



## Rapport de projet LIFPOO - Printemps 2023

Réalisé par
FAUSTMANN Lucas
SIONI Farès

# Fonctionnalités :

### **VueManager** : ~ 2-3 jours

Le VueManager est un élément clé de notre projet, car il nous permet de gérer efficacement nos différentes vues. Pour faciliter cette gestion, nous avons divisé nos vues en écrans et en composants. Chacune de ces vues nécessite des données spécifiques, met à jour les données de manière différente et a des interactions différentes avec l'utilisateur.

En tant que Singleton, le VueManager est responsable de plusieurs tâches importantes, notamment la gestion de la vue actuelle, la transition entre les différentes vues, ainsi que la gestion du curseur. Grâce à cette approche modulaire, nous pouvons facilement ajouter de nouvelles vues et modifier les comportements existants sans affecter le reste de l'application.

### **IconRepository** : ~ 1/2 jour

Nous avons créé le IconRepository pour optimiser l'affichage des icônes dans notre application. Plutôt que de recharger la SpriteSheet à chaque fois qu'une icône doit être affichée, nous avons implémenté un gestionnaire pour stocker les icônes en mémoire.

Le IconRepository utilise également le design pattern Singleton, ce qui nous permet de garantir qu'une seule instance de la classe est créée pendant toute la durée de l'exécution de l'application. Les icônes sont stockées dans des variables statiques et peuvent être aisément accessibles grâce à un Enum qui répertorie toutes les icônes disponibles.

Grâce à cette approche, nous pouvons accéder rapidement aux icônes dont nous avons besoin sans subir les coûts de performance liés au rechargement constant de la SpriteSheet.

### **Ordonnanceur** : ~ 2-3 jours

Nous avons apporté des modifications à l'ordonnanceur pour améliorer les performances de notre application. Nous avons séparé la mise à jour de l'interface utilisateur (UI) de celle du modèle pour éviter tout effet de blocage potentiel.

Pour la mise à jour de l'UI, nous avons programmé un Thread pour attendre environ 16ms (soit  $1000/60$  pour atteindre une fréquence de 60 fps) avant de procéder à la mise à jour de l'affichage. Cela permet de maintenir une fréquence de rafraîchissement fluide et d'éviter les problèmes de délai d'affichage.

Pour la mise à jour du modèle, nous avons mis en place un Timer qui appelle régulièrement la fonction correspondante toutes les X ms. Cette valeur peut être réglée grâce à un Slider disponible sur la Vue. Cette approche permet d'assurer une mise à jour régulière du modèle tout en garantissant la fluidité de l'UI.

### **Système économique** : ~ 1-2 jours

Notre système économique permet aux joueurs de gagner des pièces en cultivant et en vendant des légumes dans leurs potagers. Les pièces peuvent ensuite être utilisées pour acheter de nouveaux légumes à planter ou pour acheter de nouveaux potagers. Concernant le prix de vente des légumes, celui-ci correspond au double du prix d'achat que l'on multiplie par la qualité finale du légume.

Le système de pièces donne ainsi une dimension économique au jeu, permettant aux joueurs de prendre des décisions stratégiques sur la façon de dépenser leurs pièces.

### **SimulateurMeteo : ~ 3-4 jours**

Le SimulateurMeteo est un élément clé de notre application, car il permet de créer des conditions météorologiques dynamiques et réalistes qui ont un impact direct sur la croissance et la santé des légumes cultivés dans notre jeu. Le simulateur prend en compte différents paramètres, tels que la saison, la température, l'humidité et l'ensoleillement, pour créer des conditions météorologiques variées et réalistes.

Nous avons conçu un système de météo basé sur les saisons, chaque saison ayant ses propres caractéristiques météorologiques spécifiques. Les conditions météorologiques affectent les cases cultivables, qui ont une influence sur la qualité des légumes (croissance et état de vie). Par exemple, des températures trop élevées peuvent endommager certaines variétés de légumes, tandis que des températures trop basses peuvent causer des dommages dus au gel. De même, l'évolution des taux d'humidité et d'ensoleillement a un impact sur le devenir des légumes.

