
Projet C++ : There is no planet B

Margaux Dufresne, Solène Leclair

MAIN 4

Encadrante : Mme. Braunstein



Table des matières

1	Introduction	2
1.1	Objectifs de notre programme	2
1.2	Lancement et utilisation	2
2	Explication du code	2
2.1	Les classes	2
2.2	Utilisateur	3
2.3	L'eau	3
2.4	Les protections hygiéniques	3
2.5	L'affichage graphique et le main	4
3	Diagramme UML	6
4	Conclusion	8

1 Introduction

1.1 Objectifs de notre programme

Dans le cadre du cours de C++ encadré par Madame Braunstein, nous avons dû créer et implémenter une application en C++ mettant en avant la problématique *There is no planet B*. Le but était de mettre en pratique les notions vu en cours de façon libre, sur un projet par groupe de 2. Pour notre part, nous avons décidé de mettre en avant les économies financières et environnementales que l'on peut faire en appliquant quelques gestes écologiques. L'objectif est de sensibiliser l'utilisateur à l'écologie en mettant en avant les gains potentiels qu'ils obtiendraient à les utiliser.

Le programme compare :

- Les économies en utilisant des protections hygiéniques réutilisables
- Les bien-faits d'utiliser une gourde par rapport à une bouteille en plastique.

Le programme présente :

- 8 classes
- 3 niveaux de hiérarchie
- 2 fonctions virtuelles différentes
- 2 conteneurs différents de la STL : maps et vecteurs
- Un Makefile avec une règle "all" et une règle "clean"
- Des tests unitaires

1.2 Lancement et utilisation

Compilation Nous avons utilisé un Makefile pour compiler les différents fichiers de l'application. Cet outil permet de générer des fichiers objets de tous les fichiers sources utilisés tout en y intégrant les bibliothèques nécessaires. Le Makefile fait ensuite l'édition de lien en un exécutable appelé main.

Lancement Pour lancer notre application, il est nécessaire d'avoir SDL2 ainsi que les différentes bibliothèques l'accompagnant installées sur votre machine. Puis il suffit simplement de se placer dans le répertoire où sont enregistrés les fichiers et d'utiliser la commande `make` dans le terminal puis de lancer l'exécutable avec `./main`. Nous pouvons aussi faire `make clean` afin de supprimer tous les fichiers `.o` et l'exécutable.

Utilisation Lors du lancement du programme, nous demandons d'abord les informations sur l'utilisateur (nom et prénom), puis le type de recherche qu'il veut effectuer : se renseigner sur les gourdes ou sur les protections hygiéniques en cliquant sur l'image de la fenêtre graphique. Après cela, une question supplémentaire sera posée dans le terminal et une autre fenêtre avec les courbes d'évolution au cours du temps apparaîtra.

Tests unitaires Dans le dossier `tests` il est possible de réaliser des tests unitaires sur les principales fonctions du programme en lançant la commande `make testcase` puis `./testcase`.

2 Explication du code

Lors de la conception du projet, nous avons créé 8 fichiers `.hh` correspondant aux fichiers d'en-tête pour la création des classes auxquels sont associés un fichier `.cc`.

2.1 Les classes

Dans ce programme nous allons créer 8 classes correspondant à différents types de solutions écologiques qui sont explicités ci-dessous.

Détail des fichiers : Il y a évidemment un fichier `main.cc` qui contient le lancement des différentes fonctions puis d'autres classes sont créées :

- **Utilisateur** : Déclaration du nom et prénom de l'utilisateur et de son choix de recherche.
- **Eau** : fils de **Utilisateur** étudiant l'impact de la consommation d'eau en bouteille en plastique ou en gourde
- **Bouteille** : fils de **Eau** pour l'eau minérale et de sources consommée en bouteille
- **Gourde** : fils de **Eau** pour boire de l'eau du robinet dans une gourde
- **Protection** : fils de **Utilisateur** contient les différents types de protections hygiéniques.
- **Cup** : fils de **Protection** pour les calculs des données concernant la coupe menstruelle.
- **Jetable** : fils de **Protection** pour les calculs des données concernant les serviettes hygiéniques jetables.
- **Lavable** : fils de **Protection** pour les calculs des données concernant les serviettes hygiéniques lavables.

2.2 Utilisateur

La classe **Utilisateur** est la classe parent de ce programme. Elle définit le nom et prénom du client puis son choix de recherche. Nous avons créé un constructeur par défaut qui demande de compléter les variables directement dans le terminal, un autre qui prend en paramètre les données souhaitées et un constructeur par copie.

