MIAGE RÉSEAU DES MIAGE DE FRANCE
Nice

Roue de la Fortune

2020

Bydon Sacha
Desire Stéphane
Maugard Samuel
Mercuri Sabrina

1 Table des matières

2 Comment jouer

2.1 Prérequis

Pour lancer le projet, vous aurez besoin d'installer Maven sur votre machine.

- Sur une machine Windows la démarche est assez laborieuse mais voilà un lien qui vous explique la démarche à suivre pour installer Maven. https://www.codeflow.site/fr/article/maven__how-to-install-maven-in-windows
- Sur linux il suffit d'exécuter la commande « *apt install maven* »

Une fois que vous avez installé Maven il faut installer les dépendances avec la commande :

« *mvn clean install* »

2.2 Sur notre serveur

Il faut exécuter la classe Client.java (dans src/client), ou exécuter la commande :

« *mvn exec:java -Dexec.mainClass=client.Client* »

Puis pour pouvoir se connecter au serveur, il suffit d'entrer l'adresse du serveur déjà placé dans la zone de texte.

109.210.118.18

2.3 En local

Il faut démarrer le serveur, soit en exécutant la classe Serveur.java (dans src/serveur), soit en exécutant la commande :

« *mvn exec:java -Dexec.mainClass=server.Serveur* »

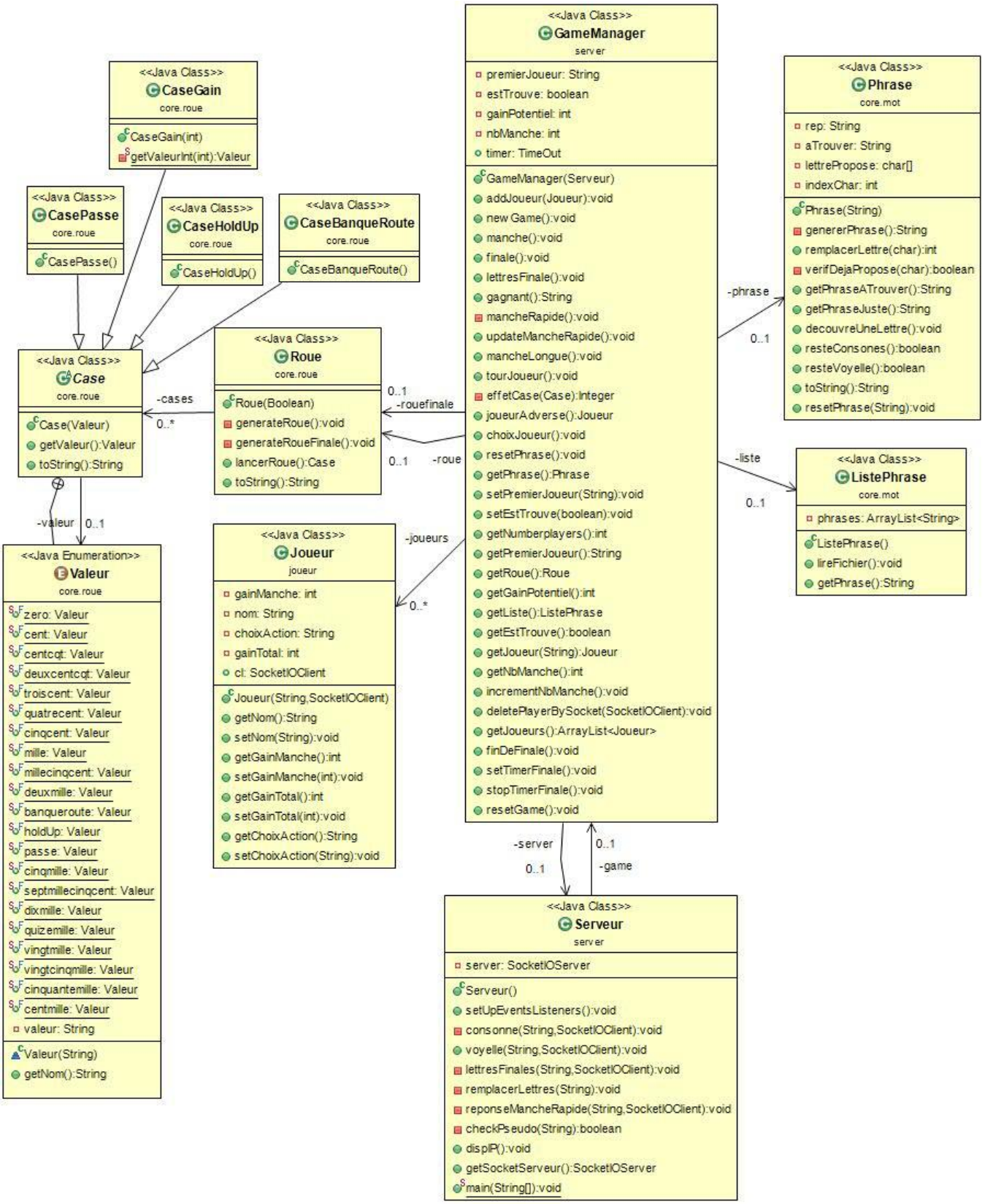
Puis pour lancer un client, il faut exécuter la classe Client.java (dans src/client), ou exécuter la commande :

« *mvn exec:java -Dexec.mainClass=client.Client* »

Ensuite pour pouvoir se connecter au serveur, il suffit d'entrer l'adresse « 127.0.0.1 » dans la zone de texte.

