

Tournoyons

Manuel technique

*Tout pour mettre en œuvre
un moteur de jeu*

**Nb : Ce document est commun à
Tournoyons.developpez.com et Tournoyons.com**

Version du dimanche 9 juin 2013

SOMMAIRE

I.	Présentation générale	3
II.	Principes d'implémentation	4
III.	Moteurs de jeu : Référence des paramètres de QueryString	5
IV.	Syntaxes propres à chaque jeu	8
	Bataille navale.....	9
	ChiFouMi	10
	Mastermind	11
	Morpion.....	13
	Pente.....	14
	Poker	15
	Reversi	18
	TicTacToe.....	19
V.	Questions de sécurité	20
VI.	Utilisation de serveurs internes à votre école.....	20
	Annexe A – Exemple de code source d'implémentation du moteur de ChiFouMi	21

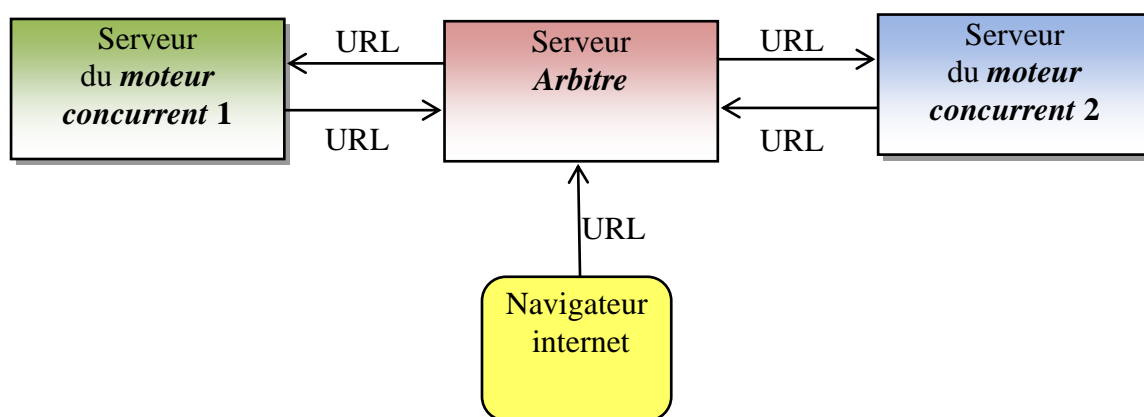
I. Présentation générale

Tournoyons est destiné à mettre en concurrence des algorithmes de joueur de jeu de plateau, dits « moteurs de jeu ».

Le serveur du site de tournois logiciels est nommé l'*Arbitre*.

Il offre un formulaire permettant d'inscrire l'URL d'un *moteur* disponible pour un type de *Jeu* (exemple *Bataille navale*)

Le site *Arbitre* permet d'organiser et de suivre en direct ou en différé des parties entre deux *Concurrents*. Ces parties sont enregistrées, et peuvent donc être revues après coup.



Un serveur *de moteur de jeu*, pour fonctionner, doit implémenter une interface appropriée, décrite ci-après.

NB : Le jeu ChiFouMi très simple à implémenter, et le jeu TicTacToe, assez simple, permettent de tester cette interface.

II. Principes d'implémentation

L'échange entre serveurs se fait via URL, les paramètres étant passés en « QueryString »

Exemple : L'*Arbitre* enverra à un *Concurrent* déclaré à l'adresse <http://equipeA.fr> la requête <http://equipeA.fr?Set=Chess&Game=CH7613&MoveId=3&Tray=Init&Referee=http://tournoyons.com/arbitre/arbitre.php>

Cette requête demande au serveur de l'équipe A d'initialiser une partie d'échecs, de nom CH7613 en tant que premier joueur (blancs), l'URL de retour pour passer le premier coup étant <http://tournoyons.com/arbitre/arbitre.php>

Dans cet exemple, le serveur de l'équipe A pourra renvoyer en retour la requête :

<http://tournoyons.com/arbitre/arbitre.php?Game=CH7613&MoveId=3&Value=E2xE4>

Indique à l'arbitre, pour la partie en question, pour le premier coup, un déplacement classique : le pion du roi avance de deux cases ([Value=E2xE4](#))

Les échanges ont toujours lieu entre *Arbitre* et *Concurrent*, jamais directement entre *Concurrents*.

