
Rapport : ENSEMENC



Auteurs : Bruyère Sarah, Sacher Juliette
Module : Programmation avancée
Ecole Nationale Supérieure de Cognitique - INP Bordeaux
Promotion 2027 - Groupe 1

SOMMAIRE

I. Introduction.....	2
II. Fonctionnement du jeu.....	3
A. Présentation du jeu.....	3
B. Déroulement d'une partie du jeu.....	3
III. Architecture objet du simulateur de potager.....	5
A. Structure globale.....	5
B. Présentation des classes.....	6
Classe MenuDebut.....	6
Classe Plante.....	6
Classe Meteo.....	7
Classe Partie.....	7
Classe PlanteProductionMultiple.....	7
Classe PlanteProductionSimple.....	7
Classe InventaireTypePlante.....	7
Classe Joueur.....	7
Classe Magasin.....	7
Classe Outil.....	8
Classe Semis.....	8
Classe Recolte.....	8
Classe Parcelle.....	8
Classe Terrain.....	8
Classe TerrainDesertique.....	8
Classe TerrainTropical.....	8
Classe TerrainVolcanique.....	8
Classe Urgence.....	9
Classe Guide.....	9
Classe Simulation.....	9
Classe Program.....	9
C. Diagramme UML.....	9
IV. Tests fonctionnels.....	12
V. Gestion du projet.....	13
A. L'organisation de l'équipe et du projet.....	13
B. Planning prévisionnel.....	14
C. Planning réel.....	15
VI. Retours d'expérience et pistes d'amélioration.....	16
A. Retours d'expérience.....	16
B. Pistes d'amélioration.....	16
VII. Conclusion.....	16

I. Introduction



Ce projet s'inscrit dans le cadre du module de programmation avancée du S6 de l'ENSC. Il a pour objectif de concevoir un simulateur de potager fonctionnant en mode console. L'utilisateur y incarne un jardinier qui débute avec un certain nombre de semis et de terrains, chacun possédant des caractéristiques propres. La simulation se déroule dans un pays choisi par les développeurs. Le pays influence directement les types de cultures disponibles et les conditions climatiques du jeu.

Le système intègre également une dimension dynamique à travers une surveillance constante du jardin. Une caméra fictive permet de signaler au joueur les événements imprévus, comme les intempéries violentes ou l'apparition d'intrus. L'ensemble vise à proposer une expérience de gestion immersive et évolutive, tout en mettant en pratique les principes fondamentaux de la programmation orientée objet.

Les contraintes techniques sont les suivantes. Le simulateur de potager a été développée en C#, sous forme de programme console, en utilisant la version 8 du framework .NET, comme spécifié. Le code s'appuie pleinement sur les principes de la programmation orientée objet, avec une structuration claire reposant sur l'encapsulation, l'héritage, le polymorphisme, les classes abstraites, ainsi que les associations entre objets. Les collections d'objets sont gérées principalement à l'aide de listes, préférées aux tableaux pour leur souplesse d'utilisation. Une attention particulière a été portée à la lisibilité et à la maintenabilité du code, en veillant à limiter les redondances et à structurer les fonctionnalités de manière cohérente. Enfin, l'ensemble du code respecte les conventions de nommage propres au langage C#, tout en étant correctement indenté et commenté lorsque cela est nécessaire.

II. Fonctionnement du jeu

A. Présentation du jeu

Prêt à faire pousser votre potager sous le soleil brûlant du Mexique ? Enfilez votre chapeau de jardinier et plongez dans une aventure agricole où chaque décision compte. Bienvenue dans ENSemenC, un simulateur de culture où la terre, le climat et votre stratégie font toute la différence !

Avant de planter quoi que ce soit, vous héritez d'un terrain : volcaniques aux sols riches, tropicaux à l'humidité constante, ou désertiques aux conditions extrêmes.

À votre disposition : une sélection de semis typiquement mexicains. Faites pousser des tomates juteuses, des piments corsés, ou encore des cactus nopales robustes. Essayez-vous à la culture de l'agave, récoltez des pastèques, des papayes ou encore des fleurs de Tithonia aux couleurs éclatantes. Haricots, avocats et ignames viendront compléter votre potager, si vous savez les cultiver au bon endroit, au bon moment.

Mais attention, un potager ne se gère pas tout seul. Météo capricieuse, intrus indésirables, manque d'eau ou excès de soleil : le jeu vous mettra à l'épreuve.

Envie d'étendre votre potager ? Passez par le magasin pour acheter de nouvelles graines, des outils, ou même des parcelles supplémentaires. Vendez vos récoltes, investissez dans de meilleurs équipements, et faites prospérer votre exploitation au fil des saisons. À vous de jouer !

B. Déroulement d'une partie du jeu

Le joueur commence l'aventure en choisissant ou non de démarrer une nouvelle partie, où il lui est demandé d'entrer un nom. Il reçoit alors les règles du jeu, un capital de départ de 200 pièces, un terrain attribué aléatoirement (par exemple un terrain Tropical) possédant 3 parcelles, et accède à diverses options pour faire prospérer son potager.

La première semaine commence alors, les conditions météorologiques et ses propriétés s'affiche sur la console.

Le joueur interagit avec le jeu via un menu principal où plusieurs actions sont disponibles chaque semaine :

1. Arroser un terrain ou une parcelle
2. Planter un semis
3. Acheter au magasin
4. Vendre les récoltes
5. Récolter une parcelle
6. Utiliser un outil
7. Retirer une plante morte
8. Afficher l'état des terrains
9. Consulter les caractéristiques des plantes
10. Relire les règles
11. Passer à la semaine suivante
12. Quitter le jeu

Le joueur peut ainsi librement gérer son potager à chaque tour en combinant stratégie et adaptation à l'environnement (météo, urgences...).

