

Tournoyons

Manuel technique

*Tout pour mettre en œuvre
un moteur de jeu ou
une interface graphique de jeu*

Et aussi pour ajouter un jeu

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

Version du vendredi 22 février 2013

SOMMAIRE

I.	Présentation générale	3
II.	Principes d'implémentation	4
III.	Moteurs de jeu : Référence des paramètres de QueryString	5
IV.	Interfaces graphiques de jeu : taille disponible et référence des paramètres de QueryString ..	8
V.	Moteurs et interfaces graphiques de jeu : Syntaxes propres à chaque jeu	10
	Bataille navale.....	11
	ChiFouMi	12
	Mastermind	13
	Morpion.....	14
	Pente.....	15
	Poker	16
	Reversi	18
	TicTacToe.....	19
VI.	Ajout de jeu – principes et syntaxe	20
VII.	Questions de sécurité	22
VIII.	Utilisation de serveurs internes à votre école.....	22
	Annexe A – Exemple de code source d'implémentation du moteur de ChiFouMi	23
	Annexe B – Exemple de code HTML de l'interface graphique de TicTacToe	24

I. Présentation générale

Tournoyons est destiné à mettre en concurrence des logiciels serveurs de jeux de plateau (moteurs de jeu et/ou interfaces graphiques).

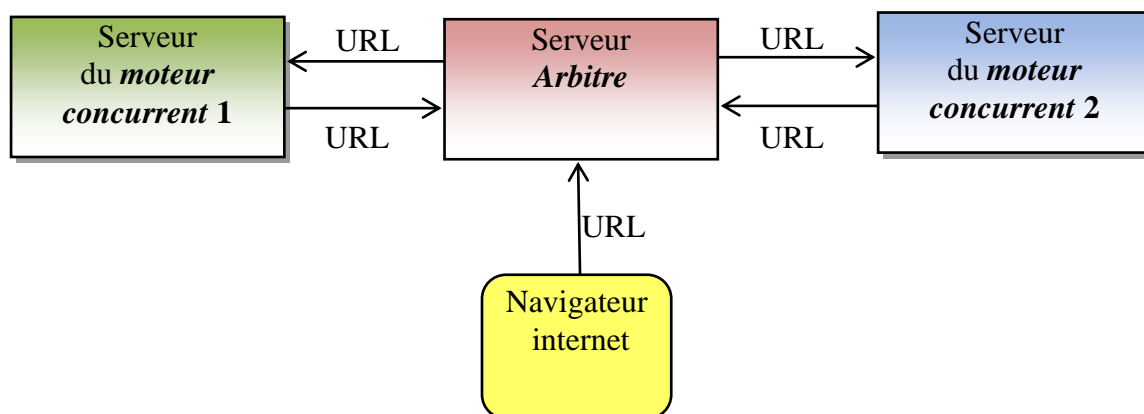
Le serveur du site de tournois logiciels est nommé l'*Arbitre*.

Il offre un formulaire permettant à des serveurs de s'inscrire en tant que :

- **moteur** disponible pour un type de *Jeu* (exemple *Bataille navale*), ou
- **interface graphique** disponible pour un type de *Jeu*.

Serveur de *moteur de jeu*

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 : Les jeux ChiFouMi et TicTacToe, très simples à implémenter, permettent de tester l'interface en tant que moteur.

Serveur d'*interface graphique*

Le site *Arbitre* permet également de visualiser une partie enregistrée via l'*interface graphique* de son choix, et de comparer ainsi les *interfaces graphiques* en concurrence.

Un serveur d'*interface graphique*, pour fonctionner, doit implémenter une interface appropriée, décrite ci-après.

II. Principes d'implémentation

Implémentation d'un serveur de *moteur de jeu*

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 »).

Implémentation d'un serveur d'*interface graphique de jeu*

L'*interface graphique* de tout jeu s'exécute dans une IFRAME du site *Arbitre*. Elle est appelée après tout changement du contexte à visualiser, ce contexte étant intégralement décrit via les paramètres passés en « QueryString »

Exemple : L'*Arbitre* enverra à un *Serveur d'Interface graphique* déclaré à l'adresse <http://serveur.fr> la requête

<http://serveur.fr?Set=Chess&Game=CH7613&Mode=standard&Name1=MissOne&Name2=MrTwo&Tray=Init&Input=One&Return=http://tournoyons.com/arbitre/retour.php>

Indique au serveur A, de produire un plateau d'échec en initialisation de partie, les « blancs » s'appelant « MissOne » et les noirs « MrTwo » et de faire jouer le premier coup aux « blancs ». La valeur du coup joué sera retournée en querystring de l'URL :

<http://tournoyons.com/arbitre/retour.php>

Attention : l'interface graphique doit être internationale, donc sans textes ajoutés.