2.3 L'eau

La classe **Eau** est parent de 2 classes **Gourde** et **Bouteille**. Elle est elle-même héritière de **Utilisateur** car c'est lui qui fait la requête.

- `std::size_t _litre_j` ; : attribut privé indiquant combien d'eau boit l'utilisateur
- `virtual void print() const` ; : permet d'afficher les informations sur les fils de **Bouteille**
- `void createTable()` : distribue les 3 cartes aux différents joueurs
- Des constructeurs (par défaut ou copie) : Demande à l'utilisateur la quantité d'eau qu'il boit chaque jour afin de la stocker dans l'attribut `_litre_j`
- Un getter pour obtenir le nombre de litre d'eau bu par l'utilisateur.

Gourde et Bouteille : Ces 2 classes héritent de **Eau**. Elles possèdent toutes les deux des constructeurs, destructeurs, getters, des fonctions de calcul et d'affichage (virtual). Pour chacune, leurs attributs sont un dictionnaire avec les prix ou la consommation en CO2 chaque année. La classe **Gourde** a aussi 2 attributs sur la durée de vie d'une gourde et le prix d'achat initial (pour calculer quand il faut la renouveler). Une gourde en inox a une durée de vie moyenne de 3 ans et vaut en moyenne 20 ans.

2.4 Les protections hygiéniques

Nous avons décidé de définir les protections hygiéniques grâce à une classe regroupant 3 classes filles représentant les différents types de protection. Ainsi on a la classe **Protection** qui est parent de 3 classes **Cup**, **Jetable**, **Lavable**. Cette classe hérite aussi de **Utilisateur** et donc en a toutes ses propriétés. Elle possède différents attributs :

- `std::size_t nb_annees` ; : attribut public déterminant sur combien d'année les résultats seront calculés (nous l'avons mise à 10 lors du projet).
- `std::size_t _nb_cycle` ; : attribut public déterminant sur combien de cycle les résultats seront calculés lors du projet nous l'avons initialisé à 13 car une personne menstruée a généralement 13 cycles par an.
- `std::size_t _nbsprotjour` ; : attribut public définit par l'utilisateur qui indique combien de protection (tampon ou serviette) iel utilise lors d'une journée.

La classe **Protection** possède aussi des constructeurs, une fonction d'affichage virtuelle ainsi qu'un getter.

Cup, Lavable et Jetable : Les 3 classes permettent de calculer le prix de ces 3 alternatives en fonction du temps (en année ou par cycle). Chacune de ses classes sont composées de deux `std::vector<float>`, qui sont des attributs publics, un représentant les différents résultats calculés selon les cycles (`res_cycle`) et l’autre les résultats selon les années (`res_an`). L’utilisation de vecteur permet une utilisation plus dynamique de l’espace sans allouer d’espace inutile. Chaque classe possède en plus des attributs privés qui dépendent de la classe.

Les attributs privés de Cup :

- `int _duree = 10 ;` : Défini la durée de vie du produit en question, ici une coupe menstruelle dure environ 10 ans.
- `int _prixuni = 20;` : Représente le prix unitaire du produit, une cup coute en moyenne 20 euros.
- `float _prixan = _prixuni/_duree;` : calcul du prix que cela représente par an.
- `float _prixcycle = _prixan/nb_cycle;` : calcul du prix que cela représente par cycle.

Les attributs privés de Lavable :

- `int _duree = 5 ;` : Défini la durée de vie du produit en question, ici une serviette hygiénique lavable dure environ 5 ans.
- `int _prixuni = 70;` : Représente le prix unitaire du produit, nous avons estimé que pour effectuer un cycle dans de bonne condition il faut environ 7 serviettes lavables à laver régulièrement, à 10 euros chacune.
- `float _prixan = _prixuni/_duree;` : calcul du prix que cela représente par an.
- `float _prixcycle = _prixan/nb_cycle;` : calcul du prix que cela représente par cycle.

Les attributs privés de Jetable :

- `float _prixuni = 0.25;` : Défini le prix moyen d’une serviette hygiénique/un tampon dans le commerce.
- `int _nb_jour_cycle = 5` : On estime que le cycle moyen chez une personne menstruée est de 5 jours.
- `float _prixcycle = (_prixuni*_nbprotjour)*_nb_jour_cycle;` : calcul du prix que cela représente par cycle.
- `float _prixan = _prixcycle * nb_cycle;` : calcul du prix que cela représente par an.

2.5 L’affichage graphique et le main

Enfin, nous avons utilisé la bibliothèque graphique SDL2. C’est une bibliothèque tierce (à installer soi-même) qui est surtout utilisée lors de la réalisation d’application graphique 2D. Cela nous a permis d’afficher la page d’accueil [1] qui permet de à l’utilisateur de faire le choix entre étudier les consommations des gourdes ou des protections hygiéniques.

