



Rapport final du projet Spin To Win

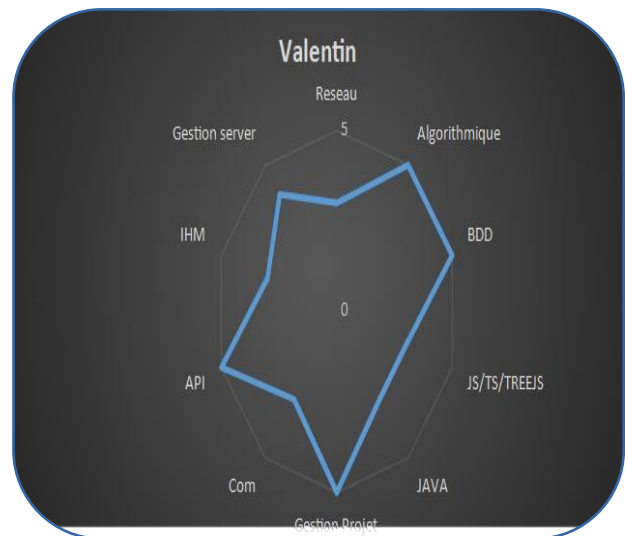
Fait le 20/06/2024

mail : vegastudio.online@gmail.com

linkedin : <https://www.linkedin.com/company/103122341/admin/feed/posts/>

Équipe VegaStudio ainsi que leurs compétences :

Valentin Lefevre

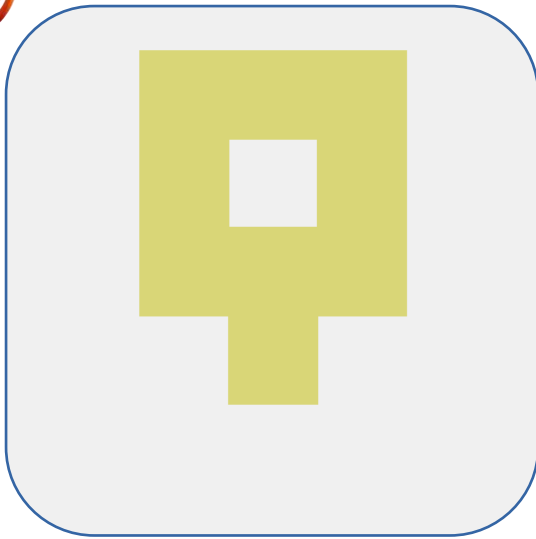


Lemine Majoub





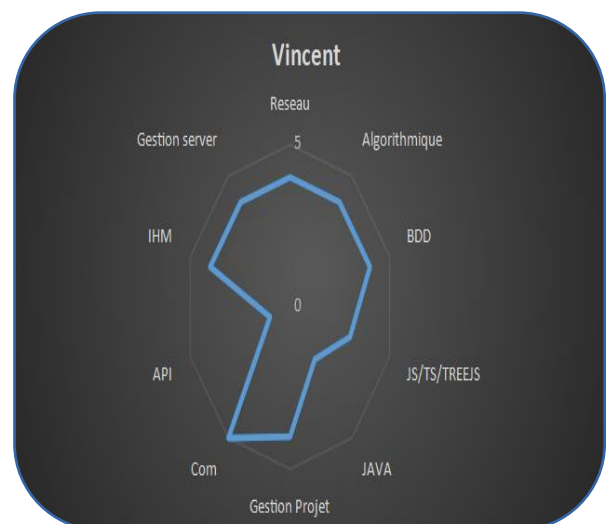
Elio Balastegui



Paul Arnaud



Vincent Chayé





Initialisation et consigne du projet :

Le projet doit être développé en Java et JavaScript, intégrant une démarche de gestion de projet pour structurer et organiser les étapes du développement, incluant planification et suivi.

Une partie du projet doit inclure des aspects de programmation réseau, comme la communication client-serveur.

Le développement de l'interface homme-machine (IHM) doit suivre les démarches étudiées en cours, en veillant à concevoir une interface ergonomique et intuitive pour l'utilisateur.

Le projet doit également intégrer une base de données pour stocker, récupérer et manipuler les données de manière efficace et sécurisée.

De plus, le projet doit incorporer des éléments en lien avec d'autres ressources du module, tels que l'utilisation de termes et de documentation en anglais, l'intégration ou l'utilisation d'outils numériques pour les statistiques, la mise en place de bonnes pratiques pour assurer la qualité du code et du projet, et la prise en compte des aspects légaux et contractuels liés aux technologies numériques.

En somme, le projet doit combiner les compétences techniques et théoriques acquises, en suivant une approche méthodique de gestion de projet, tout en intégrant des aspects réseaux, bases de données, IHM, et diverses ressources complémentaires.



