

**Journal de laboratoire du projet de web mapping de Clément Fougrouse :  
Jeu d'échec et brouillard de guerre**



ÉCOLE NATIONALE  
DES SCIENCES  
GÉOGRAPHIQUES



Commanditaire : Emmanuel Fritsch

Membres du projet : Kévin Beaupuy, Florian Bueno, Clément Fougrouse et Baptiste Riviere

## 1 Objectif principal et résultat

Mon rôle au sein de ce travail d'équipe a été de réaliser les tâches suivantes. Gameplay fonctionnel et JavaScript (JS). Partie du javascript du site portant sur les mécaniques de jeu : déplacement des pièces, promotion des pions, brouillard de guerre. C'est à dire traiter la demande de l'utilisateur quand celui veut et peut réaliser un coup. Il faut récupérer les informations du serveur et retourner des informations au serveur sur le coup choisi.

Vous pouvez visualiser le résultat final sur la plateforme principale disponible sur <https://chesswihou.herokuapp.com/>

Le code est disponible sur GitHub à l'adresse <https://github.com/baptisteRiviere/projet-web-mapping-ehecs>

## 2 Contribution personnelle

J'ai programmé les fichiers suivants entièrement :

- Le sous-dossier blanc :
  - partie\_blanc.js
  - partie\_blanc.css
  - partie\_blanc.html
- Le sous-dossier noir :
  - partie\_noir.js
  - partie\_noir.css
  - partie\_noir.html
- Le dossier images
- Promotion.js
- Promotion.html

## 3 Détails des résultats

Pour accéder directement à mes résultats il est possible de soi se mettre en local et grâce au logiciel MAMP de visé le bon chemin d'accès du dossier et de taper l'adresse suivant `localhost/partie/blanc/partie_blanc.html`

Ou sinon de taper directement sur internet l'adresse suivante `https://chesswihou.herokuapp.com/partie/blanc/partie_blanc.html`

Puis dans un deuxième onglet

`https://chesswihou.herokuapp.com/partie/noir/partie_noir.html`

Vous avez donc accès directement au plateau avec brouillard de guerre. De plus un bouton recharger est toujours présent, cela permet de recharger une partie (mettre la base de données à l'état initiale) et donc après rafraichissement des deux onglets, on peut effectuer une autre partie. Cette fonctionnalité est seulement fonctionnelle dans l'onglet des blancs.

Comme précisez précédemment j'ai codé les deux fichiers javascript qui permet l'interaction des joueurs avec la base de données. Le fonctionnement de ces deux fichiers sont très similaires.

Ce fichier permet tout d'abord la mise en place, l'initialisation des pièces, du plateau et du brouillard. Puis une fonction permet à un intervalle de temps régulier de regarder quel joueur à le trait. Quand, nous avons le trait alors une autre fonction permet de récupérer les coups possibles de joueur et le coup de l'adversaire si on peut le voir (envoie d'une requête au serveur via une méthode fetch) et permet aussi de débloquent le drag and drop des pièces (bloquer quand le joueur n'a pas le trait). Quand une pièce est sélectionnée par le joueur, les coups possibles sont montrés en mettant ces cases en surbrillance. Une fois que le coup a été choisi (fin du drop), une nouvelle requête est envoyée au serveur indiquant le coup que l'utilisateur a choisi via un fetch, le serveur nous confirme ce coup et nous donne les informations des nouvelles vues. Nous perdons également le trait et les événements de drag en drop sont supprimer jusqu'à qu'on obtienne de nouveau le trait.

Les fichiers html et css ont été effectuer avec Kévin Beaupuy pour que le site reste homogène dans la couleur et la structure.

Les Fichiers promotion.js et promotion.html sont des fichiers qui permettent de choisir le type de promotion ont souhaite obtenir quand les conditions sont réalisées. Ils sont directement appelés dans le partie\_blanc.html via une Iframe (Cela permet d'insérer plusieurs fichier html dans une même page, le script JS permet l'envoi d'information entre les deux pages).

#### 4 Détails des journées sur le projet Web-mapping

##### Lundi 13 décembre :

\*matin : Découverte des sujets lors d'une présentation effectuer par M. Fritsch. Notre groupe (Kévin, Baptiste, Florian et moi-même) avons été affecté à la programmation d'un jeu : un jeu d'échec en brouillard de guerre. J'ai été nommé chef de projet. Nous avons donc pris connaissance du sujet et identifié les différents points.

\* après-midi : Nous avons eu une réunion avec M. Fritsch pour obtenir quelques éclaircissements sur certaines interrogations que nous avons eu sur la compréhension du sujet. Nous avons également choisi de travailler sur un projet GitHub.

##### Mardi 14 décembre :

\*matin : La répartition des taches a été effectué en matinée. Je m'occuperai ce qui se passe quand le joueur joue un coup, et plus largement le coté interface graphique du javascript.

