

Un jeu par Elian BENOIS

ENSEmenc

Elian BENOIS

1ère année - groupe 4
Programmation avancée

Etablissement : ENSC
Encadrante : Mme. TETELIN

2024 - 2025

Sommaire

Introduction.....	2
Possibilités offertes au joueur et déroulement d'une partie.....	3
En mode classique :	3
En mode urgence :	3
Ajouts créatifs.....	4
Modélisation objet réalisée.....	5
Les classes abstraites principales :	5
Les classes dérivées :	5
Tests réalisés.....	6
Tests unitaires :	6
Tests d'intégration :	6
Tests manuels :	6
Organisation de l'équipe et gestion de projet.....	7
Bilan critique.....	8
Annexe.....	9

Introduction

Ce rapport présente le développement d'un jeu de simulation de potager conçu pour une interface console, développé seul dans un cadre personnel et pédagogique.

L'objectif principal était de créer une application permettant de gérer un potager de manière stratégique, tout en mettant en pratique les concepts de programmation orientée objet et la gestion d'événements aléatoires.

Le joueur peut interagir avec un ensemble de parcelles, cultiver des légumes, faire face aux contraintes climatiques et aux attaques de nuisibles.

Ce document décrit en détail l'univers et les mécanismes du jeu, le déroulement typique d'une partie, la modélisation objet adoptée, les tests réalisés pour assurer la qualité, l'organisation personnelle du travail et une réflexion critique sur le projet.

L'univers du jeu et ses spécificités

L'univers proposé est celui d'un jardin potager, espace familial et facilement compréhensible, ce qui facilite l'immersion du joueur malgré l'absence d'éléments graphiques avancés. Le potager est modélisé comme une grille de terrains, chacun possédant des coordonnées précises et un type de sol, afin d'introduire un niveau de réalisme dans la gestion des cultures. Chaque parcelle peut accueillir un légume, qui évolue selon plusieurs paramètres tels que la croissance, l'état sanitaire, la nécessité d'arrosage ou l'exposition aux risques extérieurs.

Le choix de l'interface console, imposé pour la simplicité et la rapidité de mise en œuvre, a conduit à une présentation textuelle riche en informations mais claire, réparties sur plusieurs lignes pour chaque parcelle sélectionnée. Par exemple, le type de terrain, le nom et l'image simplifiée du légume, l'état de protection contre les nuisibles ou le climat, ainsi que l'avancement de la croissance sont affichés ligne par ligne.

L'une des spécificités majeures de ce jeu est l'intégration d'événements extérieurs qui influencent directement le potager. Un système de climat génère périodiquement des situations d'urgence, comme des conditions météorologiques extrêmes, tandis qu'une menace constante est représentée par l'attaque de rongeurs pouvant endommager les cultures. Le joueur doit donc non seulement cultiver mais aussi protéger activement ses parcelles, en utilisant les ressources à sa disposition (serres, engrais, arrosage). Cette dualité gestion/réactivité crée un équilibre intéressant entre stratégie à long terme et réponses immédiates.

Possibilités des joueurs et déroulement d'une partie

Dans cette simulation, le joueur contrôle l'ensemble du potager à travers une navigation simple mais efficace. Au début de la partie, il peut choisir la taille de son potager, ce qui influence la surface cultivable et la complexité de la gestion. De plus, le joueur sélectionne le pays où se situe son potager parmi plusieurs options proposées, telles que la France, Madagascar ou Placiland. Ce choix modifie certaines conditions climatiques et défis spécifiques, offrant ainsi une expérience de jeu variée selon la localisation.

Le déplacement dans la grille se fait à l'aide des touches fléchées, avec un système de bouclage aux limites, ce qui permet de parcourir aisément toutes les parcelles. Lorsqu'une parcelle est sélectionnée, ses informations détaillées sont affichées, offrant une vision complète de son état.