SpinToWin



Spin to Win est une roulette en ligne avec des règles et des probabilités modifiées, trouvable dans le user guide de notre site.

Les règles sont simples. La roulette à 37 cases numérotées de 0 à 36. Un bille est lancé et, en fonction de là où tombe la bille, nous pouvons gagner de l'argent ou en perdre, cela dépend de la mise.

Nous misons sur une table des mises où plusieurs cases avec différents effets sont présentes. Nous déposons de l'argent sous forme de jeton sur une ou plusieurs cases et, en fonction de la position finale de la bille, nous pouvons gagner ou perdre de l'argent.

La bille est lancée toutes les 30 secondes. Une fois que la bille est lancée, il est impossible de miser jusqu'au prochain lancé.

Choix pour la V1 la V2 et la version finale :



Une fois le jeu de la roulette choisi, nous avons fait une réunion pour nous mettre d'accord sur ce que nous voulions exactement pour la première version fonctionnelle (V1) et la V2 et la partie finale (V3).

Version 1:

- Une roulette jouable sans animation
- ~~Connection~~/création a un compte
- Base de données local

Version 2:

- Animation de la roulette
- Base de données mise en place
- Jouable en réseaux
- Vue d'accueil/jeu

Version 3:

- Mise en place de statistiques
- Tchat
- Bonus

Choix de notre version le 06/03/2024

Nous nous sommes mis d'accord pour faire une roulette directement sur le site Web de notre entreprise. Avec le choix d'une roulette avec une case spéciale, un chat textuel ainsi qu'une croupière qui nous parle. De plus, pour toutes les parties, nous avons mis des dead-line pour chaque partie.



Sois nous sommes des génies, soit nous avons eu de la chance, mais nous sommes toujours restés dans les temps et aucune modification de notre plan de départ n'a été nécessaire pour arriver à la fin de notre projet.

Début du projet le 28/02/2024 :

- Annonce des équipes et première idée de jeu qui sort, mais encore rien de définitif le 28/02/2024

Véritable début du projet le 29/02/2024 :

- Choix définitif du jeu qui sera une roulette en ligne le 02/03/2024.

C'est à partir de cette date que la création a tout d'abord commencé. Nous avons notre projet en tête, chacun avec des particularités différentes, mais tous le même fond.

Pour mettre en place ce projet, nous nous sommes tous de suite dit qu'il fallait que l'on s'accorde et se coordonne le mieux possible avant de commencer à développer.

Du 03/03/2024 au 25/03/2024 :

Pour la mise en place de notre projet, nous avons commencé la recherche par de multiples réunions où les sujets étaient définis en avance :



Reunion 1 :

- Mise au point avec l'équipe car absent
- Faire mail et prepa presentation au prof.

Reunion2 :

- Montrer nos préférences sur l'attribution des tâches.
- Validations des technos.
- Création du GIT.

Reunion 3 :

- Présentation ThreeJS.
- Présentation Trello et Monday.
- Installation des environnements de dev

Du nom de l'entreprise → Du nom du jeu → Les deux logos ainsi que l'ambiance (choix des couleurs) le 13/03/2024

Puis nous avons mis en place un canal d'échange (photo ci-dessus) où on déposait toutes les idées de chaque membre, que l'on triait ensuite lors des réunions qui avaient lieu une à deux fois par semaine de cours. Pour ce genre d'échange, nous avons utilisé principalement Discord.



Quand le projet fut défini, nous avons commencé à nous renseigner sur les particularités des membres de notre équipe, les points forts et faibles. Puis nous avons défini nos tâches à accomplir ainsi que nos objectifs le 25/03/2024.

NOM	Reseau	Algorithmiq	BDD	JS/TS/TREE	JAVA	Gestion Proj	Com	API	IHM	Gestion server
Fermer		2	4	4	2	2	3	4	1	5
Elio		2	5	4	2	3	5	3	5	3
Lemine		2	5	3	5	3	4	4	3	5
Valentin		3	5	5	3	3	5	3	5	3
Vincent		4	4	4	3	2	4	5	1	4

Nous avons donc séparé notre projet en 5 parties principales :

- IHM
- API
- BDD
- Réseau
- Com/Gestion

Nous avons aussi, grâce aux différentes affinités avec les matières définies, choisi qui travaille dans quoi :

Puis nous avons élu pour chaque partie un référent qui transmet les informations et s'occupe du bon déroulement de sa partie.