Afin d'arroser ses plantes, le joueur doit choisir s'il veut arroser un terrain entier ou juste une parcelle. L'effet de l'eau s'applique ensuite sur le terrain ou parcelle choisi.

Pour planter un semi, le joueur voit son stock de semis et peut choisir lequel il veut planter, il lui est ensuite demandé de choisir le terrain et la parcelle sur lesquels il veut planter son semi.

Lorsqu'il se rend au magasin, le joueur peut acheter des semis (par exemple : des papaye, Pastèque, Tomate mais attention, les plantes ont toutes un terrain favori qui leur permet de pousser), des outils (pour faire face aux urgences et protéger ses terrains), des terrains (volcanique, désertique ou tropical), des parcelles ou quitter le magasin. Chaque item a un coût spécifique et des caractéristiques propres. Une fois un achat effectué, les stocks mis à jour s'affiche pour permettre au joueur de suivre l'évolution de ses stocks.

Ensuite, le joueur a la possibilité de vendre ces récoltes afin de gagner de l'argent pour pouvoir acheter d'autres produits dans le but de faire croître son potager et sa richesse personnelle.

En tapant 6, le joueur peut utiliser un des outils qu'il possède dans son stock afin de protéger ses parcelles des intempéries et des intrus. La protection ne dure que 2 semaines.. Le joueur peut aussi décider de retirer une plante morte afin de pouvoir en planter une nouvelle à la place.

Pour voir l'état de ses terrains et de ses plantes, le joueur peut entrer le 8. Pour la plante, sa santé, sa croissance vont s'afficher puis, pour l'état des terrains, se sont l'humidité, les précipitations et l'ensoleillement qui vont être visibles.

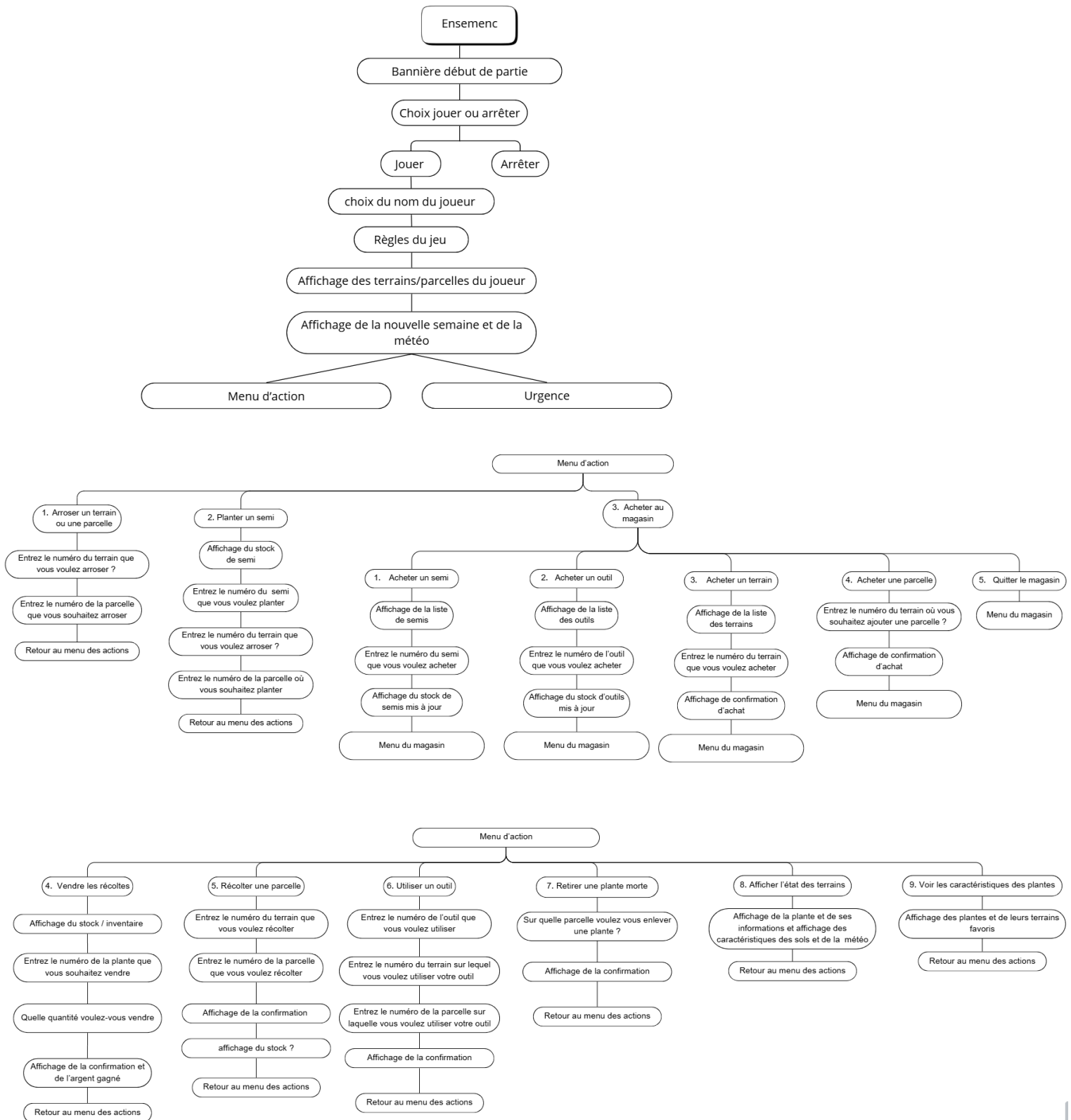
Le joueur peut aussi afficher les plantes et leurs caractéristiques ainsi que les règles du jeu. Finalement, une fois que le joueur a réalisé toutes les actions voulues, il a le choix d'arrêter complètement la simulation ou de passer à la semaine suivante pour voir la croissance de ses plantes et les nouvelles conditions météorologiques.

