

Projet POO:
Modèle Proies-Prédateurs
Cahier de Spécifications

Historique des révisions

Date	Description et justification de la modification	Auteur	Pages/ chapitre	Edition/ Revision
12/02/2018	Création	Betti		0.0
18/02/2018	Révision suite feedback	Betti		1.0

Table des matières

1. Introduction	3
2. Exigences opérationnelles	3
3. Interfaces	4
4. Exigences Fonctionnelles	5
4.1 Initialisation d'une simulation	5
4.2 Éléments d'une simulation	5
4.2.1 Terrain	6
4.2.2 Espèces	6
4.2.3 Animaux	6
4.3 Déroulement d'une simulation	6

1. Introduction

Les spécifications présentées dans ce document visent à décrire les fonctionnalités et les objectifs de l'application qui sera développée à la demande du client. Cette application permettra de simuler un modèle proie-prédateur simpliste et d'observer ladite simulation.

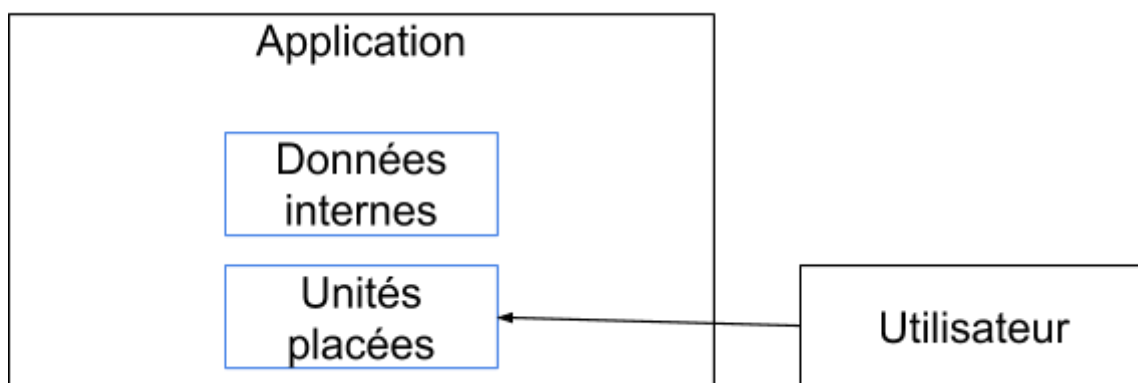
Les fonctions principales de cette application sont la création des proies et prédateurs sur un terrain, ainsi que la simulation au tour par tour de l'évolution de leurs populations, qui pourra être suivie grâce à l'affichage de l'état du terrain à la fin de chaque tour. L'utilisateur aura un certain contrôle sur le déroulement de la simulation.

2. Exigences opérationnelles

L'utilisateur se voit proposer une application *standalone*, simple d'utilisation.

Les contraintes de performance sont quasi inexistantes: l'application tourne sur un ordinateur *mainstream* (2 Go de Ram, 10 Go d'espace disque, processeur mono ou multicore à 2 Ghz) sans la moindre difficulté, attendu que la majeure partie du temps est passée à attendre entre deux tours successifs (ce point dépendant bien évidemment de la vitesse de la simulation).

Il n'y a pas de contraintes de sécurité, ni d'intégrité.