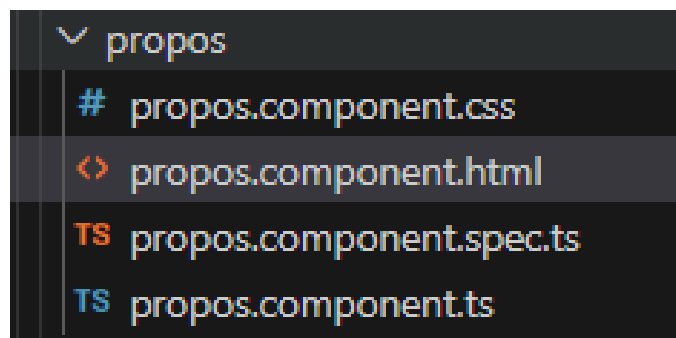
Développons maintenant nos différentes parties :

Avant ça, il faut savoir que nous avons fait le choix de passer par un framework nommé Angular pour simplifier la liaison entre les différentes parties (API Java → BDD → IHM).



IHM :

L'IHM comprend l'ensemble des tâches liées à l'interface utilisateur. Nous utiliserons donc premièrement pour l'ensemble du site de l'HTML et du CSS liés à du TypeScript (TS). Nous avons utilisé pour organiser toute notre partie frontend la méthode MVC (le modèle (Model), la vue (View) et le contrôleur (Controller)). Chaque vue de notre site est séparée des autres dans un dossier qui lui est propre avec sa page HTML, son CSS, ainsi que la vue en TS, et le contrôleur en TS. Cela nous permet de pouvoir travailler en même temps sur l'IHM sans avoir de conflit lors de l'assemblage.



Nous avons décidé pour notre jeu de partir sur un jeu en ligne. Le jeu se nomme SpinToWin et se trouve sur le site de notre entreprise (fictive) VegaStudio. Du côté de l'IHM, les pages SpinTowin sont « table » et « roulette ». Du côté des pages entreprises, nous avons « Vegastudio » pour la page d'accueil, « propos » pour la présentation de l'équipe et de l'entreprise, « jeu » pour les différents jeux que nous proposons. Enfin, il



y a les pages liées à la connexion de l'utilisateur et à ses données : « login » pour la connexion si notre compte a déjà été créé, « inscription » pour

la création du compte utilisateur et enfin la page « credit » pour les informations utilisateurs ainsi que ses statistiques liées à ses gains et ses pertes.

API :

Le projet a consisté en la création d'un service HTTP dans le but de pouvoir effectuer tous types d'appel API entre le frontend et le backend, avec une base de données H2 qui a été mise en place pour le développement. Une classe Joueur a été développée pour centraliser les informations des joueurs, et les premiers tests API ont été réalisés avec Postman, incluant la création, la mise à jour des joueurs, et l'authentification. Une classe d'interaction avec la base de données a été développée et la logique de la roulette a été isolée dans un package distinct.

Le projet, structuré en tant que projet Maven, comprend un package dédié à la logique de calcul, un jeu, un WebSocket pour la gestion des salons. Une connexion à une base de données MySQL a été établie pour la production, et un service WebSocket a été mis en place pour gérer les salons de jeu, permettant les connexions/déconnexions (jusqu'à 10 joueurs par salon). Le WebSocket permet de synchroniser chaque salon en séquençant une partie pour avoir au final 25 secondes sur la table pour miser puis 8 secondes sur la roulette et l'affichage du résultat. Un chat par salon a également été implémenté permettant de garder les 50 derniers messages. On a limité le nombre de messages pour éviter que le serveur sur lequel on se trouve plante.



*Voici le côté mathématique et statistique de notre roulette :

Génération de nombres aléatoires :

Utilisation de la fonction `Math.random()` pour simuler la roue de la roulette, produisant un entier aléatoire n tel que :

$0 \leq n \leq 36$ représentant le numéro sur lequel la bille s'arrête.

Calcul des gains :

Les gains sont calculés en fonction du type de pari :

- Plein : Pari sur un seul numéro avec une probabilité de $\frac{1}{37}$ et un paiement de 35:1
- Rouge/Noir, Pair/Impair, Manque/Passe : Chacun avec une probabilité d'environ $\frac{18}{37}$ et un paiement de 1:1
- Douzaines, Colonnes : Probabilité de $\frac{12}{37}$ et un paiement de 2:1

Probabilité et Statistiques :

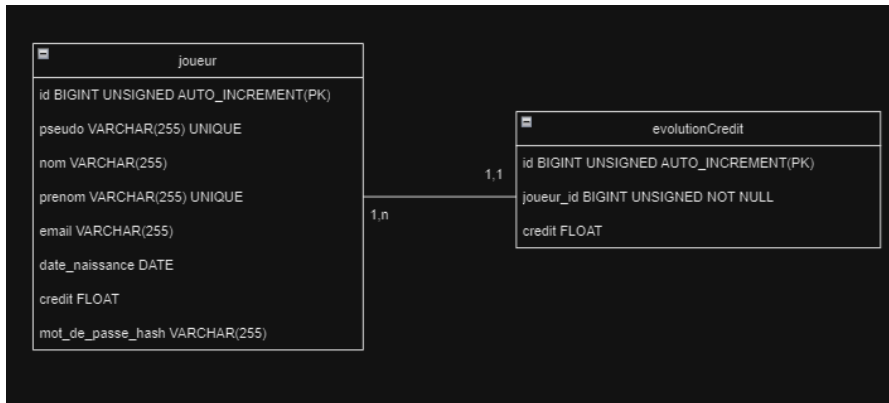
Les cotes de chaque type de pari sont basées sur la répartition des résultats possibles. Par exemple, les paris externes (couleur, parité, et manque/passe) reposent sur une probabilité de près de 50%, ajustée pour le zéro.



