

ANDRIAMISAINA Tiavina 2203040

AIT ADDI Marwan 2235395

Rapport de Conception Agile de **Projets Informatiques : Application** **Planning Poker**

Chargé d'enseignement : Lachand-Pascal Valentin

M1 Informatique 2023-2024

Introduction

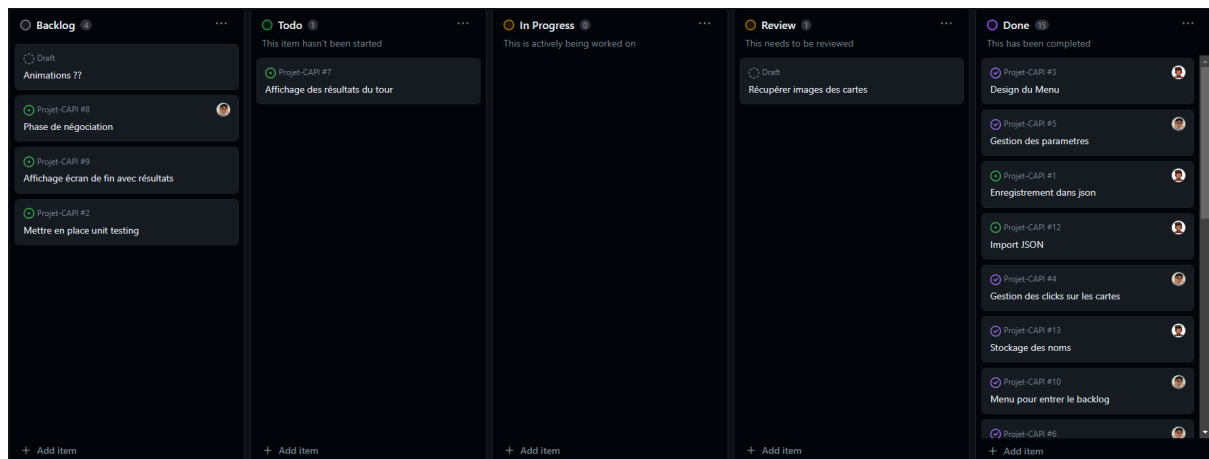
L'objectif de ce projet est de créer une application de Planning Poker fonctionnel, comportant plusieurs fonctionnalités :

- Une interface claire et agréable
- La possibilité de choisir entre plusieurs mode de jeux (ici Unanimité et Majorité)
- Pouvoir rentrer les backlogs depuis un fichier JSON
- Pouvoir rentrer les backlogs directement depuis l'interface
- Pouvoir enregistrer les backlogs dans un fichier JSON

Nous avons choisi de développer cette application avec des technologies Web (HTML, CSS, JavaScript) car celles-ci nous permettent une plus grande liberté dans l'élaboration de l'interface tout en nous permettant d'utiliser un langage puissant et polyvalent que ce soit dans l'affichage dynamique sur la page web et la gestions des différentes variables derrière. De plus, l'utilisation de CSS nous a permis de faire une première ébauche de l'interface grâce à l'outil Figma.

Méthodologie

Nous avons tout d'abord ouvert un nouveau projet sur GitHub afin de faciliter le travail collaboratif. Nous avons ensuite créé un tableau Kanban afin de pouvoir suivre l'avancement du projet et connaître les tâches qui nous restaient à faire. Afin de déterminer la difficulté des tâches nous avons fait une session de planning poker. Quant à la répartition des tâches, chacun a simplement effectué celles qui lui plaisaient le plus et/ou les plus urgentes.

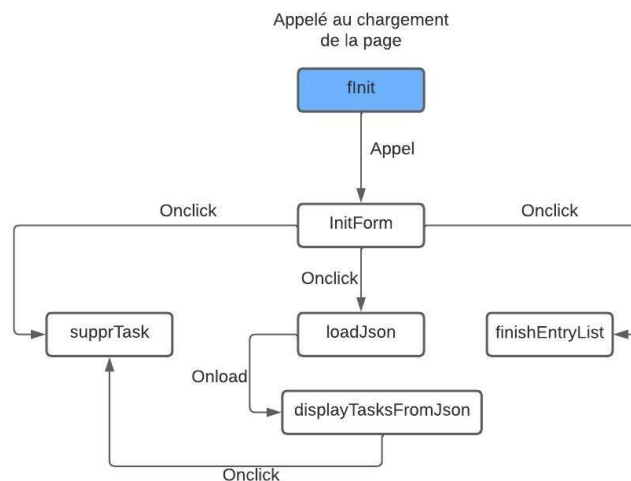


Toutes les tâches ont été converties en branches du projet et ont ensuite été testées et validées avant de l'intégrer dans le rendu final.

