

Projet de programmation

Tower Defense

Fabrice Rossi

27 mai 2010

1 Rappel sur l'organisation et la notation

Le projet se déroule du 31 mai au 4 juin. La présence est obligatoire pendant les TH affectées au projet (en général de 8h30 à 18h30). Toute absence non justifiée entraînera une réduction significative de la note de l'étudiant(e) concerné(e).

Les objectifs pédagogiques du projet sont de développer les compétences en programmation et d'initier au travail en groupe. De ce fait, le travail de programmation doit être réparti de façon équilibré au sein de chaque groupe. Chaque élève doit développer une partie du logiciel. Les autres tâches associées au projet (communication avec l'encadrant, rédaction des documents, aspects multimédia, etc.) seront aussi réparties entre les élèves, sans que cela puisse justifier un désengagement des activités de programmation.

La note est basée sur une appréciation globale du projet éventuellement modulée par la contribution de chaque étudiant. Cette appréciation tient compte de la qualité technique du logiciel final, mais aussi de la qualité du cahier des charges, de celle des maquettes et du compte rendu final (cf la section 5), de la bonne organisation du groupe, de la répartition des tâches, etc. La soutenance finale est un élément très important de la notation et compte en général pour 20 % de la note. Cette soutenance aura lieu durant la semaine du 7 juin selon un calendrier fixé avec l'encadrant pendant le déroulement du projet.

En complément de ce rappel, le site de Sylvain Malacria¹ donne des conseils très utiles pour le bon déroulement du projet.

2 Description générale

Le but de ce projet est de réaliser un jeu de stratégie en temps réel (STR) basé sur le principe des tours défensives (*Tower Defense*). Ce type de jeu est une variante simplifiée des jeux classiques de STR, popularisée par des extensions développées par des amateurs (des *mods*) pour des jeux très populaires comme *Starcraft* et *Warcraft*.

Le principe du jeu est assez simple. Le joueur doit défendre sa base contre des ennemis contrôlés par l'ordinateur, au moyen de tours défensives. L'originalité du concept réside dans la dissymétrie totale des deux camps du jeu. L'ordinateur contrôle des troupes ennemies mobiles, dont le but est soit de traverser une zone de jeu, soit de voler un trésor dans la base du joueur et de l'emporter en dehors de la zone du jeu. Ces troupes ne peuvent pas attaquer directement le joueur et ne peuvent pas franchir la zone de terrain occupée par une tour. Le joueur place sur le terrain de jeu des tours qui resteront à l'emplacement de leur construction. Ces tours attaquent et tentent de détruire les troupes ennemies qui passent à leur portée. Le joueur doit donc exploiter à la fois le terrain de jeu et ses tours afin de contrôler les mouvements des troupes ennemies, l'objectif étant de les maintenir à portée de tir des tours le plus longtemps possible. Chaque troupe ennemie détruite apporte des ressources que le joueur utilise pour construire d'autres tours ou pour renforcer les tours existantes.

D'un point de vue pragmatique, un autre intérêt du concept est que l'intelligence artificielle de l'ordinateur peut être très faible sans nuire à l'intérêt du jeu. Ceci explique le foisonnement de l'offre en jeux *Tower Defense*, depuis des réalisations en *flash* jusqu'à des mises en œuvres sur téléphone mobile.

¹<http://malacria.fr/projet/Accueil.html>

3 Règles du jeu

Cette section détaille les grands principes du jeu à programmer. Les exigences en termes de programmation sont présentées dans la section 4. Les règles sont décrites de façon imprécise afin de laisser beaucoup de liberté de choix lors de la conception du jeu et pendant son réglage. **Attention cependant, la règle de non blocage n'est pas optionnelle** (cf les sections 3.1, 3.2 et 3.3).

3.1 Déroulement du jeu

Le jeu se déroule sous forme d'une succession de vagues d'ennemis, ceux-ci devenant de plus en plus nombreux et puissants au fur et à mesure des vagues. Lors d'une vague, des troupes ennemies entrent par divers points d'entrée de la zone de jeu (aussi appelée terrain) et tentent d'atteindre un but qui dépend de la variante du jeu. Dans une première variante, la zone de jeu comporte des points de sortie : chaque troupe tente alors de rejoindre le plus rapidement possible un point de sortie. Dans la variante de capture, le terrain comporte une base contenant des trésors. Les troupes ennemies cherchent alors à voler un trésor dans la base et à le rapporter au point de sortie le plus proche.

Le joueur peut à tout moment placer une nouvelle tour, renforcer une tour existante ou détruire une tour devenue inutile. Les deux premières actions nécessitent des ressources suffisantes. Le joueur débute le jeu avec des ressources initiales. Chaque destruction d'une troupe ennemie lui rapporte de nouvelles ressources. Le joueur ne peut pas bloquer l'ordinateur : il doit toujours exister un chemin entre une entrée et une sortie (ou entre une entrée et les trésors, puis entre les trésors et la sortie).

Le jeu se termine soit par une défaite du joueur quand un nombre suffisant d'ennemis ont traversé la zone de jeu (ou volé assez de trésors), soit par une victoire du joueur quand toutes les vagues sont passées.

