NF05 - AGENDA

Rapport de projet

Thomas de Lachaux – Louis Bichet

Automne 2018

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

Table des matières

Introduction

Le projet consiste à créer un agenda, à l'aide du langage de programmation C, proposant deux modes de fonctionnement : utilisateur et administrateur.

- Le premier mode doit permettre à l'utilisateur d'effectuer les actions suivantes :
  - créer un rendez-vous
  - modifier et supprimer un rendez-vous
  - consulter l'agenda sous une forme définie par l'administrateur (par jour, semaine, mois, etc.)
- Quant au second mode, l'administrateur pourra effectuer les actions suivantes :
  - ajouter, modifier ou supprimer un utilisateur
  - ajouter, modifier ou supprimer les "journées spéciales"
  - changer le mode d'affichage de l'agenda (par jour, semaine, mois, etc.)

Ce rapport a pour but de mettre en exergue notre méthode d'approche du sujet, les algorithmes élaborés puis mis en pratique à l'aide du langage C, ainsi que l'ensemble de notre démarche comprenant les outils développés, les problèmes rencontrés, les choix faits, etc.

Nous établirons notre rapport selon le plan suivant :

1. Les algorithmes utilisés et leur fonctionnement
2. Problèmes rencontrés et solutions adoptées
3. Mode d'emploi de l'Agenda
4. Outils
5. Conclusion et documentation

Notre projet est disponible à cette adresse sur le site GitHub : https://github.com/HerelAdrastel/NF05/tree/master

1. Les algorithmes utilisés et leur fonctionnement

1.1. Menus

Les menus de notre agenda sont tous similaires : on propose plusieurs choix à l'utilisateur. Ce dernier doit entrer un nombre correspondant à son souhait. A partir du choix, on se réfère à des conditions « Si » permettant de lancer le sous-algorithme correspondant au numéro entré par l'utilisateur. Afin de ne pas sortir du menu par inadvertance, on utilise une boucle « Tant que » de sécurité définie par le numéro permettant de quitter le menu (proposé dans les choix). La structure des menus est donc équivalente à :

Algorithme : Menus

Variables : choix, quitter : entiers

Instructions :

1. quitter ← 5 // ici, 5 est donné à titre d'exemple
2. Tant que (choix != quitter)
3. Ecrire (« Votre choix : ... »!) // Dans les « ... » on propose plusieurs choix numérotés.
4. Lire (Clavier ! Choix)
5. Si (choix = 1) alors choix1 (paramètres d'entrée ! Paramètres de sortie)
6. …
7. Si (choix = 4) alors choix4 (paramètres d'entrée ! Paramètres de sortie)
8. Fin Tant que
9. Fin Menus

1.2. Stockage des rendez-vous

Afin de stocker les rendez-vous entrés par l'utilisateur, nous avons utilisé les listes doublement chaînées, afin d'éviter les tableaux et parce que celles-ci correspondaient à la nécessité de stocker des rendez-vous, d'afficher ceux-ci dans un ordre défini, ainsi que de gérer la mémoire plus précisément, ce que ne permet pas un tableau. Notre fonction permettant d'entrer un rendez-vous est équivalente à la fonction « add_first() » vue en cours.

1.3. Stockage des utilisateurs

Le stockage des utilisateurs se fait par l'intermédiaire d'une liste doublement chaînée. En effet, dès qu'un administrateur entre le nom d'un

nouvel utilisateur, on appelle un sous-algorithme qui va joindre le nouvel utilisateur à la liste. Tout comme le stockage des rendez-vous, la fonction permettant d'ajouter un utilisateur en tête de chaîne est similaire à la fonction add_first(). De plus, elle permet de vérifier les cas spéciaux, notamment si la chaîne est vide.

1.4. Affichage des rendez-vous, journées spéciales

L'affichage des rendez-vous se fait à l'aide d'une boucle « Pour » permettant de parcourir la liste chaînée. La condition d'arrêt n'est pas le pointeur dont la valeur est « NULL », mais la taille de la chaîne, puisque l'on utilise ici une boucle « Pour » et non une boucle « Tant que ».

1.5. Suppression d'un rendez-vous, utilisateur

La suppression d'un rendez-vous ou d'un utilisateur se fait par l'intermédiaire d'un sous-algorithme permettant d'appeler l'élément par la recherche du nom de l'utilisateur ou le numéro du rendez-vous. On modifie les destinations des pointeurs, afin d'enlever toute trace de la case à supprimer de la chaîne, puis on libère l'espace mémoire de la case supprimée.