Pour l'avancée du projet, nous avons choisi d'utiliser la méthode agile SCRUM comme vu en cours. Elle consiste à alterner les périodes de sprint et des réunions fréquentes. Les sprints avaient une durée d' une semaine, et à chaque fin de sprint nous nous appelions pour nous mettre à jour sur le travail que nous avons chacun effectué.

Designs patterns

Pour cette application, nous avons utilisé le design pattern façade. Ce design pattern nous permet de regrouper toutes les fonctions complexes dans une seule fonction qui sera appelée au chargement de la page. Cette méthode permet de rendre le code plus lisible et maintenable. Les fonctions contenues dans celle-ci seront donc appelées uniquement lorsqu'elles seront nécessaires.



Fonctionnement du fichier entry.js

Choix Techniques

L'enjeu principal lorsque l'on utilise uniquement du Javascript c'est de pouvoir passer des variables de pages en pages. En effet, pour que les paramètres, les pseudos et les backlogs choisis soient correctement utilisés dans la page de jeu, il a fallu trouver un moyen de faire passer ces variables dans des pages différentes de celles où elles sont créées.

Nous ne voulions pas utiliser le langage PHP dans notre code bien qu'il nous aurait permis de faire passer ces variables directement depuis l'URL ou par l'intermédiaire des formulaires.

Nous avons donc décidé de sauvegarder localement ces variables dans le navigateur grâce à la commande localStorage proposée par Javascript. Les variables sont alors sauvegardées dans un fichier JSON enregistré localement que nous n'avons qu'à appeler afin de récupérer les données. Ce fichier JSON est invisible pour l'utilisateur et ne nécessite aucune action de sa part.

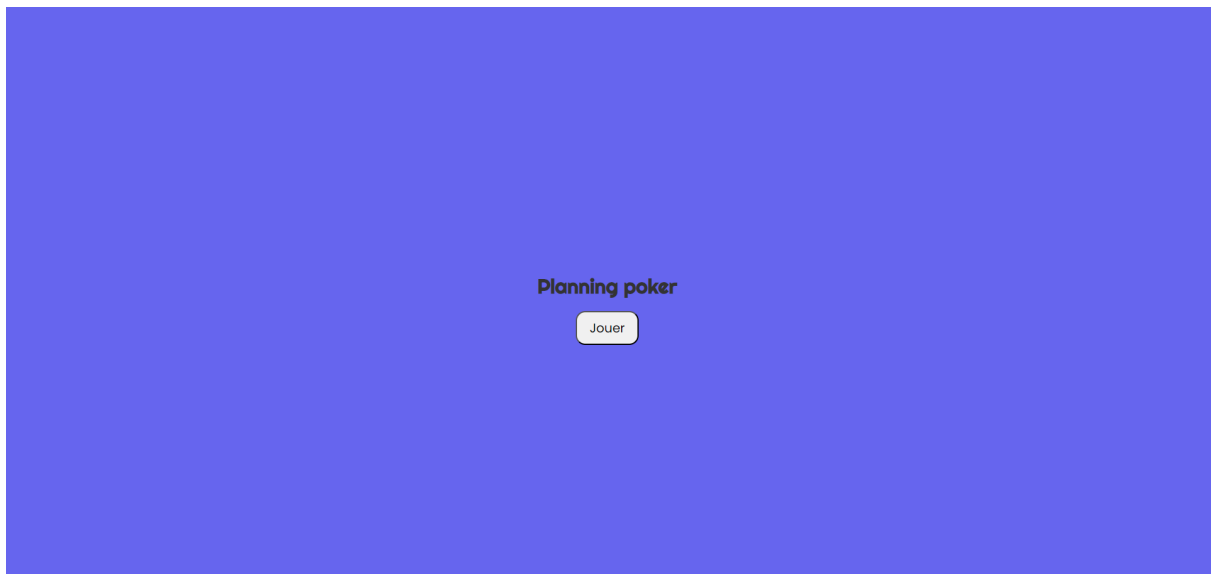
La non utilisation du langage PHP implique également que l'application ne fonctionne qu'en local, c'est -à -dire que les utilisateurs doivent être sur le même appareil. Cela rend également le déploiement du site beaucoup plus simple.

