

Sommaire

1. Introduction

- **Présentation et contexte du projet**

2. Objectifs et Contraintes

- **Attentes principales**
- **Limitations et défis**

3. Organisation et Gestion du Projet

- **Cahier des charges : Interface graphique, Version terminal, Logique de jeu**
- **Répartition des tâches**

4. Développement Technique

- **Mécanismes de mise à jour**

5. Conclusion

- **Résumé des acquis et perspectives futures**

Introduction :

Ce rapport présente notre projet de jeu de type "tower defense", inspiré par le célèbre jeu "Plants vs Zombies", mais avec une touche unique et originale. Nous avons choisi un concept où les ennemis sont des monstres insectoïdes et les défenseurs sont divers types de tours.

Le jeu se déroule sur deux axes : un mode texte pour une approche plus stratégique ou il y a une logique tour après tour (c'est-à-dire chaque tour il faut soit placer une tour ou attendre en tapant Next) et un mode graphique pour une meilleure expérience visuelle. Dans le mode texte, le joueur progresse à travers différents niveaux, chacun présentant un ensemble unique de monstres à éliminer. En mode graphique, le joueur peut choisir entre deux environnements : une forêt ou un océan, chacun créé avec soin à l'aide d'images modifiées sur Photoshop pour une esthétique distincte.

Le jeu propose plusieurs types de tours : la tour d'archer, le Mur et la tour du magique, chacun avec ses propres caractéristiques et coûts. Un système de monnaie intégré permet aux joueurs d'acheter et de placer stratégiquement ces tours sur le terrain de jeu. De plus, le jeu offre différents niveaux de difficulté ainsi que des modes de jeu variés pour enrichir l'expérience utilisateur.

L'objectif de ce projet est double : offrir un divertissement captivant tout en illustrant les concepts de la programmation orientée objet. Une attention particulière a été accordée à l'architecture et à la conception des classes, ainsi qu'à l'implémentation de la boucle de mise à jour interne du jeu, des aspects cruciaux pour le bon fonctionnement et la maintenabilité du code.

Dans les sections suivantes, nous allons :

Détailler les Objectifs : Clarifier les buts principaux du projet.

Examiner les Contraintes : Discuter des défis rencontrés.

Présenter la Gestion du Projet : Exposer notre organisation et méthodes de travail.

Explorer les Aspects Techniques : Aborder brièvement la structure du jeu et son développement.

Faire un Bilan : Conclure avec les résultats obtenus et les perspectives futures.

2. Objectifs

Notre jeu a pour but d'offrir une expérience amusante et stratégique. Voici nos principaux objectifs :

Diversité des Tours : On peut choisir parmi plusieurs types de tours, chacune ayant un coût, une puissance de tir et une portée différents. Cela permet de varier les stratégies.

Placement des Tours : Il est important de bien placer les tours. On ne peut pas mettre deux tours sur la même case, et une tour ne tirera pas sur un monstre qui est derrière elle. Il faut donc réfléchir où placer les tours pour qu'elles soient efficaces.

Sélection et Placement des Tours : On choisit les tours à partir d'un menu, puis on les place sur la carte. Quand une tour attaque un monstre, on voit des flèches, ce qui rend l'action plus dynamique.

Système de Monnaie : En éliminant des ennemis, on gagne de l'argent. Avec cet argent, on peut acheter de nouvelles tours ou améliorer celles qu'on a déjà.

Niveaux de Jeu : Le jeu comporte trois niveaux, chacun avec un nombre défini d'ennemis à battre. Si on réussit, on passe au niveau suivant.

Augmentation de la Difficulté : Avec le temps, il y a de plus en plus d'ennemis, ce qui rend le jeu plus difficile. Il faut donc constamment adapter sa stratégie pour survivre.

Choix de la Carte : On peut choisir entre différentes cartes, comme l'océan ou la forêt, chacune ayant ses propres défis.

Modes de Jeu : Il y a différents modes de jeu, comme le mode Marathon ou Normal, et on peut aussi choisir la difficulté entre facile, moyen et difficile.