NB : Au sein de *S'entraîner*, l'interface *Tournoyons* utilise une communication locale au site, tandis que *Modèle* est distante, à l'instar d'une interface développée. La différence est sensible avec le navigateur Chrome.

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>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>Login</i>	Non utilisé	Authentification du moteur
<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>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.

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 *Login*

Pour tourneyons.developpez.com uniquement ;

Le paramètre **Login** a pour valeur une chaîne indiquant le login ayant déclaré le moteur à Tournoyons. L'arbitre vérifie la correspondance pour authentifier la requête.

Le paramètre *MoveId* (*Id du coup*)

Le paramètre **MoveId** (traduire *Id du coup*) a pour valeur un entier fixé 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 *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).

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. Interfaces graphiques de jeu : taille disponible et référence des paramètres de QueryString

La taille de l'IFRAME au sein duquel s'affiche l'interface graphique est :
width='736' ; height='660'

L'inventaire des paramètres de Querystring est :

Nom du paramètre	Fonction
<i>Color</i>	Couleur du cadre environnant
<i>Count</i>	Nombre de coups joués
<i>Gain1 & Gain2</i>	Représente le gain du premier (ou second) joueur
<i>Input</i>	Solliciter un coup et retourner la valeur saisie
<i>Legend</i>	Texte à afficher, commentant la phase de jeu
<i>Method</i>	Méthode d'affichage de la partie
<i>Move1 & Move2</i>	Montrer le coup du premier (ou second) joueur
<i>Mute</i>	Couper les sons
<i>Name1 & Name2</i>	Nom du premier (ou second) joueur
<i>Set</i>	A quel jeu joue-t-on ?
<i>Tray</i>	Etat du plateau
<i>URLReturn</i>	Spécifie l'URL à émettre avec le coup joué
<i>Wait</i>	Délai entre l'affichage d'un coup et l'affichage des impacts
<i>WebTest</i>	Facultatif - Destiné aux tests, demande de jouer « à la main »

Les fonctions à implémenter sont (voir exemple en annexe) :

- Paramètre **Color** = **Couleur** : Indique la couleur, (au format #HHHHHH, sans mettre le #), du cadre environnant. Cela permet à l'interface qui le souhaite de se fondre dans le cadre environnant.
- Paramètre **Count** = **Comptage** : Afficher « [decodeURI\(Comptage\)](#) », explicitant le nombre de coups joués.
- Paramètre **Gain1** = **Gain** : Afficher la mention du score du premier joueur, déduite de la valeur **Gain**.
Paramètre **Gain2** = **Gain**: Idem second joueur.
NB : La syntaxe et les valeurs de **Gain** dépendent du jeu. (cf. chapitre suivant).
- Paramètre **Input** = **Joueur** : Si **Joueur=One**, solliciter et saisir le coup du premier joueur, sinon (**Joueur=Two**) le coup du second joueur. Pas de saisie en l'absence de ce paramètre.
NB : La valeur du coup est à retourner via une URL en IFRAME (cf paramètre **URLReturn**).

