MASTER 2
Méthodes Informatiques Appliquées à la Gestion des Entreprises

Rapport de projet applications distribuées

CONCEPTION ET REALISATION D'UN JEU DE DES (DICE GAME)

Réalisé par :

- M. EL KHAOUDI Amine
- M. JANATI IDRISSI Mohammed
- Mlle. MOULINE Safaa

Encadré par :

- M. BENALI Khalid

Année universitaire 2016-2017

TABLE DES MATIERES

INTRODUCTION

Nous sommes emmenés dans le cadre de ce projet de faire l'analyse, la conception et la réalisation d'une application de jeu de dés en utilisant les patrons de conception vus durant le cours de conception d'applications distribuées.

1. CAHIER DES CHARGES

1.1. Objectifs du projet

L'objectif de ce projet est de faire l'analyse, la conception d'une application de jeu de dés en utilisant les patrons de conception vus en cours et réaliser l'application avec des technologies au choix mais en utilisant différentes solutions de persistances (bases de données relationnelles, fichiers XML …).

1.2. Fonctionnalités de l'application

Les fonctionnalités supportées par l'application à réaliser sont les suivantes :

- Présenter au joueur une interface d'entrée dans le jeu.
- Expliquer au joueur les conditions du déroulement du jeu, quelles règles devrait-il prendre en compte.
- Guider le joueur pour découvrir les différentes fonctions de l'application.
- Le joueur doit choisir un pseudo pour pouvoir sauvegarder ses scores.
- Sauvegarder les scores de chaque joueur.
- Chaque joueur pourra comparer son score au score du meilleur joueur.

1.3. Déroulement du jeu

Tout jeu est réagi par un certain nombre de règle que le joueur devra respecter. Dans le cadre de ce jeu, les règles sont les suivantes :

- Pour jouer, le joueur doit choisir un pseudo puis commencer le jeu.
- Par défaut, le nombre maximum de lancer autoriser par partie est fixé à 10 mais le joueur a la possibilité de changer ce paramètre.
- Deux dés de 6 faces sont disponibles et à chaque lancé le joueur obtient une face aléatoire de chaque dé.
- Lorsque la somme des valeurs des faces des deux dés vaut 7 le joueur marque alors 10 points sinon 0 points.
- Le jeu se termine lorsque le joueur effectue tous les lancés.

