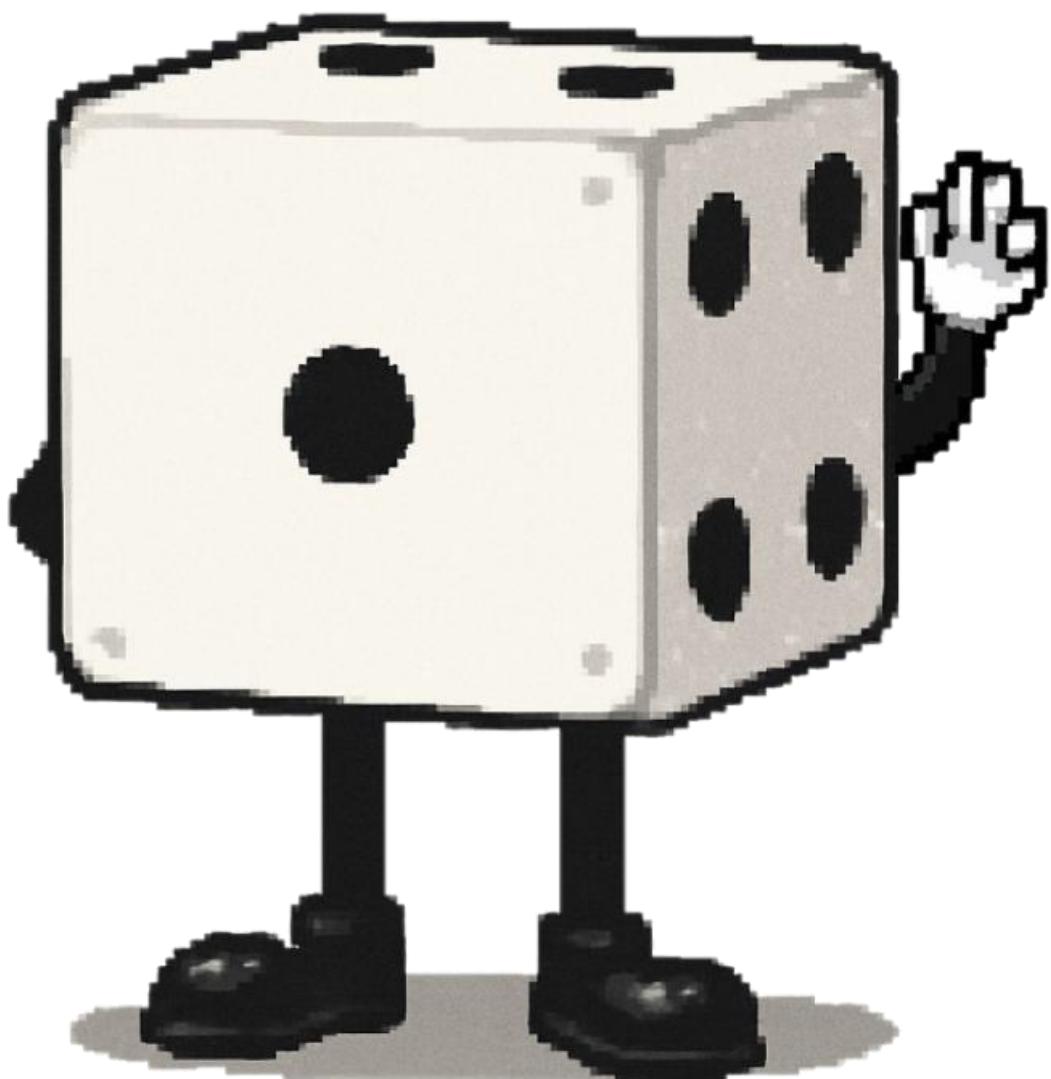


Projet « Game » :

Le dé menteur



Base de données

La base de données a été créée via PhpMyAdmin, et les tables ont toutes comme moteur de stockage InnoDB afin de pouvoir créer des relations entre elles.

Elles sont au nombre de 4 :

1. player :

La table player contient les données principales des comptes de chaque joueur ainsi que le montant de points « Elier » (Elo + liar) qui représente les points de classement obtenus à la fin d'une partie. Elle servira à connecter les joueurs, afficher leur classement ou modifier leurs infos.

