

Protocole : Mastermind

<https://fr.wikipedia.org/wiki/Mastermind>

Le Mastermind ou Master Mind est un jeu de société pour deux joueurs dont le but est de trouver un code. C'est un jeu de réflexion, et de déduction, inventé par Mordecai Meirowitz dans les années 1970 alors qu'il travaillait comme expert en télécommunications.

Membres du groupe

Eric Rakotonirina
Manambintsoa Rahajarivelo
Cyrille Dakhli
Mohamed Amine Jaouani
Alper Bulan
Sahid Ferdjaoui

Le But
=====

Il s'agit de définir la spécification d'une implémentation rudimentaire en Java utilisant le protocole TCP/IP.

Déroulement d'une partie
=====

PARTIE CONNEXION & COMMUNICATION DES CLIENTS AVEC LE SERVEUR

Lorsque les clients se connectent au serveur, ils sont placés dans une "salle d'attente", ayant chacun un identifiant permettant de les distinguer.

Chaque joueur est notifié par le serveur de son propre identifiant via un message (permet de savoir contre qui on joue et également de choisir contre qui on veut jouer)

Les options qui leur sont possibles sont les suivantes :

- soit demander au serveur la liste des actions disponibles
- soit de quitter la salle d'attente et donc le jeu (déconnexion)
=> le serveur supprime ce client (joueur) de la liste d'attente et de ses données (pas besoin de les garder en mémoire)
- soit de demander au serveur la liste des joueurs dans la salle d'attente (= joueurs connectés)
=> le serveur envoie un message au client lui envoyant la liste des joueurs connectés et disponibles
=> les joueurs étant déjà en partie ne sont pas à afficher
- soit de demander une connexion avec l'un des joueurs disponibles dans le but de jouer une partie
=> en utilisant le protocole défini (grammaire, etc.), le serveur transmet la demande d'invitation à une partie au joueur concerné

PARTIE INITIALISATION DU JEU

Après le lancement d'une partie, les joueurs devront alors définir un code secret composé de 4 couleurs parmi 8 :
rouge; jaune; vert; bleu; orange; blanc; violet; fuchsia.

Le premier client ayant défini son code se met alors en attente.
Une partie ne peut démarrer que lorsque le second joueur aura défini le sien.

PARTIE DÉROULEMENT DU JEU

Une manche commence et chaque joueur joue sa proposition. La manche suivante ne commence que si tous les joueurs aient fini de jouer et que le serveur ait fini de valider la réponse.
Il s'en suit une succession de manche durant laquelle, à tour de rôle, chaque joueur se verra demander d'entrer une combinaison de couleurs dans le but de deviner la combinaison de l'adversaire.

A la fin de chaque manche, le serveur vérifie s'il y a un gagnant ou non (= si exactement un joueur a trouvé la combinaison de l'adversaire ou non), ou s'il y a égalité (= les deux joueurs ont découvert la combinaison de l'adversaire durant la même manche)
Le système de manche permet d'assurer l'équité des chances d'être vainqueur des deux joueurs en permettant d'obtenir un match nul.

PARTIE FIN DE JEU

La partie se termine lorsqu'au moins un des joueurs a trouvé la combinaison de son adversaire.
Le serveur envoie alors un message à chacun des deux joueurs pour annoncer le(s) gagnant(s).
Enfin, le serveur renvoie les deux joueurs dans la salle d'attente (= ces deux joueurs sont de nouveaux visibles dans la liste des joueurs disponibles et d'autres joueurs peuvent leur envoyer des invitations à jouer)

Le protocole

Grammaire

Une grammaire simple est définie en texte composé d'une clé suivi de d'un espace, d'une chaîne de caractère UTF8 et du caractère de retour à la ligne '\n'.

