

Hilmi CELAYIR

Ahmed BAAROUN

université
PARIS-SACLAY

Rapport De Projet Programmation Web

Table des Matières

INTRODUCTION.....	3
ANALYSE ET CONCEPTION.....	4
RESULTAT.....	6
CONCLUSION ET PERSPECTIVES.....	7

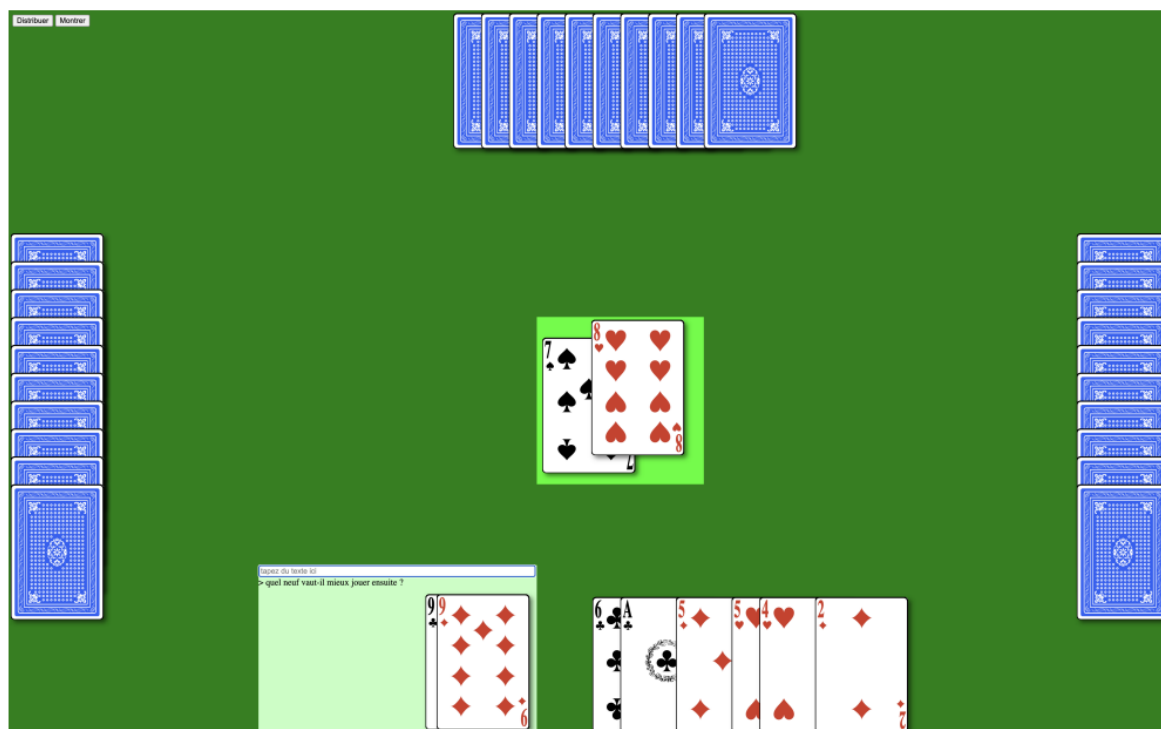
- INTRODUCTION :

L'objectif de ce projet est d'appliquer nos connaissances en développement web sous HTML, CSS, PHP, etc. afin d'en produire un projet complet, permettant d'apprendre aussi à gérer un projet, même petit. En effet, notre but est de créer un jeu de cartes, avec les règles et la variante de notre choix, sous forme de site web en ligne, et pouvoir y jouer en multijoueur, à partir de deux participants. Il faut pouvoir s'inscrire et se connecter à partir d'un formulaire.

La connexion déboucherait donc à ce moment-là sur la page de jeu.

Concernant justement ce dernier, à savoir le jeu, nous devons tout d'abord développer un moteur fonctionnel avec son affichage et son interface graphique.

Enfin, il ne restera plus qu'à modéliser les règles du jeu. Voici un exemple d'aperçu à avoir :



• ANALYSE ET CONCEPTION :

Pour résumer, le projet consistera, comme indiqué en introduction, à coder un jeu de cartes en ligne. Ainsi, quatre fonctionnalités principales sont attendues, soit :

- Un **formulaire** d'inscription/connexion avec gestion de réseau et de base de données
- Un **moteur** de jeu avec les commandes et l'affichage correspondant
- Les **règles** du jeu à modéliser et à implémenter en PHP avec des appels Ajax
- Un **chat** entre les joueurs pour pouvoir discuter

Le formulaire, est ici, une façon de réguler l'accès à une partie du jeu, avec un identifiant, une adresse e-mail, un mot de passe et même le niveau du joueur.

