

# Projet C++ - Tower Sim

Lorris Creantor

17 avril 2022

## Choix d'implémentation

Au cours du projet, le seul moment pour lequel nous avons réellement eu à effectuer des choix d'implémentations a pour réaliser la suppression des avions. Pour cette suppression j'ai d'abord fait un premier choix que j'ai par la suite été amené à revoir.

Lors de ma première approche du problème (dans la question 4 de la TASK 0), a été d'ajouter dans la classe `DynamicObject` une fonction `bool must_remove() const;` servant à déterminer si un objet être supprimé de la liste. Une autre solution aurait été de faire en sorte que la fonction `move` retourne un booléen, mais j'ai préféré séparer les deux responsabilités par souci de clarté et également car, la fonction `bool must_remove() const;` ayant une implémentation par défaut retournant `false` faisant que je n'avais à l'override que dans la classe `Aircraft`.

Par la suite (au cours de la TASK 2), lorsque j'ai été amené à introduire la classe `AircraftManager`, j'ai pu reprendre ma solution précédente. En effet, cette nouvelle classe ayant pour rôle de stocker directement les avions, j'ai pu déplacer la méthode `bool must_remove() const` directement dans la classe `Aircraft` car, le seul endroit où celle-ci était requise était lors du parcours de la liste d'`Aircraft`.

## Problèmes rencontrés

Les principaux problèmes rencontrés lors de la réalisation du projet sont liés au fait que la programmation en langage C++ diffère grandement de ce à quoi j'ai pu être habitué en utilisant d'autres langages. Je peux citer notamment le parcours des collections à l'aide des bornes `begin` et `end` ou encore la façon assez singulière de supprimer des éléments d'une collection en les déplaçant à l'aide de `remove_if`.

J'ai également rencontré des difficultés pour la TASK 4 au moment de générer la classe `Point`. J'ai d'abord tenté d'utiliser la fonction `enable_if` et la SFINAE pour bloquer certains constructeurs avant de comprendre qu'il fallait en fait supprimer les deux constructeurs spécifiques et ne garder que le générique.

## Appréciation du projet

Dans l'ensemble, je ne garde pas une très bonne opinion de ce projet. Ce ressenti est lié à plusieurs choses.

Tout d'abord, et c'est un des points principaux même s'il n'est pas lié au projet en lui-même, je n'apprécie pas du tout la programmation en C++. Je trouve que le langage même s'il est connu pour les performances qu'il offre, rend la plupart des tâches basiques (tris de collection, généricité, déclaration de champ possible de 4 manière différente...), trop compliquée, et ce, sans réel avantage. Un autre problème selon moi est la syntaxe ambiguë de C++. Les mêmes symboles pouvant représenter des choses différentes selon le contexte comme `&&`.

Ensuite, je n'ai pas apprécié la façon dont le sujet était agencé. On nous imposait trop souvent des directions à prendre, pour par la suite nous dire que ce n'était pas la bonne solution, ce qui était très frustrant. Peut-être aurait-il mieux valu de donner moins d'indication pour nous forcer à réfléchir par nous même, et, si jamais la solution semblait être dure à trouver, glisser quelques indices.

Pour terminer cette section sur une note positive, j'ai trouvé intéressant l'exercice consistant à reprendre du code étranger et à lui ajouter des fonctionnalités.

## Connaissances acquises

Je n'ai globalement pas acquis de connaissance notable au cours de ce projet, hormis ma familiarisation avec la librairie standard du langage C++. En effet, je n'ai pas réellement été amené à réfléchir à des solutions, le sujet nous laissant dans les faits peu de liberté.