2. CONCEPTION DU PROJET

2.1. Diagramme de cas d'utilisation

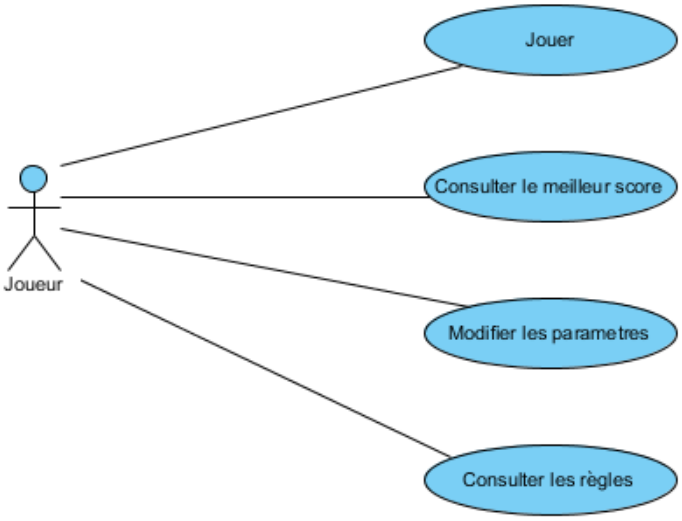

Figure 1 : Diagramme de cas d'utilisation

2.2. Diagramme de classe

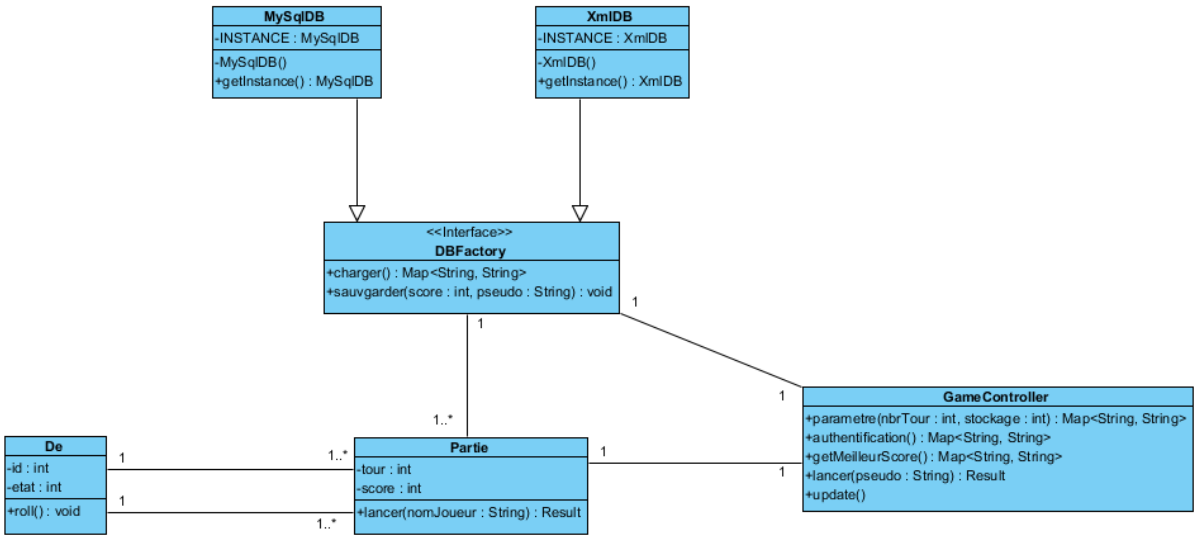

Figure 2 : Diagramme de classe

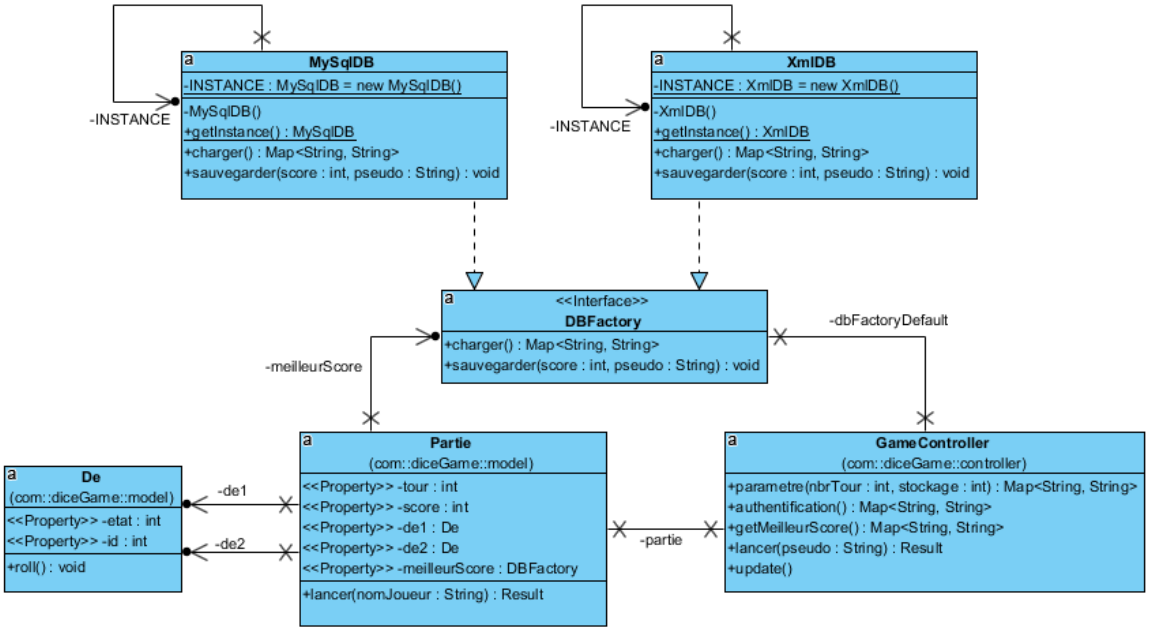

Figure 3 : Diagramme de classe après reverse

2.3. Diagramme de séquence (Jouer)

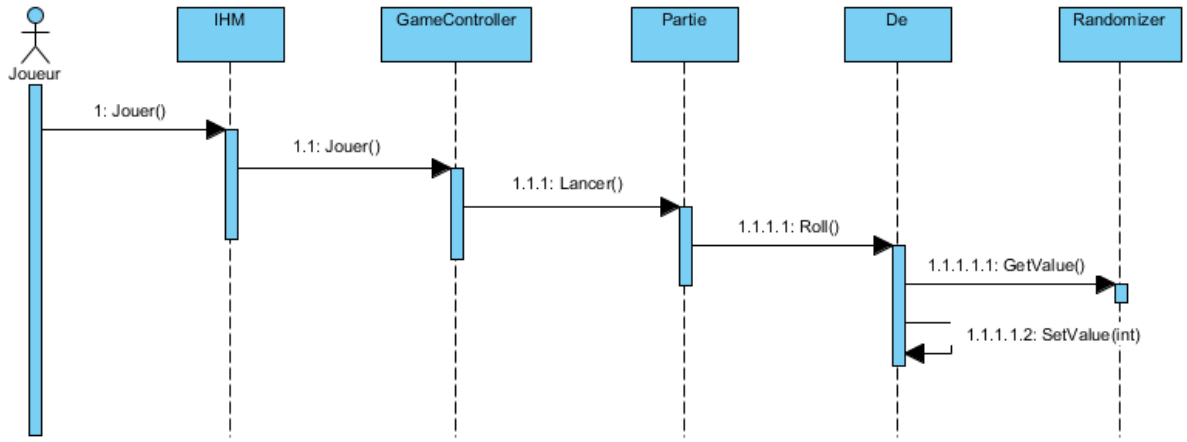

Figure 4 : Diagramme séquence (Jouer)

2.4. Diagramme d’activité (Jouer)

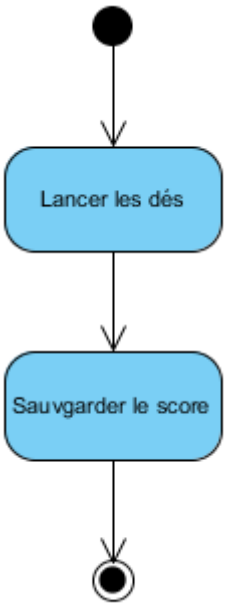

Figure 5 : Diagramme d'activité (Jouer)

3. REALISATION DU PROJET

3.1. Langages

JAVA :

Pour la réalisation de cette application nous utiliserons comme langage de développement le langage JAVA pour le traitement coté back-end.

Spring boot :

Spring est un framework bien connu des développeurs Java pour les nombreuses fonctionnalités qu'il apporte sur les aspects web, sécurité, batch ou encore accès aux données.

Spring Boot est un projet ou un micro framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.

Angular 2 :

Pour le traitement coté front-end nous avons choisis comme framework angular 2, pour éviter le chargement de l'application (SPA), et aussi il nous a poussez a utiliser un nouveau langage typé : le TypeScript.

RestFull :

Un style d'architecture qui permet un échange simple des données on se basant sur JAX-RS, utilisant un format d'échange indépendant de tout langage, comme XML ou bien JSON.

Les Web services Restfull vont nous permettre communique le back-end avec le front-end.

3.2. Outils

Eclipse :

Dernière version du fameux IDE Eclipse, il tire toute sa puissance par le fait d'être open source ce qui est visible sur la multitude de plugins disponibles.

