

# Projet Approche Orientée Objet – IATIC 3 - 2018/2019

## Wargame

Dhekra ABOUDA ([dhekra.abouda@gmail.com](mailto:dhekra.abouda@gmail.com))

L'objectif de ce projet est de réaliser une version informatique d'un jeu de Wargame tactique tour par tour. Ce sujet décrit les fonctionnalités attendues du jeu sans toutefois imposer tous les détails (univers du jeu, types d'unité, formules de calcul, . . .). Quand ce n'est pas explicitement imposé par l'énoncé, vous avez toute latitude de décider du choix le plus pertinent pour votre implémentation.

## 1 Le jeu

Un wargame (jeu de guerre) est un jeu permettant à un ou plusieurs joueurs de simuler des batailles. Un wargame est composé d'un système de jeu (règles) et de scénarios. Dans un wargame tactique, un scénario représente une bataille où évoluent des unités d'au maximum quelques dizaines de soldats (compagnie). L'univers dans lequel le jeu se déroule est libre. On peut citer par exemple : historique (Mémoire 44, War in the east), fantastique (Battlelore, Battle for Wesnoth), science-fiction (Full metal planet, Crimson \_eld), . . .

### 1.1 Plateau de jeu

Le jeu se déroule sur un espace de jeu découpé en cases hexagonales (voir la figure 1). Un hexagone représente l'unité de lieu et est d'un certain type (plaine, colline, montagne, . . .). Le jeu devra comporter au minimum cinq types différents de terrain.



Figure 1 \_ Exemple de Battle for Wesnoth

Le type de terrain a un effet sur les unités qui s'y trouvent (bonus d'attaque ou de défense, points de déplacement, . . .) (voir par exemple le tableau 1). Vous êtes libre d'adapter les types de terrain et les effets à votre contexte. Les effets peuvent varier en fonction du type d'unité (le type d'unité « elfe » se déplacera par exemple plus vite dans la forêt). Chaque hexagone ne peut accueillir qu'une seule unité.

### 1.2 Unités

Chaque unité possède un type (infanterie légère, infanterie lourde, . . .). Le jeu devra proposer au moins cinq types différents d'unité dont au minimum une disposant de capacité d'attaque à distance (« archer » par exemple).

Terrain	Points de déplacement	Bonus défense
Village	1	40%
Forteresse	1	60%
Eau profonde	-	0%
Plaine	1	20%
Forêt	2	40%
Colline	2	50%
Montagne	3	60%

« - » signifie inaccessible.

TABLE 1 – Exemples d'effets de terrain

Le type d'unité définit le potentiel d'attaque, le potentiel de défense, le potentiel de déplacement, le champ de vision et l'état initial (points de vie ou PV) de l'unité (voir par exemple la tableau 2). Vous pouvez adapter les types d'unités et la valeur des caractéristiques à votre contexte.

Unité	Att.	Déf.	Dépl.	Vision	PV
Infanterie	5	3	6	4	28
Infanterie lourde	10	10	4	4	38
Cavalerie	8	3	8	6	38
Mage	5	1	5	5	24
Archer	6	2	5	7	33

TABLE 2 – Exemples de caractéristiques d'unités

Le potentiel d'attaque et de défense définissent la puissance de l'unité durant un combat. Plus une unité dispose d'un potentiel d'attaque élevé, plus elle fera de dégâts lors d'une attaque. Plus son potentiel de défense est élevé, mieux elle résistera aux attaques adverses. Une unité peut disposer de plusieurs « armes » et donc d'un potentiel d'attaque par arme. Le potentiel de déplacement indique la vitesse de déplacement de l'unité, i.e. la distance qu'elle peut parcourir en un tour. Le champ de vision précise jusqu'à quelle distance l'unité peut voir. Au-delà, le brouillard de guerre occulte les événements. Ces quatre caractéristiques sont immuables tout au long du déroulement du scénario. L'état de l'unité va cependant varier en fonction des dégâts subits lors de la bataille.

### 1.3 Déplacement

La capacité de déplacement d'une unité se mesure en nombre d'hexagones. Durant un tour, une unité peut donc parcourir au maximum un nombre d'hexagone égal à son potentiel de déplacement initial. Lors d'un déplacement, le potentiel de déplacement courant change également en fonction du type de terrain traversé. Par exemple, une unité d'infanterie lourde pourra traverser dans le même tour deux hexagones de forêt mais pas un hexagone de forêt et un de montagne.

Lorsqu'une unité se déplace sur une case contenant une unité adverse, elle l'attaque. L'attaque n'est donc possible que si l'unité dispose encore de suffisamment de points de déplacement pour atteindre l'adversaire. Si l'unité attaquée est détruite, l'attaquant prend sa place. Sinon, aucun déplacement ne se produit.