- Paramètre **Legend** = **Texte** : Afficher « [unescape](#) (**Texte**) », texte commentant la phase de jeu.
- Paramètre **Method** = **Methode**
 Suivant les valeurs de **Methode**, la méthode d’affichage de la partie est :
 - **Double** : Affichage de deux plateaux de jeu, celui vu du premier joueur, et celui vu du second (mode utilisé uniquement pour *Bataille Navale*)
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Prepare** : Préparation de partie. (mode utilisé uniquement pour *Bataille Navale*)
 - **Standard** : Affichage standard de la partie en cours (par défaut)
- Paramètre **Move1** = **Valeur** : Montrer le premier joueur jouant le coup de valeur **Valeur**.
 Paramètre **Move2** = **Valeur** : Idem pour le second joueur
 NB : La syntaxe et les valeurs de **Valeur** dépendent du jeu. (cf. chapitre suivant).
- Paramètre **Mute** = **text** : couper les sons (ignorer **text**).
- Paramètre **Name1** = **Nom** : Afficher « [unescape](#) (**Nom**) », nom du premier joueur.
 Paramètre **Name2** = **Nom** : Idem second joueur.
- Paramètre **Set** = **Jeu** : Jeu auquel on joue.
 Valeurs possibles, à ce jour :
 - **BigTicTacToe** (traduire *Morpion*)
 - **MasterMind**
 - **Pente** (nom français, se prononce « *Peinté* »)
 - **Poker**
 - **Reversi**
 - **Rochambeau** (traduire *ChiFouMi* ou *Pierre Feuille Ciseaux*)
 - **Seafight** (traduire *Bataille navale*)
 - **TicTacToe** (morpion 3x3)
- Paramètre **Tray** = **Plateau** : Disposer le plateau d’après **Plateau**.
 NB : La syntaxe et les valeurs de **Plateau** dépendent du jeu. (cf. chapitre suivant).
- Paramètre **URLReturn** = **URL**. Pour retourner la valeur d’un **input** ou d’une préparation de partie (**Mode=Prepare**.)
 L’interface doit disposer d’un IFRAME, au sein duquel elle appellera l’URL **URL**, avec le QueryString « **?ReturnValue=Valeur** » (ou **Prepare**).
 NB : La syntaxe et les valeurs de **Valeur** et **Prepare** dépendent du jeu. (cf. chapitre suivant).
- Paramètre **Wait** = **Valeur**. Délai (en dixièmes de secondes) pour l’interface à respecter entre l’affichage d’un coup et l’affichage des impacts. (Ex : Reversi, attente entre l’affichage du coup, et le retournement des pions impactés)
- Paramètre **WebTest** = **yes** : Facultatif - Demande de pouvoir jouer ‘à la main’, et de pouvoir récupérer à tout moment le contexte et la dernière valeur. Ce paramètre est destiné au testeur d’interface et au testeur de moteur. (Voir bouton ‘Web Test’ de ces testeurs).

V. Moteurs et interfaces graphiques de jeu : Syntaxes propres à chaque jeu

Parmi les paramètres de QueryString de moteur et d'interface, certains ont une syntaxe dépendante du jeu. Il s'agit de :

Nom du paramètre	Usage	Fonction
Gain1 & gain2	Moteur	Gain de chaque joueur
	Interface	Afficher la mention du score de chaque joueur, déduite de la valeur du paramètre.
Move1 & move2	Moteur	Dernier coup joué par l'adversaire
	Interface	Montrer le joueur jouant le coup de valeur du paramètre
Method	Interface	Méthode d'affichage de la partie
Tray	Moteur	Etat du plateau à jouer
	Interface	Etat du plateau à afficher
Value	Moteur	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	Flotte du joueur concerné, pour l'interface uniquement
Move1 & move2	Valeur pour le plateau du joueur concerné
Tray	Pour un moteur, concaténation des Tirs des plateaux du joueur concerné. Pour l'interface : concaténation des Tirs des plateaux du joueur un et du joueur deux, séparés par '***'.
Value	Si Turn vaut 1, Value vaut Flotte , sinon Value vaut Case

- Paramètres possibles de **Méthod** (pour l'interface)
 - **Double** : Si jeu entre deux internautes ou entre deux moteurs
 - **End** : Fin de partie
 - **Inverse** : Si jeu entre l'internaute, premier joueur, et un moteur
 - **None** : Entre deux parties, pas de plateau

- **Prepare** : En préparation de partie. pour chaque internaute en jeu
Paramètres complémentaires :
 - **Legend** pour le nom du bouton « *Valider* », pour terminer la préparation.
 - **URLReturn** pour retourner le plateau préparé
- **Standard** : Si jeu entre l'internaute, second joueur, et un moteur

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
Gain1 & gain2	Score
Move1 & move2	Valeur
Tray	Sans objet, chaque coup étant réputé indépendant des précédents
Value	Valeur . Ce paramètre n'est pas passé dans le sens Arbitre vers joueur

- Paramètres possibles de **Méthod**
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)

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.

