
	<div>Mini Projet C++ 2017</div> <div>ISEN CIR2 Brest</div>	
---	--	---

## Lions et gazelles

Ce projet a pour objectif la réalisation en langage C++ d'une simulation graphique de relation prédateur/proie.

En partant de la partie fonctionnelle de votre TP 5 : une classe Lion et une classe Gazelle ou les lions peuvent manger des gazelles, vous devrez aboutir à une application en Qt où, en plus de l'affichage graphique du monde et du déplacement des animaux, vous devrez mettre en place des *widgets* de contrôle (boutons, *sliders*,...) permettant de paramétrer l'application.

### Cahier des charges minimal :

- Partie fonctionnelle : la taille du monde (2D) est définie au départ et configurable. Le monde est peuplé d'un nombre défini au départ (configurable) d'animaux, soit lions soit gazelles (la proportion de lions et de gazelles est aussi fixée au départ et configurable). Les animaux ont au départ une « énergie » initiale (ou points de vie) qu'ils perdent au fil du temps, sauf s'ils mangent (ils gagnent alors de l'énergie). Si l'énergie devient nulle, l'animal meurt et disparaît du monde. Une simulation tour par tour fait bouger les animaux de façon autonome. Quand un lion et une gazelle sont à la même position, le lion mange la gazelle (« prédation ») : elle meurt et le lion gagne de l'énergie. Le nombre de lions et de gazelles vivants ainsi que de gazelles mangées doit être connu en permanence (affichage dans la console au minimum).
- Affichage graphique du monde : les graphismes du « fond » du monde et des animaux sont choisis pour qu'un utilisateur comprenne immédiatement de quoi il s'agit. Les déplacements doivent être fluides et la « prédation » doit être bien explicite.
- Affichage graphique de contrôle :
  - les *widgets* de contrôle obligatoires sont :
    1. un bouton de démarrage de la simulation
    2. un bouton « Quitter »
    3. un « *slider* » de réglage de la vitesse
    4. une case à cocher pour passer en plein écran (décocher fait retourner en affichage normal).
  - Vous pouvez ajouter des *widgets* de contrôle facultatifs tels que des contrôles du nombre d'animaux initial, de la proportion lion/gazelle,...
- Configuration : le nombre d'animaux initial, la proportion lion/gazelle, l'énergie initiale, la taille du monde en X,Y doivent être configurables, soit par des *widgets* de contrôle (cf. ci-dessus), soit par un fichier de configuration, soit par des paramètres de la ligne de commande.

**Vos ajouts (facultatifs) :**

- Toute amélioration de l'«intelligence» des animaux est la bienvenue (par exemple, déplacement non aléatoire, mais fonction des prédateurs/proies les plus proches).
- Ajout d'une fonctionnalité « naissance » : si l'énergie dépasse un certain seuil, des petits peuvent naître...
- Le monde peut être plus complexe qu'un simple fond : présence d'obstacles (rochers, rivières,...) éventuellement configurables par fichier.
- L'affichage des variables nombre de lions et de gazelles vivants, nombre de gazelles mangées peut se faire dans l'interface graphique au lieu de la console, sous forme texte ou graphe.
- L'ajout de musique ou sons est possible.
- Autres ajouts selon votre imagination, à partir du moment où l'idée générale de l'application est respectée (éviter les super-héros ou les vaisseaux spatiaux par exemple, ça ne colle pas trop...).

**Contraintes de programmation :**

- Le programme devra s'exécuter sous Linux.
- Le programme devra bien sûr être conçu selon une conception «objet» et les règles de bonne programmation (pas de variables globales, nom des variables explicite,...) devront être respectées.
- Votre code doit être commenté et lisible (indenté, aéré,...).

**Livraisons et tests demandés :**

- Ce projet est noté individuellement. Les échanges d'informations (verbales) sont autorisés, mais tout échange ou recopie de code est interdit. Je passe des outils de comparaison de code et anti-plagiat/Internet. Vous risquez un 0/20.
- Les sources et fichiers nécessaires à la compilation et l'installation devront être fournis. La qualité du programme source sera notée.
- Une démonstration du programme aura lieu en fin de projet. Elle permettra de produire une partie de la note sur la qualité graphique et le bon fonctionnement des éléments de contrôle et de configuration demandés.

**Déroulement :**

- pas de créneaux de cours spécifiques en dehors des séances indiquées ci-dessous
- 2 séances encadrées prévues le 25 et le 28 avril, mais vous devrez déjà avoir bien avancé le projet pour que ces séances soient profitables.
- Démonstration et livraison lors de la dernière séance, le 9 mai.

**Barème :**

- le barème précis vous sera donné ultérieurement.
- le principe : le respect du cahier des charges minimal et de la qualité du code est noté sur 20. Vos ajouts sont des bonus. Dans le meilleur des cas vous pouvez donc avoir plus de 20, et cette note peut bonifier votre moyenne d'année en C++ (ex : S1 12/20, S2 24/20 => année 18/20), qui ne pourra cependant pas excéder 20!