



Ecole Nationale
Supérieure
de l'Electronique
et de ses Applications

Projet Advance Wars

Pierre MARIET - Joanna XANH
3^{ème} année, Option Informatique et Systèmes

26 octobre 2017

Table des matières

1	Présentation générale	2
1.1	Archétype	2
1.2	Règles du jeu	2
1.2.1	Description des unités	2
1.2.2	Les généraux et leurs pouvoirs	2
1.3	Ressources	3
2	Description et conception des états	6
2.1	Description des états	6
2.1.1	État éléments fixes	6
2.1.2	État éléments mobiles	6
2.1.3	État général	7
2.2	Conception logiciel	7
3	Rendu : Stratégie et Conception	9
3.1	Stratégie de rendu d'un état	9
3.2	Conception logiciel	9

Chapitre 1

Présentation générale

1.1 Archétype

Le projet est de recréer le jeu Advance WarsTM développé par Intelligent Systems[®] pour la Game Boy AdvanceTM de Nintendo[®].

Le jeu est en tour par tour, sur une carte découpée en cases. Le but du jeu est de capturer le Quartier Général adverse. Il est aussi possible de gagner par forfait. Les cartes pourront être créées à partir d'un fichier texte.

1.2 Règles du jeu

1.2.1 Description des unités

Sur une carte composée de différents terrains et séparée en cases s'affrontent 2 armées. Les terrains augmentent ou non la défense de l'unité qui se trouve dessus. Ces armées sont composées de différentes unités qui ont chacune des spécificités qui varient :

- Les unités aériennes peuvent se déplacer n'importe où sur la carte. Les unités de transport aérien ont une capacité de 1 et ne peuvent débarquer en mer ou en montagne.
- Les unités navales ne peuvent se déplacer qu'en mer, pas dans les rivières. Les unités de transport naval peuvent débarquer uniquement sur une plage ou dans un port.
- Les unités d'infanterie et bazookas peuvent se déplacer sur tous les sols et traverser les rivières.
- Les unités terrestres motorisées ne peuvent franchir les montagnes.

1.2.2 Les généraux et leurs pouvoirs

Chaque armée est dirigée par un général possédant un pouvoir qui donne un avantage à son armée ou un inconvénient à l'ennemi. Ce pouvoir dure un tour (sauf dans pour le Blizzard) et se recharge sur plusieurs tours. Le général donne en plus un avantage moindre, ou non, mais en rapport avec son pouvoir et qui dure toute la partie.

Réparation Redonne 2 points de vie à toutes les unités alliées. Aucune autre spécificité en dehors du pouvoir. Les statistiques de ce général représentent les statistiques moyennes.

Blizzard Provoque une tempête de neige qui dure deux tours et qui diminue l’amplitude de déplacement des unités adverses. Aucune autre spécificité en dehors du pouvoir.

Marche forcée L’infanterie peut se déplacer d’une case en plus et les bazookas ont une force de frappe augmentée de 50%. Par défaut, l’infanterie et les bazookas ont un bonus de force de frappe de 20% et de 10% de défense contre les unités d’infanterie adverse. Une unité qui capture un bâtiment a vitesse de capture de 150% de ses points de vie. La puissance de frappe des autres unités de tir direct est en revanche moins élevée que la moyenne. Les unités de transports peuvent se déplacer d’une case en plus par rapport à la moyenne.

Puissance Max Les chars d’assaut voient leur puissance de frappe augmenter de 40%, leur défense augmente de 1 point et leur amplitude de déplacement augmente de 1 case. Par défaut, les chars d’assaut ont une puissance de frappe 50% plus élevée que la moyenne.

Embuscade Les unités de tir indirect voient leur portée augmenter de 1 et leur force de frappe augmenter de 50%. Par défaut, la portée des unités de tir indirect est augmentée de 1 case. Les unités de tir direct ont une force de frappe diminuée de 20%.

Frappe éclair Les unités aériennes peuvent se déplacer une nouvelle fois pendant le tour. Par défaut, la force de frappe des unités aériennes est 15% plus élevée que la moyenne Les forces navales ont une force de frappe réduite de 20% par rapport à la moyenne.