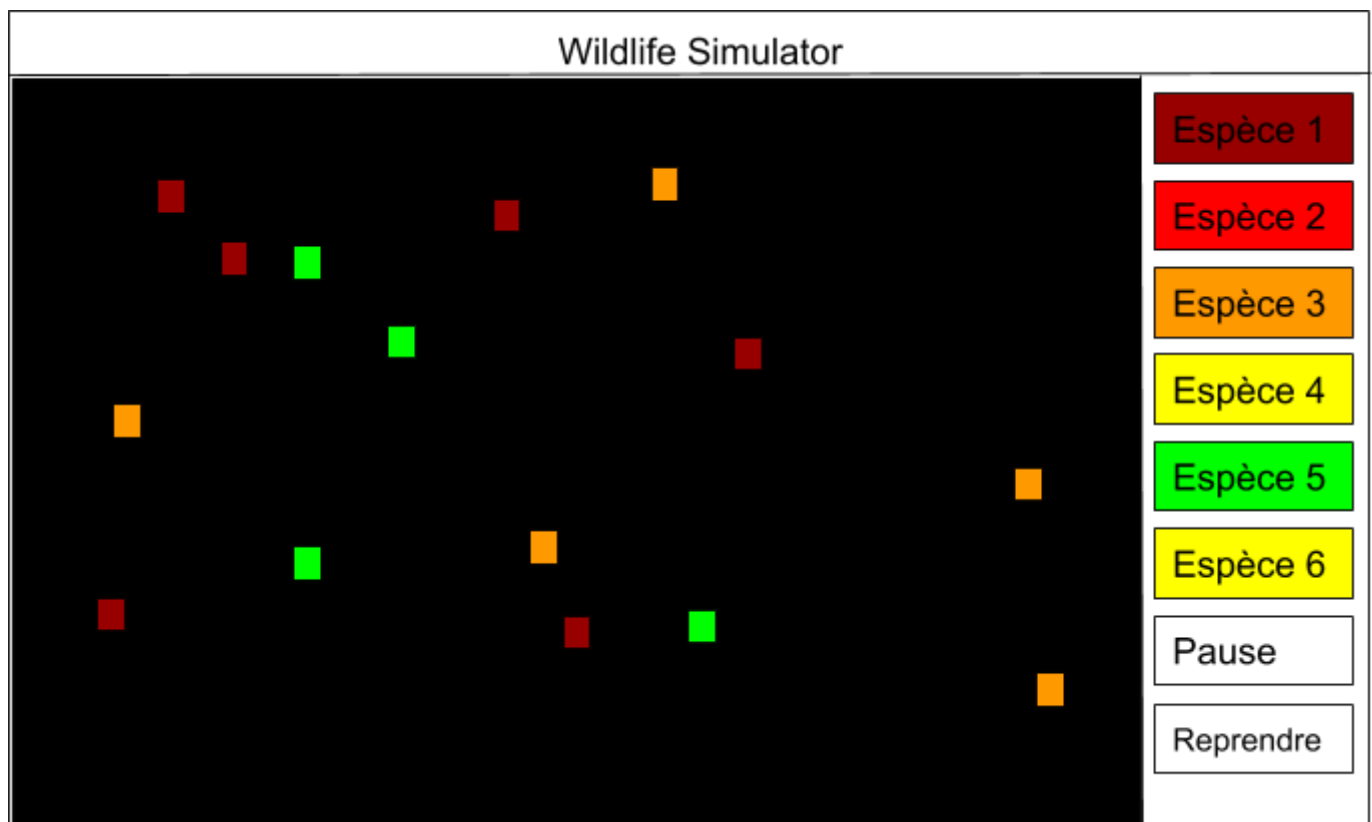
La *simulation* en elle-même est autonome et ne demande pas d'interaction postérieure à la phase d'initialisation, mais l'on souhaite garder la possibilité d'influer partiellement sur son déroulement si nécessaire.

Afin de permettre une éventuelle réutilisation du travail qui aura été fourni, l'application devra être divisée en parties les plus indépendantes possibles. Notamment, la partie logique gérant les tours, les interactions et les différentes espèces devra être indépendante de l'affichage.

3. Interfaces

L'interface homme-machine sera simple, intuitive et la plus explicite possible afin qu'aucune phase de formation ne soit nécessaire à l'utilisation de l'application.

Il n'y a à priori pas de besoin de sauvegarder ou de restaurer des données.



Exemple de fenêtre de l'application

4. Exigences Fonctionnelles

Le coeur de l'application sera basé sur la simulation d'un modèle de prédation, incluant différentes espèces pouvant représenter des proies, des prédateurs ou les 2 ,se déplaçant dans un monde fermé. La simulation se terminera dès lors que toutes les unités seront décédées.

4.1 Initialisation d'une simulation

Comme dans toute simulation, la première étape est bien évidemment l'initialisation, qui devra se dérouler de la manière la plus intuitive possible directement dans la fenêtre de l'application. Un système de drag & drop depuis une liste des différentes espèces ou de point & click est fortement recommandé.

4.2 Éléments d'une simulation

Une simulation fera interagir les différents éléments suivants:

4.2.1 Terrain

Le terrain sera un univers fermé au sein duquel évolueront les différentes espèces. Il pourra éventuellement comporter des zones plus ou moins accessibles, et divers types de ressources.

4.2.2 Espèces

Différentes espèces au comportement prédéterminé (proie, prédateur, ou les deux) seront disponibles. Chaque espèce pourra disposer, si son comportement le permet, d'une liste d'espèces qu'elle essaiera de chasser, et d'une liste d'espèces qu'elle aura tendance à fuir. Chaque espèce aura un ensemble de caractéristiques propres, comme une espérance de vie, une vitesse de déplacement, et un champ de vision.

