



Find Your Personal Product

FYPP

UTBM

Compte rendu de projet IFT

Floret Somia

Jeanpetit Chloé

Alexandre Matéo

Daghoughi Leïla

Choix du projet

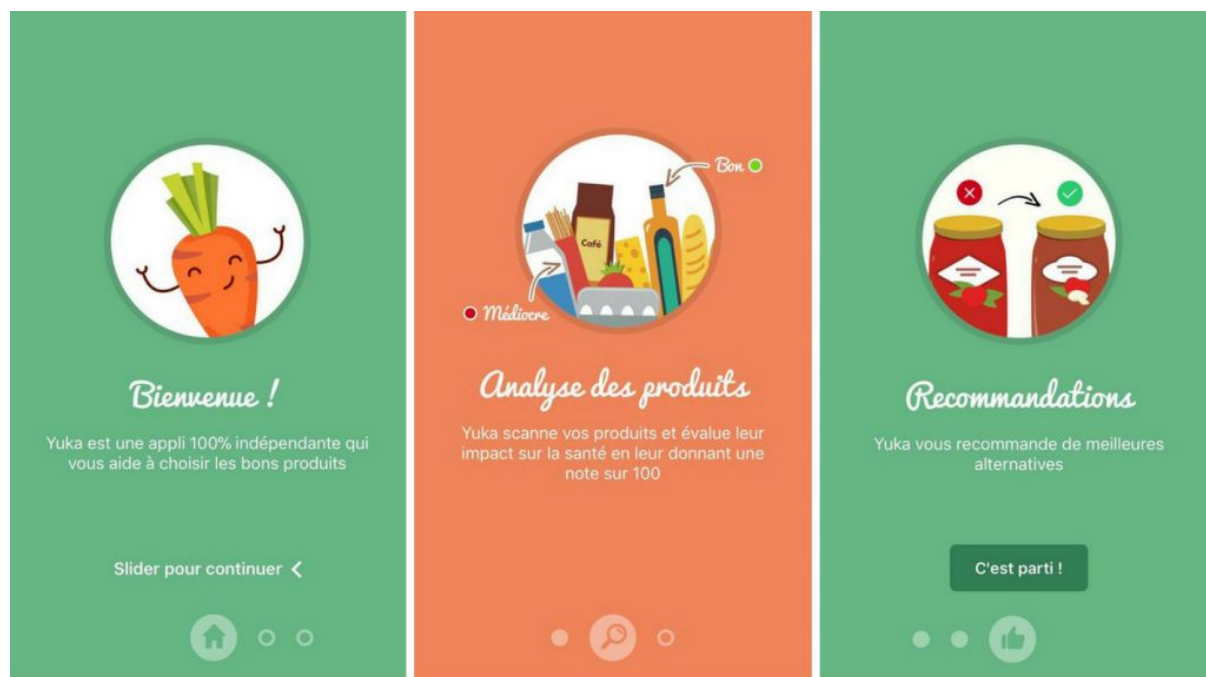
Ce projet est né de l'observation de la vie quotidienne. En effet, les grandes surfaces d'aujourd'hui débordent de produits et de marques différentes, il est donc souvent difficile de faire son choix.

Le but de notre projet est donc d'aider les clients à faire leur propre choix de produits de toute sorte.

Notre projet compare différents produits concurrents, puis les classe selon différents critères préalablement sélectionnés.

Un projet qui semble pourtant familier...

Nous nous sommes inspirés d'une application dénommée Yuka, très utilisée, elle sert à aider le consommateur à choisir leurs produits et permet également de les comparer grâce à une très large base de données qui se base sur un critère: la composition des aliments et des produits cosmétiques.



