Cours : Web Dynamique 2 (WD2)
Professeur : M. J.P. Delcroix

Compte rendu de TP n°3

Utilisation des bases de données en PHP et communication « réseau »

Bataille Navale

La Bataille Navale

Auteur : Pierre Galerneau
Année Universitaire : 2008 / 2009

Résumé

L’objectif de ce troisième TP faisant l’objet d’un compte rendu en Web Dynamique 2 était de réaliser un site permettant de jouer en réseau à la bataille navale.

J’ai adapté ce sujet afin de réaliser un site qui soit le plus agréable possible et surtout le plus ressemblant possible à un site de jeu réel. Pour cela, j’ai pris exemple sur quelques sites trouvés sur internet, sur lesquels j’ai cherché des idées, que j’ai tenté ensuite de reproduire dans la mien. J’ai notamment trouvé l’idée de réaliser un système de communication entre les joueurs (en leur donnant la possibilité de s’envoyer des invitations à jouer) en surfant sur internet.

Table des matières

Introduction

Dans ce troisième et avant dernier rapport de Web Dynamique 2; je vais présenter les différents aspects de la réalisation d'un site permettant de jouer à la bataille navale en réseau. Ce rapport va me permettre de mettre sur papier toute la réflexion mise en œuvre pour réaliser ce site, que ce soit pour les pages simples ou les plus compliquées.

Dans ce second rapport, l'un des grands objectifs était d'utiliser le PHP couplé aux bases de données MySQL. En effet, depuis maintenant quelques temps, nous sommes arrivés à l'étude des bases de données en PHP et il nous était donc demandé de les utiliser dans le cadre de ce TP. De plus, je me suis fixé l'objectif de réaliser un système de communication simple entre les différents utilisateurs du site, système reposant sur la base de données, afin de permettre aux joueurs de se proposer des parties les uns les autres, de les accepter, de les refuser,…

Comme ce projet ne s'est pas réalisé en une seule étape, il est intéressant de se poser la question suivante pour comprendre comment j'ai pu en arriver au résultat final :

Quelles étapes et quels choix ont étés nécessaires à la réalisation en bonne et due forme d'un site permettant de jouer en réseau à la bataille navale en PHP utilisant les bases de données MySQL comme sauvegarde de données

Je vais donc répondre à cette question dans les pages qui vont suivre.

1 Présentation du projet

Dans ce troisième projet, l'objectif est de programmer un jeu de Bataille Navale en utilisant la technologie PHP / MySQL. Cependant, il faut préciser que ce jeu n'est pas un jeu « simple », c'est-à-dire, un joueur jouant contre un ordinateur. C'est un jeu en « réseau » : un utilisateur joue contre un autre utilisateur connecté en même temps que lui.

Pour réaliser ce site, il m'a donc fallu réaliser différentes choses :

- La communication entre différents joueurs (envoie d'invitations à jouer,…),
- Le placement des bateaux joueur par joueur et partie par partie,
- La réalisation du jeu en lui-même,
- …

De plus, comme ce site doit permettre à plusieurs joueurs de « se voir », de s'inviter,.. il m'a fallu mettre en place un module permettant d'afficher la liste des joueurs présents sur le site au moment X où un joueur est connecté afin de lui donner la possibilité de rentrer en contact avec les autres joueurs pour pouvoir jouer.

Maintenant que les bases du projet sont posées et que l'on comprend qui pourra faire quoi sur notre site, il est nécessaire d'expliquer comment sont gérées les données et comment elles ont organisées entre elles au sein de nos tables.

2 La base de données

Dans cette section du site, je vais présenter l'organisation des données sous forme de base de donnée MySQL que j'ai mise en place pour arriver à réaliser un site permettant de jouer à la bataille navale en réseau.

Il m'a fallu enregistrer des données sur de très nombreuses entités différentes. Je vais présenter tout cela ici.

2.1 Esquisse

Pour réaliser ce projet, il nous a fallu stocker un certain nombre d'informations concernant :

- les utilisateurs,
- les parties,
- les bateaux,
- les tirs,
- les utilisateurs connectés,
- les tours (à qui est ce de jouer)
- …

La base de données à mettre en place a donc été assez compliquée puisqu'elle ne se compose pas simplement de deux ou trois tables comme c'était le cas dans les TP précédents celui-ci.

Dans le sujet, rien n'est (une fois encore) précisé quand à l'architecture des données, c'est donc à nous de la mettre en place et de l'imaginer en fonction de nos besoins.

De fait, il est apparu après analyse que nous avions besoin de sept tables pour mener à bien notre projet. Je vais les détailler une par une ci-dessous.

Remarque : Comme dans les projets précédents, les tables de celui-ci seront préfixées d'un nom faisant référence à la bataille navale afin de les différencier au sein de la base MySQL chez free. Le préfixe sera « bataille » dans ce TP.

2.2 La table bataille_users

La table vpc_Users est destinée à contenir toutes les données concernant les utilisateurs de notre site. En effet, ces données doivent pouvoir être enregistrées quelque part afin d'être retrouvées et consultées régulièrement, notamment au moment de la connexion au site.

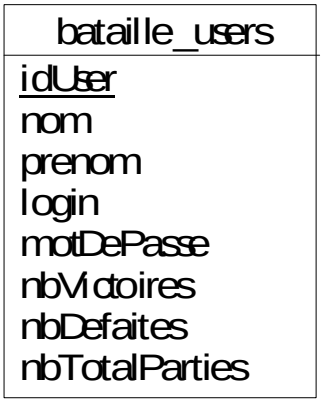
Comme l'objectif de ce TP est de réaliser un jeu de bataille navale en ligne, il faut pouvoir stocker certaines informations pour chaque utilisateur. Bien sur, il faut enregistrer les informations indispensables à la connexion d'un utilisateur comme :

- Son login,
- Son mot de passe,
- Son nom,
- …

Mais il faut également enregistrer des données plus propres au jeu telles que :

- son nombre de victoires,
- son nombre de défaites,
- son nombre de parties jouées,
- …

Pour cela, j'ai mis en place une table dont le schéma est le suivant :



Je vais maintenant décrire le type de chaque propriété de cette table :

- idUser : entier séquentiel non nul,
- nom ; chaine de caractères non nulle,
- prenom : chaine de caractères non nulle,
- login : chaine de caractères non nulle,
- motDePasse : chaine de caractères / numeriques non nulle et de taille supérieure ou égale à 6 caractères,
- nbVictoires : entier,
- nbDefaites ; entier,

- nbTotalParties : entier.

Le script correspondant à cette table (script qui permet de la créer dans la base de données associée) est le suivant :

Code

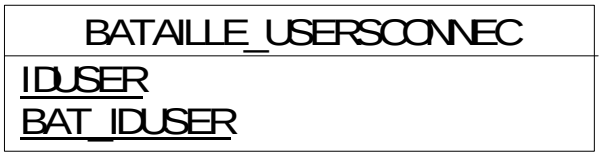
```
-- ============================================================
--   Table : BATAILLE_USERS
-- ============================================================
create table BATAILLE_USERS
(
   IDUSER              DECIMAL(6)          not null,
   NOM                 VARCHAR(50)         not null,
   PRENOM              VARCHAR(50)         not null,
   LOGIN               VARCHAR(50)         not null,
   MOTDEPASSE          VARCHAR(50)         not null,
   NBVICTOIRES         INTEGER             not null,
   NBDEFAITES          INTEGER             not null,
   NBTOTALPARTIES      INTEGER             not null
);

-- ============================================================
--   Index : BATAILLE_USERS_PK
-- ============================================================
create unique index BATAILLE_USERS_PK on BATAILLE_USERS (IDUSER asc);
```

Ce script n'est pas un script de création MySQL mais un script SQL classique. Je pense que ca ne change pas grand-chose de présenter celui-ci plutôt qu'un autre puisque le principe est d'expliquer quel type de script est utilisé pour créer une table en SQL et non de faire un cours de MySQL…

2.3 La table bataille_usersConnectes

Cette table ne contient qu'un seul champ : l'identifiant de l'utilisateur connecté. En effet, pour identifier un utilisateur connecte, il suffit d'enregistrer son identifiant dans une table à part et on sait si oui ou non il est connecté au site.

En voici le schéma !



Je visn de dire que cette table ne contenait qu'un seul champ, cependant, comme c'est une table qui dépend uniquement de la table bataille_users mais qu'elle doit exister tout de même, l'identifiant utilisateur apparait deux fois à cause du fait que c'est une association récursive.

Voici son script de création :

Code

```
-- ============================================================
--   Table : BATAILLE_USERSCONNEC
-- ============================================================
create table BATAILLE_USERSCONNEC
(
   IDUSER              DECIMAL(6)          not null,
   BAT_IDUSER          DECIMAL(6)          not null
);

-- ============================================================
--   Index : BATAILLE_USERSCONNEC
-- ============================================================
create unique index BATAILLE_USERSCONNEC on BATAILLE_USERSCONNEC (IDUSER asc, BAT_IDUSER asc);
```

Cette table est donc destinée à contenir tous les identifiants de tous les utilisateurs connectés à un certain moment sur le site.

2.4 La table bataille_messages

Cette table est destinée à servir de base de stockage pour l'envoi des messages d'un utilisateur à un autre. En effet, lorsqu'un utilisateur souhaite envoyer une invitation à un autre, il doit passer par la base de données puisqu'il ne peut pas lui parler « en direct ». Une invitation est donc crée dans la base bataille_messages et sera affichée au moyen d'un script de vérification de la base sur le compte de l'utilisateur destination au moment où il se connectera.

Pour cela, il faut donc enregistrer certaines informations afin que le message et la réponse du destinataire arrivent bien au bon utilisateur.

En voici le schéma :

Cette table contient deux champs de type entier séquentiels qui sont en réalités les identifiants de deux utilisateurs différents : l'utilisateur qui a envoyé l'invitation et celui qui est le destinataire de celle-ci. Le troisième champ est quant à lui un booléen qui permet de savoir si l'invitation a été délivrée ou non afin d'éviter qu'elle ne soit réaffichée ultérieurement.

Le script SQL correspondant est le suivant :

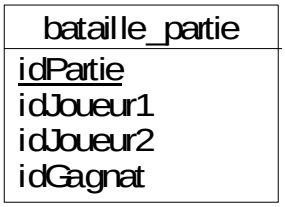
Code

```
-- ============================================================
--   Table : BATAILLE_MESSAGE
-- ============================================================
create table BATAILLE_MESSAGE
(
   IDUSERSOURCE        DECIMAL(6)          not null,
```

```
   IDUSERDESTINATION    DECIMAL(6)          not null,
   LU                   DECIMAL(1)          not null
);

-- ============================================================
--   Index : BATAILLE_MESSAGE_PK
-- ============================================================
create unique index BATAILLE_MESSAGE_PK on BATAILLE_MESSAGE (IDUSERSOURCE asc, IDUSERDESTINATION asc);
```

2.5 La table bataille_partie

Cette table est destinée à conserver les informations concernant une partie. De fait, une partie se joue entre deux joueurs et est terminée ou non. Il faut donc conserver ces informations afin que l'on puisse savoir à tout moment si un utilisateur a une partie en cours, s'il a terminé toutes ses parties, et si les parties terminées ont étés gagnées par l'un ou l'autre des joueurs.



Cette table contient uniquement des identifiants de type entier non nul. En effet, la partie doit posséder un identifiant numérique afin que l'on puisse la repérer simplement et toujours savoir de laquelle on parle quand on la cherche. De plus, une partie doit contenir les identifiants de ses protagonistes de manière à ce que l'on puisse savoir très simplement qui joue avec qui dans une partie donnée. Enfin, il faut conserver un champ dans lequel on enregistrera l'identifiant du vainqueur lorsque la partie sera terminée afn de pouvoir signifier que untel, d'identifiant X a gagné la partie n°Y.

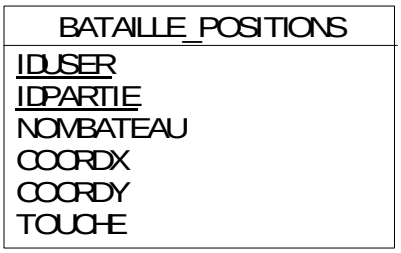
Voici son script de création :

Code

```
-- ============================================================
--   Table : BATAILLE_PARTIE
-- ============================================================
create table BATAILLE_PARTIE
(
   IDPARTIE          DECIMAL(6)          not null,
   IDUSER            DECIMAL(6)          not null,
   IDJOUEUR1         INTEGER             not null,
   IDJOUEUR2         INTEGER             not null,
   IDGAGNAT          INTEGER             not null
);

-- ============================================================
--   Index : BATAILLE_PARTIE_PK
-- ============================================================
create unique index BATAILLE_PARTIE_PK on BATAILLE_PARTIE (IDPARTIE asc);
```

2.6 La table bataille_positions

Lorsqu'un joueur démarre une partie contre un adversaire, il doit d'abord enregistrer les positions des bateaux qu'il veut placer afin de jouer. Pour cela, il dispose d'une page qui lui demande de choisir leur position, qui vérifie leur validité et qui les enregistre ensuite dans la base de données. De fait, il faut forcément les enregistrer afin que les coordonnées puissent être vérifiées lors des phases suivantes du jeu. On a donc créé une table bataille_positions contenant toutes les positions de chaque bateau de chaque joueur. Celle-ci se présente de la manière suivante :



Voici la liste des types des propriétés de la table :

- idUser : identifiant de l'utilisateur concerné (venant de la table bataille_users),
- idPartie : identifiant de la partie concernée (vient de la table bataille_partie),
- nomBateau : nom du bateau auquel appartient la coordonnée : chaine de caractères,
- coordX : entier correspondant à la position en X de cette case (abscisse),
- coordY : entier correspondant à la position en Y de cette case (ordonnée),
- touche : booléen permettant de dire si la case a été touchée par un tir de l'adversaire.

Cette table est en relation avec les deux tables bataille_users et bataille_partie car, les coordonnées d'un bateau ne sont vraies que pour un joueur donné dans une partie donnée. Il faut donc enregistrer ces données de manière à ce qu'elles soient « classées » par joueur et par partie.

Voici le script SQL correspondant :

Code

```
-- ============================================================
--   Table : BATAILLE_POSITIONS
-- ============================================================
create table BATAILLE_POSITIONS
(
   IDUSER            DECIMAL(6)          not null,
   IDPARTIE          DECIMAL(6)          not null,
   NOMBATEAU         VARCHAR(50)         not null,
   COORDX            INTEGER             not null,
   COORDY            INTEGER             not null,
   TOUCHE            INTEGER             not null
);

-- ============================================================
--   Index : BATAILLE_POSITIONS_P
-- ============================================================
create unique index BATAILLE_POSITIONS_P on BATAILLE_POSITIONS (IDUSER asc, IDPARTIE asc);
```

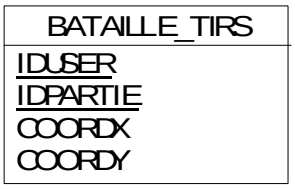
2.7 La table bataille_tirs

La table bataille_tirs permet de stocker dans la base les tirs effectués par un utilisateur donné lors d'une certaine partie. En effet, il faut pouvoir les stocker afin de « se souvenir » de ce qui a déjà été tiré. Ceci permettra notamment de pouvoir vérifier si les bateaux ont étés coulés ou bien d'afficher les grilles de jeu contenants des images permettant de signaler à l'utilisateur que son adversaire a tiré dans telle case. Ceci permettra donc d'avoir une grille qui affichera une sorte d'historique des différents tirs de l'adversaire.

Cette table doit donc conserver trois informations importantes :

- Qui a tiré (identifiant du joueur),
- Dans quelle partie est intégré ce tir,
- Ou a tiré le joueur.

Voici donc le schéma de cette table :



Les différentes propriétés qui la composent sont de type :

- idUser : entier séquentiel (identifiant de l'utilisateur, récupéré dans la table bataille_users,)
- idPartie : entier séquentiel (identifiant de la partie concernée, récupérée dans la table bataille_partie),
- coordX : coordonnée en abscisse du tir, entier non nul,
- coordY : coordonnée en ordonnée du tir, entier non nul.

Cette table est en relation avec les tables bataille_partie et bataille_users car elle contient des données spécifiques à un utilisateur au sein d'une partie précise. Il faut donc nécessairement que les données soient liées.

Voici le script de création SQL correspondant :

Code

```
-- ============================================================
--   Table : BATAILLE_TIRS
-- ============================================================
create table BATAILLE_TIRS
(
   IDUSER            DECIMAL(6)         not null,
   IDPARTIE          DECIMAL(6)         not null,
   COORDX            INTEGER                 ,
   COORDY            INTEGER
);

-- ============================================================
--   Index : BATAILLE_TIRS_PK
-- ============================================================
create unique index BATAILLE_TIRS_PK on BATAILLE_TIRS (IDUSER asc, IDPARTIE asc);
```

2.8 La table bataille_aqui

Cette table a pour objectif de permettre de savoir à qui c'est de jouer. C'est-à-dire que, lorsque l'un des deux joueurs d'une partie aura joué, son contenu sera modifié afin que l'on puisse savoir que c'est à l'autre joueur de jouer. Pour cela, on enregistre dans cette table différentes informations, notamment l'identifiant de la partie et un numéro correspondant au joueur dont c'est le tour de jouer.

Voici le schéma correspondant :

Ceci signifie que l'identifiant du joueur X correspondant à la partie Y a l'autorisation de jouer, c'est son tour.

Les données de cette table sont de type :

- idPartie : identifiant de la partie concernée (vient de la table bataille_partie),
- idJoueur : entier correspondant à l'un des deux joueurs de la partie (enregistrés dans la table bataille_partie).

Encore une fois, cette table est récursive par rapport à la table bataille_partie. En effet, elle n'a pas de raison d'être sans l'existence d'une partie, elle doit donc être en relation avec cette table. Cependant, aucune autre table ne rentre dans ce schéma puisque seules des informations de la table bataille_parties doivent être conservées afin de savoir à qui est ce de jouer.

Voici le script SQL correspondant à la création de cette table :

Code

```
-- ============================================================
--   Table : BATAILLE_AQUI
-- ============================================================
create table BATAILLE_AQUI
(
   IDPARTIE            DECIMAL(6)          not null,
   BAT_IDPARTIE        DECIMAL(6)          not null,
   IDJOUEUR            INTEGER             not null
);

-- ============================================================
--   Index : BATAILLE_AQUI_PK
-- ============================================================
create unique index BATAILLE_AQUI_PK on BATAILLE_AQUI (IDPARTIE asc, BAT_IDPARTIE asc);
```

2.9 La base de données en général

Dans ce projet, la base de données se compose donc de sept tables dont seulement deux sont indépendantes des autres. Les cinq autres sont en relation avec au moins une des deux tables principales que sont :

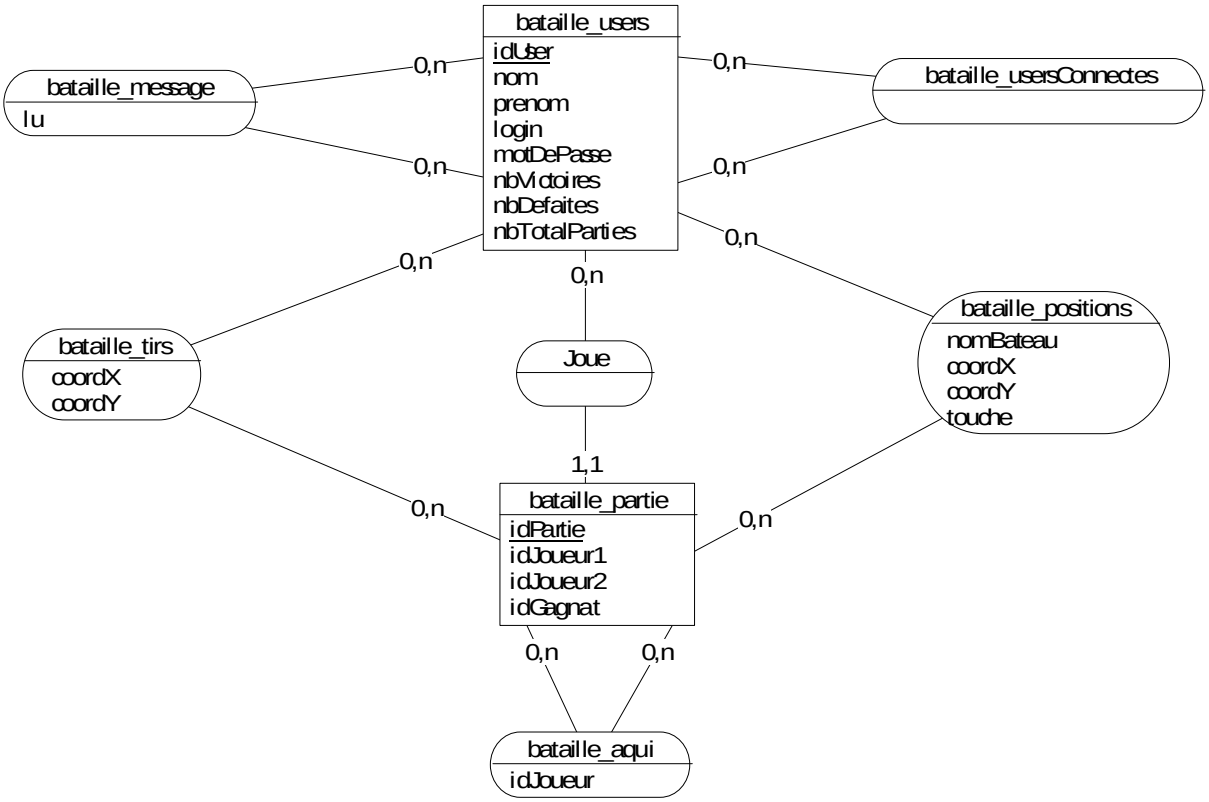
- La table bataille_users,
- La table bataille_partie.

Je vais présenter ici les modèles qui m'ont permis d'arriver à cette modélisation :

- Le Modèle Conceptuel des Données (MCD),
- Le Modèle Physique des Données (MPD).

Ces deux modèles permettent de définir les données conceptuellement puis en terme de tables SQL, ce qui permet une analyse du problème propre et claire.

2.9.1 MCD



Ce modèle est beaucoup plus compliqué que ceux que l'on a pu voir jusqu'à présent puisque la plupart des tables sont en relations entre elles et s'imbriquent plus ou moins.

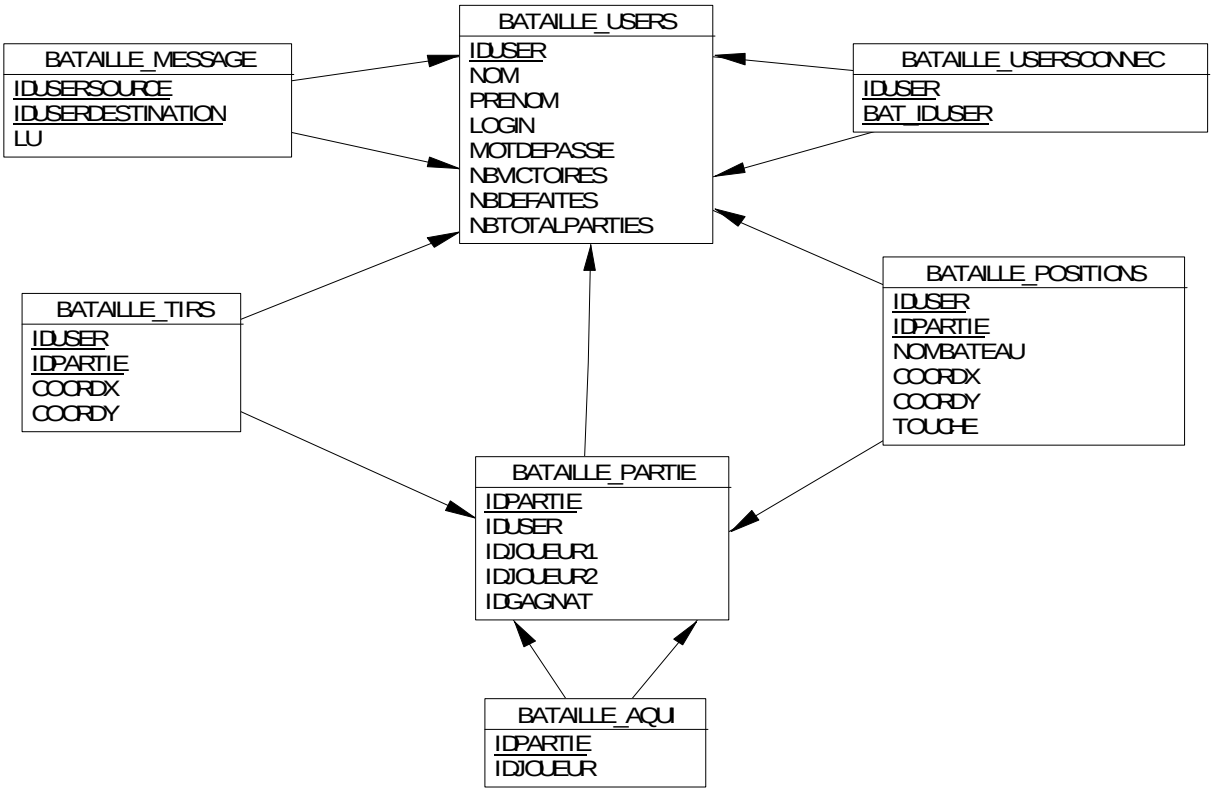
Toute fois, ce modèle retranscrit les besoins en termes de données du projet. On peut déjà supposer le nombre de tables à créer en regardant les cardinalités : lorsqu'une association a deux cardinalités se terminant par « n », une table sera nécessaire.