2. Problèmes rencontrés et solutions adoptées

2.1. Accents

2.1.1. Problématique

Par défaut, sur Windows, les consoles sont encodées avec la page de code 850. Cela correspond à la table ASCII Etendue.

Pour vérifier la page de code sur votre console, taper « chcp ».

Le code source est encodé en UTF-8 (chcp 65001). Par défaut, sur MacOS et Linux, les consoles sont encodées en UTF-8. Dans ce cas-là, on aura un affichage correct.

Mais sur Windows, le code « printf("À bientôt !"); » affichera « ├Ç bient┼┤t ! ».

On peut résoudre ce problème en exécutant l'instruction suivante : *printf("\267 bient\223t !");*.

« \267 » correspond au caractère `À` dans la table ASCII étendu en base octale. De même pour « \223 » qui correspond au « ô ». Cependant, cette phrase n'est pas très lisible et peut gêner lors du développement de l'application.

2.1.2. Résolution

Pour résoudre ce problème, nous avons développé un script en Python. Cet algorithme agira en fonction du premier argument donné :

– Si le premier argument est « encode », le script convertit tous les caractères spéciaux dans les fichiers sources en caractères octaux de la base ASCII.

– Si le premier argument est « decode », le script convertit tous les caractères octaux de la base ASCII dans les fichiers sources en caractères spéciaux.

Ainsi, quand nous lançons la compilation, l'ordinateur exécute d'abord le script en mode d'encodage, puis compile le programme, et enfin exécute le script de décodage.

2.1.3. Exécuter le script

Le script est situé dans le dossier « utils/accents ». Il est disponible au format « .py » et « .exe ».

Tout d'abord, il vous faut ouvrir un terminal et bien vérifier que le dossier courant du terminal soit celui du script ; des exceptions seront levées dans le cas contraire.

Exécuter le script à partir de l'exécutable

Prérequis : avoir un système d'exploitation Windows.

Pour encoder les fichiers sources, tapez « accents.exe encode » - Pour décoder les fichiers sources, tapez « accents.exe decode ».

Exécuter le script à partir des sources

Prérequis : Python 3.x (Testé avec Python 3.7).

Pour encoder les fichiers sources, tapez « python accents.py encode ».

Pour décoder les fichiers sources, tapez « python accents.py decode ».

Convertir le « .py » en « .exe »

Prérequis : Python 3.x (Testé avec Python 3.7).

Il vous faudra aussi *PyInstaller* installable via la commande « pip install pyinstaller ».

Entrez ensuite « pyinstaller accents.spec ».

2.2. Doxystyle

2.2.1. Problématique

En utilisant *Doxygen*, nous avons constaté que le design du site est assez vieux.

2.2.2. Résolution

Cependant nous avons pu résoudre ce problème, car *Doxygen* offre la possibilité de modifier les entêtes HTML, et d'ajouter des fichiers supplémentaires, tels que des feuilles de styles, des scripts ou des images.

Le développement est principalement basé sur le fichier « styles.less ». Pour nous aider dans la modification du design, nous sommes inspirés de celui proposé par *Bootstrap*.

Nous avons modifié les propriétés de nombreuses classes du code. À chaque modification, il fallait suffixer la ligne par « !important » pour être sûr que notre propriété soit prioritaire par rapport à celle par défaut. Au lieu de le faire manuellement, nous avons utilisé le language *LESS*. La commande « lessc » compile « styles.less » vers « styles.css ».

Quand il était impossible de styliser ou sélectionner un élément avec des règles précises en CSS, nous avons utilisé du javascript avec le framework jQuery (déjà inclus par Doxygen). Par exemple, pour ajouter des graphiques dans le README.md rapidement, nous avons utilisé [mermaid]. Il a fallu utiliser jQuery pour modifier le DOM et implémenter correctement cette bibliothèque.

2.2.3. Compiler le LESS en CSS

Prérequis : Node.js.

Installer *less* via la commande « npm install -g less ».

Pour compiler « styles.less », ouvrez un terminal, placez-vous dans le dossier *Doxystyle* et entrez « lessc styles.less styles.css ».

2.2.4. Compiler le projet à partir des sources

Le code principal est développé en *C99*. Pour le compiler, il vous faudra télécharger *CMake* ainsi qu'un compilateur C.

2.3. Collision des rendez-vous

Lorsque nous avons réalisé la fonction permettant d'entrer et de stocker des rendez-vous, nous nous sommes aperçu que nous n'avions aucune sécurité concernant la collision, la superposition des rendez-vous, de telle sorte que ceux-ci pouvaient se chevaucher. C'est pourquoi nous avons ajouté une vérification concernant la date et la durée des rendez-vous lors de l'entrée d'un rendez-vous.