\* après-midi : Cette demi-journée a été consacré au dégrossissement de chaque partie. Pour ma partie, j'ai commencé à créer un rapide code html pour créer l'échiquier. Je me suis fortement inspiré d'une interface graphique des codes déjà existant. J'ai également lu les différents codes pour voir ce qu'il a déjà été fait et ce qui pourrait m'être utile. J'ai également décidé d'utiliser des images pour la représentation des pièces. Cette journée m'a permis de creer le plateau avec les pièces.

### Vendredi 17 décembre :

\*matin : Cette matinée a permis de commencer à implémenter le coup d'un joueur. J'ai effectué comme hypothèses que le joueur a le trait et qu'il a reçu les différents coups possible (création d'une variable temporaire des coups possibles remplis manuellement pour effectuer les tests et voir le bon fonctionnement des fonctions).

### Jeudi 27 décembre :

\*matin : Petite mise au point entre Baptiste et moi pour la structure : j'ai dû revoir le nom donné à chacune des cases et l'ordre de celle-ci (la case 11 se situe en bas à gauche). J'ai également continué à créer les fonctions utiles. J'ai donc mise en place un système de surbrillance des cases possibles quand l'utilisateur prends une pièce.

\* après-midi : Commencement à créer la structure de la future requête à envoyer au serveur pour la mise à jour (maj.php). Récupération de l'index du coup jouer et déplacement de la pièce sur le plateau.

### Dimanche 2 janvier :

\*matin : Je me suis consacré au cas d'une promotion. Pour cela j'ai effectué une IFrame qui permet de rajouter une page fille dans la page mère. Plus esthétique que l'ouverture d'un popup. Création de fonction utiles pour la mise en place de l'IFrame. Également cette méthode pourra être réutilisée pour l'apparition des demandes de match nul et des abandons.

\* après-midi : Depuis le début j'ai seulement utilisé un seul échiquier avec toutes les pièces présentes (blanc et noir) sans brouillard de guerre. Dans l'optique d'afficher à termes le brouillard j'ai donc créé deux js différents : un pour les blancs et un autre pour les noirs.

### Mardi 11 janvier :

\*Après-midi : Réunion entre les quatre membres du projet pour une mise en commun des codes sur GitHub et résolutions des conflits sur GitHub. Peu de conflits ont été détectés, puis j'ai commencé avec les conseils de Baptiste sur la structure de l'envoi, les différentes requêtes pour récupérer les coups possibles et également pour indiquer le coup que le joueur a effectué.

### Samedi 15 janvier :

\*matin : Restructuration de mon code pour intégrer les différents fetchs (envoi de requête au serveur). Suppression des variables de tests (coups\_possibles, promotions ...) et réalisation des commentaires du code pour une meilleure compréhension. Mise en place de la fonction qui permet de voir toutes les secondes qui a le trait (requête sur cr.php), elle permet de savoir quand déclencher les fonctions qui permet au joueur de réaliser un coup. Après avoir réalisé un test (ouverture de deux onglets (un blanc et l'autre noir) cela semble fonctionner pour le premier tour). En effet, lors de l'envoi du deuxième coup (coup noir) la base de données ne se met pas à jour. J'ai informé Baptiste pour qu'on puisse se pencher dessus quand nous aurons un temps disponible.

\* après-midi : Commencement à mettre le brouillard pour le jeu des blancs mais un petit problème est survenu car la fonction mise en place affiche seulement les nouvelles cases vues ainsi que les potentielles

pièces mais ne remet pas de brouillard sur les cases qu'on perd la vue. Par ailleurs aucun test sur l'apparition d'une pièce ennemi a été effectué.

#### Samedi 22 janvier :

\*matin : Suite au problème évoqué précédemment j'ai informé Baptiste dans l'optique de changer un petit peu la structure des rendus des appels du fichier maj.json.php. J'ai également effectué quelques tests sur le brouillard.

#### Semaine du 21 au 27 février :

Tout au long de cette semaine, une longue discussion s'est développée entre Baptiste et moi. En effet, le développement de ma partie a permis d'identifier de nombreux problèmes dans les retours du php. A chaque problème identifié (mauvaise vues, mauvais coups possible...) j'informais Baptiste et réalisait le coup (en visio ou de mon côté) pour qu'il puisse exporter la base de données enfin de regarder les problèmes et de corriger le code. Cette semaine a été décisive pour la correction des bugs chez Baptiste et pour le développement d'une partie jouée chez moi.

Par ailleurs, Baptiste avait développé un bouton « recharger ». Il permet de remettre à zéro la partie, c'est-à-dire, de remettre en position initiale les pièces au niveau du serveur. Cela a été extrêmement pratique pour effectuer des tests lors du développement pour tester les déplacements et les autres fonctions (au lieu de remettre manuellement toutes la base de données vierge à chaque fois qu'il y avait un problème). Nous avons décidé de laisser cette fonctionnalité présente dans notre code (que chez les blancs) pour permettre de tester les différents cas dans notre code et éviter de recommencer une partie. Cependant à terme quand le projet sera abouti et tous les liens opérationnels dans le site, il devra être supprimé. Nous avons donc délibérément choisi de le laisser car notre projet n'est pas totalement abouti.