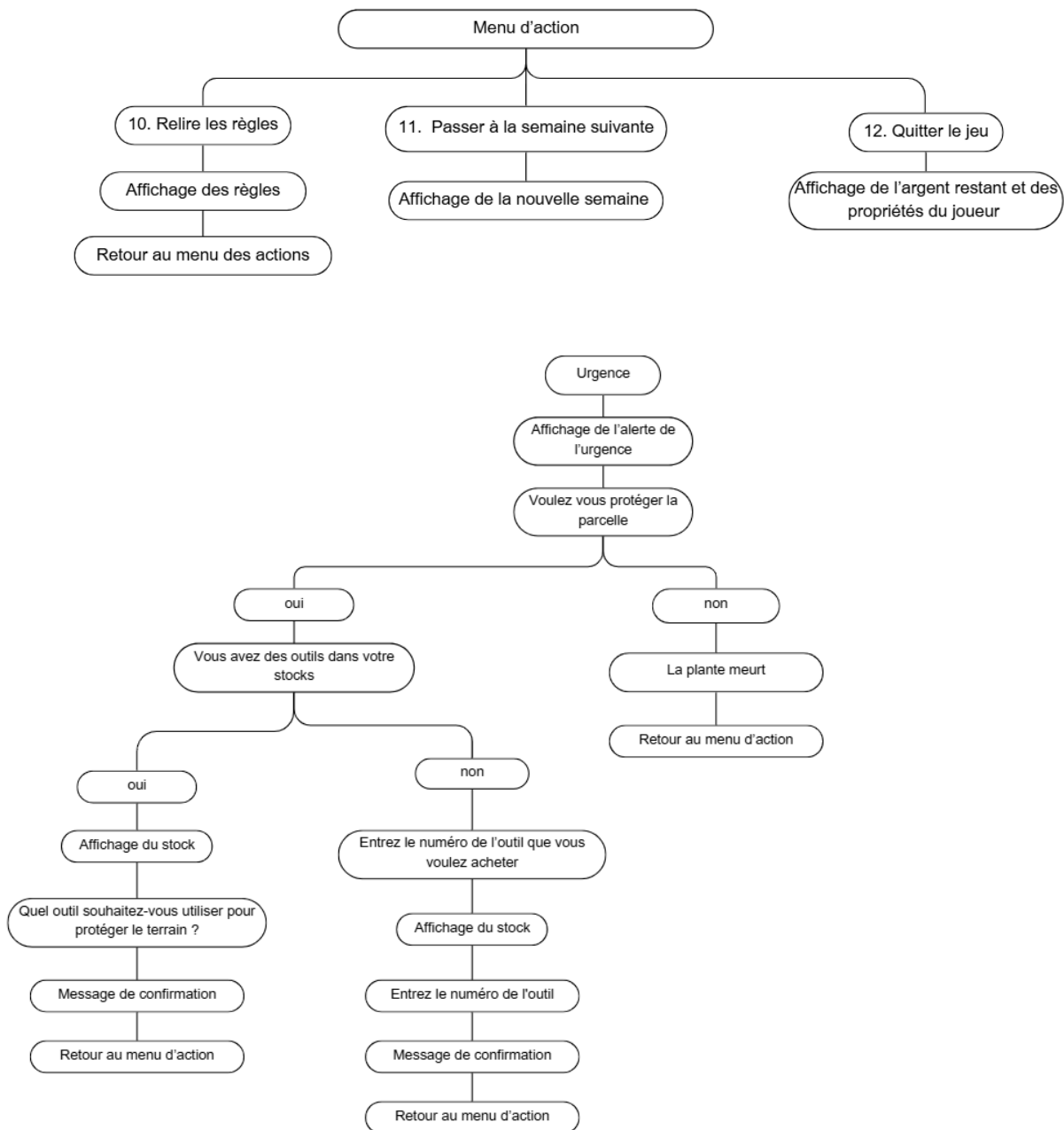
Seulement attention, des urgences peuvent survenir, que ce soit des intempéries ou des intrus, il faut que le joueur protège ses terrains avant que toutes ses plantes ne meurent. Une fois une urgence déclenchée, le joueur a le choix d'utiliser un outil présent dans son stock ou d'en acheter un puis de choisir la parcelle à protéger. Mais attention, la protection ne dure que 2 semaines. Si le joueur décide de ne rien faire, la plante concernée par l'urgence perdra de la vie à chaque tour !

Lorsque le joueur décide de quitter la simulation, son argent va s'afficher ainsi que ses récoltes.

III. Architecture objet du simulateur de potager

A. Structure globale





B. Présentation des classes

Classe MenuDebut

La classe MenuDebut gère l'affichage et la logique du menu principal du jeu. Elle propose au joueur de commencer une nouvelle partie ou de quitter l'application. En cas de lancement, elle initialise les éléments nécessaires à la simulation (joueur, météo, simulation) et retourne une instance de la classe Partie. Elle contient également une méthode dédiée à l'affichage d'une bannière d'accueil stylisée en console, renforçant l'immersion dès le démarrage du jeu.

