BDW1 - Projet PHP/MySQL
Un jeu de bataille navale

UCBL - Département Informatique de Lyon 1 – 2017

Table des matières

1 Informations organisationnelles

1.1 Description générale du projet

Le projet est à réaliser de préférence en binôme (les monômes sont acceptés mais les trinômes ne sont pas autorisés). Ce semestre, le projet porte sur la création d'un site Web permettant de jouer à la bataille navale avec la possibilité d'exploiter des bonus ou subir des malus en fonction de cartes tirées avant l'envoi d'un missile. L'application permet à plusieurs joueurs de se connecter et d'effectuer une ou plusieurs parties. L'application permettra de mémoriser les résultats des parties précédentes et de fournir des statistiques.

1.2 Rendu et notation du projet

1.2.1 Rendu du projet

Ce projet est à rendre avec les consignes de rendu suivantes :

— Le fichier rendu (un seul par binôme) aura une extension .zip et il doit inclure votre/vos nom(s) à la fois dans le nom de fichier et en entête de chacun de vos fichiers
— Ce fichier zip contiendra 3 fichiers :
  — le répertoire contenant les pages de votre site (code commenté et indenté). Le projet sera évalué avec le navigateur Mozilla Firefox, donc vérifiez le rendu de votre site avec ce navigateur !
  — un fichier .sql qui contiendra le script SQL "exécutable" de création de votre base de données ainsi que quelques tuples permettant de la peupler
  — un rapport au format pdf de 6 pages maximum (annexes inclues), qui détaillera vos choix de conception pour la BD, les problèmes rencontrés aussi bien au niveau organisationnel (travail en équipe) qu'au niveau technique, votre MCD (schéma E/A) et votre MLD (schéma Relationnel), et toute information que vous jugerez pertinente de préciser.

  Ce fichier zip est à rendre le jour de la séance de TP12 avant minuit dans la zone de dépôt TOMUSS dédiée (Rendu_projet).

1.2.2 Notation

La note de projet se calcule ainsi :

$$\text{NoteProjet} = \frac{1}{3}\text{NoteSoutenance} + \frac{1}{3}\text{NoteCode} + \frac{1}{3}\text{NoteRapport}$$

Il est important de noter que :
— Concernant la soutenance : au sein du binôme, la note de soutenance peut être différente pour les 2 étudiant(e)s
— Concernant le code : toute ressemblance trop importante entre des projets sera sanctionnée par un 0 pour la note de code et ce pour chacun des protagonistes.
— Concernant le rapport : il doit être au format PDF, de 6 pages maximum, organisé, clair, au format A4 et en police de taille 12, et l'orthographe sera prise en compte. Le contenu doit nous être instructif, par exemple pour nous permettre de comprendre des choix de conception/programmation, d'évaluer des problèmes organisationnels, ou d'obtenir des critiques constructives sur le sujet de projet. En résumé, il vaut mieux un rapport de 3 pages avec des informations pertinentes plutôt qu'un rapport de 6 pages truffé d'évidences ou de copier-coller...

1.2.3 Utilisation d'un *framework*

Il est possible d'utiliser un *framework* PHP ou CSS de votre choix, sous les conditions suivantes :
— Vous êtes autonome(s) : les chargé(e)s de TP ne sauront pas forcément vous aider en cas de problème avec le *framework*
— L'aide que procure le *framework* sera compensée par une plus grande exigence au niveau des résultats, en terme de fonctionnalités, qualité du code, etc.

1.2.4 Documentation MySQL et PHP

Rien de tel que la consultation des docs officielles pour répondre à vos questions !
— Documentation de MySQL (et les commandes SQL implémentées dans ce SGBD) :
  — `http://dev.mysql.com/doc/refman/5.0/fr/index.html`
— Documentation de HTML et CSS :
  — `https://www.w3schools.com/`
— Documentation de PHP et index des fonctions de PHP :
  — `http://www.php.net/manual/fr/`
  — `http://www.php.net/manual/fr/funcref.php`

2 Spécification de l'application

Cette section constitue le cahier des charges que vous aurez à satisfaire durant les séances de TP dédiées au projet et le travail personnel que vous aurez à faire en dehors des séances de TP.