A chaque coup, l'état du plateau est passé en querystring (clé « *Tray* ») par l'*Arbitre* au *Concurrent* concerné. Un serveur de *moteur* donné pourra donc choisir de ne pas stocker d'information entre deux sollicitations.

Attention, une partie peut démarrer directement sur un plateau déjà avancé (par exemple en cas de lancement avec « photo de plateau en cours de jeu »).

III. Moteurs de jeu : Référence des paramètres de QueryString

Nom du paramètre	Arbitre vers Concurrent	Concurrent vers Arbitre
<i>Gain1 & Gain2</i>	Représente le gain des joueurs	Ne pas envoyer
<i>Game</i>	Identifiant de la partie en cours	A renvoyer tel que reçu
<i>Level</i>	Niveau de la règle	Ne pas envoyer
<i>Move1</i> et <i>Move2</i>	Dernier coup joué par l'adversaire	Ne pas envoyer
<i>MoveId</i>	Identifiant du coup demandé	A renvoyer tel que reçu
<i>Opponent</i>	Identifiant de l'adversaire	Ne pas envoyer
<i>Referee</i>	Spécifie l'URL à émettre avec le coup joué	Ne pas envoyer
<i>Set</i>	A quel jeu joue-t-on ?	Ne pas envoyer
<i>Status</i>	Etat de la partie	Ne pas envoyer
<i>TimeOut</i>	Durée max. accordée au moteur	Ne pas envoyer
<i>Tray</i>	Etat du plateau	Ne pas envoyer
<i>Turn</i>	Nombre de coups	Ne pas envoyer
<i>Value</i>	Non utilisé	Coup joué à envoyer

Les paramètres *Gain1 & Gain2*

Les paramètres *Gain1 & Gain2* indiquent le gain de chacun des deux joueurs.

La syntaxe et les valeurs dépendent du jeu. (cf. chapitre V).

Ce paramètre n'est pas envoyé par l'arbitre pour le début de la partie.

A ne pas envoyer à l'arbitre.

Le paramètre *Game (Partie)*

Le paramètre *Game* (traduire *Partie*) a pour valeur un identifiant unique, crypté, fixé par *l'Arbitre*, chaîne de caractère (taille 10 max). Il peut servir au *Concurrent* de clé s'il souhaite stocker une description élaborée du jeu entre deux coups.

A renvoyer à l'arbitre tel que reçu.

Le paramètre *Level (Niveau)*

Le paramètre *Level* (traduire *Niveau*), introduit temporairement, uniquement pour le *Poker*, indique le niveau de la règle.

Les paramètres *Move1* & *Move2* (*Coup1* & *coup2*)

Les paramètres *Move1* & *Move2_* (traduire *Coup1* & *coup2*) indiquent la valeur du dernier coup (mouvement de pièce, pose de pièce, ...) joué par l'adversaire (*Move1* si l'adversaire est le premier joueur de la partie, *Move2* sinon).

La syntaxe et les valeurs dépendent du jeu. (cf. chapitre V).

Ce paramètre n'est pas envoyé par l'arbitre pour le début de la partie.

A ne pas envoyer à l'arbitre.

Le paramètre *MoveId* (*Id du coup*)

Le paramètre *MoveId_* (traduire *Id du coup*) a pour valeur une chaîne fixée par l'*Arbitre*.

A renvoyer à l'arbitre tel que reçu.