Visual Studio Code :

Visual Studio Code est un éditeur de code source léger mais puissant qui fonctionne sur votre bureau et est disponible pour Windows, MacOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langues (C ++, C #, Python, PHP, Go) et les temps d'exécution (tels que .NET et Unity).

3.3. Persistance

MySql :

MySQL (prononcer [maj.ɛs.ky.ɛl]) est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde3, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

XML :

Xml (eXtensible Markup Language) Est un langage HTML amélioré permettant de définir de nouvelles balises. Il permet de mettre en forme des documents grâce à des balises (markup).

3.4. Patron de conception utilisé

Observateur :

Le principe de ce patron est de définir une liste d'observateurs dans la classe qui est observée, ainsi avec la méthode notifyObserver(), tous les observateurs sont notifiés lors d'un changement.

Les observables sont les dés et la partie. L'observateur est un contrôleur Rest (GameController) où toutes les informations de la partie se trouvent. L'observateur est notifié quand les dés sont lancés (méthode 'roll()' de la classe Dé) et également à la fin de la méthode 'lancer()' de la classe Partie, qui incrémente le nombre de tours et modifie le score du joueur.

La méthode update() de la classe GameController est appelée à chaque notification. Nous vérifions l'instance de l'observable qui a invoqué cette méthode, et nous mettons à jour les éléments de la fenêtre de jeu en fonction de l'observable.

Abstract Factory (DbFactory) :

La fabrique abstraite est une interface générique qui permet de créer des objets concrets. On n'a pas à se soucier de connaître la classe concrète de l'objet en question, le code client n’interagit qu'avec la classe abstraire.

Nous avons créé une méthode getFactory(int type) qui retourne aléatoirement une des trois classes de persistance.

Singleton :

Les deux classes de persistance sont des singletons. Elles ne sont instanciées qu'une seule fois. L'instance est créée à l'initialisation.

