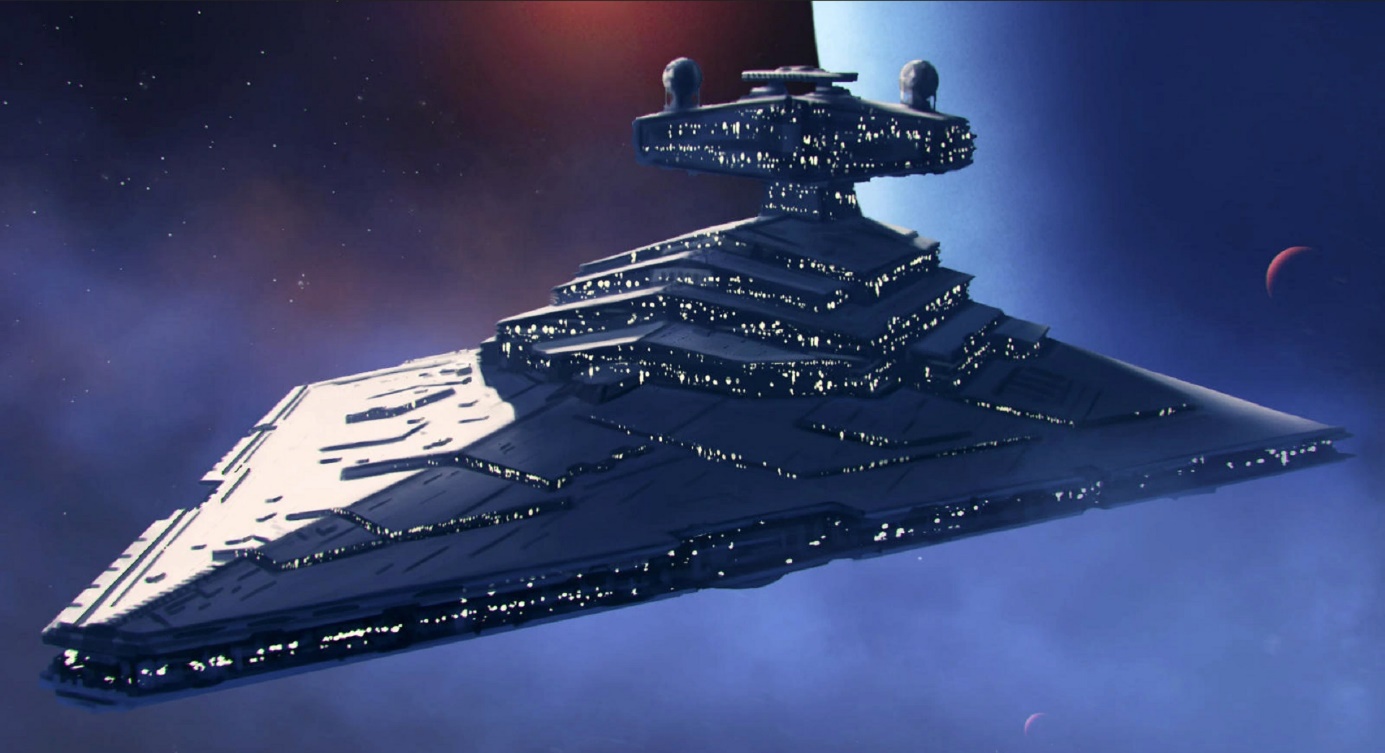
**Squadron – Labo 2**



*Durée du travail :* **Du 21.03.2024**

**au 17.04.2024**

*Auteurs :* **Demont Killian   
Graf Calvin**

*Enseignant :* **Krähenbühl Grégoire**

*Assistant :* **Decorvet Grégoire**

*Domaine d’application :* **C++**

*Lieu de travail :* **HEIG-VD | Yverdon-les-Bains**

Table des matières

[1. Introduction 3](#_Toc164272410)

[2. Développement 4](#_Toc164272411)

[2.1 Instruction de compilation 4](#_Toc164272412)

[2.2 Choix d’implémentation 4](#_Toc164272413)

[2.3 Protocole de tests 5](#_Toc164272414)

[2.3.1 Squadron 5](#_Toc164272415)

[2.3.2 Ship 6](#_Toc164272416)

[3. Conclusion 7](#_Toc164272417)

[4. Diagramme de classe 8](#_Toc164272418)

## Introduction

Ce laboratoire a pour objectif de concevoir une escadrille qui est composée de différents types de vaisseaux spatiaux représentée chacun par une classe. Nous devrons pouvoir calculer la quantité de carburant nécessaire pour parcourir une distance en fonction de la vitesse désirée et en prenant compte le type de vaisseaux qui compose l’escadrille.

De plus, nous devons gérer la règle des cinq pour la classe Squadron, l’ajout et suppression de vaisseau de différentes manières. Soit en retournant une nouvelle instance soit en modifiant l’instance qui invoque la méthode. La gestion du leader de l’escadrille, l’affichage de celle-ci et pour finir l’accès à un vaisseau de l’escadrille en fonction de son index.

Nous avons plusieurs types de vaisseaux séparés en deux catégories distinctes, les vaisseaux cargo qui peuvent avoir un chargement et les autres.

Enfin, nous devrons ajouter un programme de test cohérent et complet qui vérifie qu’il ne reste plus aucune erreur dans notre programme.