Le paramètre *Opponent* (*Adversaire*)

Le paramètre *Opponent_* (traduire *Adversaire*) est renseigné uniquement lorsque le moteur est appelé pour un match de compétition. Il identifie l'adversaire de façon permanente, ce qui peut permettre d'accumuler de l'information sur sa façon de jouer d'un match à l'autre, voire d'une compétition à l'autre ...

A ne pas envoyer à l'arbitre.

Le paramètre *Referee* (*Arbitre*) - Arbitre vers Concurrent uniquement

Le paramètre *Referee_* (traduire *Arbitre*) décrit l'URL à laquelle envoyer ses coups.

NB : si *Referee* est vide, on est dans le contexte de la page « testeur de moteur ». Afficher alors la valeur en ligne (et n'afficher que cela).

A ne pas envoyer à l'arbitre.

Le paramètre *Set* (*Jeu*) - Arbitre vers Concurrent uniquement

A ne pas envoyer à l'arbitre. A ce jour, le paramètre *Set* (traduire *Jeu*) a pour valeurs possibles :

- *BigTicTacToe* (traduire *Morpion*).
- *MasterMind*
- *Pente* (nom français, se prononce « *Peinté* »)
- *Poker*
- *Reversi*
- *Rochambeau* (traduire *ChiFouMi* ou *Pierre Feuille Ciseaux*).
NB : offre un exemple minimal d'implémentation, Voir code source Php en annexe.
- *SeaFight* (traduire *Bataille navale*).
- *TicTacToe* (*morpion 3x3*)
NB : offre un exemple simple d'implémentation.

Le paramètre *Status (Statut)* - Arbitre vers Concurrent uniquement

A ne pas envoyer à l'arbitre. Le paramètre *Status* (traduire *Statut*) permet, si la partie est terminée (en match ou en entraînement en partie « en un clic »), d'en enregistrer les caractéristiques pour ses statistiques personnelles et analyses des performances de son moteur.

Le paramètre *Status* a pour valeurs possibles :

0 : partie en cours

1 : partie gagnée en tant que premier joueur

2 : partie gagnée en tant que second joueur

3 : partie perdue en tant que premier joueur

4 : partie perdue en tant que second joueur

5 : match nul

9 : partie terminée en erreur

Si la partie est terminée, le moteur ne doit pas rappeler l'arbitre. Le paramètre *Referee* est donc absent

Le paramètre *TimeOut (Duree)* - Arbitre vers Concurrent uniquement

Le paramètre *TimeOut* indique la durée maximum autorisée, en secondes, entre l'émission par l'arbitre de l'URL d'appel du moteur, et la réception par l'arbitre de l'URL retournée par le moteur. Au-delà, l'arbitre considère qu'il y a erreur et le moteur concerné perd la partie.

Durée peut valoir entre 4s et 54s. Elle est constante pour une partie donnée.

Le paramètre *Tray (Plateau)* - Arbitre vers Concurrent uniquement

Le paramètre *Tray* (traduire *Plateau*) décrit l'état du plateau. La syntaxe et les valeurs dépendent du jeu. (cf. chapitre V)».

Par convention, la valeur d'un plateau vide (partie à débiter) est : « *Init* »

Le paramètre *Turn (Tour)* - Arbitre vers Concurrent uniquement

Le paramètre *Turn* (traduire *Tour*) indique le nombre de tours pour la partie en cours, plus précisément le nombre de coups joués (incluant le coup courant).

NB : *Turn* n'est pas valable pour le Poker, où on s'appuie sur l'étape en cours.

Donc si *Turn* est impair, le moteur appelé est premier joueur, et recevra (si $Turn > 1$) la valeur du coup joué par l'adversaire dans **Move2**.

A l'inverse, si *Turn* est pair, le moteur appelé est second joueur, et recevra la valeur du coup joué par l'adversaire dans **Move1**.

A ne pas envoyer à l'arbitre.

