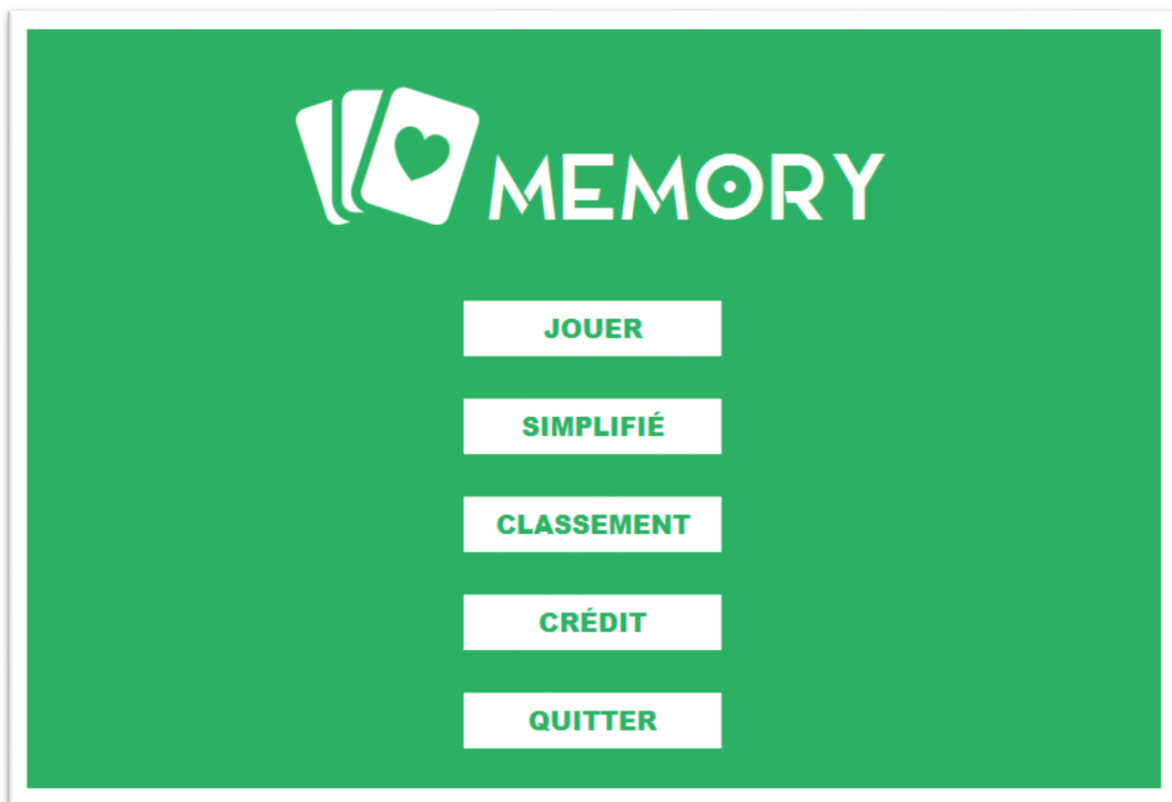


# Projet IHM S2 : MEMORY



Bilel JEGHAM  
Groupe 4A

Juin 2016

# Table des matières

I.	Présentation du Résultat final.....	2
II.	Maquettes : .....	2
	A. Page Accueil .....	2
	B. Page Jeux .....	3
	C. Page Fin de Partie .....	3
	D. Page Classement.....	4
	E. Page Crédit .....	4
III.	Diagramme des classes .....	5
	A. Classe « Fenetre ».....	6
	B. Classe « Controleur » .....	6
	C. Classe « Carte ».....	7
	D. Classe « Score » .....	7
IV.	Algorithme de fin de partie .....	7
V.	Amélioration possible .....	7

# I. Présentation du Résultat final

Le projet du jeu de Memory a pu être complété avec toutes les contraintes dans les consignes.

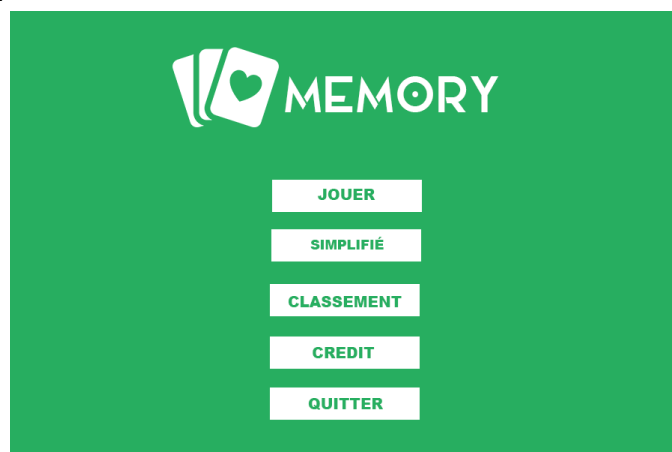
Des fonctionnalités supplémentaires, ont pu être implémentées une fois le développement de la version initiale terminé :

