

Rapport de projet : Programmation Web

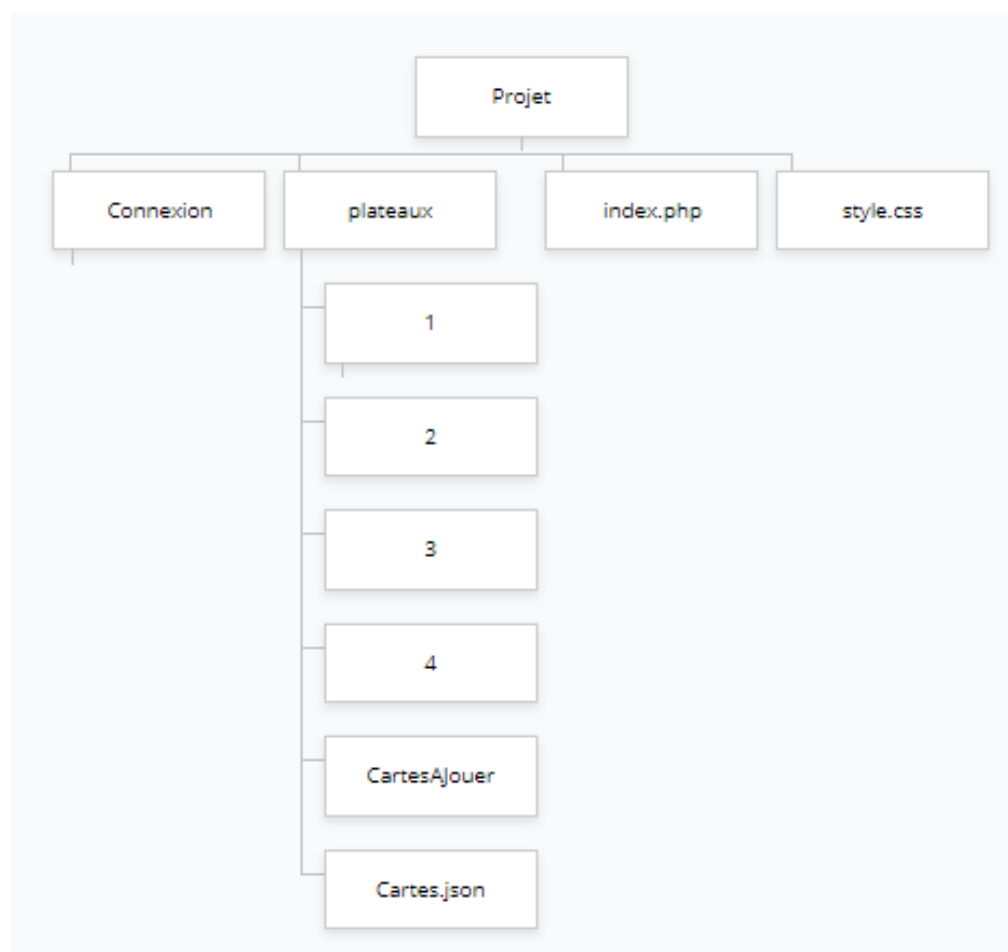
Présentation du projet

L'objectif de notre projet était de réaliser un jeu de cartes en utilisant html, css, php et javascript, nous avons décidé de coder un jeu de belote. Un des objectifs du projet était de faire en sorte que le jeu soit accessible aux nouveaux joueurs et donne la possibilité aux joueurs plus expérimentés d'expliquer aux débutants comment marche le jeu.

Pour mettre au point ce projet nous avons décidé de le séparer en 2 parties distinctes :

- La partie connexion/inscription. Cette partie prend en compte toutes les actions en rapport avec l'identification des joueurs lors de leur arrivée sur le site ainsi que la répartition de ceux-ci dans les différents plateaux.
- La partie jeu. Cette dernière correspond à l'implémentation du jeu.

Lors de la conception de notre site nous avons imaginé l'arborescence afin de ne pas être perdu lors du codage. Les dossiers 1 à 4 représentent les différents plateaux de jeu.