3 Architecture et Fonctionnalités

3.1 Représentation générale



3.2 Explication du fonctionnement

3.2.1 Core du jeu

Déroulement du jeu :

Le jeu se déroule en 5 manches. Les 4 premières manches sont composées chacune d'une manche rapide et d'une manche longue. Une manche rapide affiche des lettres dans un ordre aléatoire toutes les 2 secondes jusqu'à l'apparition du mot : le but étant de taper la réponse le plus rapidement possible pour empocher 500 de gain et de prendre la main pour la manche longue.

Pour la manche longue, la roue tourne et affiche la case obtenue. Selon la case obtenue, il passera son tour ou pourra jouer. S'il peut jouer, il pourra alors choisir une consonne (en tapant c puis la consonne), une voyelle s'il a minimum 200 de gain pour la payer (en tapant v puis la voyelle et en se délestant de 200 de gain) ou alors il pourra proposer une réponse entière (en tapant r puis la phrase qu'il pense correcte).

Nous avons fait en sorte que lorsque le joueur aura proposé toutes les consonnes, il ne pourra plus proposer de consonnes et de même pour les voyelles.

Lorsque les 4 premières manches sont terminées, le joueur ayant le plus de gains (gagnés lors des manches précédentes) pourra participer à la manche finale (le perdant pourra regarder son déroulement mais ne pourra pas jouer). Une deuxième roue est alors lancée avec des gains plus importants.

La phrase affichée aura les lettres r, s, t, l, n et e d'affichées et le joueur pourra proposer 3 consonnes et 1 voyelle. Une fois les lettres affichées (si elles sont contenues dans la phrase), il aura alors 30sec pour trouver le bon mot. Si le joueur trouve le bon mot, il gagnera les gains obtenus lors de toutes les manches et sinon il ne gagnera que les gains gagnés lors des 4 premières manches.

Explications au niveau du code :

Pour ce qui est du core du jeu, nous avons décidé de l'organiser autour du Game manager qui permet de gérer le déroulement de la partie.

Le Game manager crée 2 roues différentes au début de la partie, la roue utilisée dans toutes les manches longues ainsi que la roue utilisée pour la finale.

Nos roues sont composées de cases dont le comportement est généralisé dans la classe abstraite case.

Nous avons décidé de regrouper les cases en 4 classes qui gèrent respectivement les gains, les Holdup, les passe, les Banqueroute.

Pour ce qui est des phrases du jeu, une classe « ListePhrase » va permettre de lire le fichier qui contient toutes les phrases à trouver dans le jeu et d'en récupérer une au hasard qui servira lors d'une manche. Elle permet également de faire en sorte que la phrase choisie, ne puisse pas être choisi une deuxième fois lors de la partie.

Lorsque cette phrase est choisie, la classe « Phrase » va alors créer un doublon transformé en phrase à trous en remplaçant les lettres par des tirets « _ » et en rajoutant des espaces pour une meilleure visibilité.

Pendant la partie, lorsque le joueur va proposer des lettres, on vérifie si la lettre se trouve dans la phrase choisie et on remplace alors les tirets par la lettre (sauf si celle-ci a déjà été proposée).

3.2.2 Réseau

Pour la partie réseau, nous avons utilisé la bibliothèque socket IO en mode Web Socket qui utilise le protocole TCP. Le serveur tourne sur le port 10101. Le serveur est hébergé sur un Raspberry placé dans une DMZ avec les ports ouverts pour rendre la connexion possible.

Les échanges entre le serveur et les clients se font grâce à des messages envoyés grâce aux méthodes « emit() » (client) et « sendEvent() » (serveur). Ces messages sont ensuite réceptionnés avec des écouteurs « EventListener ».

Une partie commence lorsque 2 personnes sont connectées sur le serveur. Si un client se déconnecte au cours d'une partie, elle est donc annulée et le client restant est déconnecté du serveur. Si un client essaie de se connecter pendant une partie, la connexion lui sera refusée. Et à la fin d'une partie, le gagnant aura le choix entre relancer une partie et quitter (ce qui déconnectera les deux clients). Après la partie terminée le serveur sera de nouveau en attente de deux nouveaux clients pour lancer une nouvelle partie.