2. game :

La table game contient les données de base des parties comme le nombre de joueur au départ, le nombre de joueur encore en vie, le pari en cours (s'il y en a un), le dernier événement de la partie, la date de début et de fin ainsi que l'ID du gagnant (non utilisé mais potentiellement utile pour une amélioration future).

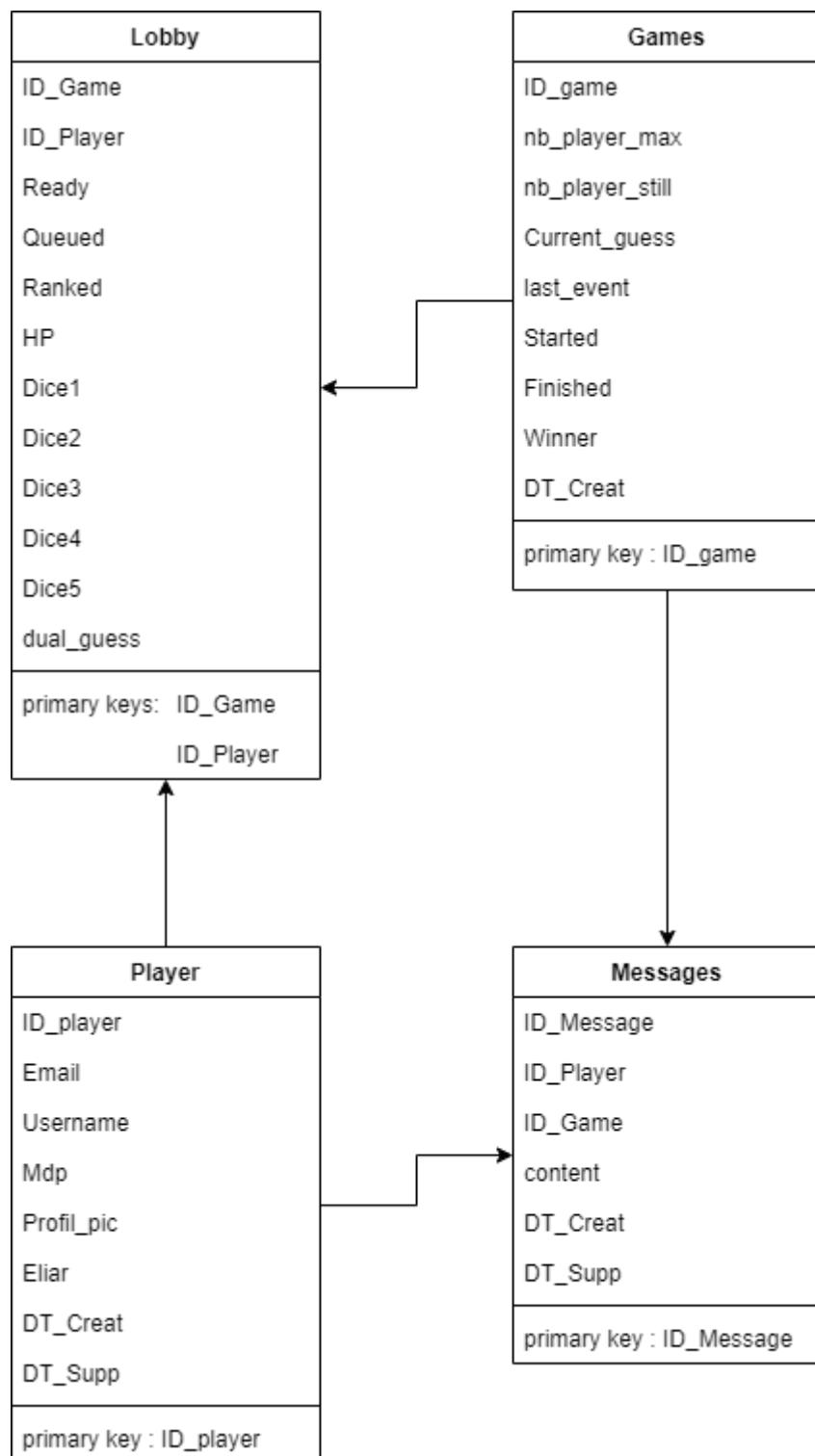
3. lobby :

La table lobby lie les ID des joueurs aux ID des parties dans lesquelles ils sont. Pour chaque entrée, elle contient le statut « prêt », les valeurs de chaque dé, le nombre de points de vie restants, la place du joueur dans la file, son classement, et la colonne « dual_guess » qui sera utile en cas de duel.