Le paramètre *Value (Valeur)*

Le paramètre *Value* (traduire *Valeur*), à envoyer à l'arbitre décrit un coup (mouvement de pièce, pose de pièce, ...). La syntaxe et les valeurs dépendent du jeu. (cf. chapitre suivant).

Non utilisé en émission par l'arbitre.

IV. Syntaxes propres à chaque jeu

Parmi les paramètres de QueryString échangés, certains ont une syntaxe dépendante du jeu. Il s'agit de :

Nom du paramètre	Fonction
Gain1 & gain2	Gain de chaque joueur
Move1 & move2	Dernier coup joué par l'adversaire
Tray	Etat du plateau à jouer
Value	Coup joué par le moteur

Voyons la syntaxe de la valeur de ces paramètres pour les différents jeux implémentés à ce jour :

- Bataille navale
- ChiFouMi (alias « Pierre Feuille Ciseaux)
- Mastermind
- Morpion
- Pente
- Poker
- Reversi
- TicTacToe

Bataille navale

A0	B0	C0
A1	B1	C1
A2	B2	C2

Le plateau (10x10) est numéroté verticalement de 0 à 9, horizontalement de A à J (cf illustration)

- **Case** désigne une case via une chaîne de deux caractères de 'A0' à 'J9' suivant la case désignée, la première lettre désignant l'abscisse (horizontale). Exemple : **C5**
- **Flotte** représente un ensemble de *navires*, éventuellement vide. Il vaut autant de chaînes de 5 caractères, concaténées.
NB : Au premier tour, le moteur 'Bataille navale' produit la flotte. Ensuite il tire.
- **Navire** est représenté par cinq caractères : **Direction, Taille, Abscisse, Ordonnée, Etat**
Exemple : **V3C5I** désigne un navire vertical de 3 cases posé en C5, intact
 - **Direction** vaut 'V' pour « Vertical » ou 'H' pour « Horizontal »
 - **Taille** indique le nombre de cases occupées et vaut donc de '2' à '5'
 - **Abscisse** (coordonnée horizontale la plus à gauche) vaut de 'A' à 'J'
 - **Ordonnée** (coordonnée verticale la plus haute) vaut de '0' à '9'
 - **Etat** indique l'état du navire. Il vaut : '1' si le navire est *intact*, '2' si il est *touché*, '3' si il est *coulé*
- **Tirs** indique les cases qui ont été tirées, via une chaîne de 100 caractères, concaténation des dix lignes horizontales, de la plus haute à la plus basse.
Chaque caractère vaut :
 - '0' si la case n'a pas été tirée,
 - '1' si elle a été tirée, dans l'eau,
 - '2' si elle a été tirée, touché,
 - '3' idem, coulé (toutes les cases correspondant au navire coulé ont alors cette valeur).
- **Valeur** désigne une case tirée avec le résultat du tir, via une chaîne de trois caractères : **Case** sur deux caractères et le résultat ('1', '2' ou '3') en troisième caractère. Exemple : **C53**

Paramètre	Syntaxe de la valeur du paramètre
Gain1 & gain2	Si partie terminée, Flotte de chaque joueur, sinon, sans objet
Move1 & move2	Valeur pour le plateau du joueur concerné
Tray	Concaténation des Tirs des plateaux du joueur concerné.
Value	Si Turn vaut 1 ou 2, Value vaut Flotte , sinon Value vaut Case

ChiFouMi









- *Valeur* vaut
 - 1 pour Pierre
 - 2 pour Feuille
 - 3 pour Ciseaux
- *Score* vaut de 0 à 5

Paramètre	Syntaxe de la valeur du paramètre
<i>Gain1 & gain2</i>	<i>Score</i>
<i>Move1 & move2</i>	<i>Valeur</i>
<i>Tray</i>	Sans objet, chaque coup étant réputé indépendant des précédents
<i>Value</i>	<i>Valeur</i> . Ce paramètre n'est pas passé dans le sens Arbitre vers joueur