Classe Plante

La classe abstraite Plante représente l'ensemble des caractéristiques communes aux différentes cultures du potager. Elle centralise les propriétés essentielles d'une plante (nom, saison de semis, type de terrain préféré, besoins en eau et en lumière, température idéale,

etc.) et gère son état de santé ainsi que son évolution au fil du temps. Elle intègre également une méthode `AnalyserSante`, qui ajuste la santé de la plante en fonction des conditions météorologiques, influençant directement sa croissance et sa survie. Enfin, la méthode abstraite `AvoirQuantiteRecolte` permet aux sous-classes de définir leur propre logique de rendement.

Classe Meteo

La classe `Meteo` modélise les conditions climatiques hebdomadaires influençant la ferme. Elle génère aléatoirement la température, les précipitations et l'ensoleillement, puis applique leurs effets sur les parcelles, en modifiant notamment leur humidité. Elle permet ainsi de simuler l'impact du climat sur la croissance des plantes dans le jeu.

Classe Partie

La classe `Partie` représente une session de jeu en regroupant un joueur et une simulation. Elle permet d'associer un joueur à son environnement de jeu, assurant ainsi le bon déroulement et le suivi de la partie. C'est une classe centrale pour l'organisation du jeu.

Classe PlanteProductionMultiple

La classe `PlanteProductionMultiple` hérite de `Plante` et représente une plante capable de produire plusieurs unités lors de la récolte. Elle évalue ses conditions de croissance (température, humidité, lumière) pour ajuster sa santé, son âge et le nombre de produits générés.

Classe PlanteProductionSimple

La classe `PlanteProductionSimple` hérite de la classe `Plante` et représente une plante dont la récolte ne produit qu'une seule unité. Elle contient une méthode `ToString` pour afficher son état (santé et croissance) et redéfinit la méthode `AvoirQuantiteRecolte` pour indiquer une production fixe.

Classe InventaireTypePlante

La classe `InventaireTypePlante` permet d'instancier différents types de plantes en fonction de leur nom. Elle centralise la création des objets `PlanteProductionSimple` ou `PlanteProductionMultiple`, selon les caractéristiques prédéfinies de chaque espèce. Cela facilite la gestion et l'ajout de nouvelles plantes dans la simulation.

Classe Joueur

La classe `Joueur` représente l'utilisateur principal du jeu, chargé de gérer une ferme. Elle contient les propriétés essentielles telles que l'argent, les terrains, les semis, les outils et les récoltes. Elle regroupe également toutes les actions possibles dans le jeu, comme planter, arroser, récolter, vendre ou encore nettoyer les parcelles.

Classe Magasin

La classe `Magasin` gère l'ensemble des interactions commerciales entre le joueur et le jeu, notamment l'achat de semis, d'outils, de terrains et de parcelles. Elle contient un menu interactif qui permet au joueur de dépenser son argent pour développer son exploitation agricole. Le magasin centralise aussi l'offre disponible, ce qui facilite l'accès aux ressources nécessaires pour progresser dans la simulation.

Classe Outil

La classe Outils représente un outil disponible dans le jeu, avec son nom, son prix d'achat et sa quantité détenue par le joueur. Elle permet de gérer les outils comme des objets acheteables et stockables dans l'inventaire du joueur. Cette classe est utilisée notamment dans le magasin pour suivre les achats d'outils et leur utilisation éventuelle.

Classe Semis

La classe Semis représente un type de graine que le joueur peut acheter et planter. Elle contient le nom de la plante, son prix d'achat, la quantité possédée et une information indiquant si la plante produit plusieurs récoltes. Elle permet de gérer les stocks de semis dans l'inventaire du joueur ainsi que leurs caractéristiques.

Classe Recolte

La classe Recoltes représente les plantes récoltées par le joueur. Elle contient le type de plante, la quantité récoltée et le prix unitaire de vente. Elle permet de gérer l'inventaire des récoltes du joueur en vue de leur stockage ou de leur vente.

Classe Parcelle

La classe Parcelle représente une unité de culture au sein d'un terrain, identifiée par un numéro et pouvant accueillir une plante. Elle stocke des informations comme l'humidité, l'ensoleillement, si elle est protégée ou vide, ainsi qu'un lien vers son terrain d'origine. Elle permet notamment d'ajouter une plante si la parcelle est libre, et peut aussi être associée à une urgence.

Classe Terrain

La classe abstraite Terrain représente un terrain cultivable dans le jeu, avec des caractéristiques comme le type de sol, l'humidité, la température et l'ensoleillement. Chaque terrain contient une liste de parcelles cultivables et peut être protégé contre certains aléas (oiseaux, grêle, parasites). Elle sert de base aux différentes sous-classes spécifiques de terrain et permet d'ajouter des parcelles ou de mettre à jour les conditions climatiques.

Classe TerrainDesertique

La classe TerrainDesertique hérite de la classe abstraite Terrain et représente un terrain de type désertique, avec des conditions climatiques spécifiques. Elle surcharge la méthode MiseAJourCondition() pour simuler un environnement sec, chaud et très ensoleillé : l'humidité est réduite de moitié, la température augmente de 5°C, et l'ensoleillement est augmenté de 20 %, tout en restant limité à 100 %.

Classe TerrainTropical

La classe TerrainTropical hérite de la classe abstraite Terrain et modélise un environnement tropical. Elle surcharge la méthode MiseAJourCondition() pour refléter un climat humide et chaud : l'humidité augmente de 20 % (sans dépasser 100 %), la température augmente légèrement, et l'ensoleillement est réduit de 10 % pour simuler les effets de la couverture nuageuse fréquente.

