

- *“La thématique du jeu proposée est celle d’une jardinerie : le joueur contrôle des jardiniers qui plantent et récoltent des fleurs de différents types pour composer des bouquets. Les bouquets rapportent de l’argent qui permettent d’embaucher plus de jardiniers ou d’acheter de nouvelles graines pour faire plus de bouquets. Malheureusement, les vaches du voisinage raffolent des fleurs et vous devrez les empêcher de venir ruiner votre jardinerie.”*
- Le projet utilise le langage Java avec Java Swing comme outil pour développer une interface interactive pour ce programme.

---

## 2. Analyse globale

- Interface graphique interactive
  - + Conception d'une interface utilisateur intuitive et esthétiquement agréable, permettant une interaction fluide avec le jeu. Les joueurs pourront observer en temps réel l'évolution de leur jardinerie, sélectionner et diriger les jardiniers, et accéder à des informations détaillées sur les différents éléments du jeu.
  - + Difficulté moyenne, priorité 1
  
- Gestion de l'état du jeu
  - + Implémentation d'un système robuste pour suivre avec précision tous les aspects du jeu, y compris les états dynamiques des unités, les interactions avec l'environnement, et les événements aléatoires affectant la jardinerie.
  - + Difficulté moyenne, priorité 1
  
- Système d'animation utilisant Thread
  - + Utilisation efficace des threads pour animer les différents éléments du jeu, assurant une expérience visuelle dynamique et réactive aux actions des joueurs.
  - + Difficulté moyenne, priorité 2
  
- Système sonore (à implémenter) - difficulté moyenne, priorité 3
- Système d'effets spéciaux (à implémenter) - difficulté moyenne, priorité 3

---

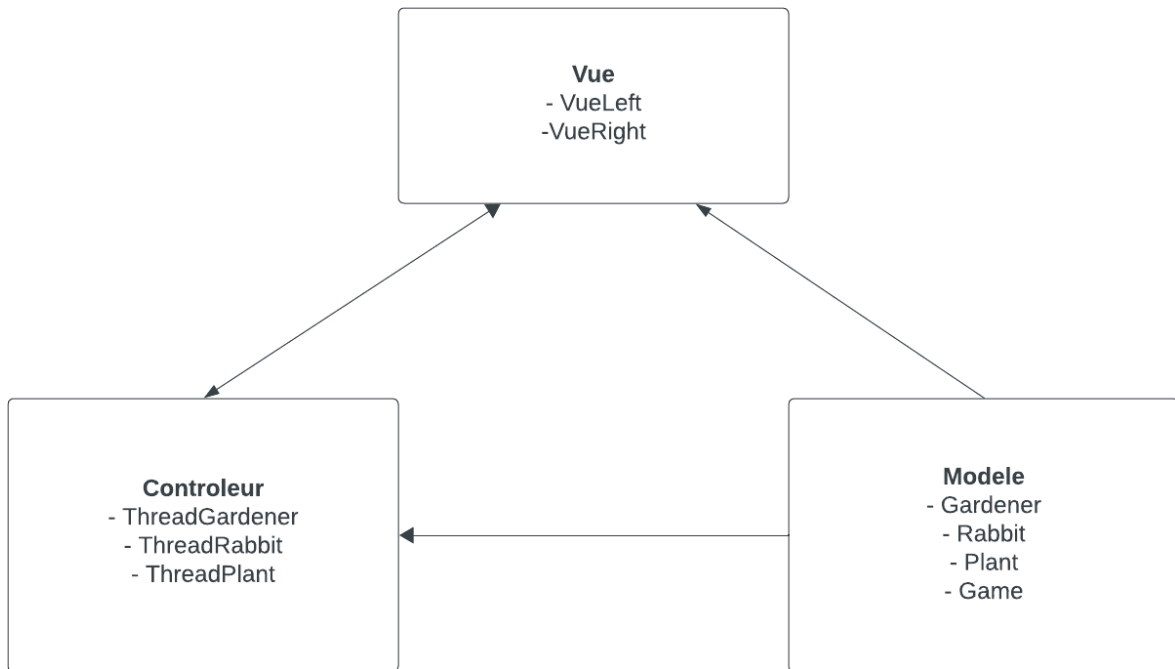
### 3. Plan de développement

- Interface graphique interactive : 4 heures
- + Mise en place des éléments graphiques pour les unités, interface utilisateur : 2 heures
- + Recherche d'une Font adaptée à l'esthétique du design : 15 minutes
- + Utilisation de Java Swing pour configurer les fenêtres d'affichage du jeu : 15 minutes
- + Utilisation de Java Swing pour la mise à jour dynamique du jeu : 1 heure
- + Écouteurs de souris pour enregistrer l'interaction de l'utilisateur : 30 minutes
  