Mastermind

Chaque plateau peut contenir huit lignes de 4 pions de couleur, et les marques associées (une marque rouge par pion bien placé, et, pour les pions restants, une marque blanche par pion mal placé)

- Chaque plateau est représenté par une chaîne concaténant la description des lignes remplies : de 0 à 8 *ValeurLigne*.
- *ValeurLigne* contient deux caractères de marque et quatre caractères de couleurs, un par pion. Le premier caractère de marque indique le nombre de pions bien placés, Le second caractère de marque indique le nombre de pions mal placés, Pour chaque pion, la couleur fournie est désignée par :

- 'A' :  blanc,
- 'B' :  bleu,
- 'C' :  vert,
- 'D' :  jaune,
- 'E' :  orange,
- 'F' :  rouge,
- 'G' :  rose,
- 'H' :  violet.

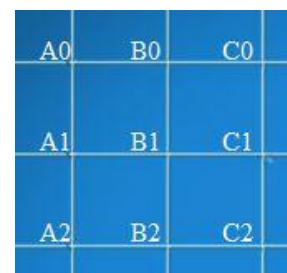
Exemple '30GCCH' indique une ligne « rose, vert, vert, violet » où trois pions sont bien placés, et aucun n'est mal placé.

- Le *Code* caché de chaque joueur est une chaîne de quatre caractères codant les 4 pions comme ci-dessus.

Paramètre	Syntaxe de la valeur du paramètre
<i>Gain1</i> & <i>gain2</i>	Sans objet
<i>Move1</i> & <i>move2</i>	Sans objet
<i>Tray</i>	Concaténation des <i>Lignes</i> du plateau adverse.
<i>Value</i>	Si <i>Turn</i> vaut 1, <i>Value</i> vaut <i>Code</i> , sinon <i>Value</i> vaut <i>ValeurLigne</i>

Morpion

Le plateau de 10x10 est numéroté verticalement de 0 à 9, horizontalement de A à J (cf illustration)



A0	B0	C0
A1	B1	C1
A2	B2	C2

- **Alignement** vaut une chaîne de quatre caractères indiquant les coordonnées de la première et de la dernière case d'un alignement réalisé.
Ex : **D3H3** indique un alignement horizontal de **D3** à **H3**
- **AlignementsJoueur** représente les alignements réalisés par le joueur. Il contient zéro à plusieurs **Alignement**.
- **Plateau** représente l'état du plateau, via une chaîne de cent caractères, concaténation des dix lignes horizontales, de la plus haute à la plus basse
Chaque caractère vaut :
 - **0** si la case est vide,
 - **1** si elle a été jouée par le premier joueur,
 - **2** si elle a été jouée par le second joueur,
- **Valeur** désigne une case via une chaîne de deux caractères de 'A0' à 'J9' suivant la case désignée, la première lettre désignant l'abscisse (horizontale). Exemple : **C5**

Paramètre	Syntaxe de la valeur du paramètre
<i>Gain1 & gain2</i>	<i>AlignementsJoueur</i>
<i>Move1 & move2</i>	<i>Valeur</i>
<i>Tray</i>	<i>Plateau</i>
<i>Value</i>	<i>Valeur</i>

Pente

Le plateau (19x19) est numéroté horizontalement et verticalement de A à S. (cf illustration)



- **Plateau** vaut une chaîne de 361 caractères, concaténation des 19 lignes horizontales, de la plus haute à la plus basse. Chaque caractère vaut :
 - **0** si la case est vide,
 - **1** si elle a été jouée par le premier joueur,
 - **2** si elle a été jouée par le second joueur,
- **PrisesJoueur** vaut de 0 à 5 : nombre de paires de pions pris à l'adversaire par le joueur
- **Valeur** désigne une case via une chaîne de deux caractères de 'AA' à 'SS' suivant la case désignée, la première lettre désignant l'abscisse (horizontale). Exemple : **CH**

Paramètre	Syntaxe de la valeur du paramètre
<i>Gain1 & gain2</i>	<i>PrisesJoueur</i>
<i>Move1 & move2</i>	<i>Valeur</i>
<i>Tray</i>	<i>Plateau</i>
<i>Value</i>	<i>Valeur</i>

Poker

La règle ayant évolué suite aux remarques des joueurs, on distingue deux niveaux. Pour un match donné, il est en niveau « 2 » si et seulement si les deux moteurs en lice le codent. Le query *Level* correspondant vaut donc 1 ou 2.