Il se compose de deux parties : l'inscription et la connexion.

L'inscription se fait avec une adresse e-mail, pour laquelle la difficulté était de ne pas pouvoir faire en sorte de vérifier son existence, néanmoins, nous avons pu vérifier son format (contenance d '@').

Si on tente d'inscrire un joueur avec un nom d'utilisateur déjà pris, cela nous est informé avec une page et un message le formulant :

Username already in use. Use another name

Puis, il faut entrer un nom d'utilisateur, un niveau de d'expérience et un mot de passe.

Nous avons par ailleurs à ce sujet un système d'encryptage de password, ainsi, si un mot de passe est erroné, un message d'erreur sera affiché comme ceci :

Wrong Password

Please Try Again [login](#)

Or, avant d'appliquer tout cela, nous avons utilisé un serveur et une base de données MySQL qui doit être manuellement créée au préalable. Nous avons utilisé MySQL car nous avons utilisé nos machines, et, n'étant pas familiers avec cela, nous avons été confrontés à certains problèmes de ports et d'accessibilité.

Les données entrées sont donc vérifiées au début de « game-mat.php », avec certaines gestions d'erreurs. Or, ici aussi, nous avons des anomalies, comme, par exemple l'apparition d'une page toute blanche en cas d'erreur, au lieu de l'affichage des messages d'erreur que nous avons prévu. Les joueurs et leurs informations sont ainsi stockés dans la base de données 'game' créée au préalable. A la suite de ces événements, des fichiers sont générés afin d'attribuer des cartes aux joueurs, et déposer des cartes sur le plateau et dans la pioche.

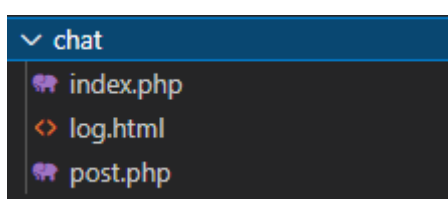
Ensuite, nous disposons d'un moteur fonctionnel avec son affichage et son contrôle. A ce sujet, deux joueurs ont été appelés dès le départ par des appels ajax, montrés ci-dessous :

```
$.ajax({
  url: "player1.html",
  cache: false,
  success: function(html) {
    player2 = html.split('<br>');
    let deck = '';
    //load 2 cards per time for the deck
    for (let i = 0; i < 8; i++) {
      id = player2[i].split('-')[1].split('.')[0];
      deck += '';
    }
    $("#two").html(deck); //Insert chat log into the #chatbox div
  }
});
```

```
$.ajax({
  url: "player2.html",
  cache: false,
  success: function(html) {
    player2 = html.split('<br>');
    let deck = '';
    //load 2 cards per time for the deck
    for (let i = 0; i < 8; i++) {
      id = player2[i].split('-')[1].split('.')[0];
      deck += '';
    }
    $("#two").html(deck); //Insert chat log into the #chatbox div
  }
});
```

Puis, grâce à d'autres appels ajax et des bibliothèques Javascript importées dans le dossier js de notre projet, nous avons codé une pioche (deck), et toutes les actions nécessaires au fonctionnement du moteur, tel qu'un système drag-and-drop. Cela nous a été possible grâce à la méthode des Listeners, permettant ainsi de lier les actions de notre jeu aux événements que l'utilisateur peut déclencher et contrôler. Pour cela, nous nous sommes inspirés de la méthode Modèle-Vue-Contrôleur, très répandue en Java.

Concernant le chat, il a été codé à part. Pour cela, nous lui avons consacré un dossier entier, afin qu'il ne puisse altérer les autres fonctionnalités, et pouvoir le remanier à notre manière. Il contient tout simplement une fonction php permettant de publier un message, vu par l'autre joueur, et une autre qui affiche le contenu du chat, qui est stocké au préalable dans un troisième fichier « log.html ». Ainsi, nous avons constaté que les messages passés des parties précédentes restaient dans le chat et devaient être effacés manuellement du fichier de stockage des messages. Or, nous ne nous sommes pas penchés plus sur le sujet, faute de temps et voulant accorder plus d'importance aux autres fonctionnalités. Voici justement l'architecture du chat :



Finalement, et malheureusement, nous avons eu du mal à mettre en relation différentes sessions de connexion, et donc une partie à part entière. Les joueurs sont donc indépendants et il n'y a aucune règle implémentée. De base, nous voulions modéliser une Huit Américain, une sorte de Uno, mais en utilisant des cartes classiques.