2.1 Règles et principes

La bataille navale classique, appelée aussi "touché-coulé", est un jeu dans lequel deux joueurs possèdent chacun deux flottes composées de 5 navires. Ces navires sont :

— un porte-avion qui occupe 5 cases,
— un croiseur qui occupe 4 cases,
— un contre-torpilleur qui occupe 3 cases,
— un sous-marin qui occupe 3 cases,
— un torpilleur qui occupe 2 cases.

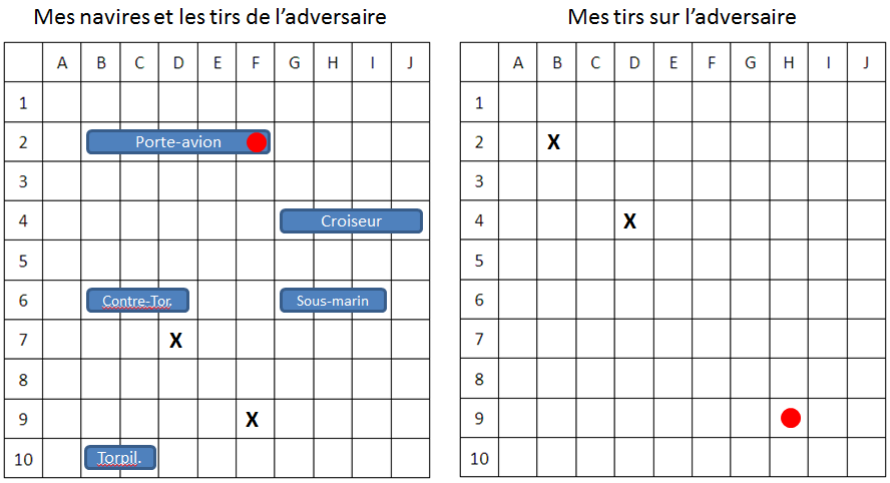

FIGURE 1 – Matrices de jeu de la bataille navale.

Les joueurs disposent de deux matrices de jeu composées chacune de 10 lignes et 10 colonnes, comme l'illustre la Figure 1. Les lignes sont numérotées de 1 à 10 et les colonnes sont indexées de A à J. La première matrice va permettre de placer les 5 navires du joueur et la seconde matrice va permettre de noter les tirs effectués pour découvrir les navires de l'adversaire. Chaque joueur doit tenter de "toucher" les navires adverses en proposant chacun leur tour des coordonnées (par exemple A2, D6...). Quand un navire a été entièrement touché, il est coulé. Le gagnant est le joueur qui parvient le premier à couler tous les navires de l'adversaire.

La version de la bataille navale que vous allez implémenter va permettre de coupler le jeu classique avec un jeu de carte bonus/malus.

FIGURE 2 – Proposition de cartes pour le jeu : huit cartes 'bonus' (bord vert) et deux cartes 'malus' (bord rouge)

Avant chaque tir, le joueur doit tirer une carte.

Parmi les cartes 'bonus', nous proposons huit cartes :

— la carte 'Missile' : cette carte permet de tirer un missile "classique" qui touchera la case indiquée. Les chances de tomber sur cette carte sont de 1 sur 4.

— la carte 'Rejoue une fois' : permet à celui qui la tire de rejouer une seconde fois. Les chances de tomber sur cette carte sont de 3 sur 20.
— la carte 'Vide ou pas vide ?' : permet à celui qui la tire de savoir avant de lancer son missile qu'une case est vide. Dans ce cas, il a la possibilité de changer de coordonnées de tir. Les chances de tomber sur cette carte sont de 1 sur 10.
— la carte 'Même pas mal !' : permet à celui qui la tire d'annuler un dégât subi sur un de ses bateaux et de le changer de place. Les chances de tomber sur cette carte sont de 3 sur 100.
— la carte 'Bateau leurre' : permet à celui qui la tire de placer un bateau leurre dans la grille. Si durant la partie le leurre est touché, il disparaît en laissant croire à l'adversaire qu'il a touché un bateau. Les chances de tomber sur cette carte sont de 3 sur 100.
— la carte 'Sauvez Willy' : permet à celui qui la tire de placer un orque dans la grille. Si durant la partie l'orque est touché, les trois plus petits bateaux non encore coulés sont alors coulés en représailles par des activistes écologistes. Les chances de tomber sur cette carte sont de 2 sur 100.
— la carte 'Méga-bombe' : permet à celui qui la tire de remplacer son prochain missile par une méga-bombe qui fera des dégâts également sur les 8 cases adjacentes à la cible (soit 9 cases d'un coup). Les chances de tomber sur cette carte sont de 3 sur 100.
— la carte 'Étoile de la mort' : permet à celui qui la tire de remplacer son prochain missile par un tir depuis l'étoile de la mort qui fera des dégâts également sur les 24 cases les plus proches de la cible (soit 25 cases d'un coup). Les chances de tomber sur cette carte sont de 1 sur 100.