Niveau 1 de la règle

Une partie se joue en douze coups aboutis (c'est-à-dire sans qu'un des deux joueurs se soit couché), avec un maximum de 99 coups en tout.

La situation du jeu est décrite à un instant donné en particulier par le *Plateau*, à savoir :

- Le *rang* du coup en cours (1 à 99),
- Le *nombre de coups aboutis* (0 à 11),
- *L'étape* (cf avancement plus loin),
- Les *deux cartes reçues par chaque joueur* (éventuellement cachées),
- Les *cartes du tableau* (communes aux deux joueurs),
- Les *misses successives* effectuées par chaque joueur pour chaque tour d'enchères.

Plus précisément, le *Plateau* est une chaîne de caractères concaténant :

Le *rang* du coup en cours est représenté par deux chiffres (*01* à *99*). Vis-à-vis des règles (cf avancement ci-dessous), si le rang est impair, le donneur est le premier joueur, sinon le donneur est le second joueur.

Le *nombre de coups aboutis* est représenté par deux chiffres (*00* à *11*).

L'*étape* est représentée par un caractère (*1* à *9*).

Les deux cartes reçues par chaque joueur forment une chaîne de caractères concaténation de quatre cartes : les deux cartes reçues par le premier joueur (donneur ou non) puis les deux cartes reçues par le second joueur.

Chaque carte est représentée par deux caractères : *Couleur Valeur* pour une carte visible ou « ?? » pour une carte cachée.

Couleur vaut :

- *S*: Pique (Spade en anglais) ou
- *C*: Trèfle (Club en anglais) ou
- *H*: Cœur (Heart en anglais) ou
- *D*: Carreau (Diamond en anglais)

Valeur vaut

- *K* : Roi (King en anglais) ou
- *Q* : Dame (Queen en anglais) ou
- *J* : Valet (Jack en anglais) ou
- *T* : Dix (Ten en anglais) ou
- *1* à *9* (*1* pour l'As)

Les **cartes du tableau** (communes aux deux joueurs)

Suivant l'avancement du coup, il y a 0, 3,4 ou 5 cartes dans le tableau. Chaque carte est représentée par deux caractères : **Couleur Valeur** (cf ci-dessus).

Nb : les tirages sont totalement indépendants (pas de sabot).

Les mises successives effectuées par chaque joueur pour chaque tour d'enchères.

Chaque tour d'enchère, correspondant à l'avancement du coup, est représenté par la lettre « **M** » suivie par la mise totale à ce stade (1 à 100) du premier joueur puis, après un tiret (45 en ASCII), par la mise totale à ce stade (1 à 100) du second joueur. Si un seul joueur a misé (cf avancement ci-dessous), la mise de son adversaire est indiquée par un '?' précédant son total de mises à ce stade.

Exemple : **M?10-12** indique que le premier joueur doit miser, en étant second à miser à cette étape.

NB : pour chaque tour d'enchère, s'il y a des relances, on trouve simplement le dernier total.

NB2 : si un joueur s'est couché, sa mise totale vaut 0.