But du projet

1. Se familiariser avec les étapes de la réalisation d'un projet
2. Apprendre à travailler en équipe et s'entraider au sein du groupe

3. Mettre en œuvre les outils découverts en cours afin d'aboutir à un résultat concret
4. Mener à bien des recherches afin de surmonter d'éventuelles difficultés

Étapes de réalisation du projet

Pour mener à bien ce présent projet, nous avons divisé le travail à faire en plusieurs étapes distinctes afin de travailler de façon progressive, voici nos étapes:

Étape 1

Se rendre dans des grandes surfaces pour observer la répartition des articles, les différentes marques, les différents prix...

Étape 2

Sélection de quelques articles, et recherche sur les différentes marques, leurs prix, les différents labels... et réalisation de tableaux excel avec les différentes informations

Étape 3

Utilisation des fichiers csv

Transformer les tableaux excel en fichier csv pour pouvoir les utiliser par la suite dans un programme.

Étape 4

Utilisation des structures

Étape 5

Détermination du système de notation des produits selon les critères sélectionnés:
Système de points et de coefficients

Étape 6

Calcul de la note d'un produit

Pour chaque marques différentes, les points sont additionnés (point du Prix et du Label), pour obtenir la note finale (différente selon le critère choisi).

Étape 7

Saisie du choix de l'utilisateur

Étape 8

Réalisation du menu

Étape 9

Création du `utils.h` et du `main.c`

Étape 10

Afficher les résultats

Étape 11

Ordonner les valeurs dans un ordre croissant (non terminé)

Algorithme du programme

Choisir le produit dans l'arborescence du menu

Choisir le critère : prix ou label

Faire

Si choix label **alors**

 multiplier par 2 la troisième colonne du tableau qui correspond au label

Sinon choix prix **faire**

 Multiplier par 2 la deuxième colonne du tableau correspondant au prix

Fin boucle si-sinon

Additionner les deux colonnes du tableau

Ordonner les données dans un ordre croissant

Afficher le produit correspondant aux résultat le plus élevé

Réalisations des étapes

Après avoir établi les étapes de notre projet, et après avoir fait quelques recherches préalables sur le rangement des produits dans des grandes surfaces, nous nous sommes ainsi répartis le projet en plusieurs tâches.

Étape 1 et 2: Recherche et réalisation des tableaux excel

Recherche des différents produits, et établissement des tableaux. Chaque tableau correspond à un type de produit, et dans chacun on y retrouve différentes marques ainsi que différents critères pour chacun, comme le prix, les labels, la composition...

Marque	Prix	Label
Nutella	6.28	0
Nustikao	3.63	0
Nocciolata	11.74	3
Patamilka	7.79	0
Jardin bio	8.15	2
Ovomaltine	7.36	0

Étape 3: Utilisation de structure

A l'origine nous avons décidé d'utiliser des tableaux multidimensionnels mais le problème de type unique nous a obligé à utiliser une structure. En effet un tableau permet de regrouper des éléments de même type, c'est à dire codés sur le même nombre de bits et de la même façon. Or, dans notre projet nous devons rassembler des éléments différents tels que des chaînes de caractères, des entiers...

Les structures permettent de remédier à cette lacune des tableaux, en regroupant des objets (des variables) au sein d'une entité repérée par un seul nom de variable.

On crée alors une structure "Produit" pour rendre les programmes plus simple. La structure va correspondre à un produit c'est à dire à une entrée dans un des fichiers CSV. Lors de la déclaration de la structure, on indique les champs de la structure, c'est-à-dire le type et le nom des variables qui la composent.

```
typedef struct Produit
{
    char marque[32];
    double prix;
    int label;
} Produit;
```

Dans cette structure il est décrit le nom de la marque (chaîne de caractère), le prix (un flottant) , et le label (un entier).

Étape 4: Utilisation de fichier csv

Les tableau excel sont convertis en fichier csv.

Le fichier csv est un format texte qui permet de représenter les données d'un tableau sous forme de valeurs séparées par des virgules. C'est un bon moyen pour faire passer les valeurs d'un tableau excel dans un programme en c.

