

## Projet C++ : Jeu Memory

---

### 1 Règle du jeu

Le jeu *Memory* est un jeu de cartes. 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 :

1. un programme C++ qui met en œuvre ce jeu de mémoire. Votre programme sera doté d'une interface graphique développée à l'aide (obligatoirement) de *gtkmm* (cf td12). La face et le dos des cartes sont laissés à voir 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 une base de données mysql (cf td13), et pourront être visualisés à tout moment. La BD sera accessible sur le serveur <http://bddetu.polytech.unice.fr>. Vous devrez également proposer des extensions (voir plus loin).
2. un rapport de projet qui décrit votre application et en particulier l'organisation des classes (diagramme de classes UML attendu) et les patrons de conception que vous aurez utilisés.

### 3 Extensions

Vous devez ajouter différentes extensions/variantes au jeu de Memory afin de le rendre plus difficile. En voici quelques unes et en vous devez en proposer d'autres de votre invention.

#### Transformation des images

À chaque fois qu'une carte est retournée, l'image sera transformée à son prochain affichage, par exemple, une dégradation qui la rend moins visible.

On pourra choisir d'autres transformations pour toute la partie ou bien changer de transformation à chaque nouvelle sélection d'une carte, ou encore choisir la transformation de façon aléatoire.

#### Déplacement des cartes

À chaque paire manquée, toutes les cartes sont déplacées (par exemple, d'un cran vers la droite puis de haut en bas, ...), ou bien les deux cartes sélectionnées sont inversées, ou encore aléatoirement.

#### Et vous ?

Que proposez-vous ? Si vous manquez d'idées, en voici encore quelques-unes :

- Plutôt que des paires, on doit retrouver des triplets, des quadruplets... de cartes.
- Utiliser un chronomètre pour limiter la partie dans le temps ;
- Plutôt que de faire évoluer les transformations précédentes à chaque paire manquée, on pourrait utiliser le chronomètre et appliquer les transformations à intervalle de temps constant (toutes les 20 secondes, par exemple).
- Appliquer des pénalités lorsque le joueur manque des paires.
- ...

### 4 Remise du projet

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

**dimanche 13 mai 2018 à 23h – aucun délai ne sera accordé –**

sous forme d'une archive **num-nom1-nom2-nom3.tar.gz** que vous déposerez sur le site jalon de l'université comme au premier semestre. Attention, cette fois-ci, de bien respecter les consignes.

Votre archive contiendra :

- uniquement les fichiers sources (**.cpp** et **.hpp**) ;
- le fichier **Makefile** permettant la compilation sans erreur du programme ;
- le fichier **rapport.pdf**, exclusivement au format pdf (sinon, il ne sera pas lu),
- les images des cartes.

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++.

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.