Validation des Entrées

Validation des paris pour vérifier qu'ils correspondent aux options disponibles et ne dépassent pas le solde du joueur. Cela inclut la vérification des montants pariés contre les crédits disponibles.

BDD :



Le schéma de donner et structurer en 2 tables une table, joueurs et une evlolutionCredit, la première stocke les informations du joueur et l'autre est là pour stocker les évolutions des crédits des joueurs BDD :

Nous avons d'abord créé le schéma de la base de données en définissant les tables et leurs relations à l'aide de scripts SQL. Ensuite, nous avons configuré une base de données H2 pour le développement, ce qui est pratique en raison de sa simplicité et de sa rapidité. Puis, nous avons mis en place une classe Java qui interagit avec cette base de données, en utilisant JDBC ou une autre technologie de persistance comme JPA/Hibernate.

Enfin, nous avons mis en place une base de données MySQL pour la phase de production. Pour ce faire, nous avons téléchargé le package nécessaire depuis le site officiel de MySQL. Vous pouvez trouver le package MySQL à l'adresse suivante :

[tps://github.com/Lemine-Mahjoub/Spin-To-](https://github.com/Lemine-Mahjoub/Spin-To-Win/blob/main/ALGO/spintowin/src/main/java/spintowin/DatabaseManager.java)

[Win/blob/main/ALGO/spintowin/src/main/java/spintowin/DatabaseManager.java](https://github.com/Lemine-Mahjoub/Spin-To-Win/blob/main/ALGO/spintowin/src/main/java/spintowin/DatabaseManager.java)



RESEAU:

Le côté réseau est énormément lié à l'API, car nous utilisons du WebSocket, ce qui nous a permis de créer nos rooms et notre channel de communication (chat).

Les WebSockets permettent une communication bidirectionnelle en temps réel entre un client et un serveur sur une seule connexion TCP. Initialement, le client envoie une requête de mise à niveau via HTTP. Si le serveur accepte, la connexion WebSocket est établie. Ensuite, les données peuvent être envoyées à tout moment par l'une ou l'autre partie, sans avoir besoin de rouvrir la connexion.

L'autre partie qui est liée à du réseau a été la mise en place du serveur, car nous avions comme idée première de prendre un serveur de l'hébergeur OVH, mais à cause de la non présence de SSH dans l'offre la moins chère, nous avons demandé l'accès à un serveur de l'IUT. Pour nous connecter à ce serveur, nous avons utilisé le Bureau à distance de Windows, ce qui nous a permis, après quelques péripéties de fonctionnement et de login, une connexion directe au serveur. Sur lequel nous avons mis en route notre jeu de manière continue. Le seul bémol d'être sur un serveur de l'IUT, c'est qu'il faut se connecter à un VPN spécifique pour trouver le site et jouer au jeu.

Droit des contrats :



Nous avons obtenu la licence d'exploitation, nos serveurs sont au sein de l'Europe et notre jeu est autorisé par L'ANJ. Notre jeu n'est pas ouvert

au mineur. Une liste de mot problématique à aussi été bannie pour qu'il n'apparaissent pas dans le chat.

COM/GESTION :

Pour la communication, nous avons, comme dit précédemment, mis un référant pour chaque partie, qui se doit de communiquer les avancements, les problèmes ainsi que de gérer les deadlines et de faire les rappels à ceux qui travaillent dessus.

Pour ce qui est de la communication extérieure, une seule personne gère cette partie, ce qui lui permet d'être au courant de tout pour pouvoir mieux ensuite repartir les informations pour le reste de l'équipe. Car si plusieurs personnes prennent et envoient les informations, il risque d'avoir des doublons d'e-mails ou bien de mauvaises transmissions d'informations.