Classe TerrainVolcanique

La classe TerrainVolcanique, dérivée de Terrain, représente un environnement influencé par l'activité volcanique. Elle redéfinit la méthode MiseAJourCondition() pour simuler un climat

chaud et sec : l'humidité est réduite de 30 %, la température augmente fortement (+6 °C), et l'ensoleillement diminue de 20 % en raison des cendres volcaniques. Ces ajustements reflètent les conditions difficiles mais fertiles des terrains volcaniques.

Classe Urgence

La classe Urgence modélise des événements imprévus (comme des attaques de nuisibles ou des catastrophes naturelles) pouvant survenir sur une parcelle. Chaque urgence a un type, une gravité (de 1 à 5) et peut impacter la santé des plantes si elle n'est pas résolue rapidement. Elle propose des méthodes pour déclencher aléatoirement une urgence, la résoudre via des outils de protection, et réinitialiser son état une fois traitée.

Classe Guide

La classe Guide fournit des informations utiles au joueur sous forme de documentation intégrée. Elle contient deux méthodes statiques : `CaracteristiquesPlantes`, qui affiche les types de plantes disponibles et leurs terrains préférés, et `ReglesJeu`, qui résume les règles du jeu, les conditions de plantation, les risques liés aux urgences, et les actions possibles pour entretenir le potager. Cette classe sert donc d'aide mémoire interactive pour faciliter la prise en main du jeu.

Classe Simulation

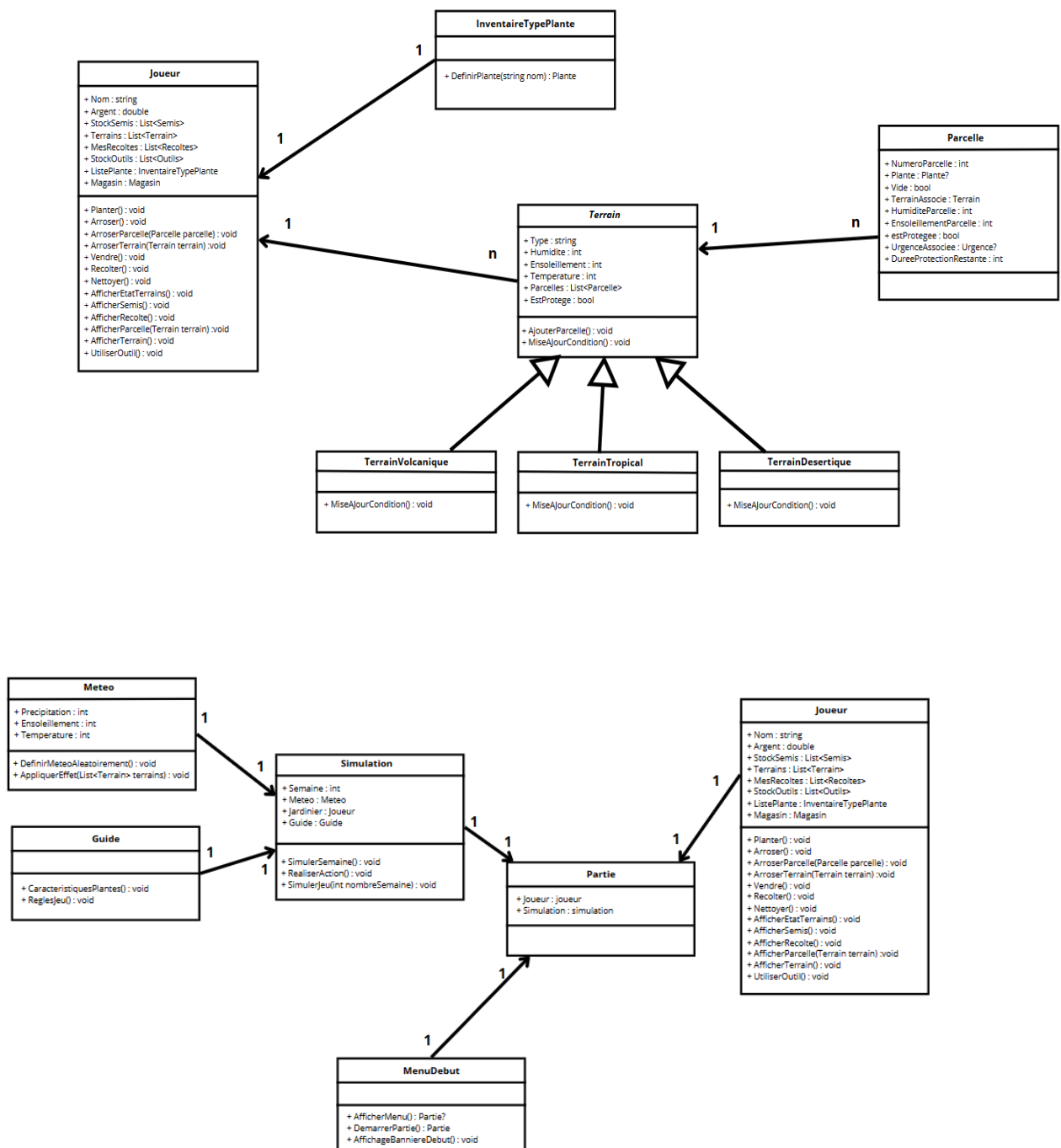
La classe Simulation est le cœur du jeu : elle orchestre le déroulement de la partie semaine après semaine. Elle gère l'évolution des conditions météo, le suivi des plantes sur les parcelles, le déclenchement et la résolution des urgences, ainsi que l'exécution des actions du joueur. Elle permet ainsi de simuler un potager dynamique et interactif, combinant aléas climatiques, entretien des cultures et stratégie de gestion.

