

Projet Java

Rapport 2: Java IO/ Base de données

SNAKE GAME

Classe : 17VP

Notre groupe:

Nguyễn Thị Quỳnh Hoa – 1752017

Lưu Phạm Mạnh Hà – 1752001

Nguyễn Hoàng Anh – 1652002

Tableau de contenu

L'histoire de versions	3
Introduction	4
Analyse et conception	5
Implémentation	6
Exemples de données	9
Résultat	10
Plan	12
Références	13

L'histoire de versions

Le nombre ordinal	Le travailler	Le réalisateur	La date
1	Analyse et conception	1752001 - Lưu Phạm Mạnh Hà	8/5/2020
2	Implémentation	1752001 - Lưu Phạm Mạnh Hà	8/5/2020
3	Résultat	1752001 - Lưu Phạm Mạnh Hà	8/5/2020
4	Références	1752001 - Lưu Phạm Mạnh Hà	8/5/2020
5	Introduction	1752001 - Lưu Phạm Mạnh Hà	9/5/2020
6	Plan	1752017 - Nguyễn Thị Quỳnh Hoa	9/5/2020
7			
8			

Introduction

[Vous présentez ici le but de votre rapport, expliquer comment vous stocker les données de votre application (soit dans les fichiers, soit un système de gestion de la base de données), donnez la raison de votre choix.]

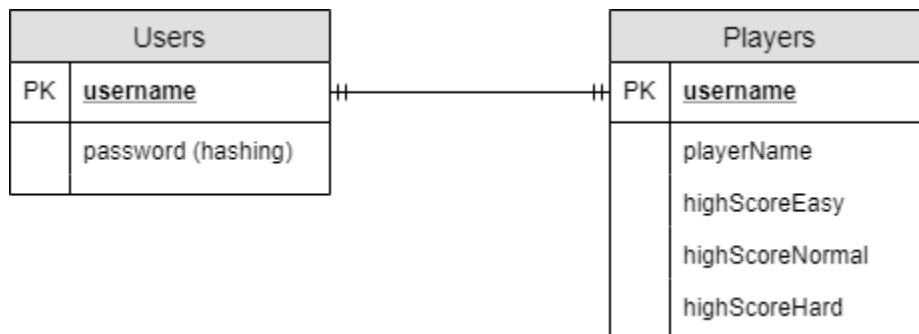
- Pour notre projet, on va stocker les données de votre application via les fichiers et le base de données MongoDB
 - Les données concernant les configurations de l'application sont stockées dans un texte fichier .txt pour récupérer facilement et rapidement.
 - Les données concernant les information de l'utilisateur sont stockées en ligne dans MongoDB pour créer la relation entre tous les utilisateur (comparer les marques entre les joueuses)

Analyse et conception

Les informations doivent être stockées:

- Les configurations de l'application sont stockées dans un fichier texte (.txt) avec le format
 <nom de la propriété> = <valeur de la propriété>
 - La taille
 - Le volume du son
- Les informations de l'utilisateur sont stockées dans MongoDB
 - Nom d'utilisateur
 - Mot de passe
 - Nom de joueur
 - Les marqueurs pour toutes les difficultés du jeu

+ Les relations entre les tableaux



Implémentation

- Utiliser `InputStream` et `OutputStream` pour charger et sauvegarder les données dans le fichier `.txt`. Utiliser la bibliothèque `java.util.Properties` pour la format du fichier

```
public static void saveConfig() {  
    try (OutputStream output = new FileOutputStream("SnakeConfig.txt")) {  
        Properties properties = new Properties();  
        properties.setProperty("scale", Integer.toString(SCALE));  
        properties.setProperty("sound", Integer.toString(SOUND));  
        properties.store(output, "Application configuration");  
        System.out.println(properties);  
    } catch (IOException ex) {  
        ex.printStackTrace();  
    }  
}
```

```
public static void loadConfig() {  
    try (InputStream input = new FileInputStream("SnakeConfig.txt")) {  
        Properties properties = new Properties();  
        properties.load(input);  
        SCALE = Integer.parseInt(properties.getProperty("scale"));  
        SOUND = Integer.parseInt(properties.getProperty("sound"));  
    } catch (IOException ex) {  
        ex.printStackTrace();  
    }  
}
```

- Utiliser MongoDB Java Driver pour connecter MongoDB via MongoClient

```
public static void connect(String connection, String database) {  
    client = MongoClient.create(connection);  
    db = client.getDatabase(database);  
}  
  
public static void close() { client.close(); }
```

- Le mot de passe est haché avant d'être stocké dans la base de données pour des raisons de sécurité. Collection.insertOne() pour stocker et Collection.updateOne/replaceOne pour mise à jour les données

```
public static boolean addUser(String username, String password) {  
    String hash = BCrypt.hashpw(password, BCrypt.gensalt());  
    Document user = new Document("username", username).append("password", hash);  
  
    MongoClient users = db.getCollection("Users");  
    Document existUser = (Document) users.find(new Document("username", username)).first();  
  
    if (existUser == null) {  
        db.getCollection("Users").insertOne(user);  
        return true;  
    }  
  
    return false;  
}
```

```
public static void update(Player player) {

    String username = player.getUsername();
    if (username == "" || username == null)
        return;
    Document playerUpdate
        = new Document("username", username)
        .append("playerName", player.getName())
        .append("highScoreEasy", player.getHighScoreEasy())
        .append("highScoreNormal", player.getHighScoreNormal())
        .append("highScoreHard", player.getHighScoreHard());

    ReplaceOptions replaceOptions = new ReplaceOptions().upsert(true)
        .bypassDocumentValidation(true);

    try {
        db.getCollection("Player")
            .replaceOne(eq("username", username),
                playerUpdate,
                replaceOptions);
    } catch (MongoException ex) {
        System.out.println("Update failed");
    }
}
```

Exemples de données

[Donnez ici un exemple de la structure de données et donnez votre explication pour chaque exemple. C'est les données que vous pouvez utiliser pour tester votre application plus tard...]

