

Amir Talal

Hadj Rabah Radouane

Nisar Rehane

PRE-ING 1, MI 2 - B

Projet sélectionné : ChenYI-TECH

L'équipe du projet " ChenYI-TECH " est composée de 3 personnes : Hadj Rabah Radouane, Nisar Rehane et Amir Talal, tous en Pré-Ing 1 en MI 2 . "ChenYI-TECH" est une application permettant la gestion d'un refuge pour animaux. Celui ci est 100% interactif dans le terminal afin de s'occuper des pensionnaires (chats, chiens, hamsters et autruches). L'application propose plusieurs options pour gérer le refuge de manière optimal, simple et efficace.

Les problèmes rencontrés :

1. Difficultés de codage par rapport à la contrainte de ne pas utiliser la librairie stdio.h . Nous avons dû créer d'autres fonctions en parallèle pour " créer nous-même la librairie ".
2. Utilisation de "Git".
3. Erreur de compilation (oubli de ; , faute d'orthographe de variable / fonctions, suppression de boucles, conditions, incrémentation , mauvaise appel de variable...
4. Chaque membre a développé ses fonctions de son côté et ont rencontré des difficultés
5. L'affichage des différents émojis et des accents " é " sur le terminal Windows ne s'affichait pas correctement lors de l'exécution de l'application.
6. Radouane du revoir le fonctionnement des méthodes printf et scanf dans la fonction Fichier_vers_Tab
7. Plusieurs corrections ont été apportées sur une même fonction suite à la découverte d'erreurs quelques temps après (ajouter.c , menu.c par exemple).

Leurs solutions :

1. Concertation en réunion de tous les membres du projet afin d'apporter notre avis sur les fonctions de chacun.
2. Corrections collectives des fonctions posant problèmes.
3. Débug jusqu'à réussite.
4. Branchements conditionnels cas par cas
5. Aide d'internet pour les problèmes les plus complexes.
6. Robustesse de chaque fonctions donnant une performance optimale

Les résultats obtenus :

1. La compilation s'effectue correctement
2. Chaque membre de l'équipe a petit a petit appris à utiliser la plateforme "Git"
3. Affichage réussi de l'interface (émojis, accent des caractères, informations des animaux)
4. Toutes les fonctionnalités développées fonctionnent complètement
5. Test de plusieurs cas de possibilités de crash du projet. Nous avons corrigé les fonctions pour éviter le maximum de crash possible, peu importe les circonstances. Ainsi, le projet ne comporte aucun crash majeur et mineur.

Organisation de l'équipe :

L'équipe a choisi GitHub comme plateforme d'hébergement et de collaboration pour leur projet. Pour communiquer, un serveur de discussion a été créé sur la plateforme " Discord " . Radouane, Talal et Rehane utilisent Windows 10 et 11. Pour compiler leur programme, les 3 membres ont utilisés " GCC " via " MSYS2 UCRT64 " disponible sur Windows Powershell.

Répartition des tâches :

Semaine / équipe	Radouane	Rehane	Talal
Semaine 1 (07/04 - 13/04)	Création de la structure Animal Création du main.c	Création de la fonction adopter.c	Création de la fonction ajouter.c
Semaine 2 (14 / 04 – 20/04)	Création de la fonction recherche.c	Amélioration de la fonction adopter.c	Amélioration de la fonction ajouter.c
Semaine 3 (21/04 - 27/04)	Création de la fonction Tab_vers_Fichiers	Création de la fonctions Fichier_vers_Tab	Création de la fonctions Fichier_vers_Tab Correction de l’affichage
Semaine 4 (28/04 - 04/05)	Création de la variante day_Clean, Debug git et Visual Studio Code	Création de la variante day_Food, Debug git et Visual Studio Code	Création de la variante day_Food Debug git et Visual Studio Code
Semaine 5 (05/05 - 11/05)	Vidéo de démonstration Création de la fonction robuste.c, inventaire.c, Menu.c, fonctions_strings.c Update de la fonction recherche.c Robustesse Update main.c	Readme Rapport du projet Débug git et Visual Studio Code Update de adopter.c Robustesse Esthétique Ajustement finaux	Update de la fonction ajouter.c Commentaires, nettoyage, lisibilité et esthétique du code Makefile Robustesse Esthétique Ajustement finaux

	Vérifications et corrections des fonctions Ajustement finaux		
--	--	--	--