4. LES INTERFACES

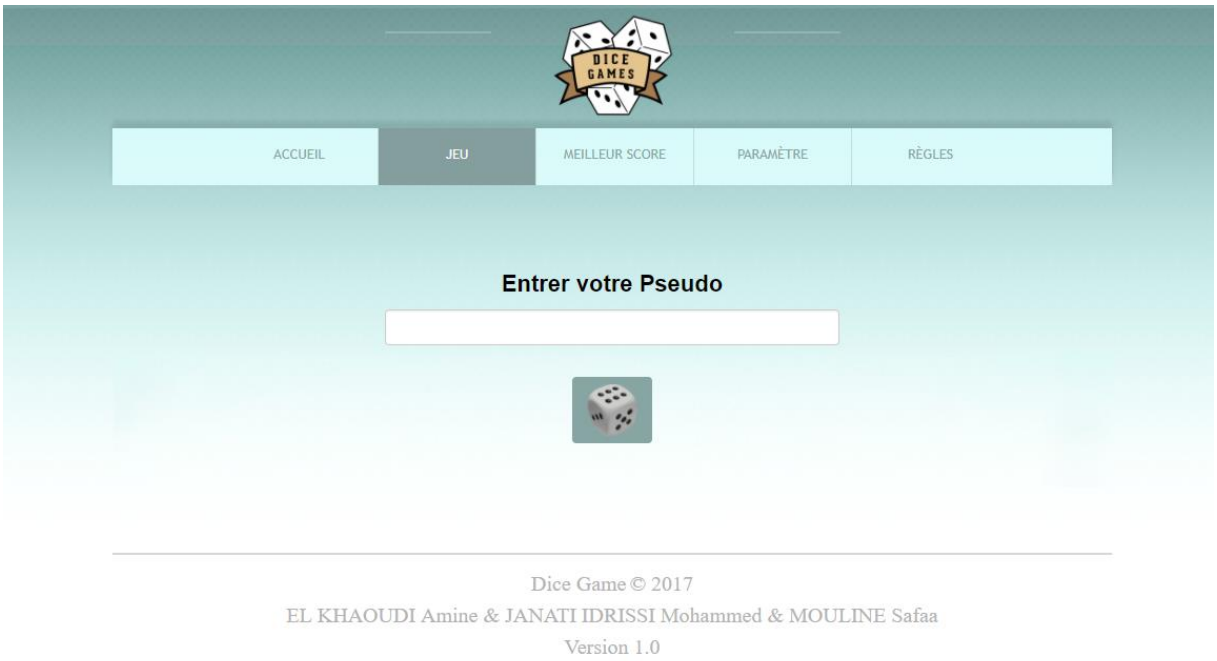

Figure 6 : Interface d'entrer le pseudo

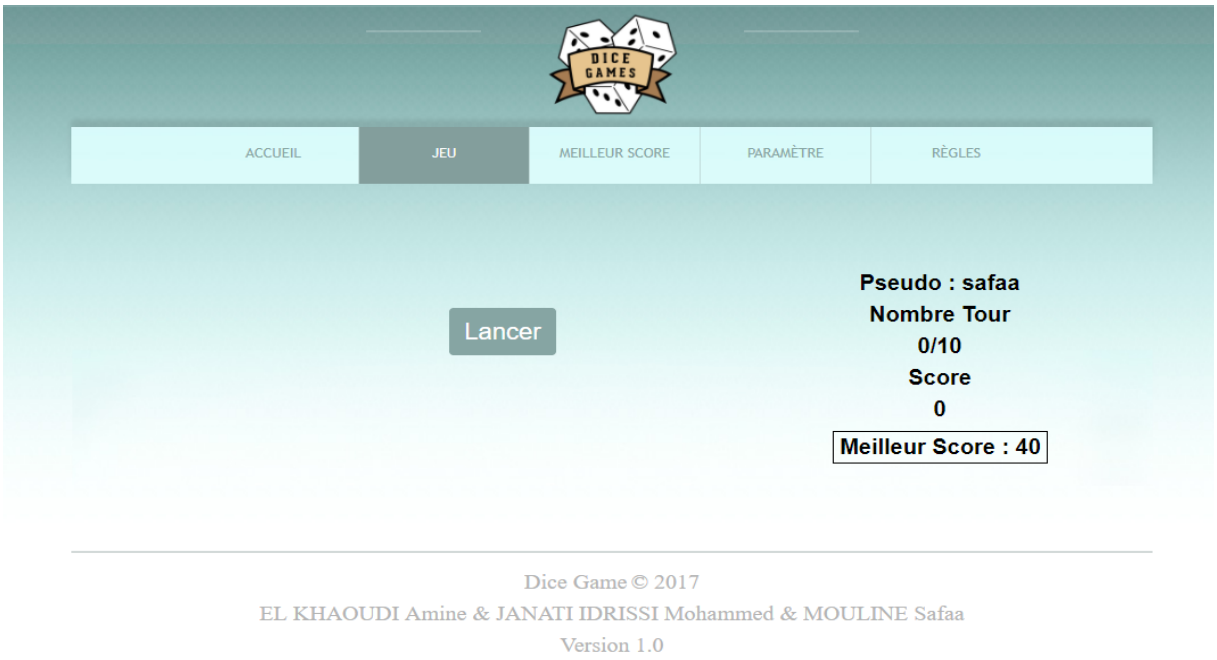

Figure 7 : Interface de le début d'une partie

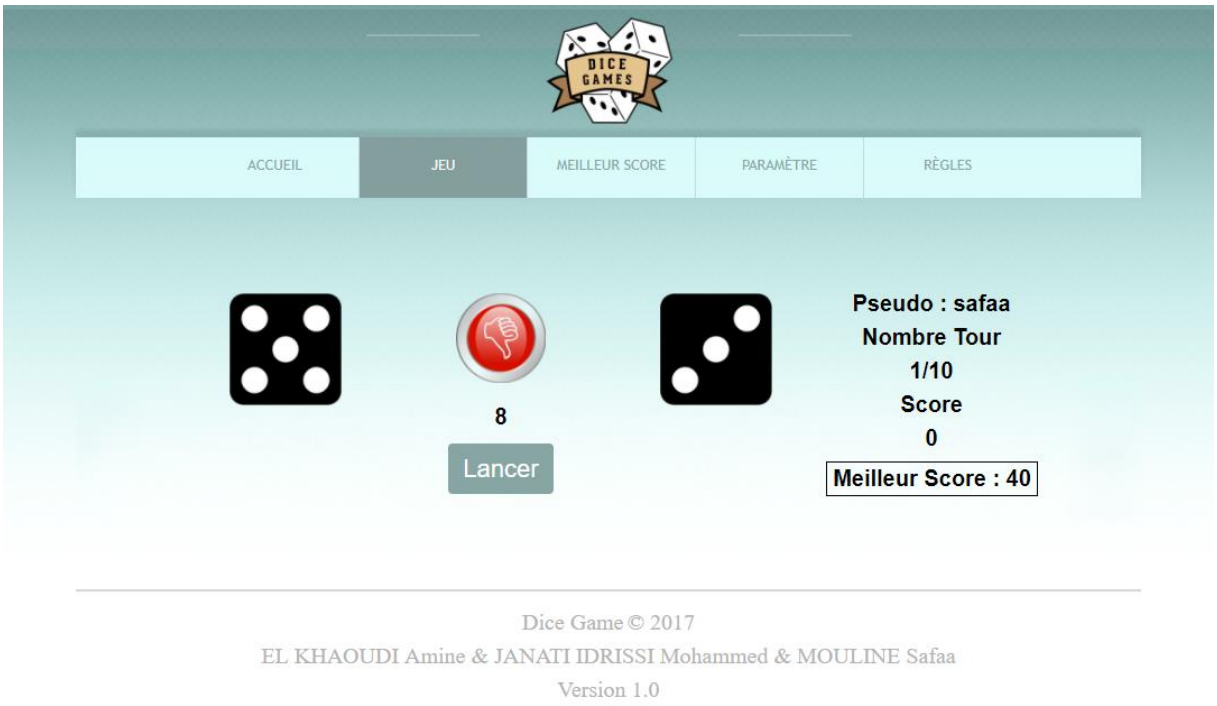

Figure 8 : Interface d'une mauvais lancer

Figure 9 : Interface d'une bon lancer

Figure 10 : Interface d'une partie perdu