Parmi les cartes 'malus', nous proposons deux cartes :

— la carte 'Passe ton tour' : oblige à celui qui la tire d'attendre que son adversaire joue deux fois. Les chances de tomber sur cette carte sont de 1 sur 10.
— la carte 'Mauvaise manip' : cette carte fait que le missile que vous vous apprêtiez à tirer a été échappé par le matelot qui devait charger le missile. C'est donc un de vos bateaux (choisi aléatoirement qui a été touché). Les chances de tomber sur cette carte sont de 3 sur 100.

Bien évidemment, vous êtes libre de proposer des cartes bonus supplémentaire en laissant libre cours à ~~vos délires~~ votre imagination : missile avec erreur de trajectoire, missile à multi-impact non adjacent...

2.2 Scénario

Un joueur souhaite faire une partie de bataille navale. Ce joueur se connecte à l'application, ce qui permet l'affichage de la page d'accueil en mode déconnecté. Si le joueur n'est pas inscrit, il a la possibilité de s'inscrire. S'il est déjà inscrit, il a la possibilité de se connecter en spécifiant un identifiant et un mot de passe. En se connectant, l'état associé au joueur passe de 'Déconnecté' à 'En ligne'. La page d'accueil de l'application en mode connecté, permet de visualiser :

— les statistiques sur les différentes parties réalisées via l'application,
— un accès à la page 'Listings' qui permet de répertorier les parties en cours, la liste des joueurs en train de jouer et la liste des joueurs connectés qui n'ont pas encore lancé de partie,
— un accès à la page 'Parie' qui permet de reprendre une partie en cours (stoppée temporairement),
— un bouton pour se déconnecter de l'application

L'accès à la page 'Listings' permet à l'utilisateur de lancer une partie. Pour lancer une partie, ce joueur doit inviter un autre joueur à jouer parmi les joueurs en lignes. Suite à cette invitation, si le joueur invité accepte l'invitation la partie est lancée. Sinon, le joueur a la possibilité d'inviter un autre joueur.

Une fois la partie lancée, chaque joueur à la possibilité de placer ses navires. Dès que le placement des navires est validé par un joueur, un message notifie à l'autre joueur que la partie commencera dès qu'il aura validé le placement de ses propres navires.

Dès que les navires des deux joueurs sont placés, l'application choisit de manière aléatoire le joueur qui commencera la partie. Un message notifie aux deux joueurs le nom du joueur qui commence. Quand un joueur joue, un message d'attente est notifié à l'autre joueur.

L'accès à la page 'Partie', permet de jouer. Un joueur commence par tirer une carte. Une fois la carte connue, le joueur peut agir (déplacement du bateau pour une carte 'Mêle pas mal', placement du leurre ou de l'orque pour respectivement les cartes 'Bateau leurre' et 'Sauvez Willy'), le joueur ensuite peut proposer les coordonnées de tir (sauf s'il a tiré une carte 'Malus') ou reproposer un second choix de coordonnées (s'il a tiré la carte 'Vide ou pas vide ?'. Les matrices des deux joueurs sont alors mises à jour. Et, c'est au tour du joueur suivant (sauf si le joueur avait tiré une carte 'Rejoue une fois').

A tout moment la partie peut être interrompue par les joueurs soit par une sortie dite "propre" en cliquant sur un bouton qui notifie à l'autre joueur que la partie est interrompue. Une sortie "non propre" est à prévoir

(fermeture du navigateur). Dans les deux cas les joueurs doivent pouvoir reprendre la partie là où la partie s'était interrompue [1].

2.3 Modélisation de la base de données

Remarque importante : Remarque importante, les indications ci-dessous vous sont proposées pour vous aider et n'ont aucun caractère obligatoire.