Pour ce qui est des présentations, nous voulons être originaux. Nous présentons un jeu, donc nous voulons que les gens s'amusent même lorsqu'il s'agit de le présenter. Voilà pourquoi pour notre premier oral nous avons fait un quiz et voilà pourquoi pour notre présentation finale nous ferons #####. (Soyez présent le 28/06/2024)

De plus, nous avons créé une page entreprise LinkedIn, ce qui nous permet de communiquer des avancées de l'entreprise. Et nous avons tous un compte Google au nom de VegaStudio, ce qui nous permet d'avoir un e-mail d'entreprise, un Google Drive commun. Enfin, nous avons l'outil

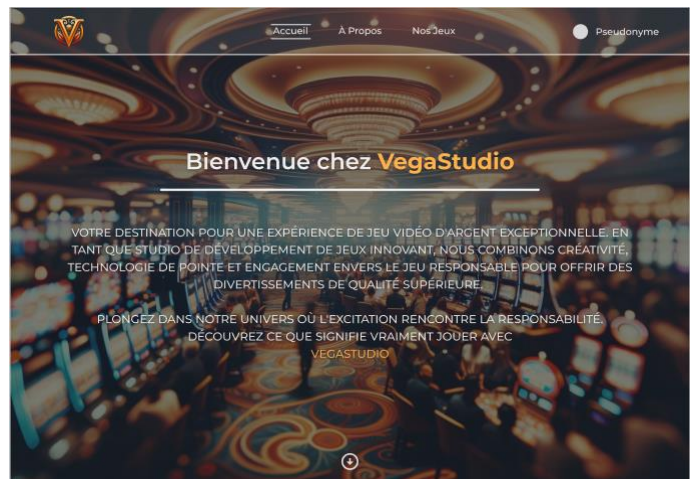
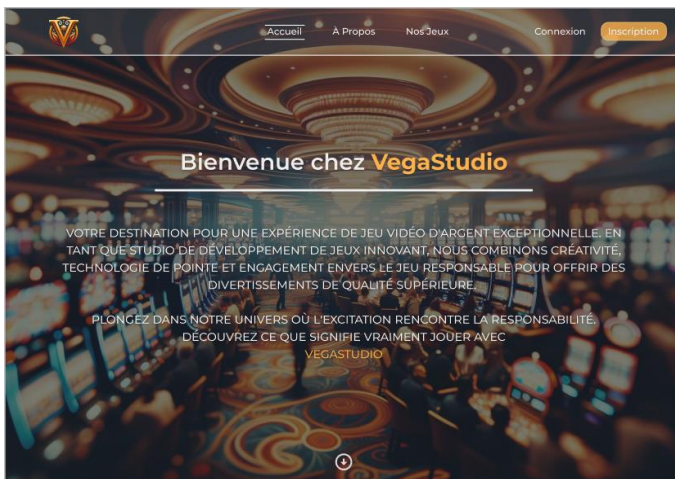


Monday qui nous permet d'avoir des tableaux de bord personnalisables, des intégrations avec d'autres outils de travail, des automatisations, et

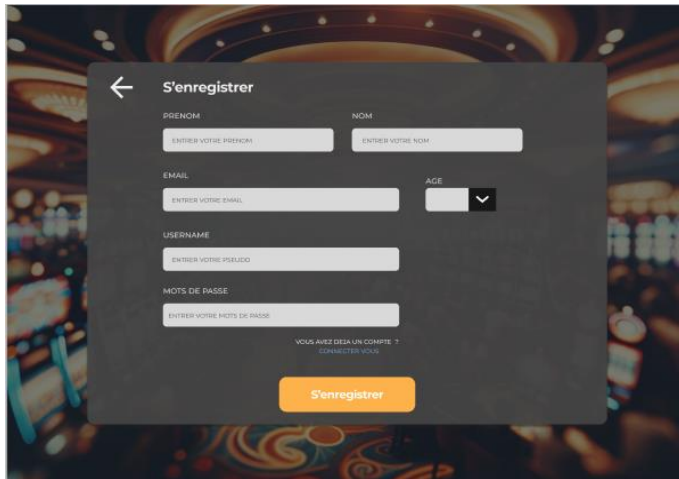
bien plus encore pour faciliter la gestion de projets et la collaboration en équipe.

Avancement de l'IHM :

Pour mettre au point cette partie du projet, nous avons commencé par créer un exemple de page Web sous Figma. C'est à ce moment que l'on a créé notre ambiance de casino :



Pour la suite, chaque page a la même image de fond et des couleurs de polices allant du blanc à l'orange/dorée. Nous gardons la même barre de navigation pour toutes les pages de présentations.



