Compte rendu

Travaux pratiques de C++

TP 3

Sommaire

1. Présentation du TP

Nous avons décidé de réaliser le TP3 à la suite du TP1 car nous trouvions cela plus intéressant au vu de la différence des questions entre le TP1 et le TP2. Cela nous semblait approprié pour augmenter nos connaissances ainsi que notre savoir-faire. De plus, le niveau de difficulté étant plus élevé, cela représente un plus grand défi et nous permet de tester nos connaissances ainsi que nos capacités sur un projet plus complexe.

1.1 Répartition et organisation du travail

Pour ce TP, nous avons tout d'abord décidé de réaliser la base de toutes les classes demandées afin d'avoir une base de travail propre. Nous nous sommes donc partagé les tâches en distribuant à parts égales le travail. L'un se focalisant sur les classes Produit, Client et l'autre sur Magasin et Commande.

1.2 Outils utilisés

Nous avons utilisé l'outils de versioning GitHub afin de faciliter le travail en binôme. Cet outils nous a été très utile pour récupérer le code écrit par l'autre, notamment pour la création des classes. Comme éditeur de texte, nous avons tous les deux utilisé VS Code. Cet éditeur a été pratique car il nous a permis d'exécuter notre code directement depuis un terminal dans l'éditeur. Il permet aussi l'ajout d'addons afin d'optimiser notre interface graphique et de mieux visualiser le code en cours de rédaction.

Nous nous sommes aussi servi de Copilot lors de la réalisation de tache répétitive comme les fonction getter.

2. Présentation du code

2.1 Organisation des classes

Organisation des classes :

- Classe Client
  - Gère les informations sur le client
  - Gère le panier des clients et modifie le stock en direct en fonction de ce que le client a dans son panier (comme un magasin physique, ce choix a été fait après un long débat)

Lors de la réalisation de cette classe, nous avons décidé de créé un affichage du panier en lien avec les autres classes qui affichent chaque produit avec la quantité voulue. Cependant, après 1h de débogage nous n'avons pas réussi à comprendre pourquoi l'exécution nous renvoie une erreur d'allocation. Nous avons donc décidé d'afficher x fois les produits dans le panier et de mettre en commentaire la fonction.

De plus, lors de la réalisation de la méthode permettant de retirer un produit du panier, nous avons rencontré plusieurs difficultés.

Dans la boucle for ligne 48 ($client.cpp$), nous parcourons un vecteur. Cependant, nous changions la taille de ce vecteur dans la suite du code car nous ajoutions/retirions des objets du vecteurs. Cela a donc entrainé une multitude d'erreur de type bad_{alloc}. Après un long temps de recherche, nous avons compris que la fonction n'éditait pas réellement le panier mais une copie de celui-ci. Nous avons donc ajouté une copie manuelle du panier d'origine afin de ne pas toucher à l'original.

- Classe Produit
  - Gère le stock des produits du magasin
  - Gère le prix des produits

Dans cette classe, nous avons décidé de faire une fonction retour ainsi qu'une surcharge afin de récupérer les diverses variables des produits. Cela permet d'avoir soit un affichage pour le magasin avec la quantité disponible soit un

affichage dans le panier sans cette quantité qui n'est pas intéressante à ce moment.

```
nom : Produit 1, description : Description 1, prix : 10.5 euro
nom : Produit 1, description : Description 1, prix 10.5 euro, quantite 85
```

Le 1er screen a été affiché via une surcharge de la classe produit et le 2ème avec une méthode dans la classe produit.

De plus, nous avons mis en place une sécurité afin qu'un client ne puisse pas prendre plus de produit dans sa commande que le stock disponible.

- Class Commande
  - Gère les commandes effectuées par les clients
  - Statut de l'état de leur commande
  - Vide le panier en cas de commande effectuée

Lors de la réalisation de la surcharge pour afficher le contenu des commandes, nous avons appris que nous pouvions faire un affichage en plusieurs ligne avec *os*. Cela a été très utile afin de faire un affichage d'une liste via un vecteur dans cette surcharge. (class *commande.cpp* ligne32)

- Class Magasin
  - Liste les clients
  - Liste les produits disponibles
  - Liste les commandes en cours
  - Permet la modification de toutes les autres classes via les vecteurs

Nous avons aussi essayé de faire un décompte du nombre de produits achetés ici mais nous n'avons pas abouti à cause d'une boucle infini. Nous avons donc aussi dû abandonner le projet à cet endroit.

2.2 Difficultés rencontrées

Lors de la réalisation de ce travail, nous n'avons pas rencontré de difficulté particulière. Nous avions une idée assez globale du fonctionnement. De plus, nous avions déjà vu le problème des surcharge dans les TP précédents.

3. Conclusion

Nous avons appris et renforcé énormément de connaissances grâce à ce TP, notamment sur l'utilisation des vecteurs qui a été un élément central tout au long de la réalisation du projet, mais aussi sur le lien entre les classes. En effet, nous avons modifié des éléments d'une classe (client) dans d'autres classes et nous avons créé énormément de liens entre toutes nos classe. Cela nous a permis d'organiser nos classes en fonction de nos besoin sans être limités par la séparation des fichiers.

Cependant, nous n'avons pas réussi à faire une interface graphique par manque de savoir-faire et par manque de temps. Nous avons donc pour objectif d'apprendre à réaliser une interface graphique utilisable afin de parfaire nos connaissances et d'avoir un travail plus abouti dans le future.