la simulation, un menu de configuration vous est présenté, permettant de définir divers paramètres pour la simulation de l'écosystème. Ces paramètres contrôlent le comportement des plantes, des herbivores et des carnivores dans la grille de simulation. Le menu de configuration fournit une interface intuitive avec des champs ajustables, chacun représentant un aspect clé de la simulation.

Paramètres configurables :

Plantes initiales : Définit le nombre de plantes initiales dans la simulation.

Plantes vert foncé initiales : Définit le nombre de plantes vert foncé initiales dans la simulation.

Herbivores initiaux : Définit le nombre d'herbivores au début de la simulation.

Carnivores initiaux : Définit le nombre de carnivores dans la population initiale.

Taux de croissance des plantes : Ajuste le taux de croissance des plantes au fil du temps. Cette valeur est un nombre flottant.

Taux de reproduction des herbivores : Définit le taux de reproduction des herbivores. Un taux plus élevé signifie plus de descendants.

Gain d'énergie des herbivores : Définit l'énergie qu'un herbivore gagne en consommant des plantes.

Perte d'énergie des herbivores : Spécifie la perte d'énergie des herbivores à chaque étape, représentant leurs déplacements, leur métabolisme et leurs actions.

Énergie initiale des herbivores : Définit le niveau d'énergie initial des herbivores au début de la simulation.

Seuil de reproduction des herbivores : Définit le seuil d'énergie nécessaire pour que les herbivores se reproduisent.

Taux de reproduction des carnivores : Ajuste le taux de reproduction des carnivores.

Gain d'énergie des carnivores : Définit la quantité d'énergie qu'un carnivore gagne en consommant des herbivores.

Perte d'énergie des carnivores : Définit la perte d'énergie des carnivores à chaque étape.

Énergie initiale des carnivores : Définit l'énergie initiale des carnivores.

Seuil de reproduction des carnivores : Définit le seuil d'énergie pour la reproduction des carnivores.

Chacun de ces champs peut être modifié directement dans le menu de configuration à l'aide de votre clavier, grâce aux fonctions `is_key_pressed()` et `is_mouse_button_pressed()`. Vous pouvez :

- Utiliser les touches Flèche haut et Flèche bas pour naviguer entre les champs.
- Modifier les valeurs en tapant des nombres, y compris des valeurs décimales pour les paramètres flottants.
- Appuyer sur Retour arrière pour supprimer les caractères et corriger les erreurs.
- Appuyer sur Entrée pour confirmer votre sélection et démarrer la simulation avec les paramètres choisis.

Une fois la configuration terminée et confirmée, la simulation commencera avec les valeurs personnalisées. Une fois la simulation lancée, vous pouvez explorer sa progression à l'aide de la fonction de navigation dans l'historique. L'historique est constitué de captures d'écran de l'écosystème à chaque étape de la simulation, vous permettant de revenir en arrière ou d'avancer dans le temps.

Contrôles de navigation dans l'historique :

Flèche gauche (←) : Naviguez vers l'étape précédente dans l'historique de la simulation.

Flèche droite (→) : Déplacez-vous vers l'étape suivante dans l'historique de la simulation. S'il n'y a pas d'étape future, la simulation avancera d'une étape et une nouvelle capture d'écran sera créée dans l'historique.

Barre d'espace : En maintenant la barre d'espace enfoncée, la simulation avancera continuellement, étape par étape, en créant de nouvelles captures d'écran dans l'historique.

Échap : Quitte la simulation et retourne au menu de configuration.

Chaque étape de l'historique contient des données sur l'état de l'écosystème, y compris les positions et les niveaux d'énergie des plantes, des herbivores et des carnivores. En naviguant dans l'historique, vous pouvez examiner comment la simulation évolue au fil du temps.

Pendant que la simulation est en cours, vous pouvez interagir avec l'écosystème et contrôler divers aspects de la simulation.

Interaction avec la grille :

La grille de simulation est composée de cellules, chacune représentant un emplacement dans l'écosystème où les agents (plantes, herbivores ou carnivores) peuvent résider. Vous pouvez interagir avec la grille de plusieurs manières :

Clic de souris : Un clic sur une cellule de la grille vous permettra de suivre un agent spécifique (herbivore ou carnivore) à cet endroit. Le statut de l'agent sélectionné, comme sa position et son niveau d'énergie, sera affiché pour le suivi.

