

Rapport de Projet : Plants vs. Zombies POO

Partie 1 : Analyse et Conception

Dans le cadre de notre projet de développement de jeu en Java, nous avons entrepris la création d'une version de Plants vs. Zombies. Cette entreprise a impliqué une analyse approfondie du cahier des charges initial, mettant en lumière les aspects essentiels à prendre en compte tout au long du processus de développement.

Objectif du Projet

Notre objectif principal était de créer une version fonctionnelle du jeu Plants vs. Zombies, en mettant un accent particulier sur la conception robuste des classes avant d'entamer la phase d'implémentation.

Analyse du Cahier des Charges

Nous avons minutieusement examiné chaque élément du cahier des charges afin de garantir une compréhension complète des fonctionnalités requises. Cela a jeté les bases pour une planification stratégique du développement.

Réflexion sur la Structure des Classes

Avant de débiter le codage, nous avons consacré du temps à une réflexion approfondie sur la structure des classes. Cette étape préliminaire visait à assurer une architecture solide et modulaire pour faciliter l'extensibilité du projet.

Résultats de la Phase d'Analyse et de Conception

Au cours de cette phase initiale, nous avons également identifié certaines zones cruciales du cahier des charges qui nécessitaient une attention particulière. Les problèmes potentiels ont été identifiés et des solutions préliminaires ont été envisagées. Parmi lesquels il y avait :

- Conception des Classes Principales

Nous avons initié le processus en concevant les classes principales du jeu, telles que celles représentant les plantes, les zombies, et l'environnement du jeu. Une attention particulière a été portée à l'établissement de relations claires entre ces entités pour assurer une gestion efficace du jeu.

- Mécanismes de Jeu de Base

Les mécanismes de base, tels que la plantation des plantes, le déplacement des zombies, et la gestion des ressources, ont été implémentés conformément aux spécifications du cahier des charges. Des tests rigoureux ont été effectués pour valider le bon fonctionnement de ces éléments fondamentaux.

- Interface textuelle jouable sur la console

L'interface textuelle a été conçue pour être simple, tout en offrant une expérience de jeu engageante. Elle permet aux joueurs de contrôler le jeu en utilisant le clavier.

- Gestion des Niveaux

La gestion des niveaux est basée sur un système d'étapes. Chaque étape comporte un certain nombre de niveaux, et chaque niveau est plus difficile que le précédent. Les joueurs doivent gagner chaque niveau pour progresser vers l'étape suivante.

Les niveaux sont générés de manière procédurale, ce qui signifie qu'ils peuvent être générés automatiquement à chaque fois. Cela permet de créer une variété de niveaux facilement et de garder le jeu intéressant.

Les niveaux peuvent contenir différents types de zombies, y compris des zombies de base avec des statistiques différentes, ainsi que des zombies spéciaux avec des capacités uniques.

Interactions Plantes-Zombies

Les interactions entre les plantes et les zombies sont basées sur un système d'attaque et de défense. Les plantes attaquent les zombies avec des projectiles ou des attaques de mêlée, et les zombies attaquent les plantes avec des projectiles ou des attaques physiques.

Les plantes et les zombies ont des statistiques différentes, telles que la santé, la vitesse, et la puissance d'attaque. Les plantes plus puissantes peuvent infliger plus de dégâts aux zombies, mais elles sont également plus vulnérables aux attaques.

Les zombies spéciaux ont des capacités uniques qui peuvent les rendre plus difficiles à battre. Par exemple, le zombie journal accélère lorsqu'il perd son journal, et le zombie sportif peut sauter au-dessus de la première plante qu'il voit.

Plantes

Les plantes du jeu sont divisées en plusieurs catégories :

- Tireurs : Les plantes tireurs tirent des projectiles sur les zombies.
- Plante Gelée : Les plantes gelées tirent des projectiles qui ralentissent les zombies.
- Tournesol : Les plantes génératrices de soleil génèrent du soleil, qui est utilisé pour acheter des plantes et des améliorations.
- Noix : Des plantes tanks sont robustes et peuvent absorber beaucoup de dégâts.
- Potatomine et Bombe Cerise : Les plantes mines explosent lorsque les zombies s'approchent ou juste après être posés.
- Chomper : Des plantes de combat rapprochées infligent de gros dégâts aux zombies lorsqu'elles sont à proximité.
-

Les bases du jeu ont été posées, couvrant les aspects cruciaux du cahier des charges. Cependant, des défis ont également émergé au cours de cette phase, notamment liés à la gestion de mémoire concurrente et à la création manuelle des niveaux. Les pistes d'extensions envisagées pour résoudre ces problèmes et améliorer le jeu seront explorées dans la partie suivante.

Remarques

- L'interface graphique n'a pas été initialement implémentée dans cette phase de développement, mais elle a été ajoutée avec succès dans une phase ultérieure du projet. Cette interface graphique offre une expérience visuelle améliorée, complétant ainsi la version textuelle jouable sur la console.
- L'équilibrage du jeu sera approfondi au cours des phases de développement.