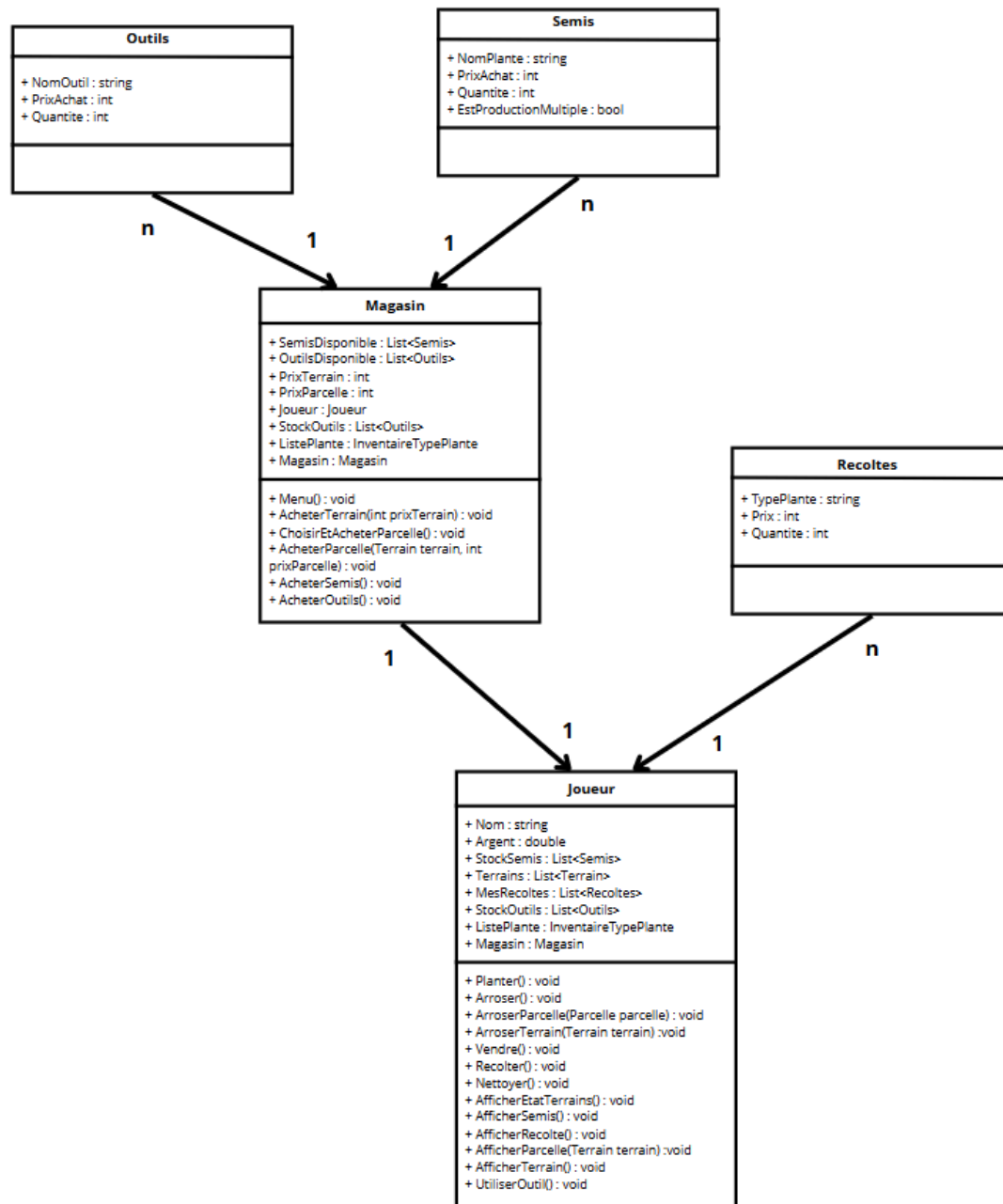
Classe Program

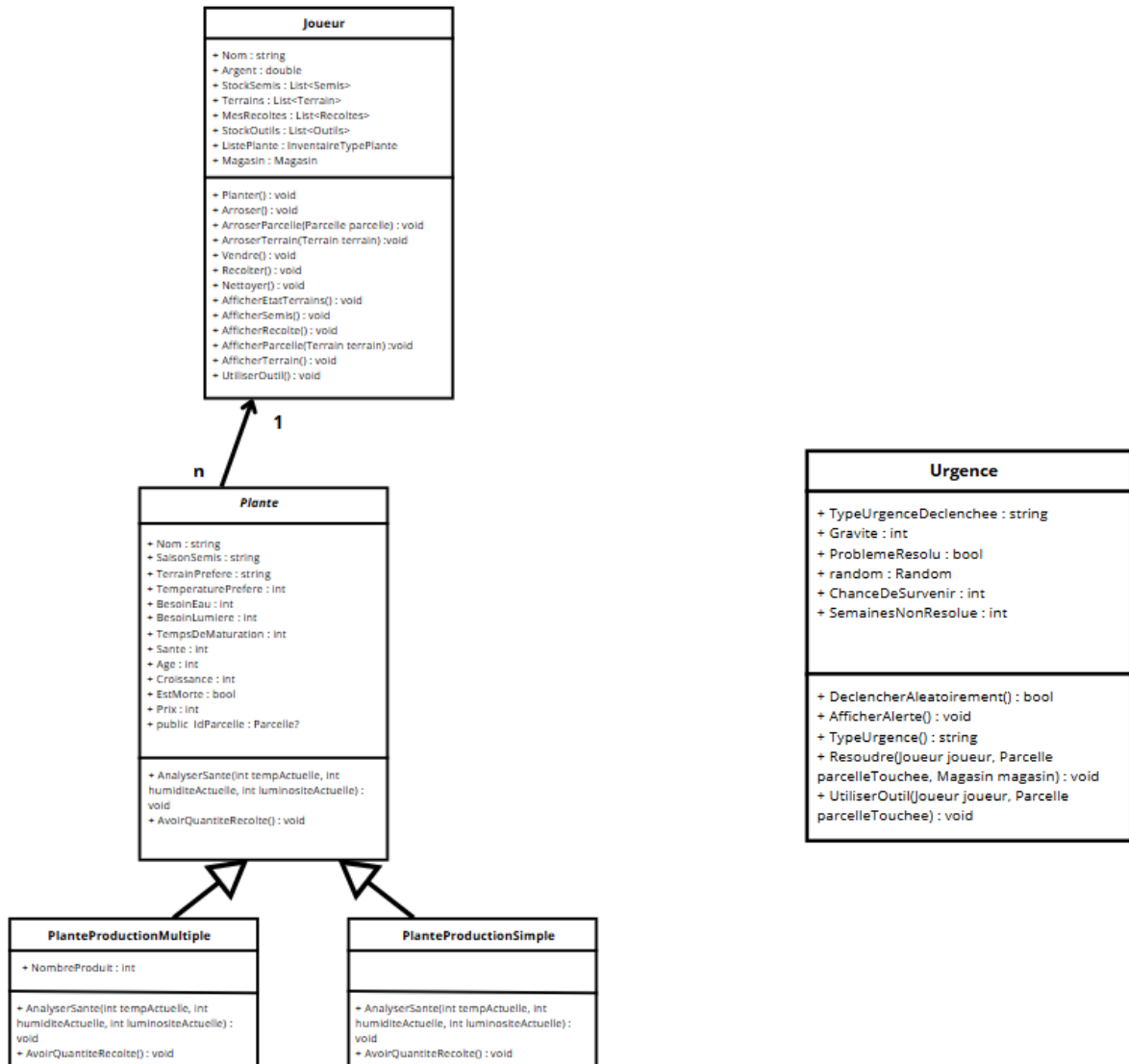
`Program.cs` correspond au lancement principal du jeu. Il initialise la partie via un menu interactif (`MenuDebut`), vérifie que les éléments essentiels (joueur et simulation) sont bien créés, puis affiche les informations initiales du joueur : nom, argent, semis en stock et terrains disponibles. Enfin, il affiche les règles du jeu et démarre la simulation pour 10 semaines, permettant au joueur de gérer son potager semaine après semaine.

C. Diagramme UML

Voici à présent un diagramme de classes UML relatif à notre projet et notre code. Par soucis de clarté, nous le présenterons en différentes parties.







IV. Tests fonctionnels

Pour garantir le bon fonctionnement de notre simulateur de potager, nous avons mené une série de tests fonctionnels couvrant à la fois les aspects unitaires, l'intégration des modules et classes, qui permettent de tester la coopération entre plusieurs modules ou classes (par exemple, s'assurer que lorsqu'un joueur plante un semis, celui-ci est bien ajouté à une parcelle du terrain, que les conditions météo affectent la plante, et que les urgences sont bien déclenchées), et différents scénarios complets de jeu. Donc chaque fonctionnalité proposée au joueur (planter, arroser, récolter, utiliser un outil, acheter, vendre, etc.) a été testée individuellement, puis dans des enchaînements réalistes, afin de vérifier leur cohérence et leur bon enchaînement.