Pour pouvoir modéliser votre base de données, nous vous proposons la spécification suivante :

La version de la bataille navale qu'il vous est demandé de développer permet à des joueurs de s'affronter durant des parties. Une partie dispose d'une état : "En cours" (dès que la partie est créée et que les joueurs jouent) , "Interrompue" (si un des joueurs à déclaré vouloir se déconnecter ou que le navigateur a été fermé) ou "Clôturée" (dès que la partie est finie avec un gagnant).

Un joueur est identifié par son pseudonyme. Pour chaque joueur, on stocke son nom, prénom, sexe, date de naissance, sa ville de résidence et la valeur de hachage de son mot de passe [2].

Un joueur peut initier une partie, il peut inviter un autre joueur à une partie, placer des navires pour une partie et jouer des tours. En effet, dans une partie, un joueur dispose de navires identifiés par un code alphanumérique. Pour chaque navire, nous stockons son type (porte-avion, torpilleur...), sa taille (en nombre de case), un lien hypertexte vers la page Wikipedia référençant le type du navire (par exemple : https://fr.wikipedia.org/wiki/Porte-avions pour les porte-avions). Nous stockons également les coordonnées de placement du navire et son état ("opérationnel", "touché" ou "coulé"). Un navire est localisé par les coordonnées d'une cellule correspondant à la proue et à un sens ("horizontal" ou "vertical"). Pour un navire placé de manière horizontale la proue correspond à la cellule la plus à gauche. Pour un navire placé de manière verticale la proue correspond à la cellule la plus haute.

Une partie se décompose en tour. Pour chaque tour effectué par un joueur, ce joueur tire une carte correspondant à un bonus ou un malus potentiel. Une ou deux actions peuvent être associées à ce tour en fonction de la carte tirée (action de déplacement, action de réparation, de changement de coordonnées...).Pour chaque tour, nous stockons les coordonnées du tir effectué par le joueur et l'information si le tir à permis ou pas de toucher un navire. En fonction des cartes bonus, il est possible d'avoir plusieurs impacts pour un tir. Un tour est identifié de manière relative par rapport à une partie via un numéro qui s'incrémente à chaque tour. Du fait de l'existence de carte 'Passe ton tour', un même joueur peut jouer plusieurs fois de suite.

L'application permet de fournir des informations et des statistiques non seulement en fin de partie, mais aussi sur l'ensemble des parties.

Pour un joueur donné, l'application permet d'afficher le nombre de parties initiées par le joueur, le nombre de parties auxquelles le joueur a participé, le nombre de fois où le joueur a gagné, le nombre de fois où il a perdu, le ratio de victoires sur le nombre total de parties jouées, le nombre de parties durant lesquelles le joueur a bénéficié d'une carte bonus rare (i.e., les cartes bonus avec une probabilité inférieure ou égale à 6/100), le nombre de parties durant lesquelles le joueur a bénéficié de la carte bonus 'Étoile de la mort'.

Pour chaque carte bonus, le nombre moyen de fois où un type de carte est tirée dans une partie. Le nombre de parties jouées sans qu'aucune carte rare (i.e., les cartes bonus avec une probabilité inférieure ou égale à 6/100) ne soit tirée. Le nombre de parties jouées durant lesquelles plus de 10 cartes rares ont été tirées.

Pour l'ensemble des parties jouées, le nombre moyen de tirs effectués par un joueur avant la victoire...

3 Jalons et aides

3.1 Planning des jalons

Pour réaliser ce projet, un travail additionnel devra être réalisé hors des séances de TP. Afin de vous organiser, vous trouverez un échéancier des différents jalons que l'on est en droit d'attendre de vous :

- TP6 : Modélisation de la base de données. Définition de l'architecture du site.
  - Jalon 1 (début de séance TP7) : Schéma EA effectué et arborescence des fichiers créées.
- TP7 : Création de la base de données et définition des fonctions PHP permettant l'inscription et la connexion d'un joueur. Création de la page à propos et de la page d'accueil en mode déconnecté.

