
TP3 : TOURS DE DÉFENCE

Introduction

Pour le troisième TP, vous allez compléter le code pour un petit jeu de type « *tower defence* ». La section suivante va décrire rapidement le jeu : le déroulement du jeu et le code déjà écrit. Ensuite, il y aura la description des éléments que vous devez ajouter. Finalement, vous trouverez les directives de construction, les directives de remise et les éléments d'évaluation.

Description

Les jeux de type « *tower defence* » sont très simples. Ils consistent à diriger la défense d'un objectif que les ennemis tentent de détruire ou d'acquérir. En général, cette défense est construite à l'aide de tours. Les tours vont tirer sur les ennemis.

Le code existant contient déjà l'interface graphique. Les ennemis sont présents et avancent vers leur objectif. Il y a un type de tour avec les augmentations possibles pour ce type de tour.

Vous allez ajouter le code pour deux nouveaux types de tours.

La sous-section suivante contient une description des classes qui sont utiles pour terminer le TP. Ensuite, les actions que vous devez implémenter seront décrites.

Description des classes utiles pour votre TP

- `Tour`. C'est la classe mère pour les différents types de tour. Une partie du TP consiste à écrire des classes filles de cette classe. Il y a deux éléments importants pour votre TP.
 - `caracteristiques` : c'est un tableau qui contient les caractéristiques qui peuvent être augmentées par le joueur. Il sera utile d'aller consulter le code de `TourArcher` pour comprendre comment construire ce tableau.
 - `animer()` : c'est une méthode qui décrit le comportement de la tour. Par exemple, dans la tour existante (`TourArcher`), c'est la méthode qui compte le temps avant de pouvoir tirer et qui tire sur les ennemis. Cette méthode est appelée automatiquement 25 fois par seconde. Un vingt-cinquième de seconde sera appelé un `tic`).
- `TourEmplacement` : cette classe contient la description des tours que le joueur peut acheter sur un emplacement. Vous devrez ajouter l'information d'achat de vos tours dans cette classe.
 - `tours` : un tableau contenant la description pour chaque type de tours qu'un utilisateur peut acheter. C'est là que vous devez ajouter l'information pour l'achat des nouvelles tours.
- `Ennemi` : cette classe décrit les ennemis. Entre autres, elle contient le nombre de `pointVie` qu'un ennemi a. Aussi, elle contient la position courante de l'ennemi, utile pour vérifier la distance entre un

ennemi et une tour. Finalement, elle contient la vitesse de déplacement d'un ennemi et l'argent qu'il donne lorsqu'il est éliminé. Vous aurez à modifier cette classe.

- Jeu : c'est la classe centrale pour la gestion du jeu.

Description des ajouts à implémenter

Voici la liste des éléments que vous devez ajouter au jeu.

- Tour de glace : un nouveau type de tour qui permet de ralentir tous les ennemis qui sont proches de la tour. Cette tour coûte 50\$ à construire. Caractéristiques :
 - Rayon : le rayon d'action de l'effet de la tour. Initialement, il est à 24. Le joueur peut l'augmenter à 32 pour 40\$ et ensuite à 40 pour 60\$.
 - Réduction : le nombre (entier) de centièmes décrivant la réduction sur la vitesse de l'ennemi. Initialement, cette valeur est à 20. Elle peut être augmentée à 40 pour 60\$. Par exemple, si l'ennemi a une vitesse de 0.8, alors sa vitesse après une réduction de 20 centièmes sera calculée comme suit : $0.8 * (1 - (20 / 100))$.

Ces tours agissent à tous les tics. Leurs effets durent un seul tic. Donc les ennemis doivent revenir à leur vitesse originale après un tic. Ces tours doivent être de couleur bleue et les ennemis qui sont ralentis doivent devenir bleus. Les ennemis reprennent leur couleur originale lorsqu'ils ne sont plus au ralenti.

- Tour de feu : un nouveau type de tour qui enflamme les ennemis. Ces tours tirent sur l'ennemi le plus proche de la tour à condition qu'il soit à l'intérieur de la distance de tir de la tour. Si l'ennemi est déjà enflammé, alors la tour tire sur le deuxième ennemi le plus proche de la tour, à condition qu'il soit à l'intérieur de la distance de tir de la tour. Et ainsi de suite. Cette tour coûte 45\$ à construire. Cette tour tire à tous les 15 tics. Lorsqu'un ennemi est touché par cette tour, il perd un point de vie à tous les 5 tics. Caractéristiques :
 - Distance : la distance à laquelle la tour peut tirer. Initialement, cette valeur est à 20. Elle peut être augmentée à 30 pour 40\$ et ensuite à 50 pour 80\$.
 - Durée : le nombre de fois que l'ennemi perd un point de vie. Initialement, cette valeur est à 3. Elle peut être augmentée à 5 pour 20\$, ensuite à 7 pour 30\$.

Par exemple, si la tour a une durée de 5, alors l'ennemi perd 1 point de vie lorsque la tour lui tire dessus. Ensuite, il va perdre 1 point de vie quatre autres fois, à raison de 1 tous les 5 tics. Ces tours doivent être de couleur rouge et les ennemis qui sont enflammés doivent devenir rouges. Les ennemis reprennent leur couleur originale lorsqu'ils ne sont plus enflammés. N'oubliez pas d'augmenter l'argent du joueur lorsqu'il élimine un ennemi (voir le code dans `TourArcher`).

Pour tous les ajouts, c'est à vous de décider si vous avez besoin de méthode ou variable supplémentaire. **Vous devez ajouter une classe pour chaque type de tour ajouté.**

Directives

1. Le TP est à faire seul ou en équipe de deux.

2. Code :
 - a. Pas de `goto`, continue.
 - b. Les `break` ne peuvent apparaître que dans les `switch`.
 - c. Un seul `return` par méthode.
 - d. Additionnez le nombre de `if` (l'opérateur ternaire compte comme un `if`), `for`, `while`, `switch` et `try`. Ce nombre ne doit pas dépasser 5 pour une méthode.
3. Indentez votre code. Assurez-vous que l'indentation est faite avec des espaces.
4. Commentaires : NE COMMENTEZ PAS MON CODE, SEULEMENT LE VÔTRE.
 - a. Commentez l'entête de chaque classe et méthode **que vous ajoutez**.
 - b. Une ligne contient soit un commentaire, soit du code, pas les deux.
 - c. Utilisez des noms d'identificateur significatif.
 - d. Une ligne de commentaire ou de code ne devrait pas dépasser 120 caractères. Continuez sur la ligne suivante au besoin.
 - e.
 - i.
 1. La première ligne d'un commentaire doit contenir une description courte (1 phrase) de la méthode ou la classe.
 2. Assurez-vous de ne pas simplement répéter le nom de la méthode, donnez plus d'information.
 - ii. Ensuite, avant de placer les **tags**, placez une ligne vide.
 - iii. Placez les **tag** `@param`, `@return` et `@throws` au besoin.
 1. **@param : décrivez les valeurs acceptées pour la méthode. Vous devez commenter les paramètres de vos méthodes.**
 - o Vous pouvez seulement importer des librairies `java.util.*`.

Remise

Remettre le TP par l'entremise de Moodle. Placez vos fichiers `*.java` dans un dossier compressé de Windows, vous devez remettre l'archive. Le TP est à remettre avant le 15 décembre 23 :59.

Évaluation

- Fonctionnalité (10 pts) : Le jeu va être testé en jouant.
- Structure (2 pts) : veillez à utiliser correctement le mécanisme d'héritage et de méthode.
- Lisibilité (2 pts) : commentaire, indentation et noms d'identificateur significatif.