

Projet de programmation

Wargame

Fabrice Rossi

8 avril 2010

1 Description générale

Le but de ce projet est de réaliser un jeu de plateau de type *wargame* dans l'esprit du logiciel libre *Battle for Wesnoth*¹.

Dans un *wargame*, deux joueurs (au moins) s'affrontent par l'intermédiaire d'armées virtuelles. Les règles des *wargames* peuvent être considérées comme une généralisation des échecs et se basent sur les grands principes suivants (détaillés dans la section 2) :

- chaque joueur dispose d'une armée constituée d'unités diverses ;
- le jeu se déroule sur un terrain constitué de cases régulières ;
- une case du terrain ne peut contenir qu'une seule unité au maximum ;
- chaque joueur joue à son tour en déplaçant ses unités ;
- une unité peut terminer son déplacement en attaquant une unité adverse, ce qui conduit à un combat.

Bien que proches des échecs, les *wargames* s'en distinguent fortement par de nombreux aspects, comme par exemple :

- à chaque tour de jeu, le joueur peut déplacer toutes ses unités ;
- le type de l'unité influence le nombre de cases qu'une unité peut parcourir dans un tour, pas la forme du déplacement ;
- le terrain peut influencer les déplacements et les combats des unités ;
- les combats sont intrinsèquement aléatoires : il n'y a pas de prise systématique comme aux échecs ;
- les armées peuvent être très variées et totalement différentes pour chaque joueur.



FIG. 1 – *Battle for Wesnoth*

2 Règles du *wargame*

Cette section détaille les grands principes du jeu à programmer. Les exigences en termes de programmation sont présentées dans la section 3. 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. Le but du projet n'est pas de reproduire toute la complexité de *Battle for Wesnoth* dont le développement a nécessité plusieurs années de travail à de nombreux programmeurs, mais simplement de s'inspirer de ce jeu.

2.1 Déroulement du jeu

Chaque joueur joue à tour de rôle. Pendant un tour, le joueur actif déplace ses unités (section 2.3) sur le terrain de jeu (section 2.2) selon les règles de déplacement (section 2.4). Chaque déplacement d'une

¹<http://www.wesnoth.org/>

unité peut se conclure par un combat (section 2.5). Le joueur termine son tour soit quand il ne peut plus déplacer aucune unité, soit s'il décide de ne pas utiliser les opportunités de mouvement qui lui restent.

2.2 Terrain

Le terrain est représenté par une grille **hexagonale** régulière comme celle de la figure 2. Chaque case du terrain est d'une nature donnée (plaine, montagne, rivière, etc.) qui influe sur les combats et sur les déplacements des unités. Par exemple, un cavalier sera défavorisé en forêt ou en montagne alors qu'il sera avantagé en plaine. Une case du terrain ne peut contenir qu'une unité à la fois (cf la section 2.4 pour les déplacements).

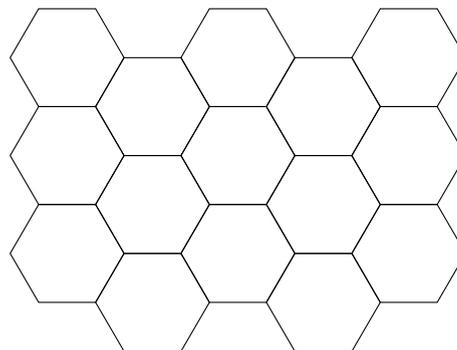


FIG. 2 – Une partie d'un terrain de jeu

2.3 Unités

Chaque unité de l'armée d'un joueur est d'une catégorie donnée (fantassin, archer, chevalier, etc.). La catégorie influe avant tout sur deux éléments fondamentaux du jeu : les déplacements (section 2.4) et les combats (section 2.5).

En outre, une unité possède des points de vie. Quand leur nombre atteint zéro, l'unité meurt et est retirée du jeu.

Plusieurs mécanismes de récupération des points de vie peuvent être envisagés, comme par exemple :

- des zones spéciales du terrain (des villages dans *Battle for Wesnoth*) permettent de récupérer des points de vie à la fin de chaque tour de jeu ;
- des unités particulières peuvent soigner les unités voisines.

2.4 Déplacement

Chaque catégorie d'unité possède des points de mouvement. Un point de mouvement donne en général la possibilité de bouger d'une case à une de ses six voisines. En fonction de la nature du terrain et de celle de l'unité, cette règle simple peut cependant être modifiée pour plus de réalisme. Un cavalier aura par exemple naturellement plus de points de mouvement qu'un fantassin. Cependant, le cavalier ne pourra pas escalader une montagne, alors que le fantassin le pourra (mais en consommant plus de points de mouvement par case).

En général, le déplacement est totalement libre : contrairement aux échecs, la forme de la trajectoire d'une unité est libre. De plus, bien qu'une case ne puisse contenir qu'une unité, il est toujours possible de traverser une case contenant une unité amie, à condition de ne pas terminer le mouvement sur cette case.