- Enregistrement du score en local dans un fichier
  - Système de classement
- Partage du score sur les réseaux sociaux

## II. Maquettes :

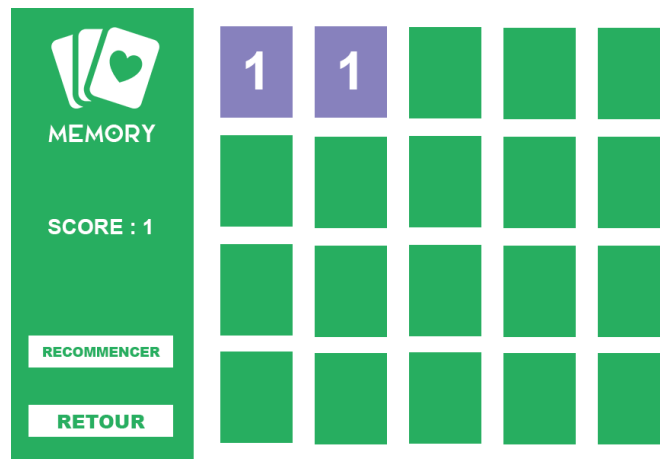
### A. Page Accueil

La page d'accueil comportera 5 boutons redirigeant vers les différentes pages du memory et pouvant quitter l'application.



## B. Page Jeux

La page de jeux comportera un bandeau à gauche contenant toutes les informations utiles (score, bouton pour retourner au menu, recommencer). Au centre, on retrouvera les 20 cartes.



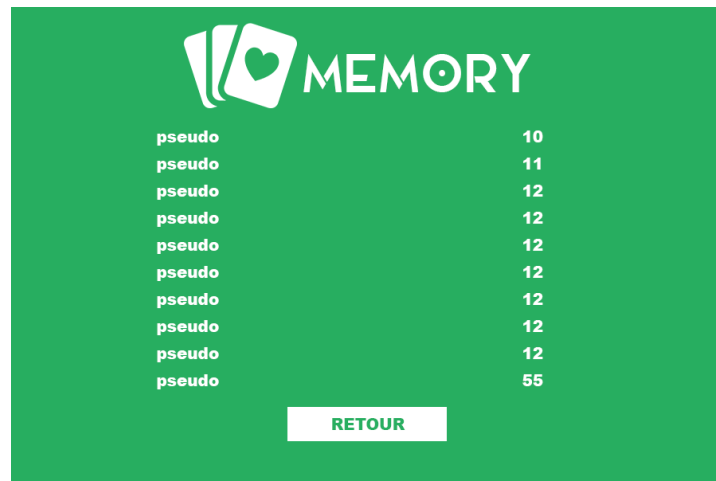
## C. Page Fin de Partie

Une page signifiant la fin de la partie, on pourra y saisir le nom de joueur et partager son score en cliquant sur les logos des réseaux sociaux.



## D. Page Classement

Une page classement comportant les 10 meilleurs scores.



