

PROJET TOWER DEFENSE

Le *tower defense* (souvent abrégée en TD) est un type de jeu vidéo où l'objectif est de défendre une zone contre des **vagues successives d'ennemis se déplaçant librement**, en **construisant et en améliorant** progressivement des tours défensives.

Le but de votre jeu sera de survivre le plus longtemps possible à l'arrivée de vagues d'ennemies de **manière infinis et programmées**, qui essayeront d'atteindre la sortie du terrain se trouvant à une extrémité de votre carte. C'est-à-dire que **suite à l'élimination complète de la vague, une autre apparaît et ainsi de suite**, un **classement des joueurs en fonctions du nombre de vagues** devra donc être disponible.

Différents types de tours pourront être construites (par exemple type eau, feu ect...), ayant des **bonus contre certains types de monstres** respectivement (par exemple une tour eau à un bonus de dégât contre les monstres de feu)

Elles peuvent aussi avoir **certains effets** comme par exemple un effet de ralentissement si c'est une tour de glace, ou de brulure si c'est une tour de feu ect **La tour pourra aussi être améliorée** pour avoir une attaque supérieure ou améliorer son effet, toujours en échange de monnaie.

Une tour peut être **construite grâce à une monnaie** (par exemple de l'or), **récoltée à chaque fois qu'un monstre est abattu**. Un **nombre raisonnable de cette monnaie** devra être accessible **au début de la partie** pour que le joueur puisse commencer à créer ses tours.

Le **placement des tours est totalement libre** sur le terrain et va donc **définir le chemin** que devront emprunter les monstres pour arriver à la sortie (**les monstres ne peuvent pas traverser les tours**).

A chaque fois qu'un **monstre atteint la sortie, une vie du joueur doit être retirée**, si ce compteur tombe à **0, la partie est terminée et il est possible de recommencer**.

Le joueur a accès à tout moment à un **menu permettant de relancer ou quitter le jeu**.

Votre code devra respecter les diagrammes UML que vous allez effectuer en amont et ils devront évoluer en adéquation.