← S'enregistrer

PRENOM NOM

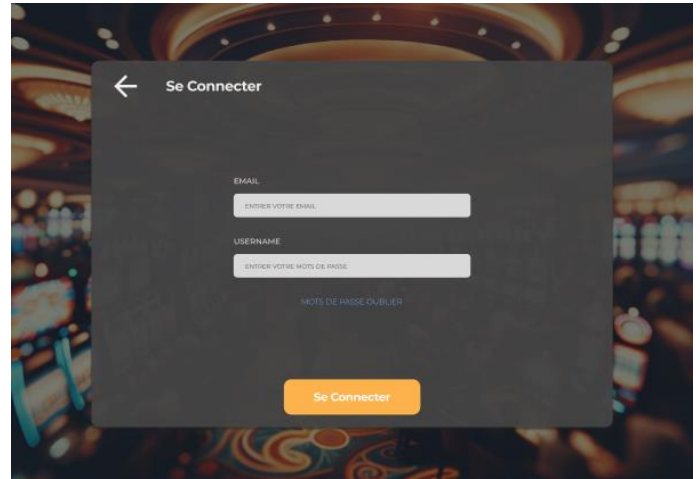
EMAIL AGE

USERNAME

MOTS DE PASSE

VOUS AVEZ DÉJÀ UN COMPTE ?
CONNECTER VOUS

S'enregistrer



← Se connecter

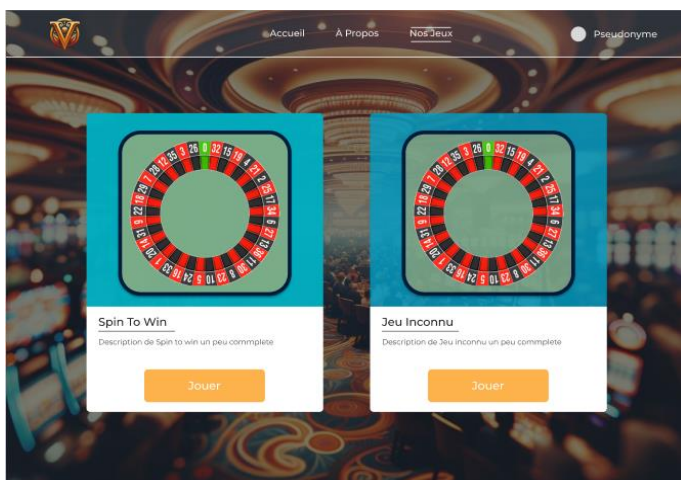
EMAIL

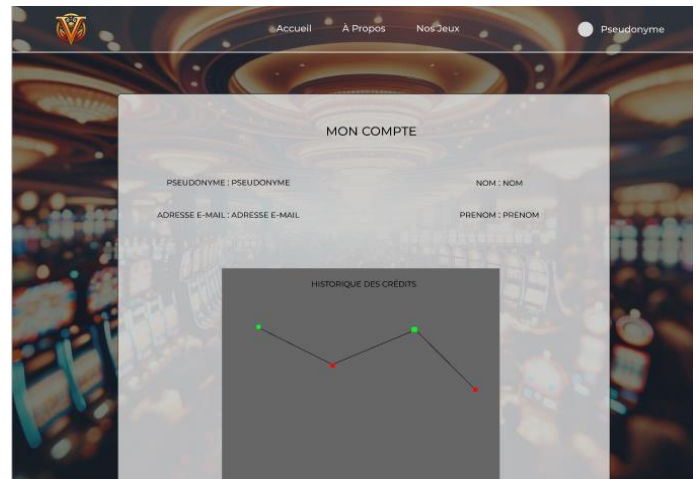
USERNAME

MOTS DE PASSE OUBLIER

Se connecter

L'ensemble des pages de présentation ont été créées sous Figma, ce qui nous a permis d'aller bien plus vite lors du développement.





Les images si dessus sont nos maquettes figma pour notre projet. Lien



figma :<https://www.figma.com/proto/WwSdnUTK3kPC19xx3zXajh/Spin-To-Win?node-id=0-1&t=4LrxwBsz8iE224OD-1>

L'ensemble du site de présentation a été fini fin mai, avec toutes les pages reliées correctement à l'API.

Pour l'affichage d'un jeu, nous avons créé une première version pour vérifier le bon fonctionnement, puis nous avons embelli la page pour arriver à notre version actuelle.

Ensuite, nous avons inclus dans la page Crédit la possibilité de voir les statistiques des gains et des pertes par un graphique.

Enfin, le channel de communication textuel est la dernière chose que nous avons implémentée. Nous avons eu plusieurs problèmes lors du placement sur la page, comme le problème de sélection pour écrire, ou bien la touche entrée non associée avec l'envoi des messages.

Avancement de l'API :

Le projet a débuté par la création d'un service HTTP pour les appels PUT et GET. Ensuite, une base de données H2 a été mise en place pour le développement initial. Une classe Joueur a été créée pour rassembler les informations des joueurs. Les premiers appels API ont été testés avec l'outil Postman, incluant la création de joueurs, la mise à jour des informations des joueurs, et l'authentification.

Par la suite, une classe a été mise en place pour interagir avec la base de données. Après la création du front de la roulette, la logique d'Elio a été



placée dans un package séparé. Une connexion à une base de données MySQL a été établie pour une utilisation en production.

