LAGNEAU Gaétan

1) Réponse aux questions

Toutes les réponses sont présentes directement sur les `TASK_N.md`.

2) Choix d'architecture

Pour signifier qu'un avion doit être détruit, je n'ai pas modifié la signature du `move()` des `DynamicObject` pour y renvoyer un booléen. En effet, cela nous aurait forcer à modifier chaque objet de ce type, pour lesquelles on aurait toujours renvoyer `false`. De plus, il ne me paraît pas évident, d'un point de vue sémentique, que move renvoie un booléen, et en particulier un booléen qui indique si l'objet doit être détruit ou pas. J'ai donc simplement rajouté un champs booléen à mon avion, car c'est le seul `DynamicObject` qui en a besoin.

En ce qui concerne les assertions, et en particulier celles qui s'assurent qu'une fonction n'est appelée qu'une fois, j'ai utilisé ceci :

```
void TowerSimulation::init_aircraft_manager()
{
    static bool is_init = false;
    assert(!is_init);
    ...
    is_init = true;
}
```

L'avantage est qu'on encombre pas les champs de la classe, en revanche on part du principe que la classe n'est instanciée qu'une seule fois.

Pour le reste, j'ai suivi les instructions en terme d'implémentation.

3) Les difficultés

J'ai passé un certain temps sur le destructeur de la classe `Aircraft`, lorsque l'on nous demande de supprimer aussi l'objet de la liste des objets affichables. Comme je n'avais encore jamais utilisé un langage à destructeurs, cette solution ne m'est pas venu à l'esprit tout de suite.

J'ai aussi eu du mal avec le ravitaillement, car j'avais certains bugs avec des avions qui restent bloqués aux terminaux, etc. Au final c'était juste un travail de débuggage.

Naturellement j'ai trouvé le langage plus difficile que les autres, en particulier en ce qui concerne les erreurs du compilateur qui sont parfois peu clémentes.

4) Ce que j'ai aimé et/ou détesté dans ce projet

Le sujet en lui même (travailler sur un aéroport avec des avions) est sympathique comparé à d'autres projets. Sinon, on est bien guidé, peut-être un peu trop, ce qui laisse peu de place à de réels choix d'implémentation.

5) Ce que j'en ai appris

Pas une connaissance en particulier, mais plus une consolidation par application de ce qu'on a vu en cours. Néanmoins comme le cours est vaste, et qu'on ne passe qu'une ou deux questions dans le projet sur chaque concept, il me faudra plus de pratique pour avoir réellement une base solide en C++.