### 1.4 Résolution des conflits et dégâts

Durant sa phase de déplacement, une unité peut attaquer une unité ennemie qui se trouve à portée. Dans ce cas, la résolution du conflit doit tenir compte des potentiels d'attaque et de défense des deux unités ainsi que du type de terrain. Chaque unité pourra subir des points de dégâts qui se déduiront de l'état de l'unité. Eventuellement, un peu de hasard pourra être introduit dans le calcul pour éviter des résultats trop répétitifs.

Une unité blessée qui ne se déplace pas et qui n'est pas attaquée lors d'un tour se repose et récupère donc une partie de ses points de vie. Une unité ne peut jamais dépasser son nombre initial de PV.

Les formules de calcul pour les dégâts et la récupération des points de vie sont laissées à votre appréciation.

**Exemple de calculs** Le conflit oppose une unité d'infanterie lourde (PAil = 10) qui attaque une unité d'infanterie (PDi = 3, PV = 28) se trouvant dans une forteresse (bonusdef = 60%). Le potentiel de défense de l'infanterie est modifié par le bonus du terrain ( $PDi = 3 + [3 * 60\%] = 3 + [1.8] = 5$ ) où  $[x]$  est la partie entière supérieure de  $x$ . Les dégâts bruts subis par l'infanterie sont  $D_{brut} = PAil - PDi = 10 - 5 = 5$ . L'application d'un peu d'aléatoire donne  $D = D_{brut} + random(-[50\% * D_{brut}], + [50\% * D_{brut}])$ . Supposons que la fonction random retourne -2 alors  $D = 5 - 2 = 3$ . Après le conflit, l'unité d'infanterie aura donc  $28 - 3 = 25$  PV.

Supposons que le taux de récupération des unités est de 10% par tour. Si l'infanterie ne se déplace pas le tour suivant et n'est pas attaquée à nouveau, elle récupère  $[10\% * 28]$  PV, soit 3 PV. L'unité retrouvera donc  $25 + 3 = 28$  PV.

## 1.5 Tour de jeu

Le jeu est donc une succession de tours de jeu. Chaque tour permet successivement aux adversaires d'effectuer diverses actions. Un joueur peut jouer tant qu'une de ses unités possèdent des points de déplacement. Quand un adversaire a terminé ses déplacements, il passe la main au joueur suivant. Eventuellement, il peut rester des points de déplacement à certaines de ces unités.

## 1.6 Conditions de victoire

Dans un scénario, chaque adversaire peut avoir des conditions de victoire différentes. Deux conditions de victoire sont possibles :

- Destruction complète de l'armée adverse : cela représente la réussite de l'attaque d'un objectif, ou
- Atteindre un numéro de tour sans avoir été détruit, i.e. après un nombre donné de tours, un joueur possède encore au moins une unité : cela correspond à la réussite de la défense d'un objectif.

# 2 Le projet

Ce projet va donc consister à proposer une version du jeu jouable sur ordinateur.

## 2.1 Jeu multi-joueurs

Le jeu doit permettre de jouer à plusieurs. Dans ce projet, les joueurs pourront être des joueurs humains ou des IA (joueurs contrôlés par l'ordinateur). Le programme fonctionnera sur une seule machine (pas en réseau). Dans le cas de plusieurs joueurs humains, ces derniers joueront leur tour l'un après l'autre.

## 2.2 Joueur ordinateur

Lors du démarrage de la partie, il sera possible de choisir le nombre de joueurs et de décider si un ou plusieurs joueurs sont contrôlés par l'ordinateur. Eventuellement, il devra être possible de démarrer une partie composée uniquement de joueurs ordinateur. Le joueur ordinateur devra bien sûr respecter les règles du jeu. Cependant, aucune contrainte n'est imposée quant à son niveau de jeu. Par exemple, il pourra choisir au hasard parmi les coups autorisés.

## 2.3 Interface utilisateur

Peu de choses sont imposées au niveau de l'interface graphique. Néanmoins, un soin particulier devra être apporté à sa clarté et à son ergonomie. Une aide devra être proposée au joueur débutant pour l'aider à assimiler les règles du jeu lors de ses premières parties. La forme que prendra cette aide n'est pas imposée.

Le programme aura également pour rôle de vérifier les actions des joueurs humains. Seules les actions valides seront possibles. Enfin, l'application ne montrera au joueur que ce qui lui est visible en respectant la règle du brouillard de guerre.