4. messages :

La table messages contient tous les messages de chaque joueur ainsi que la partie dans lesquelles ils ont été envoyés.

Schéma de la base de données



Choix techniques & conception

Le site en général :

Le site fonctionne en intégralité via une interface serveur en PHP (api.php), ce qui permet d'utiliser certaines 'actions' sur différentes pages tel que 'get_player_game' qui permet de récupérer les infos des joueurs d'une partie particulière et ainsi de les afficher dans la page 'lobby.html' ou bien dans la page 'game.html'.

Afin de pouvoir être certain que le jeu fonctionne correctement et que les changements s'affichent en direct pour tous les joueurs, une partie peut avoir 3 statuts différents :

Créée mais non commencée : Started et Finished sont null

Créée et commencée : Seul Finished est null

Terminée : Ni Started, ni Finished ne sont null

Cela me permet de faire passer tous les joueurs (qui sont tous prêt) de la page de lobby à la page du jeu, d'afficher les résultats de la partie terminée chez tous les joueurs et enfin de pouvoir déterminer où rediriger un joueur qui souhaite se reconnecter à sa partie via la page d'accueil.

Les différentes pages :

Le site contient les pages classiques de connexion, création de compte et accueil. Il n'est pas possible d'accéder à la page accueil ainsi qu'au pages permettant de jouer sans être connecté.

Hormis pour la page 'game.html', c'est le fichier 'header.js' qui fait cette vérification car il est utilisé pour toutes les pages nécessitant d'être connecté. Ces pages affichent la barre de navigation qui inclut la possibilité de modifier les infos de son profil. Elle également permet de rejoindre, de créer une partie ou bien encore de se déconnecter.

Lorsqu'on veut rejoindre une partie il y a deux possibilités :

- En créer une via la page ‘create_party.html’ :

Il suffit d'entrer le nombre maximal de joueur (min 2 / max 6).

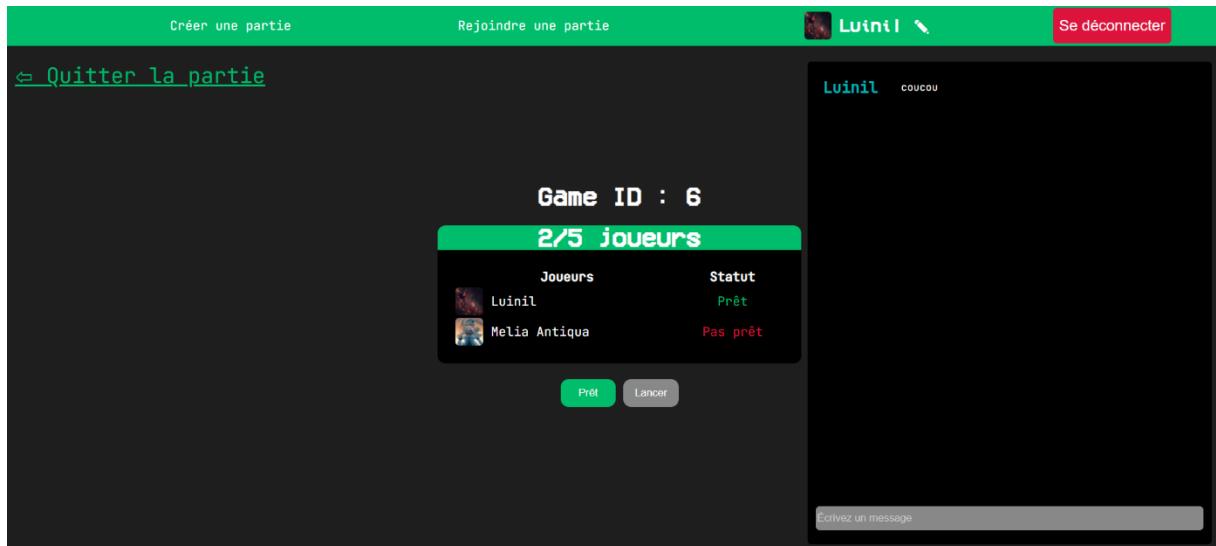
- En rejoindre une déjà créée mais non commencée via ‘join_party.html’ :

Il suffit d'entrer l'ID de la partie.

Ces deux méthodes renvoient ensuite le joueur vers la page de lobby de la partie créée ou rejointe.