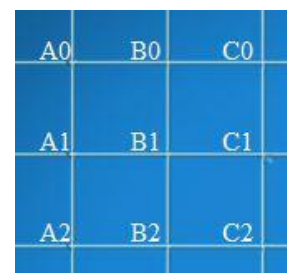
- Le **Code** caché de chaque joueur est une chaîne de quatre caractères codant les 4 pions comme ci-dessus.

Paramètre	Syntaxe de la valeur du paramètre
Gain1 & gain2	Code du joueur, pour l'interface uniquement
Move1 & move2	Sans objet
Tray	Pour l'interface : concaténation des Lignes du plateau du premier joueur, puis de '***', puis des Lignes du plateau du second joueur. Pour le moteur, concaténation des Lignes du plateau adverse.
Value	Si Turn vaut 1, Value vaut Code , sinon Value vaut ValeurLigne

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Prepare** : En préparation de partie. pour chaque internaute en jeuParamètres complémentaires :
 - **URLReturn** pour retourner le plateau préparé

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	

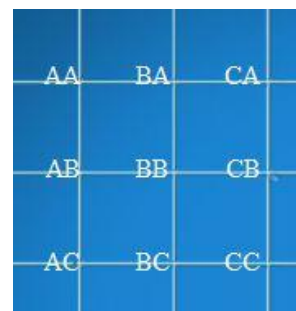
- **Alignement** vaut une chaîne de quatre caractères indiquant les coordonnées de la première et de la dernière case d'un alignement réalisé.
Ex : **D3H3** indique un alignement horizontal de **D3** à **H3**
- **AlignementsJoueur** représente les alignements réalisés par le joueur. Il contient zéro 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>

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)

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,
- **3** si gagnante premier joueur,
- **4** si gagnante second joueur,
- **5** si pion premier joueur à supprimer,
- **6** si pion second joueur à supprimer,
- **9** si interdite (pour le second coup des blancs).

NB : **Plateau** reçu par un moteur contient uniquement des **0**, **1** ou **2**, les autres valeurs étant destinées uniquement à **Plateau** transmis à une interface.

- **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
Gain1 & gain2	PrisesJoueur
Move1 & move2	Valeur
Tray	Plateau
Value	Valeur

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)

Poker

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 **misés 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

- **A** : As (Ace en anglais) ou
- **K** : Roi (King en anglais) ou
- **Q** : Dame (Queen en anglais) ou
- **J** : Valet (Jack en anglais) ou
- **T** : Dix (Ten en anglais) ou
- **2** à **9**

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.

NB3 : à une interface, seule la dernière étape de mise est transmise,

L'avancement d'un coup est constitué des étapes suivantes (sauf si un joueur se couche)

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 3). Sa mise totale doit être supérieure à deux jetons.

Etape 2 - Mise *pré-flop* de l'adversaire du donneur.

Distribution du *flop* : trois cartes visibles

Etape 3 - Mises *flop* : L'adversaire du donneur (y compris après relance en étape 5)

Etape 4 - Mises *flop* : Le donneur mise

Distribution du *turnant* : une quatrième carte visible

Etape 5 - Mises du *turnant* : L'adversaire du donneur mise (y compris après relance)

Etape 6 - Mises du *turnant* : Le donneur mise

Distribution de la *rivière* : cinquième carte visible

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

Etape 8 - Mises de la *rivière* : Le donneur mise

Etape 9 – Abattage des cartes et verdict

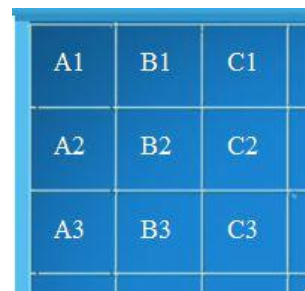
NB : le score en cours (**Gain1** & **gain2**) participe également à l'évaluation de la situation du jeu et peut influencer les mises.