Un effort a été fait sur la robustesse du système face aux erreurs de saisie. À chaque point d'interaction avec le joueur, des vérifications sont effectuées pour éviter toute entrée invalide. Par exemple, si le joueur entre du texte au lieu d'un chiffre, ou sélectionne une option inexistante dans un menu, le programme intercepte l'erreur et redemande une saisie correcte sans planter ni interrompre l'expérience. Cela permet d'éviter les comportements inattendus et de maintenir une navigation fluide tout au long du jeu.

La fin de jeu est également bien gérée : que le joueur décide d'arrêter manuellement ou que toutes les semaines soient écoulées, un résumé de la partie est affiché, avec les récoltes restantes et l'argent final. Ces tests nous ont permis d'assurer une expérience stable, sans interruption, quel que soit le comportement du joueur.

V. Gestion du projet

A. L'organisation de l'équipe et du projet

Pour débiter ce projet, nous avons d'abord défini les modalités dans lesquelles notre simulateur de potager allait se dérouler. Inspirés par l'arrivée des vacances et du soleil, nous avons choisi de situer notre projet dans un pays chaud : le Mexique.

Nous avons commencé par analyser les conditions météorologiques typiques de ce pays, les plantes qu'on peut y cultiver et les risques qui peuvent survenir. À partir de cette réflexion, nous avons identifié les principales classes nécessaires au simulateur, les actions possibles, et donc les fonctions à implémenter.

Tout ce travail de conception a été réalisé ensemble, cela a permis à chacune d'apporter son avis, ses idées et sa créativité. Après avoir identifié les classes principales, nous nous sommes réparties entre nous afin que chacun puisse commencer à structurer le code de son côté.