Partie 2 : Pistes d'Extensions implémentées et problèmes rencontrés

Dans cette section, nous explorons les pistes d'extensions que nous envisageons pour améliorer notre projet de jeu Plants vs. Zombies en Java. Ces extensions visent à introduire de nouvelles fonctionnalités et à résoudre des problèmes identifiés, notamment ceux liés à la gestion concurrente et à la création manuelle des niveaux.

Gestion Concurrente des Thread

Un problème de thread concurrent a été identifié lors de la manipulation du tableau d'entités, entraînant des problèmes de mémoire et de synchronisation. Pour résoudre cela, nous prévoyons d'implémenter un modèle de gestion concurrente, utilisant des threads pour traiter les mises à jour du tableau de manière sécurisée et efficace.

Copie du Tableau

Afin de résoudre les problèmes liés aux threads, nous considérons l'option de renvoyer une copie du tableau au lieu de manipuler directement l'original. Cette approche réduit les risques de conflits mais coûte beaucoup d'opération.

Création Manuelle des Niveaux

Au départ, la création des niveaux se fait dynamiquement. Pour introduire davantage de contrôle et de personnalisation, nous envisageons d'implémenter une fonctionnalité permettant de créer manuellement les zombies qui apparaîtront à des moments spécifiques. Cela permettrait de contrôler parfaitement la création de niveaux en décidant avec précision de quand et quel type de zombies apparaît. Cependant, cette solution n'est pas optimale, car elle nécessite un effort manuel important pour créer un grand nombre de niveaux.

Génération Procédurale de Niveaux

Pour ajouter de la variété au jeu, nous avons exploré la possibilité d'implémenter une génération procédurale de niveaux. Cette approche permettrait de créer des niveaux de manière automatique tout en s'assurant qu'ils soient identiques, simplifiant ainsi la création d'un grand nombre de niveaux.

Optimisation des Performances

Nous planifions une optimisation des performances, notamment en réduisant la charge de travail des threads, en améliorant la gestion mémoire. Cela garantira une expérience de jeu fluide même sur des systèmes moins puissants.

Arrosoir

L'arrosoir est une arme qui permet aux joueurs d'améliorer les plantes, en augmentant une de leurs statistiques. Elle peut être gagnée en tuant un ennemi, lors de la mort d'un zombie il y a une petite chance qu'un arrosoir tombe.

Partie 3 : Pistes d'Extensions Non Encore Implémentées

Système de Récompenses et Évolutions

Un système de récompenses et d'évolutions permettrait aux joueurs de gagner des récompenses pour leurs performances, telles que des nouveaux objets ou des améliorations. Les plantes pourraient également évoluer au fil du temps, devenant plus puissantes et plus résistantes.

Avantages

- Ajouterait une nouvelle dimension de progression et de motivation au jeu.
- Permettrait aux joueurs de personnaliser leur expérience et de se sentir récompensés pour leurs efforts.

Inconvénients

- Pourrait être difficile à équilibrer.
- Pourrait nécessiter un investissement de temps et de ressources important.

Mode Multijoueur

Un mode multijoueur permettrait aux joueurs de s'affronter ou de coopérer en ligne.

Avantages

- Ajouterait une nouvelle dimension de socialité et de compétition au jeu.
- Permettrait aux joueurs de jouer avec leurs amis ou de se défier les uns les autres.

Inconvénients

- Pourrait être difficile à mettre en œuvre.
- Pourrait nécessiter un investissement de temps et de ressources important.

Variété d'Ennemis et de Plantes

L'ajout de nouvelles variétés d'ennemis et de plantes ajouterait de la variété et de la difficulté au jeu.

Avantages

- Rendrait le jeu plus stimulant et intéressant.
- Offrirait aux joueurs de nouveaux défis à relever.

Inconvénients

- Pourrait être difficile à équilibrer.
- Pourrait nécessiter un investissement de temps et de ressources important.

Niveaux Bonus et Challenges

L'ajout de niveaux bonus et de challenges spéciaux offrirait aux joueurs un nouveau contenu à explorer.

Avantages

- Rendrait le jeu plus long et plus attrayant.
- Offrirait aux joueurs des défis supplémentaires à relever.

Inconvénients

- Pourrait être difficile à mettre en œuvre.

- Pourrait nécessiter un investissement de temps et de ressources important.

Équilibrage et Tests : Il serait utile d’approfondir les efforts d’équilibrage du jeu pour ne pas avoir de plantes ou de zombies juger trop puissant, en ajustant les paramètres pour garantir une expérience de jeu engageante. De plus il serait judicieux d’approfondir les test avec un module dédiés comme JUnit plutôt que de les faire a la mains.

Conclusion :

Le projet de développement du jeu Plants vs. Zombies en Java a été un succès. Les objectifs du projet ont été atteints, et le jeu est maintenant fonctionnel et jouable.

Les pistes d'extensions envisagées pour améliorer le jeu sont prometteuses. Elles permettraient d'ajouter de nouvelles fonctionnalités et de résoudre des problèmes identifiés, notamment ceux liés à la gestion concurrente car la solution actuelle n'est pas optimale.

Diagramme des Class :

