Les exercices des feuilles de TD précédentes ont permis de modéliser les propriétés des îles ainsi que les déplacements et les manoeuvres des navires. Les solutions apportées sont encore incomplètes.

L'objectif des exercices de cette feuille est de mettre en place la classe principale ***Trafalgar*** du simulateur temps réel de même nom, après une mise à niveau des classes ***_Trafalgar*** et ***Navire***.

Les exercices de la feuille de TD suivante permettront de finaliser la classe ***Ile***, notamment pour les embarquements et débarquements de troupes.

Les exercices de l'ultime feuille de TD permettront de modéliser les équipements radars portés par les navires, uniques moyens d'observation de l'environnement de la confrontation.

Exercice 1. Classe *_Trafalgar* – V 1.1.0

Installer avec succès la version **V 1.1.0** de la classe ***_Trafalgar***.

Q1 / Identifier toutes les évolutions fonctionnelles de cette classe
Q2 / Justifier l'intérêt technique de l'instruction C **enum** et ses limites ?
Q3 / Justifier le rôle fonctionnel de l'énumération ***TypesNavires***
Q4 / Justifier le rôle fonctionnel de l'énumération ***EtatsMeteo***
Q5 / Fournir plusieurs exemples d'autres types de littéraux qui restent à définir.

Exercice 2. Classe *Navire* – V 1.2.0 / Service *manoeuvrer*

Installer avec succès les codes sources du service ***manoeuvrer*** de la classe ***Navire***, solution de la feuille n°19.

Exercice 3. Classe *Navire* – V 1.3.0 / Ajout de nouvelles propriétés

Installer avec succès la version **V 1.3.0** de la classe ***Navire*** puis analyser toutes les évolutions fonctionnelles de la nouvelle version.

Q1 / Quel est l'intérêt fonctionnel pour la simulation du nouveau constructeur normal ?
Q2 / Quel est l'intérêt fonctionnel de connaître le type du navire ?
Q3 / Quel est l'intérêt fonctionnel de connaître le poids du navire ?
Q4 / Justifier l'intérêt technique de modifier le type des propriétés "nom" et "drapeau"
Q5 / Quel est le rôle de la méthode ***c_str***() de la classe ***string*** ?

Exercice 4. Classe *Navire* – V 1.3.0 / Etat d'avancement des services

Q1 / Un constructeur d'une classe C++ peut il invoquer un autre constructeur de la même classe ?
Q2 / Comment pourrait on améliorer le contrôle de validité du paramètre de la méthode ***deplacer*** ?
Q3 / Quelle contrainte de compilation imposerait cette amélioration ?
Q4 / Quelles évolutions fonctionnelles reste à apporter à la méthode ***deplacer*** ?
Q5 / Quelles évolutions fonctionnelles reste à apporter à la méthode ***manoeuvrer*** ?

Installer avec succès la version **V 1.3.0** de la classe ***Navire***.

Exercice 5. Classe *Trafalgar* – V 1.0.0 / Propriétés et comportements du simulateur

Installer avec succès la version **V 1.0.0** de la classe principale ***Trafalgar*** puis contrôler que le fichier "header" est conforme aux spécifications détaillées de la classe fournies en annexe. Cette classe exploite le mécanisme C++ RTTI.

Q1 / Justifier le choix d'une classe abstraite.
Q2 / Justifier le choix du type des attributs, en regard des comportements spécifiés pour le simulateur.

Q3 / Quelle est la justification de l'absence des cartes marines dans les propriétés de la classe ***Trafalgar*** ?
Q4 / Quelle classe de la suite logicielle **Trafalgar** "construira" les îles à partir des cartes marines ?
Q5 / Pour quelle raison aucune propriété de la classe ***Trafalgar*** ne modélise les flottes ?

Exercice 6. Classe *Trafalgar* – V 1.0.0 / Accesseur de modification *placer*

Exploiter le squelette fourni en annexe pour développer l'accesseur ***placer*** et exécuter avec succès le module de tests unitaires également fourni en annexe.

Exercice 7. Classe *Trafalgar* – V 1.0.0 / Service *start*

Exploiter le squelette fourni en annexe pour développer le service ***start*** et exécuter avec succès le module de tests unitaires également fourni en annexe.

Exercice 8. Classe *Trafalgar* – V 1.0.0 / Service *deplacer*

Exploiter le squelette fourni en annexe pour développer le service ***deplacer***.

En tests unitaires, le thread H1 est simulé par une succession d'appels de la méthode ***deplacer*** (boucle). Exécuter avec succès le module de tests unitaires également fourni en annexe.

Exercice 9. Classe *Trafalgar* – V 1.0.0 / Service *manoeuvrer*

Exploiter le squelette fourni en annexe pour développer le service ***manoeuvrer*** et exécuter avec succès le module de tests unitaires également fourni en annexe.