Nous avons aussi dressé la liste de toutes les fonctionnalités que nous souhaitons implémenté et de leurs priorités :

Fonctionnalités Prioritaires :

- chat pour la communication entre joueurs
- affichage du jeu de cartes et mise à jour de ce dernier
- déplacement des cartes (drag and drop)

Fonctionnalités moyennement prioritaires :

- Système de connexion
- Inscriptions de nouveaux membres

Fonctionnalités facultatives :

- Système de points
- Cryptages des mots de passes
- Système d'affichage de joueurs et de niveau de jeu

Partie connexion

Inscription :



La page d'inscription est un simple formulaire dans lequel on peut entrer son nom et son identifiant.

Si les champs ne sont pas remplis apparaît alors un message d'erreur indiquant à l'utilisateur que certains champs sont vides.

Ensuite une fonction crypte les mots de passe et les envoie vers le fichier password.json qui associe pour chaque personne inscrite son mot de passe crypté.

```
[{"name": "test", "mdp": "$2y$10$B5Q.Va\\dL6o10pF5DVXfV.CqRk0u1B5x.nPx79a8NXtvao42Xwye"}, {
```

Connexion :

La page de connexion ressemble beaucoup à la page inscription sauf que la redirection du formulaire se fait vers une page php qui vérifie dans le fichier password.json si un identifiant correspond à celui envoyé. Si il existe la fonction décrypte alors le mot de passe et si il est égal a celui envoyé alors cela nous redirige vers la page connexion sinon cela ramène à la page connexion.php



La partie du formulaire indiquant le niveau n'a pas encore été implémentée donc elle est pour l'instant inutile mais pourra servir à terme.

Le bouton inscription envoie vers la page inscription.php

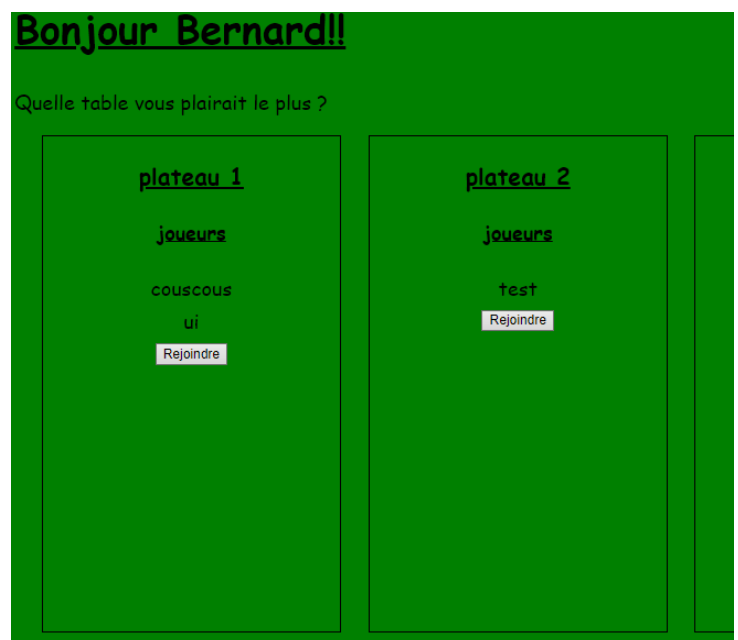
Choix du plateau de jeu :