Selon que la parcelle est vide ou occupée par un légume, différentes actions sont disponibles. Si elle est vide, le joueur peut planter une graine parmi celles disponibles dans son inventaire, qui contient diverses espèces de légumes avec des caractéristiques variées (temps de croissance, besoin en eau, valeur marchande). Le magasin propose huit variétés de légumes différentes, chacune avec ses spécificités, que le joueur peut acheter pour diversifier sa production. Si un légume y pousse déjà, il peut être arrosé, fertilisé, protégé par une serre, déterré pour enlever une culture non désirée, ou récolté une fois mature.

Le déroulement d'une partie est structuré en cycles où le joueur alterne phases d'observation, de décision et d'action. Chaque action a un impact direct sur l'état des légumes : l'arrosage accélère la croissance, le fertilisant améliore la productivité, tandis que l'absence de protection peut entraîner la perte des cultures à cause des nuisibles. Le jeu introduit aussi des éléments aléatoires pour dynamiser l'expérience, comme les attaques sporadiques de rongeurs et les variations climatiques.

Par ailleurs, le joueur dispose d'un magasin intégré au jeu, où il peut acheter des graines supplémentaires ainsi que de l'équipement tel que l'engrais ou les serres, en utilisant l'argent accumulé grâce à la vente des récoltes. Cette fonctionnalité ajoute une dimension stratégique, puisqu'il faut gérer son budget pour optimiser la production.

Enfin, le jeu permet également de sauvegarder la progression à tout moment, offrant ainsi la possibilité de reprendre la partie ultérieurement sans perdre les avancées réalisées. Cette fonctionnalité est essentielle pour un jeu de gestion, car elle donne au joueur la liberté de gérer son temps et d'expérimenter différentes approches sans contrainte.

Le joueur doit ainsi gérer ses ressources, optimiser la production, et adapter ses stratégies en fonction des événements. Ce mélange entre gestion proactive et réaction face aux aléas fait la richesse du gameplay, même s'il reste simple dans sa conception. Le choix de développer une interface console limite les interactions possibles mais facilite la compréhension rapide des mécanismes.

La modélisation objet réalisée

Pour structurer ce projet, j'ai adopté une approche orientée objet rigoureuse afin d'organiser clairement les différentes composantes du jeu et leurs interactions.

- Classe Potager : Cette classe centrale gère la grille de terrains (*ListTerrains*). Elle s'occupe de la navigation dans le potager, de l'affichage global, et coordonne les événements externes (climat, rongeurs). Elle est responsable de l'appel des méthodes sur chaque terrain selon les actions du joueur.
- Classe Terrain : Chaque terrain est identifié par ses coordonnées et son type (par exemple, sol argileux, sableux, etc.). Il peut contenir un légume et des informations spécifiques comme la présence de rongeurs. La classe encapsule les méthodes pour planter, arroser, protéger, déterrer et récolter, ainsi que pour appliquer les conséquences des événements aléatoires.
- Classe Legume : Modélise les caractéristiques d'une plante (nom, temps de croissance, croissance actuelle, état, arrosage, protection, etc.). Elle possède des méthodes permettant d'évoluer dans le temps et d'interagir avec le joueur (actions possibles selon l'état, récolte).
- Classe Climat : Simule les conditions météorologiques et leurs effets, notamment des situations d'urgence qui peuvent affecter les cultures, forçant le joueur à adopter des stratégies défensives.
- Classe Inventaire : Regroupe les ressources du joueur : graines disponibles, engrais, serres, argent. Elle facilite la gestion des ressources consommables au cours du jeu.
- Classes héritières de Legume : Chaque classe dérivée modélise un type spécifique de légume avec ses propres caractéristiques (durée de croissance, besoins en eau, résistance aux attaques, rentabilité). Ces classes spécialisées héritent des propriétés et méthodes de la classe Legume, tout en adaptant ou étendant certains comportements pour refléter leurs particularités.
- Classes héritières de Terrain : Ces classes spécialisées représentent différents types de parcelles dans le potager, chacune avec des caractéristiques spécifiques adaptées à un type de terrain particulier. Chaque classe héritière étend la classe Terrain de base en modifiant ou ajoutant des attributs et méthodes propres à son type, comme des contraintes de culture ou des réactions différentes aux conditions climatiques.
- Classe Jeu : La classe Jeu constitue le cœur de l'application, orchestrant le déroulement global de la partie. Elle gère l'état du potager, la météo, les interactions avec le joueur, ainsi que la boucle principale du jeu.