- Gestion de l'état du jeu : 2.5 heures
- + Mise en place de la logique de base du jeu : 2 heures
- + Utilisation de Thread pour gérer les mises à jour du jeu et les changements : 30 minutes
  
- Système d'animation utilisant Thread : 1 heure
- + Mise en place des éléments graphiques pour l'animation : 1 heure
- + Utilisation de Thread pour ajouter de l'animation au jeu : 10 minutes
  
- Système sonore (à implémenter)
- Système d'effets spéciaux (à implémenter)
- Rédaction du document analyse et conception : 2 heures
- Mise en place une page Github pour l'hébergement de projet : 15 minutes

---

## 4. Conception generale

Le patron MVC



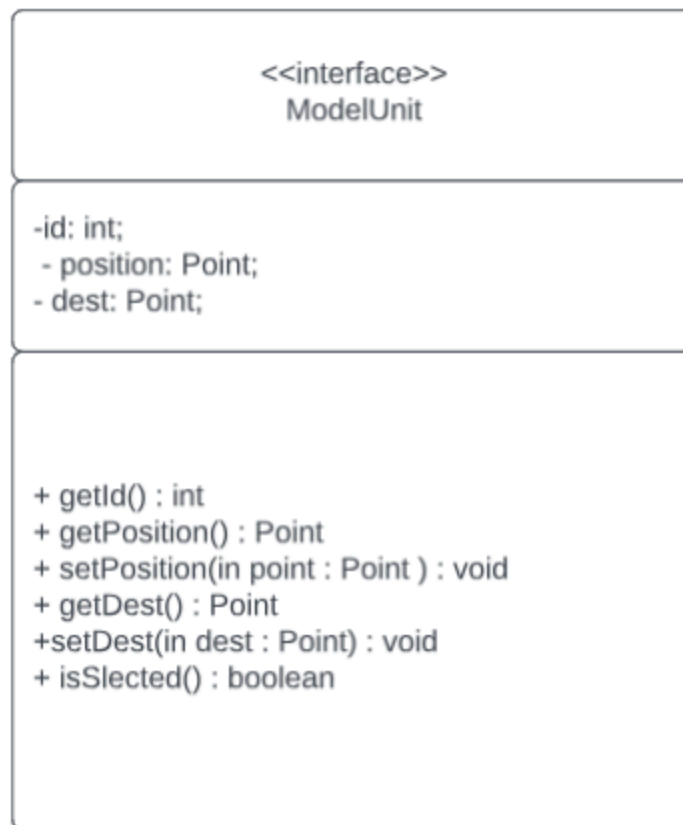
---

## 5. Conception détaillée

### A) Modèle

#### 1. Interface Unit

- utilisée comme base pour d'autres types d'objets Unité
- chaque Unité a un identifiant int, une position Point, une destination Point et un booléen isSelected avec leurs propres accesseurs et mutateurs
- Diagramme:



---

## 2. Gardener



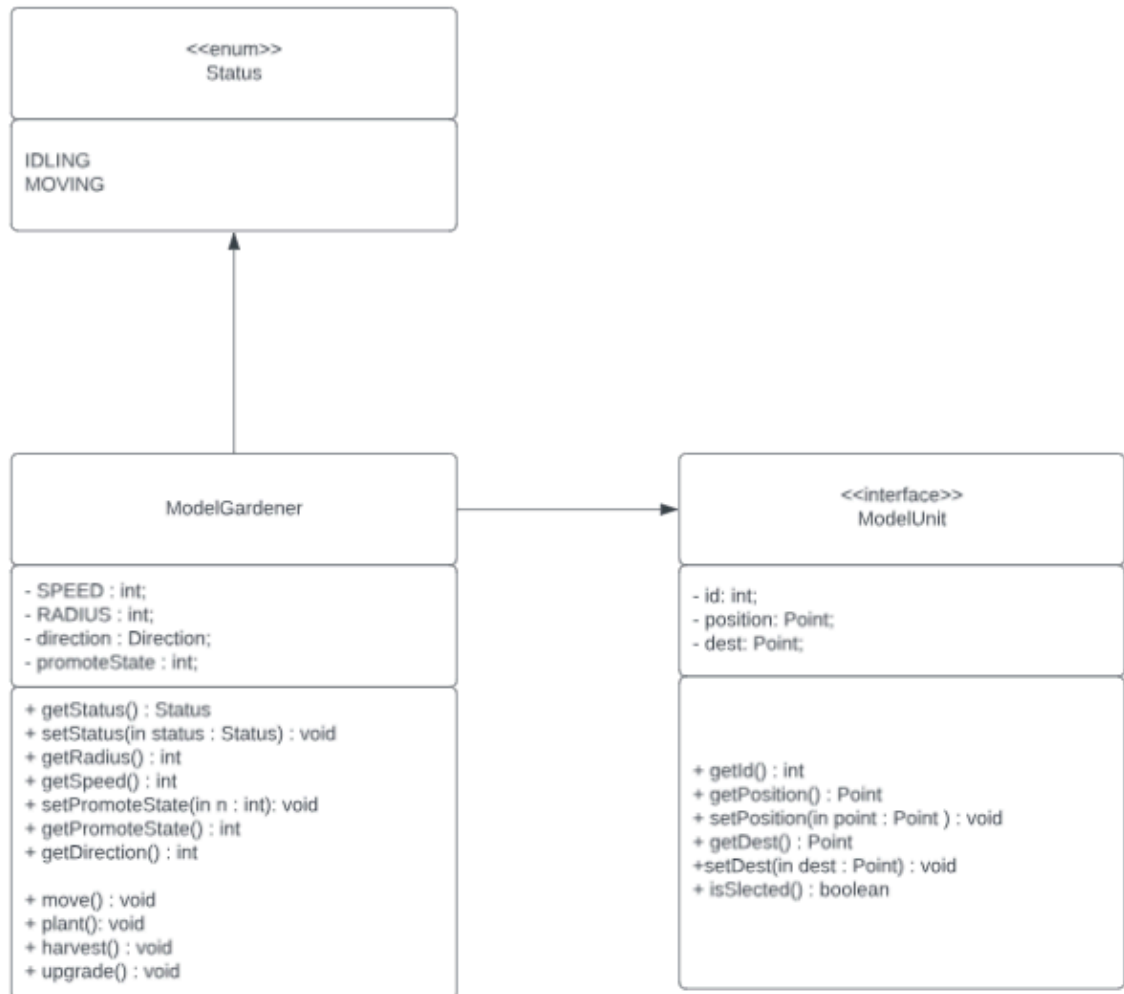
Entités jouant un rôle central dans le jeu, avec des fonctions spécifiques pour planter, récolter, et défendre les plantes contre les invasions de vaches. Chaque jardinier possède les attributs suivants :

- Identifiant (ID) : Unique pour chaque jardinier, facilitant leur gestion.
- Position : Coordonnées actuelles sur la carte du jeu.
- Destination : Coordonnées de la destination lorsqu'ils se déplacent.
- Vitesse : Détermine la rapidité de déplacement d'un point à un autre.
- État : Inclut différents états comme se déplacer, planter, récolter, ou être en repos.

Les fonctionnalités principales :

- Action de déplacement: Algorithmes pour naviguer vers des destinations spécifiques.
- + Calculer la distance entre la position et la destination.

- 
- + Si la distance est supérieure à la vitesse (cela lui prend plus de 1 seconde pour y arriver), le déplacer d'une distance égale à sa vitesse vers sa destination
  - + Sinon, le placer directement à sa destination et changer son statut en mode repos.
- 
- Action de plantation: Sélection et plantation de graines dans des zones désignées.
  - + S'il y a assez d'argent, planter un type de plante aléatoire à sa position actuelle.
  - + Sinon, rien est fait
- 
- Action de récolte: Collecter des plantes mûres pour générer des ressources.
  - + Utiliser la position actuelle pour vérifier toutes les plantes autour, si la plante est récoltable, retirer cette plante du jeu et ajouter l'argent.
  - + Sinon, rien est fait
- 
- Action de promouvoir: Amélioration des compétences et des capacités via l'utilisation de ressources.
  - + S'il y a assez d'argent, augmenter la vitesse et le rayon de récolte du jardinier.
  - + Il existe une limitation de 5 niveaux maximum de promotion pour éviter d'avoir un jardinier qui peut protéger tous les plants en même temps.