Mvc : on a le package Ui qui contient View (qui contient les classes qui s'occupent d'affichage graphique comme Render) et Contrôle(MyButton , MyMouseListener pour les interaction entre le jeu et l'utilisateur)

Contraintes et Défis du Projet ;

Contraintes Techniques

Outils Utilisés : Le jeu a été créé avec Java AWT et Swing, en respectant les principes de base de la programmation orientée objet.

Simplicité : Nous avons évité les techniques trop complexes pour rester dans notre zone de confort.

Défis Rencontrés ;

Le plus dur était d'adapter le mode terminal au mode graphique , on a dû repenser certains logiques.

Stratégie de Développement : Nous avons commencé par une version texte pour nous assurer d'avoir un jeu fonctionnel avant d'ajouter les graphiques.

Limitations Temporelles

Pression du Temps : Avec les défis rencontrés, nous avons manqué de temps pour finaliser toutes les fonctionnalités prévues.

Ressources

Images : Trouver les bonnes images était compliqué. Par exemple, nous avons dû créer nous-mêmes la carte du jeu.

Choix des Éléments Graphiques : Sélectionner les images appropriées pour le jeu était crucial pour son apparence.

Organisation et Gestion du Projet

Début du Projet

Cahier des Charges : Nous avons commencé par définir les tâches et répartir le travail.

Choix des Éléments : Au lieu de zombies, nous avons choisi des monstres, et au lieu de plantes, nous avons opté pour des tours.

Structure des Données : Pour les monstres, nous avons utilisé une liste de listes, chaque sous-liste représentant une ligne de monstres. La carte du jeu a été structurée comme une grille de tours.

Développement

Mode Texte : Jules a travaillé sur le mode texte, mettant en place la logique fondamentale du jeu.

Mode Graphique : on a commencé à développer le mode graphique et le mécanisme de mise à jour , en construisant progressivement les différents écrans et fonctionnalités.

Interface Graphique : Cette partie a pris du temps et inclut des écrans comme 'Menu', 'Playing', 'Settings', et 'Second Menu'. Chaque écran correspond à un état différent du jeu, par exemple, 'Playing' est l'état actif du jeu.

Composants Interactifs : Des éléments comme 'MyButton' ou 'MyMouseListener' ont été utilisés pour l'interaction utilisateur.

Version Texte

Logique et Jeu sans Graphiques : Cette version permet de jouer sans l'interface graphique, en utilisant les entrées clavier pour interagir et afficher l'état du jeu après chaque tour.

Logique du Jeu ;

Tower et Monster : Représentent respectivement les tours et les monstres.

Map : Gère la disposition des tours et le mouvement des monstres. Les tours sont organisées en grille et les monstres en listes de listes chaque liste représentant une ligne d'une matrice où se trouvent les monstres .

Mécaniques de Jeu : Comprend des méthodes pour le mouvement des monstres et la défense des tours.

Classe Game : Gère la logique des combats, les conditions de défaite et le placement des tours.

Développement Technique

Boucle de Jeu :

Dans GameFrame, une boucle alterne entre le rendu graphique et la mise à jour du jeu pour maintenir un rythme régulier.

Dans le menu de configuration, les joueurs peuvent ajuster les paramètres du jeu, comme la difficulté et le choix de la carte. Ou appuie sur start pour tomber sur un second menu pour choisir le mode que l'on veut

Conclusion

En conclusion, nous sommes fiers des accomplissements que nous avons réalisés dans ce projet. Nous avons réussi à développer un jeu de Tower Defense fonctionnel et attrayant, qui offre une expérience de jeu stimulante et engageante. Nous avons créé une variété de tours et d'ennemis, mis en place un système de monnaie, et conçu une progression en difficulté pour offrir un défi constant aux joueurs. De plus, nous avons réussi à implémenter une interface utilisateur intuitive et des graphismes attrayants, rendant notre jeu accessible et agréable pour les joueurs.