

Groupe Segfault

Cahier des Charges



Introduction/

Le but de ce projet est de réaliser un simulateur d'évolution accompagné d'un système d'édition de la simulation pour permettre de recréer une multitude de scénarios avec un unique programme.

Les simulations d'évolutions sont relativement nombreuses et visent des domaines différents de l'évolution. On y retrouve les modèles usuels comme ceux de dynamique de population, de dérive génétique ou encore de sélection naturelle et artificielle. À ces modèles viennent s'ajouter d'autres moins communs comme l'héritage de paterne ou encore l'automate cellulaire. On remarque cependant que parmi toutes ces simulations le modèle est fixe et ne peut pas être modifié, mis à part ces paramètres.

Ainsi, en l'état, créer un nouveau modèle de simulation requiert soit de partir de zéro, soit de se baser sur un ancien modèle et modifier son code.

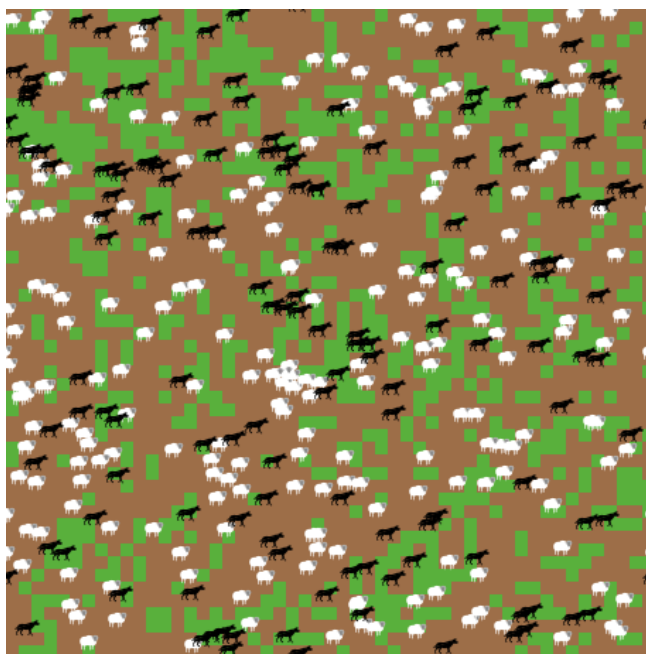


Image du modèle de prédation mouton/loup

La grande majorité de ces simulations utilisent des algorithmes évolutionnistes reposant sur la programmation génétique. Comme son nom l'indique, elle se base sur le principe de sélection naturelle pour permettre aux agents d'évoluer et de s'adapter à la situation. La principale force de cette technique est sa proximité avec la réalité, ce qui permet de rendre la simulation plus réaliste. Cet avantage vient avec l'inconvénient d'être plus coûteux en temps de calcul.

Sommaire/

- I/ Fonctionnement de la simulation
 - a. Généralités
 - b. Les agents
 - c. Les maladies
 - d. Le terrain
- II/ Génération de l'environnement
 - a. Le terrain
 - b. La nourriture
- III/ Moteur de la simulation
- IV/ Interface
- V/ Librairies utilisées
- VI/ Répartitions des tâches

I/ Fonctionnement de la simulation

a. Généralités

Comme le décrit l'introduction, nous allons simuler une population qui évolue selon son environnement. Pour cela nous allons avoir besoin de plusieurs éléments pour rendre la simulation la plus fidèle possible. Tout d'abord nous allons concevoir des agents, qui seront nos populations que nous allons simuler dans un même environnement. En parlant d'environnement nous allons avoir besoin d'un terrain pour faire évoluer nos agents. Puis nous allons aussi concevoir des maladies, qui pourront alors se propager (ou pas !) parmi nos agents.

b. Les agents

Les agents seront les individus que nous voulons simuler. Prenons un exemple, si nous voulons simuler la relation prédatrice entre les loups et les moutons, nous ferons alors deux sortes d'agents avec deux sets différents de caractéristique. Ces caractéristiques seront prédéfinies par nos soins, c'est-à-dire que nous voulons que les agents aient au minimum ces caractéristiques :

