Université de Bordeaux - Master 1 Bioinformatique

Projet de Programmation en C++ :

lesMinimesEnUnClic

Rapport et présentation du programme

Abou KOUTA et Julie PRATX

31 mars 2020

Table des matières

1 Introduction **2**

2 Analyses **3**

2.1 Contexte 3

2.2 Besoins 3

3 Conception et Fonctionnalités **4**

4 Réalisations **6**

5 Jeu d'essai **8**

6 Problèmes rencontrés **11**

7 Perspectives **12**

8 Conclusion **13**

9 Références **14**

Chapitre 1

Introduction

Crée initialement par Bjarne Stroustrup dans les annees 1980, le C++ est le descendant du langage C. Ces deux langages, bien que semblables au premier abord, sont neanmoins différents. Le C++ propose de nouvelles fonctionnalités, comme la programmation orientee objet (POO). Elles en font un langage tres puissant qui permet de programmer avec une approche différente du langage C [1].

L'objectif de ce projet consiste à implémenter des classes pour informatiser la gestion du port de plaisance de La Rochelle à l'aide d'un programme du nom de : LesMinimesEnUnClic. En effet, le gérant souhaite totalement informatiser la gestion du port et mettre en place une comptabilité exemplaire du port.

La realisation de ce projet a été réalisé a l'aide du langage C++ en s'appuyant sur les différentes fonctionnalites proposées par le sujet. L'application de gestion se controle via un terminal. Celle-ci a dû être developpée de manière à ce que l'experience de l'utilisateur soit la plus simple et la plus accessible possible.

Chapitre 2

Analyses

2.1 Contexte

Dans l'application de gestion, tous les fonctions sont caracterisées par : nbDePlaceAFlot (int), nbDePlaceVisiteurs (int), ndBdePlaceCorpsMort (int).

Chaque type de bâteau a aussi des caracteristiques specifiques définies de la facon suivante :

— Bâteau : typeDeBateau (string), taille (int), tailleMini (int), tailleMaxi (int), nombreDeCabine (int), electricite (String), eau (string), sommeVNH (int), sommeVT1 (int), sommeVT2 (int) sommeEau (int), sommeElec (int).

— Voilier : tailleMaxi (int), tailleMini (int), nombreDeCabine (int), voilier non habitable : id-VNH (int), voilier type 1 : id-VT1 (int), voilier type 2 : id-VT2 (int).

On attribue à tous les usagers un statut : je suis un abonné ou je suis de passage.

2.2 Besoins

Afin de bien comprendre le fonctionnement du programme, il faut avant tout comprendre les besoins de l'utilisateur, donc du gérant du port de La Rochelle, tant sur le plan fonctionnel que sur celui de l'interface utilisateur.

Premièrement, le gestionnaire du port doit disposer d'un catalogue d'informations qui lui permettra de conserver les données concernant les usagers. De ce fait, le programme comprend des conteneurs de la bibliotheque STL pour stocker tous les ensembles ou listes d'objets. Ceux-ci servent à stocker les informations associes à leur clients, les clients eux-mêmes, ainsi que les propositions de réservation.

Apres avoir défni nos structures de données, il nous faut répondre au deuxieme besoin qui est d'établir des intéractions entre ces structures afin de permettre à l'utilisateur de gérer le port.

Enfin, l'utilisateur doit pouvoir afficher le contenu des structures de données en fonction de certains critères s'il le souhaite. Pour cela, il lui sera possible de rechercher le type d'usagers, abonné ou de passage, de la taille du bateau ou du type de voilier (*ces fonctions ne sont pas disponibles*).

