Annexes

**I. Glossaire**

*SGBD :* Un système de gestion de base de données est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou les informations de la base de données.

*Thread :* Un thread ou fil (d'exécution) ou tâche, est similaire à un processus car tous deux représentent l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle. Toutefois, là où chaque processus possède sa propre mémoire virtuelle, les threads d'un même processus se partagent sa mémoire virtuelle. Par contre, tous les threads possèdent leur propre pile d’appel.

*Socket :* Il s’agit d’un modèle permettant la communication inter processus afin de permettre à divers processus de communiquer aussi bien sur une même machine qu’à travers un réseau TCP/IP.

Ici, les sockets permettent de communiquer entre les clients et le serveur.

*API :* Une interface de programmation (Application Programming Interface ou API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine.

Du point de vue technique une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels.

*Texas hold’em no limit:* C’est la règle la plus connue de poker, sans restriction maximale de mise ou de relance. Elle se joue avec 2 cartes en main et 5 sur la table. La meilleure combinaison de 5 cartes gagne.

*Blind/surblind :* ce sont des mises obligatoires au début d’un tour pour éviter la triche.

**II. Protocole**

//Demande de connexion

CONNECT@pseudo@mdp

Retour :

CONNECTOK: opération effectué

WPASS : mauvais password utilisé, connexion impossible

WPSEUDO: pseudo introuvable, connexion impossible

PDC : compte déjà connecté

//Demandes des parties qui ne sont pas en cours (doit être connecté)

GETLISTEPARTIE

Retour :

LISTEPARTIE@partie1/nbjouer/nbmaxjouer@partie2/nbjouer/nbmaxjouer ...

NCON si pas connecté

//Création de compte

CREATCPT@nomcompte@mdp

Retour :

CREATOK : opération effectué </li>

AUPSEUDO : pseudo déjà utilisé, ajout impossible </li>

ERREURBDD : erreur a l'ajout, veuillez recommencer </li>

WFPSEUDO : mauvais format de pseudo </li>

WFPASS : mauvais format de password </li>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Changer de pseudo

ACTPSEUDO@ancienpseudo@mdp@nouveaupseudo

Retour :

OK : opération effectué </li>

WPASS : mauvais password utilisé, changement impossible </li>

WPSEUDO : pseudo introuvable </li>

AUPSEUDO : pseudo déjà utilisé </li>

WFPSEUD : mauvais format de pseudo </li>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Changer de mot de passe

ACTPASS@pseudo@ancienmdp@nouveaumdp

Retour :

OK : opération effectué </li>

WPASS : mauvais password utilisé, changement impossible </li>

WPSEUDO : pseudo introuvable, changement impossible </li>

WFPASS : mauvais format de nouveau password </li>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Créer une partie // IF CONNECTE

CREATEPARTIE@nompartie@nbjoueursmax

Retour:

CREATPOK si c'eest bon

PAU nom de partie deja utilisé

WFP mauvais format nom de partie

AIP le joueur est déjà dans une partie

NCON si pas connecté

+ Envoi des joueurs dans la partie de la forme LISTEJOUEURSPARTIE@pseudo1@pseudo2....

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//rejoindre une partie // IF CONNECTE

REJP@nompartie

Retour:

REJOK si c'est bon

PNE nom de partie n'existe pas

TOOMANY trop de joueurs dans la partie

AIP le joueur est déjà dans une partie

PEC la partie est en cours

NCON si pas connecté

+ Envoi des joueurs dans la partie de la forme LISTEJOUEURSPARTIE@pseudo1@pseudo2....

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//demande des joueurs dans la partie

GETPLAYERPARTY

Retour:

LISTEJOUEURSPARTIE@pseudo1@$pseudoCréateur... ($ Devant le pseudo du créateur)

NIP le joueur n'est pas dans une partie

NCON si pas connecté

+ Envoi des joueurs dans la partie de la forme LISTEJOUEURSPARTIE@pseudo1@pseudo2....