Caractéristique :	Description :
Espèce	Nom de l'espèce
Temps de vie maximum	Temps de vie (en secondes)
Énergie	Points de vie de l'agent en quelque sorte. L'agent meurt s'ils tombent à 0 (Max : 100). Il pourra se reproduire s'ils dépassent un certain seuil qui sera déterminé.
Rayon de détection de nourriture	Rayon dans lequel l'agent détectera sa nourriture
Nourriture(s)	Type de nourriture dont se nourrit l'agent (pouvant elle-même être un autre agent)
Vitesse	Vitesse de déplacement de l'agent
Type de reproduction	Sexué (requiert deux agents pour produire un ou plusieurs agents) ou asexué (besoin d'un seul agent qui va se multiplier)
Taux de reproduction	Nombre de descendants lors de la reproduction (approximatif)
Résistance	Valeur entre 0 et 1, déterminant la préférence de température du climat (0 = Froid, 1 = Chaud)

À noter que chacun de ces paramètres peut être nul. Si nous reprenons l'exemple cité plus haut avec les moutons et les loups, si nous voulons ajouter un autre agent, par exemple de l'herbe, alors sa caractéristique nourriture sera nulle, car ils n'auront pas besoin de se nourrir, pareil pour la détection de nourriture. Tous ces paramètres seront initialisés par l'utilisateur avant que la simulation commence. L'intelligence artificielle des agents s'occupera de leur comportement (s'il faut aller chercher de la nourriture, s'il faut s'accoupler, ou s'il faut s'éloigner parce qu'un prédateur arrive).

C. Les maladies

Il n'y aura pas seulement cette relation de proie/prédateur que nous voulons simuler, nous allons aussi créer des maladies qui pourront ajouter un peu de piment à cette simulation. Tout comme les agents, les maladies auront des caractéristiques :

Caractéristique :	Description :
Nom	Nom de la maladie
Cible	Cible de la maladie (Transmissible via des "non-cibles" sans qu'elles subissent les effets de la maladie)
Type de transmission	Moyen de transmission parmi : <ul style="list-style-type: none"> - par aérosol - par reproduction - par proximité au moment de la mort d'un agent infecté - par consommation d'un autre agent infecté (qu'il subisse la maladie ou non)
Taux de Guérison	variable entre 0 et 100, qui détermine la probabilité qu'un agent se rétablisse de la maladie à chaque seconde
Taux de Létalité	variable entre 0 et 100, qui détermine la probabilité qu'un agent meure de la maladie à chaque seconde.
Taux de Transmission	variable entre 0 et 100, qui détermine la probabilité qu'un agent contracte la maladie à chaque seconde

d. Le terrain

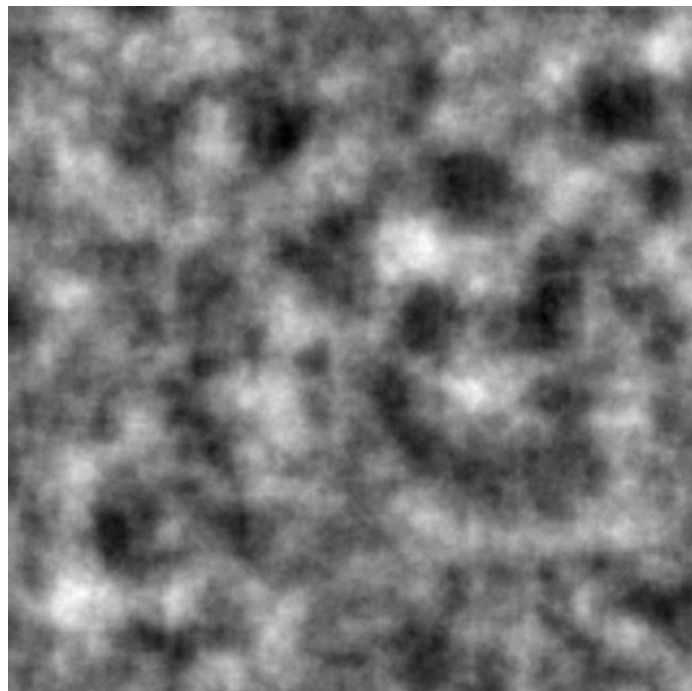
Le terrain sera simplement composé de deux éléments principaux, il y a le climat de chaque zone du terrain déterminé au lancement de la simulation par un générateur de terrain, puis il y a la génération de nourritures. Ces deux éléments sont décrits avec plus de détails juste après.