- **RESULTAT :**

Dans cette section, il est question de montrer le rendu de chaque page du projet. Il y en a deux, ce sont l'écran de départ, avec le formulaire et le système de connexion/inscription, puis le plateau de jeu avec toutes ses div.

The image shows a web form with a green background. The form is divided into two main sections. The top section is titled 'Login' and contains fields for 'Username:' and 'Password:', followed by a 'Login' button. Below this is a separator line with 'OR' in the center. The bottom section is titled 'Connect with' and contains fields for 'Email:', 'Username:', a dropdown menu labeled 'Experience status', and 'Password:', followed by a 'Connect' button.

L'utilisateur s'inscrit tout d'abord ici, et est ensuite ajouté à la base de donnée créée au départ. Un message s'affiche affirmant qu'on peut désormais se connecter.

Your account has been created successfully.

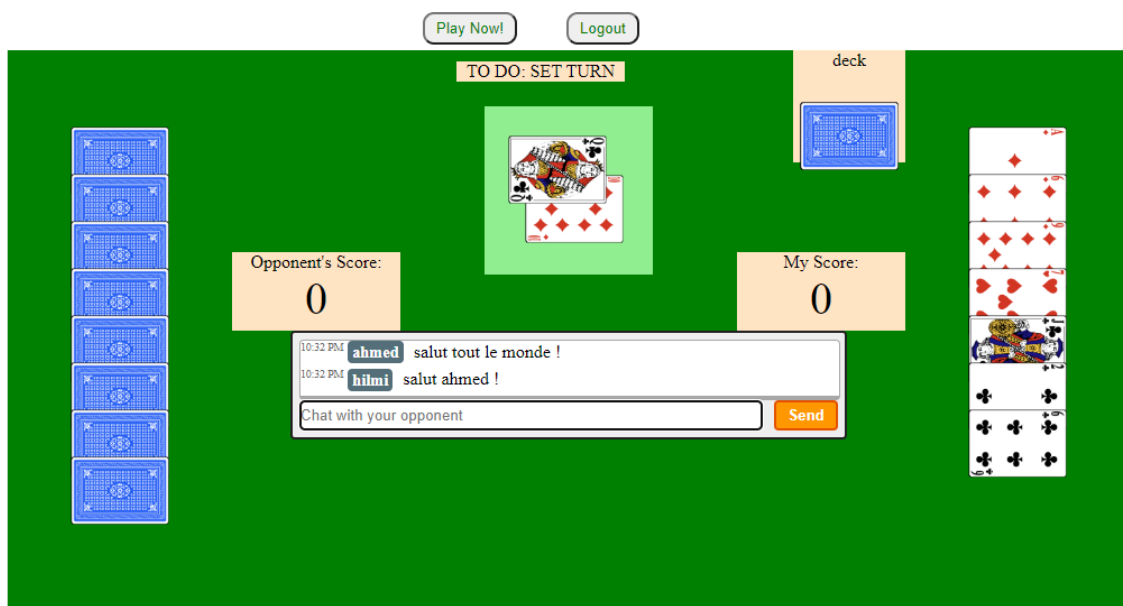
Here are your details:

Username: testrapport

Email: testrapport@testrapport

You can now [login](#)

Ensuite, après s'être connecté, on accède à l'écran de jeu. On aperçoit ainsi le jeu entier avec une partie au milieu représentant le plateau sur lequel les cartes peuvent être drag-and-droppées. Par ce système de déplacement, nous pouvons aussi piocher des cartes de la pioche (deck) et les ajouter à notre collection de carte en tant que player1. Les cartes seront montrées à ce moment-là. Le bouton Play Now! permet de commencer la partie, et Logout fait en sorte de sortir de la partie et de se trouver sur le formulaire de connexion/inscription. Le chat est aussi présent juste en dessous du plateau. Lors de la publication d'un message dessus, sa taille est mise à jour et réadaptée au fur et à mesure qu'il y ait des messages. L'heure est aussi renseignée pour chaque message envoyé.



• CONCLUSION ET PERSPECTIVES :

Pour conclure, ce projet nous a appris à mettre en application notre savoir-faire en programmation web, dans un contexte assez inédit, à savoir celui de programmer un jeu en HTML, CSS, PHP, etc.

Nous avons naturellement rencontré des difficultés, notamment pour les règles de jeu, mais nous avons innové à notre manière en intégrant un serveur et une base de données MySQL, avec une création de table, ainsi que des créations systématiques de fichiers HTML permettant de distribuer équitablement et au hasard les cartes entre les joueurs par exemple.