

# ***Projet Logiciel Transversal***

## ***Tactical Wars***

Moussa DIALLO , Maxime FARNOUD, Matthis HADDOUCHE & Timothé  
MUSETE LEKAN

## Sommaire

1 . Objectif.....	3
1.1 . Présentation générale.....	3
1.2 . Règles du jeu et Ressources.....	4
2 . Description et conception des états.....	8
2.1 . Description des états.....	8
2.1.1 . État des éléments fixes sur la carte.....	8
2.1.2 . État des éléments mobiles sur la carte.....	8
2.1.3 . État des éléments propres au joueur.....	9
2.1.4 . L'état général de la partie.....	9
2.2 . Conception Logiciel.....	9

# 1 . Objectif

## 1.1 . Présentation générale

Archétype : Advance Wars/Wargroove

- Jeu de stratégie militaire au tour par tour
- Gestion de ressources



Figure 1: Image du jeu « Advanced Wars »

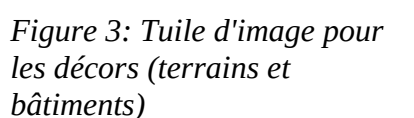


Figure 2: Image du jeu « Wargroove »

## 1.2 . Règles du jeu et Ressources

Règles :

- Condition de Victoire : Prendre le QG ennemi
- Déroulement d'une Partie :
  - Le royaume adverse a des vues sur un territoire stratégique, défendez-le !
  - Chaque joueur débute dans un coin de la map, avec seulement son Quartier Général (QG) et sans aucune unité. Ils disposent également d'un peu de mana (ressource fondamentale) pour commencer la partie.
  - Le QG sert à produire les unités ainsi que le nombre de membres (individus) la composant. Chaque unité possède un nombre maximal de membres pouvant former l'unité et lorsque ce maximum est atteint, l'unité obtient un bonus de statistiques (notamment pour l'attaque). Par conséquent, il est possible de fusionner 2 unités incomplètes.
  - Une fois une unité produite au QG le joueur peut la déplacer selon la statistique de mouvement de cette unité.
  - Au début de la partie, la carte est couverte d'un brouillard de guerre qui disparaît en fonction de la position de chaque unités et de leur statistiques de visions (chaque joueur a sa propre « vue » de la carte qui dépend de ses propres unités). Le brouillard cache complètement un case et ce qu'il y a dessus (terrain, unité, bâtiment) mais une fois disparu il ne revient pas.
  - Le joueur commence uniquement avec son QG comme bâtiment, et pourra en capturer d'autres durant la partie en y plaçant des unités. Lorsqu'une unité accomplit l'action de capture, elle ne peut pas attaquer. Un bâtiment se capture en 2 tours : 1er tour 50% de capture, 2ème tour 100% capture)
  - L'objectif est de prendre le QG adverse même si toutes les unités ennemies ne sont pas tombées.
- Il existe différents types de bâtiments :
  - Les mines de mana : permettent une récolte de mana abondante, ressource essentiel dans la partie qui permet de générer des unités.
  - Les villages : régénèrent la santé des unités se trouvant dessus et servent à l'effort de guerre en payant un impôt en mana à chaque tour.
  - Les camps d'entraînement : grâce à leurs prouesses, les officiers formateurs peuvent former des soldats performants ; ils permettent de générer des unités autre part qu'au QG et permettent donc de gagner du temps.



- Les chauve souris (unités d'éclaireur): très bonne vision mais très fragiles. Leur mobilité leur permet de s'échapper rapidement et de frapper sans être vu.



*Figure 5: Chauve-souris*

- Les magiciens : unité puissante à distance et moyenne au corps à corps. Assez fragile.



*Figure 6: Magicien*

- Les nains : de féroce combattant au corps à corps qui n'hésite pas à baisser leur garde pour attaquer l'adversaire.



*Figure 7: Nain*

- Les chevaliers : de véritables murs défensifs, ils sont prudents et se protègent derrière leurs boucliers et leur armure, cela les rends résistants mais diminue leur attaque et leur déplacement.



*Figure 8: Chevalier*

- Les golems, peu mobiles mais aussi puissants à distance qu'au corps à corps. Leur faible mobilité en font des cibles de choix pour les magiciens malgré leur formidable défense.



