

# † Travaux Pratiques / Projet *Le jeu du Memory !!!* † † †



## Principe du TP-Projet :

Comme son nom l'indique, ce sujet fait office à la fois de sujet de TP, mais aussi de sujet de projet. En terme d'organisation, le principe est le suivant :

- \* 3 TP (CLO) + 2 TP (IHS) + 1 TP (AMO)
  - o le premier TP permettant d'initier le projet,
  - o les TP suivants permettant d'avancer les différents modules de ce projet et/ou de vous présenter et/ou d'approfondir certaines notions à mettre en œuvre dans votre projet.
- \* 1 Projet
  - o A rendre avant le 26 mars 2021
- \* Notation
  - o Un livrable, noté à l'issu de chacun des TP.
  - o TP et projet comptent dans la note des ces 6 TP : moyenne des TP sur 13 + projet global sur 7 points.

**V**OUS L' aurez compris, l'objectif de ce projet est de réaliser un version revisitée du jeu de Memory. Ce jeu de société a été édité par Ravensburger pour la première fois en 1959. Si la majorité des règles resteront les mêmes, certaines vont être quelque peu modifiées, et d'autres ajoutées ...

## I - Descriptif & règles du jeu

### Les règles du jeu classique :

Le jeu se compose de paires de cartes portant des illustrations identiques. L'ensemble des cartes est mélangé, puis étalé face contre table. À son tour, chaque joueur (entre 2 et 8) retourne deux cartes de son choix. S'il découvre deux cartes identiques, il les ramasse et les conserve, ce cas de figure lui permettant également de rejouer. À l'inverse, si les cartes ne sont pas identiques, il les retourne faces cachées à leur emplacement de départ.

Le jeu se termine quand toutes les paires de cartes ont été découvertes et ramassées. Le gagnant est le joueur qui possède le plus de paires.

### Les règles de nous ... :

Bien évidemment, nous conserverons les règles de bases décrites ci-dessus. Néanmoins, nous considérerons dans notre application les ajouts/modifications suivants :



#### \* nombre de joueurs :

Nous conserverons ici la possibilité de jouer entre 2 et 8 joueurs. Cependant, pour ceux n'ayant pas tant d'amis, nous ajouterons la possibilité du jeu à 1 joueur, i.e. contre la machine.

#### \* nombre de paires & taille de plateau :

Afin de pouvoir faire varier la difficulté, nous considérerons la possibilité de jouer sur des plateaux de tailles différentes et avec un nombre de paires différentes. Ces 2 informations pourront être choisies au lancement d'une partie.

#### \* les cartes mystères :

Nous prévoyons également la possibilité de choisir l'introduction de 2, 4 ou 6 cartes mystères<sup>1</sup>. Ces cartes auront des fonctionnalités différentes :

- o **révèle-tout** : cette carte, si découverte, affiche l'ensemble des cartes encore présentes sur le plateau pendant 5 secondes. Les cartes ne sont visibles que par le joueur ayant retournée cette carte.
- o **mélange-tout** : cette carte, si découverte, provoque un mélange immédiat et aléatoire des cartes encore présentes sur le plateau. Les cartes sont mélangées pour tous les joueurs.
- o **dans-ta-face** : cette carte, si découverte, provoque la perte de toutes les paires gagnées par le joueur l'ayant retournée. Les paires sont alors repositionnées face cachée sur le plateau.

Les cartes mystères ne sont pas supprimées du plateau. Une fois l'action réalisée, la carte est remplacée face cachée sur ce dernier.

Les prochaines versions de ce futur best-seller apporteront de nouvelles cartes mystères ...

1. En fonction de la taille du plateau choisi.



- \* *le précieux :*

Dans toute partie une amulette magique et unique est présente sur le plateau, sous l'un des cartes. Lorsque cette amulette est découverte par le retournement d'une carte, elle procure au joueur l'ayant découverte un bonus de 5 tours de jeu consécutifs. Si durant ceux-ci le joueur gagne des paires, les tours de jeu supplémentaires sont cumulés. Une fois les tours de jeu bonus terminés, l'amulette est remplacée aléatoirement sous l'une des cartes restantes du plateau.