Cette règle générale de liberté souffre d'une exception très importante induite par la notion de **zone de contrôle**. Chaque unité contrôle les six cases qui entourent celle qui la contient. Quand une unité d'un joueur arrive sur une de cases contrôlées par une unité de l'adversaire, son mouvement s'arrête nécessairement, même s'il lui reste des points de mouvement.

Dans tous les cas, un déplacement peut se terminer par un combat, à l'initiative du propriétaire de l'unité qui se déplace. En d'autres termes, quand un joueur déplace une unité dans la zone de contrôle d'une unité ennemie, il peut déclencher un combat avec celle-ci.

2.5 Combats

Un combat implique deux unités, un attaquant et un défenseur. Chaque catégorie d'unité possède un ou plusieurs types d'attaque, par exemple une arme de corps à corps (une épée) et une arme de combat à distance (un arc). L'attaquant choisi le type d'attaque et le défenseur ne peut répondre que par un type correspondant. Le mécanisme exact de résolution des combats est laissé à l'initiative des étudiants, en s'appuyant sur les principes suivants :

- une arme cause des dégâts quand elle touche : ces dégâts correspondent à une perte de points de vie pour l'unité touchée ;
- la nature de l'unité touchée peut influencer les dégâts subis (possession d'une armure, par exemple) ;
- une arme est caractérisée par une probabilité de toucher l'adversaire ;
- le terrain et la nature des unités peut influencer la probabilité de toucher ;

- une arme est plus ou moins rapide (par exemple une arbalète a un rythme de tir très faible comparé à celui d'un arc) ce qui se traduit par un nombre d'attaques possibles dans un combat ;
- l'attaquant attaque en premier, puis le défenseur riposte. Cette alternance a lieu attaque par attaque (cf point précédent).

3 Fonctionnalités exigées

Le projet devra implémenter en Java orienté objet un *wargame* obéissant aux règles décrites ci-dessus. Le jeu permettra l'affrontement de deux joueurs en mode *hot seat* : le premier joueur effectue son tour puis laisse la place au second pour qu'il joue, etc. Le jeu sera contrôlé par une interface graphique simple permettant de déplacer les unités et de choisir l'arme utilisée pour un combat. On pourra utiliser les éléments graphiques de *Battle for Wesnoth*.

En terme de règles du jeu, il est nécessaire

- d'utiliser un terrain hexagonal ;
- de gérer au moins deux natures différentes de terrain (avec une représentation graphique associée et une influence sur les unités) ;
- de gérer au moins deux catégories d'unité différentes (avec une représentation graphique associée et des différences notables au niveau des déplacements et des combats) ;
- de gérer les déplacements comme indiqué, en particulier en tenant compte de la zone de contrôle ;
- de faire jouer un rôle au hasard dans la résolution des combats ;
- de gérer au moins deux attaques différentes par unité.

Les mécanismes de démarrage et de fin du jeu sont laissés à l'imagination des étudiants. Une solution simple consiste à adopter une approche inspirée du jeu d'échecs : chaque joueur possède une armée identique à celle de l'autre ; le vainqueur est celui qui conserve la (ou les) dernière(s) unité(s) vivante(s).

Il est très vivement conseillé de profiter des mécanismes orientés objets pour faciliter l'enrichissement du jeu par l'ajout de natures de terrain, de catégories d'unités, d'attaques, etc. Cependant, l'évaluation du projet se basera avant tout sur les fonctionnalités du jeu et sur la qualité de l'implémentation. Des extensions du jeu ambitieuses en terme de programmation et/ou de concepts informatiques comme l'affichage des scénarios de sortie possible d'un combat (avec les probabilités associées) ou le support du jeu en réseau seront beaucoup plus valorisées que l'ajout de nombreuses catégories d'unités, par exemple.

4 Déroulement du projet

Pour assurer le bon déroulement du projet, divers éléments d'analyse et de programmation doivent être remis de façon régulière aux enseignants chargés du suivi, selon le calendrier suivant :

Lundi 12/04, 13h30 : cahier des charges

Il s'agit d'une description du produit attendu en fin de projet (cf le site de Sylvain Malacria²).

Mardi 13/04, 8h30 : scénario d'un tour de jeu

Il s'agit d'une description précise du déroulement d'un tour de jeu, du point de vue interne au programme.

Mercredi 14/04, 8h30 : maquettes

Deux maquettes seront réalisées, sous forme de deux programmes Java indépendants.

La première maquette proposera un affichage du terrain de jeu³, en particulier une matérialisation de la grille hexagonale. Idéalement, la maquette devra permettre une interaction minimale, par exemple le déplacement d'une unité (sans respect particulier des règles du jeu à ce stade, c'est-à-dire un déplacement vers une case quelconque du terrain). On pourra aussi étudier l'affichage d'une partie du terrain quand celui-ci est plus grand que la fenêtre de l'application.

La deuxième maquette s'attachera à la représentation informatique des unités et surtout aux combats. Elle devra afficher de façon textuelle le déroulement d'un combat entre deux unités selon les règles choisies par les étudiants.

²<http://malacria.fr/projet/Accueil.html>

³Cf les exemples du cours MVC : <http://apiacoo.org/teaching/java/index.fr.html>