

Présentation du projet

SOAB – System of a Board, est une Application Web de jeux de société modernes en ligne. Fondé sur les problématiques d'organisation et de temps nécessaires pour participer à ce genre de sessions, la solution promet de retrouver un panel d'abord restreint, puis augmenté, de jeux de société modernes, retranscrit et accessible à partir d'un site web. Si la solution est d'abord une alternative, elle offrira aussi la possibilité de tester de nouveaux jeux avec ces amis ou même de parfaits inconnus.

Alors que plusieurs autres solutions du genre existent sur internet, la plupart, vieillissantes, ont un manque à gagner en terme d'optimisation, d'accessibilité et d'esthétique. Tandis que deux principaux concurrents tiennent le marché : BoardGameArena et Tabletopia. Le premier dispose de nombreux bugs, alors que le second, plus stable, propose une solution de jeux en 3D, dont la praticité de jeux peut ne pas plaire à tout le monde.

System of A Board veut proposer une solution récente à la praticité 2D, plus axée sur le fond du jeu et la partie que l'immersion 3D.

Les cibles ? Majoritairement des habitués du genre, voulant palier aux problèmes de temps et d'organisation de ce type de session, qui peuvent prendre jusqu'à une journée entière. Tout en proposant un accès à de potentiels nouveaux utilisateurs et néophytes.

Nom	Praticité	Ergonomie	Performances	Plateformes	Types de jeux
Board Game Arena	2D ✓	Moderne ✓	De nombreux bugs ✗	Navigateur ✓	Modernes ✓
Tabletopia	3D ✗	Moderne ✓	Bonnes ✓	Navigateur, Mobile, Steam ✓	Modernes ✓
Yucata	2D ✓	Ancienne ✗	Bonnes ✓	Navigateur ✓	Modernes ✓
Plato	2D ✓	Moderne ✓	Bonnes, seulement mobile ✓	Mobile ✗	Classiques ✗
TableTopSimulator	3D ✗	Moderne ✓	Bonnes ✓	Steam ✗	Classiques ✗
Solution possible	2D ✓	Moderne ✓	Bonnes ✓	Navigateur ✓	Modernes ✓

Présentation technique

Le projet va fonctionner sur deux applications, l'une frontend servira à gérer les comportements, règles de gestion et pages qui régiront le site web, tandis que l'autre, le backend métier, traitera les données sensibles pour les renvoyer au front lors de la réception d'appels API.

Le frontend fonctionnera à l'aide du framework javascript ReactJS pour la partie applicative, aidé du langage dynamique scss, ainsi que de la librairie ThreeJS pour les jeux.

ReactJS étant une librairie réputée qui utilise Node, elle sera un outil important d'organisation du projet en composants et de montée en compétence, là où l'ajout de SCSS permettra la dynamisation de l'attribution de classes, de manière à respecter une propreté de code et de limiter le travail des fichiers composants sur les feuilles de style. De son côté, le framework ThreeJS va offrir une librairie stable de modélisation visuelle javascript au challenge technique durcit, comparativement à des outils de création de jeux et d'exportation WebGL tels que Unity. Dans le but d'offrir une fluidité de réponse, la majorité des traitements asynchrones se feront sur le front, de sorte à limiter les interactions de données de jeux avec le back et ainsi limiter les temps de réponses.

Le backend fonctionnera lui à l'aide du framework javascript NodeJS Express. Il aura pour but de traiter majoritairement les données nécessitant un calcul synchrone, ainsi que de modéliser les DAO qui traiteront avec la base de données MySQL. La connexion entre joueurs se fera au moyen de la librairie Socket.io, permettant une communication bidirectionnelle en temps réel.

Au niveau Intégration Continue, il est prévu que le projet fonctionne sur un serveur virtuel distant disposant de Jenkins, de sorte à faciliter le déploiement de nouvelles versions sur le Git, mais aussi d'effectuer des scans de code, notamment via Sonar ; les tags de version git serviront à valider ou non les déploiements de version. Un repo github disposant des droits CI/CD disposera de pipelines servant à valider les merge request d'US avant qu'elles soient envoyées sur la branche de feature. Aussi, chaque push se fera vers deux repositories, celle de l'ETNA et celle du Github créé pour l'occasion.

Parmi les limites analysées, ReactJS est un excellent framework qui, comme la majorité des framework JS récents, automatisent une grande partie du fonctionnement du projet. Il ne sera pas possible de maîtriser au plus bas le fonctionnement du frontend et une architecture irréprochable sera demandée. Il en va de même pour NodeJS Express, dont l'architecture des models, services, controllers, DAO et DTO devra être maîtrisée. Néanmoins, la principale limite de notre projet sera le temps. Proposer un POC de librairie de jeux signifiera de montrer, non seulement que l'application web est fonctionnelle, mais qu'elle fonctionnera sans problèmes en ligne, tout en ayant un minimum de jeux à présenter.

L'équipe et son organisation

Le groupe est composée de 7 personnes qui ont des rôles pour définir leurs taches pour ce projet.

Groupe

Quentin (spécialité Réseau) : Game designer

Jeremy (spécialité Web) : Game designer

Sabri (spécialité Mobile) : Game designer

Julien (spécialité Web) : Développeur Back

Adrien (spécialité Web) : Développeur Back

Jean (spécialité Web) : Développeur front

Justine (spécialité Mobile) : Développeur front

Rôle et Tâches

Les Games designer seront charger de la conception des jeux.

Les Développeurs Back se chargeront de création et gestion de l'API.

Les Développeurs Front se chargeront de gérer création et gestion de visuel client.

Compétence

Les games designer devront découvrir Three Js

Les développeurs back devront apprendre le NodeJS avec Express

Membre du groupe	Quentin	Jean	Jeremy	Juien	Justine	Sabri	Adrien
Compétence	Réseau	Web	Web	Web	Mobile	Mobile	Web
Taches repartie selon les roles	Game Desigr	Front	Game Desigr	Back	Front	Game Desigr	Back
Compétence a apprendre	Three js	React	Three js	Node	React	Three js	Node