Tout au long du projet, la communication a été primordiale. Comme plusieurs classes devaient interagir entre elles, nous avons régulièrement fait des mises en commun pour ajuster nos codes respectifs et éviter les incohérences. La communication a été assurée en grande partie en présentiel lors des séances de TP ou lors de rendez-vous fixés préalablement. Autrement, nous avons majoritairement utilisé WhatsApp pour communiquer et Google Drive pour partager nos documents. Le code quant à lui était sur GitHub, ce qui permettait d'avoir les modifications du code de notre partenaire.

Les séances de TP prévues dans l'emploi du temps ont été très utiles. Elles nous ont permis de faire régulièrement le point sur l'avancement du projet, de clarifier certaines parties plus techniques, et de terminer et répartir les prochaines tâches à réaliser. À chaque séance, nous fixions de nouveaux objectifs concrets à atteindre avant la prochaine.

B. Planning prévisionnel

	01/04 au 06/04	07/04 au 13/04	14/04 au 20/04	21/04 au 27/04	28/04 au 04/05	05/05 au 11/05	12/05 au 18/05	19/05 au 23/05
Tâches	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7	Semaine 8
Choix du lieu, des plantes ...								
Identification des classes principales								
Structure des classes identifiées: plantes, terrain, joueur, magasin, parcelle...								
Ajout des fonctions dans chacune des classes								
Mode Urgence (classe Urgence)								
Classe Simulation : affichage...								
Optimisation du code (améliore la robustesse)								
Ajout fonctionnalités bonus								
Rédaction du rapport								

C. Planning réel

	01/04 au 06/04	07/04 au 13/04	14/04 au 20/04	21/04 au 27/04	28/04 au 04/05	05/05 au 11/05	12/05 au 18/05	19/05 au 23/05
Tâches	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7	Semaine 8
Choix du lieu, des plantes ...								
Identification des classes principales								
Structure des classes identifiées: plantes, terrain, joueur, magasin, parcelle...								
Ajout des fonctions dans chacune des classes								
Mode Urgence (classe Urgence)								
Classe Simulation : affichage...								
Optimisation du code (améliore la robustesse)								
Ajout fonctionnalités bonus								
Rédaction du rapport								

Au début du projet, nous avons rencontré des difficultés à bien identifier toutes les classes nécessaires et à nous projeter dans l'architecture du code. Cette première phase a donc pris plus de temps que prévu, ce qui a entraîné un retard dès les premières semaines.

Cependant, une fois cette base clarifiée, le développement s'est accéléré. Par exemple, nous avons pu implémenter des fonctionnalités comme le magasin beaucoup plus tôt que prévu. En revanche, certaines fonctionnalités bonus n'ont pas pu être réalisées, notamment la sauvegarde de la partie et un mode IA. Si nous avons pu démarrer le développement plus rapidement, ces fonctionnalités auraient pu être explorées et potentiellement intégrées au projet. Malgré ce retard en début de projet, nous avons tout de même réussi à fournir une simulation de potager fonctionnel à la date demandée.

VI. Retours d'expérience et pistes d'amélioration

A. Retours d'expérience

Malgré un démarrage un peu difficile, notamment à cause d'un sujet très libre qui nécessitait de bien cadrer les idées pour ne pas s'éparpiller, ce projet nous a permis de monter en compétences. Nous avons progressé en programmation orientée objet, notamment sur la communication entre les classes et la structuration d'un projet complet. Un des éléments les plus positifs de ce projet a été la qualité de la communication dans notre binôme. Travailler de manière collaborative, bien s'organiser et échanger régulièrement nous a permis d'être plus productives, mais aussi d'avancer de façon régulière. Enfin, nous avons également pu échanger nos compétences : quand l'une d'entre nous rencontrait des difficultés, l'autre apportait son aide, et inversement. Ce soutien mutuel a enrichi notre apprentissage.

B. Pistes d'amélioration

Si nous avons du temps pour poursuivre ce projet, plusieurs aspects pourraient être ajoutés ou améliorés. Tout d'abord, nous aimerions proposer un affichage plus visuel et ludique pour représenter l'état du potager. L'ajout de maladies affectant les plantes à des moments aléatoires permettrait de rendre la simulation plus réaliste. Aussi, un ajout majeur serait d'offrir la possibilité de choisir différents pays, chacun avec ses propres conditions climatiques et de temporalités différentes, pour varier les scénarios de jeu. Dans notre code actuel, nous avons implémenté un mode classique ainsi qu'un mode urgence. Cependant, nous aurions aimé améliorer le mode Urgence pour associer un outil avec un type d'urgence. Enfin, selon nous, une des premières améliorations à apporter serait l'implémentation d'une fonction de sauvegarde. En effet, à l'heure actuelle, il est nécessaire de recommencer une partie depuis le début à chaque session, ce qui peut être contraignant pour les joueurs qui disposent de peu de temps. Cette fonctionnalité permettrait de reprendre facilement le potager là où il avait été laissé.

VII. Conclusion

Le projet Ensemenc nous a permis de mobiliser l'ensemble des compétences acquises en programmation orientée objet à travers un simulateur de potager complet, interactif et immersif. Ce travail nous a offert une expérience concrète de développement logiciel en équipe, de la conception à la réalisation, en passant par les phases de test et de débogage. Nous avons su relever les défis techniques liés à la gestion d'un système complexe tout en veillant à proposer une expérience utilisateur cohérente et fluide. Ce projet a été à la fois formateur, stimulant et enrichissant, et ouvre la voie à de nombreuses pistes d'amélioration futures.