Venons en maintenant au MPD permettant de visualiser réellement les tables qui seront crées et qui permettront de stocker toutes les données permettant de réaliser le jeu de bataille navale en ligne :

2.9.2 MPD

Le MPD correspond, lui, au besoin en terme de tables, du projet. C'est grâce à lui que l'on peut visualiser les tables à mettre en place afin que les données soient ordonnées le plus proprement et clairement possible au sein de notre site.

Le voici :



Les flèches correspondent aux « flux » de données entre les tables. Cela correspond au fait que, lorsqu'une flèche part d'une table vers une autre, la première contient des informations relatives à la seconde, nécessaires à la compréhension et à la cohérence des données.

Maintenant que la structure de la base de données nécessaire au bon fonctionnement du site et celle des tables la composant a été expliquée, je vais décrire les choix que j'ai eu à faire lors de la programmation de ce projet.

3 Mes choix de programmation

Pour réaliser ce site, le choix du langage ne nous a pas été donné puisque nous le réalisons dans le cadre du cours de Web Dynamique 2, dans lequel nous étudions le PHP / MySQL. Cependant, nous avons eu de nombreux choix à faire afin de mener à bien le projet. Nous avons notamment du décider de :

- La manière de proposer des parties entre les utilisateurs,
- L'architecture et l'organisation des données,
- L'interface graphique à présenter,
- …

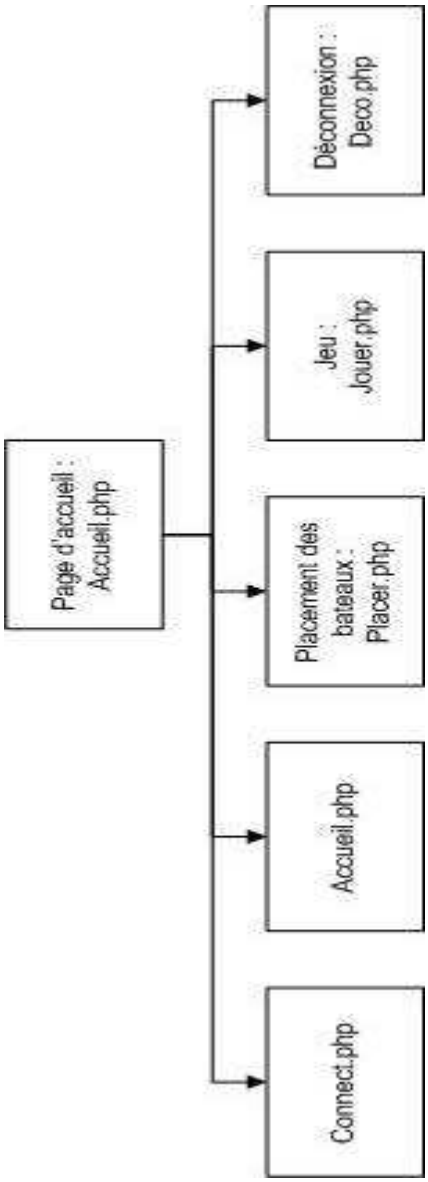
L’objectif principal était de réaliser un site pratique à utiliser et dans lequel les utilisateurs se sentent bien afin qu’ils n’hésitent pas à revenir jouer.

3.1 Choix d’architecture

Dans la réalisation de ce site, je me suis vite rendu compte qu’il n’y avait pas besoin de mettre en place un grand nombre de pages PHP pour qu’il fonctionne. Cependant, de nombreux algorithmes se répétaient, c'est-à-dire que nous en avions besoin à plusieurs endroits. Pour éviter de les recopier et de faire des copier / coller inutiles, j’ai donc décidé de réaliser ces algorithmes dans des pages à part. Ceci me permet en effet de les utiliser quand bon me semble en utilisant simplement l’instruction PHP include à l’endroit où j’en ai besoin. Ceci me permet donc de simuler des fonctions générales comme on pourrait le faire si l’on programmait de manière objet. Ceci permet donc de simplifier la lisibilité du code et de séparer les parties répétitives afin de ne pas avoir besoin de les répéter. C’est en effet toujours le même code qui s’exécute, ce qui évite les erreurs et les répétitions.

3.1.1 Architecture des pages

Toutes les pages de mon site s’enchainent à partir du moment où un utilisateur se connecte sur mon site. Je vais expliquer l’enchainement de ces pages au moyen d’un diagramme :



Même si l'arborescence des pages est présentée ci-dessus et qu'elle parait très simple par rapport à ce que nous avons pu voir jusqu'à maintenant, il est important que j'explique certains autres choix que j'ai fais concernant l'architecture du site notamment au niveau des portions de codes qui se répétaient régulièrement.

3.1.2 Fichiers de code externes

Comme je me suis rapidement rendu compte que certaines portions de codes se répétaient et que leur répétition rendait la lisibilité du code très mauvaise, j'ai cherché un moyen de simuler des fonctions de « classe », c'est-à-dire des fonctions accessibles de partout, publiques.

Après avoir cherché quelques temps, j'ai pensé à faire des fichiers de code PHP externes, ne contenant que le code correspondant à une fonction spéciale. Ceci me permet en effet d'avoir accès à la fonction concernée en incluant simplement le fichier la contenant à l'endroit où j'en ai besoin.

Ceci permet de simplifier la lecture du code des pages du site mais également celle du code des fonctions. De fait, ces fonctions sont assez importantes et peuvent parfois atteindre 100 à 150 lignes de code pur. Il est donc beaucoup plus clair et pratique de ne pas avoir à les réécrire et de pouvoir les appeler en une seule ligne.

Si on prend l'exemple de la fonction permettant de vérifier si un des deux joueurs a gagné la partie en cours (fonction enregistrée dans le fichier verif_gagne.php), son appel sera fait de la manière suivante :

Code

```
…
Include(« verif_gagne.php ») ;
…
```

Ceci permet donc un gain de lisibilité énorme et en même temps, cela permet de ne pas avoir à modifier toutes les occurrences de la fonction lorsque l'on désire la modifier. En effet, il suffit, dans ce cas, de modifier le fichier contenant et tous les appels de la fonction appelleront le code modifié ! C'est donc également un gain de précision et de simplicité.

J'ai utilisé ce principe de fichiers de code externes afin de simuler un principe objet. En effet, je n'aime pas du tout avoir à réécrire certaines fonctions déjà créées car je trouve que c'est une perte de temps et que cela risque d'entrainer des problèmes si jamais on décide de modifier ces fonctions. J'ai notamment utilisé ce principe lors de :

- La vérification qu'un utilisateur a gagné ou perdu,
- La vérification du tour de jeu (à qui est ce de jouer),
- La vérification des messages (l'utilisateur a-t-il reçu une invitation ?),
- …

Maintenant, je vais présenter en détail les différentes pages du site et surtout les différents affichages produits par celles-ci. En effet, même si le site ne compte pas un nombre de pages très élevé (moins de 10 si on ne compte pas les pages de code PHP pur), il est important de présenter les différentes choses qui peuvent s'afficher car il y a en fait un certain nombre de choses, notamment au niveau des messages et de leur réception.

3.1.3 Les différentes pages du site

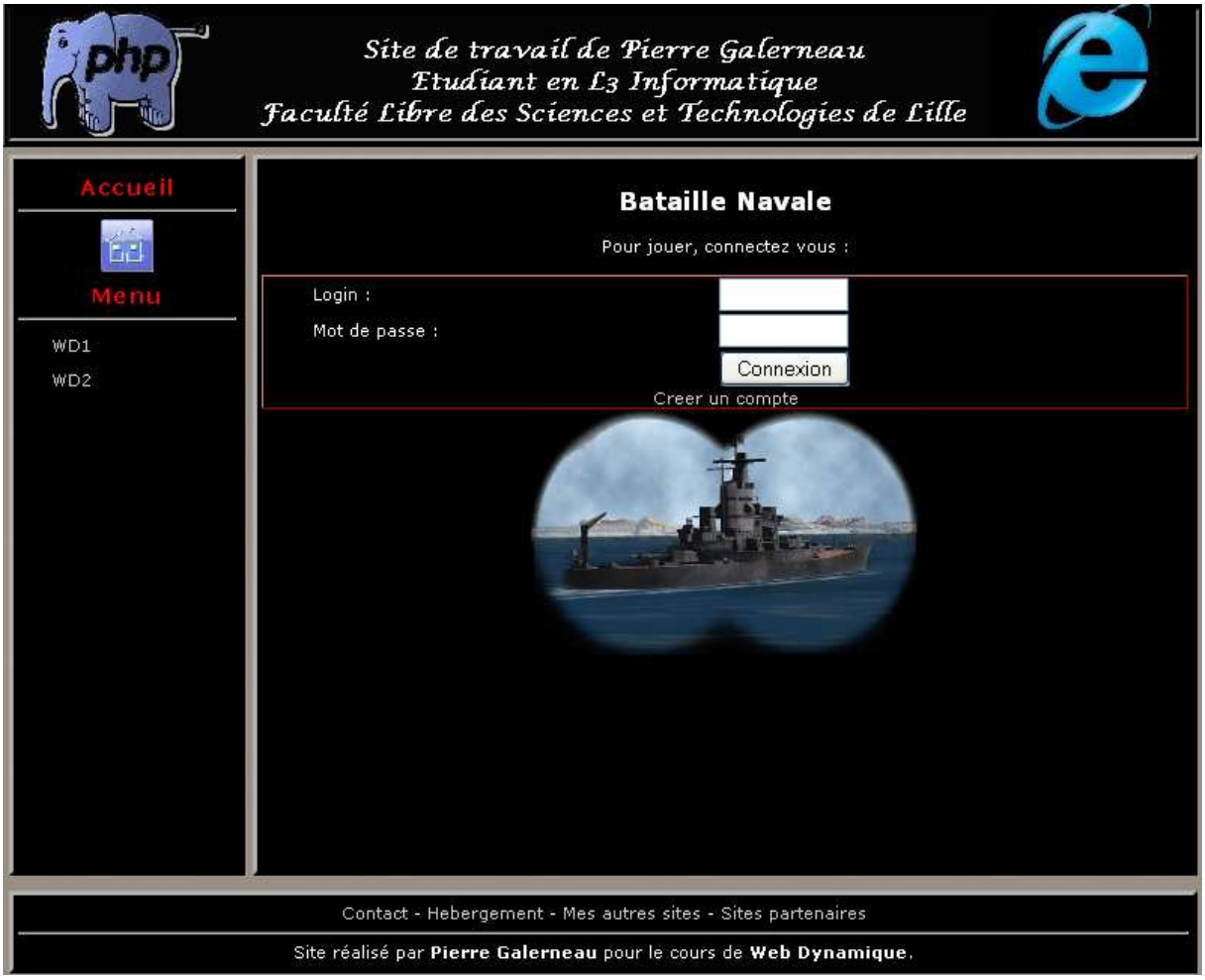
J'ai déjà présenté en gros les différentes pages accessibles dans la section précédente, cependant, je pense qu'il est bon que je fasse une section dans laquelle je présente plus en détail chaque page de manière à présenter les choix graphiques que j'ai fais et les possibilités qu'un utilisateur a sur chaque page du site. Cette section n'est pas destinée à présenter la manière dont les pages sont générées par le serveur, mais seulement les choses qu'elles proposent aux utilisateurs, les différentes fonctions et possibilités proposées.

La page accueil.php

Lorsqu'un utilisateur arrive sur le site et qu'il souhaite jouer à la bataille navale, il a deux possibilités. Soit il possède déjà un compte et il peut alors se connecter simplement, soit il n'e possède pas et il doit en créer un.

La page d'accueil affiche quant à elle un formulaire de connexion assemblé à lien permettant de créer un compte.

Voici l'allure de cette page lorsque l'utilisateur n'est pas connecté et viens juste d'arriver !

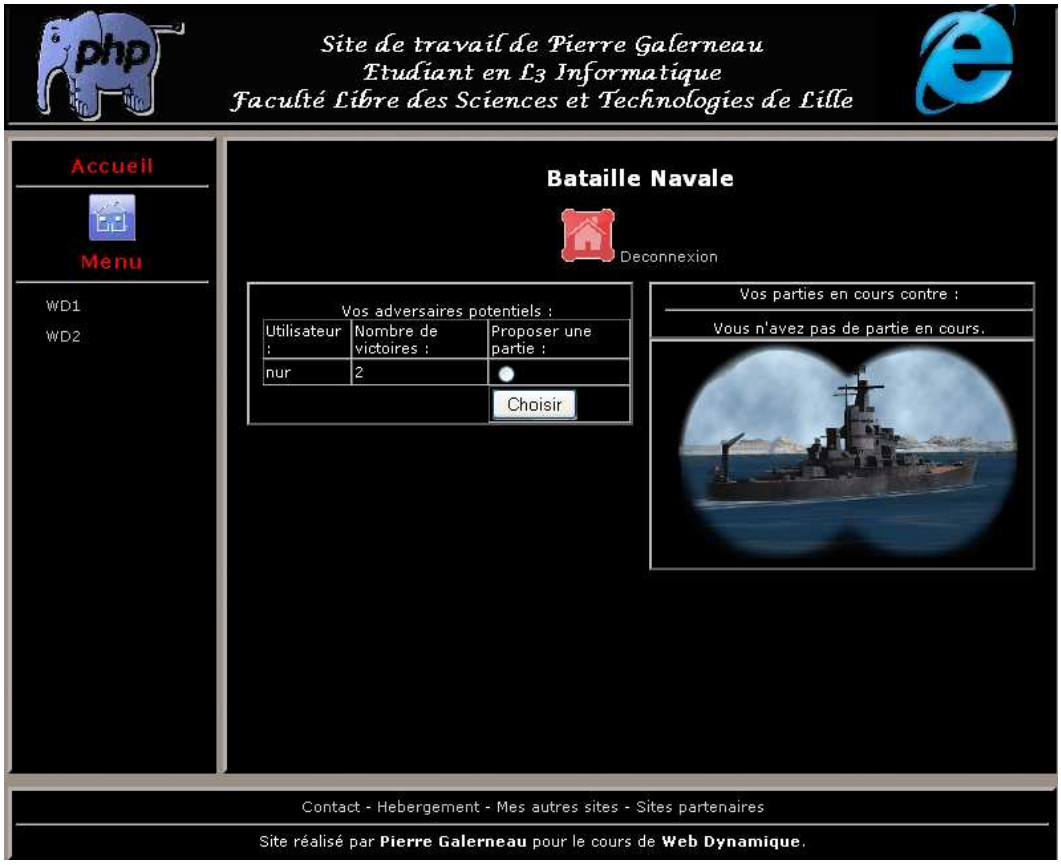


Cette page permet donc de se connecter au site mais également d'afficher d'autres choses. En effet, lorsqu'un utilisateur est connecté sur son compte, il peut voir s'afficher sur cette page (en s'y rendant à l'aide du menu) :

- La liste des utilisateurs connectés en même temps que lui,
- La liste des parties qu'il a en cours.

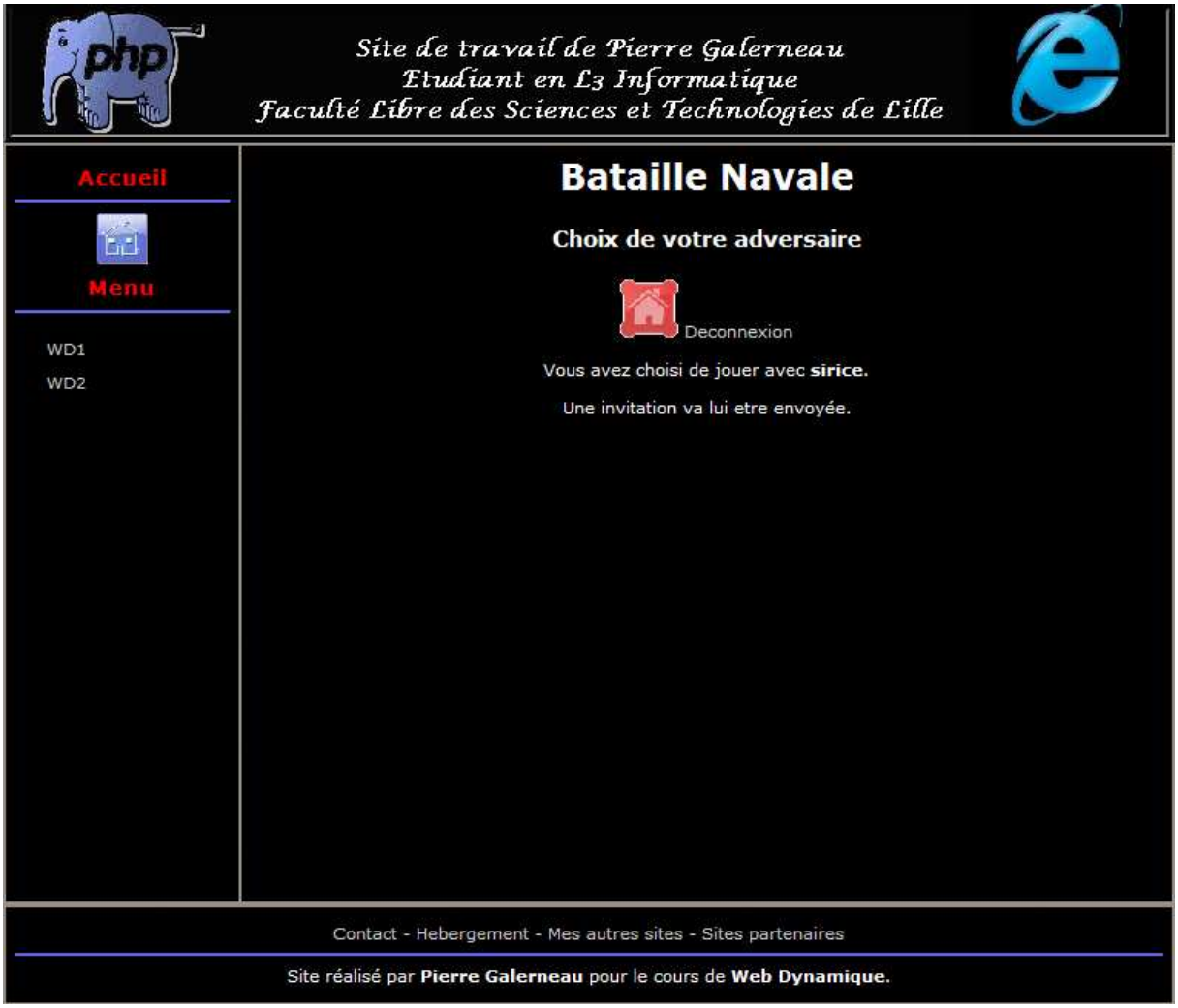
Voici l'affichage correspondant :

On voit ici l'affichage correspondant au fait qu'il n'y ai personne de connecté, cependant, si quelqu'un d'autre se connecte, on aura l'affichage suivant :



L’utilisateur peut alors choisir de jouer avec un utilisateur connecté en même temps que lui en lui envoyant une invitation au moyen du formulaire associé à l’affichage des autres joueurs présents. Lorsqu’une invitation sera envoyée, un message sera affiché signifiant l’envoi de l’invitation et l’attente de la réponse du destinataire.

Voici l’affichage signifiant qu’une invitation a été envoyée :



Sur la page du destinataire, une phrase apparaitra alors, signifiant qu’une invitation est arrivée et luis proposant – au moyen de liens – de l’accepter ou de la refuser.

Voici celui qui permet de voir que l’on a reçu une invitation :

Remarque : les messages signifiants qu'un utilisateur a reçu une invitation s'afficheront de la même manière que l'utilisateur se trouve sur la page d'accueil ou sur n'importe quelle autre page du site. Ceci est fait au moyen d'une page de code externe qui vérifié, à chaque actualisation de la page, si l'utilisateur a reçu une invitation ou non.

La page de création de compte

Lorsqu'un utilisateur est sur la page d'accueil du site mais qu'il ne possède pas de compte lui permettant de s'y connecter, il peut cliquer sur le lien « Créer un compte » qui l'enverra vers la page « créer_compte.php ».

Celle-ci est en fait un formulaire HTML permettant à l'utilisateur de saisir les informations nécessaires à la création de son compte au sein de la base de données. Voici l'affichage correspondant à cette page :

Lorsque le nouvel utilisateur aura créé son compte au moyen de cette page, il pourra alors retourner sur la page d'accueil afin de se connecter au site.

Cette connexion sera effectuée au moyen de la page connect.php qui est présentée ci-dessous.

La page de connexion

Lorsqu'un utilisateur souhaite se connecter, il lui faut saisir les informations demandées dans le formulaire de connexion présent sur la page d'accueil. Une fois ces données saisies, il valide le formulaire et est alors renvoyé vers la page connect.php qui les vérifie et l'autorise ou non à se connecter. Si il est autorisé à se connecter, alors l'utilisateur est renvoyé vers la page d'accueil sur laquelle les informations présentées ci-dessus concernant les autres utilisateurs connectés ou les parties en cours de l'utilisateur sont affichées. L'utilisateur nouvellement connecté peut alors voir et choisir ce qu'il veut faire, c'est-à-dire :

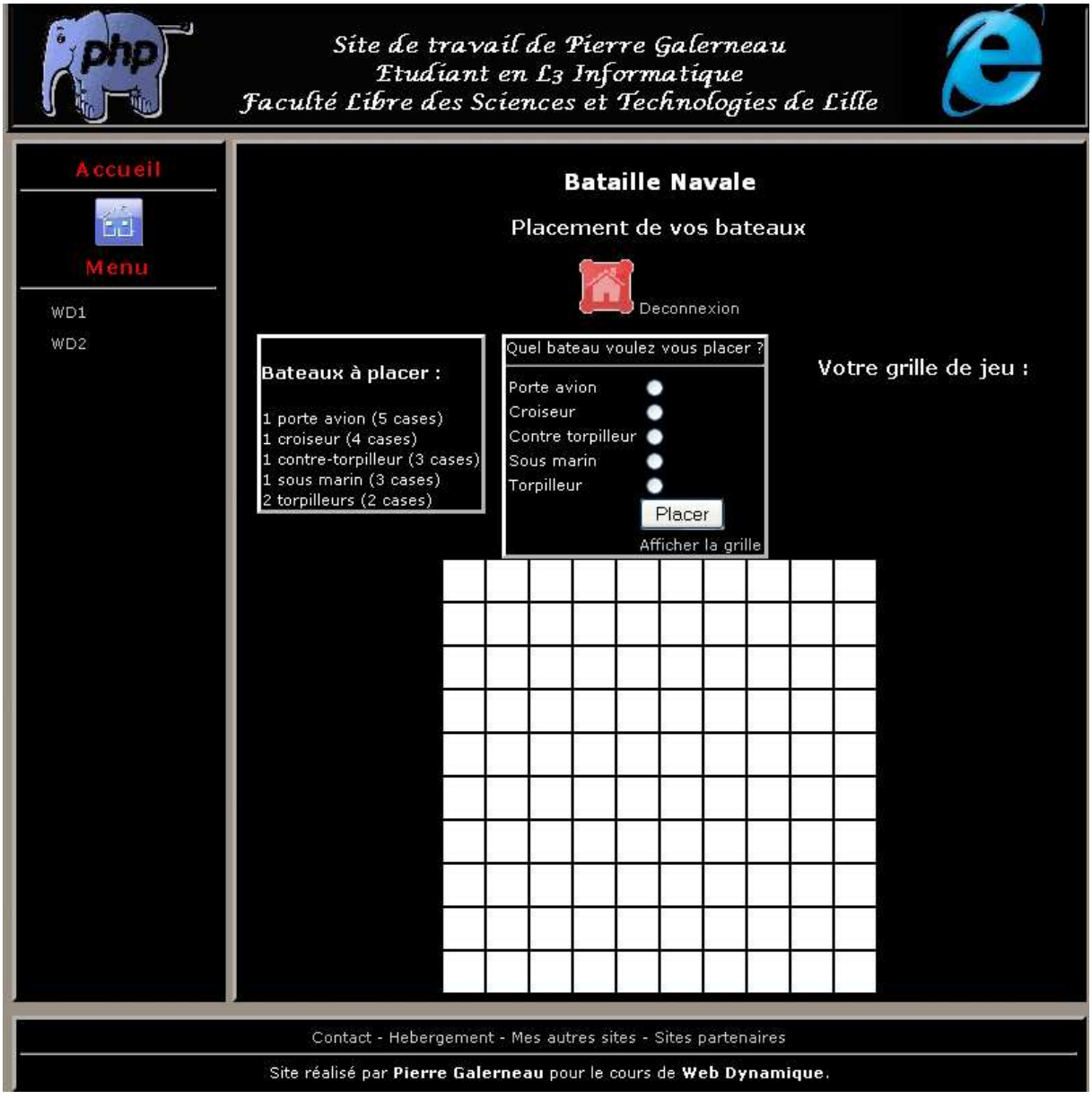
- Demander une nouvelle partie,
- Reprendre une partie non terminée.

