

The background of the cover is white and features several large, abstract, rounded geometric shapes in shades of teal and grey. These shapes are scattered across the page, with some overlapping. The main title is centered in a bold, dark teal font. The authors' names are centered below the title in a smaller, dark grey font.

Rapport Tower Defense

ANNE PASSELÈGUE & GUILHEM POTIES

INTRODUCTION

Nous détaillerons dans ce rapport les différentes fonctionnalités du jeu de tower defense qu'a réalisé notre binôme. Nous ferons également une analyse du travail fourni, avec ce qui a bien fonctionné et les difficultés rencontrées. Nous évoquerons aussi les améliorations que nous aurions pu réaliser si nous avions eu plus de temps.

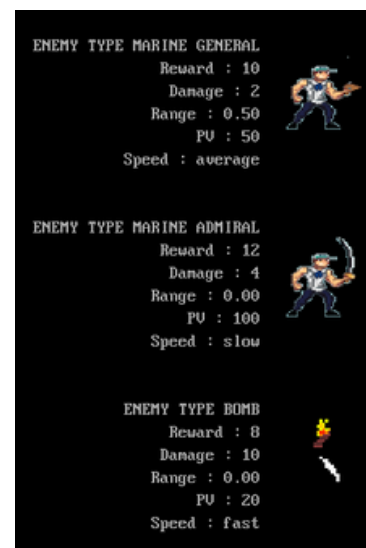
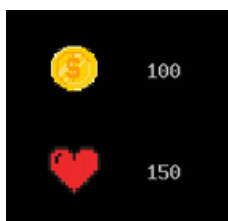
Vous pourrez retrouver notre projet sur le dépôt Git suivant :
https://github.com/sakured/Tower_Defense.

PRÉSENTATION DU JEU

Le principe de notre tower defense est le suivant. Le joueur devra empêcher des ennemis d'attaquer une cible. Les ennemis suivront le chemin le plus optimisé de la carte d'un point de départ (là où ils apparaîtront) jusqu'au point d'arrivée. Pour les en empêcher, le joueur pourra construire des tours de défense qui attaqueront les ennemis situés à une certaine distance des tours. Celles-ci peuvent être construites n'importe où en dehors du chemin en payant une certaine somme d'argent. La destruction d'un ennemi fera gagner au joueur de l'argent.

Lorsqu'un ennemi arrive au point d'arrivée, il tire et fait perdre un certain nombre de points de vie au joueur. Lorsque le joueur n'a plus de points de vie, il a perdu.

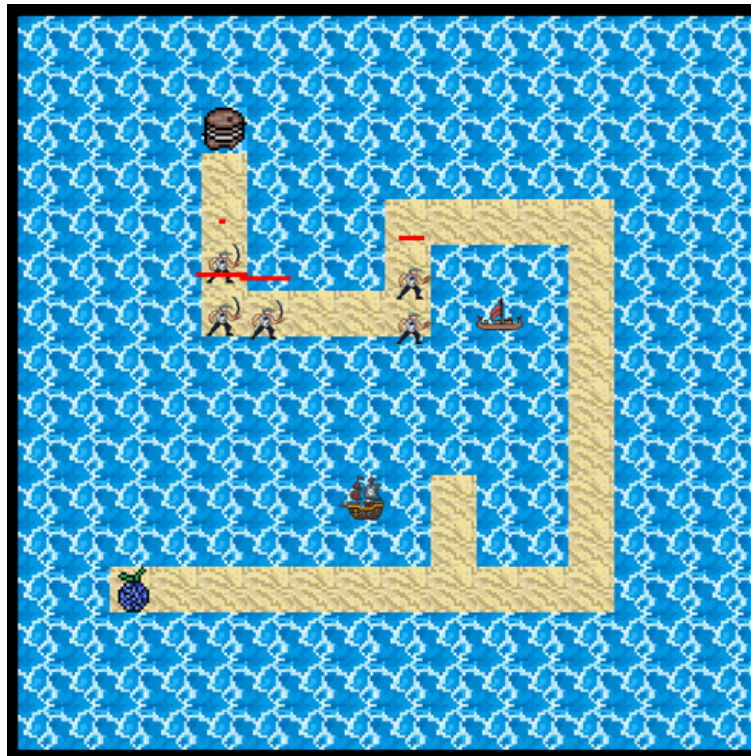
Il existe plusieurs types de tours avec des paramètres différents (coût, distance de tir, dommages faits aux ennemis) et plusieurs types d'ennemis (différents dégâts, vitesse, distance de tir, points de vie, gain rapporté au joueur en cas d'élimination). Ces informations sont affichées sur l'écran du joueur à tout moment.



Pour créer une tour, le joueur doit cliquer sur la touche correspondant au type de tour souhaité (ex : touche "1" pour une tour de type 1) puis cliquer à l'endroit où il veut placer sa tour.