L'avancement d'un coup est constitué des étapes suivantes (si un joueur se couche on passe directement à l'étape 9)

Blind : Le donneur mise automatiquement un jeton, et son adversaire deux jetons

--- Distribution de deux cartes à chaque joueur, non visibles de l'autre joueur ---

Etape 1 - Mise pré-flop du donneur (y compris après relance en étape 2). Il peut suivre à deux jetons ou relancer

Etape 2 - Mise pré-flop de l'adversaire du donneur ou pas (check) de façon à égaliser ou relancer --- Distribution du flop : trois cartes visibles ---

Etape 3 - Mises flop : L'adversaire du donneur mise (y compris après relance), ou pas (check)

Etape 4 - Mises flop : Le donneur mise (ou check) de façon à égaliser ou relancer

--- Distribution du tournant : une quatrième carte visible ---

Etape 5 - Mises du tournant : L'adversaire du donneur mise (y compris après relance), ou pas

Etape 6 - Mises du tournant : Le donneur mise (ou check) de façon à égaliser ou relancer

--- Distribution de la rivière : cinquième carte visible ---

Etape 7 - Mises de la rivière : L'adversaire du donneur (y compris après relance), ou pas

Etape 8 - Mises de la rivière : Le donneur mise (ou check) de façon à égaliser ou relancer

Etape 9 – Abattage des cartes et verdict

NB : le score en cours (**Gain1 & gain2**) participe à l'évaluation de la situation du jeu et peut influencer les mises.

NB2 : En étape 9, les deux cartes individuelles de chaque joueur sont masquées si un joueur s'est couché et visibles sinon, en affichage comme dans le **Plateau** reçu par les moteurs. Le moteur recevant l'étape 9 doit simplement acquiescer, peu importe alors **Value**.

Paramètre	Valeur du paramètre
Gain1 & gain2	Cumul des gains (négatif si perte) des coups terminés
Move1 & move2	Valeur mise
Tray	Coup en cours
Value	Mise totale du joueur à miser (0 si abandon)

Niveau 2 de la règle

Par rapport au niveau 1 :

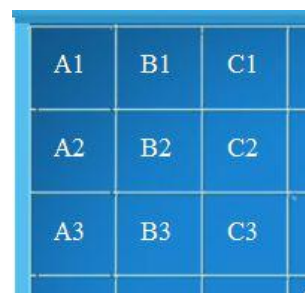
1. Chacun part avec un **capital** de **200** jetons,
2. La mise sur un coup donné est plafonnée à 100 jetons maximum et est aussi plafonnée au capital restant le plus faible,
3. La partie s'arrête sur l'une des conditions suivantes : si un joueur a perdu son capital (devenu inférieur à la grosse blinde), ou si **quinze** coups aboutis sont joués, ou, à l'extrême, au bout de 99 coups,
4. La **petite blinde** démarre à un jeton, puis **augmente** de deux jetons tous les huit coups (la grosse blinde étant double de la petite),
5. La **relance minimum** égale la petite blinde (sous réserve du point 2)

Paramètre	Valeur du paramètre
<i>Gain1 & gain2</i>	Capital de chaque joueur

NB : A terme, une fois migrés l'essentiel des moteurs en compétition, nous prévoyons que seul le niveau deux soit conservé.

Reversi

Le plateau (8x8) est numéroté en colonnes (comme Excel) de A à H, et en lignes de 1 à 8.(cf. illustration ci-contre).



A1	B1	C1
A2	B2	C2
A3	B3	C3

- **Plateau** vaut une chaîne de 64 caractères, concaténation des 8 lignes horizontales, de la plus haute à la plus basse
Chaque caractère vaut :
 - **0** si la case est vide
 - **3** si elle contient un pion noir,
 - **4** si elle contient un pion blanc,
- **Valeur** soit
 - Désigne une case via une chaîne de deux caractères de '**A1**' à '**H8**' suivant la case désignée, la première lettre désignant l'abscisse (horizontale). Exemple : **C5**, soit
 - Vaut '**XX**' indiquant « Je passe » (si aucune case ne peut être jouée)

Paramètre	Syntaxe de la valeur du paramètre
Gain1 & gain2	Sans objet
Move1 & move2	Valeur
Tray	Plateau
Value	Valeur
Legend	Si Value=XX , Legend = « Les noirs passent » (ou traduction) ou « Les blancs passent » (ou traduction) Si partie non terminée, sans possibilité de jeu pour le joueur actif, Legend = « Je passe » (ou traduction)

TicTacToe

Le plateau est numéroté de 1 à 9.(cf. illustration ci-contre).

