Polytech-Sophia mai 2019
Elec3 Durée : 1,5 mois

Projet C : Jeu Memory

Avant de commencer : la qualité des commentaires, avec notamment la présence des antécédents, des conséquents, des invariants de boucle, les rôles de chacune des fonctions, ainsi que les noms donnés aux variables, l'emploi à bon escient des majuscules et la bonne indentation rentreront pour une part importante dans l'appréciation du travail. Ce projet doit permettre de montrer votre autonomie et votre compréhension tant dans la conception du programme que dans sa réalisation. Enfin, si les codes de plusieurs projets se trouvent être identiques, ou être copiés depuis le web, tous les projets concernés seront immédiatement sanctionnés par un zéro.

1 Règle du jeu

Le jeu *Memory* est un jeu de cartes (voir le site `http://www.jeu-test-ma-memoire.com/jeux-de-memory`). Les cartes sont présentées par paires et sont disposées faces cachées en début de jeu. Lors d'un coup, le joueur sélectionne deux cartes et les retourne. Si les cartes sont identiques, elles restent visibles, sinon, elles sont retournées face cachées. Le jeu s'arrête lorsque toutes les paires sont visibles.

2 Travail à faire

Votre travail consiste à écrire un programme en C qui met en œuvre ce jeu de mémoire. Votre programme sera doté d'une interface graphique développée à l'aide, obligatoirement, de *libsx*. La face et le dos des cartes sont laissés à voir choix.

La taille de la grille devra être variable, et laissée au choix de l'utilisateur.

Votre programme devra identifier les joueurs et proposer une gestion de scores. Les 10 (par exemple) meilleurs scores avec les noms des joueurs seront mémorisés dans un fichier et pourront être visualisés à tout moment.

3 Remise du projet

Votre projet est à faire en trinôme et à rendre au plus tard le :

samedi 10 juin 2019 à 23h - aucun délai ne sera accordé –

1. vous enverrez à `vg@unice.fr` une archive `memory-n1-n2-n3.tar.gz` avec `n1`, `n2` et `n3` les noms des trois étudiants du groupe.

L'archive devra contenir :

- les fichiers sources (`.c` et `.h`) correctement documentés (chaque fonction doit avoir un commentaire, les invariants de boucle doivent être marqués), indenté, et codé (les noms de variables explicites, éviter les trop longues fonctions) ;
- le fichier `Makefile`
- un fichier `rapport` exclusivement au format pdf et décrivant le fonctionnement général du programme, les algorithmes, ainsi que les choix de programmation ;

On rappelle que le code source doit être correctement indenté, commenté et qu'il doit être clair et lisible. Vous devrez utiliser au mieux les propriétés vues en cours et TD du langage C. La compilation avec les options `-Wall -pedantic` ne doit pas donner de *warning*.

Il ne devrait pas être nécessaire de rappeler que le travail doit être personnel et que toute ressemblance entre des projets sera sévèrement sanctionnée. Mieux vaut donc un projet modeste personnel qu'un très beau projet copié.

Il ne devrait d'ailleurs pas être nécessaire de rappeler que le but premier d'un tel projet est de vous faire progresser en programmation en vous confrontant à une expérience de plus grande envergure qu'un simple TD.