---

### 3. Plant



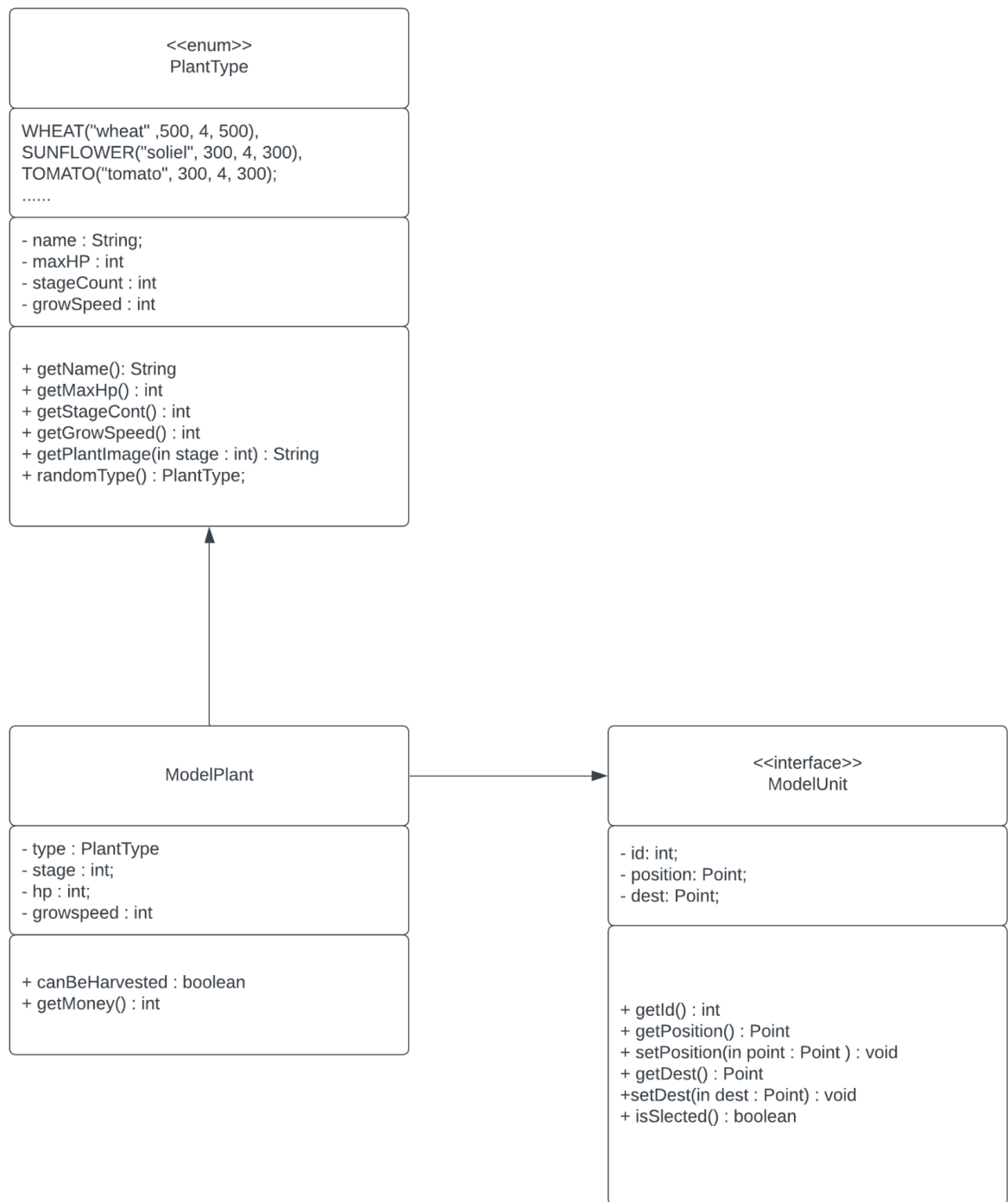
Utilisez une énumération PlantType pour représenter les différents types de plantes, chacun avec son propre nom, son art, sa santé maximale (sa résistance face à être mangé par les ennemis), sa vitesse de croissance et le nombre d'étapes qu'elles doivent franchir avant d'être récoltables. Le taux de croissance du plant et le temps nécessaire pour qu'il mûrisse peuvent ajouter un élément de gestion des ressources au jeu.

Les plantes représentent les ressources primaires du jeu, avec différents types offrant divers bénéfices.