Paramètre	Syntaxe de la valeur du paramètre
Gain1 & gain2	Cumul des gains (négatif si perte) des coups terminés
Move1 & move2	Valeur misee
Tray	Pour une interface : description du coup en cours (avec uniquement la mise totale de chaque jouer) Pour un moteur : coup en cours
Value	Mise totale du joueur à miser (0 si abandon)

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)

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
 - **1** si la case est vide, autorisant la pose d'un pion noir (premier joueur),
 - **2** si la case est vide, autorisant la pose d'un pion blanc,
 - **3** si elle contient un pion noir,
 - **4** si elle contient un pion blanc,
 - **7** si pion à retourner de blanc à noir,
 - **8** si pion à retourner de noir à blanc.

NB : **Plateau** reçu par un moteur contient uniquement des **0**, **3** et **4**, les autres valeurs étant destinées uniquement à **Plateau** transmis à une interface.

- **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)

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)

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,
 - **3** si gagnante premier joueur,
 - **4** si gagnante second joueur
- **Valeur** vaut de '**1**' à '**9**' suivant la case désignée (cf. illustration)

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

- Paramètres possibles de **Méthod**
 - **End** : Fin de partie
 - **None** : Entre deux parties, pas de plateau
 - **Standard** (par défaut)


VI. Ajout de jeu – principes et syntaxe

Lorsque l'on utilise un « jeu ajouté », Tournoyons appelle chacun à son tour :

- L'interface modèle (ou l'interface choisie),
- L'arbitre,
- Le moteur modèle (ou le ou les moteurs choisis).

Tous ces programmes ont été programmés par des utilisateurs de Tournoyons.

Tous ces programmes conversent avec Tournoyons par URL.

Pour les moteurs et interfaces, voir les chapitre précédents, plus la syntaxe spécifique au jeu, censé être décrite dans la fiche de jeu (accessible via le  en face du nom du jeu dans la page « Nouvelle partie »).

Attention, l'arbitre doit **impérativement** émettre son URL (de retour à Tournoyons) depuis son frame d'appel (à priori donc, par un script de type `location.href=URL`).

Arbitre : référence des paramètres de QueryString échangés avec Tournoyons

L'inventaire des paramètres de Querystring est :

Nom du paramètre	Tournoyons vers Arbitre	Arbitre vers Tournoyons
<i>Gain1 & Gain2</i>	Représente le gain des joueurs avant le coup en cours	Idem, après le coup en cours
<i>Game</i>	Identifiant de la partie en cours	A renvoyer tel que reçu
<i>Method</i>	Spécifique jeu à deux plateaux	A renvoyer tel que reçu
<i>Move1</i> et <i>Move2</i>	Dernier coup joué	A renvoyer tel que reçu
<i>MoveId</i>	Identifiant du coup demandé	A renvoyer tel que reçu
<i>ReturnCode</i>	Non utilisé	Code retour de l'arbitre
<i>ReturnURL</i>	Spécifie l'URL à émettre en retour	Ne pas envoyer
<i>Tray</i>	Etat du plateau avant le coup en cours	Idem, après le coup en cours

Les paramètres *Gain1 & Gain2*

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

La syntaxe et les valeurs dépendent du jeu ajouté.

L'arbitre doit donc faire passer ces valeurs de « avant » à « après » le coup en cours

Le paramètre *Game (Partie)*

Le paramètre ***Game*** (traduire *Partie*) a pour valeur un identifiant unique, crypté, fixé par **Tournoyons**, chaîne de caractère (taille 10 max).

A renvoyer à Tournoyons tel que reçu.

Le paramètre *Method* (Méthode)

Le paramètre *Method* (traduire *Methode*) n'a d'intérêt que pour les jeux à deux plateaux.

En phase de préparation du plateau, **Method** = **Prepare**

A renvoyer à Tournoyons tel que reçu.

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 (*Move1* si le joueur est premier joueur, *Move2* sinon).

La syntaxe et les valeurs dépendent du jeu ajouté.

A renvoyer à Tournoyons tel que reçu.

NB : Dans le cas d'un jeu de nature 'Simultanée', *Move1* & *Move2* sont tous deux renseignés

Le paramètre *MoveId* (Id du coup)

Le paramètre *MoveId* (traduire *Id du coup*) est fixé par *Tournoyons*.

A renvoyer à Tournoyons tel que reçu.

Le paramètre *ReturnCode* (Code de retour) - Arbitre vers Tournoyons uniquement

Le paramètre *ReturnCode* (traduire *Code de retour*) décrit le statut de la partie, déterminé par l'arbitre.