L'interaction entre ces classes permet de créer un système modulaire et flexible. Par exemple, le potager interroge chaque terrain pour afficher son état ou appliquer des actions, tandis que les légumes exposent les actions possibles en fonction de leur état interne. Cette modélisation facilite aussi l'extension future, par exemple pour ajouter de nouveaux types de légumes ou d'événements.

Un diagramme UML a été réalisé pour clarifier cette architecture, mettant en lumière les relations d'agrégation, d'association ainsi que les méthodes principales de chaque classe.

Tests réalisés pour vérifier le bon fonctionnement de l'application

La validation de cette application a reposé sur une série de tests manuels et quelques tests unitaires simples réalisés dans le cadre du développement personnel:

- Tests unitaires des méthodes de gestion des légumes : Par exemple, la méthode de plantation vérifie que le légume est bien ajouté dans la parcelle, que les ressources dans l'inventaire sont décomptées correctement, et que les propriétés du légume (croissance, état) sont initialisées. De même, les méthodes d'arrosage, fertilisation et récolte ont été testées indépendamment pour vérifier leur impact sur l'état du légume.
- Tests de navigation dans le potager : Il était crucial de s'assurer que le déplacement entre les parcelles fonctionne correctement, avec une gestion appropriée du passage aux bords du potager (bouclage). Ces tests ont été faits en simulant plusieurs parcours de navigation, en vérifiant que la sélection correspond toujours aux coordonnées attendues.
- Tests d'affichage : La cohérence entre l'état réel des objets et les informations affichées dans la console a été vérifiée, notamment pour les différentes lignes d'informations affichées dans la méthode d'affichage du légume.
- Tests des événements aléatoires (climat et nuisibles) : En lançant plusieurs cycles, j'ai observé l'application correcte des événements aléatoires, comme la perte aléatoire des légumes non protégés par les rongeurs, ou les alertes climatiques. Ces tests ont permis de s'assurer que le jeu reste stable même lorsque des événements perturbateurs surviennent.
- Tests d'intégration : Enfin, des parties complètes ont été jouées, simulant une succession d'actions diverses, pour vérifier la cohérence globale et détecter d'éventuels bugs liés à l'interaction des différentes classes.

Ces tests ont permis d'identifier quelques problèmes mineurs (boucle infinie lors de certaines actions, affichage incorrect de certaines informations) qui ont été corrigés au fur et à mesure. L'absence de tests automatisés plus poussés est une limite, mais compréhensible dans le cadre d'un projet individuel et à petite échelle.

Gestion de projet : organisation personnelle et planning de réalisation

Étant seul sur ce projet, la gestion a été entièrement autodidacte et autonome. J'ai suivi une démarche structurée pour éviter de me disperser et assurer une progression régulière sur une période totale de 50 jours.

Le projet a été divisé en plusieurs phases :

Phase d'analyse et conception (environ 10 jours) :

Durant cette phase initiale, j'ai étudié en détail le cahier des charges afin de bien comprendre les besoins et spécificités du jeu. J'ai défini les fonctionnalités principales à implémenter, ce qui m'a permis de cadrer le projet. J'ai ensuite élaboré le diagramme UML pour modéliser les classes principales, en planifiant notamment la structure des terrains, légumes et interactions. Cette étape a été cruciale pour poser des bases solides avant le développement.

Phase de développement (environ 30 jours) :

Le cœur du projet a consisté en une implémentation progressive des classes et fonctionnalités. J'ai commencé par la modélisation des terrains et des légumes, en développant leurs propriétés et méthodes. Puis, j'ai intégré la navigation utilisateur dans la grille du potager, ainsi que les actions possibles selon les états des parcelles. J'ai ensuite ajouté la gestion des événements extérieurs, comme les conditions climatiques et les attaques de nuisibles. L'approche a été itérative, avec des tests fréquents pour valider chaque nouvelle fonctionnalité avant de passer à la suivante. Par ailleurs, j'ai développé les fonctionnalités complémentaires comme le magasin, la sauvegarde des parties et la gestion de l'inventaire.