II/ Génération de l'environnement

a. Le terrain

L'environnement correspond au terrain dans lequel nos agents vont évoluer. Celui-ci est composé de trois types de climats : le climat chaud, le climat froid ou le climat tempéré. Ces trois climats vont alors influencer les agents et la nourriture s'y trouvant (en influençant par exemple le taux de reproduction, la vitesse de déplacement, etc...).

Pour générer ce terrain, nous allons utiliser le Bruit de Perlin (ou Perlin Noise en anglais). Lors de la génération du Bruit de Perlin, les valeurs se rapprochant d'un (noir) correspondront aux zones chaudes, celles se rapprochant de zéro (blanc) correspondront aux zones froides et les nuances entre les deux, seront les zones tempérées.



Exemple de Bruit de Perlin

b. La nourriture

La nourriture, qu'on peut assimiler à un genre de plante, va donc pousser sur ce terrain. Comme vu précédemment, la nourriture permet de remplir l'énergie des agents herbivores. Pour permettre la survie d'une espèce, les plantes doivent donc se propager. Pour ce faire, si une plante reste un certain temps (défini à l'avance) en vie, alors une autre plante poussera à côté. Ce temps est défini pour la zone tempérée et celui pour les deux autres zones, est calculé en fonction de ce temps de base. Plus le climat est extrême, plus le temps à attendre est long (ceci dépendant bien entendu de la résistance de cette plante).

III/ Moteur de la simulation

Notre simulation sera rendue grâce à la bibliothèque SDL. Grâce à SDL, nous allons pouvoir assez simplement créer une boucle principale qui nous permettra de mettre à jour les états de nos agents, maladies et de notre environnement. De plus, nous souhaiterions ajouter une fonctionnalité de play/pause à notre simulation et de pouvoir accélérer ou ralentir la simulation.

IV/ Interface

L'interface contient l'éditeur de la simulation qui permet de modifier la simulation pour créer un modèle particulier. C'est avec elle que l'utilisateur pourra créer de nouveaux agents dans la simulation. Comme on peut le voir dans le croquis ci-dessous, nous pouvons créer un agent puis par la suite éditer ses paramètres pour représenter le modèle.

<div>Agent 1</div> <div>Agent 2</div> <div>+</div>	<h3>Paramètres</h3> <div>Paramètre 1</div> <div>Paramètre 2</div>
LANCER SIMULATION	

Croquis de l'interface

Ainsi, pour recréer le modèle de prédation mouton/loup précédemment énoncé, nous créons un agent mouton qui a comme paramètre de nourriture l'herbe, puis un agent loup, qui lui a pour paramètre de nourriture l'agent mouton. Puis on peut ensuite lancer la simulation et observer son déroulement.

Une fois la simulation terminée, une nouvelle interface s'offre à l'utilisateur lui permettant de visualiser l'évolution de la simulation sous forme de graphes. Ces graphes représentent l'évolution du nombre d'individus au sein d'une population et l'évolution de leurs caractéristiques

V/ Les librairies utilisées

Pour ce projet, nous avons décidé d'utiliser SDL et GTK 3.0. Ces librairies nous permettront de gérer facilement les interfaces utilisateur grâce à GTK 3.0 et le dessin grâce à SDL.



Logo de SDL



Logo de GTK 3.0

VI/ Répartition des tâches

Tâche	BIGORRE	CAPMARTIN	DURFORT	MASSET
Interface				x
moteur		x		
simulation	x		x	
terrain :	x	x		
Site Web				x

VII/ Planning

Première soutenance :

- Interface : un premier prototype
- moteur fait à 100%
- simulation : agents ainsi que leurs comportements déjà fini
- terrain : un premier prototype (au moins génération de nourriture)
- site web : un premier prototype à compléter

Deuxième soutenance : projet fin

Conclusion/

Notre simulation n'est évidemment pas représentative de la réalité et beaucoup d'approximations sont faites. Cependant, nous espérons pouvoir mener à bien ce projet et observer des résultats intéressants sur l'évolution de nos populations d'individus.

Sources/

https://www.wikiwand.com/fr/Algorithme_%C3%A9volutionniste

<http://ccl.northwestern.edu/rp/beagle/index.shtml#:~:text=Simulated%20Evolution%20is%20the%20umbrella,and%20learn%20about%20evolutionary%20processes.>