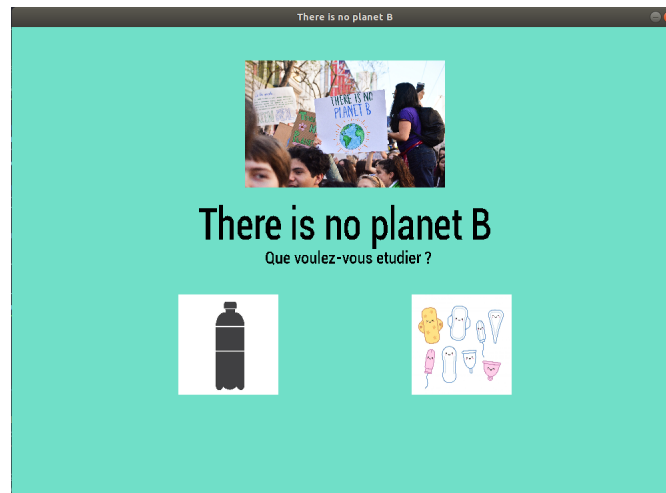


FIGURE 1 – Page d'accueil

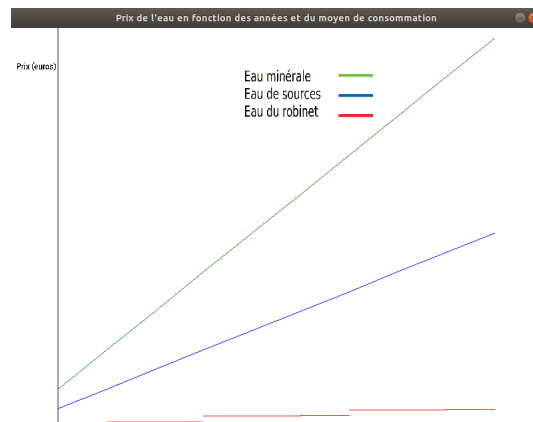


FIGURE 2 – Affichage des courbes

Il faut ensuite répondre à une question sur le terminal pour voir afficher le graphique souhaité [2].

Pour avoir les valeurs exactes, le terminal [3] donne l'évolution des prix et de la consommation en CO2 par année.

```
Nom ?
leclair
Prénom ?
solene
Combien buvez vous de litres d'eau par jour ?
1
L'année 0 il a dépensé : 21.0958 €
L'année 1 il a dépensé : 22.1915 €
L'année 2 il a dépensé : 23.2873 €
L'année 3 il a dépensé : 44.383 €
L'année 4 il a dépensé : 45.4788 €
L'année 5 il a dépensé : 46.5745 €
L'année 6 il a dépensé : 67.6702 €
L'année 7 il a dépensé : 68.766 €
L'année 8 il a dépensé : 69.8617 €
L'année 9 il a dépensé : 90.9575 €
      Résultat minéral :
L'année 0, il a dépensé : 146.1 €
L'année 1, il a dépensé : 292.2 €
L'année 2, il a dépensé : 438.3 €
L'année 3, il a dépensé : 584.4 €
L'année 4, il a dépensé : 730.5 €
L'année 5, il a dépensé : 876.6 €
L'année 6, il a dépensé : 1022.7 €
L'année 7, il a dépensé : 1168.8 €
L'année 8, il a dépensé : 1314.9 €
L'année 9, il a dépensé : 1461 €
      Résultat source :
L'année 0, il a dépensé : 73.05 €
L'année 1, il a dépensé : 146.1 €
L'année 2, il a dépensé : 219.15 €
L'année 3, il a dépensé : 292.2 €
L'année 4, il a dépensé : 365.25 €
L'année 5, il a dépensé : 438.3 €
L'année 6, il a dépensé : 511.35 €
L'année 7, il a dépensé : 584.4 €
L'année 8, il a dépensé : 657.45 €
L'année 9, il a dépensé : 730.5 €
      Une bouteille plastique consomme comme CO2 :
L'année 0 : 143.543 kg
L'année 1 : 287.087 kg
L'année 2 : 430.63 kg
L'année 3 : 574.173 kg
L'année 4 : 717.716 kg
L'année 5 : 861.26 kg
L'année 6 : 1004.8 kg
L'année 7 : 1148.35 kg
L'année 8 : 1291.89 kg
L'année 9 : 1435.43 kg
```