Architecture de l'application

L'application consiste en plusieurs pages, ayant toutes une seule fonction afin de rendre la navigation et l'interface la plus claire possible.

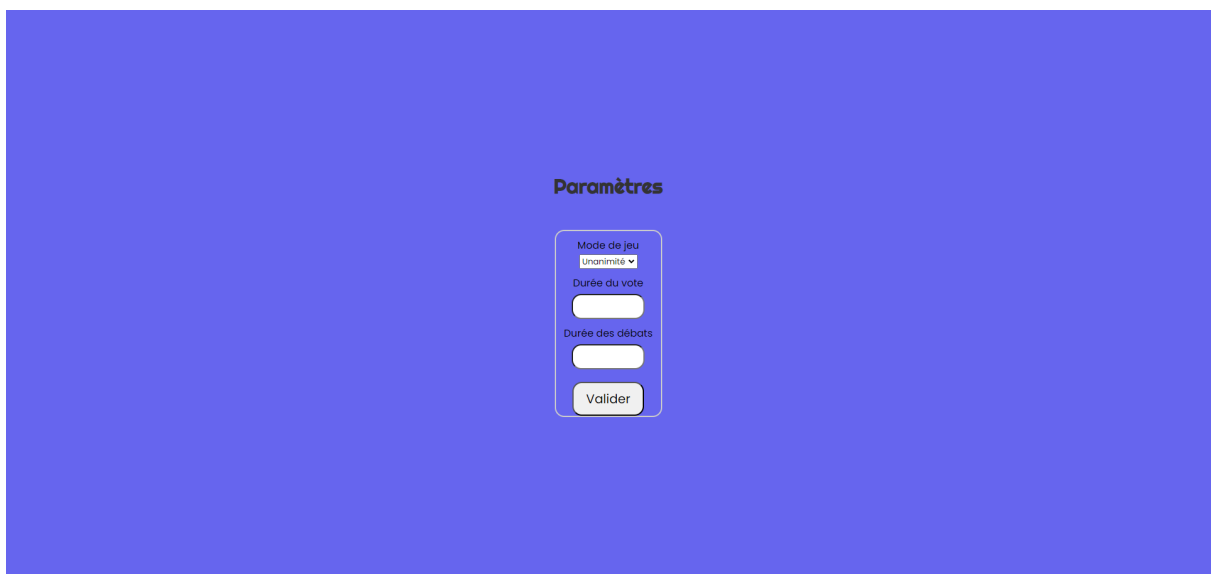
Index :

Cette page sert de page d'accueil. Elle ne contient qu'un seul bouton permettant d'accéder aux paramètres du jeu.



Setup :

Cette page permet de paramétrer le jeu. L'utilisateur peut choisir le mode de jeu qu'il souhaite : unanimité ou majorité. Elle permet aussi de paramétrer le chronomètre pour déterminer le temps imparti pour chaque joueur lors de son tour et le temps imparti pour débattre.



Pseudo :