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//quitter la partie

EXITPARTIE

Retour:

EXITOK client a bien quitté la partie

ERROR erreur quelconque

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//demande les infos sur un joueur

GETINFO@pseudo

Retour:

SETINFO@pseudo@partiesGagné@partiesPerdu@dateInscription

PI pseudo introuvable

NCON si pas connecté

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//lancement de la partie par le créateur (met encours à 1)

DEBUTPARTIE

Retour:

NC n'est pas le créateur de la partie

PAJ pas assez de joueurs pour lancer la partie

NIP n'est pas dans une partie

PEC partie est déjà en cours

NCON n'est pas connecté

Envoi à toute la partie : AREUREADY pour dire de changer de fenêtre en fenêtre partie et de renvoyer IAMREADY

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//le joueur précise qu'il est prêt pour la partie

IAMREADY

Envoi à toute la partie : DEBUTPARTIE lorsque tous les joueurs sont prêt ou au bout d'un certain temps (kick ceux qui n'ont pas répondu)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//jetons de tous les joueurs de la partie

Envoi à toute la partie : JETONJ@pseudo1/jetonsTotal/jetonsPosés@pseudo2/jetonsTotal/jetonsPosés ....

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi des cartes de la main

Envoi à un joueur : CARTEM@carte1@carte2 ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi d'un message aux autres joueurs de la partie

MESSAGE@pseudo@message

Retour:

NIP n'est pas dans une partie

PNE partie n’est pas en cours

NCON n'est pas connecté

Envoi à toute la partie : MESSAGE@pseudo@message

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi des cartes de la table

Envoi à toute la partie : CARTET@nombreCarte@carte1@carte2 ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi joue et les paramètres de savoir ses choix

Envoi à un client: JOUE@JetonsMin@JetonsMax@booléen

Le booléen correspond savoir si relancer est disponible

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//le client envoie son choix au serveur

CHOIX@numéroDeChoix@NombreJetons

numéroDeChoix:

1: se coucher (mettre NombreJetons à 0)

2: suivre (mettre NombreJetons à 0)

3: relancer

Retour:

PAT pas au tour de se client de jouer

MJET mauvais nombre de jetons

MNC mauvais numéro de choix

JOK ça s'est bien passé et ce n'est plus à lui de jouer

NIP n'est pas dans une partie

NCON n'est pas connecté

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi des jetons de la table

Envoi à toute la partie : JETONT@nombreJetons

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi des cartes de ceux qui ne sont pas couchés

Envoi à toute la partie : MONTREC@pseudo1/carte1/carte2@pseudo2/carte1/carte2 ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi le gagnant du tour (celui qui ramasse des jetons)

Envoi à toute la partie : GAGNANTT@pseudo1@pseudo2 ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi le gagnant de la partie (celui qui gagne une victoire++)

Envoi à toute la partie : GAGNANTP@pseudo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi un perdant de la partie (celui qui gagne une defaite++)

Envoi à toute la partie : PERDU@pseudo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi le choix du joueur au pseudo "pseudo"

Envoi à toute la partie : JCHOIX@pseudo@choix

Choix est un entier: 1==se coucher

2==suivre

3==relancer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//envoi le joueur a qui c'est le tour de jouer à tous les joueurs

Envoi à toute la partie : JOUEURJ@pseudo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Retour si erreur syntaxe : ERROR

**III. Manuel utilisateur**

**Manuel utilisateur : Serveur :**

1. Démarrage du *SGBD* : MongoDB

Pour que le serveur puisse fonctionner, le *SGBD* doit être en route. Pour cela, il faut ouvrir un terminal (ou créer un .bat), se placer dans le dossier qui contient mongoDB, puis exécuter la commande suivante :

mongod --dbpath Path

Où Path est le chemin vers le dossier où vous voulez mettre la base de données.

Il est également possible de changer les paramètres de la base de données (authentifications ...) en utilisant la commande « mongo ». Le site officiel MongoDB met en détail toutes les possibilités.

1. Démarrage de Poker.jar

Pour lancer le serveur, il faut exécuter le .jar. Pour cela, il faut avoir java d'installé et la commande java dans les variables d’environnement (normalement automatique).

Il faut ensuite ouvrir un terminal ( ou créer un .bat), puis exécuter la commande suivante :

java -jar Path Port

Path est le chemin vers le .bat.

Port est le numéro du port que le serveur doit utiliser. Si aucun port n'est indiqué, le port 6667 sera utilisé.

Pour qu'un client puisse se connecter au serveur, il doit connaître l'IP de la machine sur lequel est le serveur, et le port à utiliser.

De plus, si les machines ne sont pas reliées par un réseau local, le port utilisé doit être redirigé vers la machine qui lance le serveur.