Suivi des agents :

Clic gauche de la souris : Lorsque vous cliquez sur un emplacement de la grille contenant un agent (soit un herbivore, soit un carnivore), cet agent sera sélectionné pour

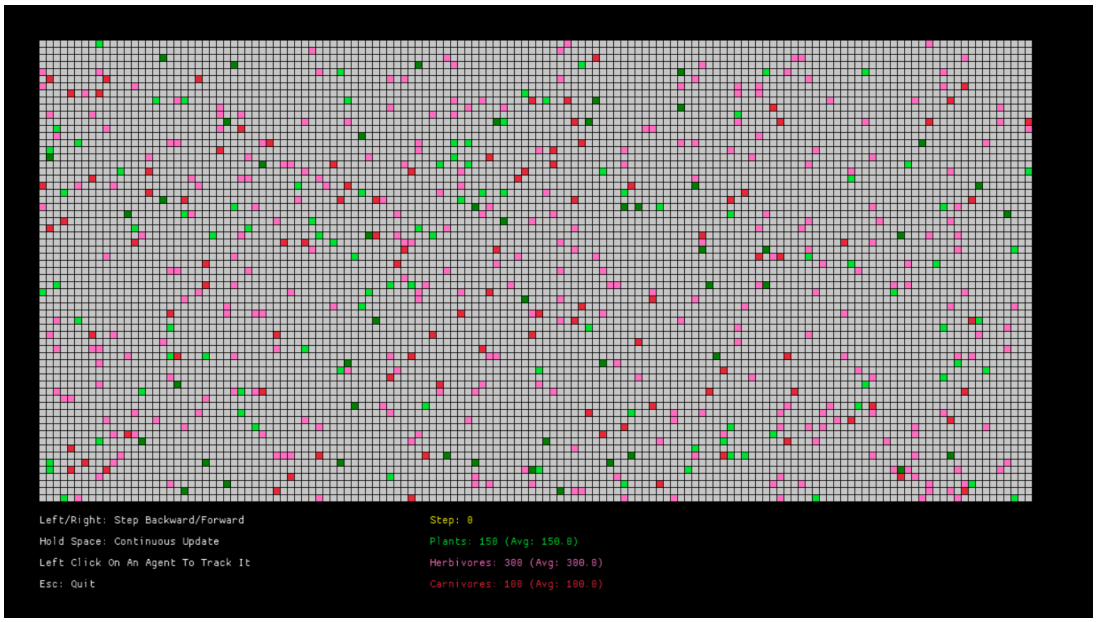
être suivi. Les détails de l'agent, y compris sa position, son niveau d'énergie et son statut de décès (le cas échéant), seront affichés à l'écran.

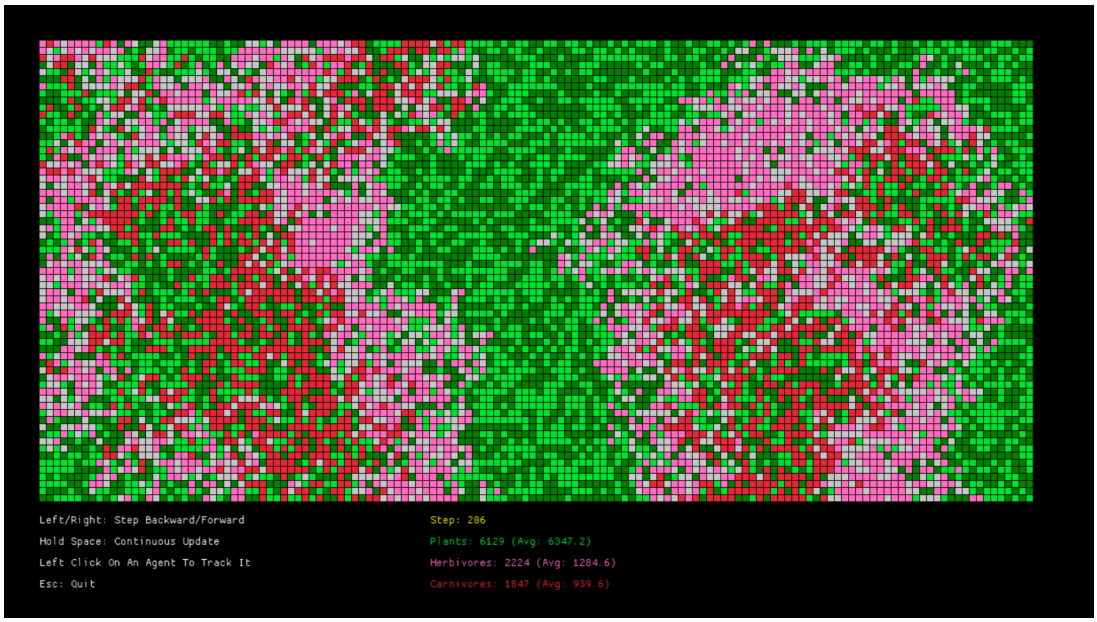
Une fois qu'un agent est suivi, ses détails sont affichés sur le côté de l'écran :

— **Né** : La position d'origine où l'agent est né.
— **Position** : La position actuelle de l'agent dans la grille.
— **Énergie** : Le niveau d'énergie actuel de l'agent, qui diminue avec le temps ou les actions.
— **Mort** : Si l'agent meurt, la cause de la mort sera affichée (par exemple, « Épuisement d'énergie »). Si l'agent meurt (par exemple, à cause d'un manque d'énergie ou d'autres raisons), la cause de la mort sera indiquée à côté de ses détails suivis.

Nous pensons que nous avons réussi à implémenter la base des éléments avec lesquels on peut interagir dans la simulation. Il reste cependant à implémenter un bouton pour revenir au menu, puis à améliorer l'esthétique de l'application en rajoutant potentiellement des couleurs pour différencier les différents agents modifiables ou pour rendre l'application simplement plus présentable et plaisante lorsque l'on interagit avec le menu ou les boutons.