Elle peut se définir via la regex suivante:

```
[A-Z_]+\s?\w*\n
```

Exemples d'événement valident:

```
MY_KEY\n
MY_KEY1\n
```

MY_KEY 12\n
MY_KEY Je suis en Master 2\n

Exemples d'événement non valident:

MY_KEY 3\n
my_Key: 1\n

PARTIE HELP

Pour afficher la liste des actions disponibles dans la salle d'attente, un client pourra tout simplement envoyer au serveur la clef "HELP"

PARTIE DÉCONNEXION

Un client désirant se déconnecter du serveur n'aura qu'à envoyer la clef "QUIT". Il sera alors déconnecté du serveur et ce dernier gèrera sa déconnexion.

PARTIE AFFICHAGE DE LA LISTE DES JOUEURS

Pour afficher la liste des joueurs disponibles avec qui il est possible de lancer une partie, un client pourra envoyer au serveur la clef "LIST".

Cette liste sera constituée :

- d'une colonne correspondant aux identifiants des joueurs
- (optionnellement d'une colonne correspondant aux noms des joueurs (pseudo), dans une version ultérieure)
- d'une colonne correspondant aux invitations envoyées par les autres joueurs

Exemple : (pour 3 joueurs d'identifiants 1, 2 et 3, avec une invitation de J1 envoyée à J2)

- Si J2 envoie au SERVEUR : "LIST"

- Alors SERVEUR envoie à J2 :

id	invitation
1	X
3	

- Si J3 envoie au SERVEUR : "LIST"

- Alors SERVEUR envoie à J3 :

id	invitation
1	
2	

- Si J1 envoie au SERVEUR : "LIST"

- Alors SERVEUR envoie à J1 :

id	invitation
2	
3	

PARTIE CONNEXION AVEC UN AUTRE JOUEUR

- La connexion avec un autre joueur se fait via la clef "CONN" suivi d'un espace, puis de l'identifiant du joueur (référéncé lors de l'affichage des joueurs au moyen de la commande "LIST").

Exemple :
CONN 2

- Si le joueur donne un identifiant n'étant pas référencé, ou bien son propre identifiant, le serveur lui répond avec un message d'erreur ([-*-*-* PARTIE MESSAGE D'ERREUR DU SERVEUR -*-*-*])

Exemple : (deux joueurs sont présents d'identifiant 1 et 2)

J1 envoie au SERVEUR : "CONN 2" => OK

J1 envoie au SERVEUR : "CONN 3" => MESSAGE D'ERREUR DU SERVEUR car il n'y a pas de joueur d'identifiant 3 disponible

J1 envoie au SERVEUR : "CONN 1" => MESSAGE D'ERREUR DU SERVEUR car le joueur 1 ne peut pas s'inviter lui-même

- Si le joueur fait une demande valide de connexion avec un autre joueur, le serveur le notifie en lui envoyant sur sa thread de lecture le message "CONNECT_OK",

- Pour que le joueur invité accepte une invitation, il procède comme suit :

- Il demande au serveur de lui afficher la liste des joueurs et des invitations liées par le biais de la commande "LIST"

- Le joueur invité peut alors voir quel(s) joueur(s) l'a/ont invité

- Il envoie alors au serveur le mot-clef "ACPT" suivi d'un espace et de l'identifiant d'un joueur l'ayant invité.

- Le serveur notifie alors les deux joueurs du début de partie et celle-ci commence.

Exemple : (trois joueurs sont présents d'identifiant 1, 2 et 3, et le joueur 1 invite le joueur 2 à jouer)

- Si J2 envoie "LIST" au SERVEUR, il va recevoir :

id | invitation

1 | X

3 |

- Si J3 envoie "LIST" au SERVEUR, il va recevoir :

id | invitation

1 |

2 |

- Si J1 envoie "LIST" au SERVEUR, il va recevoir :

id | invitation

2 |

3 |

- Si J2 envoie au SERVEUR "ACPT 1" :

alors le SERVEUR envoie à J1 "Début de la partie avec J2..."

et il envoie à J2 "Début de la partie avec J2..."

Et la partie commence

- Si J2 envoie au SERVEUR "ACPT 3" :
alors le SERVEUR envoie à J2 un message d'erreur

- Il est également possible pour un joueur ayant envoyé une invitation de l'annuler via la clef "REFU".