Dans le cas d'une box d'un particulier, il suffit de s'y connecter et de changer les paramètres.

Attention: dans le dossier où se trouve Poker.jar doit se trouver un dossier lib contenant la librairie (.jar) de mongoDB.

3) Utilisation du serveur

Le serveur se lance dans une console : de nombreuses traces sur le déroulement du jeu s'affichent.

De plus, quelques commandes sont disponibles. Pour avoir sa liste, il suffit de taper HELP dans la console.



Ainsi, un utilisateur peut voir la liste des clients, des parties, des données dans la base de données, déconnecter un client, voir les crédits du projet ou fermer le serveur.

Le lecteur de commande est dans un *thread* à part, donc les commandes peuvent être tapés à n'importe quel moment.

**Manuel utilisateur : Client Web**

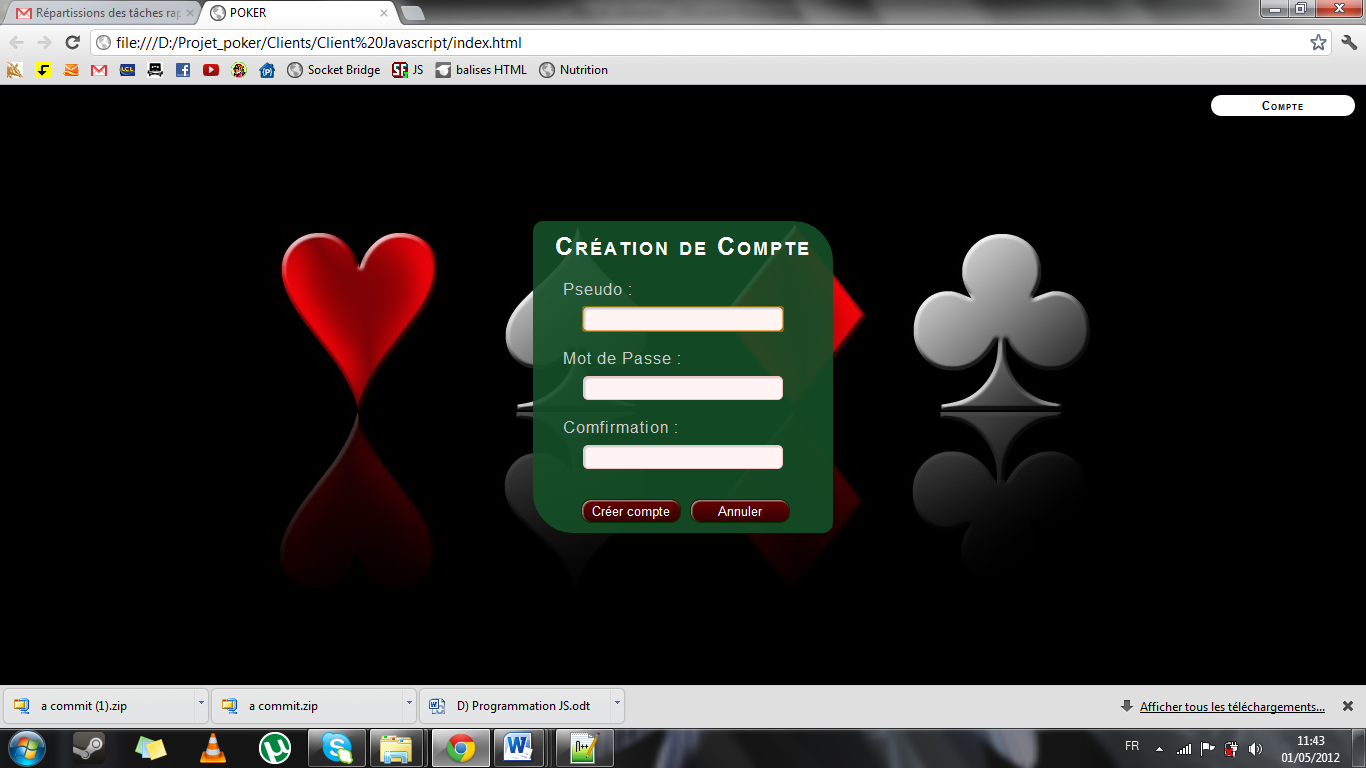
1. Démarrage

Le programme de par son orientation web se présente sous la forme d’une arborescence HTML. Pour le démarrer, il suffit donc d’ouvrir l’élément « index.html » avec un navigateur internet. Pour des soucis de réalisation, la compatibilité est garantie de manière optimale seulement avec les navigateurs Chrome et Safari.