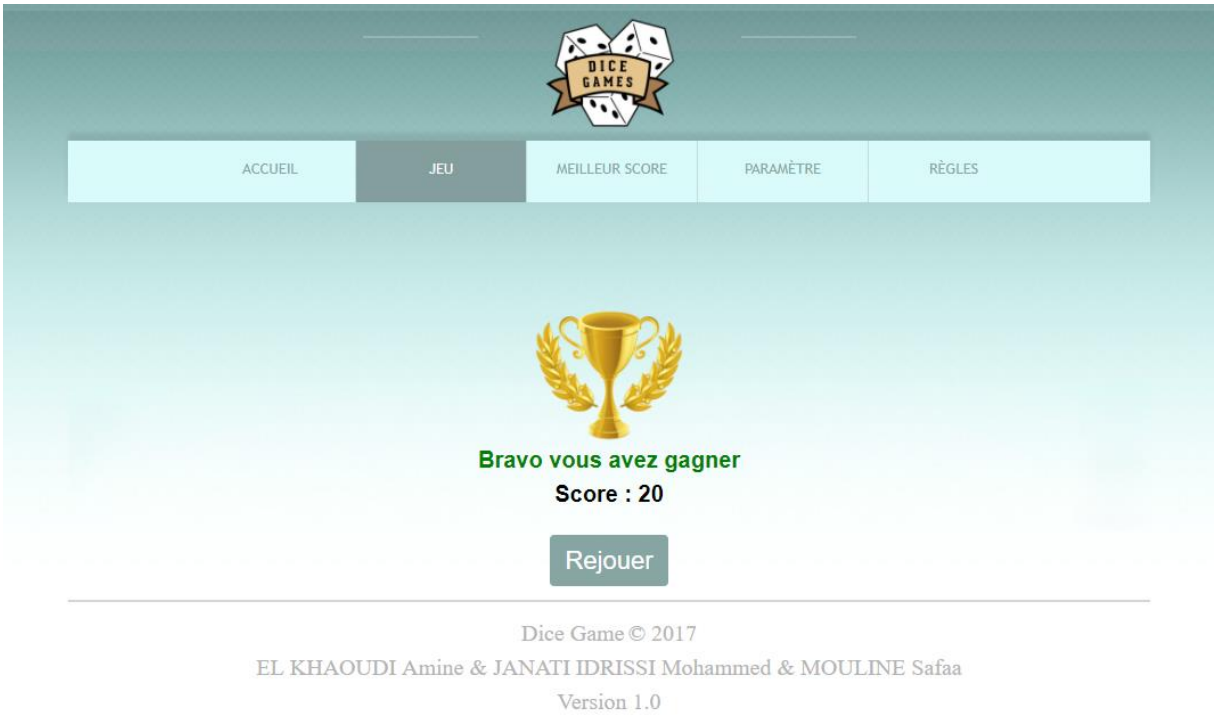

Figure 11 : Interface d'une partie gagner

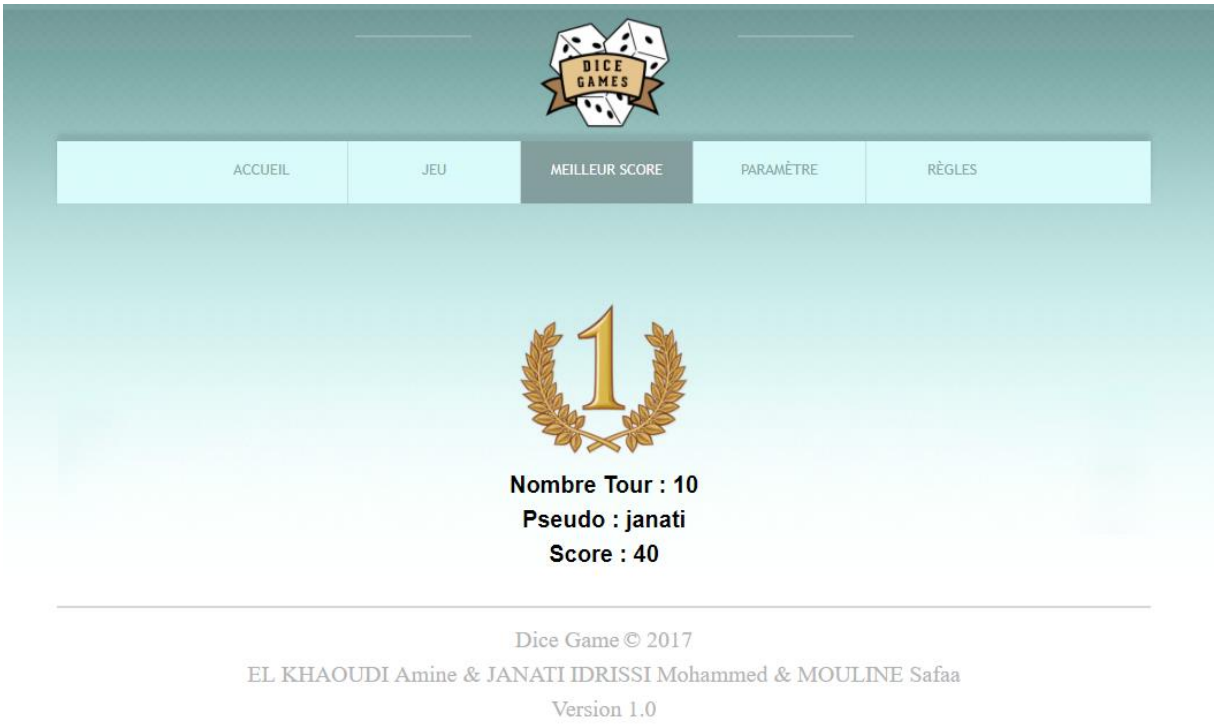

Figure 12 : Interface de meilleur score

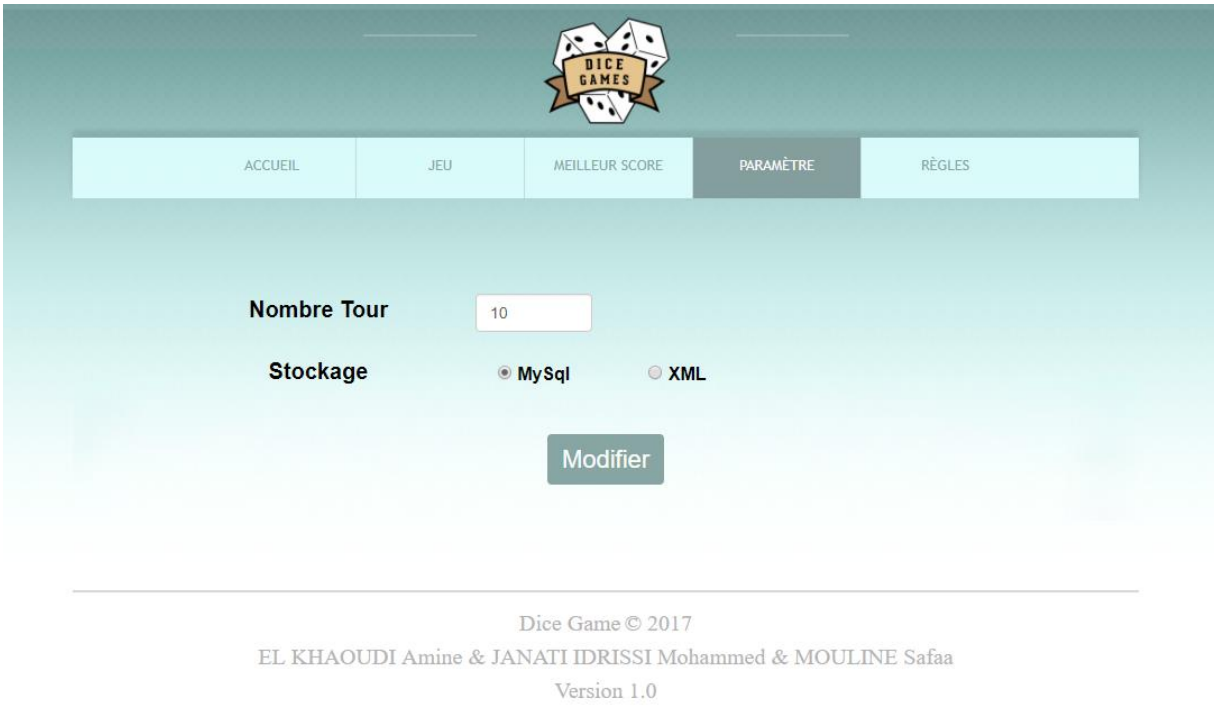

Figure 13 : Interface du paramètre

CONCLUSION

Le but de ce projet était de concevoir et développer une application de jeu de dés en utilisant des technologies aux choix tout en mettant en œuvre les différentes connaissances vues durant le cours d'applications distribuées sur les design patterns.

Durant la réalisation de ce projet, nous avons pu apprendre de nombreuses notions comme la persistance des données dans différentes bases de données, le développement des interfaces graphiques en utilisant Angular 2 et la communication entre deux application avec le web service RestFull.

TABLE DES FIGURES

REFERENCES

1. https://fr.wikipedia.org/wiki/MySQL
2. https://code.visualstudio.com/docs
3. http://www.groupeafg.com/spring-boot-kesako/
4. http://blog.xebia.fr/2015/12/14/angular-2-presentation/