*Figure 9: Golem*

Le tableau suivant présente les valeurs des différentes statistiques des unités. Ces valeurs ne sont que des valeurs de référence selon une échelle arbitraire et servent essentiellement à établir les proportions entre les différentes unités. Évidemment, ces valeurs sont vouées à être modifiées à l'avenir après les premiers tests dans l'idée de mieux équilibrer le jeu :

Characters	PV	Portée d'Attaque	Dégâts	Portée de déplacement	Coût en mana
Magicien	30	100	70	60	70
Goblin	20	10	20	60	10
Golem	100	70	100	10	100
Chauve-souris	10	50	10	100	20
Nain	50	10	50	40	50
Chevalier	70	20	30	30	50

## 2 . Description et conception des états

### 2.1 . Description des états

Un état est essentiellement composé d'une carte en 2D quadrillée. Sur cette carte se trouvent des éléments fixes (les terrains et les bâtiments) et des éléments mobiles (les unités). Tous ces éléments ont une propriété commune qui est leurs coordonnées (x,y) qui correspond à leur position sur le quadrillage.

En plus de cela, un état doit prendre en compte les informations spécifiques aux joueurs (noms, ressources, vue de la carte, quantité de mana).

#### 2.1.1 . État des éléments fixes sur la carte

La carte est formée d'une grille d'éléments nommés « cases ». La taille de la grille est définie à l'avance et est directement lié au nombre de case (par exemple une grille 128x128 contient 16 384 cases). Chaque case de la grille correspond à une position sur la carte (avec les coordonnées x et y correspondantes).

Les éléments fixes sur la carte se partagent en deux catégories :

- **Les éléments pouvant être possédés par un joueur :**
  - Les bâtiments et leurs différents types (camp d'entraînement, mine de mana, QG et village) et leur statut de contrôle (« neutre » ou « possédé » par un des deux joueurs). À noter qu'il n'y a que le QG qui n'est jamais neutre et qui appartient toujours au même joueur (jusqu'à sa destruction).
- **Les éléments ne pouvant pas être possédés par un joueur :**
  - Les environnements et leurs différents types (montagne, forêt et plaine). Chaque case de la grille, à part celles où se trouvent des bâtiments, contient un type d'environnement.

Il faudra donc dans un état donné, avoir toutes les informations liées à une coordonnée du quadrillage : le type d'unité qu'il y a dessus (s'il y en a), le type d'environnement qui s'y trouve et le bâtiment situé à cet emplacement (s'il y en a un). Les informations liées aux bâtiment et aux unités sont également à connaître : qui les contrôle et de quel type il s'agit.

#### 2.1.2 . État des éléments mobiles sur la carte

Les unités sont les seules entités qui ont la capacité de se déplacer sur le quadrillage. Elles ont également la possibilité d'influer sur les bâtiments et les autres unités (ils peuvent prendre le contrôle des bâtiments et peuvent attaquer les unités ennemies). Nous devons donc connaître à



chaque état, leur position et la valeur de leurs différents attributs.

Exemple: Une unité de nains possède 3 combattants, la santé globale de l'unité est de 150 HP, sa portée de déplacement est de 30 et sa portée d'attaque est de 8 et la valeur de son attaque est de 40 (les valeurs ici sont purement arbitraire et servent juste d'exemple). La vision de l'unité doit également être prise en compte pour le joueur qui la contrôle afin de savoir quelle partie du brouillard de guerre il peut voir.

### 2.1.3 . État des éléments propres au joueur

Les éléments propres au joueur permettent de distinguer les deux adversaires et de leur associer des attributs liés au gameplay :

- La perception que le joueur aura sur la carte, c'est à dire si sa vision est masquée ou non par le brouillard.
- Les éléments identifiant les joueurs : les noms des joueurs ainsi que leur couleur. Ces deux éléments servent à distinguer les unités et les bâtiments contrôlés par les joueurs. Typiquement, pour un état donné nous pourrions avoir : Joueur 1 - équipe rouge contre Joueur 2 - équipe Bleu.
- Le quantité de mana que chacun des joueurs possèdent à un état donné. Cette quantité sera amenée à fortement évoluer lors d'une partie. Elle peut être nulle, mais cependant elle n'a pas de limite supérieure et ne peut être négative.
- L'état de l'armée du joueur doit être connu. Un joueur a des unités dans son armée, on veut avoir accès au nombre de ces unités, au nombre totale des membres de l'unité, les états de chacune des unités et toutes les infos correspondantes.

