

Projet Logiciel Transversal

Élevage Numérique

M. GOSSELIN

Sommaire

I. Présentation du logiciel.....	3
Fonctionnement du logiciel.....	3
Environnement de jeu.....	4
Ressources.....	4
II. États du programme.....	5
Description des états, Conception logiciel.....	5
III. Rendu du Logiciel.....	7
Conception Logiciel.....	7
Diagramme des classes de rendu.....	7

I. Présentation du logiciel

Nom du logiciel : Élevage Numérique

Principe du logiciel : Le logiciel Élevage Numérique est un logiciel de gestion d'élevage ou de populations précises d'animaux.

Fonctionnement du logiciel

Un fermier (ou même un zoo), élève des animaux d'une race bien précise.

Il doit donc pouvoir gérer la population d'animaux de cette race, les nourrir, les vendre, les faire s'accoupler en évitant un maximum les problèmes de consanguinité, acheter d'autres animaux, gérer les frais vétérinaires, tuer les bêtes malades ou âgées, tout en assurant un gain maximal (pour un zoo, il n'y a évidemment pas de gain possible mais que des dépenses).

À l'initialisation du logiciel, il faut créer une nouvelle population d'animaux. Il est possible de créer plusieurs populations d'animaux.

Quand on crée une population d'animaux, il faut renseigner :

- la race animale élevée,
- les animaux qui appartiennent à cet élevage.

Si la race de cet élevage, n'a jamais été élevée dans la ferme, il faut alors créer une fiche de renseignements sur la race en question, comprenant :

- le nombre moyen de petits qui naissent dans une portée,
- le poids moyen d'un mâle et d'une femelle adultes de cette race,
- l'espérance de vie de l'espèce (pour le mâle et la femelle),
- l'âge moyen de maturité de cette race,
- la durée moyenne de gestation de la race,
- le prix d'une bête à la vente (pour une bête enfant, et adulte)
- le prix d'une ration de nourriture par unité de temps pour chaque bête,
- le nombre N de générations minimal pour que la présence d'un ancêtre commun entre deux individus de sexes différents ne crée pas de réel problème génétique, dans le cas d'accouplement.

Ensuite, pour chaque individu de cette espèce animale, il faut renseigner l'âge et le sexe. Un numéro est attribué à chaque animal automatiquement.

L'éleveur peut décider à tout instant de vendre ou tuer un animal, d'acheter un nouvel animal pour enrichir le patrimoine génétique du cheptel, ou de faire se reproduire des animaux de sexes opposés (si les deux bêtes à accoupler ont un ancêtre commun, l'accouplement est impossible).

Quand deux animaux s'accouplent, les animaux résultant de cet accouplement, sont automatiquement ajoutés à la base de donnée, au bout de la durée de gestation (dans un but de simplification, le nombre d'animaux qui naissent est systématiquement le même et correspond toujours au nombre moyen d'enfants par portée pour cette race animale, le sexe des animaux nés, est attribué aléatoirement ou défini manuellement).

Quand tous les animaux d'un élevage sont tous supprimés de la base de donnée (tous vendus ou morts), l'élevage est automatiquement supprimé du logiciel.

Il est possible de créer un élevage d'une espèce déjà exploitée dans le passé, il n'est alors pas nécessaire de créer une fiche de renseignements pour cette race.

L'unité de temps est le mois.

Le logiciel doit pouvoir gérer au moins 10 espèces animales différentes composées au minimum chacune de 500 individus.