Chapitre 3

Conception et Fonctionnalités

Le programme contient plusieurs classes et sous-classes. Il est composé de la sorte :

1. le **programme principal** : il contient le main. Il permet à l'utilisateur d'intéragir avec le programme en appelant des fontions comme, par exemple, la création d'un usager, ou encore, l'attribution d'une place dans ou à l'extérieur du port, ou encore, d'éditer la facture.
2. la **classe Bateau** : elle regroupe toutes les caractéristiques des bateaux, à savoir, la taille (minimale et maximale), le nombre de cabine et le racordement à l'eau et à l'électricité. Elle contient un constructeur par défault (contient les variables initialisées), un constructeur avec un argument (la taille du bateau) et un destructeur. Elle a également des fonctions qui permettent de gérer tout ce qui a un rapport avec le port :
   - **nbDePlaceDispo** qui prend en argument le type de bâteau, le type de visiteur et sa place dans le port ou à l'extérieur, accroché à une bouée à l'extérieur du port (corps mort).
   - **facturationVoilierNonHabitable** qui prend en argument le nombre de jour passé dans le port et le type de visiteur (abonné ou de passage).
   - **facturationVoilierType1** qui prend en argument le nombre de jour passé dans le port et le type de visiteur (abonné ou de passage).
   - **facturationVoilierType2** qui prend en argument le nombre de jour passé dans le port et le type de visiteur (abonné ou de passage).
   - **facturationCorpsMort** qui prend pour argument le nombre de jour passé hors du port.
   - **facturationEau** qui a pour argument un booléen si l'utilisateur veut être raccordé ou non à l'eau moyennant un supplément.
   - **facturationElectricite** qui a pour argument un booléen si l'utilisateur veut être raccordé ou non à l'électricité moyennant un supplément.
   - **totalAPayer** qui prend pour argument le type de bâteau émet une facture à la demande de l'utilisateur. Il regroupe donc le temps passé au port ou hors du port, le supplément en eau et/ou en électricité.
   - **afficheBateau** sans argument, affiche le type de bâteau de l'usager enregistré.

     Toutes ces fonctions sont publiques. Les variables comme la taille ou le nombre de cabine (entre autres) sont privées.
3. la **sous-classe VoilierNonHabitable** : est une classe fille de Bateau. Elle hérite donc de toutes les fonctions de Bateau. Elle possède un constructeur par défault, un constructeur avec un argument (la taille) et un destructeur. Ces constructeurs sont publics. Tandis que la taille maximale (par défault à 9), le nombre de cabine (par défault à 0) et le raccordement à l'eau et à l'électricité qui sont par défault à "non", sont des données privées.
4. la **sous-classe VoilierType1** : est une classe fille de la classe Bateau. Elle hérite donc de toutes les fonctions de Bateau. Elle possède un constructeur par défault, un constructeur avec un argument (la taille) et un destructeur. Ces constructeurs sont publics. Tandis que la taille minimale (par défault à 10), la taille maximale (par défault à 25), le nombre de cabine, le raccordement à l'eau et à l'électricité qui sont par défault à "oui", sont des données privées.

5. la **sous-classe VoilierType2** : est une classe fille de la classe Bateau. Elle hérite donc de toutes les fonctions de Bateau. Elle possède un constructeur par défault, un constructeur avec un argument (la taille) et un destructeur. Ces constructeurs sont publics. Tandis que la taille, le nombre de cabine, le raccordement à l'eau et à l'électricité qui sont par défault à "oui, sont des données privées.
6. la **classe GestionDuPort** : sert à garder une trace du passage des usagers et des bâteaux qui sont présents dans le port. Elle est utile pour savoir, en temps réel, le nombre de places disponibles (à flot, visiteur ou corps mort). De ce fait, elle contient un constructeur par défault, un destructeur ansi que :
   - **afficheListeClient** (*non disponible*) : sert à afficher la liste des clients présent dans le port ou sur attachés à un corps mort.
   - **afficheListePlacesDispo** : sert à afficher le nombre de place disponibles sur chacun des 3 supports (à flot, visiteur et corps mort).
     Mais également, des données privées telles que le nombre de places totales de places à flot (4 700), de places visiteurs (300) et de corps mort disponibles (30).
7. la **classe Usager** : sert à répertoirier tout ce qui a un rapport avec l'usager, à savoir, son identifiant (ID), son type (abonné ou de passage) et le type de bâteau qu'il possède. Elle contient donc un constructeur par défault ainsi qu'un destructeur. Mais aussi des fonctions :
   - **afficheID** : a pour argument le type de bâteau de l'utilisateur et affiche l'identifient de l'utilisateur en fonction de son bâteau. S'il possède un voilier de type 1, son identifiant sera de type VT1XXXX où X correspond à un chiffre entre 0 et 9.
   - **abonnes** : sert à afficher à l'écran "Je suis un abonné".
   - **passagers** : sert à afficher à l'écran "Je suis de passage".
   - **id** (*ne fonctionne pas correctement*) : sert à générer un identifiant pour chaque nouvel usager.