La communication avec le serveur est gérée par une applet java. C’est pourquoi il est nécessaire que Java soit installer sur la machine du client et d’autoriser l’exécution de l’applet au démarrage de la page.

1. Mode non connecté

A l’ouverture, la page est munie seulement un menu « Compte » contenant quatre éléments : « Connexion », « Créer un compte », « Changer pseudo » et « Changer mot de passe ». L’action sur l’un des éléments génère l’affichage d’un formulaire comme suit.



1. Mode connecté

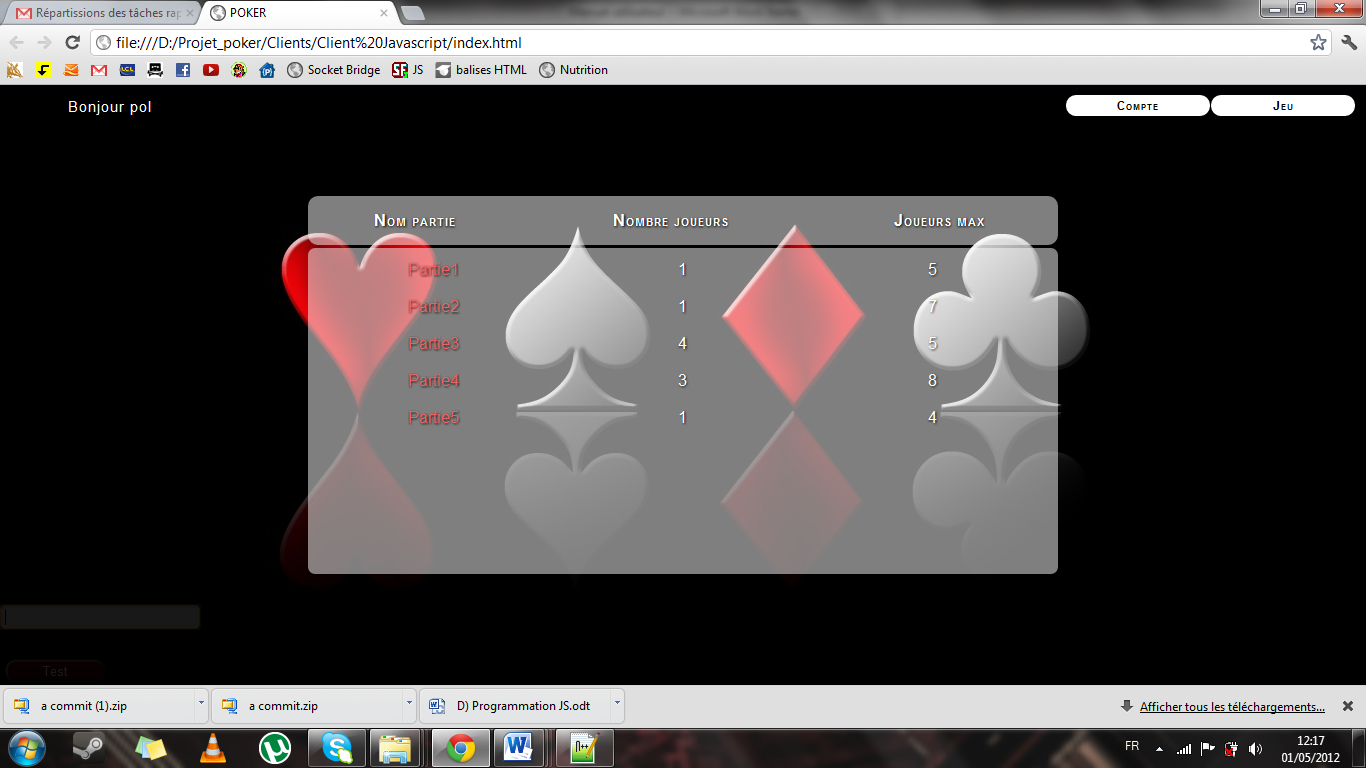
Après une demande de connexion validée par le serveur, le menu est modifié et comprend maintenant deux onglets. Le premier, « Compte » contient les éléments « Infos personnelles » et « Deconnexion ». Le second, « Jeu », en comprend deux également : « Créer une partie » et « Parties en cours ».