Placement des bateaux

Lorsqu'un utilisateur crée une nouvelle partie et que son adversaire accepte de jouer avec lui, ils doivent tous les deux placer leurs bateaux. Pour cela, la page placer.php est appelée et leur donne la possibilité de choisir quel bateau ils veulent placer.

Pour plus de clarté, j'ai mis en place un tableau expliquant quelle taille occupe chaque bateau (nombre de cases) afin que tous les joueurs puissent savoir simplement ce qu'il faut cocher pour placer tel ou tel bateau. De plus, un formulaire permet de spécifier à l'utilisateur quels bateaux il n'a pas encore placé. Ceci permet d'éviter qu'un utilisateur ne tente de placer plusieurs fois le même bateau, etc.

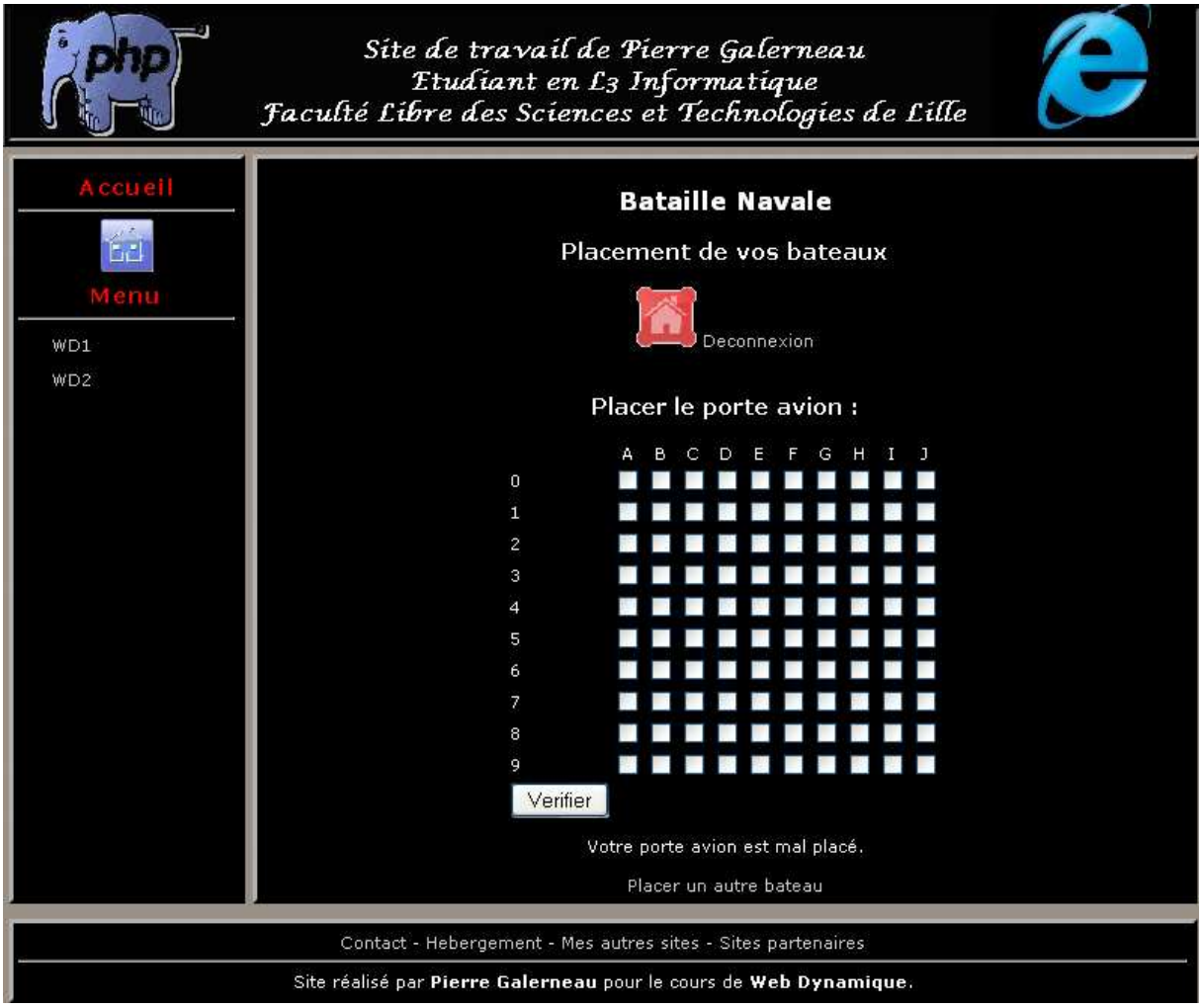
Voici l'affichage de la page placer.php à l'arrivée de l'utilisateur (quand aucun bateau n'est placé) :



L'utilisateur peut alors choisir de placer les bateaux qu'il souhaite, un par un. Lorsqu'il choisi un bateau, un formulaire composé de cases à cocher (de type checkbox), est affiché, ce qui lui permet de placer ses bateaux comme il le souhaite. Les vérifications nécessaires sont alors effectuées afin de garantir que les bateaux placés dans la base de données bataille_positions sont bel et bien valides. Un message apparait alors spécifiant à l'utilisateur si oui ou non le

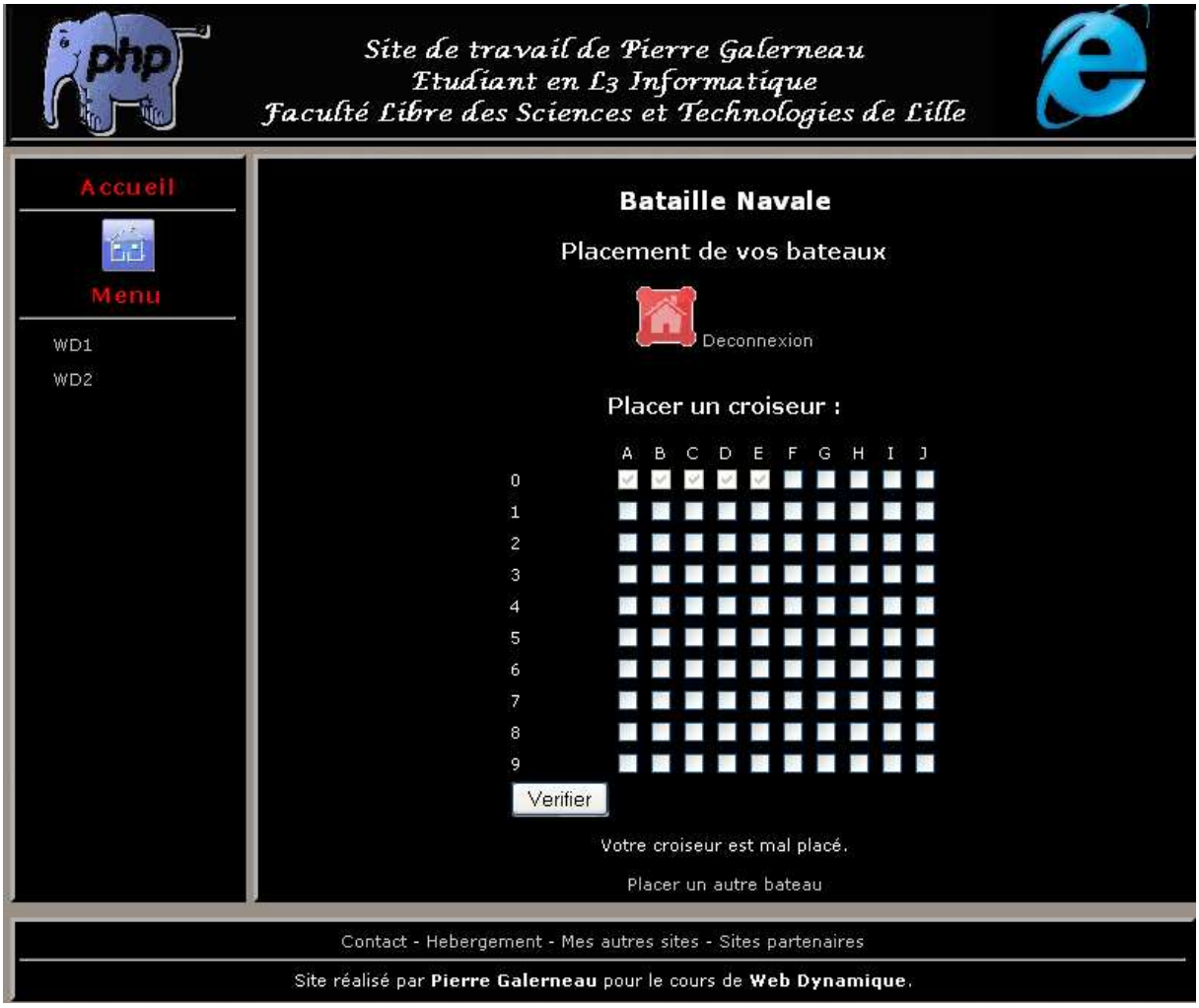
bateau est bien placé. Il peut alors choisir de placer un autre bateau en cliquant sur le lien « placer un autre bateau » qui redonne lui permet de retourner au choix du bateau à placer tout en voyant la grille qu'il a composé pour l'instant.

Voici l'affichage correspondant au moment où un utilisateur demande le placement d'un bateau alors qu'il n'en a placé aucun :



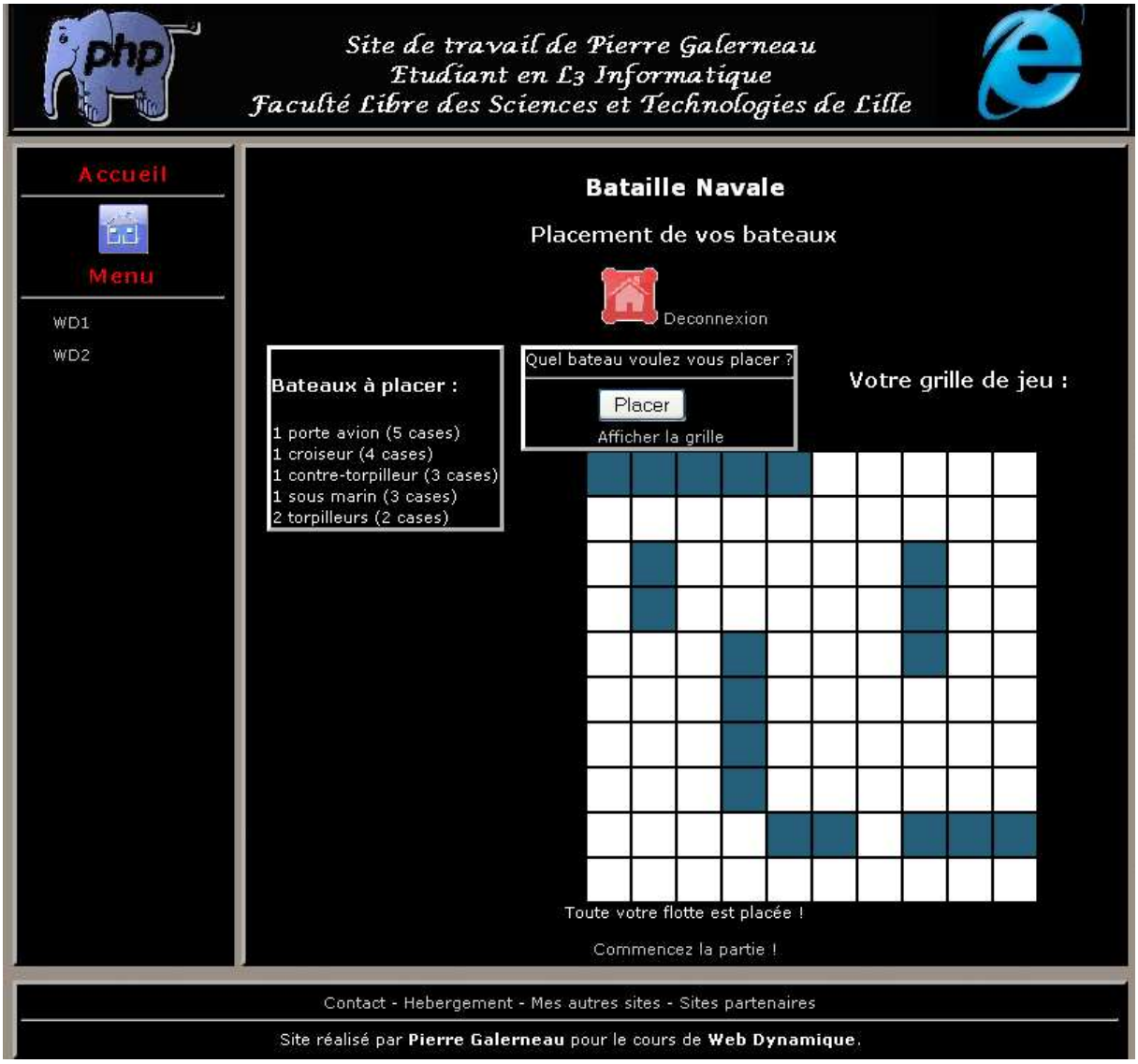
Lorsque des bateaux sont déjà placés et que l'utilisateur demande le placement d'un autre bateau, l'utilisateur voit apparaitre la même grille que présentée ci-dessus, cependant, les cases déjà occupées par un bateau placé précédemment sont cochées et mises en « disable », c'est-à-dire grisées afin d'éviter que l'utilisateur ne puisse les réutiliser. Cette technique permet en réalité de les désactiver afin de faciliter la vision de l'utilisateur des cases qu'il a déjà utilisées.

Voici un affichage correspondant :



Ceci permet, je trouve, de simplifier le travail de mise en place des bateaux de l'utilisateur.

Enfin, voici l'affichage de la page lorsque tous les bateaux sont placés et que l'utilisateur peut commencer la partie :



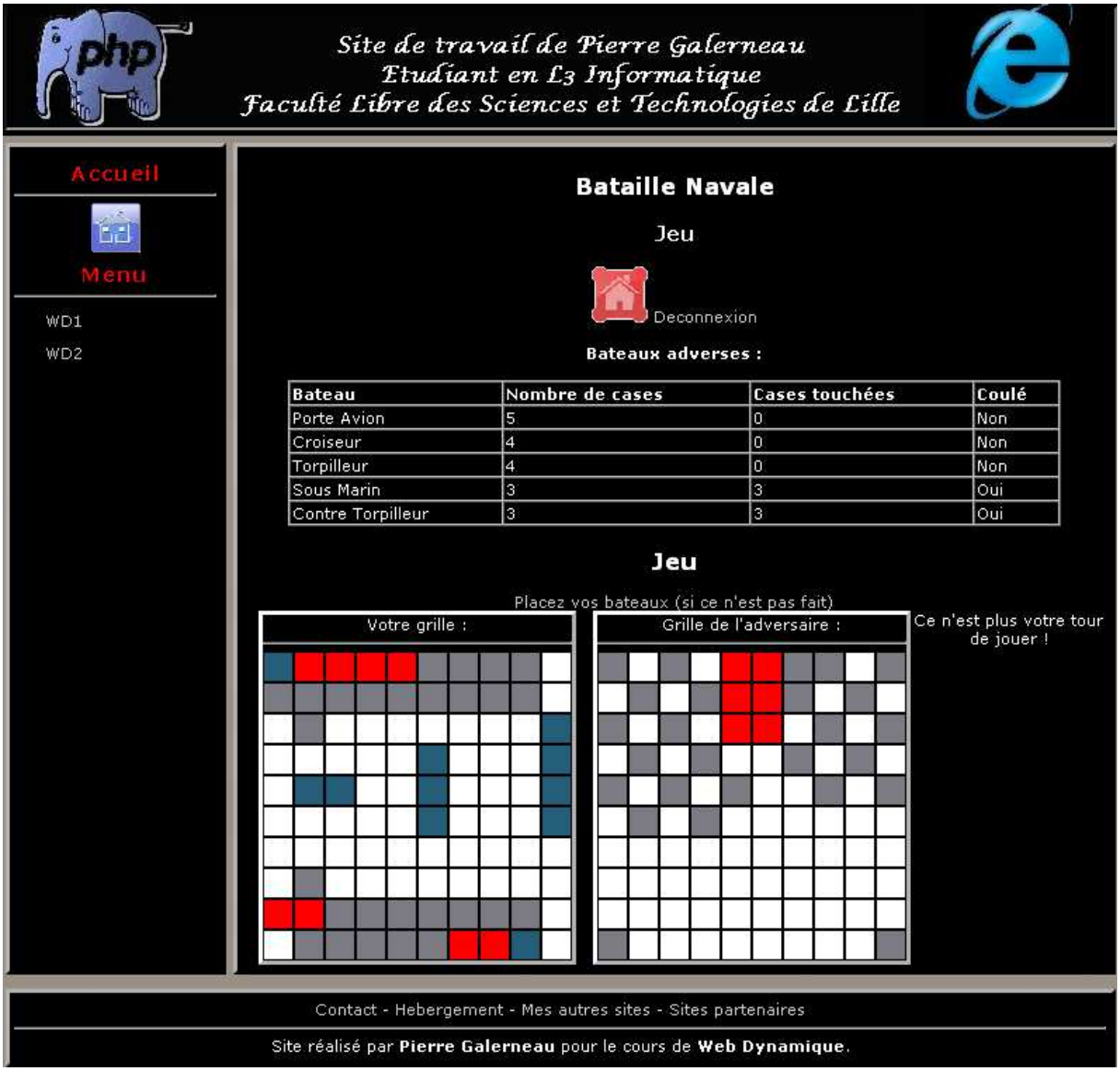
Cette vue permet à l'utilisateur de visualiser les bateaux qu'il a mis en place. Il peut alors, à ce moment là, passer à la phase de jeu réel au moyen du lien « commencer la partie » qui est apparu en bas de la page.

Celui-ci l'envoie vers :

La page jouer.php

Cette page est celle qui est « commune » aux deux utilisateurs. C'est ici que les deux joueurs vont s'affronter et vont pouvoir tirer, examiner leur grille et celle de l'adversaire, etc. Pour cela, j'ai mis en place plusieurs choses :

Lorsqu'un utilisateur n'a pas le droit de jouer (ce n'est pas son tour), il ne voit s'afficher que sa grille, celle de ses tirs et les bateau qu'il a touchés de la manière suivante :

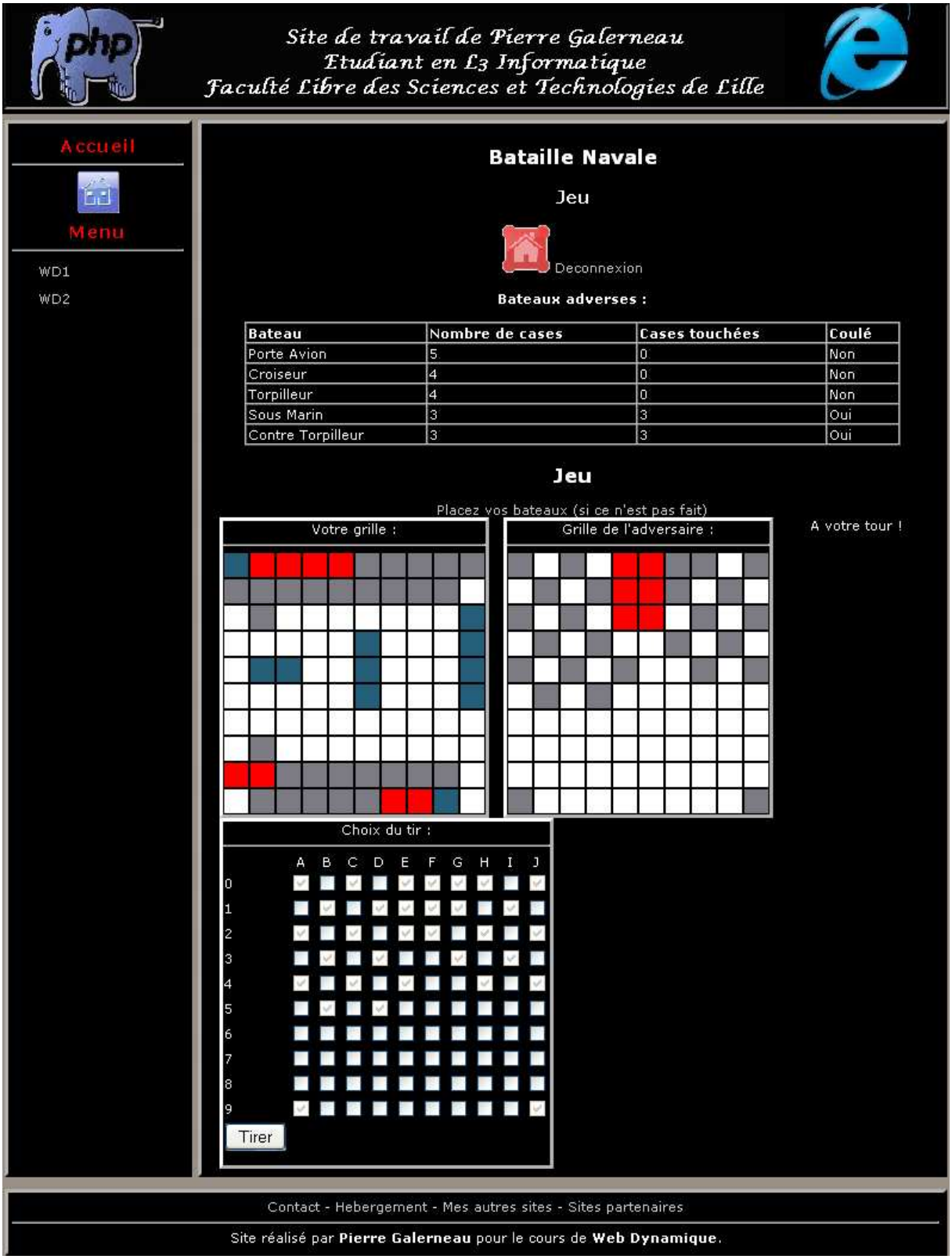


Sur cette page, on voit donc s'afficher différentes choses :

- Un tableau montrant tous les bateaux possédés par l'adversaire, leur nombre de cases et le nombre de cases touchées (et donc si le bateau est coulé ou non),
- un tableau d'images dont les cases correspondent au placement des bateaux du joueur courant et sur lequel apparaissent les tirs de l'adversaire,
- un tableau dans lequel les tirs du joueur apparaissent afin de préciser sur quelle case du jeu de l'adversaire le joueur courant a déjà tiré,
- un message spécifiant si c'est le tour de l'utilisateur ou non.

Lorsque c'est le tour de l'utilisateur courant de jouer, alors le message change en disant « c'est votre tour » et un affichage supplémentaire apparait alors. Celui-ci permet au joueur dont c'est le tour de choisir sur quelle case il va tirer.

Encore une fois, lorsque l'utilisateur a déjà tiré sur certaines cases de la grille adverse, celles-ci seront grisées dans la grille de tir afin de permettre à l'utilisateur de visualiser plus facilement ce qu'il a déjà fait et de lui éviter de retirer sur une case déjà « explorée ».