### 2.1.4 . L'état général de la partie

L'état général de la partie doit comporter :

- Le numéro du tour actuel de jeu. (identique pour les deux joueurs)
- Un tour est composé de deux phases : la phase de jeu du Joueur 1 et la phase de jeu du Joueur 2. L'état du jeu doit connaître la phase pour savoir qui doit jouer.
- Le statut de défaite ou victoire des joueurs une fois la partie terminée.

## 2.2 . Conception Logiciel

Le diagramme des classes pour les états est présenté en Figure 10, dont nous pouvons mettre en évidence les groupes de classes suivants :

➔ La classe **State** récapitule l'état générale du jeu :

- Qui est le joueur 1 ?
  - Qui est le joueur 2 ?
  - Quel est le nombre de tours ?
  - Le jeu a-t-il commencé et si oui qui joue ?
- ➔ On donne ensuite une description de la classe **Player** qui possède un état : joue, a gagné, a perdu. Il détient également une liste de ressources (un identifiant unique qui indique le type de ressource), une liste de bâtiments (liste de Building), une armée (Army), une couleur. Au niveau des méthodes on retrouve :
- Des add et remove pour ajouter des objets dans les différentes listes.
  - Des getter et setter pour accéder ou modifier les attributs de la classe.
- ➔ La classe **Ressource** est composée d'un unique attribut *amount* (entier) ainsi que d'un setter et un getter. De cette classe hérite **Mana**.
- ➔ La classe **Building** est composée des attributs suivants :
- D'une *position* de type **Position**
  - D'un identifiant de contrôle, permettant de savoir si le bâtiment est neutre ou s'il appartient à un Joueur.
  - D'un entier *amountMana* qui permet de mesurer la production en mana du **Building**
  - Un identifiant *BuildingID* qui permet d'identifier le bâtiment de manière unique et d'un *typeID* qui permet de distinguer le type de bâtiment.
  - Des setters et getters pour modifier et accéder aux différents attributs.

De cette classe hérite : **Headquarter, ManaMine, Town, TrainingCamp**.

- ➔ La classe **Unit** qui contient plusieurs attributs :
- *size* qui contient le nombre de combattants dans l'unité,
  - *positions* qui continent la position de l'unité,
  - *singleHpUnit* qui contient les points de santé d'un unique membre de l'unité,
  - *globalHp* qui contient les points de santé de l'unité complète,
  - *attackrange* qui permet d'obtenir la portée de l'attaque d'une unité,
  - *sightRange* qui permet d'obtenir la puissance de la vision d'un membre de l'unité,
  - *globaleDamage* qui permet de quantifier les dégâts totaux infligés par une unité,
  - *moveRange* qui permet de quantifier les déplacements d'une unité et finalement
  - *unitID* pour identifier l'unité de façon unique
  - *typeId* pour identifier la race de l'unité.

La classe contient également plusieurs méthodes :

- une méthode *init* pour initialiser l'unité
- une méthode *move* pour se déplacer
- une méthode *place* pour déplacer arbitrairement l'unité à une Position donnée
- une méthode *attack* pour attaquer
- Les différents getter et setter.

De cette classe héritent les classe suivante :

- **Bat**
  - **Gobelin**
  - **Dwarf**
  - **Knight**
  - **Wizard**
  - **Golem**
- ➔ La classe **Army** possède des listes stockant des différents types d'unité ainsi qu'un attribut *workForce* qui contient la population entière de l'armée.
- ➔ Une classe **Environnement** avec deux attributs : *typeID* qui identifie le type de biome, *allPositions* pour accéder à la liste de Positions que couvertes par le biome. La classe contient bien sûr les getter et setter de ces attributs. De cette classe héritent :
- **Mountain**
  - **Forest**
  - **Plain**
- ➔ Une classe **Position** qui contient deux attributs x et y (entiers), et plusieurs méthodes :
- une méthode *changePlace* pour se déplacer de façon arbitraire sur le quadrillage
  - une méthode *move* pour se déplacer
  - des getters et setters