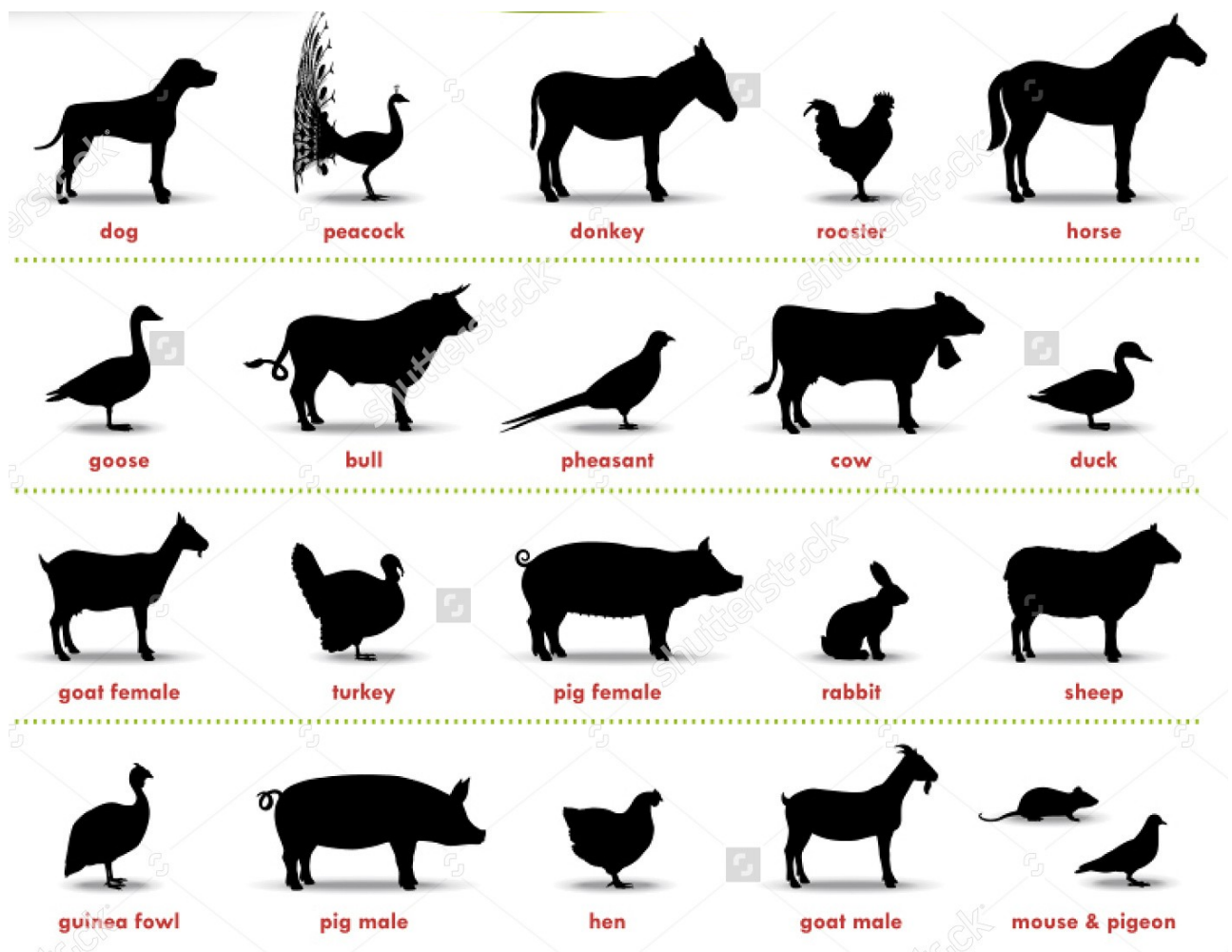
Pour simplifier les données :

On considère au début, que les animaux ont tous le même âge et n'ont aucun ancêtres communs (donc qu'on peut commencer à reproduire les animaux ensembles dès le lancement du programme), et qu'il n'y a aucune différence de poids moyen adultes et d'espérance de vie, entre mâles et femelles.

Environnement de jeu

L'environnement de développement du logiciel est NetBeans IDE.

Ressources



II. États du programme

Description des états, Conception logiciel¹

Le logiciel présenté ici sert à une Ferme. La classe **Ferme** est caractérisée par les élevages qui la composent, les espèces d'animaux élevés dans ces élevages, et son budget.