#### Mardi 22 février :

\*matin : Baptiste m'a informé qu'il avait plus ou moins fini sa partie. Cela me permet de réaliser la mienne. En effet, même si j'ai pu effectuer des essais en créant des variables locales des coups possibles pour effectuer le déplacement des pièces, je dois attendre les résultats de Baptiste pour pouvoir avancer. Tout d'abord, la structure des données à envoyer au serveur est différente, de même pour les retours. J'ai donc dû revoir toutes les fonctions d'envoi au serveur.

\* après-midi : Cependant, le premier fetch (envoi de données au serveur) pour récupérer les premiers coups possibles pour les blancs, ne fonctionne pas. J'ai donc informé Baptiste de ce problème et après vérification de ses tests unitaires, il y a bien un problème sur les premiers coups.

#### Mercredi 23 février :

\*matin : Identification de nouveaux problèmes (des coups impossibles et cela est dû à une mauvaise position d'une pièce d'est le début coté php). De même lorsqu'une pièce est mangée. Baptiste a donc modifié le retour du json pour que je puisse ainsi obtenir toutes les informations nécessaires pour l'interface. J'ai donc créé de nouvelle fonction quand on se fait manger une pièce ou quand on mange une pièce. J'ai également aidé Florian en visio pour régler un petit souci de son côté.

\*après-midi : La plus grande partie de l'interface graphique est terminée : on peut jouer normalement des coups et prendre correctement des pièces ennemies et se faire manger correctement. Il reste néanmoins

le brouillard à implémenter ainsi que d'autres options comme les coups spéciaux, les cimetières et les abandons et demande de nul.

#### Jeudi 24 février :

\*matin : Mise en place du brouillard de guerre avec la création de nombreuses fonctions destinées aux vues débloquées et à l'affichage du brouillard. Celui-ci fonctionne plus ou moins car dans certains cas, il y avait des problèmes. Une fois Baptiste au courant, il commence les correctifs pour remédier aux problèmes.

\*après-midi : J'ai continué la mise en place du brouillard de guerre avec un résultat plutôt concluant. Et j'ai commencé à m'attaquer aux coups spéciaux. En effet, la structure des coups spéciaux est légèrement différente (plus d'informations à traiter) lors des renvois du serveur. Il faut donc les traiter séparément des coups classiques.

#### Vendredi 25 février :

\*matin : Cette matinée a été destinée à la fin des coups spéciaux (les roques et les promotions). De même, elle a aussi conduit à la mise à jour des cimetières. Cependant la prise en passant n'a pas été codée graphiquement. En effet, lorsque je voulais réaliser ce coup, celui-ci n'était pas renvoyé par le serveur.

\*après-midi : Tout le groupe s'est retrouvé dans une réunion pour le merge (la mise en commun des codes sur git) et la résolution des conflits. Même si j'ai beaucoup travaillé avec Baptiste et que nos codes sont vraiment complémentaires, il faut les relier avec les codes de Florian et de Kevin. J'en ai aussi profité pour reprendre des modifications du css et de l'html de Kevin ainsi que la récupération du pseudo de Florian. Une fois que la réunion finie, j'ai fini ma soirée à coder la fin de partie ainsi que l'abandon d'un des joueurs.

#### Dimanche 27 février :

\*matin : Mise en place d'un bandeau d'information (à qui de jouer, fin de partie...) et aussi à une bonne documentation du code. J'ai seulement documenté le code du javascript des blancs car en effet les deux javascripts sont très similaires avec beaucoup de fonctions similaires.

## En conclusion

Je me suis uniquement occupé de l'interface graphique. Cette interface graphique permet de réaliser une partie presque correctement (quelques bugs persistent et certaines fonctionnalités sont manquantes (prise en passant, demande de nul.)). Elle permet de récupérer à chaque tour les coups possibles et le coup que l'adversaire à effectuer en mettant le plateau à jour et bien sûr de permettre à l'utilisateur d'effectuer un coup, tout en envoyant au serveur les informations nécessaires.

Cependant plusieurs pistes d'amélioration sont envisageables. En effet, les deux codes JavaScripts sont très similaires, il serait peut-être envisageable d'en effectuer qu'un seul.

Par ailleurs, pas mal de fonctionnalités sont manquantes (de mon côté (js et graphiquement) et du côté serveur) comme la sauvegarde des parties en cours, la reprise d'une partie, la visualisation de l'arbitre (sans le brouillard et les deux camps en même temps). Aussi, il sera envisageable de créer une sorte de sauvegarde

temporaire des positions de l'échiquier car son on rafraichit la page, on perd la position de nos pièces et la partie est injouable.

Pour finir, il serait envisageable de continuer les liens entre ma partie et le site en général En effet, les informations des parties sont actuellement rentrées en dur dans le code et il sera important de pouvoir les récupérer via une méthode car pour l'instant seul, le site ne permet pas à lui tout seul de généré des parties et de pouvoir jouer.