Chaque plant possède les attributs suivants :

- Type de Plante (PlantType) : Différencier les plantes par leurs attributs et leur valeur.
- Position : Emplacement dans le jardin.
- Santé : Indicateur de la durabilité de la plante, influencée par les attaques de vaches.
- Croissance : Étapes de développement avant que la plante ne soit prête à être récoltée.

Les fonctionnalités principales



---

#### 4. Rabbit



utilise la texture d'une vache pour représenter une vache affamée venant de chez votre voisin pour se régaler de vos plantes.

L'unité vache n'est pas contrôlable par l'utilisateur, elles sont toutes autonomes. Les vaches sont des éléments perturbateurs qui visent à consommer les plantes du joueur.

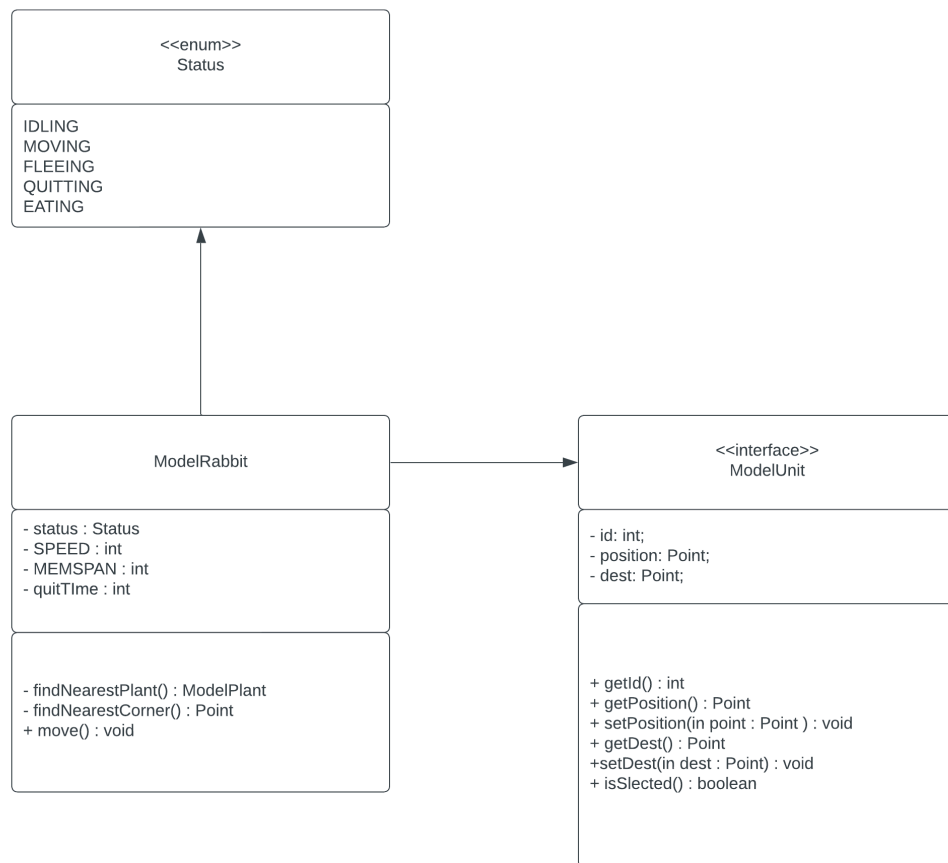
Chaque vache a 5 principaux états :

- En Attente : la vache reste immobile
- En Mouvement
- En Fuite : lorsqu'une vache se trouve dans le rayon d'un jardinier, elle s'enfuira vers le coin le plus proche du champ
- En Quitte : après un certain temps sans manger, la vache perd toute sa patience et quitte le champ
- En Train de Manger : mange quand elle est près d'une plante