L’action sur « Deconnexion » entrainera le retour au mode non connecté.

Celle sur « Infos personnelles » provoque l’envoi au serveur d’une demande d’informations sur le compte, qui sont affichées dès leur réception.

Cliquer sur « Créer une partie », génère l’affichage d’un formulaire à deux champs de la même forme que ceux du mode non connecté.

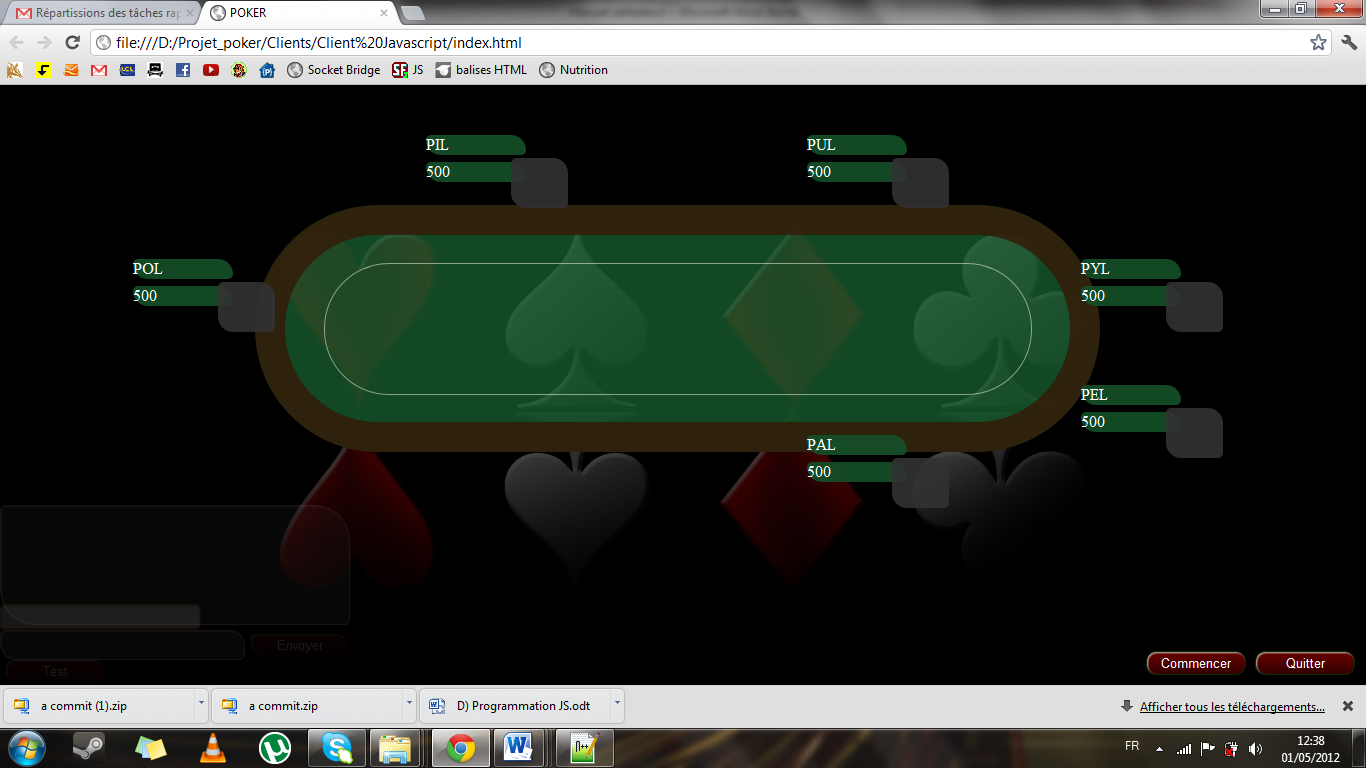
Enfin, l’action sur « Partie en cours » demande la liste des parties au serveur. Lors de la réception de celle-ci l’affichage d’un tableau comme suit :



L’utilisateur peut alors cliquer sur la partie de son choix et une demande pour la rejoindre est envoyée au serveur.