Marque	Prix Kg	Label	
Nutella	6.28	0	
Nustikao	3.63	0	
Nocciolata	11.74	3	
Patamilka	7.79	0	
Jardin bio	8.15	2	
Ovomaltine	7.36	0	

===== >>

Marque,Prix Kg,Label
Nutella,6.28,0
Nustikao,3.63,0
Nocciolata,11.74,3
Patamilka,7.79,0
Jardin bio,8.15,2
Ovomaltine,7.36,0

Après avoir converti les tableaux en fichier en csv, on récupère le contenu de ces fichiers pour ensuite stocker les informations dans un tableau.

```
int parse_csv_product(char *nomFichier, Produit *listeProduits)
```

La fonction parse_csv_product prend un tableau de Produit (vide) pour ensuite y ajouter les valeurs dedans. Cette méthode consiste à faire passer une valeur en paramètre (argument) pour récupérer le résultat sur cette même valeur (ici le tableau).

Dans la fonction parse_csv_product on lit ligne par ligne et pour chaque ligne on sépare les mots délimités avec une virgule "," à l'aide de la fonction strtok().

```
char *token = strtok(ligne, ",");
```

Pour chaque ligne, on ajoute les champs du produit au tableau de produits, qui est passé en paramètre. Ainsi, on a récupérer les valeurs d'un fichier csv vers une structure de donnée exploitable en C.

De plus en valeur de retour elle retourne le nombre de produit qu'elle a pu trouver dans le fichier CSV (sachant qu'on a une macro qui correspond au nombre de produit max NB_PRODUITS_MAX).

Étape 5 et 6: Notation des produits

Les produits sont notés selon un nombre de points et des coefficients selon le critère choisi.

A partir de la structure de donnée, on veut calculer la note en fonction du critère.

```
void calculer_note_produits(Produit *produits, int nbProduits, int
critere, double *notes_sortie)
```

Pour cela on utilise la fonction `calculer_note_produits()` qui va prendre en paramètre:

- la liste des produits (qui ont été extraits du fichier csv)
- le nombre de produits à classer (`nbProduits`)
- le critère (on a créé deux macro `CRITERE_LABEL` et `CRITERE_PRIX` pour représenter les valeurs de ces critères)
- et enfin comme pour la fonction `parse_csv_product()` on met un tableau en paramètre pour récupérer les notes qui y ont été calculées dedans

On crée une boucle `if`:

Dans le 1er cas, l'utilisateur choisi le critère du prix:

La note du prix est alors multiplié par 2

Dans le 2ème cas, l'utilisateur choisi le critère du label:

La note du label est alors multiplié par 2

```
for (int i = 0; i < nbProduits; i++)
{
    if (critere == CRITERE_PRIX)
    {
        notes_sortie[i] = (prixMax - produits[i].prix) * 2 +
produits[i].label;
    }
    else if (critere == CRITERE_LABEL)
    {
        notes_sortie[i] = prixMax - produits[i].prix +
produits[i].label * 2;
    }
}
```

La note final de chaque produit est calculée par la somme de la note du prix et de la note du label.

Étape 7: Saisie du choix de l'utilisateur

Pour finir, il faut récupérer la saisie de l'utilisateur. Pour cela on crée une structure "ChoixMenu", qui correspond au choix de l'utilisateur à la fin du menu.


```
typedef struct ChoixMenu
{
    char *fichier;
    int critere;
} ChoixMenu;
```

Le champ fichier correspond au chemin vers le fichier CSV et "critère" au critère que l'utilisateur a choisi. Pour simplifier les choses, on a défini tous les chemins vers les fichiers en tant que macro dans menu.c.

```
#define PATE_A_TARTINER "./resources/Pate_a_tartiner.csv"
#define OEUFS "./resources/oeufs.csv"
#define LIEGEOIS "./resources/liegeois.csv"
#define JAMBON_BLANC "./resources/jambon_blanc.csv"
```

Étape 8 : Réalisation du menu

Au démarrage du programme: un menu à plusieurs choix est présenté à l'utilisateur afin de le diriger vers la catégorie qui lui correspond.