Notre simulateur météorologique permet donc de créer un environnement dynamique et réaliste pour les légumes cultivés dans notre jeu, en prenant en compte les conditions météorologiques réelles qui influent sur la croissance des plantes. Cela permet aux joueurs de comprendre les conditions idéales pour chaque type de légume et d'adapter leur stratégie de culture en conséquence.

### **Panneau d'information de la météo : ~ 1/2 jour**

Le panneau d'information de la météo affiche la saison en cours ainsi que la température, l'humidité et l'ensoleillement actuels. Ces informations sont mises à jour régulièrement en fonction des changements météorologiques dans le simulateur de potager. Le panneau est conçu pour aider les joueurs à comprendre les conditions actuelles et à planifier leurs activités de jardinage en conséquence.

### **Système de sauvegarde : ~ 1-2 jours**

Notre système de sauvegarde a été conçu pour permettre aux utilisateurs de notre application de sauvegarder et charger facilement leur progression. Nous avons mis en place une classe SaveAndLoad qui permet de gérer les sauvegardes et chargements.

Lorsque l'utilisateur souhaite charger une sauvegarde, notre système vérifie tout d'abord si un fichier de sauvegarde existe à l'emplacement spécifié. Si tel est le cas, il tente de charger les données à partir du fichier. Si le fichier est valide, les données sont chargées et le SimulateurPotager est initialisé avec les données sauvegardées, sinon une nouvelle sauvegarde est créée.

Si aucun fichier de sauvegarde n'est trouvé, un nouveau SimulateurPotager est créé avec des valeurs par défaut, et l'utilisateur peut commencer une nouvelle partie.

Lorsque l'utilisateur souhaite sauvegarder son SimulateurPotager, notre système enregistre les données dans un fichier spécifié par l'utilisateur. Les données sont stockées sous forme de SaveData, qui contient les données de l'inventaire de l'utilisateur et du SimulateurPotager.

Notre système de sauvegarde permet à l'utilisateur de reprendre facilement sa progression, même après avoir quitté l'application ou redémarré son ordinateur. De plus, notre système assure que les données sauvegardées soient toujours en format valide, afin de garantir une expérience utilisateur cohérente.

### **Affichage des légumes et cases selon leur état : ~ 1-2 jours**

Nous affichons les légumes dans le potager en fonction de leur état. Les légumes sont affichés avec une échelle et une teinte correspondant à leur croissance et à leur qualité. Cela permet aux joueurs de savoir rapidement quelles plantes sont prêtes à être récoltées et de choisir les légumes les plus mûrs pour une meilleure qualité de récolte. Le même système est en place pour l'affichage des cases cultivables, afin d'indiquer leur taux d'humidité.

### **Tooltip d'information sur la case : ~ 1/2 jour**

Lorsqu'un joueur survole une case avec le curseur de la souris, le tooltip d'information s'affiche automatiquement.

Le tooltip contient des informations sur la case, son humidité, son ensoleillement et les informations sur le légume, le cas échéant (variété et qualité).

### **Plusieurs actions sur les potagers :** ~ 2-3 jours

Le système de gestion de potager implique plusieurs fonctionnalités qui permettent aux utilisateurs de prendre soin de leur jardin. Tout d'abord, l'utilisateur peut planter différents types de légumes dans les cases de son potager. Il peut également les arroser pour les maintenir en bonne santé et surveiller la croissance des légumes. Une fois qu'ils ont atteint leur maturité, l'utilisateur peut les récolter pour obtenir des pièces. Le curseur de la souris change en fonction de l'action sélectionnée, permettant une meilleure immersion à l'utilisateur.

### **Édition des informations d'un potager :** ~ 1-2 jours

Nous permettons aussi aux utilisateurs de gérer leurs potagers en leur offrant la possibilité de personnaliser leurs informations. En effet, ils peuvent changer le nom du potager, ainsi que la couleur du bouton qui lui est associé pour une meilleure identification visuelle. De plus, ils peuvent par ailleurs supprimer un potager qui ne leur est plus utile ou qu'ils ne souhaitent plus entretenir.