Un service WebSocket a été développé pour gérer la gestion des salons, incluant les connexions et déconnexions avec un maximum de 10 joueurs par salon. Ce service a également permis de synchroniser le déroulement des parties dans chaque salon, garantissant que tous les joueurs voient le même numéro tomber sur une période de 25/8 secondes. De plus, un chat par salon a été mis en place, basé sur une file de taille 50.

Enfin, un schéma de données basé sur la classe Joueur a été créé pour structurer les informations dans la base de données.

Avancement :

Les avancements représentent les parties qui ont duré sur le long terme. Seuls l'API et l'IHM ont été long à faire.

Les ressentis de l'équipe :

Lémine :

Je ressentais une grande joie de travailler dans un nouveau projet de groupe, sur un projet commun que l'on appréciait tous et où on était tous dans le domaine qu'on aimait, malheureusement j'ai dû changer de projet en cours de route pour des projets personnels qui me tenaient beaucoup à cœur, mais cela ne change en rien mon amour pour le projet SpinToWin que l'on a tous façonné à la main avec dur labeur.

Vincent :



Au début du projet, j'étais assez stressé sur le déroulement que l'on mettrait en place, car durant la première SAE, c'est ça qui a posé problème dans ma première SAE. Mais finalement, tout s'est passé à la perfection et l'ambiance de l'équipe a beaucoup y a grandement participé.

Elio :

Au début du projet, j'appréhendais le rendu final au vu de la complexité du casino en ligne, de son économie et de sa gestion multijoueur. Mais avec quelques alternatives et une contribution en groupe, nous sommes parvenues à un résultat satisfaisant dont je, et nous, sommes fiers.

Paul :

Au début de notre projet de SAÉ, la charge de travail me semblait décourageante. Cependant, grâce à une répartition efficace des tâches et à une bonne entente au sein de notre équipe, j'ai pu me remotiver. La bienveillance et la coopération de mes coéquipiers m'ont aidé à assumer cette charge de travail. Je suis très content de cette fin de projet qui me semble très satisfaisante en vue de l'écart de difficulté entre la première SAÉ et cette deuxième.

Valentin :

Au début du projet, j'avais peur de la charge de travail et de la difficulté du projet, puis au fil du temps, le projet a avancé de mieux en mieux, ce qui m'a reboosté et motivé à travailler encore plus pour qu'au final, on se



retrouve avec des fonctionnalités qu'on pensait impossibles à intégrer au début. Je suis content d'avoir participé à ce projet.

Future du projet :

