

RAPPORT DE PROJET C++



21/12/20
16

Tower Defense

Ce rapport présente le projet, résume les choix de conception
et explique comment utiliser le jeu.

Rapport de projet C++

TOWER DEFENSE

Réalisé par Hamdi Gazzah

INTRODUCTION :

Le but du projet est de concevoir et développer un jeu de stratégie où l'objectif est de défendre une zone contre des vagues successives d'ennemis.

Ce projet a été proposé par notre enseignant afin de renforcer nos compétences en programmation orientée objet en C++.

J'espère que vous prendrez autant de plaisir à lire ce rapport que j'en ai pris durant tout le déroulement de ce projet.

PRESENTATION DU JEU :

Le Tower Defense est un style de jeu de stratégie, où le but est de défendre une zone contre des vagues successives d'ennemis se déplaçant sur un itinéraire, en construisant et améliorant progressivement des tours défensives.

Trois type de tours seront à disposition :

- « **Blue Tower** » qui est caractérisé par des tirs plus fréquent mais moins puissant.
- « **Green Tower** » qui tire des missile qui ralentissent l'ennemi.
- « **Red Tower** » qui possède des tirs qui ricochent sur les deux ennemis les plus près avec à chaque fois moins de puissance.

Les vagues des monstres peut comporter 3 types des monstres :

- « **Strong Monster** » qui est plus résistant mais plus lent.
- « **Fast Monster** » qui est plus faible mais plus rapide.
- « **Explosive Monster** » qui s'explode lorsqu'il meurt (change sa couleur en rouge) et blesse ses camarades.

ANALYSE :

Le Tower Defense fera partie de la catégorie des jeux avec chemin. En effet, il faut de faire suivre aux monstres un chemin prédéfini. L'idée de départ était de faire déplacer les monstres case par case, sans animations. Par la suite, j'ai décidé de gérer le déplacement en temps réel pour un meilleur rendu visuel.

En ce qui concerne la partie graphique, le style que j'ai choisi est très simpliste. Je ne possède pas les compétences nécessaires à l'élaboration d'un design plus proche de ce que l'on peut voir dans les autres jeux du genre. L'idée était de se contenter de formes géométriques simples. De plus, rester minimaliste m'a permis de nous concentrer davantage sur le code.

Lorsque j'ai imaginé ce projet, c'était avant tout en tant que joueur. En effet, avant de réfléchir au code, je pense au côté ludique, et agréable du jeu. Au moment de dresser la liste de mes objectifs, il m'est donc apparu clairement que les points primordiaux concerneraient ces sujets.

Les mécaniques de jeu se devaient d'être intéressantes, et facile à appréhender. Effectivement, le jeu ayant une durée de vie très limitée, le joueur devait dès les premières secondes comprendre et prendre en main la partie.

De plus, pour que l'utilisateur apprécie l'expérience, l'interface graphique ne devait pas le repousser. Sans chercher à être particulièrement travaillée, elle devait rester esthétique. Un des points indiscutables concernait la fluidité du rendu graphique. On ne peut pas apprécier un jeu à sa juste valeur si des défauts techniques sont omniprésents.

DEROULEMENT D'UNE PARTIE :

Une partie se déroule comme une suite de tours de jeu. Chaque tour de jeu se déroule en 3 phases :

Phase de choix de la carte :

Après avoir créer une nouvelle partie en cliquant sur « New », le joueur aura le choix de choisir une carte parmi 4 cartes prédéfinies, il peut basculer entre les cartes pour avoir une idée sur chacune puis cliquer sur « Start Game » pour lancer la partie, il peut aussi créer une carte personnalisée en définissant un chemin quelconque, sachant que tous ces cartes seront modifiables depuis des fichiers de configuration.

Une fois qu'il a terminé la création il doit cliquer sur « Save Map », là, une phase de test sera exécutée afin de s'assurer qu'il a bien choisit une case de départ et une d'arrivée. Si le test ne réussit pas le jour ne peut pas lancer la partie. Notant que le bouton « Clear all » permet de tout effacer.

Phase de placement des tours :

Durant cette période, le joueur positionne de nouvelles tours de défense en choisissant d'abord le tour souhaité et la position souhaitée sur la carte. Le prix du tour choisi sera décompté de la somme d'argent qu'il possède (indiquée par « Coins »).

Le jour peut, durant toute la partie, améliorer les tours déjà existantes toute en cliquant sur le tour elle-même, une petite fenêtre apparaît qui indique le niveau, la distance de tire, la puissance, et le prix de l'amélioration de tour. De plus un petit bouton s'affiche qui permettre d'améliorer le niveau le tour, bien-sûr en fonction de l'argent qu'il possède.

Une fois qu'il a terminé, il le signale au jeu en appuyant sur un bouton « Next Wave », ce dernier sera responsable au lancement des vagues des monstres et à cette instant l'action commence.

Phase d'action :

Durant cette phase, une nouvelle vague de N Monstre apparaît à une zone d'entrée l'un derrière l'autre et se déplace vers la zone de sortie, les tours de défense commencent aussi à tirer. Cette phase peut être vue comme la boucle des étapes suivant :

- demander de « Wave Manager » de chargé monstre par monstre depuis le fichier de configuration

- les monstres commencent à suivre le chemin.

- si le monstre est dans la portée d'un tour, le tour dirige son missile vers le cible et commence à tirer

- si le monstre n'a plus des points d vie (qui seront traduisent par une barre au-dessus de monstre) il sera détruit et le joueur reçoive l'argent qu'il a.

Le jeu se termine soit par une défaite du joueur quand un nombre suffisant d'ennemis ont traversé tout le chemin qui sera traduit par l'épuisement de nombre de vie, soit par une victoire du joueur quand toutes les vagues sont passées.

Durant cette phase le joueur peut sauvegarder l'état du jeu en cliquant sur « Save » pour qu'il puisse reprendre sa partie ultérieurement en choisissant « Load » au début du jeu.

CONCLUSION :

La première chose à souligner en ce qui concerne ce projet, c'est le plaisir que j'ai pris à le réaliser. Cela est bien entendu lié à la satisfaction d'avoir atteint les objectifs demandés.

D'un point de vue plus pédagogique, j'ai énormément appris tout le long du développement. J'ai désormais la sensation d'être réellement à l'aise avec les concepts de la programmation orientée objet, et plus encore avec la programmation orientée objet en C++.