1. Il sera donc nécessaire de persister tous les états et toutes les actions en base
2. L'idée est de ne jamais stocker en clair le mot de passe d'un joueur mais seulement le résultat du hachage de son mot de passe. cf `http://php.net/manual/fr/function.hash.php`

  - Jalon 2 (début de séance TP8) : Un joueur doit pouvoir s'inscrire et se connecté.
- TP8 : Gestion du lancement d'une partie et du placement des navires et création des cartes bonus.
  - Jalon 3 (début de séance TP9) : Un joueur pré-inscrit doit pouvoir se connecter, afficher les matrices du jeu, placer les navires et afficher une carte bonus retournée.
- TP9 : **(Évaluation intermédiaire)** Gestion de la partie et de la synchronisation des affichages des écrans.
  - Jalon 4 : (début de séance TP10) : Un joueur pré-inscrit doit pouvoir se connecter, créer une partie, inviter un joueur à une partie, lancer la partie (bataille navale classique)
- TP10 : Gestion des modifications de règles en fonction des cartes bonus tirées.
  - Jalon 5 : (début de séance TP11) Le jeu est fonctionnel dans sa version complète
- TP11 : Gestion et affichage des statistiques et réalisation de tests pour s'assurer du bon fonctionnement de l'application. Correction de bugs. Définition du scénario pour la démonstration de la soutenance
  - Jalon 6 : (début de séance TP12) L'application est fonctionnelle. Vous êtes prêt(e)s pour la soutenance

3.2 Modélisation de la base de données

3.2.1 Produire le schéma entité/association

Vous êtes libres d'utiliser le logiciel de votre choix pour concevoir le schéma entité association, avec le formalisme Merise. Vous inclurez dans votre rapport une exportation de votre schéma entité/association de qualité suffisante pour être lisible.

3.2.2 Produire le modèle Relationnel

Selon l'outil de modélisation choisi à l'étape précédente, il est possible de générer le modèle Relationnel automatiquement, mais il est recommandé de bien vérifier le résultat obtenu et de le modifier si nécessaire. Sinon, la transformation du MCD vers le MLD se fait à partir des règles énoncées en cours. Vous inclurez une exportation de votre modèle Relationnel dans le rapport.

3.2.3 Produire le script SQL de création de la base

Selon l'outil de modélisation choisi à l'étape précédente, il est possible de générer le script de création des tables automatiquement (mais il est recommandé de bien vérifier le résultat obtenu et de le modifier si nécessaire). Sinon, écrivez les requêtes de création de vos tables.

TODO: Schéma entité/association + modèle relationnel

3.3 Architecture du site.

L'objectif de cette section est de vous inviter à mettre en place l'architecture de votre site.

TODO: Dans le répertoire racine, créer le sous-répertoire 'jeu'. Dans le répertoire 'jeu', créer les sous-répertoires 'css', et 'images'. Ajouter dans le répertoire 'jeu' les fichiers 'accueil.php' (le nom index.php est évidemment possible), 'listings.php', 'partie.php' et 'statistiques.php'. Dans le sous-répertoire 'css', créer le fichier 'style.css'. Dans le sous-répertoire 'images', vous pourrez stocker les images correspondant à vos cartes (si vous utilisez des images), à votre logo d'application...
Cette architecture évoluera au fur et à mesure de votre avancée dans le projet.

3.4 Connexion à la base de données.

Votre application va nécessiter de vous connecter à la base de données que ce soit au niveau de l'accueil, de la page de jeu ou de la page des statistiques. Comme différentes pages vont nécessiter une connexion à la base, vous allez regrouper au sein d'un fichier le paramétrage et l'ouverture de la connexion dans un fichier.

TODO: Créer le fichier *init.php* dans lequel vous créerez la connexion à votre base de données.).
Votre fichier *init.php* devra donc contenir le code suivant :

```php
<?php
        $bd = 'L2IFxx_BD';   //remplacer xx
        $user = '<loginTOMUSS>' ;
        $passwd = '<passwordTOMUSS>' ;
        $machine = 'localhost';

    $connexion = mysqli_connect($machine,$user,$passwd,$bd);

    if(mysqli_connect_errno()){ // erreur si > 0
       printf("Echec de la connexion : \%s", mysqli_connect_error());
        }
?>
```

Pour fermer proprement la connexion, il sera nécessaire de faire appel à la primitive *mysqli_close()*.

TODO: Créer le fichier *end.php*, dans lequel vous ajouterez la commande de fermeture de la connexion.

Toute page nécessitant un accès à la base de données inclura le fichier *init.php* en début de page et le fichier *end.php* en fin de page.

TODO: Comme test de connexion, vous pourrez implémenter les fonctions 'mystères' vues en TD7.

TODO: Définir une fonction PHP qui permet d'afficher une carte sous forme de 'div' avec la valeur de la carte et son nombre de points comme nous l'avons vu dans le TD8.