Les valeurs possibles sont :

Signification	Valeur
Partie en cours	3
Fin de la partie, victoire du 1 ^{er} joueur	1
Fin de la partie, victoire du 2 ^{ème} joueur	2
Fin de la partie, match nul	4
Fin de la partie, due à une erreur constatée sur le coup reçu (*)	5

En cas d'erreur constatée, le paramètre *Tray* doit contenir le message d'erreur correspondant (ex : « *coup joué sur une case déjà occupée* »)

Le paramètre *ReturnURL* (URL de retour) - Tournoyons vers Arbitre uniquement

Le paramètre *ReturnURL* (traduire *URL de retour*) décrit l'URL à laquelle envoyer son retour.

A ne pas envoyer à Tournoyons.

Le paramètre *Tray* (Plateau)

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

L'arbitre doit donc faire passer ces valeurs de « *avant* » à « *après* » le coup en cours

NB : En cas d'erreur constatée, le paramètre *Tray* doit contenir le message d'erreur correspondant (ex : « *coup joué sur une case déjà occupée* »)

NB2 : En cas de jeu double, suivant le contexte, un seul plateau est donné, ou bien les deux. Dans ce dernier cas, les chaînes correspondant à chaque plateau sont et/ou doivent être séparées par '***'.

Attention : les caractères '/' et '|' ne doivent jamais être présents au sein de *Tray*

VII. 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).

VIII. 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.

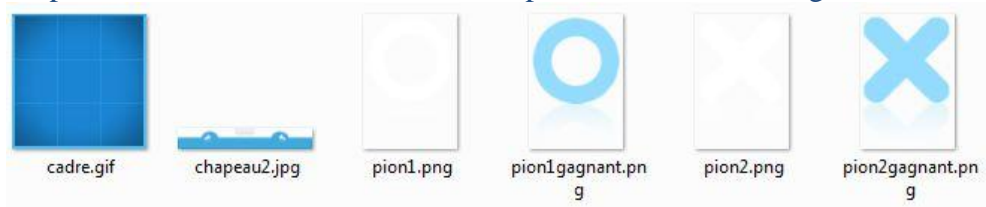
Annexe A – Exemple de code source d'implémentation du moteur de ChiFouMi