Cet enchainement de page se continue jusqu'à ce que l'un des deux joueurs gagne la partie.

Il faut remarquer que les pages du site s'actualisent automatiquement toute les 60 secondes ce qui permet à l'utilisateur de visualiser simplement le moment où c'est son tour de jouer. En effet, actualiser la page soit même n'est pas très pratique donc cette technique permet de le faire simplement et d'éviter à l'utilisateur de devoir passer sa vie à actualiser la page. J'ai

choisi d'actualiser toutes les 60 secondes, ce qui est déjà un laps de temps assez long, car, sinon, je trouvais que c'était un peu trop court, cela risquait de gêner l'utilisateur dans son jeu. En effet, si la page s'actualise très rapidement, l'utilisateur peut être gêné au moment de tirer par exemple par le fait que l'affichage « saute » tout le temps, ce qui l'empêche de pouvoir bien visualiser ce qui se passe et l'endroit où il souhaite tirer. C'est donc pour une question de simplicité que j'ai choisis d'actualiser toute les minutes même si cela peut paraitre un peu long par moment.

Enfin, lorsque la partie est terminée, c'est-à-dire que l'un des deux joueurs a remporté la partie en coulant tous les bateaux de l'autre, un message s'affiche de manière à prévenir les deux joueurs que la partie est terminée. Lorsque l'utilisateur courant a gagné, la page prend la forme suivante :

Celle-ci permet donc à l'utilisateur de voir qu'il a gagné, mais également de l'empêcher de rejouer puisque les formulaires de jeu de d'affichage ne s'affichent plus.

Lorsque l'utilisateur courant a perdu, l'affichage est exactement le même. La seule différence est que le message est « Vous avez PERDU ! Désolé » à la place de celui présenté ici.

Il ne reste donc plus aucune page servant à jouer à présenter. En effet, la seule chose dont je n'ai pas parlé est le fait que l'on puisse reprendre une partie commencée mais non terminée en cliquant sur le lien associé dans la page d'accueil. Cependant, cela ne sert pas à grand-chose

de présenter cela puisqu'en réalité cette page ne fait que renvoyer vers la page de jeu : « jouer.php » qui récupère l'identifiant de cette partie et permet donc à l'utilisateur de la reprendre là ou elle en était.

Enfin, il reste la page permettant la déconnexion de l'utilisateur :

La page deco.php

La page de déconnexion permet à l'utilisateur de quitte le site au moyen d'une destruction de sa session. Il voit alors apparaitre un message lui signifiant qu'il s'est bel et bien déconnecté du site ainsi qu'un lien lui permettant de retourner au formulaire de connexion de la page d'accueil s'il le souhaite.

Voici l'affichage correspondant :

Cette page permet également, comme nous le verrons ensuite, de supprimer l'utilisateur de la table bataille_usersconnectes afin qu'il n'apparaisse plus lorsque l'on demande l'affichage des utilisateurs en ligne.

3.2 Les sessions dans la bataille navale

Lorsque j'ai eu à décider comment j'allais organiser les données au sein de mon site et comment j'allais différencier les utilisateurs afin qu'ils puissent avoir leurs statistiques propres (nombre de victoires,…), j'ai décidé d'utiliser, logiquement, l'objet PHP « session ». Cependant, contrairement aux TP précédents, je n'ai pas eu à mettre en place des sessions permettant de différencier certains utilisateurs de certains autres. En effet, dans ce site, je n'ai pas vu l'intérêt de mettre en place des comptes administrateur puisque en réalité, aucun administrateur n'existe. Ce site est simplement destiné à permettre à différentes personnes de jouer à la bataille navale entre elles et en ligne. Aucun utilisateur n'a plus de privilèges que les autres en dehors du créateur du site qui peut accéder à toutes les données par l'intermédiaire de la base de données.

Comme je n'ai pas vu l'intérêt de créer un compte administrateur et de lui octroyer des droits spéciaux, je me suis contenté de créer un type d'utilisateur unique. Ceci m'a donc évité d'avoir à faire de nombreux tests afin de déterminer si oui ou non l'utilisateur essayant de se connecter à telle ou telle page était bien du type autorisé. En effet, aucune restriction de ce type ne s'appliquait dans ce site puisque tous les utilisateurs devaient être du même type.

J'ai donc simplement, pour chaque utilisateur, créé une session contenant au départ :

- Son login,
- Son identifiant.

Ceci s'est fait de la manière suivante :

A la connexion, lorsque l'utilisateur demande la connexion, on vérifie dans la base de données si le login existe. Si oui, alors on vérifie si le mot de passe correspondant est égal à celui qui a été saisi. Si ce n'est pas le cas, alors on affiche une erreur, sinon, on affiche un message spécifiant que l'utilisateur est connecté sous tel ou tel pseudonyme tout en enregistrant les données précédemment vérifiées dans le tableau associé à la session de l'utilisateur de la manière suivante :

Code

```
$_SESSION['login']=$login ;
$_SESSION['id']=$id ;
```

Cette session est ensuite transmise de page en page, au fur et à mesure que l'utilisateur se déplace au sein du site grâce à la ligne suivante :

Code

```
Session_start() ;
```

De fait, cette ligne permet la création d'une nouvelle session cependant, elle permet également d'autres choses. En effet, elle cherche à récupérer une session existante dans la page précédente. Si elle n'en trouve pas, alors elle en créera une nouvelle. C'est cette propriété que nous utilisons dans notre page de connexion pour la création de la session. Cependant, dans les autres pages du site, comme une session existera déjà dans la page précédente, cette ligne de code la récupérera et donc les variables enregistrées dans celle-ci seront accessibles de n'importe quelle page du site à partir du moment où cette ligne se trouve bien à la tête du fichier correspondant.

Voila donc comment sont créées les sessions lors de la connexion d'un utilisateur. Cependant, elles ne restent pas éternellement comme cela au fur et à mesure de la navigation de l'utilisateur sur notre site. En effet, lorsqu'un utilisateur choisit de jouer une partie ou d'en reprendre une qu'il avait en cours auparavant, on va récupérer au moyen de requêtes SQL les données correspondants à la partie et à son adversaire et les enregistrer dans la session de l'utilisateur courant.

Ceci permet en effet de ne pas avoir à rechercher à longueur de temps ces informations en recopiant bêtement des requêtes effectuant cela. En effet, quelle que soit la page du site, le joueur « contiendra » toujours les informations correspondants à sa partie et à son adversaire, ce qui permet de les avoir sous la main facilement et de manière plus fiable que lorsque l'on doit aller les chercher nous même à chaque début de page

Ceci permet donc un gain en lisibilité du code et surtout en clarté au moment de la programmation puisque cela évite de devoir copier / coller certaines portions de code Sauvegarder ces informations dans la session permet de les transférer de page en page et donc de n'avoir aucun problème de récupération de celles-ci Il suffit pour cela d'accéder à la case correspondante du tableau $_SESSION de la manière suivante :

Code

```
$variable=$_SESSION['partie'] ;
```

La variable $variable contiendra alors la valeur contenu dans le tableau session correspondant au numéro de la partie jouée actuellement par l'utilisateur.

De même, lorsque l'on souhaite retirer ces informations du tableau $_SESSION (notamment lorsqu'un utilisateur a terminé sa partie), il suffit de remettre la case correspondante à 0 ou de la supprimer au moyen de la ligne suivante :

Code

```
session_unregister("variable");
```

3.3 Quelle présentation adopter

L'un des objectifs principaux de notre travail en cours de Web Dynamique est de réaliser des sites les plus professionnels possibles. Il faut donc, pour cela, réaliser les interfaces les plus conviviales possibles afin que l'utilisateur ai du plaisir à naviguer sur notre site et n'hésite pas à y revenir régulièrement.

Pour commencer, j'ai donc décidé de mettre un titre. Comme dans le TP précédent, je n'ai pas fais de bannière afin de ne pas surcharger le design du site. Cependant, si ce site devait être réalisé dans un autre contexte, il serait certainement agrémenté d'une bannière, notamment si c'était un site indépendant, c'est-à-dire qu'il ne soit pas intégré au sein d'un site contenant comme c'est le cas ici.

Voici le titre :

Bataille Navale

De plus, j’ai décidé de mettre un petit menu permettant à l’utilisateur de se rendre sur les pages accessibles de partout, c'est-à-dire :

- La page d’accueil,
- La page de déconnexion.

J’ai réalisé ce menu au moyen d’images (que j’ai déjà utilisées pour d’autres sites présentés dans les comptes rendus précédents). J’ai fais cela dans le but d’embellir la page du site et surtout d’offrir à l’utilisateur l’impression qu’il est sur un site professionnel.

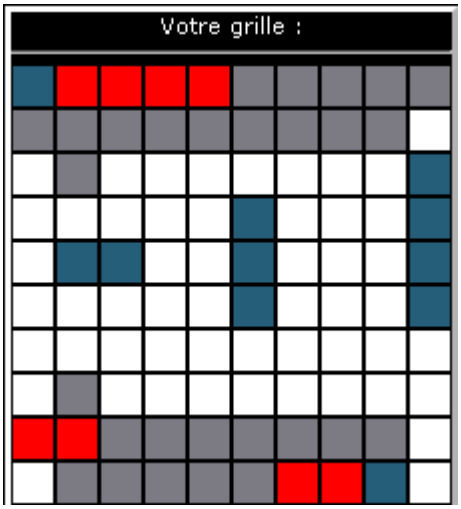
Voici le menu :

Enfin, ce site étant un site de jeu, j’ai décidé de rendre l’affichage des pages de jeu le plus convivial possible. Pour cela, j’ai créé des grilles (affichant la position des bateaux du joueur, etc) à l’aide d’images :

- Un carré blanc pour une case vide,
- Un carré bleu pour une case contenant un morceau de bateau,
- Un carré rouge pour un morceau de bateau touché,
- Un carré gris pour une case vide touchée (un tir dans l’eau).

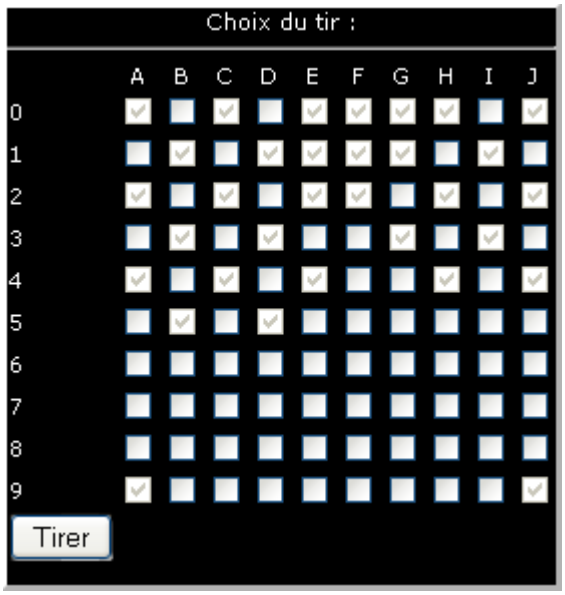
Ceci permet, je trouve, de rendre le site plus attractif et plus agréable à utiliser. De plus, le jeu est plus « réaliste » grâce à cela. Je veux dire par là que ce jeu ressemble plus à tous ceux que l’on peut rencontrer d’habitude de cette manière.

Voici cet affichage :



De même, j’ai réalisé une grille de cases à cocher permettant de choisir graphiquement la case dans laquelle on veut tirer et pour que le jeu soit plus agréable et plus simple, j’ai décidé de griser les cases dans lesquelles l’utilisateur a déjà tiré. Ceci ne sert à rien dans l’absolu pour le jeu, cependant, c’est dans un but de faciliter le jeu et d’améliorer le confort de jeu du joueur que j’ai fais cela. Cela évite en effet de devoir calculer combien de cases on doit « sauter » pour tirer dans la bonne case. Cela simplifie grandement, je trouve, l’utilisation du site et donc le jeu et le rend vraiment plus pratique à mon gout.

Voici l'affichage de la grille de tir contenant les cases grisées dont je viens de parler :



Enfin, j'ai choisi d'afficher un tableau contenant la liste de tous les bateaux afin que l'utilisateur puisse visualiser les bateaux de l'adversaire et qu'il puisse avoir sous les yeux le nombre de bateaux restants à couler et le nombre de cases de chaque bateau.

Voici ce tableau :

Bateaux adverses :

Bateau	Nombre de cases	Cases touchées	Coulé
Porte Avion	5	0	Non
Croiseur	4	0	Non
Torpilleur	4	0	Non
Sous Marin	3	3	Oui
Contre Torpilleur	3	3	Oui

L'utilisation d'un tableau comme celui-ci est à double tranchant. En effet, cela permet de visualiser tres simplement là où m'on en est dans la partie et c'est donc pratique ; cependant, cela simplifier aussi beaucoup le jeu dans le sens où cela permet de déterminer quel bateau viens d'être touché par notre tir. On sait donc si on a touché un porte avion ou un torpilleur.

Cependant, je trouve que cela est pratique du point de vue du joueur et je ne trouve pas cela très gênant de savoir quel bateau on vient de toucher puisque les deux joueurs d'une partie ont accès à ce tableau. Comme tout le monde a accès aux mêmes informations, cela n'avantage personne et donc je ne vois pas pourquoi cela serait géant.

Maintenant que toutes mes motivations et mes choix sont expliqués, il me faut présenter et expliquer la programmation réalisée.

4 Explication du code

4.1 Arbres programmatiques et explication hors programmation

Dans ce paragraphe, je vais expliquer les algorithmes principaux hors langage de programmation à l'aide d'arbres programmatiques afin de rendre accessible à tous la compréhension des algorithmes nécessaires au bon fonctionnement de ce site.

4.1.1 Algorithme de connexion

Cet algorithme est couplé avec la page d'accueil du site. En effet, cette page contient un formulaire de connexion qui permet à l'utilisateur de saisir les données nécessaires à sa connexion au site. Lorsqu'il valide son formulaire, l'utilisateur est alors renvoyé vers cet algorithme de connexion dont voici le fonctionnement.

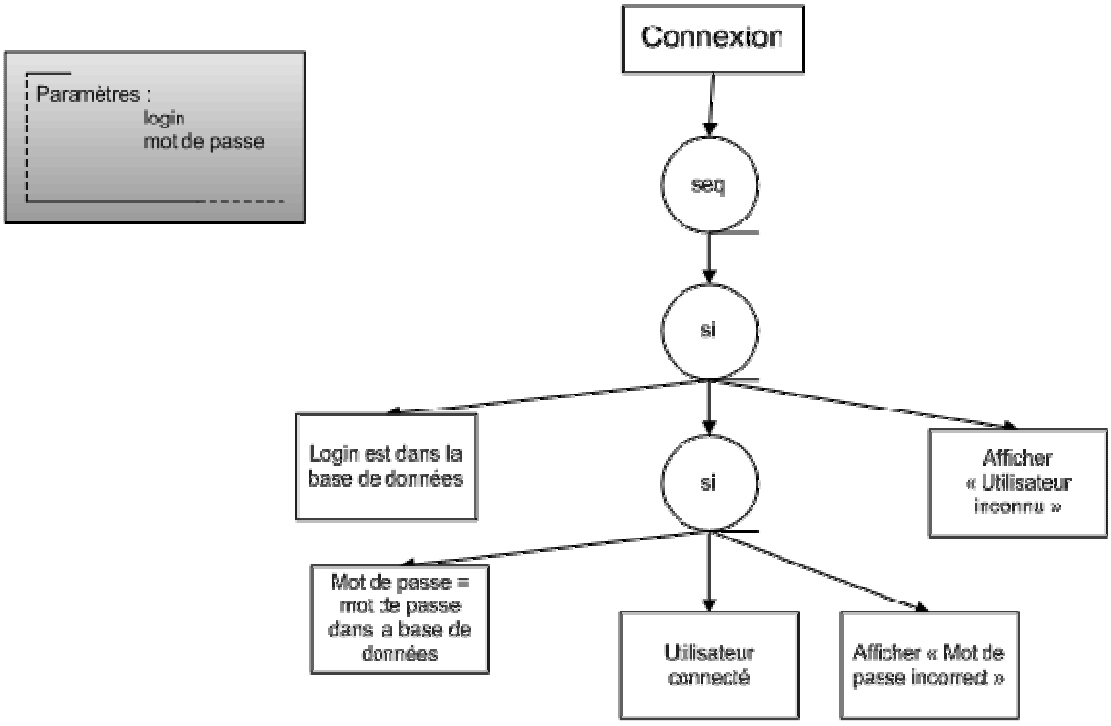
Cet algorithme est simple. En effet, la vérification des informations saisies par l'utilisateur est plus simple à réaliser lorsque l'on utilise des bases de données que lorsque l'on les simule à l'aide de fichiers texte.

L'algorithme commence donc par récupérer les valeurs que l'utilisateur a saisies dans le formulaire de connexion. Il récupère donc deux valeurs :

- Un login ;
- Un mot de passe.

Ces valeurs doivent maintenant être vérifiées. En effet, il faut que le login existe dans la base de données pour que l'on puisse connecter l'utilisateur. De plus, le mot de passe qu'il a saisi doit être identique à celui qui est enregistré dans la base.

Il suffit donc de « chercher » dans la base bataille_users une ligne dont le login est égal à celui qui a été saisi par l'utilisateur. S'il n'y en a pas, alors on affiche un message spécifiant que le login n'existe pas. Par contre, si une ligne est trouvée, alors on vérifie que le mot de passe enregistré est bien identique à celui que l'utilisateur a saisi dans le formulaire de connexion. Si c'est le cas, alors l'utilisateur est connecté, sinon, un message lui est affiché, signifiant que son mot de passe est incorrect.

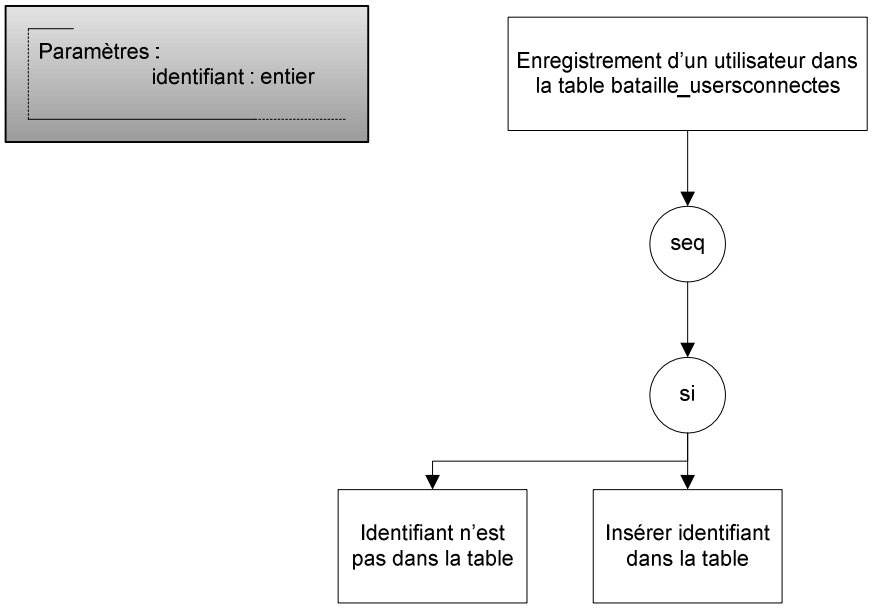


L'objectif de cet algorithme est, comme nous l'avons dis plus haut, de connecter un utilisateur au site ou, tout au moins, de vérifier que les informations qu'il a entré comme informations de connexion sont valides.

Une fois connecté, il faut sauvegarder les informations qui permettront de reconnaitre cet utilisateur de page en page dans les variables de session.

Quand il est connecté, l'utilisateur peut accéder à toutes les pages du site qu'il souhaite, cependant, il n'est pour l'instant visible par personne et ne peut donc pas être invité à jouer. Il peut donc difficilement jouer une partie. Afin de le rendre « visible », il faut l'ajouter à la table bataille_usersconnectes afin que son identifiant apparaisse dans cette table. Cela signifiera alors qu'il est connecté et qu'il est disponible pour une éventuelle partie.

L'algorithme permettant de l'ajouter à la table bataille_usersconnectes est tres simple. En effet, il s'agit d'une simple insertion dont le schéma est le suivant :



Lorsque ces valeurs sont sauvegardées et que l'utilisateur est connecté, la page de connexion (connect.php) redirige automatiquement l'utilisateur vers la page d'accueil. Ceci est « une nouveauté » dans la réalisation de mes TP. En effet, jusque maintenant je ne savais pas le faire et ma page de connexion ressemblait donc fortement à ma page d'accueil dans le sens où elles affichaient la même chose afin que, lorsque l'utilisateur se connecte, il ai l'impression d'être sur la page d'accueil. Ici, j'ai trouvé comment rediriger dès la connexion, l'utilisateur vers la page d'accueil ce qui m'évite donc les répétitions de code habituelles. Ceci est fait au moyen d'une portion de code JavaScript dont voici la teneur :

Code

```
<script language=javaScript
document.location.replace("accueil.php");
</script>
```

Je reviendrais plus longuement là-dessus dans les parties suivantes de mon rapport.

Je vais maintenant expliquer un autre algorithme important : celui qui s'exécute lors du chargement de la page d'accueil du site.

4.1.2 La page d'accueil du site

Comme je l'ai dis précédemment, la page d'accueil a deux rôles :

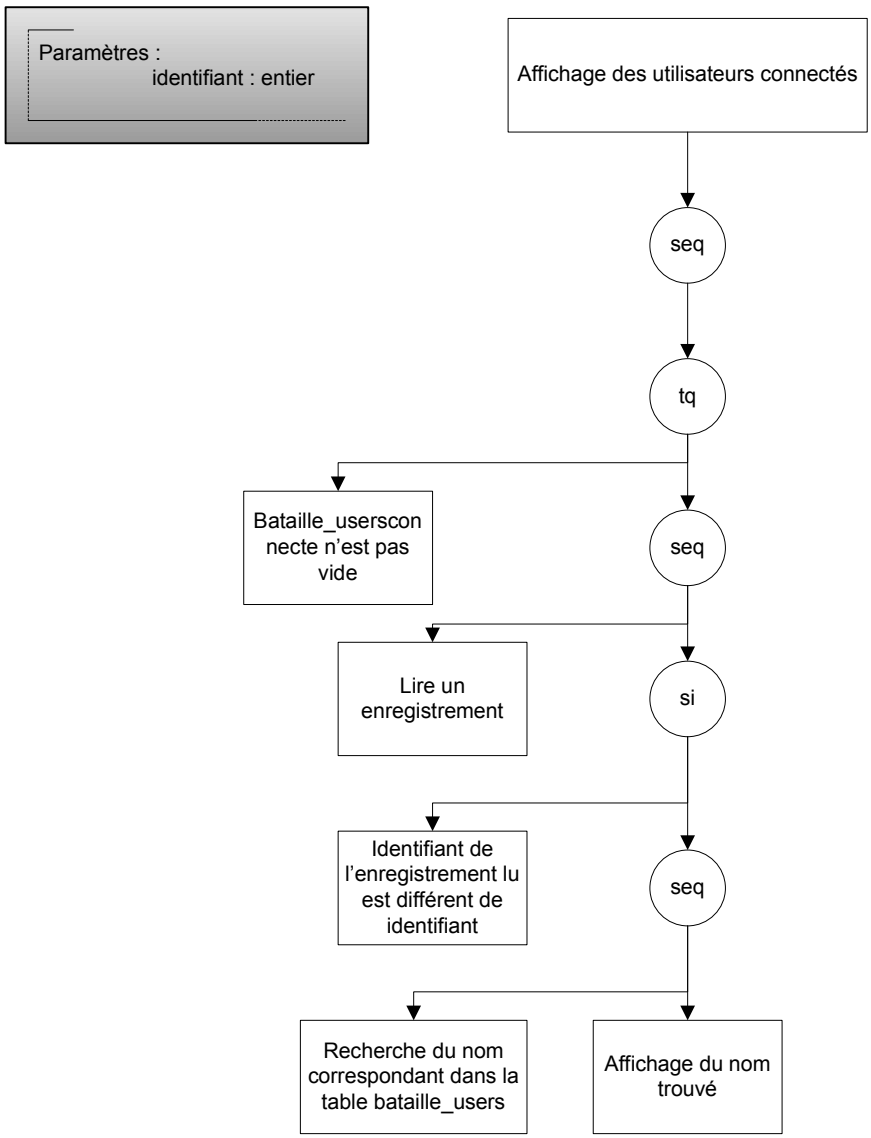
- Permettre à l'utilisateur de se connecter au moyen du formulaire de connexion qui renvoie vers la page connect.php,
- Afficher la liste des utilisateurs connectés et la liste des parties en cours du joueur.