1	2	3
4	5	6
7	8	9

- **Plateau** vaut une chaîne de neuf caractères correspondants aux neuf cases dans l'ordre de l'illustration.
Chaque caractère vaut
 - **0** si la case est vide,
 - **1** si elle a été jouée par le premier joueur,
 - **2** si elle a été jouée par le second joueur,
- **Valeur** vaut de '1' à '9' suivant la case désignée (cf. illustration)

Paramètre	Syntaxe de la valeur du paramètre
<i>Gain1 & gain2</i>	Non utilisé
<i>Move1 & move2</i>	<i>Valeur</i>
<i>Tray</i>	<i>Plateau</i>
<i>Value</i>	<i>Valeur</i>

V. Questions de sécurité

Concernant les moteurs de jeu

Cela a du sens, par précaution, de vérifier au sein de votre moteur de jeu, à la réception d'une URL que l'IP de l'émetteur émetteur (arbitre) = IP de Tournoyons (à ce jour 87.98.130.52).

VI. Utilisation de serveurs internes à votre école

Pour les moteurs de jeu

Les échanges entre arbitre et moteur de jeu, décrits plus haut, sont basés sur de simples URL croisées, asynchrones, et induisent donc un hébergement plutôt de type externe, comme ceux permettant aux élèves de mettre en ligne leurs propres pages Web. (Ce dispositif permet du coup une implémentation par des individuels comme des structures, avec les moyens les plus divers). Dès lors qu'une école souhaite concourir via des serveurs de son réseau interne, une architecture spécifique à l'école doit être envisagée. Dans tous les cas, il n'y a qu'un seul arbitre prévu, hébergé sur le serveur physique de Tournoyons.com.

En pratique, Tournoyons peut parfaitement appeler par exemple un Webservice que vous exposeriez sur une DMZ, et négocier avec vous une implémentation précise des paramètres en XML, JSON ou autres. Une clé de sécurité et son mécanisme d'actualisation peuvent être convenus entre développeurs. Un tel Webservice aurait a priori une seule entrée pour les moteurs de jeu, portant l'identification du moteur concerné chez vous, et le contexte de jeu.

La fonction correspondante retournerait de façon synchrone le coup joué, avec le même timeout (prochainement personnalisable) que pour la méthode publique.

Votre documentation interne d'accès au service par les moteurs de vos élèves et professeurs pourrait leur être mise à disposition via Tournoyons, dès lors qu'ils sont dûment identifiés. Plus généralement vous pouvez compter sur la totale collaboration de Tournoyons, et de sa volonté de s'adapter autant qu'il faudra à votre contexte.

Annexe A – Exemple de code source d'implémentation du moteur de ChiFouMi

Exemple de code source PHP implémentant le moteur de ChiFouMi

```
<?php
//Entrées
$Partie=$_REQUEST['Game']; //On récupère le nom de partie
$MoveId=$_REQUEST['MoveId']; //On récupère le MoveId
$Arbitre=$_REQUEST['Referee']; //On récupère l'URL de l'arbitre
//Elaboration du jeu
$Valeur=mt_rand(1,3); //On choisit son coup
//Sorties : On construit l'URL cible
if ($Arbitre=='') {print $Valeur;} else {
$URL=$Arbitre.'?Game='.$Partie.'&MoveId='.$MoveId.'&Value='.$Valeur;
fopen($URL,"r"); //On transmet à l'arbitre
//si fopen pose problème, essayer readfile()
?>
```