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()
?>
```

Annexe B – Exemple de code HTML de l'interface graphique de TicTacToe

Le code ci-après est issu du site [tournoyons](http://www.tournoyons.com) : interface graphique par défaut pour TicTacToe
En parallèle de ce fichier, on trouve le répertoire de nom « images », contenant :



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">
<head>
<link href='http://fonts.googleapis.com/css?family=Jockey+One' rel='stylesheet' type='text/css'>
<meta name="robots" content="noindex" />
<style>
TD {
font-family: Arial;
font-size:18px;
}
</style>
<script language='javascript'>
var NomJoueur1=""
var NomJoueur2=""
var Pion1=""
var Pion2=""
var PionGagnant1=""
var PionGagnant2=""
var Chemin="";
var LePlateau=['0','0','0','0','0','0','0','0','0','0'];
var LePlateauChaineInit="000000000"; //Pour le testeur de moteur
var EnSaisie=false;
var Interfacage='none'
var IsPremier=true;
var Valeur='';
var URLRetour='';
var Webmaster_Joue=false;
var WebTest=false;

function Affiche() {
    ContenuPlateau="<TABLE cellpadding=0 cellspacing=0 border=0>"
    for (Ligne=0;Ligne<3;Ligne++) {
        ContenuPlateau+="<TR height='150px'>"
        for (Colonne=0;Colonne<3;Colonne++) {
            Ind=Colonne+(3*Ligne)
            ContenuPlateau+="<TD width='154px' align='center' valign='bottom'>"
            switch (LePlateau[Ind]){
                case '0' :
                    if (EnSaisie) {
                        ContenuPlateau+="<A
Href='javascript:Input ("+(Ind+1)+") '>"
                        ContenuPlateau+="<IMG src='"+Chemin+"vide.gif' bor-
der=0></a>";
                    } else {
                        ContenuPlateau+="<IMG src='"+Chemin+"vide.gif' bor-
der=0>";
                    }
                    break;
                case '1' :
                    ContenuPlateau+=Pion1;
                    break;
                case '2' :
                    ContenuPlateau+=Pion2;
```



```

        break;
    case '3' :
        ContenuPlateau+=PionGagnant1;
        break;
    case '4' :
        ContenuPlateau+=PionGagnant2;
        break;
    }
    ContenuPlateau+="</TD>"
}
ContenuPlateau+="</TR>"
}
ContenuPlateau+="</TABLE>"
document.getElementById("Plateau").innerHTML=ContenuPlateau;
if (EnSaisie) {if (IsPremier) {
    LeNomJoueur1='<u>'+NomJoueur1+'</u>';LeNomJoueur2=NomJoueur2
    else {LeNomJoueur2='<u>'+NomJoueur2+'</u>';LeNomJoueur1=NomJoueur1}}
else {LeNomJoueur1=NomJoueur1;LeNomJoueur2=NomJoueur2}
document.getElementById("Joueur1").innerHTML=LeNomJoueur1
document.getElementById("Joueur2").innerHTML=LeNomJoueur2
}
function Affiche_Vide () {
    document.getElementById("Plateau").innerHTML='';
    document.getElementById("Joueur1").innerHTML=''
    document.getElementById("Joueur2").innerHTML=''
}

function Init() {
    var QueryNoms=new Array();
    var QueryValeurs=new Array();
    var QueryNombre=0;
    var Move1=0;
    var Move2=0;
    Private_Init()
    Query = window.location.search.substring(1);
    if (Query.indexOf('=') > 0) {
        Params = Query.split("&");
        QueryNombre=Params.length
        for (i = 0; i < QueryNombre; i++) {
            Paire = Params[i].split("=");
            QueryNoms[i]=Paire[0].toLowerCase()
            QueryValeurs[i]=Paire[1];
            //alert(i+' ' +QueryNoms[i]+'='+QueryValeurs[i])
        }
        Interfacage='standard'
        for (i = 0; i < QueryNombre; i++) {
            Valeur=QueryValeurs[i]
            switch(QueryNoms[i]) {
                case 'color': document.body.style.backgroundColor='#'+Valeur;break;
                case 'count': document.getElementById("Tirage").innerHTML=unescape(Valeur);break;
                case 'input' :
                    EnSaisie=true;IsPremier=(Valeur.toLowerCase()=='one');break;
                case
                    'legend':document.getElementById("Legende").innerHTML=unescape(Valeur);break;
                case 'method': Interfacage=Valeur.toLowerCase();break;
                case 'move1': if (Valeur!='') {Move1=Valeur;IsPremier=false;break;}
                case 'move2': if (Valeur!='') {Move2=Valeur;IsPremier=true;break;}

                case 'name1' : NomJoueur1=unescape(Valeur);break;
                case 'name2' : NomJoueur2=unescape(Valeur);break;

                case 'tray': if (Valeur != '') {for (j=0;j<9;j++)
                    {LePlateau[j]=Valeur.substring(j,j+1)}};break;
                case 'webtest' :
                    EnSaisie=false;WebTest=true;
                    Mess="<INPUT Type='Button' value='Joue'
                    onclick=Webmaster('joue')>";
                    Mess+="<INPUT Type='Button' value='Valeur plateau'
                    onclick=Webmaster('affiche')>";
                    document.getElementById("Legende").innerHTML=Mess
                    break;
                case 'urlreturn' : URLRetour=Valeur;break;
            }
        }
        if (Move1 > 0 ) {LePlateau[Move1-1]='1';}
        if (Move2 > 0 ) {LePlateau[Move2-1]='2';}
        switch(Interfacage) {