---

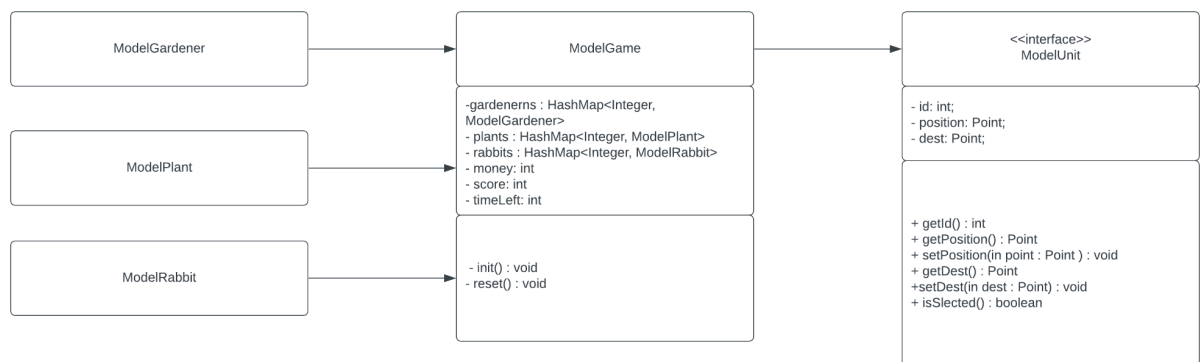
Les fonctionnalités principales :

- Action de déplacement:
  - + Une vache apparaît aléatoirement depuis le bord du champ
  - + si la patience de la vache est à 0, elle quitte le champ par le coin le plus proche
  - + Si elle est à portée d'un jardinier et qu'elle n'est pas déjà en fuite ou en train de quitter, elle commencera à fuir vers l'un des 4 coins du champ selon lequel est le plus proche
  - + Si elle n'est affectée par aucune de ces conditions, elle essaiera d'atteindre la plante la plus proche qui n'est pas gardée par un jardinier.
  - + si aucune plante n'est accessible, elle se déplacera aléatoirement
  - + pour rendre la vache un peu plus réaliste, j'ajoute une durée de mémoire à chacune d'elles, donc elles seront un peu plus imprévisibles et dynamiques



## 5. ModelGame

- Utilisez 3 HashMap avec id, unité pour conserver les informations sur les jardiniers, les plantes, les lapins. Contient également des informations sur l'argent, le score et le temps restant dont dispose le joueur.
- Système de score : est mis à jour à chaque fois qu'un jardinier récolte une plante. Le score est calculé en fonction de la vitesse de croissance de cette plante (plus elle met de temps à pousser == plus elle a de chances d'être mangée par une vache, donc donne plus de points).
- Système monétaire : à chaque fois qu'une plante réalise une récolte, ajoutez la même somme d'argent que le score de cette plante. L'argent est utilisé pour acheter des jardiniers, planter de nouvelles graines ou améliorer les jardiniers.
- Fin du jeu : Le jeu se termine automatiquement après une session de 5 minutes ou les joueurs n'ont plus les moyens d'augmenter leur argent (pas assez d'argent pour acheter de nouvelles graines), le seul facteur décisif étant le score obtenu lors de la récolte des plantes.