2.3.1. Concept

On compare donc le nouveau rendez-vous à tous les autres rendez-vous de l'utilisateur sur deux critères : la date puis l'heure et la durée. Le but est de vérifier que le jour n'est pas le même, puis si les heures se chevauchent ou non. S'il y a chevauchement, le programme renvoie un message et un menu permettant de changer le nouveau rendez-vous ou de modifier l'ancien rendez-vous collisionnant. Le fonctionnement de l'algorithme est le suivant.

2.3.2. Sous-algorithme : collision

Paramètres d'entrée : nombredeRDV : entier ; rdvUser : article de type listeRDV.

Paramètres de sortie : 0 ou 1.

Variables :

- i, choix : entier
- current : article de type rendez-vous

Instructions :

1. choix = 0
2. Pour i allant de 0 à nombreDeRDV
3. Si (….) // vérification si collision
4. Ecrire (« Vous avez déjà un rendez-vous » !)
5. Ecrire (« Que voulez vous faire ? »!)

```
6.      Tant que (choix < 1 OU choix > 3)
7.          Ecrire ( « 1. Modifier le nouveau rendez-vous » ! )
8.          Ecrire ( « 2. Modifier l’ancien rendez-vous » ! )
9.          Ecrire ( « 3. Supprimer l’ancien rendez-vous »! )
10.         Lire ( Clavier ! Choix )
11.         Fin Tant que
12.         Si (choix = 1) alors…
13.         Si (choix = 2) alors…
14.         Si (choix = 3) alors…
15.         Fin Si
16.     Fin Si
17.     Fin Collision
```

3. Mode d'emploi

3.1. Ouverture de l’agenda

Trois exécutables sont disponibles à la racine du projet :

- *Agenda.exe* : c’est le programme par défaut. Il permet notamment d’afficher les accents des consoles récentes de Windows (telles que *PowerShell* ou *Cmder*).
- *Agenda-encode.exe* : il s’agit du même programme qu’*Agenda.exe* à la différence que celui-ci permet l’affichage des accents sur la console Windows (*cmd.exe*).
- *GUI.exe* : c’est l’interface graphique.

Choisissez donc l’exécutable que vous souhaitez lancer selon votre console ou votre préférence graphique. Vous accédez donc à l’Agenda.

3.2. Menu principal

Vous arrivez dans un menu vous demandant si vous êtes un utilisateur ou un adminstrateur. Entrez le numéro correspondant à votre statut et votre profil. Un numéro permettant de quitter le programme est également disponible.

Remarque : Si vous êtes un utilisateur mais que vous ne trouvez pas votre nom dans la liste des choix, veuillez contacter un administrateur afin qu’il rentre votre profil dans l’agenda.

3.3. Profil utilisateur

En tant qu’utilisateur, vos actions n’impactent que votre agenda personnel. Vous avez donc la possibilité de réaliser les actions suivantes :

1. Ajouter un rendez-vous
2. Modifier / supprimer un rendez-vous
3. Afficher les rendez-vous selon le mode d’affichage (choisi par l’administrateur)
4. Afficher les journées spéciales

Lorsque vous entrez dans votre profil, le programme affiche un nouveau menu permettant de choisir l’action que vous souhaiter effectuer. Entrez le numéro correspondant pour l’action choisie.

3.3.1. Ajouter un rendez-vous

Lorsque vous entrez dans cette option, le programme vous propose trois choix permettant de définir le jour du rendez-vous. Entrez le numéro correspondant. Vous pourrez ensuite entrer tous les détails de ce rendez-vous, à savoir :

1. L’heure
2. La durée
3. Le nom
4. Le lieu
5. Les personnes présentes

Cela fait, vous serez redirigés vers le menu principal utilisateur.

3.3.2. Lister les rendez-vous

Ce menu vous permettra de lister les rendez-vous selon le mode d’affichage choisi par l’administrateur. L’Agenda va vous proposer le contenu qu’il doit afficher selon la date des rendez-vous. Vous pourrez par la suite avoir accès à un nouveau menu permettant de :

1. Modifier un rendez-vous
2. Supprimer un rendez-vous
3. Sauvegarder un rendez-vous
4. Retourner au menu principal

Tout comme les autres menus, il vous faudra entrer le numéro correspondant à votre souhait.

