

Projet Prog 2 : Tower Defense

Compte Rendu 1

Raphaël LE BIHAN, Rida LALI

19 avril 2020

1 Présentation du projet

1.1 Présentation générale

L'objectif du projet est de réaliser un jeu de Tower Defense.

Le jeu présente une carte 2D où est située une tour appelée tour principale, reliée au bord de l'écran par un chemin. Lors de différentes manches, des ennemis arrivent du bord de l'écran et tentent de prendre d'assaut la tour principale.

L'objectif du joueur est de défendre la tour principale de l'assaut ennemi. Pour cela il peut disposer différentes tours de défense sur la carte, qui attaqueront les ennemis à proximité.

1.2 Format du rendu

Les explications qui suivent dans ce compte rendu, (en particulier l'organisation du code, et les détails de plusieurs classes et objets) sont détaillées dans différents autres documents joins avec celui-ci. Il n'est pas nécessaire de tout lire pour comprendre l'organisation du code source et les mécanismes du jeu. Les parties importantes à lire dans les autres documents seront signalées dans la suite du compte rendu.

2 Organisation du code

Détail important : presentation.pdf

Le code est organisé en différents packages :

- jeu : mécanismes généraux pour le déroulement d'une partie (carte, magasin, ticks, définition d'un ennemi/d'une tour, etc..)
- affichage : affichage d'une partie dans une fenêtre. Le package est divisé en deux sous-packages :
 - composants (les différents composants de la fenêtre, et leur disposition)
 - comportement (les effets des boutons, événements, etc..)
- strategies : algorithmes généraux pour les actions des ennemis et tours (recherche de cibles à attaquer/soigner, pathfinding, etc..)

- ennemis, tours, effets : implémentation des différents ennemis, tours, effets
- test : partie test, qui est lancée à l'exécution du programme

Dans les packages jeu et affichage, il a été choisi de ne pas implémenter les méthodes directement dans les différentes classes (abstraites ou pas), mais dans des objets (qui sont “comme” des objets compagnons, mais avec un nom différent parce qu'on ne peut pas séparer un objet compagnon et une classe dans deux fichiers en Scala). Ceci a été fait afin de diminuer la taille des fichiers (comme conseillé dans le sujet, sinon certains fichiers devenaient énormes).

3 Mécanismes : affichage et jeu

Détail important : presentation.pdf et mecanismes.pdf

4 Tours, ennemis, effets, manches implémentés

Les tours et ennemis implémentés ont différents comportements.

Les ennemis se déplacent différemment : certains choisissent d'emprunter seulement les chemins accessibles (avec aucune tour sur le passage), d'autres prennent le chemin le plus court même s'il contient des tours, il essaieront de les détruire en avançant.

Les ennemis et les tours choisissent des cibles différentes, par exemple la cible la plus proche, ou la tour principale en priorité, ou la cible la plus dangereuse, etc..

L'architecture choisie (organisation en package + utilisation de l'héritage) permet d'ajouter facilement des tours, ennemis et effets supplémentaires au jeu.

4.1 Tours

Au total 8 tours ont été implémentées, certaines peuvent être achetées directement, d'autres sont accessibles en améliorant une tour déjà présente (voir Magasin et Contrat dans detail_jeu.pdf).

- TourPrincipale : Non achetable, la tour principale est la tour à défendre
- Defenseuse : Tour classique d'attaque, qui attaque l'ennemi le plus proche
- Sniper : Evolution de Defenseuse, a une grande portée et attaque l'ennemi le plus dangereux
- Mortier : Evolution de Defenseuse, attaque l'ennemi le plus proche en faisant des dégâts de zone
- Lampadaire : Tour de soin, qui soigne la tour à portée ayant le moins de PV, et qui puisse être soignée
- Gluant : Evolution de Lampadaire, inflige un effet de ralentissement aux ennemis à proximité
- Yogi : Evolution de Lampadaire, elle est capable à la fois d'attaquer et se soigner

- Barrière : Tour inoffensive mais avec un grand nombre de PV, qui permet de bloquer le passage aux ennemis

4.2 Ennemis

Au total 5 ennemis ont été implémentés, avec des comportements différents

- Fourmi : Ennemi peu puissant, se déplace vers la tour principale en prenant un chemin sans tour.
- Racailou : Ennemi lent mais plus puissant, prend le chemin le plus court vers la tour principale.
- Kamikaze : Ennemi rapide qui évite les tours et rush vers la tour principale pour se suicider. Il n'attaque pas les tours mais explose et inflige des dégâts aux tours et ennemis proches lorsqu'il meurt.
- Soignant : Ennemi inoffensif mais qui se déplace vers les ennemis ayant subi des dégâts pour les soigner.
- Johnson : C'est le boss final, avec un grand nombre de PV.

4.3 Effets

Seul un effet a été implémenté : Ralenti, qui divise par deux la vitesse de déplacement d'un endommageable pendant un instant. Cet effet est infligé aux ennemis par la tour Gluant.

4.4 Manches

Le jeu contient pour l'instant une partie de quatre manches (trois manches classiques + le boss de fin). Comme pour les tours, ennemis et effets l'architecture permet de changer/ajouter facilement d'autres manches.

5 Affichage

L'interface graphique comporte différents boutons et commandes dont l'utilisation est (voulue) assez intuitive.

Le comportement général est détaillé dans `detail.affichage.pdf`

Le jeu comporte deux timers, `TimerAff` pour l'affichage (à fréquence fixe), `TimerJeu` (à fréquence réglable, s'arrête entre les différentes manches), afin de pouvoir régler la vitesse de déroulement du jeu.

Le magasin (`Inventaire`) et la zone d'affichage de la carte (`ZoneGrille`) héritent de la classe `Selectionneur`, qui permet de sélectionner une case en cliquant dans un quadrillage.

On peut utiliser la molette, ou les boutons pour zoomer, dézoomer et se déplacer sur la carte. De plus, au survol par la souris des informations sont affichées dans la zone inférieure droite de la fenêtre (`ZoneInfos`).

Les boutons `ATTAQUE`, `ACHETER`, `DETRUIRE` permettent de lancer la prochaine manche, acheter/faire évoluer ou supprimer une tour.

6 Evolution du projet

Certaines parties concernant l’affichage du jeu n’ont pas encore été implémentées par manque de temps, notamment :

- l’affichage des messages (ZoneGrilleM),
- le drag n drop pour se déplacer sur la carte (ZoneGrilleDND),
- les boutons pause et paramètres
- l’affichage de messages sur toute la fenêtre (par exemple pour afficher VICTOIRE/GAME OVER ou PAUSE)
- affichage des attaques des ennemis/tours sur la carte

On pourra aussi rajouter plus de tours, ennemis, effets ou tuiles avec de nouvelles mécaniques de jeu (ex : cases d’eau, tirs en ligne droite comme proposés dans l’énoncé).

Une autre fonctionnalité qui pourra être rajoutée est la lecture d’une partie (carte + magasin + manches) directement depuis un fichier au lieu de l’implémenter dans le code source.