Chapitre 4

Réalisations

Les différentes parties qui vont suivre sont des illustrations de la phase finale de l'application. Celle-ci est affichée dans le terminal.

Un exemple d'utilisation de l'application est presenté ci-dessous. Cette figure nous donne le type d'usager, abonné ou de passage, des informations sur la taille et le type de bateau, la présence ou l'absence de branchement à l'eau ou à l'électricité. On a aussi des renseignements sur le type de place dans ou hors du port : à quai ou en mer ainsi que le nombre de place disponible et la facture que devra régléger l'usager (**Figure 4.1**).

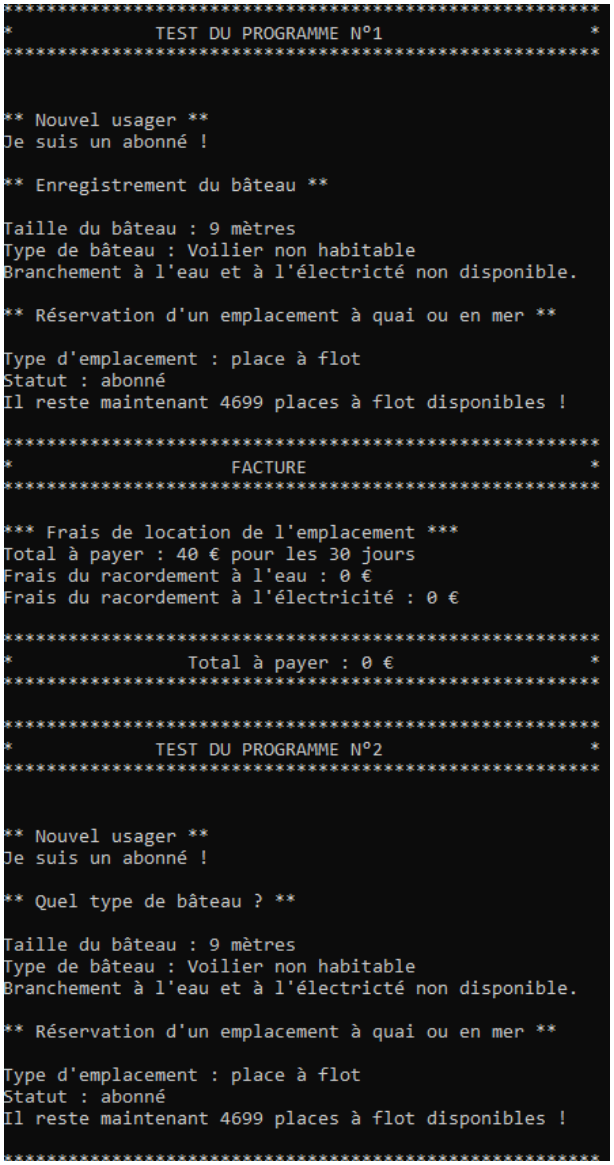
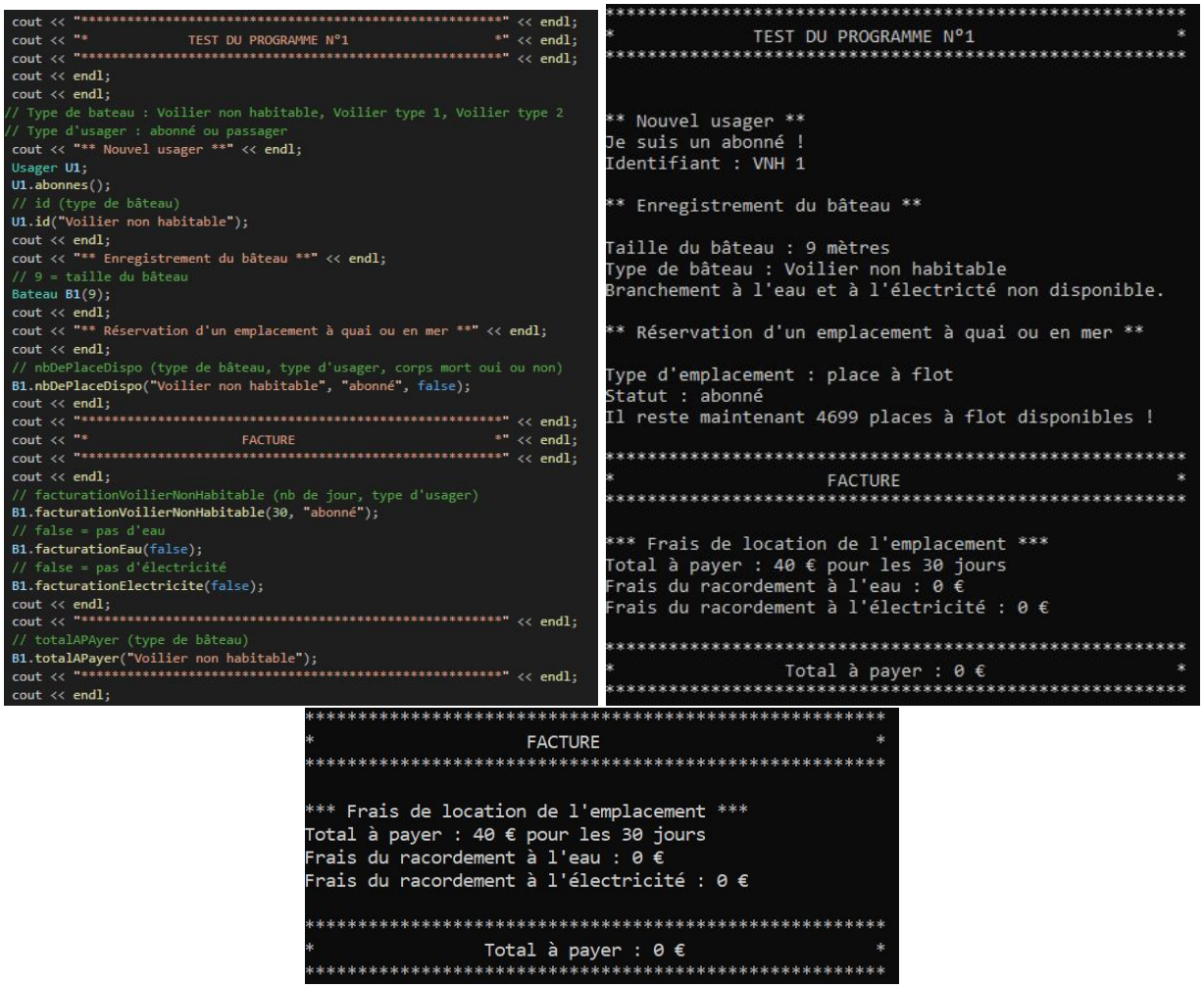