La page lobby permet de regrouper les joueurs avant de lancer une partie et de s'assurer que le nombre maximum de joueur prêt est bien atteint pour lancer la partie sans encombre. Elle affiche également certaines infos comme l'ID de la partie mais aussi le nombre de joueur présent, le nombre de joueur nécessaire, le pseudo et statut de chaque joueur.



Il est possible de quitter une partie non lancée via le lien en haut à gauche.

Le chat est disponible sur la page de lobby et celle du jeu, c'est pourquoi il est créé grâce au fichier chatbox.js qui va chercher tous les messages d'une partie en particulier dont l'ID est récupéré dans l'adresse URL. Il affiche ensuite ces messages par ordre de création. Ce même fichier prend également en charge l'ajout de message.

La page home.html est la page d'accueil, si on essaye d'accéder à la page index.html, ce dernier nous renvoie soit vers la page de connexion, soit vers la page d'accueil. Cette dernière contient l'historique des parties du joueur connecté ainsi que les résultats si la partie est terminée, dans le cas contraire, un lien permettant de rejoindre le lobby ou la page du jeu en fonction du statut de la partie (cf. page précédente).

The screenshot shows a dark-themed web application. At the top, there are navigation links: "Créer une partie", "Rejoindre une partie", a profile icon for "Luinil", and a "Se déconnecter" button. Below this, on the left, is a section titled "Votre Historique" (Your History) containing a table of past games. The table has columns for Date, Classement (Placement), and Elier (Score). One row shows a game from June 7, 2025, with a "Rejoindre le lobby" (Join lobby) button instead of a score. On the right, there is a "Classement Top 6" (Top 6 Ranking) table listing six players: Luinil, Melia Antiqua, Flolink, Laeknir, Ciri, and Midna, each with their rank and score.

Classement Top 6		
Place	Username	Eliar
1	Luinil	40
2	Melia Antiqua	39
3	Flolink	14
4	Laeknir	0
5	Ciri	0
6	Midna	0

Elle contient également le classement des 6 meilleurs joueurs, ceux qui ont le plus de « Elier ».

Le chiffre 6, pour ce classement ainsi que pour le nombre maximal de joueur dans une partie, fait référence aux 6 faces d'un dé.

La page de jeu affiche les joueurs de la partie ainsi que leurs points de vie dans un tableau, sur lequel on peut remarquer que le pseudo d'un joueur peut être de différente couleur :

Vert : C'est le joueur actuellement connecté à la page.
Il reste de cette couleur même à son tour.

Bleu : C'est le joueur qui joue actuellement.

Blanc : Un joueur encore non éliminé qui attend son tour.

Rouge : Joueur éliminé.

The screenshot shows a game interface with a dark background. On the left, there's a table titled "Joueurs" with columns "Joueurs" and "Dés". It lists four players: Luinil (blue), Flolink (white), Ciri (white), and Melia Antiqua (green). Below the table, a message says "Vous êtes le Prochain !" (You are the next). In the center, it says "Aucun pari lancé, le Prochain joueur doit obligatoirement en soumettre un nouveau". Below this, it shows "Vos dés : [dice icons]" and "Faites un pari : [dice icon] de [dice icon] Parier OU Menteur !". On the right, there's a sidebar with player status messages:

- Melia Antiqua: tutut
- Ciri: test chat à 4
- Flolink: pas de soucis à priori
- Luinil: bientôt terminé
- Melia Antiqua: super

At the bottom right, there's a text input field with "Ecrivez un message" and a small button below it.

À la droite de ce tableau on peut trouver 2 éléments :

1. Le statut du tour du joueur connecté, il affiche « Votre tour » quand il doit jouer, « Vous êtes le prochain » ou « Veuillez patienter » quand ce n'est pas le cas. Il affiche également « Vous êtes éliminé » pour les joueurs sans point de vie
2. Le dernier évènement de la partie, cela permet de donner une info plus claire sur comment un joueur a terminé son tour, s'il a fait un nouveau pari ou si un joueur a perdu un point de vie (pas visible sur la capture).

Ensuite sur le dessous on peut retrouver le pari actuellement lancé (s'il y en a un), les dés du joueur connecté, les champs à remplir pour lancer un nouveau pari, les boutons pour lancer ce dernier ou accuser le précédent de menteur et enfin un PNJ proposant son aide.

Ce dernier permet, lorsqu'on clique dessus, d'afficher les règles du jeu. Aucun temps limite n'est imposé par joueur lorsque c'est son tour donc il peut lire à son aise.