*II - Architecture logicielle globale*

Nous souhaitons mettre en œuvre une architecture permettant de jouer via différents clients :

- ★ client lourd : on utilisera ici la bibliothèque graphique JavaFx
  - ★ client riche : on utilisera ici VueJs
  - ★ client mobile : on réalisera ici la version Android

Pour ce faire, nous envisageons une architecture telle que montrée sur la figure 1 : Elle sera composée d'éléments distincts :

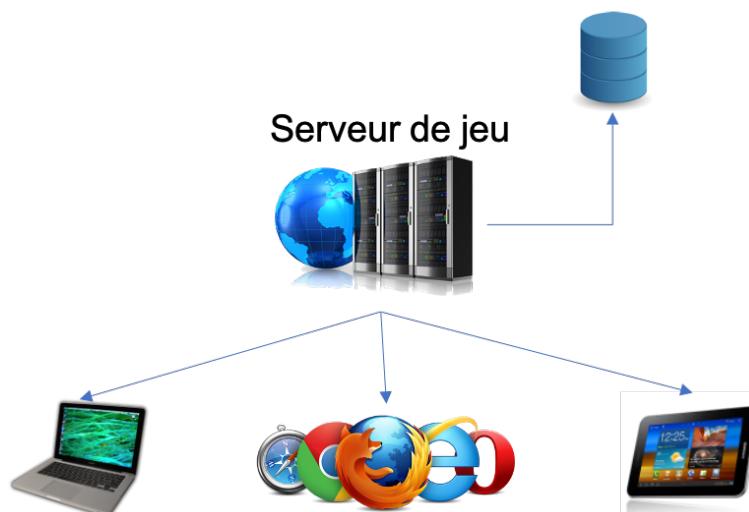


FIGURE 1 – Architecture globale de l'application Memory

- ★ les clients dont nous avons parlé précédemment
  - ★ le serveur de jeu :
    - cet élément sera responsable de la gestion des différentes parties et des différents joueurs.
    - Précisément, le serveur proposera toujours des nouvelles parties auxquelles les joueurs pourront se connecter. Une fois une partie lancée, le serveur sera également responsable du déroulement et de la gestion de la partie. À la fin d'une partie, les résultats seront enregistrés dans une base de données.
    - Un joueur sera modélisé par un nom, un prénom, un mail et un pseudo et devra également être enregistré dans la base afin de pouvoir s'inscrire à une partie.

*III - TP n°1 - Le noyau fonctionnel*

Lors de ce premier TP, vous devez débuter la conception et l'implémentation du noyau fonctionnel de ce petit jeu en réinvestissant l'ensemble des (nombreuses) connaissances vues en cours et TD de CLO.

De plus, lors de l'implémentation, vous mettrez en œuvre, autant que faire se peut, une démarche de développement dirigée par les tests<sup>2</sup>.

Pour l'implémentation, vous mettrez en place une structure de **projet maven en module**, c'est à dire un projet *maven* global composé, pour ce premier TP du projet *maven* correspondant à la couche métier.

2 2 2 2 2 2 2 2 2

#### **Documents à produire :**

Vous devez, avant le lundi 11 janvier/mardi 12 janvier 2021 - 13H30, rendre<sup>3</sup>, dans une *unique archive*<sup>4</sup>, à votre très cher encadrant, le workspace comprenant :

## 2. Tests unitaires (**JUnit**) et tests d'intégration (**JMockit**).

2. Tests unitaires (JUnit) et tests d'intégration
3. Via la plate-forme Moodle de l'école

<sup>5</sup> Le nom de l'archive devra obligatoirement contenir le nom des deux élèves constituant le binôme.



- le projet UML contenant :
    - votre conception actuelle du noyau fonctionnel sous la forme d'un ou plusieurs diagrammes de classes UML.
  - le projet (ou les projets) Maven de votre noyau fonctionnel :
    - Les fichiers sources commentés du logiciel réalisé.
    - Les fichiers sources des tests réalisés
    - L'API (javadoc) de votre noyau fonctionnel

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

BON COURAGE & BONNE ANNÉE À TOUS !!!