1. Déroulement d’une partie

Après la création d’une partie ou l’acceptation du serveur pour en rejoindre une, les menus disparaissent, une table, un chat et un bouton « Quitter » sont affichés. Un bouton « Commencer » apparait également si l’utilisateur est le créateur de la partie. Le joueur est placé sur la table et des jetons lui sont attribués.



Lorsque le créateur a lancé la partie, les cartes sont distribuées et quand c’est son tour, trois choix sont proposés à l’utilisateur : se coucher, suivre ou relancer.



A tout moment l’utilisateur peut dialoguer avec les autres joueurs via le chat situé en bas à gauche.



L’utilisateur peut également demander des informations sur ses adversaires en cliquant sur leur pseudo. Celles-ci s’afficheront en bas à droite au-dessus des boutons et s’effaceront lorsque l’utilisateur enlèvera son curseur du pseudo

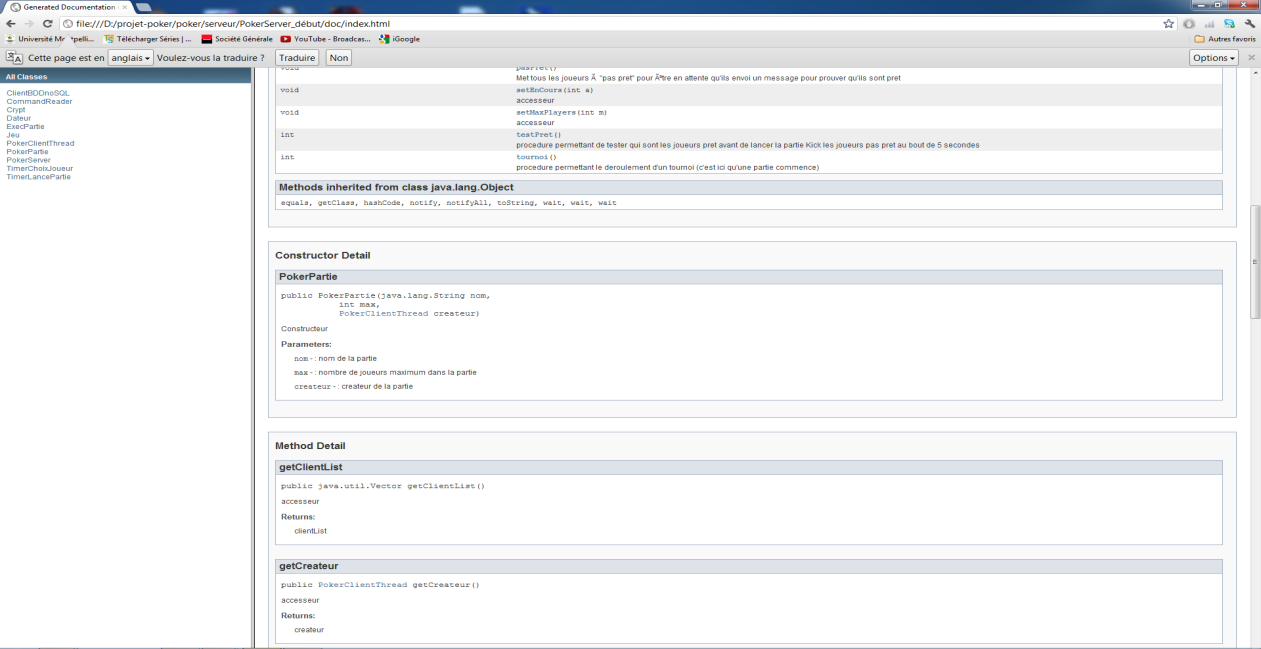
**IV. Cahier de maintenance**

**Cahier de maintenance serveur**

1. La reprise du code :

Reprendre le code d’une application c’est jamais aisé. Cependant, ici quelques actions ont étés faites pour faciliter la chose :

Tout d’abord une documentation complète est disponible permettant de connaître quelles sont les classes à modifier pour ajouter de nouvelles fonctionnalités.



De plus, le code est entièrement commenté. Cela permet de compléter facilement le code.



Enfin, Le protocole est disponible afin de connaître quel messages il faut modifier, ajouter ou enlever.

1. Tester l’application :

Un manuel d’utilisation du serveur est disponible. Ainsi, vous pourrez installer et utiliser le serveur et le *SGBD* en toute simplicité.