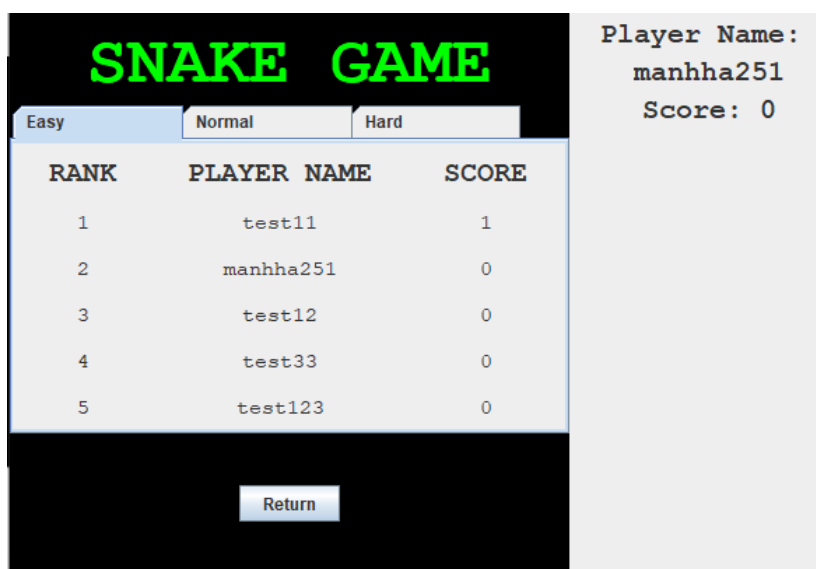
Résultat

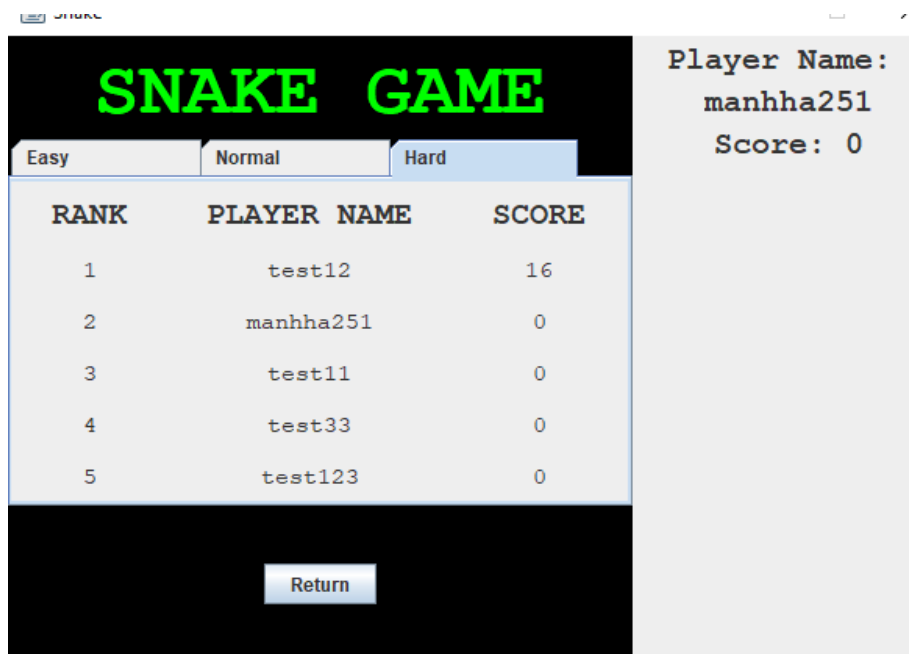
[Expliquez ce que vous obtenez jusqu'à présente, des avantages, désavantages et les solutions prévues (si c'est possible)]

- Ajoute une utilisateur



- Afficher les tableaux des meilleurs scores pour chaque difficultés





Plan

[Donnez le plan de votre projet (en détaille) jusqu'à la fin du projet: qui fait quoi, la durée de chaque tâche]

Semaine	De jour - En jour	Le travailler	Le réalisateur
1	23 - 29/03/2020	Conception de l'interface	Hoàng Anh, Quỳnh Hoa.
2	30/03 - 05/04/2020	Conception de base de backend de jeu	Mạnh Hà.
3	06-12/04/2020	Jeu d'essai (Aucune interface)	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
4	13-19/04/2020	Jeu de base complet avec des interfaces	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
5	20-26/04/2020	Objets de jeu supplémentaires	Hoàng Anh, Quỳnh Hoa.
6	27/04 - 03/05/2020	Ajouter le menu	Mạnh Hà (Hoàng Anh, Quỳnh Hoa.)
7	04-10/05/2020	Ajouter la base de données	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
8	11-17/05/2020	Retester et terminer le jeu	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.

Références

[Listez tous les ressources à utiliser dans votre projet, y compris les codes existantes, les algorithmes utilisés, les livres, les rapports, les liens à consulter, etc.]

<https://mkyong.com/java/java-properties-file-examples/>

<https://mongodb.github.io/mongo-java-driver/4.0/>