Je viens de présenter, dans la partie précédente, la partie « connexion » de cette page, je vais donc maintenant m'attarder sur l'affichage des données lorsque la page est appelée par un utilisateur connecté.

Cet algorithme permet l'affichage de deux grandes catégories de données :

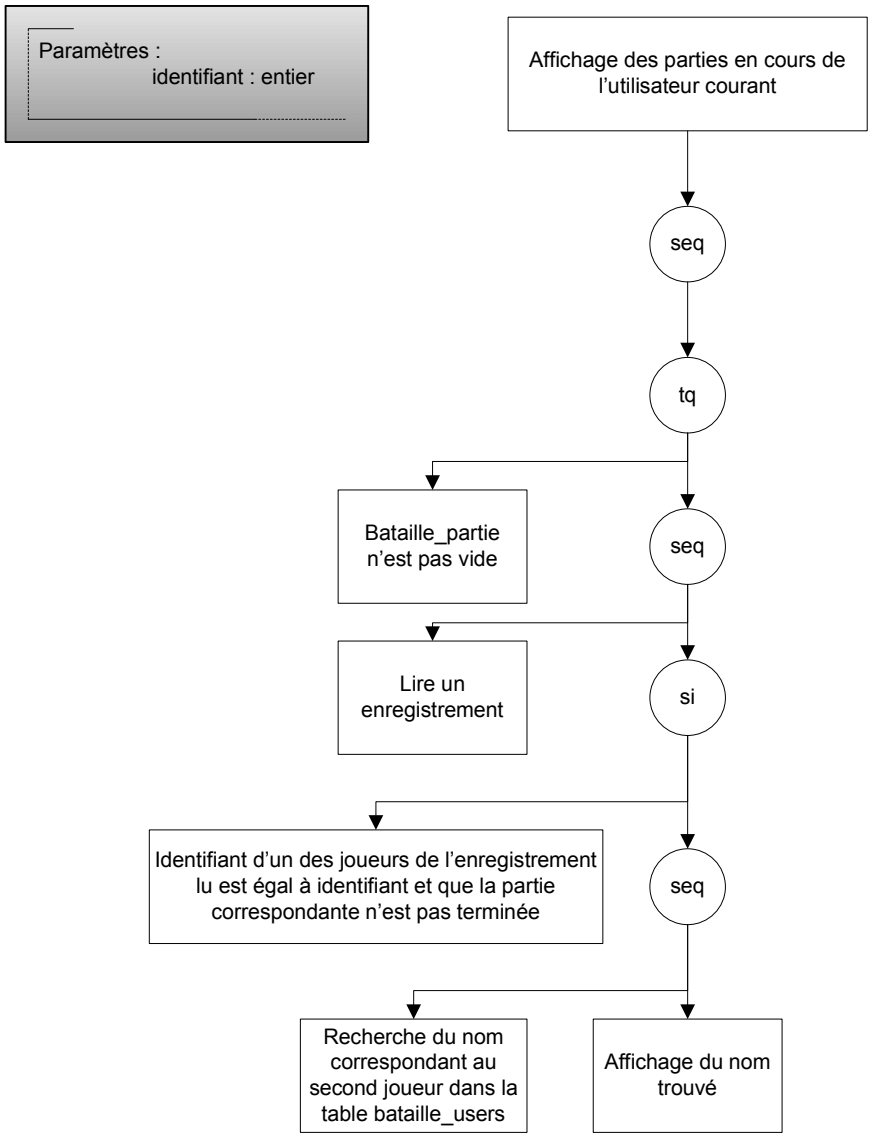
- Les utilisateurs connectés en même temps que l'utilisateur courant,
- Les parties en cours de l'utilisateur courant.

Je vais commencer par présenter l'algorithme d'affichage des utilisateurs connectés en même temps que le joueur courant. Cet algorithme se présente de la manière suivante :



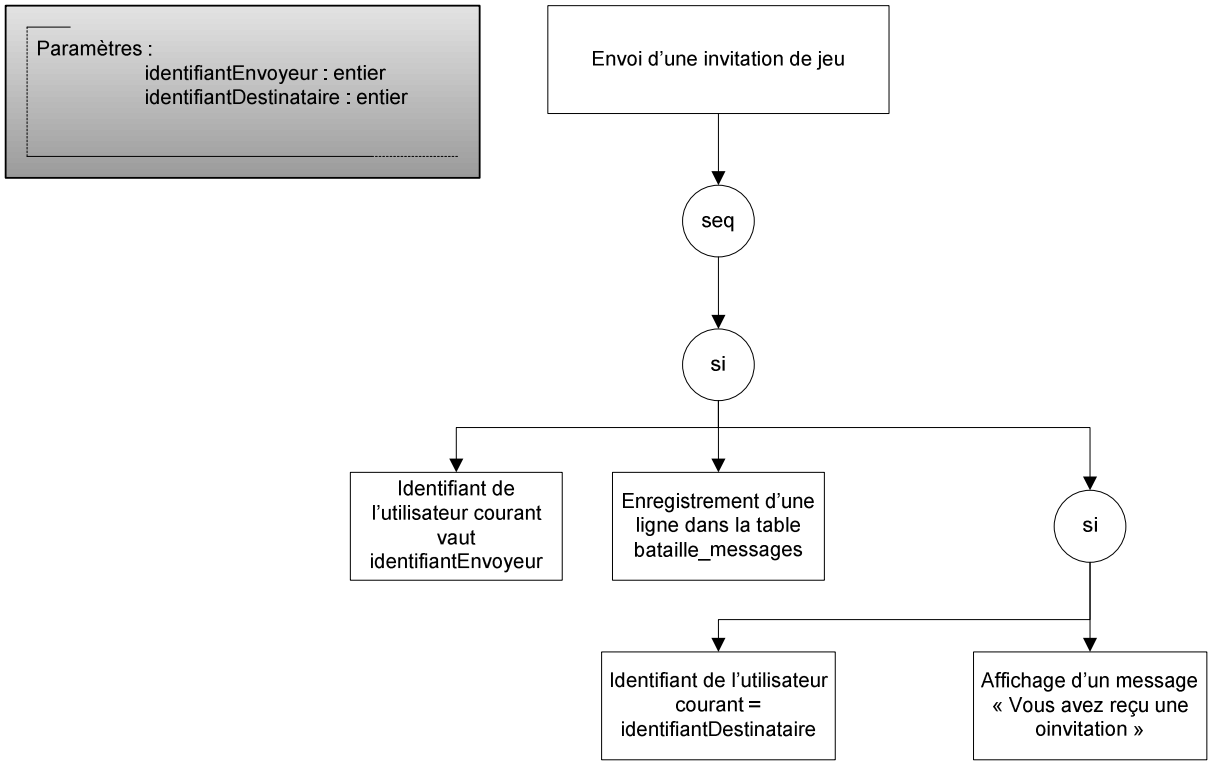
Cet affichage est fait à l'aide d'un DIV HTML dans lequel j'intègre tous les résultats de l'algorithme précédent.

Juste après l'exécution de cet algorithme, s'exécute l'algorithme permettant l'affichage des parties en cours de l'utilisateur courant. En voici la forme :



Cet algorithme se contente de lire la table bataille_partie en recherchant un enregistrement dans lequel l'un des deux identifiants de joueurs est égal à celui de l'utilisateur courant. Lorsqu'une ligne comme celle là est trouvée, alors on regarde si la partie est terminée. Si c'est le cas, on passe à l'enregistrement suivant. Sinon, on l'affiche dans la liste des parties en cours accompagné d'un lien permettant de la reprendre.

Comme je l'ai expliqué dans la page précédente, un utilisateur connecté peut voir tous les autres utilisateurs connectés en même temps que lui. Il peut également leur proposer une partie en leur envoyant une invitation au moyen du formulaire associé à la liste de noms. Lorsque ceci est fait, l'algorithme utilisé est le suivant :



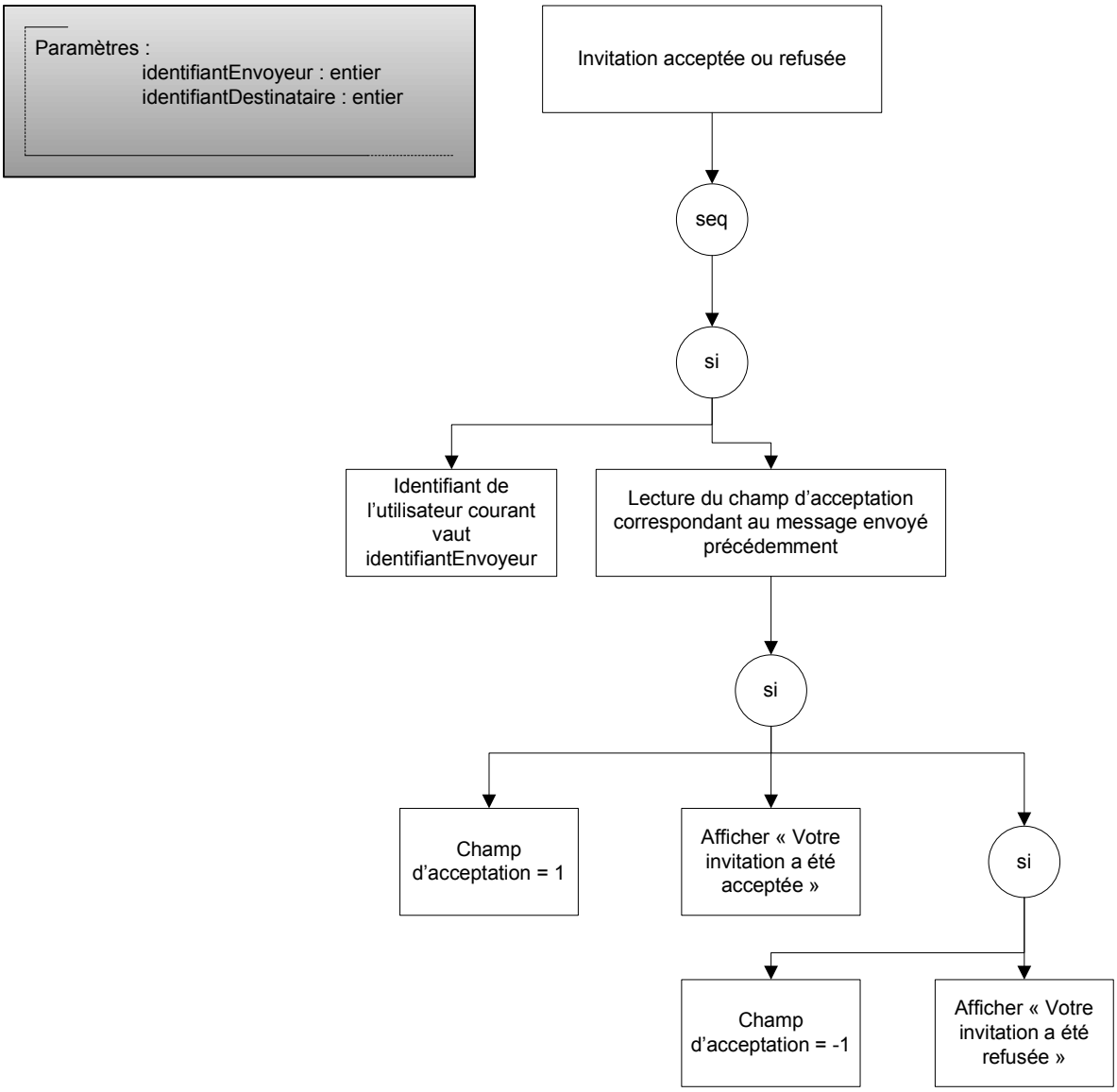
Cet algorithme permet donc, par l'intermédiaire de la table bataille_messages de faire passer l'invitation à l'utilisateur destinataire. Lorsqu'il reçoit

L'invitation, celui-ci voit alors s'afficher un message contenant le nom de l'utilisateur qui lui a envoyé l'invitation, associé à deux liens :

- Un permettant d'accepter l'invitation,
- Un permettant de la refuser.

Pour cela, des valeurs doivent être enregistrées dans la table bataille_messages afin que l'on puisse spécifier à l'utilisateur qui a envoyé l'invitation si le destinataire l'a acceptée ou refusée.

Voici l'algorithme permettant d'afficher à l'utilisateur si son adversaire potentiel a accepté ou refusé son invitation :



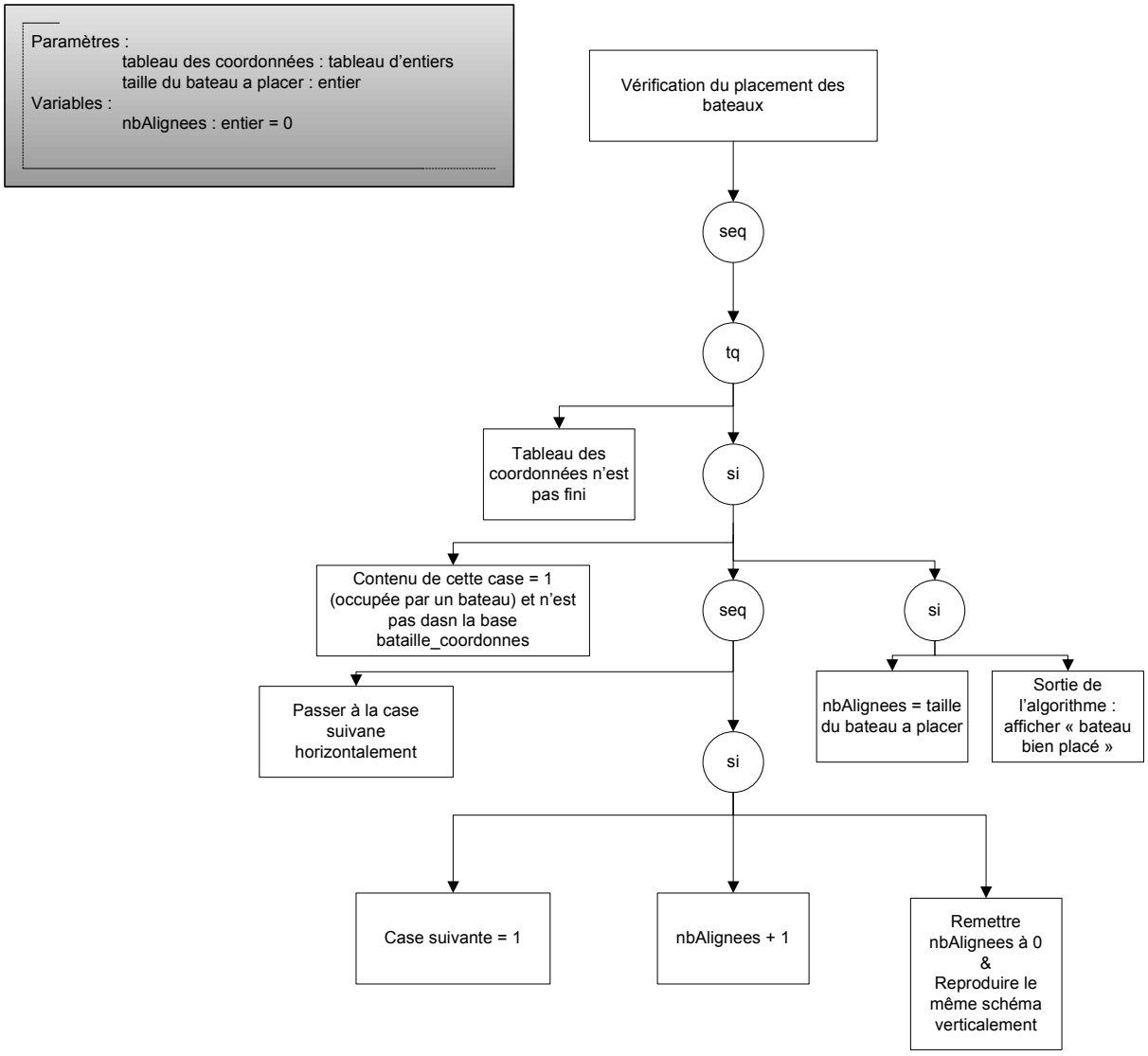
L'utilisateur est donc, de ce fait informé de ce que son adversaire potentiel a répondu à son invitation. Lorsque l'invitation est acceptée, un lien est alors affiché afin de permettre aux deux utilisateurs de commencer la partie. Celui-ci renvoie vers la page placer.php qui permet de placer les bateaux. Je vais d'ailleurs maintenant présenter cet algorithme de placement des bateaux.

4.1.3 Placement des bateaux, la page placer.php

Cette page permet à l'utilisateur qui commence une partie de placer mes bateaux qu'il n'a pas encore placés sur sa grille de jeu. Pour cela, lorsqu'il se présente sur la page, un formulaire lui est présenté afin de lui montrer quels bateaux il n'a pas encore placé sur sa grille de jeu. Il peu d'ailleurs, grâce à ce formulaire, choisir le bateau qu'il va placer au tour suivant. En validant ce formulaire, il voit alors apparaitre une grille lui permettant de placer le bateau sélectionné. Pour cela, il lui suffit de cocher les cases dans lesquelles il souhaite que le bateau soit. Une fois ceci fait, il valide le formulaire, ce qui entraine l'exécution d'une fonction de vérification. En effet, un bateau ne doit pas faire plus de cases que ce qui est prévu (5 pour un porte avion par exemple), mais il ne doit pas non plus en faire moins ! Deux bateaux ne doivent pas

pouvoir contenir la même case afin que l'on n'ai pas de superposition de bateaux, des bateaux ne doivent pas pouvoir être mis en diagonale, etc.

Voici le principe simplifié de cet algorithme :



Cet algorithme permet donc de vérifier l'alignement des cases du bateau et permet donc de valider, ou non le placement d'un bateau sur la grille de jeu. Lorsque le bateau est bien placé, l'algorithme redirige l'utilisateur vers la page placer.php dans laquelle le choix permettant de placer le bateau qu'il vient de placer a été supprimé. Il peut alors créer un nouveau bateau s'il n'a pas terminé ou simplement commencer la partie lorsque sa grille est complètement remplie (tous les bateaux sont placés correctement).

Lorsque l'utilisateur choisi de jouer, il clique sur le lien l'envoyant vers :

4.1.4 La page jouer.php

C'est sur cette page que l'utilisateur va réellement pouvoir jouer et tirer sur les bateaux de l'adversaire. Pour cela, il disposera d'une grille de cases à cocher. Il pourra alors en sélectionner une (la cocher), et c'est dans cette case que sera effectué le tir.

Les coordonnées du tir seront alors enregistrées dans la table bataille_tirs et, lorsque l'un des bateau de l'adversaire sera touché, la case correspondante dans la table bataille_positions sera passé à touché, c'est-à-dire que le champ touche sera actualisé et passé à 1.

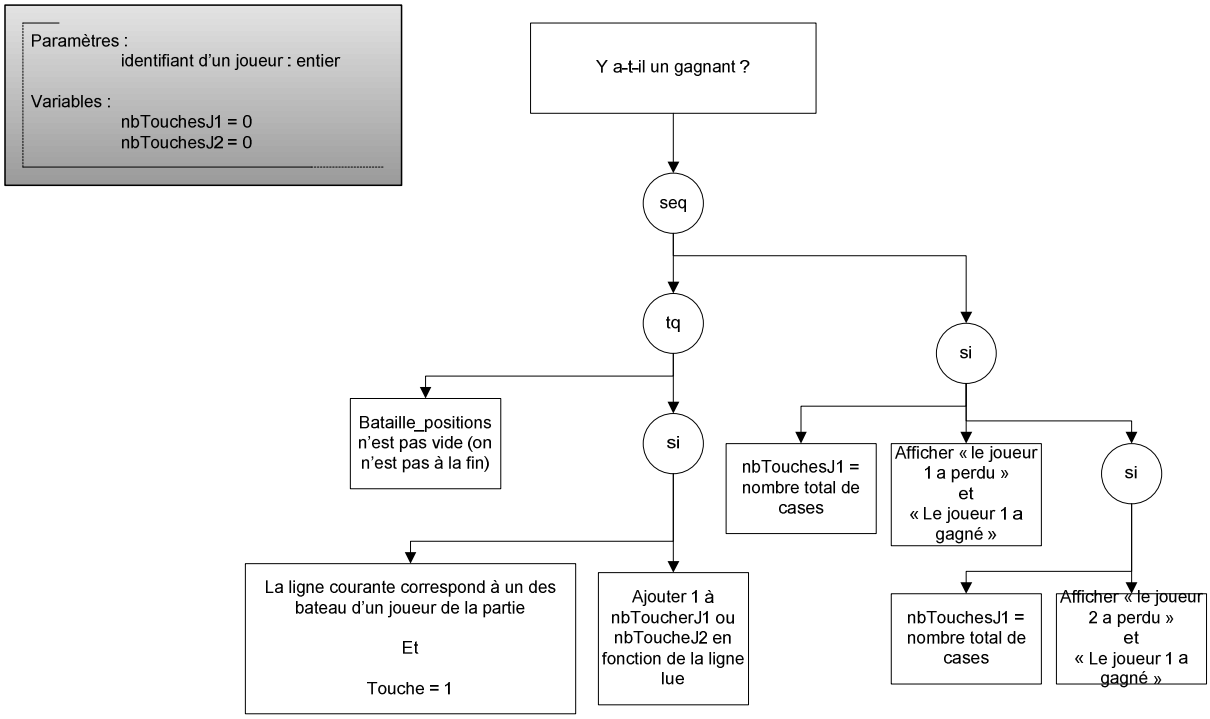
L'algorithme de jeu n'est pas intéressant à présenter puisqu'il se contente en faite d'insérer les coordonnées du tir du joueur dans la table bataille_tirs, sans même vérifier que l'utilisateur n'a pas déjà tiré à cet endroit. L'algorithme permettant le tir et ceux permettant les affichages des grilles (la grille du jeu de l'utilisateur courant et celle de son adversaire) ne sont pas très intéressants car ils sont identiques à ceux que j'ai présenté ci-dessus. Je ne vais donc pas faire d'arbres programmatiques pour les expliquer mais je vais plutôt expliquer l'algorithme permettant de déconnecter l'utilisateur courant ainsi que celui permettant de vérifier que l'un des deux utilisateur a gagné..

4.1.5 Algorithme de vérification du gagnant

A chaque tour de jeu, lorsqu'un utilisateur « passe la main » à l'autre, il faut savoir si l'utilisateur qui vient de jouer a gagné ou non. Si il a gagné, alors il faut « arrêter » le jeu en interdisant aux deux joueurs de jouer, sinon il faut leur permettre de jouer à nouveau.

Cet algorithme de vérification du gagnant se base simplement sur la table bataille_positions. Afin de déterminer s'il y a un gagnant, l'algorithme « regarde » dans la table bataille_positions si toutes les cases des bateaux d'un joueur sont touchées. Si c'est le cas, alors ce joueur a perdu et son adversaire a gagné !

Voici sa forme :



On enregistre ensuite l'identifiant du gagnant dans la table bataille_parties afin que l'on puisse « se souvenir » de qui a gagné quelle partie. De même, on met à jour le nombre de victoires et de défaites des participants de la partie à l'aide d'une requête de mise à jour de type update sur la table bataille_users.

4.1.6 La déconnexion : la page deco.php

Lorsqu'un utilisateur a terminé de jouer ou qu'il n'a pas trouvé de partenaire, il peut se déconnecter au moyen du lien « déconnexion » du menu. Ce lien l'envoie alors vers la page deco.php qui lui permet de supprimer sa session et de disparaitre de la table bataille_usersconnectes. En effet, si il se déconnecte, il ne doit évidemment plus apparaitre en temps qu'utilisateur connecté dans cette table de la base de données !

L'algorithme permettant cela est très simple. En effet, il se contente de supprimer la session de l'utilisateur au moyen de la ligne suivante :

Code

```
Session_destroy();
```

Puis il supprime l'utilisateur de la table bataille_usersconnectes. Ceci se fait au moyen d'un algorithme simple :

- On recherche dans la table bataille_usersconnectes la ligne correspondant à son identifiant,
- On supprime cette ligne.

L'utilisateur est alors complètement déconnecté et n'apparaitra plus – tant qu'il ne se reconnectera pas – dans la liste des utilisateurs connectés actuellement.