(optionnel : Dans ce cas-là, le serveur notifie le joueur ayant reçu l'invitation en lui envoyant le message "CANCEL_FROM id_joueur_hôte")

Exemple : (deux joueurs sont présents d'identifiant 1 et 2)

1. J1 envoie au SERVEUR : "CONN 2"

2. (optionnel) SERVEUR envoie à J2 un message contenant le port de l'hôte"

3. J1 envoie au SERVEUR : "REFU"

4. (optionne) SERVEUR envoie à J2 un message indiquant que le joueur 2 a refusé l'invitation

La partie commence

Après la connexion à l'hôte, les joueurs envoient leur combinaison à deviner avec le mot clef PASSWORD.

Le serveur envoie PASSWORD:

Les clients doivent envoyer par exemple PASSWORD RVBO

Tant que les combinaisons ne sont pas valides (ex : PASSWORD RVBZ), le serveur redemande d'envoyer une combinaison.

La partie se déroule en manche, où à chacune d'elle, le serveur demande aux clients d'envoyer leur tentative de deviner le code de l'adversaire au moyen du mot-clef "ATTEMPT".

Le client qui reçoit "ATTEMPT:" devra alors répondre au serveur avec "ATTEMPT" suivi d'un espace et d'une combinaison de 4 caractères correspondant à 4 couleurs parmi les 8 possibles (l'ordre compte !)

- Le SERVEUR envoie aux deux joueurs :

ATTEMPT: (vide)

- Les Clients (joueurs) doivent répondre en envoyant le code "ATTEMPT" suivi d'un espace et d'une combinaison de 4 couleurs choisies dans un ordre précis :

ATTEMPT: (tentative du client, code couleur de 8 caractères)

Exemple :

- SERVEUR envoie à J1 et J2 : "ATTEMPT:"

- J2 envoie au SERVEUR : "ATTEMPT BOWP"

- J1 envoie au SERVEUR : "ATTEMPT RVBO"

- Une fois qu'un joueur a envoyé sa proposition, il ne peut plus la changer. S'il essaie, le serveur lui répond en lui disant que sa proposition a déjà été prise en compte et qu'il ne peut plus la changer.

- Une fois que les deux joueurs ont envoyé leur proposition, on passe à la manche suivante où le serveur commence par donner le résultat de leur proposition à chaque joueur, avant de leur demander de formuler une nouvelle proposition.

Le serveur est responsable de faire le ping pong entre les clients et le client d'attendre de recevoir ATTEMPT pour demander au joueur de faire un essai.

Victoire d'un des joueurs

A la fin de la manche gagnante (= où le serveur a détecté la présence d'au moins un gagnant), le serveur envoie alors un message à chacun des deux joueurs pour annoncer le(s) gagnant(s). Enfin, le serveur renvoie les deux joueurs dans la salle d'attente (= ces deux joueurs sont de nouveaux visibles dans la liste des joueurs disponibles et d'autres joueurs peuvent leur envoyer des invitations à jouer)

Si un des joueurs trouve le bon code le serveur lui envoie :

WINNING: (vide)

Le perdant reçoit:

LOSING: (code entré par le joueur gagnant)

Si les deux joueurs ont fait une égalité, le serveur leur envoie :
DRAW:

Les deux clients sont alors ramenés à la salle d'attente

Code couleur

- R: rouge
- J: jaune
- V: vert
- B: bleu
- O: orange
- W: blanc
- P: violet
- F: fuchsia

Code réponse du serveur à une proposition d'un joueur

- '-' : Mauvaise couleur
- 'X' : Bonne couleur
- 'O' : Bonne couleur et à la bonne place

Gestion des erreurs:

Les clients seront tenus de gérer les erreurs (ex: couleur entrée par l'utilisateur non-valide ou code secret invalide (8 caractères) mais le serveur pourra toujours répondre:

ERROR: (message d'erreur)

Le client pourra alors reformuler sa requête correctement.