FIGURE 4.1 – Affichage, depuis le terminal, du main.cpp

Chapitre 5

Jeu d'essai

Les différentes figures présentées ci-dessous (**Figure 5.1 à 5.3**), montrent différents tests du programme avec des paramètres différents. La légende indique ces différents paramètres. La **Figure 5.4** montre le nombre de places disponibles dans le port. La **Figure 5.5** montre la liste des usagers qui sont présent dans ou hors du port.

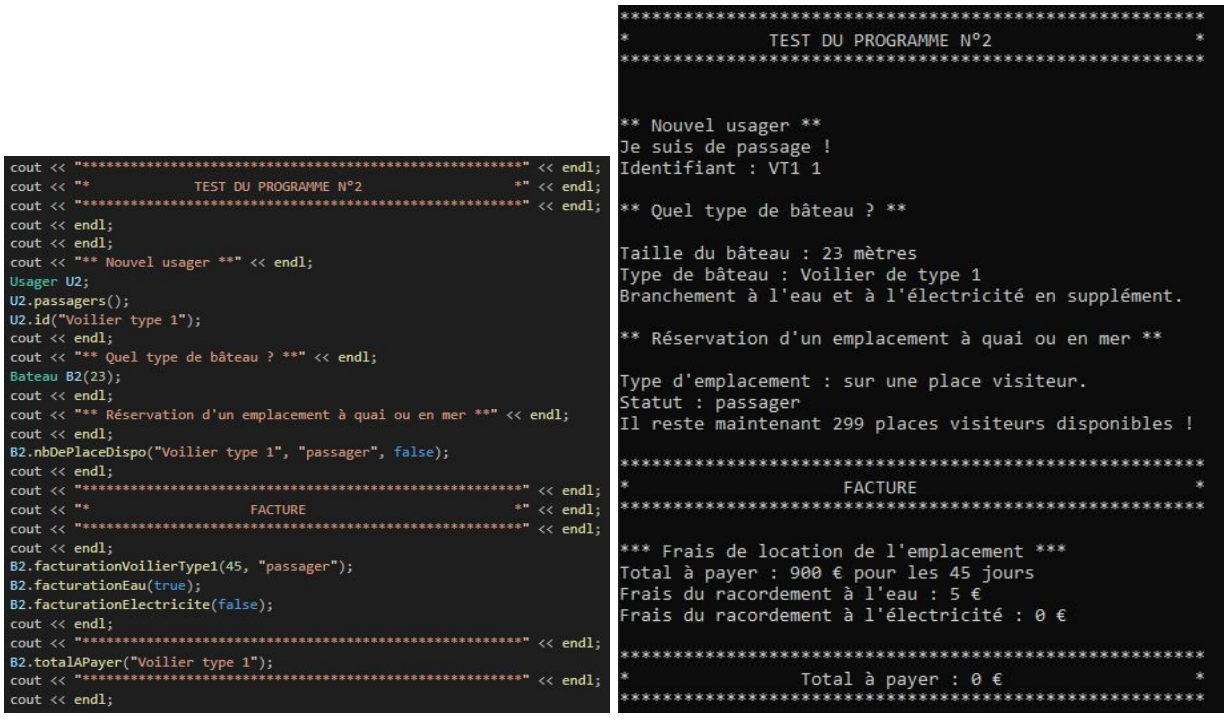


```
cout << "*****************************************************" << endl;
cout << "*             TEST DU PROGRAMME N°1                 *" << endl;
cout << "*****************************************************" << endl;
cout << endl;
cout << endl;
// Type de bateau : Voilier non habitable, Voilier type 1, Voilier type 2
// Type d'usager : abonné ou passager
cout << "** Nouvel usager **" << endl;
Usager U1;
U1.abonnes();
// id (type de bâteau)
U1.id("Voilier non habitable");
cout << endl;
cout << "** Enregistrement du bâteau **" << endl;
// 9 = taille du bâteau
Bateau B1(9);
cout << endl;
cout << "** Réservation d'un emplacement à quai ou en mer **" << endl;
cout << endl;
// nbDePlaceDispo (type de bâteau, type d'usager, corps mort oui ou non)
B1.nbDePlaceDispo("Voilier non habitable", "abonné", false);
cout << endl;
cout << "*****************************************************" << endl;
cout << "*                   FACTURE                         *" << endl;
cout << "*****************************************************" << endl;
cout << endl;
// facturationVoilierNonHabitable (nb de jour, type d'usager)
B1.facturationVoilierNonHabitable(30, "abonné");
// false = pas d'eau
B1.facturationEau(false);
// false = pas d'électricité
B1.facturationElectricite(false);
cout << endl;
cout << "*****************************************************" << endl;
// totalAPAyer (type de bâteau)
B1.totalAPayer("Voilier non habitable");
cout << "*****************************************************" << endl;
cout << endl;
```

```
*****************************************************
*             TEST DU PROGRAMME N°1                 *
*****************************************************


** Nouvel usager **
Je suis un abonné !
Identifiant : VNH 1

** Enregistrement du bâteau **

Taille du bâteau : 9 mètres
Type de bâteau : Voilier non habitable
Branchement à l'eau et à l'électricté non disponible.

** Réservation d'un emplacement à quai ou en mer **

Type d'emplacement : place à flot
Statut : abonné
Il reste maintenant 4699 places à flot disponibles !

*****************************************************
*                   FACTURE                         *
*****************************************************

*** Frais de location de l'emplacement ***
Total à payer : 40 € pour les 30 jours
Frais du racordement à l'eau : 0 €
Frais du racordement à l'électricité : 0 €

*****************************************************
*                Total à payer : 0 €                *
*****************************************************
```

```
*****************************************************
*                   FACTURE                         *
*****************************************************

*** Frais de location de l'emplacement ***
Total à payer : 40 € pour les 30 jours
Frais du racordement à l'eau : 0 €
Frais du racordement à l'électricité : 0 €

*****************************************************
*                Total à payer : 0 €                *
*****************************************************
```