## Développement

* 1. Instruction de compilation

Nom du compilateur : Mingw-w64 gcc

Version du compilateur : 13.2.0

* 1. Choix d’implémentation

Concernant la classe Ship, comme l’implémentation de la règle des cinq n’était pas demandée contrairement à Squadron, nous avons décidé d’interdire leur utilisation à l’aide du mot-clé « delete » pour éviter par exemple la création d’un vaisseau par copie qui aurait la même ID que le vaisseau originel. Nous avons uniquement laissé le destructeur en « default » car la consigne du laboratoire le demandait.

Pour les vaisseaux, il est possible de créer un vaisseau sans nom, c’est pourquoi au lieu de créer un deuxième constructeur, nous avons mis une valeur par défaut à vide au nom. Cela nous permet de réutiliser notre code.

Nous avons ajouté un fichier « Constants » qui regroupe les unités utilisées dans notre application afin de pouvoir changer facilement. Nous avons décidé d’ajouter l’unité de distance bien que nous ne l’utilisons pas afin de savoir quelle unité nous utilisons pour notre calcul. Comme ça, si nous souhaitons l’afficher dans le futur, nous utiliserons la bonne unité.

* 1. Protocole de tests

1. Squadron

|  |  |
| --- | --- |
| **Test** | **Résultat** |
| Créer une escadrille avec les différents constructeurs | OK |
| Ajouter un vaisseau avec la méthode d’ajout de vaisseau | OK |
| Ajouter un vaisseau avec += | OK |
| Retourner une nouvelle escadrille contenant le nouveau vaisseau avec : escadrille + vaisseau | OK |
| Retourner une nouvelle escadrille contenant le nouveau vaisseau avec la méthode d’ajout de vaisseau par copie | OK |
| Ajouter un vaisseau déjà dans l’escadrille ne fait rien | OK |
| Déplacer une escadrille avec le constructeur par déplacement | OK |
| Retirer un vaisseau avec la méthode pour retirer un vaisseau (peu importe son emplacement dans l’escadrille) | OK |
| Retirer un vaisseau avec -= (peu importe son emplacement dans l’escadrille) | OK |
| Retourner une nouvelle escadrille sans le vaisseau avec : escadrille - vaisseau (peu importe son emplacement dans l’escadrille) | OK |
| Retourner une nouvelle escadrille sans le vaisseau avec la méthode pour retirer un vaisseau par copie | OK |
| Retirer un vaisseau qui n’est pas dans l’escadrille ne fait rien | OK |
| Accéder au 1er vaisseau | OK |
| Accéder à un vaisseau au milieu de l’escadrille | OK |
| Accéder au dernier vaisseau | OK |
| Modifier un vaisseau directement via la méthode pour récupérer un vaisseau | OK |
| Accéder à un index invalide retourne une erreur | OK |
| Attribuer un leader à une escadrille (pas de doublon avec les membres) | OK |
| Retirer le leader d’une escadrille (ne le retire pas de l’escadrille) | OK |
| Remplacer le leader d’une escadrille (l’ancien leader devient membre) | OK |
| Attribuer comme leader un vaisseau qui est déjà leader ne fait rien | OK |
| Retirer le leader d’une escadrille qui n’a pas de leader ne fait rien | OK |
| Copier une escadrille à l’aide du constructeur par copie fonctionne | OK |
| Modifier le leader, les membres ou le nom de la copie ne modifie pas l’originale et vice-versa | OK |
| Le calcul de la consommation totale de l’escadrille en fonction de sa distance et des vitesses des différents type de vaisseau est correcte | OK |
| Le calcul de la consommation totale de l’escadrille retourne une erreur si la vitesse fournie et trop haute pour l’un des vaisseaux de l’escadrille | OK |
| Tester que l’affichage est correct | OK |
| Vérifier que la précision des digits est correcte | OK |

1. Ship

|  |  |
| --- | --- |
| **Test** | **Résultat** |
| Créer un vaisseau de tous les types | OK |
| Créer un vaisseau sans nom est possible | OK |
| Créer ou modifier un vaisseau avec une cargaison invalide retourne une erreur (Ex : -12) | OK |
| Créer ou modifier un vaisseau avec cargaison plus élevée que la limite du vaisseau retourne une erreur | OK |
| Le calcul de la consommation d’un vaisseau en fonction de la distance et sa vitesse (pour tous les types de vaisseaux) est correcte | OK |
| Le calcul de la consommation d’un vaisseau avec une vitesse qui dépasse sa vitesse maximum retourne une erreur | OK |
| Tester que l’affichage est correct | OK |
| Vérifier que la précision des digits est correcte | OK |

Constante : Nous avons refait tous les tests listés ci-dessus avec des Ship et Squadron constant lorsque cela est possible. C’est-à-dire pour chaque méthode constante.

## Conclusion

La classe Squadron permet l’ajout et la suppression de membres ou de leader, le calcul de la consommation de l’escadrille et la possibilité de définir un nom. Les différents types de vaisseaux ont tous été implémenté avec chacun leur particularité et la possibilité de calculer la consommation pour une distance et vitesse donné pour chacun.

La conception des classes opérations a été implémenté en factorisant le plus possible afin de faciliter l’implémentation d’éventuelles nouveaux types de vaisseau dans le futur. La règle des 5 a été mis en place ainsi que l’opérateur d’écriture dans le flux pour Squadron et Ship. Tous les points du cahier des charges ont été remplis et testés avec succès.

## Diagramme de classe

