Synthèse du projet informatique

# Simulation d’Ecosystème

## Sommaire

### -Partie Organisation

-Ce que nous voulions

-Répartition des taches

-Difficultés rencontrées

### -Partie Informatique

-Architecture du projet

-Comment faire fonctionner le projet

### -Sources

## Organisation :

* L’Objectif premier du projet était de parvenir à avoir une modélisation dynamique d’un écosystème dont l’utilisateur ne visualise que le résultat. Un résultat parfait aurait été une interface sur laquelle l’utilisateur définie ses règles et sur lequel il peut interagir en temps réel. Nous donc défini nous même des règles basique sur lesquelles commencer notre code.

Règle :

Un environnement composé d’herbe, de forêt et d’eau généré aléatoirement. Chaque type de terrain à une influence sur ceux qui le parcourent.

Les différentes espèces (herbivore, carnivore et charognard) y sont placées aléatoirement.

Le déplacement de ces dernières doit se faire de manière logique, ainsi un herbivore ira là où il y a à manger, un carnivore où il y a des proies herbivores et un charognard cherchera des cadavres.

Les zones d’eau ne peuvent être occupées.

S’ils le peuvent, les animaux se reproduisent.

* Pierre et moi ayant des méthodes de code très différentes, il a été décidé de séparé le travail très distinctement et de ne pas intervenir dans le travail de l’autre sauf si nécessaire. Ainsi Pierre s’est occupé de la partie génération de l’environnement ainsi que de la gestion de l’interface tandis que je m’occupais des interactions entre espèces puis de la mise en commun des deux codes pour fournir l’outil final.
* Les premières difficultés rencontrées ont été la modélisation des interactions environnement/espèce. Notre visualisation initiale était en effet inverse au principe final. Nous avons du nous forcer par soucis de complexité à imaginer un environnement agissant sur des espèces plutôt que des espèces se déplaçant sur un environnement. La seconde difficulté fut la prise en main et l’installation des packages pour générer les images et GIF de rendu. Finalement, la complexité des codes nous à couter un certain temps lors de la phase de contrôle des fonctions. L’utilisation de boucle étant systématique pour le parcours de l’environnement, certain bugs n’ont été vu que tardivement.
* Ce projet nous a permis d’appréhender toute la programmation objet avec nos méthodes de code ainsi que l’utilisation de module axé interface. De plus la phase de débug à l’aide de pydev debugger fut d’une grande utilité.

## Partie informatique

### Architecture :

Le dossier se présente sous la forme d’un script lançant la simulation puis de package comprenant les objets et les fonctions leur étant propre, de plus le dossier Résultats comporte après utilisation du programme les images générées par tour, la base de donnée du dernier tour ainsi que le GIF récapitulatif.

Ainsi on trouve par dossier :

-BDD : La requête permettant de sauvegarder l’ensemble des espèces avec les terrains sur lesquels elles se situent et leurs coordonnées sur l’environnement. Nous n’avons pas pu écrire le script permettant de générer un environnement par rapport à cette base de données par manque de temps.

-Environnement : Contient les scripts des classes Environnement et Terrain.

Terrain comporte la classe abstraite terrain dont hérite les classe Foret Rivière et Herbe, une liste d’espèces occupe le terrain ou est vide lorsque non occupé.

Environnement et la combinaison d’une partie génération avec génération du terrain par watershading et attribution dans une matrice d’objet terrain et d’une partie interactions entre les espèces avec les fonctions regarder, se\_deplacer\_manger, vivre et tour\_brut.

* espece : Contient le script de la classe Espece avec les fonctions lui étant interne, c'est-à-dire celle jouant sur son état indifféremment du terrain sur lequel il se trouve.
* Interface : Contient les script qui vont être affichés à l’utilisateur que ce soit du texte (interface utilisateur) ou des images (test\_utilitaire) ou des GIF(Thegif)
* Simulation : contient le script effectuant l’ensemble des taches.

## Application

Pour se servir du simulateur il faut lancer le script main\_appel.py.

Même si un traitement d’erreur à été effectué, il se peut que le programme termine dès le lancement avec ce message « Process finished with exit code -1073741571 (0xC00000FD) »

Le programme ayant une grande complexité, ce dernier s’auto termine pour ne pas nuire à la machine. Cette erreur dépendant des capacités de la machine on ne peut pas prédire quand elle aura lieu ou non.

Après terminaison normale le dossier Résultat contiendra un gif ainsi qu’une base de données. Les images pouvant être en grand nombre elles sont dans un sous dossier.

Bug de dernières minutes, les animaux ont tendance à trop se reproduire entrainant des arrêts du programme.