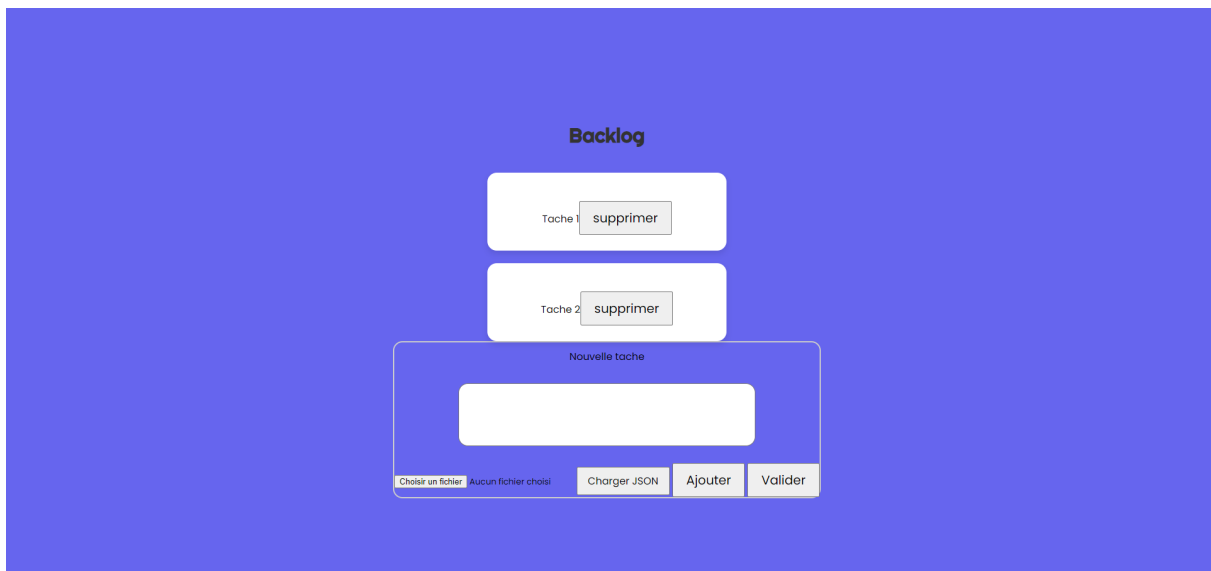
Cette page permet à l'utilisateur de rentrer les pseudos de chaque joueur. Elle affiche les pseudos enregistrés et permet à l'utilisateur de les supprimer. Il peut enregistrer jusqu'à 10 joueurs.



The screenshot shows a web interface with a blue background. At the top, the text "Veuillez entrer vos pseudos" is displayed. Below it is a text input field followed by an "Ajouter" button. In the center, there is a white box titled "Joueurs" containing a list of three players: "Pierre", "Paul", and "Jacques". Each name is followed by a "Supprimer" button. At the bottom of the white box is a "Continuer" button.

Backlog-entry :

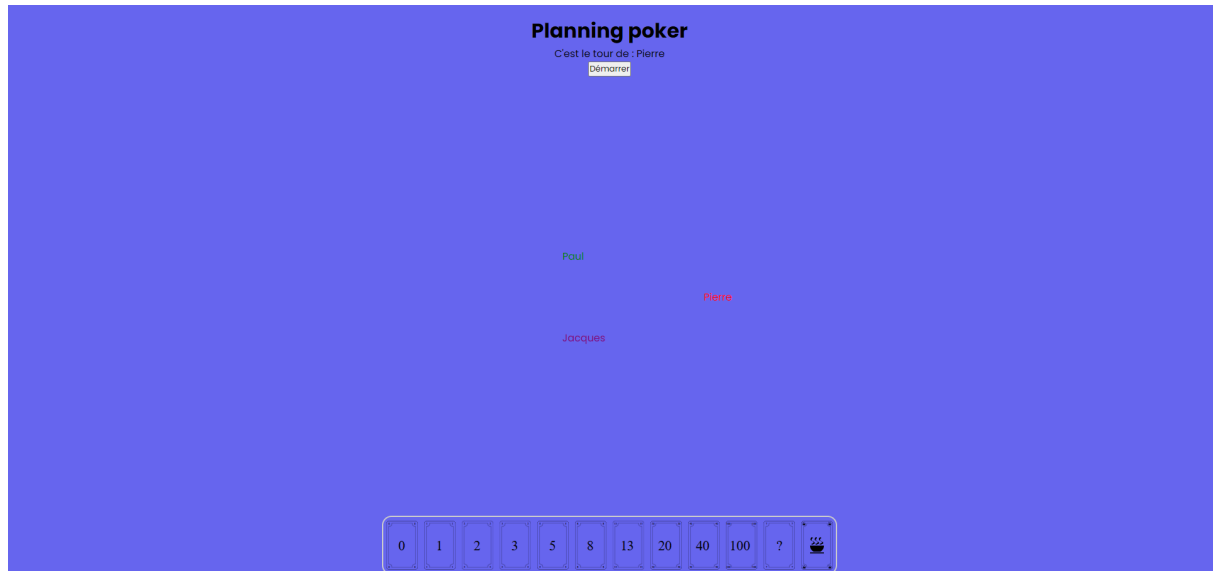
Cette page permet d'enregistrer les backlogs à évaluer par les joueurs. L'utilisateur peut rentrer les backlogs grâce au formulaire ou charger un fichier JSON contenant une liste de backlogs.