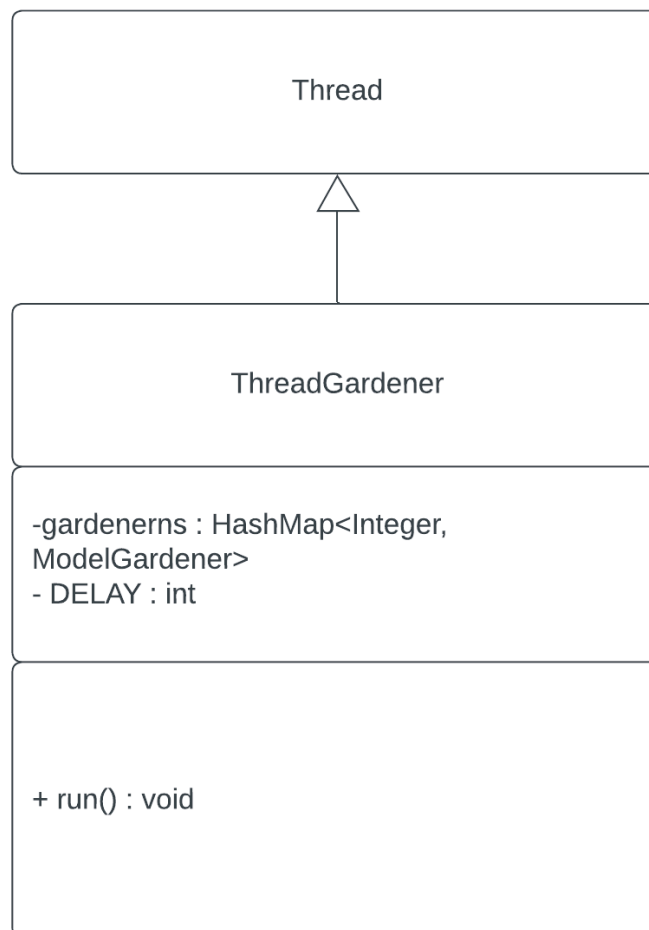


---

## B) Contrôleur

### 1. ThreadGardener

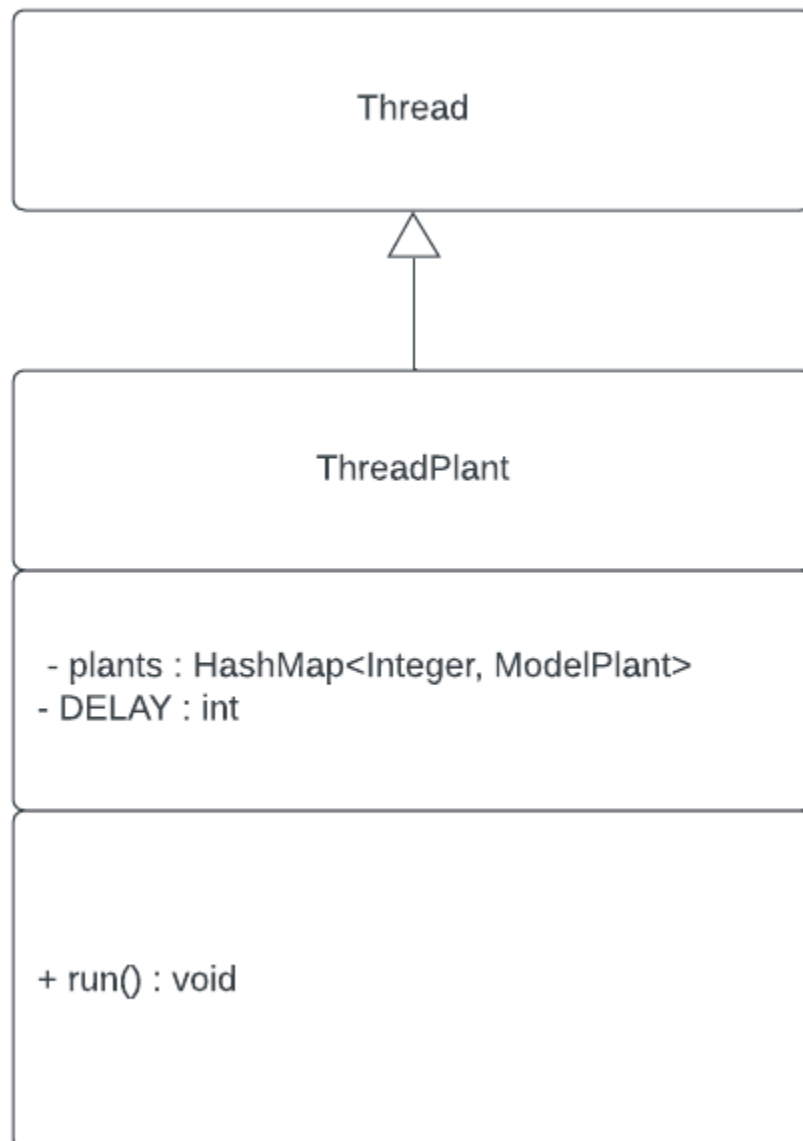
- Extension du Thread
- Après chaque boucle d'un DELAY constant, déplacez tous les gardeners



---

## 2. ThreadPlant

- Extension du Thread
- Après un délai constant, faites pousser toutes les plantes. Si une plante est trop vieille, ou si sa santé est réduite à 0, retirez-la du jeu.