4.2.3 Animaux

Chaque animal représenté appartiendra à une unique espèce, et héritera des caractéristiques de son espèce. Il pourra en outre vieillir, et son intérêt pour ses prédateurs ainsi que sa condition pourront éventuellement évoluer au cours du temps.

4.3 Déroulement d'une simulation

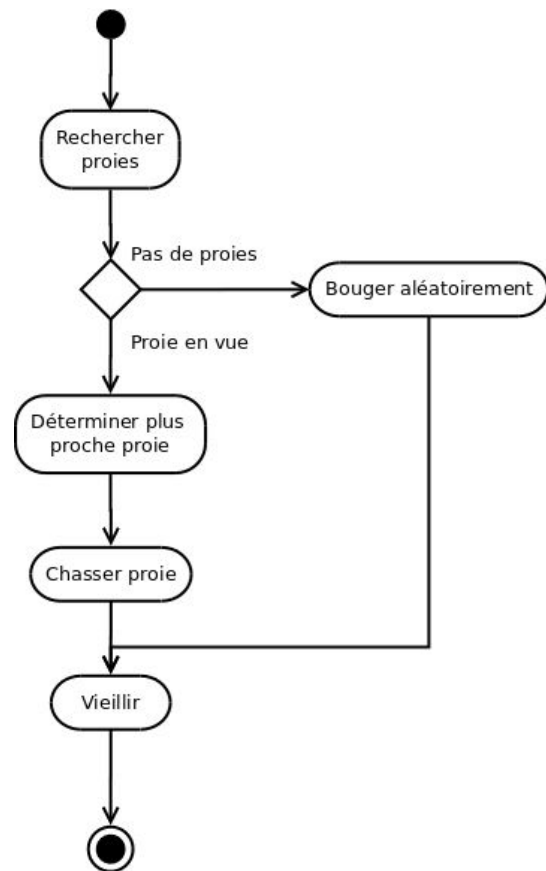
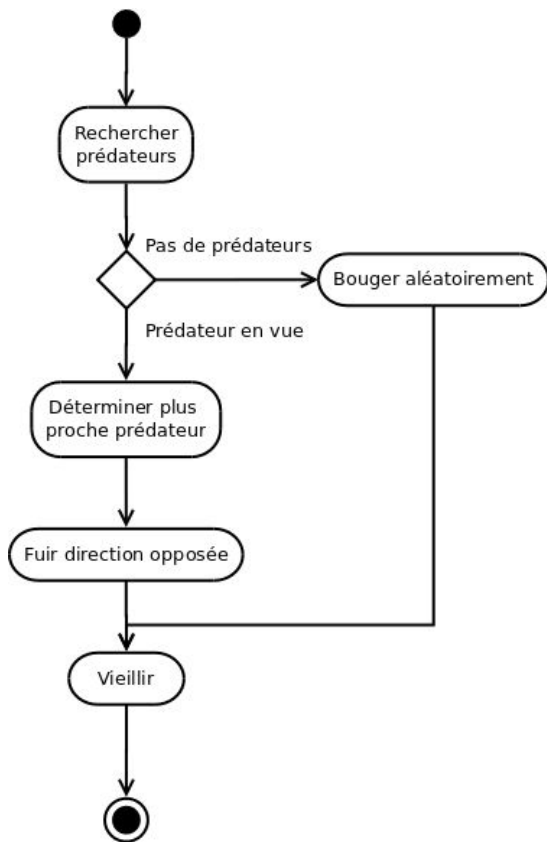
La simulation se déroule selon un système tour par tour, de manière autonome. Elle s'achève lorsqu'il n'y a plus d'animaux sur le terrain. Les unités peuvent interagir de différentes manières. Un tour d'une unité passe par les étapes suivantes:

- détection des unités (proies et/ou prédateurs) géographiquement proches
- mouvement adapté à cette détection:
 - fuite si un animal d'une espèce crainte est présent
 - chasse d'une proie si l'une d'entre elles est à portée
- si rien n'a été détecté, mouvement aléatoire à la découverte du monde

Pour une proie:

Pour un prédateur:

Modèle Proies-Prédateurs



Modèle Proies-Prédateurs

Pour un animal étant à la fois proie et prédateur:

