

## Projet C : Jeu Memory

---

### 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 votre choix. 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 binôme et à rendre au plus tard le :

**samedi 30 juin 2018 à 23h – aucun délai ne sera accordé –**

- vous enverrez à vos enseignants [dmei2006@xidian.edu.cn](mailto:dmei2006@xidian.edu.cn), [jyliu@xidian.edu.cn](mailto:jyliu@xidian.edu.cn) une [archive memory-n1-n2.tar.gz](#) avec **n1** et **n2** sont les deux derniers chiffres de numéro d'étudiant.
- vous enverrez à [vg@unice.fr](mailto:vg@unice.fr) un rapport en français de 2 pages (pas moins, pas plus) exclusivement au format pdf (sinon, il ne sera pas lu) qui décrit votre projet.

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** en chinois 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.