### **Différentes familles de légumes :** ~ 1-2 jours

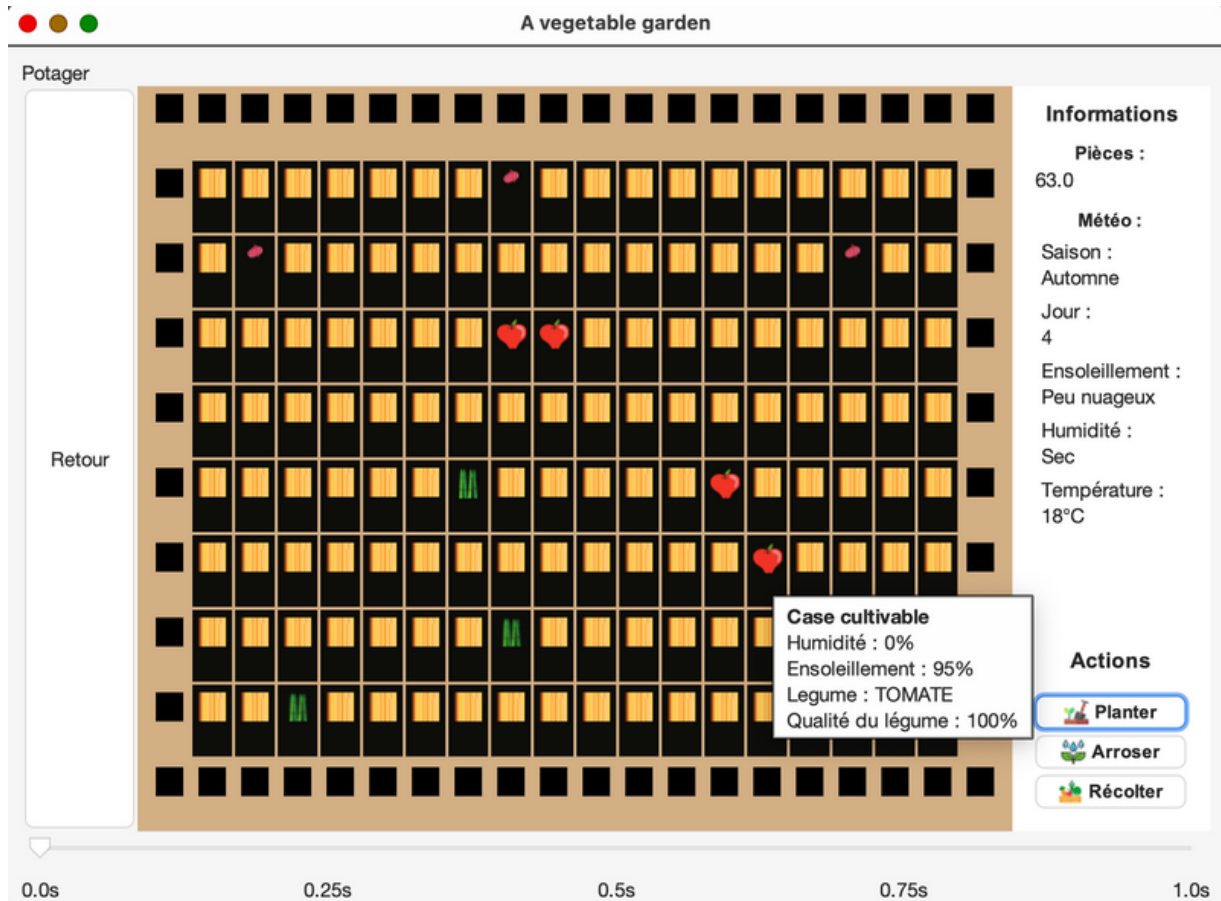
Nous avons créé un système de familles de légumes, chacune ayant ses propriétés spécifiques en fonction des conditions environnementales de la case. Par exemple, les légumes de la famille des "racines" sont plus sensibles à l'humidité. Les légumes de la famille des "feuilles" ont besoin de beaucoup d'ensoleillement. Les joueurs doivent donc tenir compte de ces différences et choisir les légumes les plus adaptés aux conditions de chaque case pour maximiser leur production, rendant la simulation encore plus immersive.

