



Projet SIMULION

Membres du groupe :

- MEDJDOUB Karim
- OKAT Elias
- SANGARE N'famara

Encadrant :

- JACQUIER Yohann

Présentation générale :

SIMULION est un programme qui vise à simuler un écosystème composé de plusieurs animaux afin d'observer leurs comportements et interactions.

Les bibliothèques qui ont été utilisées pour la réalisation de ce projet sont :

- **tkinter** : pour la gestion des fenêtres (fenêtre principale, fenêtre d'aides, fenêtre de résultats).
- **pygame** : pour l'affichage de la simulation.
- **matplotlib** : pour la réalisation du graphique qui expose les résultats en fin de simulation.
- **random** : pour la gestion du pseudo aléatoire.
- **math** : pour utiliser quelques fonctions mathématiques.

Manuel utilisateur :

- Pour lancer le programme il suffit d'exécuter le fichier main.py, il faudra avoir au préalable les bibliothèques citées précédemment installées sur votre machine.
- Une page HELP est disponible lors du lancement du programme pour expliquer comment utiliser ce dernier.
- Le programme fonctionne sous Windows et Linux, cependant pour une meilleure expérience il sera préférable de l'exécuter sous Windows (quelques bugs d'affichage sous linux).

- Python n'étant pas un langage très optimisé pour les programmes gourmands, je vous invite à ne pas abuser sur le nombre d'animaux dans la simulation (au risque d'avoir quelques lags et ralentissements du programme).

Fonctionnalités :

Voici la liste détaillée des fonctionnalités développées :

Fenêtre d'accueil : fenêtre d'accueil qui sert d'introduction à l'application composée de 3 parties. L'écran principal qui sert d'accueil, l'écran d'aides qui fournit les aides nécessaires à la bonne utilisation de l'application, ainsi que l'écran de lancement qui sert à paramétrer la simulation.

Interactions intelligentes : les animaux interagissent les uns avec les autres ainsi qu'avec l'environnement (attaque, reproduction, alimentation, déplacements, mort).

Affichages dynamiques : les éléments constituant de la simulation possèdent tous des affichages dynamiques basés sur une horloge interne à leurs sprites.

Affichages différenciés : les éléments constituant de la simulation posséderont des rendus différents en fonction de plusieurs facteurs (genre de l'animal, mutation génétique qu'il porte, etc.).

Caméra dynamique : l'utilisateur peut déplacer la caméra pour recouvrir une grande zone de la map.

Système de fullscreen : si l'utilisateur le souhaite il peut lancer la simulation en plein écran.

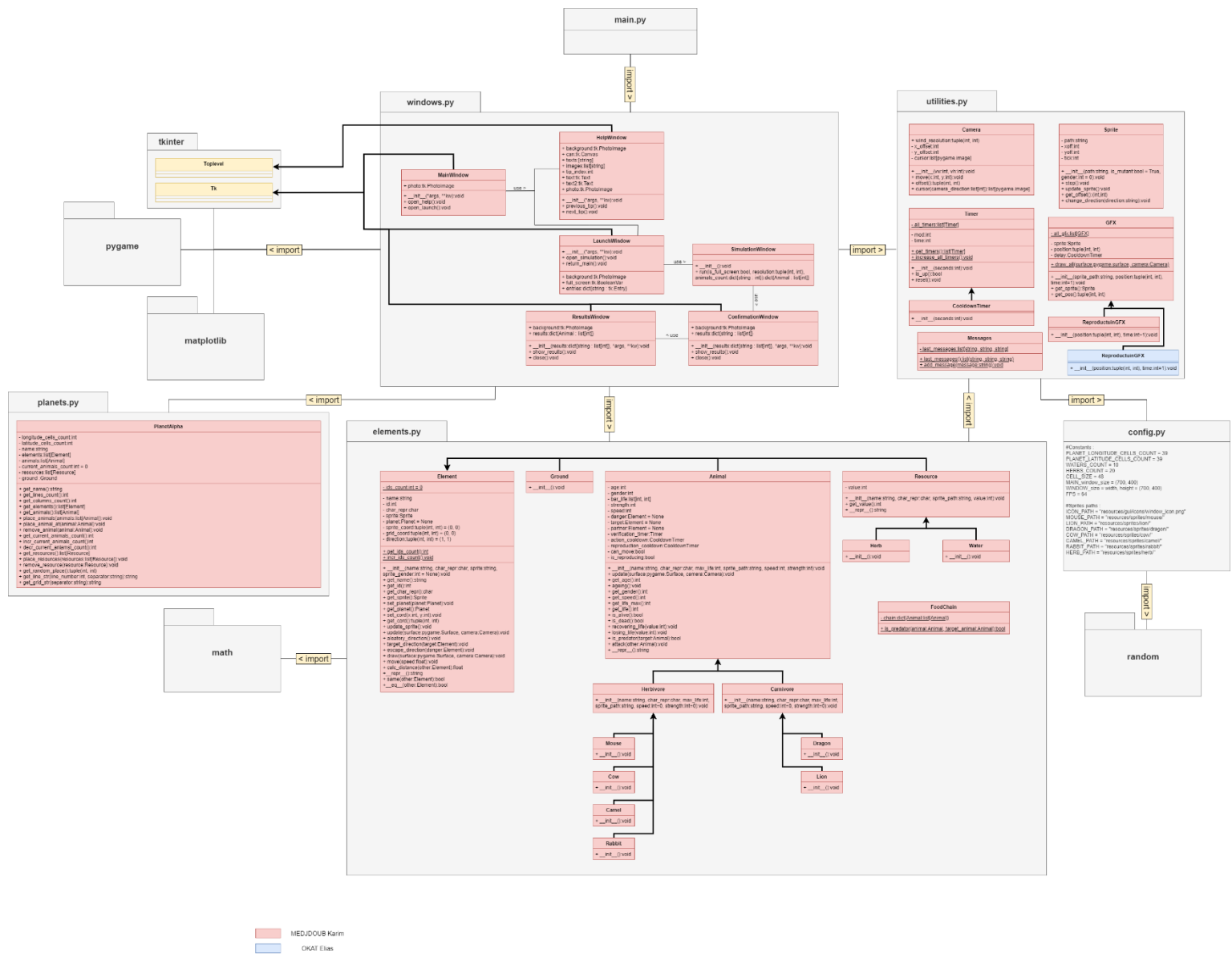
GUI : affichage des informations importantes dans la fenêtre du jeu (âge de la planète, nombre d'animaux, nombre d'images par seconde, informations sur la simulation, etc.).

Mode apocalypse : mode qui transforme la planète en un décor post apocalyptique.

Résultats de simulation : graphique en fin de simulation qui résume l'évolution des populations d'animaux au fil du temps.

Diagramme des classes correspondantes aux fonctionnalités :

(Voir les clés en bas du diagramme pour connaître l'auteur des classes)



Une copie du diagramme sera remise avec ce rapport pour une meilleure lisibilité.