Le programme attend que l'utilisateur entre le numéro correspondant à son choix, ce choix mène à un sous menu, qui, de même, propose plusieurs choix. Avec une arborescence l'utilisateur arrive aux produits qu'il souhaite. Le menu est réalisé avec des imbrications de SWITCH, application généralisé de IF. Chaque instruction SWITCH inclut plusieurs étiquettes CASE, correspondant aux différentes possibilités.

```
switch (i)
{
    int i;
    case 3:
        fprintf(stdout, "=== CONFITURE,MIEL,PATE A TARTINER===\n\n");
        fprintf(stdout, "1. Confiture \n");
        fprintf(stdout, "2. Miel \n");
        fprintf(stdout, "3. Pate à tartiner\n");
        fprintf(stdout, "\nVotre choix ? \n\n");
        scanf("%d", &i);
```

Peu importe le produit final, le menu se termine sur le choix du critère. L'utilisateur choisit soit le prix (CRITERE_PRIX=1) soit le label (CRITERE_LABEL=2).

Le menu est relié à la structure ChoixMenu ainsi qu'au fichier CSV des différents produits.

```
return (ChoixMenu){PATE_A_TARTINER,i};
break;
```

La syntaxe(ChoixMenu){PATE_A_TARTINER, i} permet de créer et initialiser la structure. Entre parenthèse on retrouve la structure Choixmenu et dans les accolades on retrouve les valeurs des champs .fichier. Ici nous avons le fichier PATE_A_TARTINER (que l'on a défini préalablement).

```
default;
    fprintf (" Vous n'avez pas entré un nombre correcte. \n\n");
}
break;
return (ChoixMenu){NULL, 0};
}
```

Et si jamais aucun fichier ne correspond, la fonction retourne la structure ChoixMenu initialisé avec {fichier= NULL, critère=0}

Lors des choix si l'utilisateur ne rentre pas un chiffre correspondant à un choix, nous avons créé une instruction default qui affiche le message suivant: Vous n'avez pas entré un nombre correcte, et le programme s'arrête.

Voici ci-dessous, l'arborescence du chemin pour arriver au choix de la pâte à tartiner.

I. Quelle gamme de produits ? II. Quels produits alimentaires ?

1. Produits alimentaires
2. Boissons
3. Hygiène et beauté
4. Bébé
5. Textile et chaussures
6. Scolaires
7. Technologie
8. Produits surgelés
9. Animalerie
10. Loisir
11. Entretien

Choix: 1. Produits alimentaires

1. épicerie salée
2. épicerie sucrée
3. produits frais
4. fruits et légumes

Choix: 2. Epicerie sucrée

III. Quel produit sucré ?

1. cafés, thé, infusions
2. sucre, farines, ingrédients
3. céréales
4. biscuits, pâtisserie
5. confiture, miel, pâte à tartiner
6. chocolat tablette, en poudre
7. Biscottes, pain

Choix: 5. Confiture, miel, pâte à tartiner

IV. Quels produits ?

1. Confiture
2. Miel
3. Pâte à tartiner

Choix: 3. Pâte à tartiner

V. Quelles pâte à tartiner?

1. Nocciolotta
2. Nutella
3. Ovomatine
4. Jardin Bio
5. Patamilka
6. Nustikao

Étape 9: Création du `utils.h` et du `main()`

Une fois les programmes terminés, on répertorie toutes les fonctions utilisées dans le fichier `utils.h` pour leur création, puis on lie toutes les fonctions dans le `main()`.