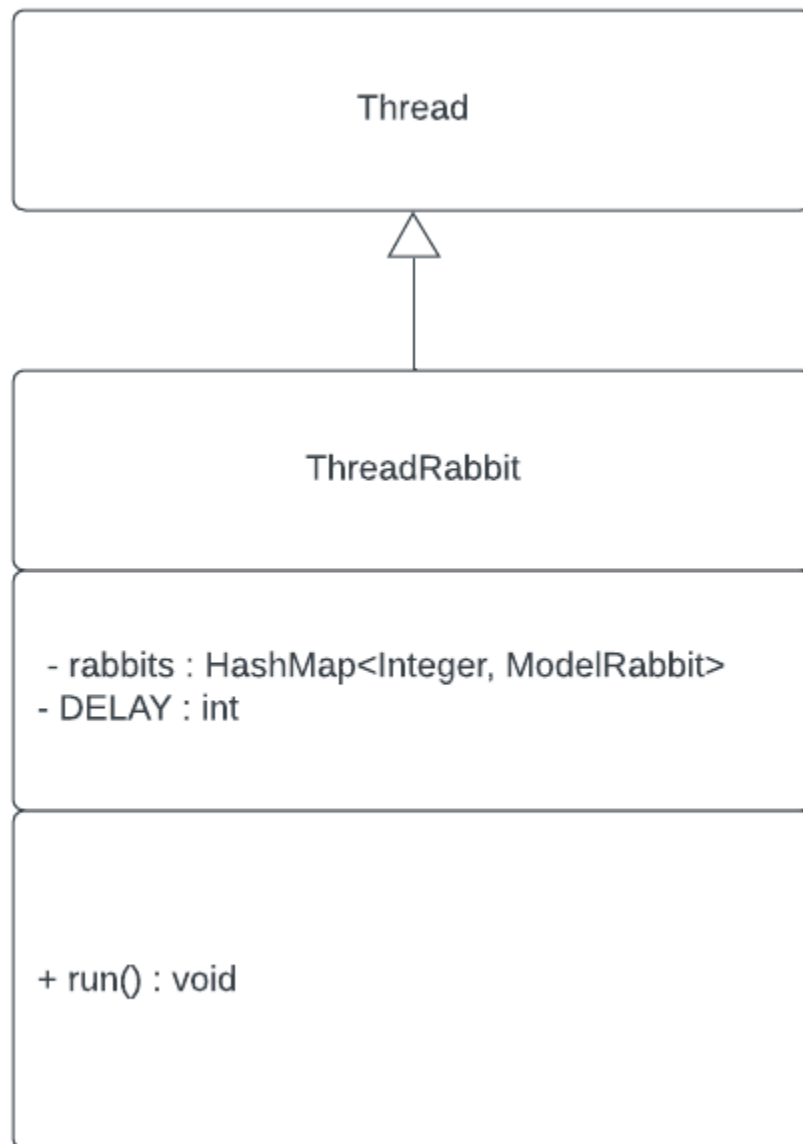




---

### 3. ThreadRabbit

- Extension du Thread
- Après chaque boucle d'un DELAY constant, déplacez tous les vaches



## C) Vue



---

La vue principale du jeu se compose de 2 sections principales : la section de gauche et la section de droite.

- La section de gauche (surlignée en bleu clair) montre l'état actuel du jeu, avec les jardiniers, les vaches, les plantes, votre score et votre argent disponible.
- La section de droite (surlignée en vert clair) affiche les informations d'un objet sélectionné (jardinier, vache ou plante.) Si rien n'est sélectionné, elle affiche la boutique où le joueur peut acheter des objets.

Les fonctionnalités principales :

- La section de gauche écoute les entrées de l'utilisateur pour sélectionner et désélectionner des objets. Si rien n'est sélectionné, cette section obtient la position du clic gauche de la souris, si cette position est en corrélation avec la hitbox d'un objet, sélectionnez cet objet. Si quelque chose est déjà sélectionné, un clic gauche le désélectionne si vous cliquez sur un espace vide ou sélectionnez un nouvel objet sur lequel vous cliquez.
- La section de droite utilise l'objet sélectionné dans la section de gauche pour afficher des informations ou afficher la boutique si rien n'est sélectionné.

## 6. Résultat





## 7. Documentation utilisateur

- Prérequis : Java avec un IDE  
(ou Java tout seul si vous avez fait un export en .jar exécutable)
- Mode d'emploi (cas IDE) : Importez le projet dans votre IDE, sélectionnez la classe Main à la racine du projet puis cliquez sur l'icône « Run ».
- Mode d'emploi (cas .jar exécutable) : double-cliquez sur l'icône du fichier .jar
- Cliquez sur la fenêtre pour faire monter l'ovale

---

## 8. Documentation développeur

- La classes principale : Main.java
- Les packages :
  - + assets : les images
  - + control : les classes des contrôleur
  - + main : contient Main.java
  - + model : les classes des modèle
  - + view : les classes des vue

## 9. Conclusion et perspectives

- J'ai mis en œuvre avec succès toutes les exigences de base du proje
- Je prévois d'ajouter des effets sonores et des effets spéciaux.
- Je dois également améliorer l'optimisation de mon code pour améliorer la performance.