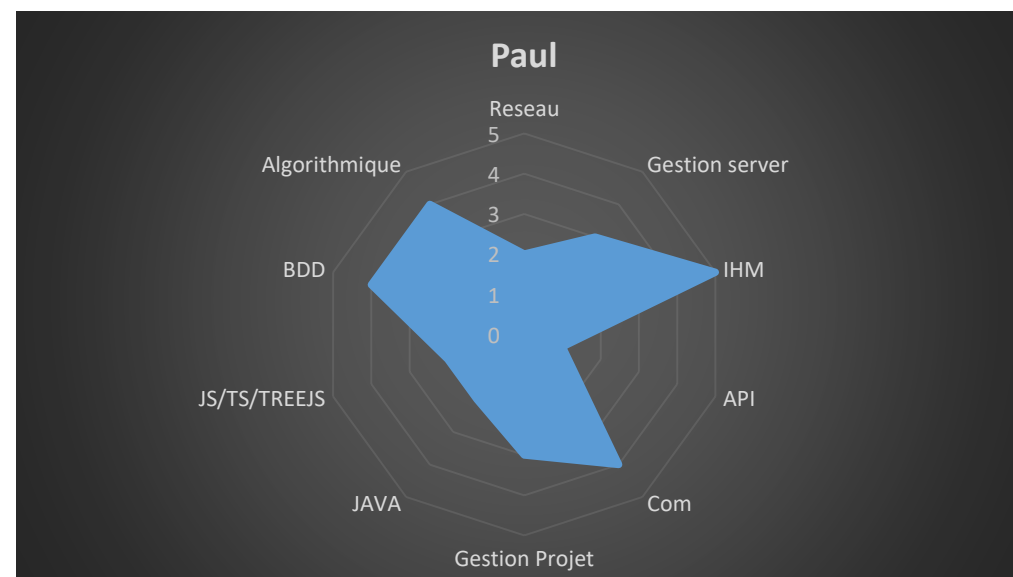
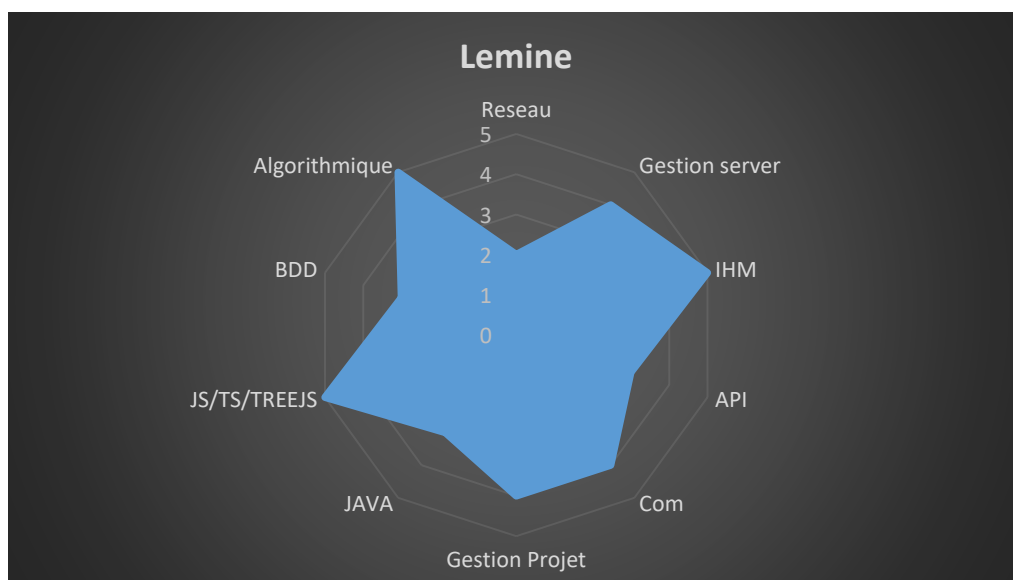
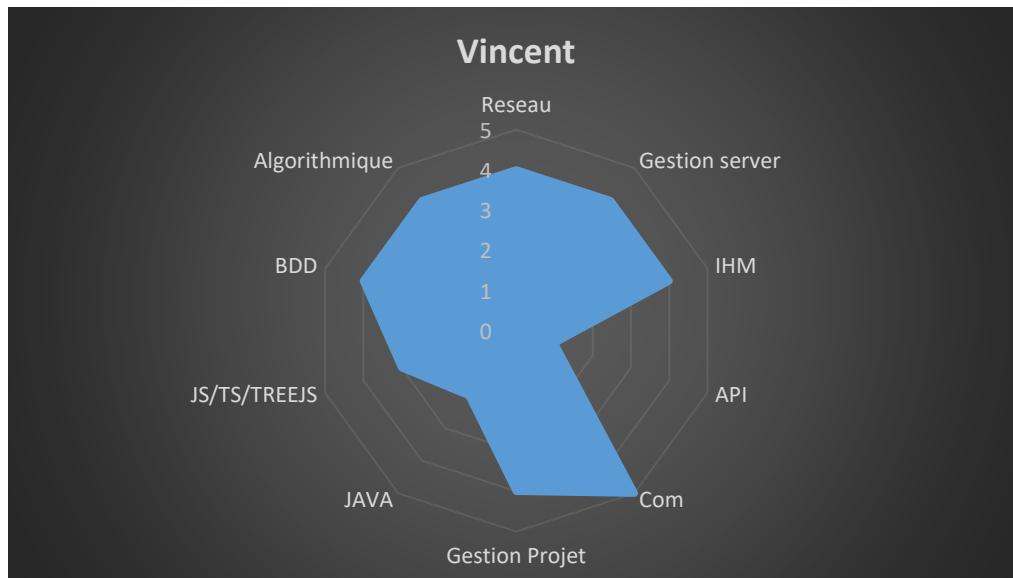
Le future de ce projet ne serais pas de développer SpinToWin plus que ça mais plutôt de faire un autre jeu du casino. Comme par exemple le blackjack.

Conclusion :

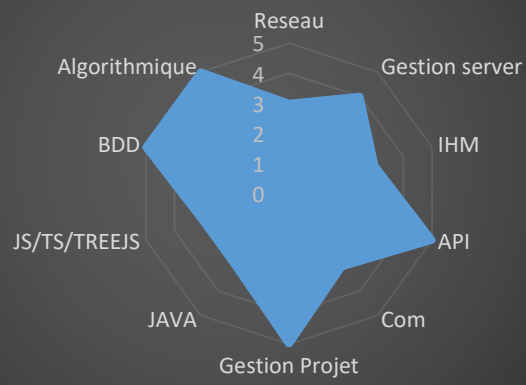
Finalement, ce projet qui à duré 4 mois, c'est passé sans gros problème et pour un résultat dont nous sommes très fières. Le travail en équipe a été un vrai plaisir et nous retiendrons tous les moments de partage et d'amusement qu'a entraîné ce projet. Si nous, l'équipe, nous en sommes très fières, c'est maintenant à vous de vous faire votre propre avis sur ce jeu qui, on espère, saura vous plaire autant qu'à nous.

Bon voyage dans le monde sensationnel de VegaStudio !

Annexe au cas où les images ne sont pas visibles entièrement :



Valentin



Elio