Lorsque l'on se connecte, on arrive sur la page de choix de terrain

On y trouve des boutons dirigeant vers les 4 différents plateaux

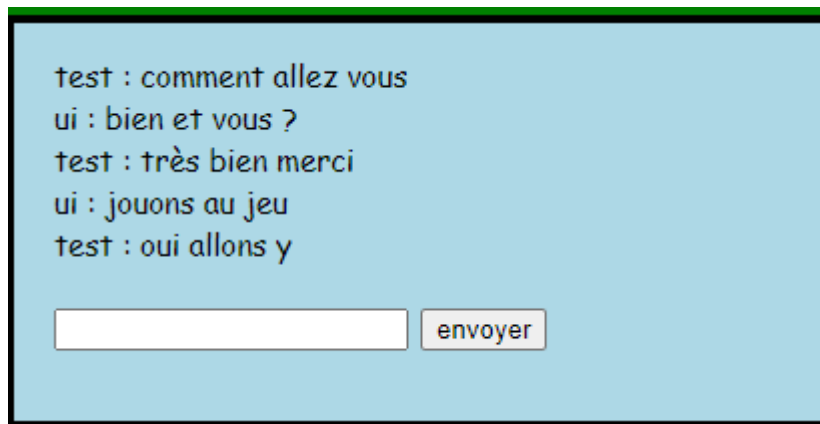
Le pseudo du joueur qui se connecte est passé à la page en utilisant la méthode request.

Depuis le fichier waiting.json on récupère la liste des joueurs dans chaque salle pour les afficher.



La page est mise à jour toutes les 5 secondes utilisant la fonction reload.location qui recharge toute la page. elle est utilisée avec la fonction setTimeout.

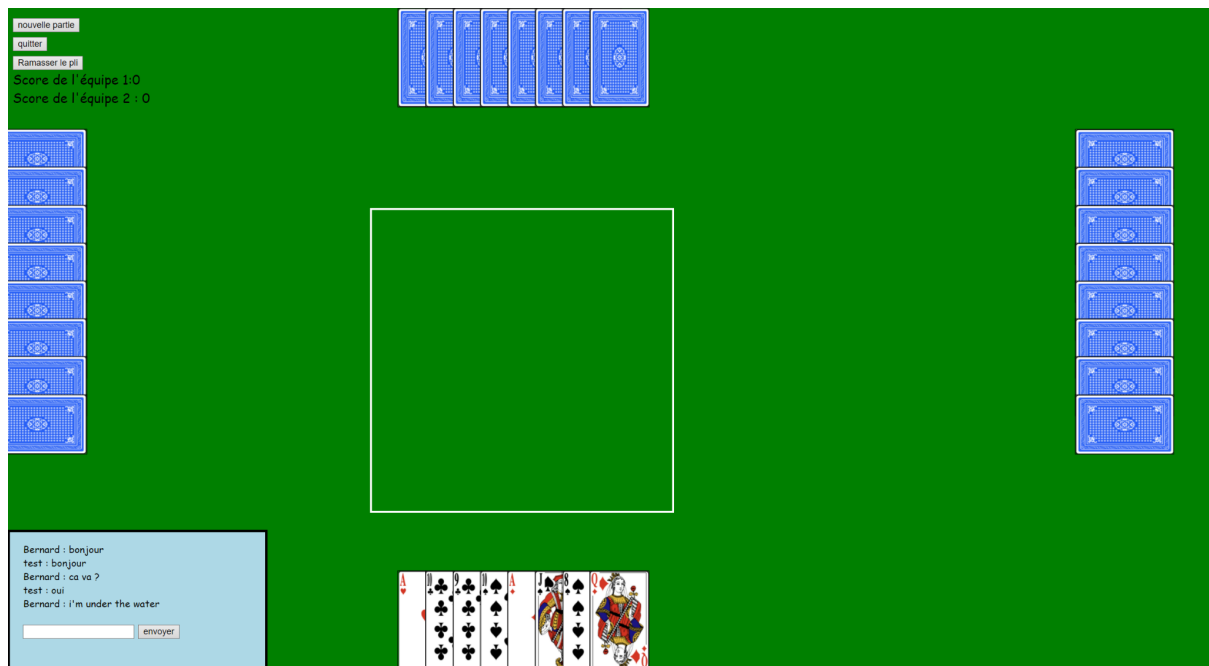
Chat :



Pour le chat se situe en bas à droite du terrain de jeu, les messages sont précédés du nom de l'expéditeur. Un maximum de 5 messages sont affichés simultanément.

Le chat est géré grâce à un fichier json qui répertorie les 5 derniers messages envoyés, puis ce fichier est mis à jour avec des requêtes AJAX.