## 2.4 Sauvegarde des parties

Les parties pouvant durer longtemps, le logiciel devra fournir un moyen de sauvegarder le jeu en cours. Ces sauvegardes pourront bien sûr être rechargées par la suite pour continuer la partie.

## 2.5 Extensions facultatives

Une ou plusieurs des extensions facultatives ci-dessous pourront être intégrées au logiciel :

**Éditeur de scénario** Ce mode permet de créer ou de modifier des scénarios.

**Campagne** Une campagne est une succession de scénarios. L'intérêt d'une campagne réside dans le fait que les unités gagnent de l'expérience dans un scénario et que l'on choisit en début de scénario, les troupes enrôlées. Lors d'un scénario ultérieur de la campagne, les unités expérimentées pourront être réutilisées.

**Evènements extérieurs** Un scénario peut être « perturbé » par des évènements extérieurs (arrivée de renfort, évènement climatique, ...).

**Actions d'opportunité** Lors du tour d'un joueur, la main peut passer temporairement à un autre joueur. Par exemple, une unité qui entre dans le champ de vision d'une unité ennemie pourrait alors être attaquée par cette dernière.

**Ligne de tir** Les attaques à distance ne sont possibles que si la ligne de visée est suffisamment dégagée (pas d'autres unités, pas de montagnes).

**Pression du temps** Les joueurs doivent réaliser leur tour de jeu dans un temps donné. Eventuellement, ce temps pourrait être différent pour chaque joueur. Lorsque le temps est écoulé, la main passe automatiquement au joueur suivant.

## 2.6 Contraintes techniques

Le projet sera réalisé en langage Java. Les fonctionnalités disponibles dans le langage (collections, exceptions, ...) devront être utilisées au mieux. L'interface graphique utilisera la bibliothèque Swing. L'utilisation de bibliothèques tierce est autorisée (Google Guava, ...). Le code source devra respecter les conventions de codage de SUN. La documentation du code sera conforme à l'outil standard javadoc. Le projet final devra fonctionner et se compiler en dehors de tout IDE (avec l'outil maven par exemple). Idéalement, le projet sera fourni sous la forme d'une archive au format jar et « exécutable ».

## 3 Organisation et évaluation

Ce projet est à réaliser par groupe comme suit :

Groupe 1	Groupe 2	Groupe 3	Groupe 4	Groupe 5	Groupe 6	Groupe 7	Groupe 8	Groupe 9
AISSAMI	TERUEL	OUZZMAN	MBELE NZUE	DOLIMAN	DUREY	FANG	DEMINE	GABIOU
COCHET	ANDRIANAIVOJAONA RASON	THIOUNE	RAJOSOA	GRATADOUR	KETTEMAYER	KHLIF	FERRATON	KRIAA
DONA	CONZATTI	APETE	THOMAS	LAKSIR	MIRI	MONTERO	KOUNDOUL	MOURAO
HAMMAR	DORGAA	DALESME	BEN MABROUK	NDOYE	SAMENE-TIAH	ROUMEC	MOUHANI MIKASSISSA	SIDIBE
LORIOT	HOUDAYER	DU COULOMBIER	DANG	SIMON-ROUQUIE	BENCHIKH	TOUMI	ROUSSEL	TSEDE
OUASTI	MARLIER	JEUATSA LANDO			DELVAL	BENMANSOUR	DIALLO	HELLALA

Trois rapports seront à rendre sous forme électronique par mail :

**Rapport d'analyse** Il décrira en détail un cahier des charges du projet (ce que vous envisagez de faire) : il comprendra une description détaillée des futures fonctionnalités, des scénarios d'utilisation, une description de l'interface (éventuellement avec des « maquettes » d'écrans, un planning prévisionnel (réunion, livrables), la répartition des tâches entre les membres, ...)

**Rapport de conception** Il présentera les choix techniques de réalisation (comment vous allez le faire) : il comprendra une description des classes et de leurs interactions, l'explication des algorithmes importants, les changements apportés depuis le rapport d'analyse, ...

**Rapport final** Il reprendra les précédents rapports en mettant en évidence les changements et décrira l'implémentation du programme : il comprendra les deux précédents rapports mis à jour, une description de l'implémentation, un manuel de l'utilisateur (avec captures d'écran), le planning et la répartition des tâches, ...

## 4 Dates importantes

- \_ avril 2019 : lancement du projet
- \_ 10 mai 2019 : rendu du rapport d'analyse
- \_ 19 Mai 2019 : rendu du rapport de conception
- \_ 09 Juin 2019 : remise du rapport final
- \_ 19 Juin 2019 : soutenances
- \_ 23 juin 2019 : remise du code source du projet, des binaires et du support de la présentation