3.2 Terrain

Le terrain (ou zone) de jeu est représenté par une grille rectangulaire constituée de cases carrées. Chaque case du terrain peut être

- franchissable : les troupes ennemies peuvent passer sur cette case ;
- constructible : le joueur peut placer une tour sur cette case ;
- une zone d'entrée : les troupes ennemies peuvent entrer dans le terrain de jeu par cette case ;
- une zone de sortie : les troupes ennemies peuvent sortir de le terrain de jeu par cette case ;
- une zone de trésors : les troupes ennemies peuvent voler des trésors à cet emplacement.

Les propriétés des cases ne sont pas toujours exclusives. Une case franchissable peut parfaitement être aussi constructible. De plus, la nature des cases peut dépendre des objets concernés. Certaines troupes peuvent être autorisées à franchir des cases infranchissables pour d'autres. De même on peut permettre la construction de certaines tours seulement sur certaines cases. Comme indiqué dans la section précédente, le blocage des troupes ennemies est interdit : une case normalement constructible peut perdre cette nature si elle est devenue le seul point de passage possible entre les entrées et les sorties du terrain. Notons enfin qu'une case peut être à la fois une entrée et une sortie dans le cas de la variante de vol de trésor (dans tous les cas, le terrain doit comporter au moins une entrée et une sortie, éventuellement confondues).

3.3 Troupes ennemies

Chaque vague d'assaillants comprend diverses troupes de différentes catégories. Chaque troupe est représentée par un icône sur le terrain. Le comportement des troupes est très simple. Une troupe se déplace horizontalement et verticalement seulement (pas en diagonale), depuis le centre d'une case vers le centre d'une case adjacente (la transition entre deux cases doit être régulière). En fonction du type de jeu, une troupe doit se déplacer soit vers une sortie, soit vers un trésor puis vers une sortie. Dans tous les cas, le déplacement se fait selon le chemin le plus court entre la position actuelle et le but. Une troupe ne peut bien sûr pas franchir une case infranchissable, mais, en revanche, plusieurs troupes peuvent occuper la même case, qu'elles soient de même nature ou pas. La construction d'une tour sur une case rend celle-ci infranchissable pour la plupart des troupes.

La catégorie d'une troupe influe essentiellement sur deux paramètres : la résistance de la troupe aux attaques des tours (cf la section 3.4) et sa vitesse de déplacement sur le terrain. L'état de santé d'une troupe

est représenté par des points de vie qui décroissent à chaque attaque. Quand leur nombre atteint zéro, la troupe est détruite et retirée du jeu. Elle rapporte au joueur des ressources qui dépendent de sa nature. Si elle portait un trésor volé, celui-ci est rendu au joueur selon un mécanisme à définir.

Comme indiqué dans la section 3.2, la catégorie de la troupe peut aussi influencer l'accessibilité des cases du terrain (on peut considérer par exemple des avions et des fantassins).

3.4 Tours

Comme les troupes, les tours sont réparties en catégories. La catégorie d'une tour influence les éléments suivants :

- son coût de construction ;
- ses attaques, en termes de nature, portée, rythme et dégâts occasionnés ;
- les zones sur lesquelles elle peut être construite ;
- ses capacités de renforcement.

Quand le joueur dispose de ressources suffisantes, il peut construire une tour. Il choisit d'abord la catégorie de la tour, puis positionne celle-ci sur le terrain, exclusivement sur une zone constructible pour cette catégorie et en respectant la règle de non blocage. La tour, représentée par une icône sur le terrain, devient alors opérationnelle : elle attaque toutes les troupes ennemies qui passent à sa portée, selon ses caractéristiques.

Le rythme des attaques détermine le temps que doit attendre la tour entre deux attaques. En général, les attaques sont ciblées sur une troupe précise, mais on pourra considérer des tours d'artillerie qui attaquent une case et donc toutes les troupes qui s'y trouvent. La nature de l'attaque dépend de la catégorie de la tour. On pourra considérer par exemple :

- des tours anti-aériennes, qui n'attaquent que les troupes « volantes » ;
- des tours d'artillerie ;
- des tours ralentisseuses qui ne causent pas de dégât aux troupes ennemies mais qui les ralentissent pour une certaine durée.

Plus généralement, l'intérêt du jeu dépend des choix stratégiques qui s'offrent au joueur. Un bon principe de conception consiste à proposer des tours qui sont efficaces sur certaines troupes mais pas sur d'autres, ce qui oblige le joueur à équilibrer son dispositif de défense. *A contrario*, une tour efficace contre toutes les troupes limite grandement l'intérêt du jeu. Il est important en outre que l'utilisation de plusieurs tours soit nécessaire pour détruire une troupe puissante.

Les tours sont modifiables et destructibles. À tout moment, le joueur peut décider de détruire une tour, ce qui lui rapporte une fraction des ressources utilisées pour construire la tour. D'autre part, les tours peuvent être renforcées, moyennant ressources, ce qui améliore leurs caractéristiques (rythme des attaques, portée et/ou dégâts).