FIGURE 1.1 – Sprites des généraux

1.3 Ressources

Les ressources graphiques utilisées sont celles du jeu d’origine. Néanmoins, en considérant le temps de développement des autres paramètres du jeu, seuls la carte, les unités sur celle ci ainsi que les animations de capture seront affichées.

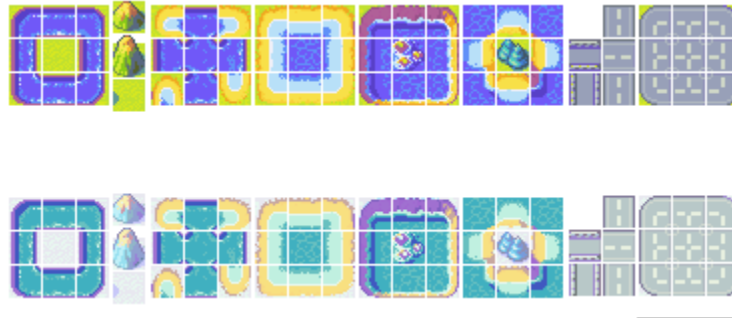


FIGURE 1.2 – Sprites pour les tuiles de la carte



FIGURE 1.3 – Sprites des différents batiments de la carte



FIGURE 1.4 – Sprites des unités ainsi que les informations de statut

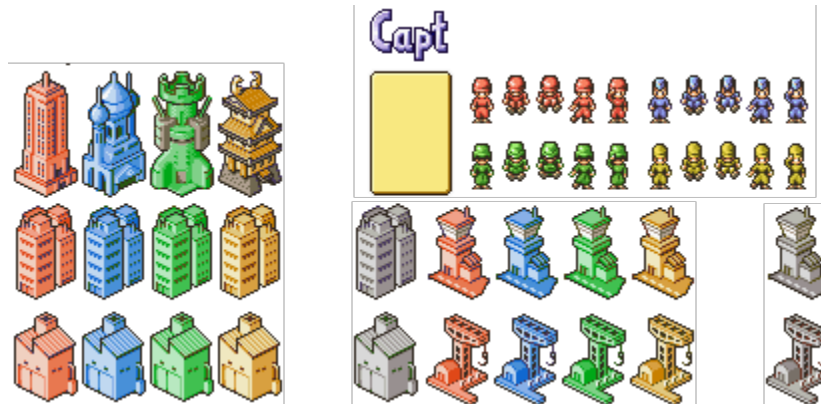


FIGURE 1.5 – Animations de capture



FIGURE 1.6 – Sprites des fenêtres d'attaque

Chapitre 2

Description et conception des états

2.1 Description des états

Un état du jeu est représenté par deux types d'éléments, ceux fixes qui sont les terrains et bâtiments et ceux mobiles qui sont les unités déployées. Ces éléments ont comme point commun des coordonnées et un identifiant pour savoir s'ils sont fixes ou statiques.

2.1.1 État éléments fixes

La carte est une grille d'éléments dont les dimensions sont fixées en début de partie. Toutes les cases ont des attributs communs comme le niveau de défense et le nombre de points de mouvements nécessaires pour franchir ce terrain. Il existe cependant deux types de cases :

Les cases terrain

Il en existe deux grand types : sol et eau. Pour les cases terrain sol, aucune unité navale ne peut se déplacer sur ces cases :

- Plain + Forest + Road + Bridge : Toutes les unités terrestres peuvent traverser. La particularité des cases Bridge est qu'aucune unité navale ne peut la traverser.
- Mountain + River : Seules les unités à pied peuvent traverser.
- Beach : Terrain particulier puisque les Lander peuvent se déplacer sur cette case pour charger et décharger. Toutes les unités terrestres peuvent traverser.

Pour les cases terrain eau, aucune unité terrestre ne peut se déplacer sur ces cases.

2.1.2 État éléments mobiles

Les éléments mobiles peuvent se déplacer sur la carte en fonction de leur condition et du nombre de points de mouvement qu'ils ont à disposition. Les unités peuvent passer par dessus d'autres unités alliées pour se rendre sur une autre case. Il ne peut y avoir qu'une unité par case. Deux types d'unités existent :

- **Les unités standard** : Elles peuvent attaquer, capturer, se déplacer selon leur capacité de mouvement et selon terrain sur lequel elles se déplacent.