La classe **Elevage** est un ensemble d'animaux d'une même espèce. Chaque élevage qui compose la ferme est caractérisé par un identifiant, une espèce animale (ou race) à laquelle appartient tous les animaux de l'élevage, et par les animaux qui composent l'élevage

La classe **Race** (ou espèce) est celle qui représente les caractéristiques générales de tous les animaux de cette race, comme par exemple espérance de vie moyenne des mâles et des femelles de cette race, un poids moyen des bêtes adultes mâles/femelles, etc.

Une Race animale est donc caractérisée par des données qui lui sont propres.

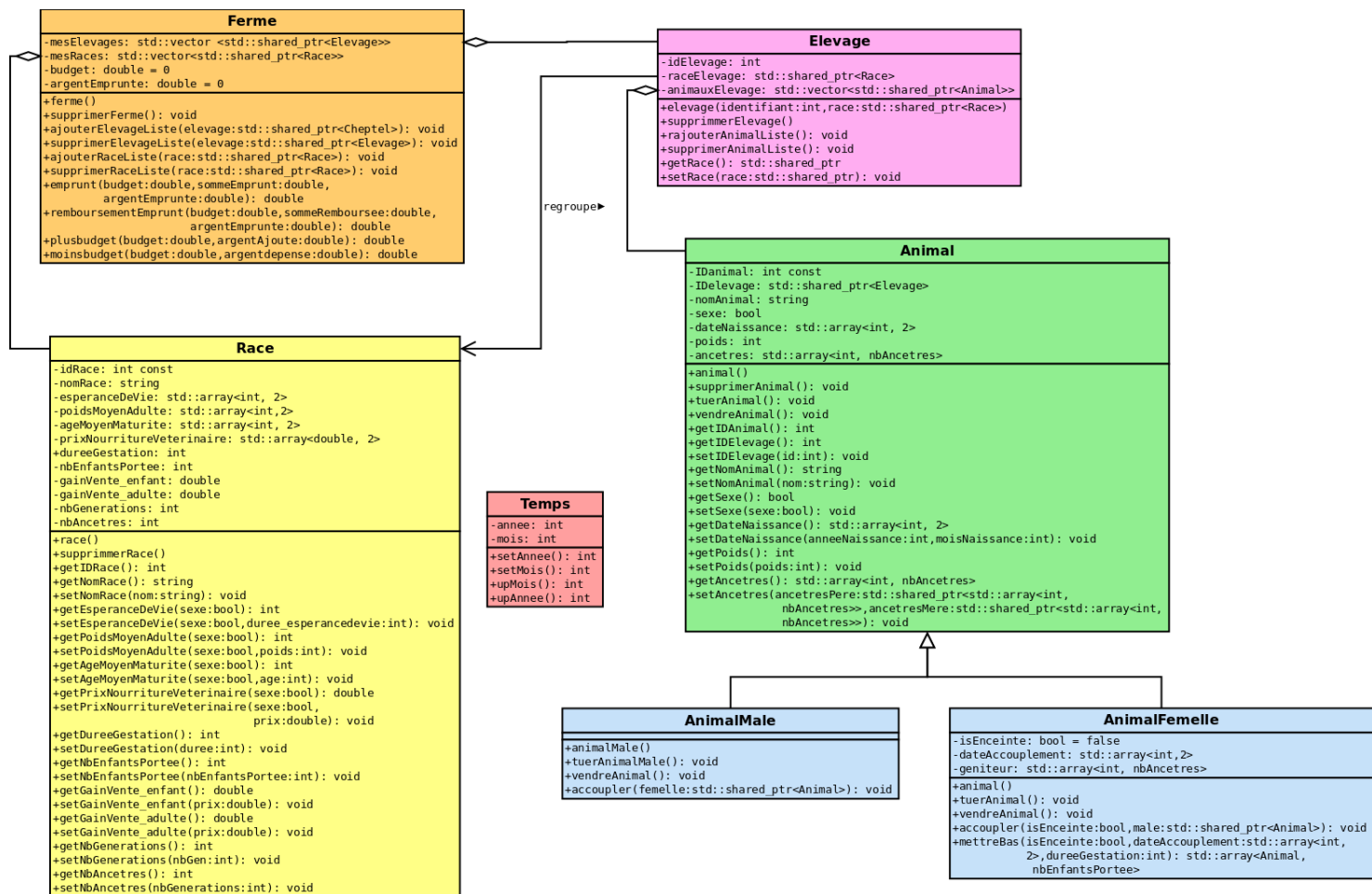
Comme une élevage ne peut contenir qu'une race, il était question d'intégrer le contenu de la classe **Race** dans la classe **Elevage**, mais il doit être possible d'arrêter un élevage, puis de recommencer un élevage de cette même race sans avoir besoin de renseigner à nouveau toutes les données de cette race.