### **Tests unitaires :** tout le long du projet

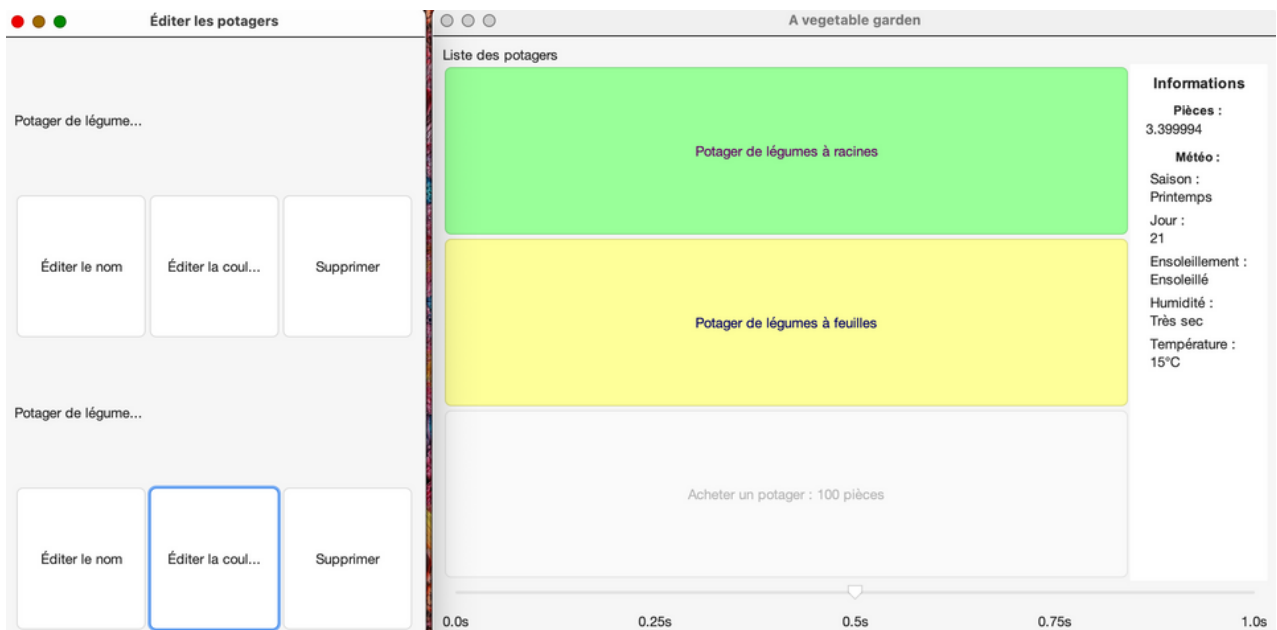
Les tests unitaires sont une pratique essentielle en développement logiciel pour valider le comportement de chaque unité de code de manière isolée et de manière répétable. Dans notre projet, nous utilisons JUnit pour les tests unitaires. Ces tests nous permettent de valider que les valeurs utilisées dans notre code sont correctes et de détecter rapidement tout comportement inattendu.

## Diagramme UML :

# ANNEXES :



*Vue d'un potager avec les cases cultivables (et les légumes plantés) au centre, les informations (météo et pièces) ainsi que les diverses actions à droite, le slider de temps en bas et enfin le tooltip suite au survol de la case avec la tomate.*



*Vue d'ensemble des différents potagers dans la fenêtre de droite, avec au centre les quelques potagers de l'utilisateur, et dans la fenêtre de gauche le menu d'édition des potagers*