4.2 Détail des fonctionnalités de chaque page

Maintenant que j'ai expliqué les principaux algorithmes et leur forme ci-dessus, je vais parcourir tous les fichiers du projet pour les expliquer un à un afin de m'arrêter plus particulièrement sur chaque particularité de programmation dans ce que j'ai développé. Je vais donc tout expliquer en détail afin que tout le coté « code » soit expliqué et clair.

4.2.1 La page d'accueil

Lorsqu'un utilisateur se connecte sur mon site de jeu de bataille navale, il a plusieurs choix dès son arrivée sur la page de connexion du site. Il peut

- Se connecter (page accueil.php[1]),
- Créer un compte (page créer_compte.php).

Si il choisit de se connecter, alors il lui suffit de remplir le formulaire de connexion composé de deux champs HTML (de type texte et de type password) et de le valider afin d'être redirigé vers la page connect.php qui effectuera les tests validant ou non la connexion. Ce formulaire de connexion HTML ne s'affiche que dans le cas où l'utilisateur demandant l'affichage de cette page n'est pas connecté. Ceci est fait au moyen de la ligne de test suivante :

Code

```
if(!isset($_SESSION['login'])){
```

Qui permet de vérifier que la session est initialisée ou non dans la page courante. Si ce n'est pas le cas, donc, le formulaire de connexion est affiché et on a donc le code suivant :

[1] Le code correspondant à la page accueil.php est disponible en annexe 1.

Code

```
Pour jouer, connectez vous :<br><br>
<div style="{border-width:1px; border-color:red; border-style:outset;}">
<table class="table">
<form method=post action=connect.php>
<tr><td>Login :</td><td><input type=text name=login size=10></td></tr>
<tr><td>Mot de passe :</td><td><input type=password name=mdp size=10></td></tr>
<tr><td></td><td><input type=submit value="Connexion"></td></tr>
</form>
</table>
<a href=creer_compte.php>Creer un compte</a></div>
```

Celui ci permet l'affichage du code correspondant à l'affichage du formulaire HTML de connexion ainsi que d'un lien permettant d'accéder à la page de création de compte dont je détaillerais le fonctionnement plus tard.

Lorsque l'utilisateur accédant a cette page possède déjà une session initialisée, qu'il est connecté donc, la page d'accueil n'affichage pas de formulaire de connexion puisque le test présenté ci-dessus est alors faut. Cependant, l'affichage n'est pas vide ! En effet, cette page permet l'affichage de la liste de tous les utilisateurs connectés au site (contenus dans la table bataille_usersconnectes). Ceci se fait simplement au moyen d'une requête SQL de type select dont la syntaxe est la suivante :

Code

```
select bataille_users.idUser, bataille_users.login, bataille_users.nbVictoire from bataille_usersConnectes, bataille_users where bataille_users.idUser=bataille_usersConnectes.idUser;
```

L'exécution de cette requête permet donc, au moyen d'une jointure, de récupérer des données telles que :

- Le login de l'utilisateur,
- Le nombre de victoires de l'utilisateur,
- L'identifiant de l'utilisateur,

Concernant tous les utilisateurs présents dans la table bataille_usersconnectes. Celle-ci permet donc, en une seule requête, d'obtenir toutes les informations nécessaires à la réalisation de la liste des utilisateurs connectés.

Cette liste est alors formée de la manière suivante :

- Le login d'un utilisateur connecté,
- Un champ de type bouton radio.

Ceci permet de donner la possibilité à l'utilisateur connecté de « choisir » un autre utilisateur afin d'interagir avec lui au moyen du formulaire associé.

Remarque : lorsque l'utilisateur arrive sur cette page d'accueil et que la liste des utilisateurs connectés s'affiche, son propre pseudonyme ne s'affiche pas. En effet, il devrait s'afficher puisque, comme les autres, il est contenu dans la table bataille_usersconnectes. Cependant, j'ai mis en place un test évitant cet affichage car il est inutile qu'un utilisateur se voie et puisse se proposer une partie à lui-même ! Le test est donc le suivant

- Si l'identifiant de l'utilisateur à afficher est égal à celui de l'utilisateur courant connecté, on ne l'affiche pas,
- Sinon on affiche.

Enfin, cette page permet, de la même manière, d'afficher la liste des parties en cours de l'utilisateur connecté. Ceci se fait, une fois de plus, au moyen d'une requete SQL de type select. Celle-ci sélectionne les parties de la table bataille_parties pour lesquelles l'un des deux identifiants de joueur est égal à celui de l'utilisateur connecté demandant l'affichage. De plus, elle vérifie que l'identifiant du gagnant de la partie n'est pas encore renseigné. Voici la forme de cette requête :

Code

```
select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;
```

Les résultats de cette requête sont affichés de la même manière que ceux de la requête précédente.

Maintenant que j'ai présenté cette page, je vais présenter la possibilité qui est offerte à un utilisateur de se créer un compte. Ceci se fait au moyen de :

4.2.2 La page créer_compte : création de compte de jeu

Lorsque l'utilisateur arrivant sur le site y vient pour le première fois, il ne possède pas de compte et n'a donc pas le droit de se connecter au site. Il faut donc lui donner la possibilité de s'en créer un et c'est à cela que sert le lien placé sur la page d'accueil « Créer un compte ». Celui-ci renvoie vers la page créer_compte.php qui est composée d'un simple formulaire de saisie. Celle-ci va permettre à l'utilisateur de saisir toutes les informations nécessaires à la création de son compte dans la base de données.

Lorsque ce formulaire est validé par l'utilisateur (et que donc il a rempli tous les champs), la page créer.php[2] est appelée. C'est en réalité celle-ci qui va réellement créer le compte du nouvel utilisateur.

Cette page commence bien sur par récupérer les données que l'utilisateur vient de saisir dans le formulaire. Pour cela, la ligne :

Code

```
$variable=$_POST["nom"];
```

Est utilisée autant de fois qu'il y a de variables (valeurs) à récupérer. Une fois celles-ci stockées dans des variables, il faut vérifier la validité des informations saisies par le nouvel arrivant. En effet, les données doivent avoir les propriétés suivantes :

- Nom : chaine de caractère non vide,
- Prénom : chaine de caractère non vide,
- Login : chaine de caractère non vide,
- Mot de passe 1 : chaine de caractères non vide et supérieure ou égale à 6 caractères,
- Mot de passe 2 : chaine de caractères non vide égale à la chaine mot de passe 1.

[2] Le code correspondant à la création de compte (pages creer_compte.php et créer.php) est disponible en annexe 2.

Si les données saisies par l'utilisateur ne sont pas de ce format, alors un message d'erreur sera affiché afin de le prévenir des erreurs qu'il a pu commettre. Il pourra alors les corriger et retenter de créer son compte.

Par contre, lorsque les données seront valides, le compte sera créé au moyen d'une requête SQL de type insert dont la syntaxe sera la suivante :

Code

```
insert into bataille_users values('', '$nom', '$prenom', '$login', '$pwd1', 0, 0, 0);
```

Une fois cette requête exécutée au moyen des lignes suivantes :

Code

```
$requete="insert into bataille_users values('', '$nom', '$prenom', '$login', '$pwd1', 0, 0, 0);";base=mysql_connect("localhost", "root", "");
if(mysql_select_db("galerneau_pierre", $base)){
        if($reponse = mysql_query($requete)){
                echo "Votre compte a été créé, connectez vous :";
        }
}
Mysql_close();
```

L'utilisateur sera enregistré et pourra donc se connecter au site. Un formulaire de connexion lui sera donc affiché sur la page créer.php.

Maintenant que l'on sait de quelle manière un utilisateur peut se créer un compte, il faut voir comment le programme va le connecter lorsqu'il demandera la connexion au moyen du formulaire de connexion présent sur la page d'accueil du site.

4.2.3 Connexion au site : la page connect.php[3]

Cette page a pour objectif de déterminer si les informations saisies par l'utilisateur tentant de se connecter au site dans le formulaire de la page d'accueil sont valides, c'est-à-dire bien présentes dans la base de données et plus particulièrement dans la table bataille_users.

Pour cela, l'algorithme commence par récupérer les valeurs saisies dans le formulaire précédent au moyen des lignes suivantes :

Code

```
$login=$_POST["login"];
$pwd=$_POST["mdp"];
```

Une fois ces valeurs récupérées, il faut les vérifier. En effet, pour qu'un utilisateur puisse être officiellement connecté, son login doit exister dans la base de données et le mot de passe associé doit être le même que celui saisi par l'utilisateur dans le formulaire de connexion.

Ces vérifications sont effectuées au moyen de requêtes SQL de type select. En effet, on cherche à sélectionner tous les enregistrements de la table bataille_users pour lesquels le login est égal à celui qui a été entré par l'utilisateur. Si il n'en existe pas, c'est que l'utilisateur a fait une faute de frappe où qu'il ne possède pas de compte. De plus, il ne peut pas y avoir plusieurs résultats puisque, à la création du compte, on vérifie que le login enregistré n'existe

[3] Le code correspondant à la page connect.php est disponible en annexe 3.

pas déjà. Lorsque l'utilisateur entre un login correct (qui existe dans la base), il faut alors tester le mot de passe associé. Ces tests sont faits de la manière suivante :

Requête de vérification de l'existence du login :

Code

```
select * from bataille_users where Login='$login';
```

On récupère la ligne correspondant au login saisi. Si il y en a une, alors on teste la validité du mot de passe à l'aide de la ligne suivante :

Code

```
if($reponse['motDePasse']==$pwd){
…
}
```

Enfin, lorsqu'un utilisateur est connecté, un message lui est affiché afin de lui signifier cet état de fait, et une dernière requête est exécutée. Celle-ci consiste à enregistrer l'identifiant de l'utilisateur qui vient de se connecter dans la table bataille_usersconnectes afin qu'il puisse apparaitre comme un adversaire potentiel des autres joueurs connectés. C'est une requête de type insert donc la syntaxe est la suivante :

Code

```
insert into bataille_usersConnectes values('', ".$reponse['idUser']."')
```

Enfin, lorsque cette dernière requête est exécutée avec succès, l'utilisateur est redirigé vers la page d'accueil sur laquelle il va voir apparaitre les données indispensable à son jeu, c'est-à-dire la liste des utilisateurs connectés ou bien celle de ses parties en cours. Cette redirection est utile car elle permet d'éviter que l'on ne doive recopier le contenu de la page d'accueil dans la page de connexion pour permettre l'affichage de ce qu'il faut..

Comme il est impossible en PHP de rediriger une page (PHP étant un langage serveur, interprété), il est indispensable d'utiliser un langage différent tel que JavaScript pour réaliser cette redirection. Ceci se fait donc grâce au code suivant :

Code

```
<script language=javaScript>
document.location.replace("accueil.php");
</script>
```

Ce qui permet de remplacer l'url courante par celle qui est spécifiée ici, soit accueik.php.

Lorsqu'un utilisateur est connecté et qu'il est enregistré dans la table des utilisateurs connectés, il ne lui reste plus qu'à se trouver un adversaire et à jouer. Pour cela, il a différentes options. Il peut :

- Envoyer une invitation à un autre joueur connecté,
- Reprendre une partie en cours.

Je vais commencer par présenter l'envoi d'invitation, la réception et la réponse à celles-ci.

4.2.4 Les invitations à jouer : la page verifMess.php

Cette page n'est pas une page du site à part entière. En réalité, c'est une page de code PHP externe qui est destinée à être intégrée dans le site au moyen de la fonction include de PHP :

Code

```
include("verifMess.php");
```

Cette page permet en effet de vérifier si un utilisateur a reçu un message d'un autre et de lui afficher, le cas échéant. Elle ne doit donc pas apparaitre que sur une seule page puisque l'utilisateur doit pouvoir recevoir ses messages, quelle que soit la page sur laquelle il se trouve au sein de notre site de jeu.

En réalité, cette page a plusieurs fonctions :

- Elle permet l'envoi d'un message lors de la validation du formulaire d'envoi d'invitation de la page d'accueil,
- Elle permet l'affichage des messages reçus par un utilisateur,
- Elle permet l'affichage d'un « formulaire » permettant de répondre au message reçu,
- Elle permet d'afficher la réponse au message envoyé sur la page de l'utilisateur émetteur du message.

Tout ceci se fait au moyen de vérifications dans la base de données et plus précisément dans la table bataille_messages. Je vais décomposer l'explication selon les différentes choses que cette page permet :

- Enregistrement / envoi d'un message :

Lorsqu'un utilisateur choisi d'envoyer un message à un autre joueur connecté, il sélectionne son nom au moyen de la case de type bouton radio du formulaire de choix et valide le formulaire. La page choixAdv.php[4] est alors appelée. Elle récupère alors l'identifiant de l'utilisateur à qui le message est destiné et exécute une requête SQL de type insert afin d'enregistrer le fait que l'utilisateur X a envoyé un message à l'utilisateur Y.

Voici cette requête :

Code

```
insert into bataille_messages values('', '".$_SESSION['id']."','$adv', 0);
```

Ce qui crée, effectivement l'invitation dans la table bataille_messages.

Maintenant, il faut spécifier à l'utilisateur à qui l'invitation est destinée qu'il en a reçu une. Pour cela, la page verifMess.php[5] est indispensable.

Celle-ci permet de vérifier, à chaque chargement de page, si un utilisateur (l'utilisateur courant) a reçu ou non un message. Pour cela, une requête SQL de type select est exécutée et elle est de la forme suivante :

[4] Le code correspondant à la page choixAdv.php est disponible en annexe 4.
[5] Le code correspondant à la page verifMess.php est disponible en annexe 5.

Code

```
select bataille_users.idUser as id, bataille_users.login from bataille_users, bataille_messages where
bataille_messages.idUserDestination='".$_SESSION['id']."'and
bataille_users.idUser=bataille_messages.idUserSource and bataille_messages.lu=0
```

Cette ligne permet de sélectionner toutes les invitations que l'utilisateur courant a reçu Pour chacune, un message sera alors affiché précisant quel est l'expéditeur de l'invitation et affichant deux liens permettant :

- D'accepter ou
- De refuser

L'invitation reçue. En cliquant sur le lien permettant d'accepter, un nouveau message sera affiché en spécifiant que l'invitation a été acceptée et que donc, la partie peut commencer. En parallèle de cela, la table bataille_messages sera mise à jour afin que l'expéditeur de l'invitation puisse être prévenu de l'acception de son adversaire. Ceci sera fait au moyen de la requête suivante :

Code

```
update bataille_messages set lu=1 where idUserDestination='".$_SESSION['id']."';
```

L'utilisateur ayant envoyé l'invitation sera alors mis au courant du choix de son adversaire au premier rechargement de la page sur laquelle il se trouve. Cette page verifMess.php sera ré exécutée et elle trouvera que le message envoyé par le propriétaire de la page a eu une réponse positive. Un message sera donc affiché afin de permettre à l'utilisateur de commencer la partie à son tour.

Lorsque le destinataire d'un message refuse une invitation, cela se passe de la même manière, simplement le champ lu de la table bataille_messages est passé à -1 au lieu de 1 et un message en conséquence est donc affiché aux deux utilisateurs, interdisant le début de la partie.

Maintenant que j'ai présenté le mécanisme d'envoi et d'acceptation de message entre les membres du site, je vais présenter la manière dont les joueurs ont la possibilité de :

4.2.5 Placer les bateaux, la page placer.php

Lorsqu'un joueur a démarré une partie, il doit placer les bateaux qui vont lui permettre de jouer, c'est-à-dire créer sa grille de jeu. Pour cela, lorsque la partie commence, il dispose d'un lien qui lui permet d'accéder à la page placer.php[6], sur laquelle il va pouvoir placer ses bateaux. C'est également celle-ci qui va en vérifier le placement et lui signaler qu'ils sont bien ou mal placés. Une fois un bateau bien placé, l'algorithme insérera les cases correspondantes dans la base de données (la table bataille_positions) afin de sauvegarder ces données.

Cette page présente plusieurs choses. Lorsqu'un utilisateur arrive sur celle-ci, il voit apparaitre une liste de bateaux lui signifiant combien de bateau de chaque type il va avoir le droit de placer et combien de cases occupe chaque bateau concerné. Ceci est fait au moyen d'un simple DIV HTML dans lequel est intégré du texte, ce qui permet d'afficher simplement les informations sans qu'elles soient modifiables.

A coté de ce premier élément DIV, un second est généré grace à du code PHP. Celui-ci permet d'afficher ou non le nom d'un bateau au sein d'un formulaire afin de ne l'afficher que

[6] Le code correspondant à la page placer.php est disponible en annexe 6.

si le nombre autorisé de ce bateau n'est pas encore placé sur la grille. Par exemple, on n'affichera le lien « porte avion » que tant qu'il n'y aura aucun porte avion d'enregistré sur la grille pour cette partie. Ceci permet donc d'empêcher un utilisateur d'enregistrer un nombre de bateau plus important que ce qui est autorisé. De fait, ces noms de bateaux sont intégrés au sein d'un formulaire HTML composé de cases à cocher de type radio. Lorsqu'un utilisateur choisi de placer un bateau (en le sélectionnant dans ce formulaire), on ne peut placer que celui là. La fonction de vérification n'est appelée que pour ce type de bateau, ce qui interdit tout autre placement.

L'affichage de ce formulaire ce fait à l'aide du test suivant :

Code

```
if(compteNbBateau("Porte Avion", 5, $partie)<1){
```

C'est-à-dire que l'on appelle une fonction qui permet de compter le nombre d'occurrences d'un bateau donné (dont le nom est passé en paramètre) dans la base de données pour la partie en cours. On n'affiche alors le lien permettant au joueur de créer ce bateau dans le seul cas où le résultat renvoyé par la fonction est inférieur au nombre de bateaux autorisé pour ce type de bateau précis.

Si le lien apparait et que l'utilisateur choisit de placer le bateau en question, alors une grille de cases à cocher de type checkbox apparaitra. Celle-ci permettra donc à l'utilisateur de placer le bateau sélectionné sur la grille.

Remarque : lorsque des bateaux sont déjà placés, cette grille apparait et elle contient des cases grisées : celles qui sont déjà occupées par des bateaux dans la grille du joueur concerné. Ceci est fait au moyen du test suivant :

Code

```
if(isset($tableau)){
	if(in_array("$i$j", $tableau)){
		$table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\" checked=true
disabled=false></td>";
	}else{
		$table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
	}
}else{
	$table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
}
```

Ce qui signifie que, si le tableau $tableau est initialisé (contenant les cases déjà utilisées dans la base de données), alors on désactive les cases dont l'indice est dans ce tableau et on affiche les autres normalement. Ceci permet de simplifier l'affichage de la grille et donc le placement de l'utilisateur et surtout, cela lui évite de placer des bateaux les uns sur les autres. J'avais, avant de penser à faire cela, vérifié lors de la vérification de la validité du placement des bateaux que les cases ne se chevauchaient pas, cependant, ceci est plus simple et certainement plus agréable à utiliser pour l'utilisateur !

Lorsqu'un utilisateur place donc un bateau et qu'il valide la grille de placement contenant ce bateau, la fonction de vérification du placement est alors appelée. On lui donne en paramètre le nom du bateau, sa taille et la partie pour laquelle on joue. Elle effectue donc les vérifications nécessaires à la vérification du placement du bateau, c'est-à-dire que :

- Aucune case n'est déjà présente dans la base de données (pas de chevauchement des bateaux),
- Un nombre de cases égal à la taille du bateau est aligné verticalement ou horizontalement.

Lorsque ces vérifications sont effectuées, si un bateau bien placé est trouvé, on renvoie un booléen dont la valeur est placée à vrai et on affiche que le bateau est bien placé, sinon on passe ce booléen à faux et on affiche que le bateau est mal placé.

Ce processus est donc utilisé pour le placement de chaque bateau ce qui permet de garantir un placement valide de tous les bateaux de la flotte du joueur.

Enfin, lorsque celui-ci est terminé (le placement d'un bateau ou de toute la flotte), une grille est affichée afin de montrer à l'utilisateur comment se présente sa grille de jeu. Celle-ci est identique à celle que je vais présenter dans la partie « jeu » du site à la seule différence qu'elle ne compte que des cases correspondants à du « vide » ou à un bateau et pas de cases de tir.

Enfin, lorsque toute la flotte d'un joueur est placée, il lui faut pouvoir accéder à la page suivante afin de commencer à jouer. Pour cela, un lien s'affiche signifiant que la partie est créée et que l'utilisateur peut commencer à joueur. Celui-ci renvoie alors vers

4.2.6 La page jouer.php[7], le jeu en lui même

Lorsque l'utilisateur arrive sur cette page, il vient de commencer une partie et de placer sa flotte ou alors, il vient de reprendre une partie en cliquant sur le lien « reprendre » de la page d'accueil. Dans le cas où il reprend une partie, il doit avoir la possibilité de placer ses bateaux, au cas où il ne les aurait pas encore placés. Pour cela, un lien est affiché sur cette page de eu afin de donner à l'utilisateur la possibilité de retourner à la page de placement des bateaux. Celui-ci sera toujours actif, même lorsque ses bateaux seront placés cependant, comme je l'ai expliqué précédemment, rien ne sera modifiable puisque la vérification affichant la liste des bateaux à placer renverra des valeurs correspondant au fait qu'il ne reste plus rien à placer. L'arrivée sur cette page avec une flotte déjà placée ne fera donc rien a part afficher le placement de la flotte actuelle !

Lorsqu'il voudra jouer, la page de jeu affichera différentes choses :

- La grille du joueur contenant ses bateaux et le résultat des tirs de l'adversaire,
- La grille de l'adversaire contenant le résultat des tirs du joueur,
- Un message signifiant si c'est le tour du joueur courant de jouer ou non,
- En fonction du message précédent, une grille de tir (ou rien).

Lorsqu'un joueur place ses bateaux et termine tout le placement de sa grille, une ligne est insérée dans la table bataille_aqui qui permet de définir à qui est le tour de jouer. J'ai décidé que, lorsqu'un joueur aurait terminé le premier, il donnerait la main à son adversaire pour le premier tour de jeu. Ceci est fait au moyen d'une requête SQL de type insert.

Sur la page de jeu, un algorithme est donc inclus à l'aide l'instruction PHP include, et il permet de définir l'utilisateur à qui est le tour de jouer. Pour cela, il récupère la ligne correspondant à la partie en cours dans la table bataille_aqui et teste la valeur de la colonne « idJoueur ». Si cette valeur est égale à l'utilisateur courant, alors un message signifiant que

[7] Le code correspondant à la page hjouer.php est disponible en annexe 7.

c’est à lui de jouer apparait, ainsi que la grille de tir, sinon, un message signifiant que ce n’est pas à lui de jouer apparait. Ceci est exécuté à chaque tour de jeu et, lorsqu’un utilisateur joue, la valeur du champ concerné dans la table bataille_aqui est bien sur modifié afin de passer la main à l’adversaire !

De plus, lorsqu’un utilisateur choisit de tirer dans une case donnée du jeu de l’adversaire, certaines vérifications sont faites notamment au niveau du contenu de cette case. En effet, si cette case ne contient pas de partie de bateau, alors il faut ajouter un enregistrement dans la table bataille_tirs pour l’utilisateur concerné, sinon, il faut, en plus, modifier la valeur de l’attribut « touche » de cette case contenue dans la table bataille_positions.

Enfin, à chaque chargement de la page, un script externe est exécuté afin de déterminer si l’utilisateur courant a gagné ou perdu. Pour cela, on vérifie que toutes les cases occupées par ses bateaux dans la table bataille_positions possèdent un attribut « touche » dont la valeur vaut 1. Si c’est le cas, alors on affiche un message lui signifiant qu’il a perdu (‘l’adversaire a coulé tous ses bateaux). Par contre, on vérifie également l’inverse, c'est-à-dire que tous les bateaux de l’adversaire sont coulés ou non. Si c’est le cas, alors on affiche que l’utilisateur courant a gagné ! Si aucun de ces cas de figure n’est détecté, alors on ne fait rien et on passe au tour de jeu suivant.

Maintenant, le jeu est terminé puisque l’on peut :

- Choisir un adversaire,
- Choisir le placement de ses bateaux,
- Jouer,
- Gagner ou perdre.

Il faut simplement remarquer que, lorsqu’un utilisateur gagne, son nombre de victoire est incrémenté de 1 grâce à une requête SQL de type update. Si il permet, son nombre de défaite est également augmenté de 1 ! L’affichage de la page d’accueil sera donc mis à jour conformément au nombre de victoires ou de défaites réels.

