Rapport projet ChenYl-Tech

Brnak Noé

Lopes Diego

Moro Chloé

Organisation :

Nous avons décidé de travailler sur le projet "ChenYl Tech". Nous nous sommes partagé les tâches pour avancer le plus rapidement possible. En quelques mots, nous avons tous participés à la réalisation du projet.

Choix importants :

- Nous avons décomposés nôtre JEU en 3 parties, une principale qui contient toutes les fonctions et les structures (à cause de problèmes fréquents, nous n'avons pas réussi à faire plusieurs .h et .c comme demandés), un .h qui contient les bibliothèques nécessaires, et un fichier .txt qui regroupe la totalité des pensionnaires.

- Nous avions choisi un tri fusion pour ranger les différentes espèces dans l'ordre décroissant (l'une des variantes) car il s’agit de l’un des tris les plus rapides et dont sa fonction est robuste et stable. Malheureusement, la fonction qui contient ce tri n'a pas été utilisée par manque de temps.

- Nous avions également créé des tableaux pour chaque espèce et pour chaque animal pour classer les animaux en fonction de leurs caractéristiques (âge, espèce, nom). Cette fonction devait fonctionner en parallèle avec le tri fusion.

- Nous avons utilisé des boucles pour les conditions dans presque toutes les fonctions pour demander à l’utilisateur de saisir les caractéristiques de l’animal, d’adopter ou d’accueillir l’animal dans le chenil.

- Nous avons utilisé des numéros pour chaque espèce (chien = 0, chat = 1, hamster = 2, autruche = 3) pour les identifier plus facilement.

Limite :

- Nous avons rencontré des difficultés lors du projet ChenYl Tech ; tout d'abord, le manque d'organisation nous a causé des problèmes de temps.

- Nous avons rencontré des difficultés sur la structure (et l'énumération). En effet, comme nos parties de codes sont dépendantes de la partie sur la structure (et énumération), des problèmes de types nous ont fait ralentir le projet. Ce problème a fini par être réglé.

- Nous avons également rencontrés des difficultés sur la partie où on doit trier des animaux par espèce. En effet, le choix du tri n’est pas facile et le tri fusion n’est pas évident à coder. Nous avons effectué des recherches sur Internet pour le faire sur un de nos codes en C.

- La variante dayfood et la variante ordredécroissant, qui sont dépendantes des tableaux, n'ont pas pu être raccordées au code principal par manque de temps.

- Nous avons mis toutes les fonctions et les structures sur une seule et même page (455 lignes), ce qui fait beaucoup et n'est pas forcément très lisible.

Conclusion :

Pour conclure, faire ce projet nous a donné du fil à retordre, mais principalement à cause du manque de temps (faute de nôtre part), malgré ça nous avons appris beaucoup en "peu de temps", par exemple le fait de travailler sur plusieurs fichiers et de les relier entres eux. Pour finir, le JEU peut compiler sur Linux et Windows (voir readme pour plus d'informations), même s'il manque une grande partie du code : le dayfood, l'ordredécroissant, les tableaux, les fonctions pour revenir en arrière dans le jeu et peut-être aussi un peu de couleur et d'esthétisme.

Bonus :

Nous avons créé le menu principal pour l’interface du jeu afin de rendre notre projet plus animé et joli pour l'utilisateur.