Terrain de jeu



Le terrain de jeu est composé de plusieurs parties:

-En haut à gauche, on trouve les boutons avec des actions diverses. Le premier permettant de lancer une nouvelle partie, il permet de redistribuer les cartes aux joueurs et de remettre les scores à 0. Le second permet de quitter la salle pour revenir à l'écran de choix du plateau de jeu. Et le dernier permet de ramasser le pli, c'est-à-dire de "nettoyer" la zone de jeu et d'incrémenter les scores.

Les scores sont également notés en haut à gauche sous les boutons, ils sont gérés par un json qui sert à stocker leur valeur et donc à les actualiser.

-Au centre, on trouve le terrain de jeu, c'est ici que les cartes peuvent être déposées pour être jouer ou bien pour être montrer aux autres joueurs.

-Autour du terrain, on trouve les jeux des différents joueurs, notre jeu est bien sur à découvert uniquement pour nous alors que les autres jeux sont cachés, il en est évidemment de même pour les autres joueurs.

Précisions:

- Lorsque l'on drag and drop une carte elle est déplacée dans le fichier jeu.json qui répertorie l'état actuel du jeu.
- Les joueurs sont en équipe de 2, les 2 premiers à rejoindre la salle sont dans la même équipe , puis les 2 suivants dans la seconde équipe. Les joueurs de la même équipe sont situés face à face sur le terrain de jeu.
- Les jeux sont représentés comme une liste d'ID, ces ID sont associés à des cartes liées dans un fichier json ,le fichier cartes.json, à partir de celui-ci on peut, par exemple, associer un id avec un visuel de carte pour pouvoir l'afficher.
- Le plateau est mis à jour toutes les 5 secondes par une fonction location.reload(true)
- Le bouton "Ramasser le pli" accorde les points à l'équipe de la personne qui à cliquer sur le bouton, il faut donc déterminer qui à gagner le pli avant de cliquer sur le bouton. (Il aurait fallu calculer qui à remporter le pli pour attribuer automatiquement les points)
- Le bouton "Ramasser le pli" fonctionne avec l'aide d'un appel ajax, qui appelle une fonction php, cette fonction détermine quel joueur a cliqué pour attribuer les points obtenus à l'équipe de celui-ci.
- Les personnes qui rejoignent la salle alors que celle-ci est pleine ne peuvent pas interagir avec le jeu, ils ne peuvent qu'observer la partie et parler dans le chat.

Conclusion

Notre projet s'est achevé et nous avons rempli toutes les fonctionnalités principales que nous souhaitons implémenter.

La principale difficulté rencontrées lors de ce projet ont été, tout d'abord, le fait que notre fichier waiting.json devenait null à certains moments. Nous pensons que cela était dû au fait que parfois nous faisons plusieurs écritures dans ce fichier en même temps car ce dernier la fonction écrivant dans le fichier se lançait au chargement de la page plateau.php. Nous avons donc ajouté une page "escale" entre la salle d'attente et le plateau. Ce problème nous a fait perdre beaucoup de temps car nous avons eu beaucoup de mal à comprendre pourquoi il se produisait et, de plus, nous l'avons détecté très tard car il ne se produisait pas si souvent.

Lors de ce projet, nous avons notamment appris à utiliser javascript et php. Mais aussi à manipuler le format json.

Cependant, il nous reste quelques pistes d'amélioration pour ce projet :

- L'implémentation de l'atout , il n'y a pour le moment aucun moyen de déterminer quelle couleur sera l'atout de la partie, en conséquence les points sont pour l'instant calculés avec leur valeur "NonAtout" c'est-à-dire comme des cartes standards.
- Il faudrait pouvoir détecter quand un joueur à quitter la page pour pouvoir le supprimer de la liste des personnes en attente dans les plateaux.
- Nous voudrions pouvoir rafraîchir uniquement le jeu sans le chat et non toute la page comme actuellement car cela complique l'utilisation du chat. En effet, le délai de 5 secondes étant très court pour écrire mais lent en termes de fluidité pour une expérience de jeu agréable.