3.4. Profil Administrateur

En tant qu’administrateur vous aurez accès à différentes fonctionnalités permettant de gérer l’Agenda, à savoir :

1. Ajouter un utilisateur
2. Lister les utilisateurs
3. Ajouter un jour spécial
4. Lister les jours spéciaux

Le statut administrateur entraîne certaines responsabilité, il faut donc veiller à la bonne utilisation des paramètres qui sont à votre disposition.

3.4.1. Ajouter un utilisateur

Cette option vous permet d’ajouter un utilisateur en entrant son nom. Il sera ensuite disponible dans la liste des utilisateurs proposée dans le menu principal de l’agenda.

3.4.2. Lister les utilisateurs

Cette option permet de lister les utilisateurs et d’effectuer deux actions sur ceux-ci : renommer l’utilisateur ou le supprimer. Vous n’avez qu’à entrer le numéro correspondant pour lancer l’action.

3.4.3. Ajouter une journée spéciale

Ce sous-menu vous octroie la possibilité d’ajouter une journée dite « spéciale » pour tous les utilisateurs. Vous pouvez donc nommer et choisir la date de celle-ci.

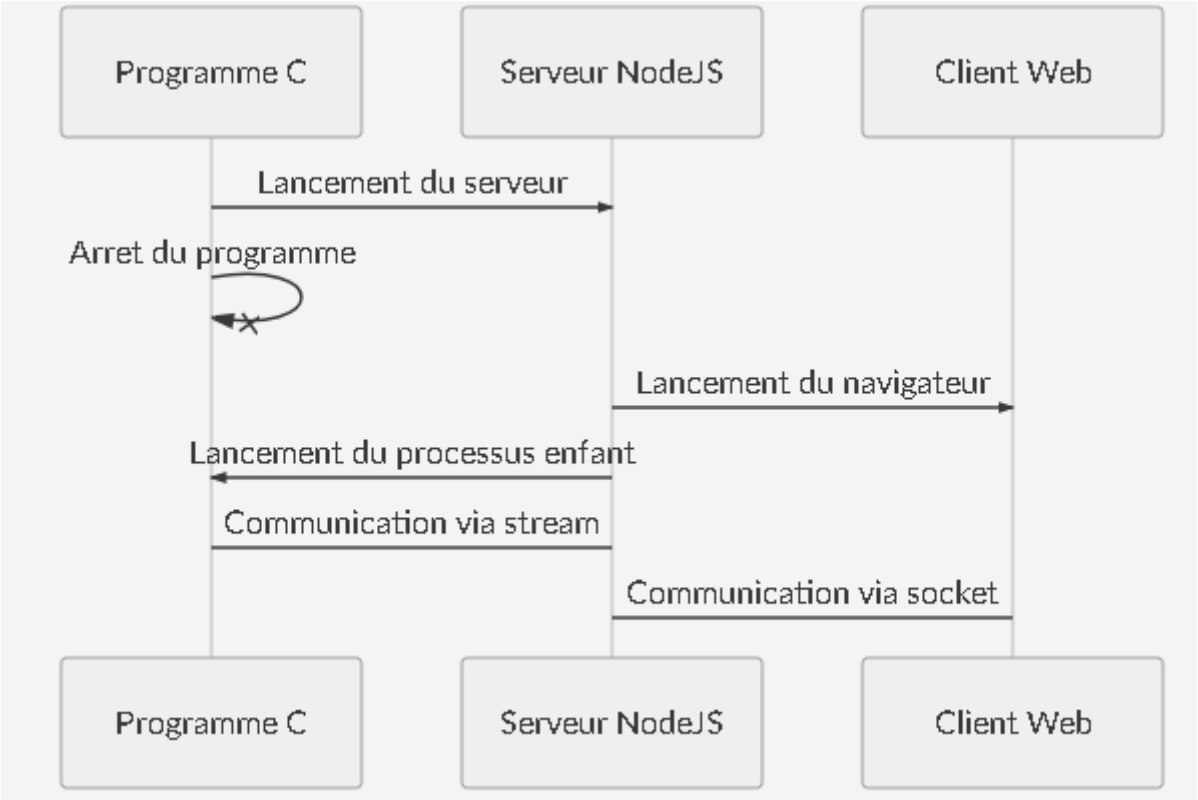
3.4.4. Lister les journées spéciales

Vous pouvez lister les journées spéciales entrées dans le menu précédemment cité, dans le but de faire leur inventaire, de les modifier ou encore de les supprimer.

4. Outils

4.1. Interface utilisateur

Une interface utilisateur est disponible pour augmenter l'ergonomie du programme. Voici un schéma expliquant le fonctionnement de l'interface graphique.



