

Projet : Évolution

Liste des fonctionnalités implémentées :

L'univers est constitué d'animaux (moutons, loups) et de végétaux (herbe, sels minéraux).
Le déplacement des animaux est aléatoire.
Les fonctions vitales tels que l'alimentation, la reproduction ou encore la mort des animaux sont gérées.
Les deux types d'interfaces (graphique et texte) sont disponibles.
Un système de sauvegarde/chargement est accessible depuis le jeu.
Il est également possible de mettre la simulation en pause et de la reprendre.
La vitesse d'exécution de la simulation est réglable (de 0,5 à 5 secondes par tours).
Le rythme de reproduction est un paramètre modifiable depuis le menu.

Fonctionnement du projet :

Le projet doit initialement être compilé grâce à la commande *javac Main.java*. Après avoir lancé le projet grâce à la commande *java Main*, une fenêtre représentant le menu principal du jeu va apparaître. Il sera alors possible de modifier les différents paramètres du jeu avant de commencer la partie, ou de charger une partie déjà existante.

Remarque : Le projet terminera son exécution si les paramètres ne sont pas correctes ou si le fichier contenant la sauvegarde d'une partie n'est pas valide.

Une fois que les paramètres sont confirmés, la fenêtre du menu principal se fermera pour laisser sa place au plateau de jeu constitué de la simulation et de différentes options. Par défaut, la vitesse de la simulation est au minimum (5 secondes entre chaque tours). Il est alors possible de modifier la vitesse du jeu grâce au *slider* situé en dessous la simulation. À gauche de la simulation sont disponibles plusieurs options :

- Possibilité de mettre en pause la simulation (lorsqu'elle est en pause, le même bouton servira à la reprendre ;
- Possibilité de revenir au menu principal pour redéfinir les paramètres ;
- Possibilité de sauvegarder/charger une partie ;
- Possibilité de quitter la partie en cours.

Remarque : Lors du retour au menu principal ou à la fermeture du jeu par le bouton *Quitter*, toute partie non sauvegardé sera perdue. Lors de la sauvegarde ou du chargement d'une partie, la simulation se mettra automatiquement en pause le temps du choix du fichier à sauvegarder/charger.

Erreurs éventuelles :

Le projet risque de ne pas fonctionner correctement dans les cas suivant :

- Si le répertoire *Sprite* n'est pas situé dans le même répertoire que celui de *Main.java* ;
- Si les paramètres tels que la taille de l'univers ou le nombre d'animaux ne sont pas correctes (nombre d'animaux supérieur à la taille de l'univers par exemple) ;
- Si le nombre d'animaux est très élevé par rapport à la taille de l'univers, certains animaux n'auront probablement aucune possibilité de déplacement, ils seront alors considérés comme morts ;
- Si le nombre d'animaux est très élevé par rapport à la taille de l'univers, certaines naissances ne pourront pas avoir lieu, dû à un possible manque de cases libres. La naissance n'aura donc pas lieu.
- Si la taille de l'univers est extrêmement élevée, la simulation sera quasiment illisible (et peut causer certains problème comme le dépassement de la taille du type int).

Classes utilisées :

Classe Main (ligne 38) : La classe principale qui permet d'ouvrir un nouveau menu.

Classe Menu (ligne 51) : La fenêtre du menu principal. La gestion de la fenêtre se fait dans la sous-classe *Fenetre* (ligne 62). La gestion des *threads* se fait dans la sous-classe *Tlauch* (ligne 134).

Classe Plateau (ligne 163) : La fenêtre du plateau de jeu, contenant la simulation. La gestion de la fenêtre se fait dans la sous-classe *Fenetre* (ligne 247).

Classe *Univers* (ligne 1095) : L'univers de la simulation.

Classe *Animal* (ligne 1100) : Hérite de la classe *Univers*.

Classe *Mouton* (ligne 1168) : Hérite de la classe *Animal*.

Classe *Loup* (ligne 1181) : Hérite de la classe *Animal*.

Classe *Vegetal* (ligne 1194) : Hérite de la classe *Univers*.

Classe *Herbe* (ligne 1199) : Hérite de la classe *Vegetal*.

Classe *Sels* (ligne 1204) : Hérite de la classe *Vegetal*.