3.5 Gestion d'une session.

La conséquence direct de ces rafraîchissements périodique est qu'il sera nécessaire de passer par des variable de session pour pouvoir conserver les informations d'une page à la page rafraîchie.

Pour utiliser les sessions, il faut exécuter l'instruction suivante avant d'afficher la moindre chose dans la page PHP, même pas un espace ou un retour à la ligne :

```php
<?php session_start(); ?>
```

En fait, l'exécution de cette instruction créer le tableau associatif $_SESSION dont le contenu est sauvé à chaque fin d'exécution d'une page PHP avec session et restauré au moment de l'exécution de session_start().

Rappel : pour supprimer une session, il faut faire :

```php
<?php session_destroy(); ?>
```

TODO: Modifier la page d'accueil, pour permettre le stockage du pseudo du participant inscrit dans la variable de session après l'avoir mis dans la base de données. On en profitera pour afficher sur la page un 'Bonjour <pseudo>' où <pseudo> correspond au pseudo du participant.

Pour résumer : Le principe est donc de créer un formulaire contenant un onglet et un bouton permettant de soumettre un pseudo, le formulaire a pour action de rester sur la page d'accueil. Dans le cas où le formulaire est soumis, une requête SQL interroge la base pour savoir si ce pseudo est déjà utilisé pour la partie courante. Si le pseudo est déjà utilisé, alors on affiche un message d'erreur indiquant le problème et on laisse la possibilité à l'utilisateur de soumettre un nouveau pseudo. Si le pseudo est valide, alors le pseudo est inséré dans la table gérant les participants de la partie, le pseudo est mis en variable de session et la page affiche un 'Bonjour' au participant.

3.6 Gestion de la synchronisation

Comme une partie nécessite des interactions entre les deux joueurs, l'action d'un joueur devra enclencher le rafraîchissement le l'interface de l'autre joueur.

3.6.1 Rafraîchissement 'basique' des pages des participants

Pour pouvoir tenir compte des modifications (affichage des tirs effectués et des cartes sélectionnées par les participants, les interfaces de chacun des participants vont devoir se rafraîchir pour se synchroniser. Or il n'est

pas possible que l'initiateur du jeu déclenche le rafraîchissement des navigateurs des participants [3].

Une solution possible est de lancer périodiquement et automatiquement le rafraîchissement de la page des participants. Ceci peut ce faire par l'ajout dans les méta-données de la page de la balise :

<**meta** http−equiv="refresh" **content**="NS">

où *NS* correspond au nombre de secondes entre deux rafraîchissements.

Cependant, si vous affectez une valeur trop petite pour *NS*, votre page se rechargera trop souvent et il risque d'y avoir un effet de clignotement désagréable, et vous n'aurez pas le temps de soumettre des coordonnées saisies dans un onglet que votre page se rafraîchira. D'un autre côté, si vous mettez une valeur trop grande pour *NS*, la partie risque d'être un peu laborieuse. Il est donc important de bien calibrer la valeur de *NS*.

3.6.2 Rafraîchissement 'avancée' des pages des participants

Pour éviter les rafraîchissements intempestifs, il serait nécessaire de déclencher le rafraîchissement des pages uniquement quand celui-ci est nécessaire. Pour cela, vous pouvez créer une table pour la synchronisation qui contiendra la liste des participants ainsi que l'état de synchronisation associé. Dans un premier temps, l'état peut être un booléen qui vaudra *1*, si la page du participant doit être rafraîchie et *0* sinon. Le principe, illustré sur la Figure 4, est donc que pour chaque participant, l'application interroge la table de synchronisation pour savoir s'il doit se rafraîchir. La consultation de la table se fera via l'utilisation de code javascript. Nous utiliserons la librairie JQuery `http://jquery.com/`