4.3 Les erreurs et problèmes rencontrés

Durant toute la durée de mon projet, j’ai eu des choix à faire, des erreurs à corriger et des problèmes à régler. Ces problèmes et erreurs ont puent être de toutes natures, de la simple faute de frappe à l’erreur de logique. Il m’a donc été nécessaire réfléchir, de chercher comment résoudre mes problèmes en tous genres. Dans cette partie, je vais présenter les principales choses qui m’on demandé réflexion (en dehors des choix que j’ai présenté plus haut).

4.3.1 Communication entre utilisateurs

Au début de la réalisation de ce projet, j’ai décidé de donner la possibilité aux différents utilisateurs de communiquer entre eux, notamment afin de pouvoir choisir agréablement l’utilisateur avec qui ils allaient jouer. Pour cela, je n’avais pas d’idées de la manière dont j’allais procéder, j’ai donc cherché sur internet. Après quelques recherches, j’ai trouvé différentes solutions :

- Utiliser les sockets,
- Utiliser un fichier texte,

- Utiliser une base de données spéciale stockant l'état du message.

J'ai opté après différents tests pour la dernière méthode puisque que je l'ai trouvé plus simple d'utilisation. J'ai donc créé une table dans ma base de données, la table bataille_messages dans laquelle j'ai stocké tous les messages que je voulais envoyer d'un utilisateur à un autre. Il ne me restait donc plus qu'à intégrer une portion de code permettant de vérifier l'état des messages reçus ou envoyés par un utilisateur afin de savoir si je devais en afficher un ou non. Ceci m'a donc permis d'effectuer l'échange de messages comme j'en avais envie et comme je l'avais imaginé. Je pense que cela aurait certainement été « plus propre » en utilisant les sockets seulement, je n'ai pas trouvé comment adapter ce système afin d'avoir un échange direct entre les deux utilisateurs et non plus :

- Entre un utilisateur et le serveur puis
- Entre le serveur et l'autre utilisateur.

Je pense donc que la solution choisie était la « meilleur » en l'état actuel de mes compétences.

4.3.2 Vérification du placement des bateaux

Cette partie ne concerne pas réellement une erreur que j'ai rencontrée mais surtout un point de réflexion très important. En effet, en fonction de la manière dont j'ai imaginé le placement de mes bateaux, je me suis, il est vrai, créé quelques problèmes moi-même au départ. De fait, je n'avais pas pensé à désactiver les cases déjà utilisées par d'autres bateaux et j'ai donc du chercher afin de vérifier que chaque case de mon bateau n'était pas déjà enregistrée dans la table bataille_positions pour la partie et l'utilisateur en cours.

Lorsque j'ai pensé à ce système de désactivation des cases, je me suis réellement « enlevé une belle épine du pied » malgré que j'avais déjà réglé le problème au moyen de test en PHP… Je ne l'ai donc mis en place que dans un soucis de clarté pour l'utilisateur!

Conclusion

Au cours de la réalisation de ce troisième et avant dernier projet de Web Dynamique 2, je pense que j'ai pu explorer un maximum les possibilités offertes par PHP et MySQL au niveau de la réalisation de sites de tous types (vente, jeux,…). Ce projet en particulier m'a permis d'utiliser de manière très approfondie la base de données MySQL et notamment de revoir avec précision les notions de modélisation de base de données. En effet, il était impossible dans ce projet, de ne pas faire une base de données cohérente et claire puisque les tables contenaient trop de données.

J'ai donc pu investir les connaissances acquises dans deux matières différentes afin de réaliser un site qui soit le plus professionnel et le mieux réalisé possible Après ce projet, je commence à apercevoir ce à quoi peut ressembler la programmation d'un site commercial (destiné à être vendu) en PHP. En effet, il m'a fallu ici mettre en place un système de résonnement et une logique proche de celle que j'aurais eu si je devais réaliser ce projet dans le cadre d'un stage ou d'une mission d'entreprise puisque le site à réaliser devait être le plus propre et le plus complet possible.

Après la fin de mon projet, je pense qu'il reste tout de même quelques améliorations à apporter afin que le site permette de jouer de manière grand public à la bataille navale. En effet, je pense que, notamment au point de vue des graphismes et des tests, ce que j'ai fait est améliorable. Je me suis en effet rendu compte à la fin de la réalisation de ce projet que j'avais par moment des répétitions de codes qui seraient à éviter dans l'objectif de faire un site le plus compétitif et rapide possible.

Tout de même, à la fin de ce projet, j'ai réalisé mon objectif principal qui était de réaliser un site permettant de jouer en réseau à la bataille navale en mettant en place un système de communication réel entre les utilisateurs afin qu'ils puissent se proposer des parties entre eux. J'ai réussi à faire reposer ce système sur la base de données MySQL mise en place pour l'occasion, ce qui faisait partie de mon objectif principal.

Annexes

Annexe 1 : la page accueil.php, l'arrivée sur le site

Code

```
<?php
session_start();

?>
<html>
<head>
<title>Bataille Navale</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="../";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>
</td>
<td class="corps">
<center>
<h2>Bataille Navale</h2>
<?php
//Affichage des produits en vente

if($_SERVER["HTTP_REFERER"] =="http://galerneau.pierre.free.fr/Fiche 18/deco.php"){
        echo "<div class=erreur>Vous avez été déconnecté de votre compte.</div><br>";
}

if(!isset($_SESSION['login'])){
        ?>
        Pour jouer, connectez vous :<br><br>
        <div style="{border-width:1px; border-color:red; border-style:outset;}">
        <table class="table">
        <form method=post action=connect.php>
        <tr><td>Login :</td><td><input type=text name=login size=10></td></tr>
        <tr><td>Mot de passe :</td><td><input type=password name=mdp size=10></td></tr>
        <tr><td></td><td><input type=submit value="Connexion"></td></tr>
        </form>
        </table>
        <a href=creer_compte.php>Creer un compte</a></div>
        <?php
}else{
        include("menu.php");
        include("verifMess.php");
        echo "<br><br>";
        $requete="select bataille_users.idUser, bataille_users.login, bataille_users.nbVictoire as nbVict from
bataille_usersConnectes, bataille_users where bataille_users.idUser=bataille_usersConnectes.idUser;";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
```

```
                        echo "<div><span>";
                        if(mysql_num_rows($reponse)==1){
                                echo "<br>Il n'y a aucun utilisateur connecté a part vous !";
                        }else{
                                //Liste des utilisateurs connectés
                                echo "<br>Vos adversaires potentiels :";
                                echo "<table border=1 class=table1><tr><td>Utilisateur :
</td><td>Nombre de victoires :</td><td>Proposer une partie :</td></tr>";
                                echo "<form method=post action=choixAdv.php>";
                                while($donnees = mysql_fetch_array($reponse)){
                                        if($donnees['idUser']==$_SESSION['id']){
                                        }else{
                                                echo
"<tr><td>".$donnees['login']."</td><td>".$donnees["nbVict"]."</yd><td><input type=radio name=partie
value=".$donnees['idUser']."></td></tr><br>";
                                        }
                                }
                                echo "<tr><td></td><td></td><td><input type=submit
value=Choisir></form></td></tr></table>";
                        }

                }else{
                }
        }else{
                echo "Echec de la connexion.";
        }
        echo "</span>";
        $requete="select * from bataille_partie where (idJ1=".$_SESSION['id']." or
idJ2=".$_SESSION['id'].") and idGagnant=0;";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
                        echo "<div><span>Vos parties en cours contre :<hr>";
                        if(mysql_num_rows($reponse)==0){
                                echo "Vous n'avez pas de partie en cours.";
                        }else{
                                $requete2="select * from bataille_users where
(idUser='".mysql_result($reponse, 0, 1)."' or idUser='".mysql_result($reponse, 0, 2)."');";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($requete2)){
                                                echo "<table class=table1
><tr><td></td><td></td></tr><tr><td></td><td></td></tr>";
                                                while($donnees=mysql_fetch_array($reponse)){
                                                        if($_SESSION['id']==$donnees['idUser']) {

                                                        }else{
                                                                echo
"<tr><td>".$donnees['login']."</td><td><a href=\"jouer.php\"><font
color=red>Reprendre</font></a></td></tr>";
                                                        }
                                                }
                                                echo "</table>";
                                        }else{
                                                echo mysql_error();
                                        }
                                }
                        }
                }
                echo "</span></div>";
```

```
        }

        echo "<meta http-equiv='refresh' content='60';URL=accueil.php?refresh=60'>";
}?>
<span><img src="../Images/bataille-navale-logo.jpg" alt="Bataille Navale" height="170px"></span></div>
</td>
</table>
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```

Annexe 2 : création d'un compte utilisateur

Page creer_compte.php :

Code

```
<html>
<head>
<title>Bataille Navale - Créer un compte</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="../";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>
</td>
<td class="corps" float=center>
<center align=center>
<h2>Bataille Navale</h2>
<h3>Créer un compte</h3>
<?php
if($_SERVER["HTTP_REFERER"] =="http://localhost/WD1/fiche%2017/creer_compte.php"){
        echo "<div class=erreur>Ce login existe déja !</div><br>";
}
?>

<form method=post action=creer.php>
<table class=table>
<tr><td>Nom : </td><td><input typ=text name=nom>*</td></tr>
<tr><td>Preom : </td><td><input typ=text name=prenom>*</td></tr>
<tr><td>Login : </td><td><input typ=text name=login>*</td></tr>
<tr><td>Mot de passe : </td><td><input type=password name=pwd1>*</td></tr>
<tr><td>Confirmation : </td><td><input type=password name=pwd2>*</td></tr>
<tr><td></td><td><input type=submit Value=Enregistrer></td></tr>
</table>
</form>
NB : (*) signifie que le champ correspondant est obligatoire !
</td>
</table>
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```

Page creer.php :

Code

```
<html>
<head>
<title>Bataille Navale - Créer un compte</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="../";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">

<?php
include("$racine/menu.php");
?>
</td>
<td class="corps" float=center>
<center align=center>
<h2>Bataille Navale</h2>
<h3>Créer un compte</h3>
<?php
$nom=$_POST["nom"];
$prenom=$_POST["prenom"];
$login=$_POST["login"];
$pwd1=$_POST["pwd1"];
$pwd2=$_POST["pwd2"];

//Vertification de la validité des données saisient
if($nom=="" || !is_string($nom) || $prenom=="" || !is_string($nom) || $login=="" || $pwd1!=$pwd2 || strlen($pwd1)<6){
        echo "<div class=erreur align=left>Les données ne sont pas valides : elles doivent etre du type suivant : <br>
        <ul>
        <li>Nom : chaine de caractere</li>
        <li>Prenom : chaine de caractere</li>
        <li>Login : chaine de caractères mixte (chiffres et lettres)</li>
        <li>Mot de passe : chaine de caracteres mixte (chiffres et lettres) de 6 caractères minimum</li>
        </ul>
        </div>";
}else{
        //Insertion dans la base de données users
        $requeteV="select login from bataille_users where login='$login';";
        $requete="insert into bataille_users values('', '$nom', '$prenom', '$login', '$pwd1', 0, 0, 0);";
        $base=mysql_connect("sql.free.fr", "galerneau_pierre", "zaXXDXKy");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requeteV)){
                        if(mysql_num_rows($reponse)>0){
                                echo "<div class=erreur>Ce login existe deja !</div><br><br><a href=creer_compte.php>Retour</a>";
                        }else{
                                if($reponse = mysql_query($requete)){
                                        echo "Votre compte a été créé, connectez vous :";
                                        ?>
                                        <div style="{border-width:1px; border-color:red; border-style:outset;}"><form method=post action=connect.php>
```

```
                                        <table class=table>
                                        <tr><td>Login :</td><td><input type=text name=login size=10></td></tr>
                                        <tr><td>Mot de passe :</td><td><input type=password name=mdp size=10></td></tr>
                                        <tr><td></td><td><input type=submit value="Connexion"></td></tr>
                                        </table>
                                        <a href=creer_compte.php>Creer un compte</a>
                                        </form>
                                        </div>
                                        <?php
                                }else{
                                        echo "Il y a eu une erreur lors de la création, réessayez.".mysql_error();
                                }
                        }
                }
        }else{
                echo "Erreur de connexion au serveur, réessayez plus tard.";
        }
        mysql_close();
}
?>

</td>
</table>
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```

Annexe 3 : connexion d'un utilisateur : la page connect.php

Code

```
<?php
session_start();
?>
<html>
<head>
<title>Bataille Navale - Connexion</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="../";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>
</td>
<td class="corps" float=center>
<center align=center>
<h2>Bataille Navale</h2>
<h3>Bienvenue</h3>
<?php
include("menu.php");
echo "<br><br>";
$login=$_POST["login"];
$pwd=$_POST["mdp"];

$requete="select * from bataille_users where Login='$login';";
$base=mysql_connect("localhost", "root", "");
if(mysql_select_db("galerneau_pierre", $base)){
        if($reponse = mysql_query($requete)){
                if(mysql_num_rows($reponse)==0){
                        echo "Ce login n'éxiste pas !";
                }else{
                        $reponse = mysql_fetch_array($reponse);
                        if($reponse['motDePasse']==$pwd){
                                //Ouverture de session
                                $_SESSION['id']=$reponse['idUser'];
                                $_SESSION['login']=$login;
                                $requete1="select count(*) from bataille_usersConnectes where idUser='".$reponse['idUser']."';";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse1 = mysql_query($requete1)){
                                                if(mysql_result($reponse1, 0)==0){
                                                        $requete2="insert into bataille_usersConnectes values('', ".$reponse['idUser'].");";
                                                        $base=mysql_connect("localhost", "root", "");
                                                        if(mysql_select_db("galerneau_pierre", $base)){
```

```
                                                    if($reponse = mysql_query($requete2)){

                                                    }
                                                }

                                            }


                                        }
                                    }
                                    echo "Vous êtes connecté sous le pseudonyme <b>".$_SESSION['login']."</b>.<br><br>";

                                    include("verifMess.php");
                                    include("verifAdv.php");
                                    echo "</span>";
                                    mysql_close();
                                    ?>
                                    <script language=javaScript>
                                    document.location.replace("accueil.php");
                                    </script>
                                    <?php
                                }else{
                                    echo "Mot de passe erroné !";
                                }
                            }
                        }
                    }
                    ?>

                    </td>
                    </table>
                    <table height="10%"><center>
                    <td class="signe">
                    <?php
                    include("$racine/bas.php");
                    ?>
                    </td></center>
                    </table>
                    <!--C'est fini !-->
                    </body>
                    </html>
```

Annexe 4 : envoi d'un message : la page choixAdv.php

Code

```
<?php
session_start();
?><html>
<head>
<title>Bataille Navale - Choix de l'adversaire</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="..";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>

</td>
<td class="corps"><center>
<h2 align=center>Bataille Navale</h2>
<h3>Choix de votre adversaire</h3>
<?php
include("menu.php");
include("verifMess.php");
$adv=$_POST["partie"];
$bl=true;
//Si on a deja envoye une invitation à cet adv
$requete="select count(*) from bataille_messages where bataille_messages.idUserSource='".$_SESSION['id']."' and bataille_messages.idUserDestination=$adv;";
$base=mysql_connect("localhost", "root", "");
if(mysql_select_db("galerneau_pierre", $base)){
        if($reponse = mysql_query($requete)){
                if(mysql_result($reponse, 0 ,0)!=0){
                        $bl=false;
                }
        }
}


if($adv==""){
        echo "<br><br>Vous n'avez choisi aucun adversaire, retournez en choisir un sur la page d'accueil du site !";
}else{
        if($bl){
                $requete="select * from bataille_users where idUser=".$adv.";";
                $base=mysql_connect("localhost", "root", "");
                if(mysql_select_db("galerneau_pierre", $base)){
                        if($reponse = mysql_query($requete)){
                                while($donnees = mysql_fetch_array($reponse)){
                                        echo "<br><br>Vous avez choisi de jouer avec
```

```
<b>".$donnees['login']."</b>.";
                                        }
                                                //Enregistrement de l'utilisateur à qui envoyer le message
                                                $requete="insert into bataille_messages values('
'".$_SESSION['id']."','$adv', 0);";
                                                $base=mysql_connect("localhost", "root", "");
                                                if(mysql_select_db("galerneau_pierre", $base)){
                                                        if($reponse = mysql_query($requete)){
                                                                echo "<br><br>Une invitation va lui etre
envoyée.";
                                                        }
                                                }
                                                mysql_close();

                                        }
                                }
                        }else{
                                echo "<br><br>Une invitation identique est deja envoyée !";
                        }
}

echo "<meta http-equiv='refresh' content='60';URL=choixAdv.php?refresh=60'>";

?>

</td>
</table>
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```

Annexe 5 : transmission des messages : la page verifMess.php

Code

```
<?php

echo <<< Verification

Verification;
$b=true;
$choix=$_GET['a'];
if($choix==""){
	//Envoi de l'invitation :
	$requete="select bataille_users.idUser as id, bataille_users.login from bataille_users, bataille_messages where bataille_messages.idUserDestination='".$_SESSION['id']."' and bataille_users.idUser=bataille_messages.idUserSource and bataille_messages.lu=0";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			while($donnees = mysql_fetch_array($reponse)){
				$_SESSION['adv']=$donnees['id'];
				echo "<br><font size=3 color=red>Une invitation vous a été envoyée par : ".$donnees['login']."</font>";
				echo "<br><a href=accueil.php?a=ok>Accepter</a> <a href=accueil.php?a=non>Refuser</a>";
				//$id=$donnees['id'];
			}
		}else{
			echo mysql_error();
		}
	}else{
		echo mysql_error();
	}
}else if($choix=="ok"){
	//Accepter ou refuser l'invitation
	echo "<br><br>";
	$requete="update bataille_messages set lu=1 where idUserDestination='".$_SESSION['id']."';";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			echo "Invitation acceptée.<br><br><a href=\"placer.php\"><h3 align=center>Commencez la partie !</h3></a>";
			//Creation d'une partie
			$requete="select count(*) from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and (idJ1=".$_SESSION['adv']." or idJ2=".$_SESSION['adv'].") and idGagnant=0;";
			$base=mysql_connect("localhost", "root", "");
			if(mysql_select_db("galerneau_pierre", $base)){
				if($reponse = mysql_query($requete)){
					if(mysql_result($reponse, 0, 0)>0){
						$b=false;
					}else{
						if($_SESSION['adv'] == $_SESSION['id']){
							echo "Impossible de jouer contre vous meme !";
						}else{
							$req="insert into bataille_partie values(",
```

```
".$_SESSION['adv']."; ".$_SESSION['id']."',0);";
                                                        $base=mysql_connect("localhost", "root", "");
                                                        if(mysql_select_db("galerneau_pierre",  $base)){

                                                                if($reponse = mysql_query($req)){
                                                                        $re="select id from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and (idJ1=".$_SESSION['adv']." or idJ2=".$_SESSION['adv'].") and idGagnant=0;";

        $base=mysql_connect("localhost", "root", "");

        if(mysql_select_db("galerneau_pierre", $base)){
                                                                                if($reponse = mysql_query($re)){
                                                                                        echo "Partie créée !";

        $_SESSION['partie']=mysql_result($reponse,0 ,0);
                                                                                }
                                                                        }
                                                                }
                                                        }

                                                                }

                                                        }
                                                }else{
                                                        echo mysql_error();
                                                }
                                        }
                                }
        }
}else if($choix=="non"){
        $requete="update bataille_messages set lu=-1 where idUserDestination='".$_SESSION['id']."';";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
                        echo "Invitation refusée !";
                }
        }
}
//Prevenir l'envoyeur du resultat
$requete="select bataille_messages.idUserSource from bataille_users, bataille_messages where bataille_messages.idUserSource='".$_SESSION['id']."' and bataille_messages.lu=1;";
$base=mysql_connect("localhost", "root", "");
if(mysql_select_db("galerneau_pierre", $base)){
        if($reponse = mysql_query($requete)){
                if(mysql_num_rows($reponse)>0){
                        if(mysql_result($reponse, 0, 0)==$_SESSION['id']){
                                echo "<br><br>Votre invitation a été acceptée !<br><br><a href=\"placer.php\"><h3>Commencez la partie !</h3></a>";
                                //Creation d'une partie
                                $req="select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and (idJ1=".$_SESSION['adv']." or idJ2=".$_SESSION['adv']." and idGagnant=0);";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($req)){
                                                $re="select max(id) from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;";
```

```
                                        $base=mysql_connect("localhost", "root", "");
                                        if(mysql_select_db("galerneau_pierre", $base)){
                                                if($reponse = mysql_query($req)){
                                                        echo "<br><br>Votre partie a été créée !";

        $_SESSION['partie']=mysql_result($reponse,0 ,0);
                                                }
                                        }
                                }
                        }

                        /*      $requete="select count(*) from bataille_partie where idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id']." and  idJ1=$adv or idJ2=$adv and idGagnant=0;";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($requete)){
                                                if(mysql_result($reponse, 0, 0)==1){
                                                        $_SESSION['partie']=mysql_result($reponse,  0, 0);
                                                        echo "Partie : ".$_SESSION['partie'];
                                                }
                                        }else{
                                                echo mysql_error();
                                        }
                                }               */
                                if($b){
                                // $requeteP="insert into bataille_partie values (", ".$_SESSION['id']., $adv, 0);";

                        /*              $base=mysql_connect("localhost", "root", "");
                                        if($adv == $_SESSION['id']){
                                                echo "Impossible de jouer contre vous meme !";
                                        }else{
                                                $req="insert into bataille_partie values(", $adv, ".$_SESSION['id'].",0);";
                                                $base=mysql_connect("localhost", "root", "");
                                                if(mysql_select_db("galerneau_pierre",  $base)){

                                                        if($reponse = mysql_query($req)){
                                                                echo "Partie créée !";
                                                                $re="select max(id) from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;";

        $base=mysql_connect("localhost", "root", "");

        if(mysql_select_db("galerneau_pierre", $base)){
                                                                                        if($reponse = mysql_query($req)){

        $_SESSION['partie']=mysql_result($reponse,0 ,0);
                                                                                        }
                                                                                }
                                                                        }
                                                                }

                                                        }               */
                                }else{
                                        echo "Vous jouez deja une partie contre cet adversaire !";
                                }
```

```
                        }
                    }
                }
            }
            $requete="select bataille_messages.idUserSource from bataille_users, bataille_messages where bataille_messages.idUserSource='".$_SESSION['id']."' and bataille_messages.lu=-1;";
            $base=mysql_connect("localhost", "root", "");
            if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
                    if(mysql_num_rows($reponse)>0){
                        if(mysql_result($reponse, 0, 0)==$_SESSION['id']){
                            echo "Votre invitation a été refusée !";
                        }
                    }
                }
            }



            mysql_close;
```