- **Les unités de transport** : Ces unités peuvent transporter certains types d'unités

2.1.3 État général

Pour l'ensemble des éléments, nous avons :

- les jours : qui correspond aux nombres de tours de jeux.
-

2.2 Conception logiciel

Le diagramme des classes pour les états est présenté en Figure 2.1, les groupes de classes sont les suivants :

- **classe Element** : c'est la classe mère des éléments statiques et mobiles. C'est une classe abstraite puisqu'il ne peut y avoir d'instances de cette classe dans le jeu. Elle contient uniquement une variable permettant d'identifier si l'élément de la classe fille est statique ou non.
- **classe Commander** : cette classe est associée aux classes Building et Unit car elle peut agir sur leur état.
- **classe ElementTab** : c'est une classe qui regroupe plusieurs éléments dans un tableau 2D. Ce tableau est dynamique afin de pouvoir charger un monde à partir d'un fichier texte. Les coordonnées du tableau sont celles qui permettent d'identifier les positions des éléments contenus.
- **classe State** : cette classe regroupe deux tableaux de la classe ElementTab : un pour les éléments statiques, un pour les éléments mobiles. De plus, elle contient un compteur de jours (équivalent des tours) et un timer pour indiquer le temps restant pour jouer un tour.

Tableau des classes :

Classe	Rôle de la classe
Élément	Classe mère de tous les éléments d'états du jeu
Land	Créer les terrains du jeu
Unit	Créer les unités de l'armée
Building	Créer les bâtiments présents dans le niveau
Transport	Créer les unités de transport
Commander	Créer les officiers du jeu qui peuvent apporter des bonus à leur armée
State	Rassemble le tableau des éléments fixes et le tableau des éléments mobiles du jeu
ElementTab	Contient un tableau 2D d'éléments