FIGURE 5.1 – Abonné, voilier non habitable, facture



```
cout << "*****************************************************" << endl;
cout << "*             TEST DU PROGRAMME N°2                 *" << endl;
cout << "*****************************************************" << endl;
cout << endl;
cout << endl;
cout << "** Nouvel usager **" << endl;
Usager U2;
U2.passagers();
U2.id("Voilier type 1");
cout << endl;
cout << "** Quel type de bâteau ? **" << endl;
Bateau B2(23);
cout << endl;
cout << "** Réservation d'un emplacement à quai ou en mer **" << endl;
cout << endl;
B2.nbDePlaceDispo("Voilier type 1", "passager", false);
cout << endl;
cout << "*****************************************************" << endl;
cout << "*                   FACTURE                         *" << endl;
cout << "*****************************************************" << endl;
cout << endl;
B2.facturationVoilierType1(45, "passager");
B2.facturationEau(true);
B2.facturationElectricite(false);
cout << endl;
cout << "*****************************************************" << endl;
B2.totalAPayer("Voilier type 1");
cout << "*****************************************************" << endl;
cout << endl;
```

```
*****************************************************
*             TEST DU PROGRAMME N°2                 *
*****************************************************


** Nouvel usager **
Je suis de passage !
Identifiant : VT1 1

** Quel type de bâteau ? **

Taille du bâteau : 23 mètres
Type de bâteau : Voilier de type 1
Branchement à l'eau et à l'électricité en supplément.

** Réservation d'un emplacement à quai ou en mer **

Type d'emplacement : sur une place visiteur.
Statut : passager
Il reste maintenant 299 places visiteurs disponibles !

*****************************************************
*                   FACTURE                         *
*****************************************************

*** Frais de location de l'emplacement ***
Total à payer : 900 € pour les 45 jours
Frais du racordement à l'eau : 5 €
Frais du racordement à l'électricité : 0 €

*****************************************************
*                Total à payer : 0 €                *
*****************************************************
```

FIGURE 5.2 – Passager, Voilier type 1, supp eau, facture

```
cout << "*********************************************************" << endl;
cout << "*             TEST DU PROGRAMME N°3                     *" << endl;
cout << "*********************************************************" << endl;
cout << endl;
cout << endl;
cout << "** Nouvel usager **" << endl;
Usager U3;
U3.passagers();
U3.id("Voilier type 2");
cout << endl;
cout << "** Quel type de bâteau ? **" << endl;
Bateau B3(27);
cout << endl;
cout << "** Réservation d'un emplacement à quai ou en mer **" << endl;
cout << endl;
B3.nbDePlaceDispo("Voilier type 2", "passager", true);
cout << endl;
cout << "*********************************************************" << endl;
cout << "*                    FACTURE                            *" << endl;
cout << "*********************************************************" << endl;
cout << endl;
B3.facturationVoilierType2(3, "passager");
B3.facturationEau(false);
B3.facturationElectricite(false);
cout << endl;
cout << "*********************************************************" << endl;
B3.totalAPayer("Voilier type 2");
cout << "*********************************************************" << endl;
cout << endl;
```

```
*********************************************************
*             TEST DU PROGRAMME N°3                     *
*********************************************************


** Nouvel usager **
Je suis de passage !
Identifiant : VT2 1

** Quel type de bâteau ? **

Taille du bâteau : 27 mètres
Type de bâteau : Voilier de type 2
Attention : places disponibles limitées !
Branchement à l'eau et à l'électricité en supplément.

** Réservation d'un emplacement à quai ou en mer **

Type d'emplacement : sur un corps mort.
Il reste maintenant 29 corps mort disponibles !

*********************************************************
*                    FACTURE                            *
*********************************************************


*** Frais de location de l'emplacement ***
Total à payer : 60 € pour les 3 jours
Frais du racordement à l'eau : 0 €
Frais du racordement à l'électricité : 0 €

*********************************************************
*                Total à payer : 0 €                    *
*********************************************************
```