Programme utils.h:

```
#include "utils.c"

int parse_csv_product(char *nomFichier, Produit *listeProduits);
void print_produit(Produit produit);
void print_liste_produits(Produit *produits, int nbProduits);
void classer_produits(Produit *listeProduits, int nbProduits);
int max_prix_produit(Produit *produits, int nbProduits);
void calculer_note_produits(Produit *produits, int nbProduits, int
critere, double *notes_sortie);
```

Programme main():

```
int main()
{
    Produit produits[NB_PRODUITS_MAX];
    double notes[NB_PRODUITS_MAX];
    ChoixMenu choix = menu();
    if (choix.fichier == NULL)
    {
        fprintf(stdout, "ERREUR : Ce cas n'a pas encore été géré.\n");
        return EXIT_FAILURE;
    }
    int nbProduits = parse_csv_product(choix.fichier, produits);
    calculer_note_produits(produits, nbProduits, choix.critere, notes);
    fprintf(stdout, "Voici la liste des meilleurs produits selon vos
critères :\n");
    for (int i = 0; i < nbProduits; i++)
    {
        fprintf(stdout, "- %s\t%0.2f€\t[Score = %0.2f]\n",
produits[i].marque, produits[i].prix, notes[i]);
    } return 0; }
```

Étape 10: Afficher le résultats

```
Lecture de ./resources/Pate_a_tartiner.csv
Voici la liste des meilleurs produits selon vos critères :
- Nutella      6.28€   [Score = 10.92]
- Nustikao     3.63€   [Score = 16.22]
- Nocciolata  11.74€   [Score = 3.00]
- Patamilka    7.79€   [Score = 7.90]
- Jardin bio   8.15€   [Score = 9.18]
- Ovomaltine   7.36€   [Score = 8.76]
```

Après avoir rempli ces choix dans le menu, le programme affiche la liste des produits, avec les scores selon les critères choisis. Le produits ayant la meilleur note correspond au meilleur produit.

Étape 11: Ordonner les scores dans l'ordre croissant

La fonction ordonner les résultats est encore inachevée. En effet, l'idéal serait d'afficher les produits dans l'ordre croissant, ce qui permettrait une lecture plus rapide.

Nous avons tout de même réalisé le programme pour ordonner un tableau unidimensionnel de taille 6, 6 nombres à virgule. On choisit donc double comme type de donnée.

```
double tableau[6] = {6.28,3.63,11.74,7.79,8.15,7.36}
```


Ce programme prend chaque valeur, et la compare à la valeur précédente. Si la valeur 2 est supérieure à la valeur 1, elle se place dans la case supérieure et si elle est inférieure, elle se place dans la case précédente.

On utilise ce principe pour chaque valeur. Et au final on obtient, un tableau avec les valeurs rangées dans l'ordre croissant.

Problèmes rencontrés

Ce projet était pour nous l'un des premiers projets d'informatique que nous réalisions. Il était donc difficile au début de définir les différentes étapes de notre projet.

De plus, à l'origine nous avons décidé d'utiliser des tableaux multidimensionnels mais le problème de type unique nous a obligé à utiliser une structure. Nous avons donc perdu à certain temps sur la tentative de réaliser le projet avec des tableaux. Ce qui a amplifié notre mauvaise gestion du temps.



Travailler en groupe sur un projet, nous a fait prendre conscience de l'importance de l'organisation, du rangement dans les zones d'échange et de créer des programmes les plus clairs et propres possible pour qu'ils soient simples à analyser et à utiliser.

Conclusion

Ce projet a été pour nous quatre très enrichissant.

Nous avons dans l'ensemble validé les objectifs à l'exception de la fonction ordonner.

Après quelques mois passés sur ce projet, on peut remarquer que beaucoup de parties peuvent être simplifiées, comme le menu...

De plus, ces programmes fonctionnent pour une petite quantité de données, mais à l'échelle d'une grande surface, avec des prix, des labels qui changent très souvent, il sera difficile de répertorier tous les produits dans des tableaux. Ensuite, le système de notation est très limité et ne permet pas d'avoir une idée réelle de la note d'un prix.

Pour conclure, nous avons appris à travailler en groupe, chacun avait des responsabilités, tout en ayant l'aide des autres personnes du groupe.

Pour écrire nos programmes nous nous sommes aidé des programmes du projet learn iT (projet des années précédente).