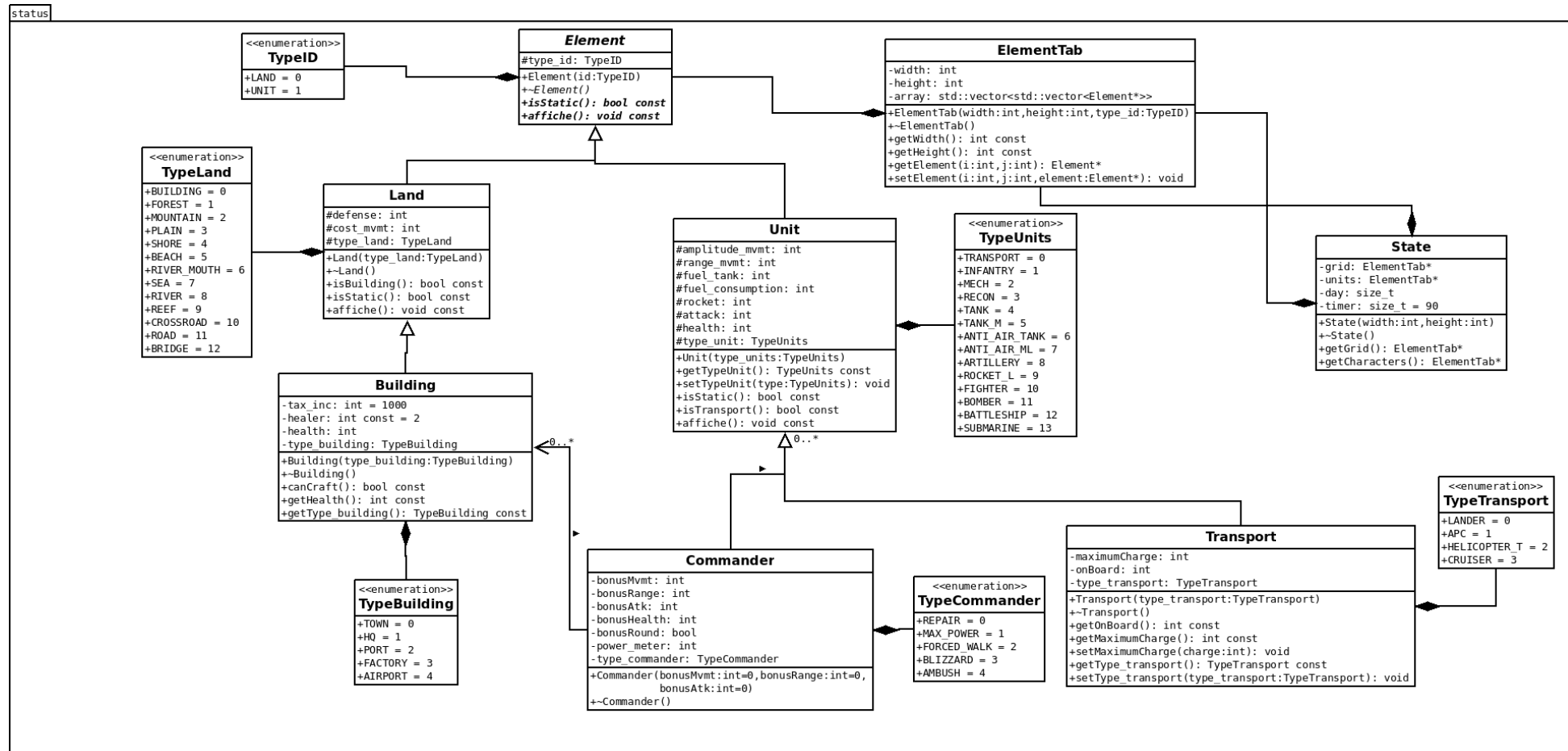


FIGURE 2.1 – Diagramme des classes d'état

Chapitre 3

Rendu : Stratégie et Conception

3.1 Stratégie de rendu d'un état

La stratégie que nous avons établi pour le rendu d'un état est une stratégie de bas niveau. Nos scènes sont coupées en plans : un plan pour le niveau qui contient différents paysages (plaine, montagne, forêt, etc.), un plan d'unités mobiles (infanterie, tank, navire, etc.), et un plan d'informations d'état du jeu (vie, jauge de pouvoir, etc.). Chaque plan possède différentes informations : une unique texture contenant les tuiles, une unique matrice avec la position des éléments, un tableau de nos plans, le nom du fichier texte à partir duquel nous générons notre état, un tableau des différentes textures, les dimensions de la fenêtre ainsi que les dimensions de tuiles.

L'organisation du rendu se fait à partir de l'état et de la détection d'un changement. Si le changement dans l'état donne lieu à un changement permanent dans le rendu, on met à jour le morceau de la matrice du plan correspondant. Pour les changements non permanents, comme les éléments mobiles, nous tiendrons à jour une liste d'éléments visuels à mettre à jour (\Rightarrow modification de la matrice du plan) automatiquement à chaque rendu d'une nouvelle frame.

3.2 Conception logiciel

Le diagramme du rendu est présent en Figure 3.1.

Plans : Le principal objectif des instances de Layer est de donner les informations basiques pour former les éléments bas-niveau à transmettre à la carte graphique ; ces informations sont données à une instance de Surface, et la définition des tuiles est contenue dans une instance de TileSet.

La méthode `setSurface()` fabrique une nouvelle surface à partir de notre fichier texte, grâce aux informations contenues dans ce fichier on peut lui demander de charger la texture, et d'initialiser la liste des sprites. Pour afficher le niveau, elle demande les dimensions de la fenêtre avec `initQuads()`. Puis, pour chaque cellule du niveau, elle se leur position avec `setQuadPosition()` et leur tuile avec `setQuadTextureCoordinates()`.

Surfaces : Chaque surface contient une texture du plan et une liste de paires de quadruplets de vecteurs 2D. Les éléments `texCoords` de chaque quadruplet contiennent les coordonnées des quatre coins de la tuile à sélectionner dans la texture. Les éléments position de chaque

quadruplet contient les coordonnées des quatre coins du carré où doit être dessiné la tuile à l'écran.

Tuiles : La classe `TileSet` contient un tableau de toutes les tuiles d'un même plan. Elle peut grâce à la méthode `getTile()`, récupérer la tuile correspondant à un indice de son tableau, et à partir de cette tuile on peut savoir ses dimensions ainsi que l'emplacement de sa texture dans nos dossiers. Pour obtenir ces informations, un client dispose des méthodes `getWidth()`, `getHeight()` et `getAdressFile()`.