FIGURE 5.3 – Passager, Voilier type 2, corps mort, facture

```
cout << "*********************************************************" << endl;
cout << "*           PLACES DISPONIBLES DANS LE PORT             *" << endl;
cout << "*********************************************************" << endl;
GestionDuPort G1;
G1.afficheListePlacesDispo();
cout << endl;
```

```
*********************************************************
*            PLACES DISPONIBLES DANS LE PORT            *
*********************************************************
Liste des places disponibles :
Places à flot : 4700
Places visiteurs : 300
Corps mort : 30
```

FIGURE 5.4 – Place disponible dans le port

```
cout << "*********************************************************" << endl;
cout << "*                 LISTE DES USAGERS                     *" << endl;
cout << "*********************************************************" << endl;
G1.afficheListeClient();
```

```
*********************************************************
*                  LISTE DES USAGERS                    *
*********************************************************
Liste des clients :
ID VNH :
ID VT1 :
ID VT2 :
```

FIGURE 5.5 – Liste des usagers du port

Chapitre 6

Problèmes rencontrés

— Au niveau de l'établissement de la facture, il est impossible d'additionner les résultats obtenus dans les fonctions précédentes, à savoir, la somme à payer pour la place dans le port, l'ajout éventuel du raccordemnt à l'eau ou à l'électricité. Le totale reste désespérément à zéro après divers essais. Il faudrait mettre toutes les données obtenues dans un tableau pour pouvoir ensuite les additionner mais on n'y est pas arrivé.

— Au niveau du compteur du nombre de place disponible. Le problème c'est que le compteur se réinitialise à chaque création d'un nouvel usager avec attribution d'une place.

— Même problème pour l'attribution d'un identifiant pour l'usager.

Chapitre 7

Perspectives

Les compétences que l'on ne métrise pas encore sont :

— la création et l'utilisation des pointeurs et des références : addition sur des pointeurs (notre problème pour l'édition de la facture), incrémentation des pointeurs (pour notre problème d'identifiant) et la mise à jour en temps réel des places disponibles au port.

— la manipulation des tableaux : pour justement pouvoir faire des additions sur les pointeurs et gérer ainsi qu'afficher la liste des usagers du port.

— l'utilisation des destructeurs : on sait que ça sert à libérer la mémoire mais on ne sait pas s'en servir correctement.

De plus, le programme n'est pas très ergonomique. La créaation d'un menu serait un plus. L'ajout d'un manuel serait également une bonne idée. Mettre trop de commentaire sur le script risque plus de le rendre illisible que de le rendre efficace.

Chapitre 8

Conclusion

Ce projet de gestion sur le port de La Rochelle lesMinimesEnUnClic nous a permi de mieux comprendre le langage C++.

En effet, c'est un langage très complet mais aussi très compliqué à apprendre et à manipuler. C'est également un langage orienté objet qui permet de développer des specificités très intéressantes.

De plus, développer ce programme en bînome est un travail très intéressant qui permet d'apprendre autrement. Nous avons réalisé notre projet à l'aide d'un dépôt Git [2]. Ce fût plus qu'indispensable pour coordonner nos codes et ainsi réaliser notre projet plus aisément (au vue des conditions actuelles qui ne permettent aucun contact ni rassemblement).

Chapitre 9

Références

[1] C++, Wikipédia, (2020)

[2] lesMinimesEnUnClic github