Le serveur *NodeJS* joue le rôle de messager entre le programme et le client Web. Le programme est exécuté avec des arguments différents d'un lancement sous console, pour avoir une sortie formatée en HTML. La sortie est récupérée par *NodeJS* qui l'envoie au client web via des sockets.

Le chemin est effectué dans le sens inverse quand un l'utilisateur appuie sur un bouton sur le client Web, ou lorsqu'un texte est entré.

Exécuter depuis les sources

Prérequis : NodeJS

Commandes pour installer les outils nécessaires :

1. npm install -g yarn pkg nodemon
2. yarn

Pour exécuter l'interface : « yarn start »

Pour exécuter l'interface en live-reload : « yarn dev »

Pour compiler l'interface pour Windows : « yarn builddev »

Pour compiler l'interface pour Windows, Mac et Linux : « yarn build »

4.2. Git

Dans le cadre de notre projet collaboratif, nous avons souhaité optimiser notre temps de travail. Pour cela, il nous fallait pouvoir tester des bouts de codes dans notre code principal et partager celui-ci rapidement s'il était fonctionnel. Nous avons donc utilisé un outil de travail collaboratif, Git, et plus précisément GitHub.

Cette plateforme permet de créer des branches parallèles à partir d'une branche principale, puis de fusionner celles-ci afin de recréer une branche principale. Cela permet entre autres de se partager les tâches plus efficacement et de pouvoir travailler chacun de son côté sans être dépendant de son binôme.

Ainsi, pour chaque nouvel ajout de fonctionnalités, nous commencions par créer une branche parallèle où nous essayions d'insérer un code fonctionnel, puis nous fusionnions avec la branche principale. Ci-dessous se trouve le lien de notre projet disponible sur GitHub.

GitHub : https://github.com/HerelAdrastel/NF05/tree/master

4.3. Clion

Afin de coder plus efficacement et d'avoir le même outil de travail, nous avons tous deux installé Clion gratuitement grâce aux licences étudiantes, ce qui nous a permis d'être plus rapides en terme de transfert et de codage.

4.4. Outils externes utilisés

4.4.1. Interface graphique

- **[NodeJS]** : cet outil sert à lancer un serveur en JavaScript.
- **[ExpressJS]** : permet de gérer plus facilement les requêtes et réponses du serveur.
- **[Yarn]** : permet de gérer plus facilement les dépendances que NPM.
- **[socket.io]** : permet de communiquer avec le client web.
- **[opn]** : permet de lancer le navigateur par défaut lors de lancement de l'Agenda avec l'interface graphique.
- **[nodemon]** : permet de « live-reload » le code du serveur.
- **[pkg]** : permet de portabiliser le code sur Windows, Mac et Linux.
- **[Bootstrap]** : permet de simplifier la stylisation de l'interface.

- **[jQuery]** : permet de faire des retouches textes rapidement.

4.4.2. Accents

- **[Python 3.7]** : permet de coder un petit script rapidement (ici les accents).
- **[PyInstaller]** : permet de portabiliser le code sous Windows.

4.4.3. Doxystyle

- **[LESS] :** Pour éviter la redondance du CSS.

Conclusion

Ce projet fut assez conséquent en terme de travail et d'implication, du fait de sa complexité et de nos connaissances sur le sujet. Cependant, Thomas avait déjà eu l'occasion de coder en langage C, ce qui nous a permis de nous diviser les tâches équitablement. Grâce à cela, nous avons eu la possibilité de développer des outils en parallèle, afin de mettre en valeur notre code en C et obtenir un visuel plus adapté (accents et interface graphique). Il est à noter que nous avons commencé le projet très tôt, ce qui nous a permis de répartir les tâches régulièrement et nous octroyer des marges conséquentes pour les jalons. Les connaissances de Thomas sur la programmation et sa facilité d'adaptation ont été d'une grande aide pour réaliser le projet.

Les choix faits lors de la réalisation de l'Agenda pourraient être discutés, mais ils reposent sur plusieurs critères personnels, tels que nos connaissances, le résultat attendu, le temps de réalisation imposé, etc. Enfin, le travail collaboratif a été grandement facilité par l'outil GitHub, qui nous a permis de travailler sur des tâches différentes tout en utilisant le code préalablement établi. Le lien de notre projet sur GitHub se trouve ci-dessous :

https://github.com/HerelAdrastel/NF05/tree/master

Vous trouverez à cette adresse la documentation du code de l'Agenda réalisée à l'aide du Doxygen, mais dont l'interface a été quelque peu modifiée.