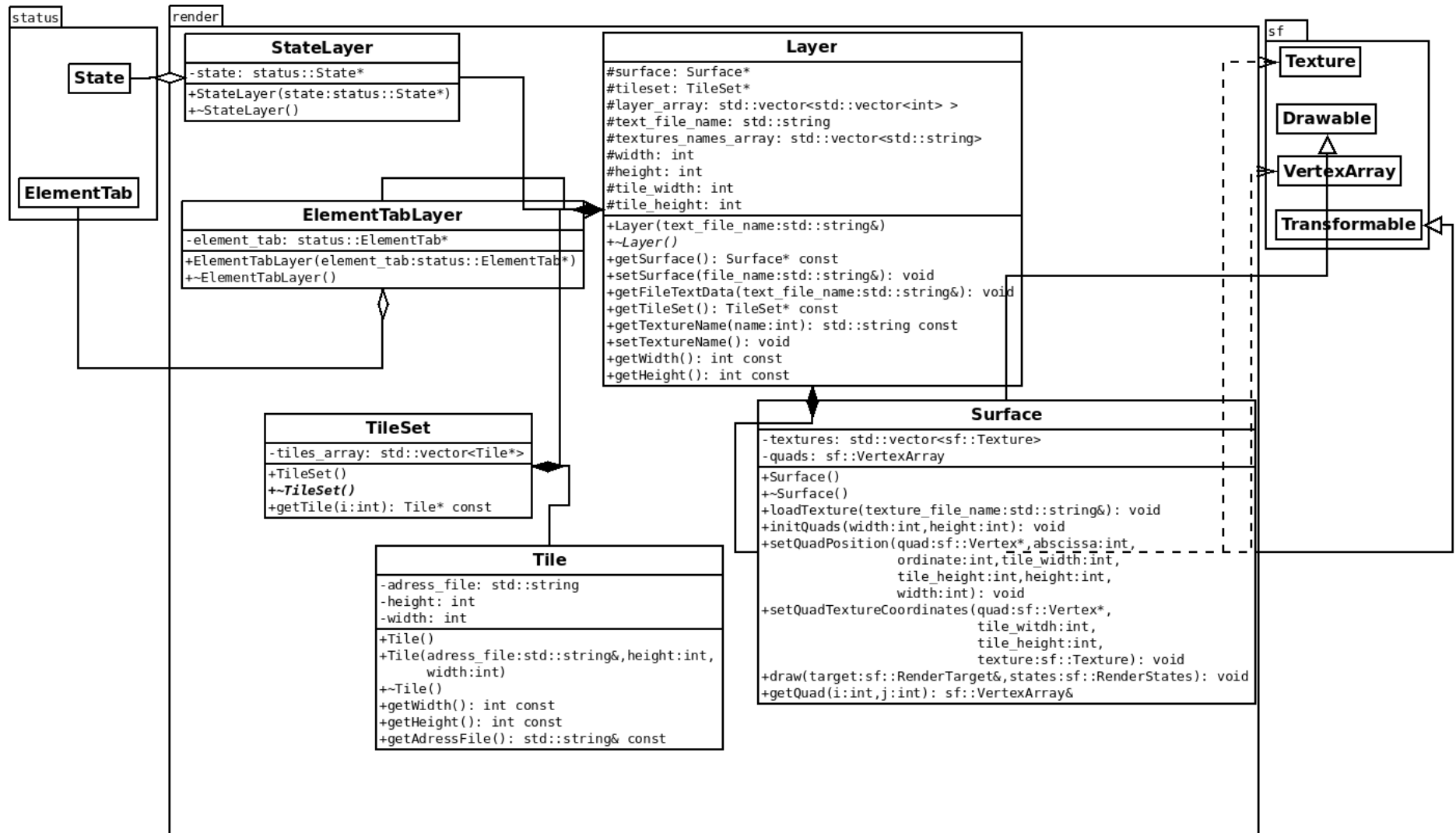


FIGURE 3.1 – Diagramme du rendu