4 Fonctionnalités exigées

Le projet devra implémenter en Java orienté objet un *tower defense* obéissant aux règles décrites ci-dessus. Le jeu sera contrôlé par une interface graphique simple permettant de placer des tours, de les renforcer et de les détruites.

En terme de règles du jeu, il est nécessaire

- de respecter la règle de non blocage ;
- de gérer le fait que les cases peuvent être franchissables ou pas, ainsi que inconstructibles ou pas ;
- de proposer au moins deux types de troupes ennemies ;
- de proposer au moins deux types de tours ;
- de gérer le renforcement des tours.

Il est très vivement conseillé d'organiser le programme de sorte à faciliter l'enrichissement du jeu par l'ajout de natures de cases, de catégories de troupes et de tours, d'attaques, etc.

5 Déroulement du projet

Pour assurer le bon déroulement du projet, les étudiants respecteront le calendrier suivant :

Lundi 31/05, 11h45 : organisation du groupe

Toute la communication entre le groupe et l'encadrant se fait par l'intermédiaire d'un des élèves, choisi par le groupe. La réalisation des trois maquettes prévues (cf plus bas) se fera de façon largement indépendante, ce qui conduit à la création de trois sous-groupes. L'organisation choisie est communiquée à l'encadrant dans la première matinée consacrée au projet.

Lundi 31/05, 18h30 : cahier des charges

La première tâche du groupe consiste à rédiger un cahier des charges. Il s'agit d'une description du produit attendu en fin de projet. La description contiendra notamment les éléments suivants :

- un « scénario » pour le jeu qui précisera le contexte choisi (par exemple, médiéval fantastique comme *Warcraft* ou futuriste comme *Starcraft*) ;
- une description détaillée des règles du jeu, dans le respect des principes généraux donnés dans la section 3. On décrira par exemple en détail les types de case choisis, les troupes et les tours ;
- une maquette graphique du jeu, c'est-à-dire un ou plusieurs dessins représentant l'organisation de l'interface qui sera proposée au joueur et précisant les zones d'interaction, les boîtes de dialogue éventuelles, etc.
- un embryon de mode d'emploi précisant l'interaction prévue avec l'interface, et les choix généraux proposés au joueur (par exemple le choix du terrain, du mode de jeu (avec ou sans trésor), etc.).
- une liste de quelques fonctionnalités optionnelles qui pourront être réalisées si l'avancement du travail le permet.

Le document sera transmis par mail à l'encadrant au format pdf à la fin de la première journée du projet.

Mardi 01/06, 18h30 : maquettes

Trois maquettes seront réalisées, sous forme de trois programmes Java relativement indépendants, développés en parallèle par trois sous-groupes (sans exclure des interactions).

La première maquette s'intéressera au calcul du plus court chemin entre deux cases d'un terrain de jeu dont les cases sont soit franchissable (coût de déplacement 1), soit infranchissable (coût de déplacement infini). On détectera en particulier le cas où aucun chemin n'existe entre deux cases.

La deuxième maquette proposera l'animation du déplacement d'une troupe en fonction d'un chemin. Les deux premières maquettes partageront des représentations informatiques des chemins et du terrain de jeu qui devront donc être réalisées en commun.

La troisième maquette s'intéresse à la gestion d'une file d'événements. Il s'agit de permettre à un programme de demander l'exécution de différentes tâches dans le futur, après un temps d'attente déterminé. La maquette s'appuiera sur les `Timer` de *Swing*, mais n'offrira pas d'interface graphique complexe.

Les trois maquettes seront transmises à l'encadrant par mail, sous forme d'export de projet Eclipse, à la fin de la deuxième journée de projet.

Mercredi 02/06, matinée : revue des maquettes

La matinée de la troisième journée sera consacrée à une revue des maquettes par les encadrants. Le but de cette revue est d'identifier d'éventuels problèmes (erreur de conception, bugs, etc.) et de proposer des pistes pour l'intégration des maquettes.

Mercredi 02/06, après-midi : intégration des maquettes

L'après-midi de la troisième journée sera consacré à l'intégration des maquettes, sans ajout de fonctionnalité. Le but est de disposer d'une maquette logicielle complète qui réalise le mouvement automatique des troupes ennemies en s'appuyant sur le gestionnaire de file d'événements pour lancer les vagues d'assauts.

Vendredi 04/06 : produit final

Les deux dernières journées sont consacrées à l'ajout des fonctionnalités manquantes dans la maquette pour la transformer en un produit final, ainsi qu'aux tests et réglages du logiciel. Les étudiants devront aussi rédiger un court compte rendu précisant de façon synthétique les différences entre le produit annoncé dans le cahier des charges et le produit livré. Le logiciel sera envoyé par mail à l'encadrant (sous forme d'un export de projet Eclipse), accompagné du compte rendu au format pdf, le vendredi soir.

Les étudiants qui le souhaitent pourront transmettre une version améliorée de leur projet au plus tard le dimanche 06/06 au soir. Cette version, ou à défaut celle de vendredi, servira de base à l'évaluation technique du projet.