Application Web
Rapport

Sommaire :

SAUDE David & FREITAS Thomas

1.Notice d'utilisation

Pour jouer au Tower Defense, exécuter le fichier tower_defense.html . Choisissez ensuite la carte sur laquelle vous désirez jouer. Cela fait apparaître un canvas sur lequel on pourra jouer. On peut sélectionner des tourelles en cliquant sur les carrés avec leurs noms. On a à disposition un menu qui affiche les informations sur le joueur. On peut placer des canons sur le canvas en cliquant au endroit où il est possible d'en placer, c'est à dire quand le cercle de placement devient bleu. Les ennemis apparaissent automatiquement, et de plus en plus vite. La partie s’arrête lorsque vous n'avez plus de points de vie. Rechargez la page pour relancer une partie, que ce soit sur la même carte ou sur une autre.

2.Description des structures de données

Joueur : la classe qui indique les caractéristiques du joueur, c'est à dire son or, ses ennemis battus et ses ennemis ratés (ses points de vie). C'est cette classe qui gère la pose de nouveau canon.

Canon : la classe qui définit les canons. Ces derniers gèrent eux même le fait de cibler un ennemi ou de tirer dessus. Les canons possèdent chacun un interval, toutes les (firerate) secondes dès leur création, ceux-ci visent un ennemi s'ils en ont pas ou détermine si l'ennemi visé est encore à portée, puis tire. Les canons ont une méthode qui leur sert à se dessiner et à dessiner leurs tirs.

Ennemi : la classe qui définit les ennemis. Les ennemis gèrent leurs déplacements sur les tableaux de mouvements. Ils peuvent indiquer s'ils sont morts ou s'ils ont gagné. Ils possèdent aussi une méthode servant à gérer leur perte de points de vie. Les ennemis ont une méthode qui leur sert à se dessiner ainsi que leur barre de vie.

La gestion des mouvements : Il s'agit en fait de comment sont gérés les mouvements en fonction des différents types de plateaux. Nos mouvements sont gérés par des tableaux de position de points. Quand

on passe à l'index suivant dans le tableau, cela correspond au point suivant le précédent. Les points doivent être perpendiculairement alignés (sinon problème relatif à la carte).

3.Algorithmes principaux

On en compte 2 principaux :

loadMap(x): la fonction démarre le jeu. Elle change le contenu de la page de façon manuelle, c'est à dire en modifiant l'attribut innerHTML de notre div principal. Ainsi, le chargement du background des cartes est géré via le CSS et en modifiant l'id du canvas. La carte est ensuite initialisée (avec la fonction suivante). Puis la fonction lance le démarrage du jeu, et boucle afin d'obtenir l'affichage et le moteur de jeu. La fonction gère aussi la fin de partie avec l'affichage de l’écran de fin.

-initMap(x) : la fonction servant à initialiser ce qui est relatif à la carte sélectionnée. Les tableaux de points sont donc initialisés, ainsi que le générateur de générateur d'ennemis.

4.Répartition des tâches

David : S'est principalement occupé de la transition du menu des cartes au canvas du jeu, donc de tower_defense.js et moveRelated.js, mais aussi des plus gros algorithmes du programme.

Thomas : A participé à l'implémentation des classes et le design des cartes. A proposé des idées pertinentes sur comment commencer le projet.

5.Difficultés rencontrées

En général, peu de difficultés trouvées. Les parties les plus dures à créer ont été le mouvement des ennemis, les conditions a réunir pour le bon placement des tourelles et le dessin du viseur des tourelles (beaucoup de notions de trigonométrie).

6.Remerciements spéciaux

Remerciements tout d'abord aux professeurs (de Web notamment), pour nous avoir transmis leur savoir sur la programmation Javascript.
Remerciements aux autres élèves de L3 INFO, pour leurs aides par ci par là.
Merci à tous.