Le jeu :

Le jeu est un dé menteur. Chaque joueur possède 5 dés, équivalent à leur nombre de vie.

Ces 5 dés ont des valeurs attribuées aléatoirement au début de la partie.

Le principe du jeu est de parier sur le nombre minimum de dé ayant une certaine valeur parmi l'ensemble des dés de tous les joueurs. Le joueur suivant pourra alors surenchérir avec un pari plus risqué ou accuser le précédent de « menteur ».

En cas d'accusation, le joueur ayant fait un pari erroné perd un dé (et donc un point de vie). En cas de pari gagnant, c'est l'accusateur qui perd un dé.

Si un joueur n'a plus de dés, il est éliminé.

Tous les dés restant sont ensuite à nouveau tiré aléatoirement et le joueur ayant perdu une vie, s'il n'est pas éliminé, commence la nouvelle manche.

En fin de partie, un cas particulier peut avoir lieu :

S'il ne reste que 2 joueurs n'ayant chacun qu'un seul dé, un duel commence ;

Les 2 joueurs vont maintenant devoir parier sur la somme des deux dés restants. Et ce toujours l'un à son tour. Le joueur s'en rapprochant le plus gagne.

En cas d'égalité, les joueurs recommencent le duel avec les mêmes dés.

En fin de partie, des points Eliar sont attribués en fonction du classement et du nombre de joueur en début de partie.

Voici la formule :

$$Eliar = (N + 1) * (N - K)$$

Avec : N = Nombre de joueur en début de partie.

K = Classement dans la partie.

Ainsi, des parties avec plus de joueurs rapporteront plus en cas de première place, mais toujours 0 en cas de dernière place.

Les affichages des changements dans la partie, que ce soit dans la page de lobby avec les joueurs, dans la page de jeu avec les paris ou encore les messages dans le chatbox se font via des intervalles qui vont lire les infos nécessaires dans la DB pour ensuite les afficher.

Tableau des heures réalisées :

LDB Projet "Game"		Florian Marchal	
Parties		Heures prévues	Heures passées
Maquette Figma		2	2,5
Schéma & Conception DB		2	1
Création de la DB sur PhpMyAdmin + Ajout de données		1	1
Création de la structure du projet (arborecence des fichiers, etc)		1	1
Style général du site (fonts, couleurs utilisées, etc)		1,5	3
Page de Login		4,5	3
	Structure HTML de base	0,5	0,5
	CSS de la page	0,5	0,5
	Js (ajout des events et fetch)	1,5	1
	Php (ajout des case à l'api et les requêtes sql)	2	1
Page de Creation de compte		4,5	2
	Structure HTML de base	0,5	0,2
	CSS de la page	0,5	0
	Js (ajout des events et fetch)	1,5	0,5
	Php (ajout des case à l'api et les requêtes sql)	2	0,8
Header + Déconnection		3	3
	Structure HTML de base	0,5	0,5
	CSS de la page	0,5	0,5
	Js (ajout des events et fetch)	1	1
	Php (ajout des case à l'api et les requêtes sql)	1	1

Page de Home		8,5	9,5
	Structure HTML de base	1,5	1,5
	CSS de la page	1	1
	Js (ajout des events et fetch)	3	4
	Php (ajout des case à l'api et les requêtes sql)	3	3
Page de Création de Partie		5	3,5
	Structure HTML de base	0,5	0,5
	CSS de la page	0,5	0,5
	Js (ajout des events et fetch)	2	1
	Php (ajout des case à l'api et les requêtes sql)	2	1,5
Page Rejoindre une partie		3	2,2
	Structure HTML de base	0,5	0,2
	CSS de la page	0,5	0
	Js (ajout des events et fetch)	1	1
	Php (ajout des case à l'api et les requêtes sql)	1	1
Page de Lobby		9	23,3
	Structure HTML de base	1,5	1,5
	CSS de la page	1,5	1,8
	Js (ajout des events et fetch)	3	10
	Php (ajout des case à l'api et les requêtes sql)	3	10
Page de Partie		11	35
	Structure HTML de base	1,5	3
	CSS de la page	1,5	2
	Js (ajout des events et fetch)	4	15
	Php (ajout des case à l'api et les requêtes sql)	4	15
Fonction de chat		Non prévu	4
Rédaction du rapport		5	4
Total		109,5	179