Le joueur pourra mettre le jeu en pause à tout moment en cliquant sur le bouton "pause", et le redémarrer quand il le voudra.

Comme le thème de notre jeu est le célèbre manga One Piece, les tours seront représentées par des bateaux pirates et les ennemis par des membres de la marine. Nous avons dessiné nos sprites nous-même, excepté les personnages et les bateaux. Des barres de vies sont affichées au-dessus des ennemis afin de savoir s'il leur reste beaucoup de PV avant d'être éliminés par les tours.



Avant que le jeu commence, les règles sont affichées à l'écran afin que le joueur puisse comprendre le fonctionnement du jeu et les touches à utiliser.



Lorsque le joueur survit à huit salves d'attaques, il gagne la partie. Les salves sont lancées toutes les dix secondes et sont de plus en plus violentes.

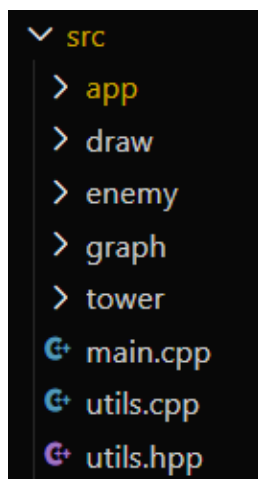
WELL DONE, YOU WON!

Si trop d'ennemis ont réussi à attaquer le point d'arrivée et qu'il n'a plus de points de vie, le joueur perd.

GAME OVER

ORGANISATION DU PROJET

Nous avons scindé notre projet en plusieurs sous dossiers dans le dossier src, afin de mieux organiser notre code.



La partie *app* contient les fichiers relatifs au paramétrage de l'application, avec entre autres l'appel des fonctions d'affichage, la mise à jour des déplacements ennemis, la détection des clics du joueur...

La partie *draw* contient toutes les fonctions qui vont dessiner l'interface de notre jeu.

La partie *enemy* contient tout ce qui est relatif aux ennemis. De même, la partie *tower* contient tout ce qui est relatif aux tours.

La partie *graph* contient tout ce qui concerne la gestion du graph de la map (afin que les ennemis puissent trouver le chemin le plus optimisé), ainsi qu'un fichier permettant de lire le fichier .idt qui permettra ensuite de construire le graph.

ANALYSE DU TRAVAIL

Nous avons au départ eu assez peur de la quantité de travail à fournir. De plus, on avait du mal à savoir par où commencer et comment organiser notre projet. Une fois la structure du projet établie, nous avons eu plus de facilité à nous lancer.

Étant donné que nous nous sommes basés sur une bonne structure, il a été relativement simple de savoir où coder quoi.

Notre principale difficulté a été l'affichage, que ce soit les informations du niveau ou l'affichage des tours au bon endroit. Une fois celles-ci affichées, l'affichage des ennemis a été assez simple, les deux fonctions d'affichages étant très similaires.

Une autre difficulté a été de calculer de manière optimisée le meilleur chemin pour les ennemis.

L'optimisation de la gestion des attaques des tours et des ennemis a aussi été assez complexe.

AMÉLIORATIONS POSSIBLES

Si nous avions eu plus de temps, nous aurions pu implémenter d'autres fonctionnalités.

Tout d'abord, nous aurions pu faire des boutons pour créer des tours plutôt que de les créer avec des touches du clavier. Cela aurait été plus ergonomique. Il en va de même pour mettre en pause le jeu, cela aurait été plus pratique de cliquer sur un bouton plutôt que sur la touche P.

Nous aurions aussi pu afficher des projectiles lancés par les tours pour attaquer les ennemis, ainsi que des projectiles lancés par les ennemis pour attaquer le point d'arrivée.

Nous aurions pu également gérer l'orientation des textures des ennemis selon leur direction avec plus d'implication sur la partie graphisme du jeu et un système de gestion du chargement des textures plus complet.

Nous aurions également pu améliorer la fonction de lecture du fichier .itd afin de vérifier sa validité. En effet, cela aurait permis de vérifier qu'il y avait bien le nom d'un fichier de map à lire, des couleurs définies et existantes pour les cases de type "in", "out" et "path", de vérifier qu'il y avait bien même nombre de nœuds qu'indiqué mais également qu'il existait au moins un chemin entre le point de départ des ennemis et leur point d'arrivée.

Enfin, nous aurions pu créer plusieurs maps avec plusieurs niveaux de difficulté, et implémenter la possibilité d'enchaîner plusieurs maps afin d'avoir plusieurs niveaux qui s'enchainent dans notre jeu.

Dans les niveaux plus compliqués, on aurait pu imaginer que les ennemis pouvaient aussi attaquer les tours et les détruire au bout d'un certain nombre d'attaques.

Dans l'ensemble nous sommes relativement satisfait du résultat de notre projet, bien que nous ayons un petit sentiment de frustrations vis à vis des améliorations ci-dessus que nous aurions pu réaliser avec un petit peu plus de temps.