Phase de tests, corrections et améliorations (environ 10 jours) :

Durant cette dernière phase, j'ai réalisé des tests manuels intensifs, couvrant tous les aspects du jeu, de la navigation à la gestion des cultures. J'ai corrigé les bugs détectés, amélioré la fluidité de navigation, et optimisé l'affichage dans la console pour assurer une meilleure lisibilité. J'ai également peaufiné les interactions pour rendre l'expérience plus intuitive.

La gestion du temps a constitué une difficulté majeure, car le travail personnel impose de jongler avec d'autres responsabilités. Pour rester efficace, j'ai organisé des sessions de travail courtes mais régulières, en priorisant les fonctionnalités essentielles avant d'aborder les raffinements.

J'ai utilisé des outils simples mais efficaces : un gestionnaire de tâches (To-Do list) pour suivre l'avancement des étapes, et Git pour gérer les versions de mon code. L'absence de collaboration a simplifié la communication mais a aussi nécessité une rigueur accrue dans la documentation pour ne pas perdre le fil du projet et assurer une bonne traçabilité des évolutions.

Bilan critique sur le projet

La réalisation de ce projet m'a apporté une expérience précieuse en programmation orientée objet, gestion de projet individuel et conception d'une application interactive en console. Le jeu fonctionne de manière satisfaisante, offrant une expérience ludique de gestion potagère avec un équilibre intéressant entre actions et événements aléatoires.

Cependant, plusieurs limites et axes d'amélioration apparaissent clairement. L'interface console, bien que fonctionnelle, limite la richesse visuelle et l'immersion ; un passage à une interface graphique permettrait d'améliorer l'ergonomie et le plaisir de jeu. La gestion des événements reste assez simple et répétitive, une complexification avec plus de types d'aléas serait souhaitable.

Du point de vue technique, certaines méthodes, notamment celles effectuant des recherches dans la liste des terrains, pourraient être optimisées (exemple : utilisation d'une structure de données plus adaptée, comme un dictionnaire pour accéder plus rapidement aux parcelles). Par ailleurs, l'ajout de tests unitaires automatisés et de tests d'intégration formels serait un atout important pour renforcer la qualité et faciliter la maintenance future.

Enfin, en travaillant seul, j'ai bénéficié d'une autonomie complète mais j'ai aussi ressenti le manque de retours extérieurs, qui auraient pu aider à identifier plus tôt certains défauts ou à enrichir les fonctionnalités. Malgré cela, ce projet reste une réussite personnelle et une base solide pour des évolutions ultérieures.

Conclusion

Pour conclure, ce projet de simulation de potager en console constitue une réalisation technique complète, intégrant des concepts fondamentaux de la programmation orientée objet et de la gestion événementielle. En développant seul cette application, j'ai pu concevoir un univers cohérent et stratégique, offrant au joueur une expérience de gestion mêlant anticipation, adaptation et prise de décision.

Le travail réalisé a permis de maîtriser la modélisation, la navigation utilisateur et la gestion d'événements aléatoires, tout en faisant face aux contraintes liées à une interface textuelle. Bien que perfectible, notamment au niveau de l'ergonomie et de la robustesse des tests, cette simulation démontre la pertinence d'une approche modulaire et évolutive. Elle ouvre la voie à des améliorations futures, notamment vers une interface graphique, une diversité accrue des événements, et une meilleure optimisation des performances.

Ce projet a surtout renforcé mes compétences personnelles en gestion complète d'un développement logiciel, de la conception à la livraison, tout en nourrissant mon intérêt pour la programmation orientée objet et la création d'applications interactives.

Annexe

Planning :

Tâches	1ère semaine	2ème semaine	3ème semaine	4ème semaine	5ème semaine	6ème semaine	7ème semaine
Phase d'analyse et conception							
Phase de développement							
Phase de tests, corrections et améliorations							