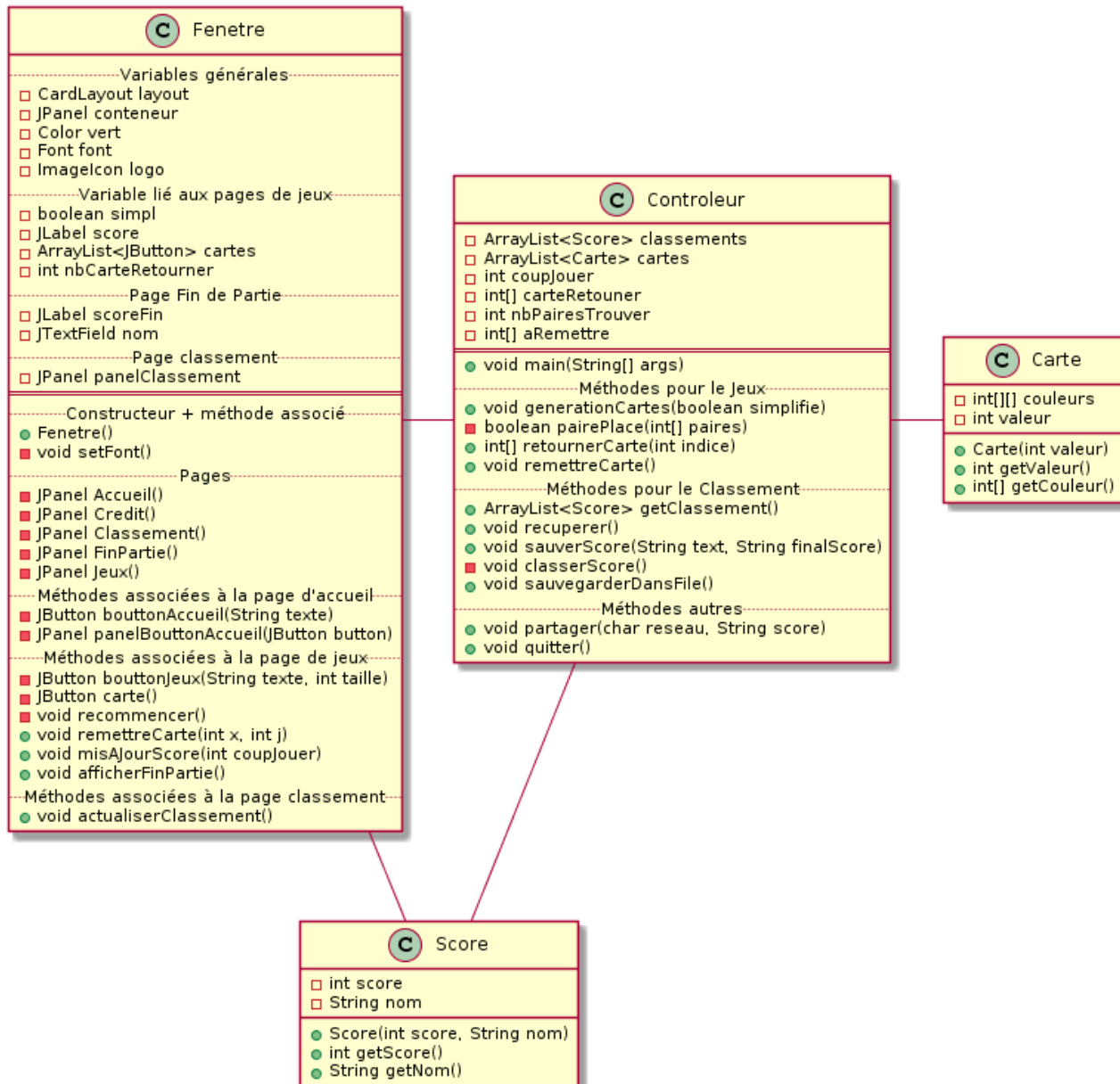
## E. Page Crédit

Une page de crédit avec une phrase de présentation.



### III. Diagramme des classes

Pour le projet, j'ai souhaitée respectée au maximum le MVC en réalisant 3 classes principales :



- Une classe Vue « Fenetre »
- Une classe Contrôleur « Controleur »
- Une classe Métier Cartes

La classe Score est arrivée après pour simplifier le développement, j'ai cherché à créer une variable comportant un 'int' et un 'String'.

Pour toutes informations sur les méthodes, la javadoc est générée dans le projet eclipse.

## A. Classe « Fenetre »

La classe de vue se base sur un CardLayout « layout » comportant un JPanel de chaque page du Memory qui sera introduit dans la variable conteneur.

Pour simplifier le développement, on définit des variables qui vont contenir les éléments réutilisé sur toutes les pages de l'IHM tels que la couleur principale « vert », la police « font », le logo « logo ».

La page « jeux » comporte différentes variables associées qui sont manipulées par les méthodes de la classe :

- Un booléen « simpl » informe si le mode simplifiée est demandé par l'utilisateur.
- Un JLabel « score » comportant le score mis à jour.
- Un ArrayList de JButton « cartes » représentant les cartes du plateau de jeu
- Un nombre entier « nbCarteRetourner » contenant le nombre de carte actuellement retourné.

La page « FinPartie » comporte différentes variables associées qui sont manipulées par les méthodes de la classe :

- Le score de Fin affiché dans un JLabel « scoreFin ».
- Un JTextField pour saisir le nom de l'utilisateur « nom ».

La page « Classement » comporte différentes variables associées qui sont manipulées par les méthodes de la classe :

- Un JPanel qui contient au maximum 10 scores inscrit sur l'application.

## B. Classe « Controleur »

La classe Controleur comporte des variables réservé à la partie du Jeux :

- Un ArrayList de Cartes « cartes » qui comporte toutes les cartes du plateau
- Un entier « coupJouer » qui est utilisé comme score.
- Un tableau d'entier « carteRetourner » qui comporte les indices des cartes qui sont visible, il est utilisé pour déterminée si une paire a été trouvé.
- Un entier « nbPairesTrouver », cette variable est utilisé pour déterminé la fin de partie.
- Un tableau d'entier « aRemettre » qui comporte les indices des cartes visibles qu'il faudra remettre face cachée.

### C. Classe « Carte »

La classe Métier « Carte » comporte :

- La liste des couleurs des paires au format rgb
- La valeur de la carte compris entre 0 et 9 inclus

### D. Classe « Score »

La classe « Score » me permet d'associé un entier « score » à un String « nom ».

## IV. Algorithme de fin de partie

A chaque fois que deux cartes sont retournées, la méthode 'retournerCarte(int indice)' cherche les valeurs sur les cartes 'getValeur()'. Si elles sont identiques alors on incrémente *nbPairesTrouver*. Si toutes les paires sont trouvées (**if** (*nbPairesTrouver* == 10)) alors on affiche la page de fin de partie Fenetre.*afficherFinPartie()*.

## V. Améliorations possibles

- Enregistrement des scores spécifiques pour le mode simplifié.
  - Actuellement, il est mélangé avec le mode de jeu 'normal'.
- Relié le système de score à une base de donnée afin de pouvoir comparer son score avec les autres joueurs.