6 Visuels du projet




Left/Right: Step Backward/Forward
Hold Space: Continuous Update
Left Click On An Agent To Track It
Esc: Quit
Step: 206
Plants: 6129 (Avg: 6347.2)
Herbivores: 2224 (Avg: 1284.6)
Carnivores: 1847 (Avg: 939.6)


Step: 206
Plants: 6129 (Avg: 6347.2)
Herbivores: 2224 (Avg: 1284.6)
Carnivores: 1847 (Avg: 939.6)
Tracked Agent Info:
Born: (98, 26)
Position: (98, 26)
Energy: 61
Died: Not Yet

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 :** 80% **En cours**
- **Implémentation des plantes :** 60% **En cours**
- **Implémentation des animaux :** 30% **En cours**
- **Tests unitaires :** 20% **En cours**
- **Documentation :** 10% **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** qui puisse reproduire, de la manière la plus fidèle et la plus flexible possible, les dynamiques complexes régissant la vie de multiples espèces (animales et végétales) au sein d'un même environnement. L'enjeu est de **fournir un outil de recherche, d'enseignement et d'apprentissage** dont la richesse fonctionnelle permette aussi bien de tester des hypothèses écologiques pointues que d'illustrer de manière ludique et visuelle les principes fondamentaux de la biologie et de l'écologie.

Pour atteindre cet objectif, notre projet se fonde à la fois sur **des bases théoriques solides** (modèles mathématiques et approches multi-agents) et sur **une architecture logicielle performante et sécurisée** (grâce au langage Rust). Cette double approche offre plusieurs avantages :

- **Réalisme des interactions :** La modélisation s'appuie sur des mécanismes biologiques reconnus (prédation, compétition, symbiose, pollinisation, etc.) et sur la prise en compte d'éléments environnementaux (climat, ressources, catastrophes naturelles). Ainsi, les scénarios produits peuvent s'approcher des dynamiques réellement observées sur le terrain.
- **Évolutivité et modularité :** La flexibilité de la plate-forme permet d'ajouter ou de retirer des espèces, de configurer différents paramètres (taux de reproduction, vitesse de croissance, etc.) et de simuler des situations variées (introduction d'une espèce invasive, disparition d'un prédateur, variation du climat). Cette adaptabilité rend la simulation utile pour une large gamme de projets de recherche ou d'études de cas pédagogiques.
- **Performance et fiabilité :** Le choix du langage Rust garantit une gestion fine des ressources et une exécution rapide, même pour des simulations à grande échelle (nombre important d'agents, multiples interactions simultanées). De plus, Rust apporte une sécurisation mémoire et une robustesse qui facilitent la maintenance et la durabilité du projet.

Du point de vue pédagogique, le simulateur propose une **visualisation claire et interactive** du fonctionnement d'un écosystème. Les étudiants peuvent ainsi manipuler différents paramètres (disponibilité des ressources, taux de mortalité, pression de prédation) et observer en temps réel l'émergence ou la disparition d'espèces, la modification des chaînes alimentaires, ou encore la fluctuation des populations au fil des saisons. Cette manipulation concrète permet de dépasser le cadre théorique pour développer **une compréhension intuitive et expérimentale** des phénomènes biologiques et écologiques.

Du point de vue de la recherche, le simulateur offre un **terrain d'expérimentation virtuel** pour tester et affiner des hypothèses écologiques. Les chercheurs peuvent mettre en place des scénarios complexes (multiples espèces, grande variété d'interactions, événements aléatoires) et analyser de manière détaillée l'influence de chaque paramètre. Les résultats obtenus peuvent ensuite nourrir des modélisations plus spécifiques ou orienter de futures études de terrain, permettant un gain de temps et de ressources considérable.

Ainsi, en **conciliant précision scientifique, performance logicielle et ergonomie**, ce projet s'impose comme un outil **transdisciplinaire** à fort potentiel. Il ambitionne d'être **accessible** (open source, compatible Linux) et **facilement appropriable** par un large public (enseignants, étudiants, chercheurs, curieux de nature). La **dimension éducative** est soutenue par des modules de visualisation didactiques, tandis que la **dimension**

scientifique trouve sa force dans la possibilité de repousser les limites de la complexité modélisée et du réalisme des scénarios explorés.

En définitive, la **création de ce simulateur** répond à un double besoin : **améliorer la compréhension des systèmes écologiques** et **développer un outil novateur, performant et collaboratif**, susceptible d'éclairer les défis contemporains en matière d'écologie et de biodiversité.

Bilan global : La démarche adoptée jusqu'à présent a permis de réaliser des avancées significatives dans la conception et la mise en place des différents modules de la simulation. Cependant, quelques retards subsistent sur certains aspects, principalement dus à la complexité technique et à la nécessité de multiples itérations pour atteindre une cohérence globale. Les prochaines étapes consisteront à finaliser et à tester les modules restants, en vue d'une version intégrée et fonctionnelle.