FIGURE 3 – Affichage des valeurs dans le terminal

3 Diagramme UML

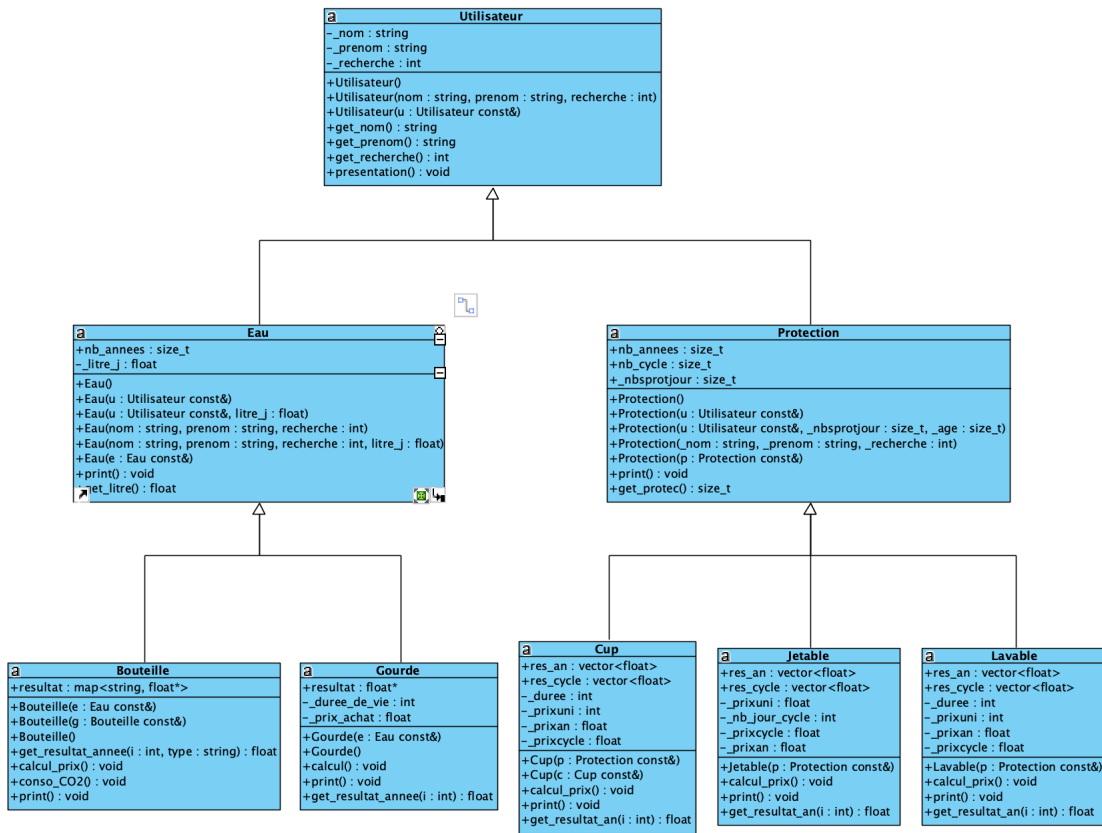


FIGURE 4 – Diagramme UML de l'application

4 Conclusion

Lors de ce semestre nous avons découvert la programmation objet en C++. Nous y avons appris les classes, les hiérarchies, les conteneurs, etc.

Réaliser un projet aussi libre, nous a permis de mettre facilement en pratique et de comprendre ces notions en les abordant selon l'angle qui nous plaisait. Nous avons aussi utilisé la librairie graphique SDL que nous avons déjà un peu manipulé l'année dernière. L'environnement et l'écologie sont des thèmes qui nous intéressent et tiennent à coeur, c'est pourquoi faire des infographies sur des solutions écologiques nous a de suite plu. Ce projet nous a fait prendre conscience du coût des règles dans la vie d'une femme mais également du prix de l'eau. Ces dépenses peuvent paraître anodines mais on voit qu'elles représentent un budget important sur le long terme.

C'était un projet très enrichissant qui nous a fait beaucoup réfléchir en groupe. Nous pensons maintenant avoir de solides bases en C++ pour pouvoir les réutiliser lors de futurs projets.

Références

- [1] Les petits calculs savants : Je passe à la gourde : <https://les-petits-calculs-savants.fr/les-infographies/#jp-carousel-314> (consulté le 10/12/2021)
- [2] Les petits calculs savants : Je passe à la cup : <https://les-petits-calculs-savants.fr/les-infographies/#jp-carousel-341> (consulté le 10/12/2021)
- [3] Code source : <https://github.com/soleneleclair/EISE-MAIN-4-C-Projet.git>