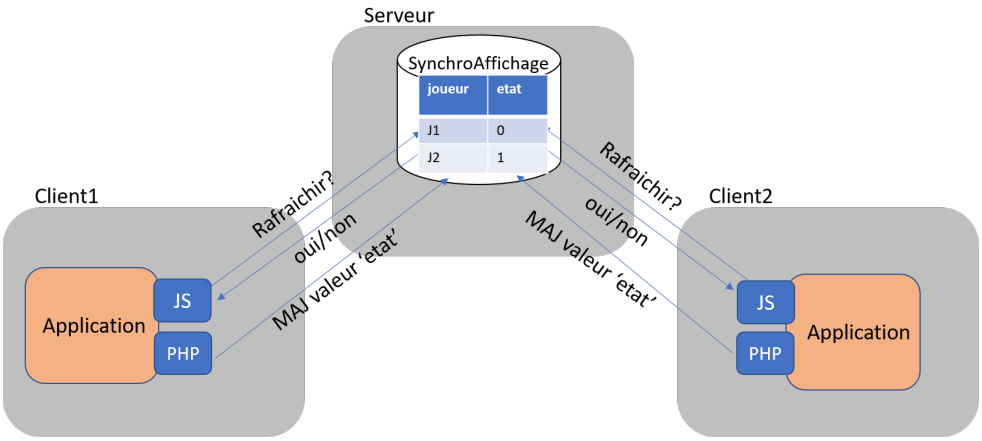

FIGURE 3 – Gestion de la synchronisation

Dans votre répertoire 'jeu', créer le sous-répertoire 'lib'. Récupérer la version courante de JQUERY (1.12.x) dans l'archive de ressources et placer le fichier 'jquery-1.12.1.js' dans le répertoire 'lib'.
Modifier le fichier d'entête pour ajouter dans son 'head' la metatdonnée suivante :

`<script src="./lib/jquery-1.12.1.js"></script>`.

Vous pouvez à présent utiliser la bibliothèque JQUERY. Pour ce faire, nous allons créer un script qui regroupera vos commandes JQUERY. Récupérer le fichier 'monCodeJQuery.js' fourni dans l'archive 'ressources.zip' sur la page de l'UE, puis modifier votre entête pour ajouter dans son head la metatdonnée suivante :

`<script src="./monCodeJQuery.js"></script>`.

Le fichier 'monCodeJQuery.js' contient du code qui va vous permettre d'interroger une table 'synchroAffichage' pour récupérer l'état associé à l'utilisateur courant. En ajoutant le code suivant dans le 'head' de votre fichier partie.php :

```
<script type="text/javascript">
$(document).ready(function(){
    setInterval(rafraichirOuPas, 1000, "<?php echo $_SESSION['joueur']; ?>");
});
</script>
```

3. Plus tard, en Master 1 vous verrez l'utilisation de WebSockets qui permettrait de le faire mais ceci dépasse le cadre des enseignements BDW1

l'interrogation de la base se fera toutes les secondes (i.e. 1000 millisecondes) sans que la page soit rechargée, puisque que ce code est évalué côté client. Le pseudo du participant courant est supposé être stocké dans la variable de session `$_SESSION['joueur']`. L'interrogation de la base de données se fera du côté serveur, c'est pour cela vous avez le fichier 'interrogerEtat.php' qui contient le code permettant de retourner l'état associé au pesudo du participant.

TODO: Créer la table *synchroAffichage(joueur,etat)*.
TODO: Modifier votre code pour que l'inscription à une partie enclenche l'ajout de l'utilisateur dans la table 'synchroAffichage' (la valeur de l'état sera par défaut à *0*).
TODO: Créer une fonction PHP qui permet de mettre à 1 l'état de tous les joueurs dans la table 'synchroAffichage'. L'appel de cette fonction permettra de notifier que toutes les interfaces doivent se synchroniser.
TODO: Tester le code.

3.7 Algorithme

3.7.1 Primitives d'affichages

TODO: Définir la fonction PHP qui permet d'afficher votre espace de jeu (2 matrices) correspondant à la grille contenant vos navires et les tirs de l'adversaire et à la grille contenant les impacts de vos tirs.

TODO: Définir la fonction PHP qui permet de tirer et d'afficher une carte.

3.7.2 Traitement d'un tour

Le traitement d'un tour nécessite d'effectuer certains traitements qui sont déclenchés par les participants. Ces traitements permettent de mettre à jour la base de données. L'objectif ici est d'expliquer en pseudo-code la manière dont vous gérez un tour.

TODO: Définir l'algorithme de traitement d'un tour → rapport

3.7.3 Traitement d'une partie

Une fois que vous avez défini clairement la façon dont se déroule une manche, vous pouvez l'inscrire dans le déroulement d'une partie. L'objectif ici est donc d'expliquer en pseudo-code la manière dont vous gérer une partie.

TODO: Définir l'algorithme de traitement d'une partie → rapport