Annexe 6 : placement des bateaux, le fichier placer.php

Code

```
<?php
session_start();
?><html>
<head>
<title>Bataille Navale - Placement de vos bateaux</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="..";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>

</td>
<td class="corps"><center>
<h2 align=center>Bataille Navale</h2>
<h3>Placement de vos bateaux</h3>
<?php

function placeBateau($nom, $taille, $nb, $partie){
	//Recuperation des coordonnées
	for($i=1; $i<11; $i++){
		for($j=1; $j<11; $j++){
			$val=$_POST["bateau$i$j"];
			if("$i$j"=="$val"){
				$tab[$i][$j]=1;
			}else{
				$tab[$i][$j]=0;
			}
		}
	}
	//Verification des Placements
	$bool=false;
	$bienP=false;
	$i=1;
	$cpt=0;
	$bool2=false;

	//Les coordonnées ne sont pas deja utilisées ?
	//Recuperation du contenu de la base bataille_positions
	$requete="select * from bataille_positions where idJoueur=".$_SESSION['id']." and idPartie=".$partie.";";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse=mysql_query($requete)){
			$i=0;
```

```
                while($donnees=mysql_fetch_array($reponse)){
                        $tableau[$i]=$donnees['coordX'].$donnees['coordY'];
                        $i++;
                }
        }
}
if( compteNbBateau($nom, $taille, $partie)>=$nb){
        return false;
}

//Verification que les coordonnées saisies ne sont pas deja dans la base
if(isset($tableau)){
        $i=1;
        $nbDejaUt=0;
        while($i<11){
                $j=1;
                while($j<11){
                        if(in_array("$i$j", $tableau)){
                                $tab[$i][$j]=0;
                        }
                        $j++;
                }
                $i++;
        }
}
$i=1;
while($i<11 && !$bool){
        $j=1;
        while($j<11 && !$bool){
                if($tab[$i][$j]==1){
                        $k=0;
                        $cpt=0;
                        //Horizontal
                        while($k<$taille && !$bool2){
                                if($tab[$i][$j+$k]==1){
                                        $enreg[$i][$j+$k]=1;
                                        $cpt++;
                                }else{
                                        $bool2=true;
                                }
                                if($cpt==$taille){
                                        $bool=true;
                                        $bienP=true;
                                }
                                $k++;
                        }
                        //Vertical
                        $cpt=0;
                        $k=0;
                        while($k<$taille && $bool2){
                                if($tab[$i+$k][$j]==1){
                                        $enreg[$i+$k][$j]=1;
                                        $cpt++;
                                }else{
                                        $k=$taille;
                                }
                                if($cpt==$taille){
                                        $bool=true;
                                        $bienP=true;
                                }
```

```
                                        $k++;
                                    }
                                }
                                $j++;
                            }
                            $i++;
                        }
                        if($nbDejaUt>0 || !$bienP || compteNbBateau($nom, $taille, $partie)>$nb+1){
                            return false;
                        }else{

                            //Enregistrement dans la BDD bataille_positions
                            for($i=1; $i<11; $i++){
                                for($j=1; $j<11; $j++){
                                    if($enreg[$i][$j]==1){
                                        $requete="insert into bataille_positions values('', '$nom', $i, $j, 0,
'".$_SESSION['id']."', ".$partie.");";
                                        $base=mysql_connect("localhost", "root", "");
                                        if(mysql_select_db("galerneau_pierre", $base)){
                                            if($reponse = mysql_query($requete)){
                                            }else{
                                                echo mysql_error();
                                            }
                                        }else{
                                            echo mysql_error();
                                        }
                                        mysql_close();

                                    }
                                }
                            }
                            return true;
                        }
                    }

                    function affChoix($bat){
                        //Recuperation du contenu de la base bataille_positions
                        $requete="select * from bataille_positions where idJoueur=".$_SESSION['id']." and
                    idPartie=".$_SESSION['partie'].";";
                        $base=mysql_connect("localhost", "root", "");
                        if(mysql_select_db("galerneau_pierre", $base)){
                            if($reponse=mysql_query($requete)){
                                $i=0;
                                while($donnees=mysql_fetch_array($reponse)){
                                    $tableau[$i]=$donnees['coordX'].$donnees['coordY'];
                                    $i++;
                                }
                            }else{
                                echo mysql_error();
                            }
                        }
                        $table= "<center><div><form method=post action=placer.php?v=ok><center><table class=table1
                    border=0>";
                        for($i=0; $i<11; $i++){
                            $table.= "<tr>";
                            for($j=0; $j<11; $j++){
                                if($i==0 && $j>0){
                                    $table.= "<td align=center>".chr(ord('A') + $j-1)."</td>";
                                }else if($j==0 && $i>0){
                                    $table.= "<td>".chr(ord('0') + $i-1)."</td>";
```

```
                        }else if($i>0 && $j>0){
                                if(isset($tableau)){
                                        if(in_array("$i$j", $tableau)){
                                                $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\" checked=true disabled=false></td>";
                                        }else{
                                                $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
                                        }
                                }else{
                                        $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
                                }


                        }else{
                                $table.= "<td></td>";
                        }
                }
                $table.= "</tr>";
        }
        $table.= "<input type=hidden name=bat value=$bat><td><input type=submit value=Verifier></td></table></form></div></center>";
        return $table;
}

function affGrille($partie){
        //Affichage grille
        $tableau;
        $requete="select * from bataille_positions where idJoueur=".$_SESSION['id']." and idPartie=".$partie.";";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
                        $i=0;
                        while($donnees = mysql_fetch_array($reponse)){
                                $tableau[$i]=$donnees['coordX'].$donnees['coordY'];
                                $i++;
                        }
                }
        }
        mysql_close();
        echo "<table class=table1>";
        for($i=1; $i<11; $i++){
                echo "<tr>";
                for($j=1; $j<11; $j++){
                        if(count($tableau)>0){
                                if(in_array("$i$j", $tableau)){
                                        echo "<td><img src=../Images/mer.jpg height=\"30px\"></td>";

                                }else{
                                        echo "<td><img src=../Images/vide.gif height=\"30px\"></td>";
                                }
                        }else{
                                echo "<td><img src=../Images/vide.gif height=\"30px\"></td>";
                        }
                }
                echo "</tr>";
        }
        echo "</table>";
```

```
}

function compteNbBateau($nom, $taille, $partie){
	$requete="select * from bataille_positions where nom='".$nom."' and idJoueur=".$_SESSION['id']." and idPartie=".$partie.";";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse=mysql_query($requete)){
			return mysql_num_rows($reponse)/$taille;
		}else{
		//	echo mysql_error();
		}
	}else{
		echo mysql_error();
	}
	mysql_close();
}


$requete="select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;";
$base=mysql_connect("localhost", "root", "");
if(mysql_select_db("galerneau_pierre", $base)){
	if($reponse = mysql_query($requete)){
		$_SESSION['partie']=mysql_result($reponse, 0, 0);
		$partie=mysql_result($reponse, 0, 0);

	}
}

include("menu.php");
//include("verifMess.php");
echo "<br><br>";

$creer=$_POST["bat"];
if($creer==""){
?>

<div>
<span1>
<h4 align=center>Bateaux à placer :</h4>
<table class=table1>
<tr><td>1 porte avion (5 cases)</td></tr>
<tr><td>1 croiseur (4 cases)</td></tr>
<tr><td>1 contre-torpilleur (3 cases)</td></tr>
<tr><td>1 sous marin (3 cases)</td></tr>
<tr><td>2 torpilleurs (2 cases)</td></td></tr></form>
</table>
</span1></div>
<div>
<span1>
Quel bateau voulez vous placer ?
<hr>
<table class=table1>
<form method=post action=placer.php>
<?php
if(compteNbBateau("Porte Avion", 5, $partie)<1){
?>
<tr><td>Porte avion</td><td><input type=radio name=bat value=pa></td></tr>
<?php
```

```
}
if(compteNbBateau("Croiseur", 4, $partie)<1){
?>
<tr><td>Croiseur</td><td><input type=radio name=bat value=c1></td></tr>
<?php
}
if(compteNbBateau("Contre Torpilleur", 3, $partie)<1){
?>
<tr><td>Contre torpilleur</td><td><input type=radio name=bat value=ct1></td></tr>
<?php
}
if(compteNbBateau("Sous Marin", 3, $partie)<1){
?>
<tr><td>Sous marin</td><td><input type=radio name=bat value=sm1></td></tr>
<?php
}
if(compteNbBateau("Torpilleur", 2, $partie)<2){
?>
<tr><td>Torpilleur</td><td><input type=radio name=bat value=t1></td></tr>
<?php
}
?>
<tr><td></td><td><input type=submit value="Placer"></td></tr>
</form>
<tr><td></td><td></td></tr>
<tr><td></td><td><a href="placer.php?v=aff">Afficher la grille</a></td></tr>
</table>
</span1></div>
<?php
echo "<h3 align=center>Votre grille de jeu :</h3>";
affGrille($partie);

}else if($creer=="pa"){
        echo "<h3 align=center>Placer le porte avion :</h3>";
        echo affChoix($creer);
        $verif=$_GET["v"];
        if($verif="ok"){
                $test=placeBateau("Porte Avion", 5, 1, $partie);
                if($test){
                        echo "Votre porte avion est bien placé !<br><br><a href=placer.php>Placer un
autre bateau</a>";
                                ?>
                <script language=javaScript>
                document.location.replace("placer.php");
                </script>
                <?php
                }else{
                        echo "Votre porte avion est mal placé.<br><br><a href=placer.php>Placer un
autre bateau</a>";
                }
        }
}else if($creer=="c1" || $creer=="c2"){
        if($creer=="c1"){
                echo "<h3 align=center>Placer un croiseur :</h3>";
        }

        echo affChoix($creer);
        $verif=$_GET["v"];
        if($verif="ok"){
                $test=placeBateau("Croiseur", 4, 1, $partie);
```

```
        }
        if($test){
                echo "Votre croiseur est bien placé !<br><br><a href=placer.php>Placer un autre
bateau</a>";
                ?>
                <script language=javaScript>
                document.location.replace("placer.php");
                </script>
                <?php
        }else{
                echo "Votre croiseur est mal placé.<br><br><a href=placer.php>Placer un autre
bateau</a>";
        }
}else if($creer=="ct1"){
        if($creer=="ct1"){
                echo "<h3 align=center>Placer un contre-torpilleur :</h3>";
        }

        echo affChoix($creer);
        $verif=$_GET["v"];
                $test=placeBateau("Contre Torpilleur", 3, 1, $partie);
        if($verif="ok"){
        }
        if($test){
                echo "Votre contre-torpilleur est bien placé !<br><br><a href=placer.php>Placer un autre
bateau</a>";
                        ?>
                <script language=javaScript>
                document.location.replace("placer.php");
                </script>
                <?php
        }
        else{
                echo "Votre contre-torpilleur est mal placé.<br><br><a href=placer.php>Placer un autre
bateau</a>";
        }
}else if($creer=="sm1"){
        if($creer=="sm1"){
                echo "<h3 align=center>Placer un sous marin :</h3>";
        }

        echo affChoix($creer);
        $verif=$_GET["v"];
                $test=placeBateau("Sous Marin", 3, 1, $partie);
        if($verif="ok"){
        }
        if($test){
                echo "Votre Sous Marin est bien placé !<br><br><a href=placer.php>Placer un autre
bateau</a>";
                        ?>
                <script language=javaScript>
                document.location.replace("placer.php");
                </script>
                <?php
        }else{
                echo "Votre Sous Marin est mal placé.<br><br><a href=placer.php>Placer un autre
bateau</a>";
        }
}else if($creer=="t1"){
        if($creer=="t1"){
```

```
                echo "<h3 align=center>Placer un Torpilleur :</h3>";
        }

        echo affChoix($creer);
        $verif=$_GET["v"];
        $test=placeBateau("Torpilleur", 2, 2, $partie);
        if($verif="ok"){
        }
        if($test){
                echo "Votre Torpilleur est bien placé !<br><br><a href=placer.php>Placer un autre bateau</a>";
                        ?>
                <script language=javaScript>
                document.location.replace("placer.php");
                </script>
                <?php
        }else{
                echo "Votre Torpilleur est mal placé.<br><br><a href=placer.php>Placer un autre bateau</a>";
        }
}
//
//Compter tout de contenu de la grille
$cpt=compteNbBateau("Porte Avion", 5, $partie);
$cpt+=compteNbBateau("Croiseur", 4, $partie);
$cpt+=compteNbBateau("Contre Torpilleur", 3, $partie)+compteNbBateau("Sous Marin", 3, $partie)+compteNbBateau("Torpilleur", 2, $partie);
//echo "[".$partie."]";

if($cpt==6){
        echo "Toute votre flotte est placée ! <br><br><a href=\"jouer.php\">Commencez la partie !</a>";
}
?>



</td>
</table>
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```

Annexe 7 : le jeu en lui-même, la page jouer.php

Code

```
<?php
session_start();
?><html>
<head>
<title>Bataille Navale - Jeu</title>
<link rel="stylesheet" type="text/css" href="../style.css">
</head>
<body bgcolor="#cdcdcd">
<center>
<?php
$racine="..";
include("$racine/bandeau.php");
?>
<table height=75%>
<td class="menu">
<?php
include("$racine/menu.php");
?>

</td>
<td class="corps"><center>
<h2 align=center>Bataille Navale</h2>
<h3>Jeu</h3>

<?php
function grilleTir(){
	$i=0;
	$requete="select * from bataille_tirs where idUser=".$_SESSION['id']." and idPartie=".$_SESSION['partie'].";";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			while($donnees = mysql_fetch_array($reponse)){
				$tableau[$i]=$donnees['coordX'].$donnees['coordY'];
				$i++;
			}
		}else{
			echo mysql_error();
		}
	}

	$table= "<center><div><form method=post action=jouer.php?t=1><center><table class=table1>";
	for($i=0; $i<11; $i++){
		$table.= "<tr>";
		for($j=0; $j<11; $j++){
			if($i==0 && $j>0){
				$table.= "<td align=center>".chr(ord('A') + $j-1)."</td>";
			}else if($j==0 && $i>0){
				$table.= "<td>".chr(ord('0') + $i-1)."</td>";
			}else if($i>0 && $j>0){
				if(isset($tableau)){
					if(in_array("$i$j", $tableau)){
```

```
                                        $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\" checked=true disabled=false></td>";
                                    }else{
                                        $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
                                    }
                                }else{
                                    $table.= "<td><input type=checkbox name=\"bateau$i$j\" value=\"$i$j\"></td>";
                                }
                            }else{
                                $table.= "<td></td>";
                            }
                        }
                        $table.= "</tr>";
                    }
                    $table.= "<input type=hidden name=bat value=$bat><td><input type=submit value=Tirer></td></table></form></div></center>";
                    return $table;
                }

                function grilleAdv($partie){
                    //Affichage grille
                    $tableau;
                    $requete="select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;";
                    $base=mysql_connect("localhost", "root", "");
                    if(mysql_select_db("galerneau_pierre", $base)){
                        if($reponse = mysql_query($requete)){
                            if(mysql_num_rows($reponse)==0){

                            }else{
                                if(mysql_result($reponse, 0, 1)==$_SESSION['id']){
                                    $i=0;
                                    $requete="select * from bataille_positions where idJoueur=".mysql_result($reponse, 0, 2)." and idPartie=$partie;";
                                    $base=mysql_connect("localhost", "root", "");
                                    if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($requete)){
                                            while($donnees = mysql_fetch_array($reponse)){

                    $tableauB[$i]=$donnees['coordX'].$donnees['coordY'];
                                                $i++;
                                            }
                                        }
                                    }
                                }else if(mysql_result($reponse, 0, 2)==$_SESSION['id']){
                                    $i=0;
                                    $requete="select * from bataille_positions where idJoueur=".mysql_result($reponse, 0, 1)." and idPartie=$partie;";
                                    $base=mysql_connect("localhost", "root", "");
                                    if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($requete)){
                                            while($donnees = mysql_fetch_array($reponse)){

                    $tableauB[$i]=$donnees['coordX'].$donnees['coordY'];
                                                $i++;
                                            }
                                        }
                                    }
```

```
                                        }
                                }
                        }
                }
                mysql_close();
                $requete="select * from bataille_tirs where idUser=".$_SESSION['id']." and idPartie=$partie;";
                $base=mysql_connect("localhost", "root", "");
                if(mysql_select_db("galerneau_pierre", $base)){
                        if($reponse = mysql_query($requete)){
                                $i=0;
                                while($donnees=mysql_fetch_array($reponse)){
                                        $tableau[$i]=$donnees['coordX'].$donnees['coordY'];
                                        $i++;
                                }
                        }
                }
                echo "<table class=table1>";
                for($i=1; $i<11; $i++){
                        echo "<tr>";
                        for($j=1; $j<11; $j++){
                                if(count($tableau)>0){
                                        if(in_array("$i$j", $tableau)){
                                                if(!in_array("$i$j", $tableauB)){
                                                        echo "<td><img src=../Images/plouf.gif height=\"20px\"></td>";
                                                }else{
                                                        echo "<td><img src=../Images/boum.gif height=\"20px\"></td>";
                                                }
                                        }else{
                                        //echo "$i$j<br>";
                                                echo "<td><img src=../Images/vide.gif height=\"20px\"></td>";
                                        }
                                }else{
                                        echo "<td><img src=../Images/vide.gif height=\"20px\"></td>";
                                }
                        }
                        echo "</tr>";
                }
                echo "</table>";
}


function affGrille($partie){
                //Tirs de l'adversaire
                $tableau;
                $requete="select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and id=$partie;";
                $base=mysql_connect("localhost", "root", "");
                if(mysql_select_db("galerneau_pierre", $base)){
                        if($reponse = mysql_query($requete)){
                                $partie=mysql_result($reponse, 0, 0);
                                if(mysql_result($reponse, 0, 1)==$_SESSION['id']){
                                        $i=0;
                                        $requete="select * from bataille_tirs where idUser=".mysql_result($reponse, 0, 2)." and idPartie=$partie;";
                                        $base=mysql_connect("localhost", "root", "");
                                        if(mysql_select_db("galerneau_pierre", $base)){
                                                if($reponse = mysql_query($requete)){
                                                        while($donnees = mysql_fetch_array($reponse)){
```

```
		$tableauB[$i]=$donnees['coordX'].$donnees['coordY'];
                                                $i++;
                                        }
                                }else{
                                        echo mysql_error();
                                }
                        }
                }else if(mysql_result($reponse, 0, 2)==$_SESSION['id']){
                        $i=0;
                        $requete="select * from bataille_tirs where idUser=".mysql_result($reponse, 0, 1)." and idPartie=$partie;";
                        $base=mysql_connect("localhost", "root", "");
                        if(mysql_select_db("galerneau_pierre", $base)){
                                if($reponse = mysql_query($requete)){
                                        while($donnees = mysql_fetch_array($reponse)){

		$tableauB[$i]=$donnees['coordX'].$donnees['coordY'];
                                                $i++;
                                        }
                                }else{
                                        echo mysql_error();
                                }
                        }
                }

        }
	}
	mysql_close();

	//Affichage grille
	$tableau;
	$requete="select * from bataille_positions where idJoueur=".$_SESSION['id']." and idPartie=$partie;";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			$i=0;
			while($donnees = mysql_fetch_array($reponse)){
				$tableau[$i]=$donnees['coordX'].$donnees['coordY'];
				$i++;
			}
		}
	}
	mysql_close();
	echo "<table class=table1>";
	for($i=1; $i<11; $i++){
		echo "<tr>";
		for($j=1; $j<11; $j++){
			if(count($tableau)>0){
				if(count($tableauB)>0){
					if(in_array("$i$j", $tableauB) && in_array("$i$j", $tableau)){
						echo "<td><img src=../Images/boum.gif height=\"20px\"></td>";
					}else if(in_array("$i$j", $tableau)){
						echo "<td><img src=../Images/mer.jpg height=\"20px\"></td>";
					}else if(in_array("$i$j", $tableauB)){
						echo "<td><img src=../Images/plouf.gif height=\"20px\"></td>";
```

```
                                        }else{
                                                echo "<td><img src=../Images/vide.gif height=\"20px\"></td>";
                                        }
                                }else{
                                        if(in_array("$i$j", $tableau)){
                                                echo "<td><img src=../Images/mer.jpg height=\"20px\"></td>";
                                        }else{
                                                echo "<td><img src=../Images/vide.gif height=\"20px\"></td>";
                                        }
                                }
                        }else{
                                echo "<td><img src=../Images/vide.gif height=\"20px\"></td>";
                        }
                }
                echo "</tr>";
        }
        echo "</table>";
}

function tirer($a, $b, $adv, $partie){

        $requete="insert into bataille_tirs values('', ".$_SESSION['id']."', $partie, $a, $b);";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){

                }else{
                        echo mysql_error();
                }
        }else{
                echo mysql_error();
        }
        $requete="select * from bataille_positions where coordX=$a and coordY=$b and idJoueur=$adv and idPartie=$partie;";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse = mysql_query($requete)){
                        if(mysql_num_rows($reponse)==1){
                                $requete2="update bataille_positions set touche=1 where coordX=$a and coordY=$b and idJoueur=$adv and idPartie=$partie;";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($requete2)){
                                        }else{
                                        echo mysql_error();
                                        }

                                }
                        }

                }else{
                        echo mysql_error();
                }
        }else{
                echo mysql_error();
        }
}
```

```
function auSuivant($adv,$partie){

	$requete="select * from bataille_aqui where idPartie=$partie";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			if(mysql_num_rows($reponse)==0) {
				$requete="insert into bataille_aqui values('', $partie, ".$_SESSION['id'].");";
				$base=mysql_connect("localhost", "root", "");
				if(mysql_select_db("galerneau_pierre", $base)){
					if($reponse = mysql_query($requete)){

					}else{
						echo mysql_error();
					}
				}else{
					echo mysql_error();
				}
			}else{
				$requete="update bataille_aqui set idJoueur=".$adv." where idPartie=$partie;";
				$base=mysql_connect("localhost", "root", "");
				if(mysql_select_db("galerneau_pierre", $base)){
					if($reponse = mysql_query($requete)){

					}else{
						echo mysql_error();
					}
				}else{
					echo mysql_error();
				}
			}
		}
	}
}

function refresh(){
	?>
	<script language=javaScript>
	document.location.replace("jouer.php");
	</script>

<?php
}
if(!isset($_SESSION['partie']) or !isset($_SESSION['adv'])){
	$requete="select * from bataille_partie where (idJ1=".$_SESSION['id']." or idJ2=".$_SESSION['id'].") and idGagnant=0;";
	$base=mysql_connect("localhost", "root", "");
	if(mysql_select_db("galerneau_pierre", $base)){
		if($reponse = mysql_query($requete)){
			$_SESSION['partie']=mysql_result($reponse, 0, 0);
			//Qui est l'adversaire .
			if(mysql_result($reponse, 0, 1)==$_SESSION['id']){
				$_SESSION['adv']=mysql_result($reponse, 0, 2);
			}else if(mysql_result($reponse, 0, 2)==$_SESSION['id']){
				$_SESSION['adv']=mysql_result($reponse, 0, 1);
			}else{
				echo "Erreur !";
```

```
                        }
                }else{
                        echo mysql_error();
                }
        }else{
                echo mysql_error();
        }
}

$bool=true;
include("menu.php");
//include("verifMess.php");
include("verif_conn.php");
include("verif_gagne.php");

?>
<h2 align=center>Jeu</h2>

<a href="placer.php">Placez vos bateaux (si ce n'est pas fait)</a>

<?php
if(!$bool){
        include("verif_gagne.php");
}else{
        ?>
        <div><span1>
        Votre grille :<hr>
        <?php
        affGrille($partie);
        ?>
        </div></span1>
        <div><span1>
        Grille de l'adversaire :<hr>
        <?php
        grilleAdv($partie);
        ?>
        </span1></div>
        <?php
        $partie=$_SESSION['partie'];
        $requete1="select * from bataille_aqui where idPartie=".$_SESSION['partie'].";";
        $base=mysql_connect("localhost", "root", "");
        if(mysql_select_db("galerneau_pierre", $base)){
                if($reponse8 = mysql_query($requete1)){
                        if(mysql_num_rows($reponse8)==0){
                                $req="insert into bataille_aqui values('', $partie, ".$adv.");";
                                $base=mysql_connect("localhost", "root", "");
                                if(mysql_select_db("galerneau_pierre", $base)){
                                        if($reponse = mysql_query($req)){
                                        }
                                }
                                ?>
                                <script language=javaScript>
                                document.location.replace("jouer.php");
                                </script>
                                <?php
                        }else if(mysql_result($reponse8, 0, 2)!=$_SESSION['id']){
                                echo "Ce n'est plus votre tour de jouer !";
                        }else{
                                echo "A votre tour !";
                                ?><div><span1>
```

```
                                Choix du tir :<hr>
                                <?php
                                echo grilleTir();
                                ?>
                                </div></span1>

                                <?php
                                $cpt=0;
                                $tir=$_GET['t'];
                                if($tir==1){
                                        $tirs;
                                        for($i=1; $i<11; $i++){
                                                for($j=1; $j<11; $j++){

                                                        $tirs[$i][$j]=$_POST["bateau$i$j"];

                                                                if($tirs[$i][$j]!=0){
                                                                $cpt++;
                                                                }

                                                }
                                        }
                                        if($cpt==1){
                                                for($i=1; $i<11; $i++){
                                                        for($j=1; $j<11; $j++){
                                                                if($tirs[$i][$j]!=0){
                                                                        echo "Vous avez tiré en $i$j!";
                                                                        echo "Partie :
".$_SESSION['partie'];
                                                                        echo "<br>Adversaire :
".$_SESSION['adv'];
                                                                        tirer($i, $j, $_SESSION['adv'],
$_SESSION['partie']);
                                                                        auSuivant($_SESSION['adv'],
$_SESSION['partie']);
                                                                        refresh();
                                                                }
                                                        }
                                                }
                                        }else{
                                                echo "On ne joue pas plusieurs fois par tour !";
                                        }

                                }


                        }
                }
        }
        mysql_close();
        echo "<meta http-equiv='refresh' content='30';URL=accueil.php?refresh=60'>";
}


?>



</td>
</table>
```

```
<table height="10%"><center>
<td class="signe">
<?php
include("$racine/bas.php");
?>
</td></center>
</table>
<!--C'est fini !-->
</body>
</html>
```