

Spécifications techniques détaillées de la classe *Ile*

Edition A – Rev. 0

Mission

Le simulateur **Trafalgar** gère en temps réel un ensemble prédéfini de navires qui se déplacent dans un plan "illimité" contenant un ensemble prédéfini d'îles, réparties en plusieurs **zones géographiques prédéfinies**.

Les positions absolues et les caractéristiques des différentes îles sont issues de cartes marines, communiquées à toutes les flottes engagées dans la confrontation. Ces cartes seront fournies par zone.

Chaque île appartient à une seule zone et est modélisée par un triangle de taille prédéfinie, dont les coordonnées sont relatives à un point origine unique de la zone, qui est le coin nord ouest de la carte marine correspondante.

Un des sommets du triangle peut être l'emplacement de l'unique port de l'île. Certaines îles n'ont pas de port.

Au démarrage du jeu, des troupes d'une ou plusieurs flottes pourront être pré positionnées sur certaines îles, au gré du coordinateur du jeu (processus "**arbitre**"). Le coordinateur ne pourra plus intervenir ensuite sur les effectifs et les positions des troupes dans les îles pendant la phase de confrontation.

Pendant la phase de confrontation, les navires pourront débarquer sur les îles des troupes qu'ils ont à bord par des opérations de débarquement terrestre, sur ordre des joueurs qui les pilotent. Chaque flotte pourra ainsi à son gré stationner des troupes en un point unique ou en plusieurs points de chaque île, en plus des troupes déjà pré positionnées au démarrage du jeu.

Le débarquement de troupes sur une île occasionne des pertes pour l'assaillant et pour les éventuels défenseurs, qui sont fonction de :

- la distance du navire au point de débarquement
- des conditions météo dans la zone de débarquement
- l'effectif des troupes adverses déjà présentes au voisinage du point de débarquement

Un débarquement **dans le port** d'une île diminuera notablement les pertes subies par l'assaillant vis à vis de deux premiers critères : distance et météo.

Distance de débarquement des troupes

Pour réussir un débarquement, la distance du navire au rivage doit être comprise entre :

- un minimum en deça duquel le bateau s'échoue (ce qui n'empêche toutefois pas le débarquement)
- un maximum au delà duquel toutes les troupes à débarquer périssent

La distance d'un navire au point de débarquement est calculée par la projection orthogonale de sa position courante sur le rivage adjacent. Les bornes de l'intervalle de réussite seront propres à chacun des îles.

Conditions météo

Les conditions météo aux abords des rivages d'une île sont locales et peuvent différer sensiblement de celles plus au large. Quand les conditions météo autour d'un navire sont différentes de celles autour d'une île de débarquement, ce sont ces dernières qui prévalent pour le calcul des pertes.

Effectifs des troupes adverses

La présence de troupes adverses autour du point de débarquement sera appréciée par rapport à un rayon d'action qui sera fonction de leur effectif courant. Les troupes de la flotte de débarquement déjà présentes près du point de débarquement limiteront l'action des troupes adverses, dans la proportion de leurs effectifs déjà stationnés.

Opérations d'embarquement de troupes

Pendant la phase de simulation et de façon analogue aux opérations précédentes, les navires pourront embarquer des troupes à leur bord depuis le rivage d'une île proche et sur ordre des joueurs qui les pilotent. Les conditions de succès et les calculs de perte seront identiques à ceux des opérations de débarquement.

Les navires ne pourront embarquer que des troupes de leur flotte d'appartenance.

I- Exigences fonctionnelles

Les propriétés d'une île sont :

1. l'identification de l'île	Objet	<i>m_nom</i>	<i>string</i>
2. la description topologique de l'île	Objet	<i>m_lieux</i>	<i>Triangle</i>
3. l'état de la météo autour de l'île	int	<i>m_etatMeteo</i>	<i>etatMeteo</i>
4. les effectifs des troupes de chaque flotte	Objet	<i>m_effectifs</i>	<i>map <string, list <int>></i>
5. les positions des troupes de chaque flotte	Object	<i>m_positions</i>	<i>map <string, list <Point>></i>

Outre le constructeur par défaut, la classe possèdera un constructeur normal qui recevra en paramètre le nom de l'île et la description des lieux (instance de la classe *Triangle*).

Aucun champ n'est modifiable globalement après création.

La classe fournira les accesseurs de consultation *nom*, *lieux*, *meteo* et *effectifs*.

Un accesseur de modification *ajouterTroupes* permettra de mettre à jour l'effectif courant des troupes d'une flotte cible en un point d'une île support, en augmentant ses effectifs locaux. La méthode sera invoquée après chaque débarquement de nouvelles troupes. Elle n'intégrera pas les calculs des pertes subies par les belligérants au point de débarquement.

Un accesseur de modification *retirerTroupes* permettra de mettre à jour l'effectif courant des troupes d'une flotte cible en un point d'une île support, en diminuant ses effectifs locaux. La méthode sera invoquée notamment après chaque embarquement de troupes. Elle n'intègre pas les calculs des pertes subies par les belligérants au point d'embarquement.

La méthode *retirerTroupes* sera également invoquée pour appliquer tous les calculs de pertes.

Service calculerPertes

Le service *calculerPertes* calculera les pertes subies par une flotte cible désignée par paramètre, en un point de transfert de troupes d'un des belligérants (débarquement ou d'embarquement). Le point cible sera fourni en paramètre.

Service débarquement

Le service *debarquement* exécutera les consignes de débarquement fournies en paramètre, en y intégrant les calculs des pertes subies par tous les belligérants présents au voisinage du point cible.

Service embarquement

Le service *embarquement* exécutera les consignes d'embarquement fournies en paramètre, en y intégrant les calculs des pertes subies par tous les belligérants présents au voisinage du point cible.

II- Contraintes de développement

La classe devra satisfaire aux spécifications techniques générales du package ***_Trafalgar*** et respecter toutes les contraintes de conception, de codage et de tests imposées pour les TP du module APO C++.