La classe **Animal** est celle qui définit chaque bête de la ferme par ses données personnelles : son âge, son poids, etc. Chaque bête appartient à un seul élevage. Les données regroupées dans la classe animal ne font pas de différences entre les sexes, ce sont les données que tous les animaux ont.

Les animaux de sexes différents ont en plus des attributs et méthodes différentes : les classes **AnimalMale** et **AnimalFemelle** diffèrent par le fait qu'une femelle peut mettre au monde des petits.

Une classe **Temps**, gère la date, ce qui permet de faire évoluer les données de chaque animal.

¹ Le logiciel n'étant pas un jeu vidéo, il n'y a pas d'états du jeu à proprement parler, comme ce qu'on peut trouver sur l'exemple donné sur Moodle



III. Rendu du Logiciel

Le logiciel de gestion d'élevage, comporte une interface graphique dite « classique », ou interface « desktop ». Pour ce faire, j'utilise le framework Qt, qui convient parfaitement à ce genre d'application, et plus particulièrement la classe Qwidget, et nombreuses de ses classes filles. Quand on lance le logiciel, une fenêtre principale s'ouvre. Elle contient une barre de Menu, une barre de Statut, une Toolbox. Au centre de la fenêtre (donc ce qui est la véritable raison du logiciel) se trouvent la liste des élevages contenus dans la ferme (à gauche de l'écran), et la description graphique de l'élevage (à droite de la liste), qui apparaît sous forme de tableau. Quand on crée un nouvel élevage, une nouvelle fenêtre apparaît, et permet de sélectionner des caractéristiques de l'élevage (comme par exemple la race animale qui sera élevée dans cet élevage). Il faut ensuite directement ajouter des animaux à cet élevage (car un élevage sans animal ne peut exister). Quand on veut créer un nouvel animal à cet élevage, une nouvelle fenêtre apparaît. C'est une fenêtre de création d'animal dans laquelle on renseigne des informations sur l'animal. Si, quand on crée un élevage, la race qu'on élève est inconnue, une nouvelle fenêtre apparaît, dans laquelle on enregistre toutes les caractéristiques de cette race, desquelles dépendront tous les animaux de l'élevage.

Conception Logiciel

Fenêtre principale : Le cœur du rendu réside dans cette classe, car elle crée la fenêtre principale du logiciel. Elle hérite des propriétés de la classe QMainWindow proposée par le framework Qt. Son unique méthode est son constructeur, qui construit aussi les éléments graphiques de la fenêtre.

Les classes **NouvelleFenetreBete**, **NouvelleFenetreRace**, **NouvelleFenetreElevage**, ont pour rôle de créer les fenêtres qui servent à créer les nouveaux élevages, les nouvelles races élevées, et les nouvelles bêtes dans les élevages. Ces classes ne contiennent que leurs constructeurs respectifs, qui leur permet de se créer, ainsi que les Widgets qu'elles contiennent.

Diagramme des classes de rendu