```

```

                case 'end' : Affiche();break;
                case 'standard' : Affiche();break;
                case 'none' : Affiche_Vide();break;
            }
        }
    }

function Input(Valeur) {
    if (IsPremier) {
        LeNomJoueur1='<u>'+NomJoueur1+'</u>';LeNomJoueur2=NomJoueur2
        else {LeNomJoueur2='<u>'+NomJoueur2+'</u>';LeNomJoueur1=NomJoueur1}
        document.getElementById("Joueur1").innerHTML=NomJoueur1
        document.getElementById("Joueur2").innerHTML=NomJoueur2
        document.getElementById("Legende").innerHTML="&nbsp;";
    }
    if (IsPremier) {LePlateau[Valeur-1]='1'} else {LePlateau[Valeur-1]='2'};
    EnSaisie=false;
    Affiche()
    //alert(URLRetour+'?ReturnValue='+Valeur)
    document.getElementById("Retour").src=URLRetour+'?ReturnValue='+Valeur
}

function Private_Init() {
    Chemin=location.pathname
    Pos=Chemin.lastIndexOf('/',Chemin.lenght)
    Chemin=Chemin.substring(0,Pos)
    Chemin='http://'+location.hostname+Chemin+'/images/'
    Pion1="<IMG src='"+Chemin+"pion1.png'">"
    Pion2="<IMG src='"+Chemin+"pion2.png'">"
    PionGagnant1="<IMG src='"+Chemin+"pion1gagnant.png'">"
    PionGagnant2="<IMG src='"+Chemin+"pion2gagnant.png'">"
}

function Webmaster(Ordre) {
    switch (Ordre) {
        case 'affiche' :
            LePlateauChaine=''
            for (i=0;i<9;i++) {LePlateauChaine+=LePlateau[i]}
            EnSaisie=false;
            prompt('LePlateau avec Valeur = '+Valeur,LePlateauChaine)
            break;
        case 'tray' :
            LePlateauChaine=''
            for (i=0;i<9;i++) {LePlateauChaine+=LePlateau[i]}
            return(LePlateauChaine)
            break;
        case 'joue' :
            Webmaster_Joue=true;
            EnSaisie=true;
            Affiche()
            break;
    }
}
</script>
<style>
TD {
font-size:1px;
}
.Div1 {
background: url('images/chapeau2.jpg') no-repeat #ffffff;
background-position: bottom;
z-index:20;
position:relative;
padding-bottom:4px;
margin-bottom:-6px;
}
.Div2 {
background: url('images/cadre.gif') no-repeat #ffffff;
background-position: top;
z-index:12;
position:relative;
}
#Legende {
font-family: 'Jockey One', arial, serif;
font-size:20px;
color:#1a85d3;
}
.Nom {
font-family: 'Jockey One', arial, serif;

```

```

font-size:18px;
font-weight:bold;
color:#ffffff;
}
.Score {
font-family: arial, serif;
font-size:36px;
font-weight:bold;
color:#000;
}
#Tirage {
font-family: arial, serif;
font-size:12px;
font-weight:bold;
color:#000;
}
</style>
<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE"
</head>
<body onload='Init()' bgcolor='#ffffff' >
<CENTER>
<IFRAME id='Retour' width='0' height='0' frameborder='0' src='about:blank'></IFRAME>
<DIV Class='Div1'>
<TABLE>
<TR>
<TD align='left' valign='top'><SPAN class='Score'
id='ScoreJoueur1'>&nbsp;</SPAN></TD>
<TD width='100px' align='center' valign='middle' cellspacing='0'><SPAN
id='Tirage'>&nbsp;</SPAN></TD>
<TD align='left' valign='top'><SPAN class='Score'
id='ScoreJoueur2'>&nbsp;</SPAN></TD>
</TR>
<TR><TD height='6px' >&nbsp;</TD></TR>
</TABLE>
<TABLE>
<TR><TD align='center' width='240px'><SPAN class='Nom' id='Joueur1'>&nbsp;</SPAN>&nbsp;</TD>
<TD width='40px'>&nbsp;</TD>
<TD align='center' width='230px'><SPAN class='Nom'
id='Joueur2'>&nbsp;</SPAN>&nbsp;</TD></TR>
</TABLE>
</DIV>
<DIV class='Div2' >
<TABLE>
<TR><TD colspan='3' width='470' height='470' align='center' valign='middle'><SPAN
id='Plateau'></SPAN></TD></TR>
<TR><TD colspan='3' height='40' align='center'><SPAN id='Legende'>&nbsp;</SPAN></TD></TR>
</TABLE>
</div>
<div id='PionBlanc' style='visibility:hidden;position:absolute;top:0px;left:0px;;z-index:30'><IMG
SRC='images/pion1.gif'></div>
<div id='PionNoir' style='visibility:hidden;position:absolute;top:0px;left:0px;;z-index:30'><IMG
SRC='images/pion2.gif'></div>
<HEAD>
<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">
</HEAD>
</body>
</html>

```