The screenshot shows a web interface with a blue background. At the top, the text "Backlog" is displayed. Below it, there are two white boxes, each containing a task name ("Tache 1" and "Tache 2") and a "supprimer" button. Below these is a larger white box labeled "Nouvelle tache" containing a text input field. At the bottom, there is a row of buttons: "Choisir un fichier", "Aucun fichier choisi", "Charger JSON", "Ajouter", and "Valider".

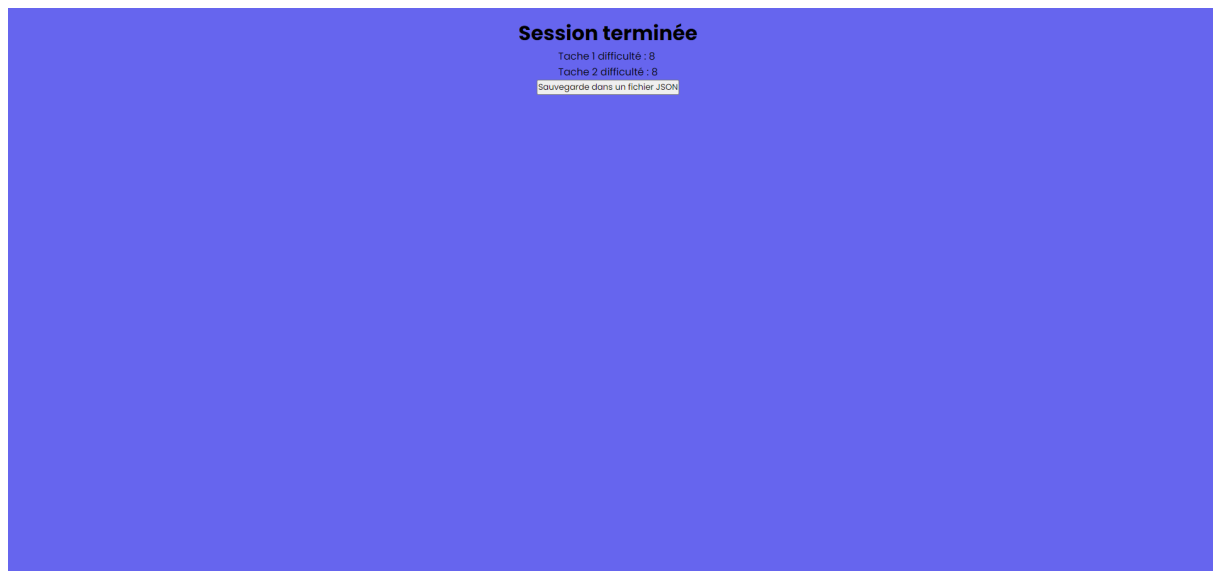
Jeu :

Cette page contient le jeu en lui-même. Les joueurs choisissent à tour de rôle une carte pour définir la difficulté de la tâche. Quand tous les joueurs ont voté, selon le mode de jeu choisi, le vote est relancé ou une autre tâche est débattue.



Fin-Jeu :

Enfin, cette page affiche les résultats du jeu et permet à l'utilisateur de les sauvegarder dans un fichier JSON téléchargeable.



Intégration continue

Nous voulions effectuer des tests unitaires pour chaque fonction contenues dans notre code avec la librairie Jest. Cependant, nous nous sommes rendus compte que la majorité de nos fonctions ne servent qu'à afficher des éléments sur la page HTML. Nous ne trouvions donc pas pertinent de les effectuer.

Cependant, pour les fonctions retournant des valeurs numériques, les tests effectués nous renvoient des résultats contradictoires. Nous avons ainsi décidé de ne pas les intégrer dans le rendu final mais ils sont tout de même disponibles dans la branche unittest.

Conclusion

Ce projet nous a permis de mettre en pratique les notions vues en cours. Les méthodes agiles permettent de développer un projet complexe plus efficacement et plus rapidement tout en optimisant le travail collaboratif.

Il est à noter que nous ne nous sommes pas du tout penché sur la sécurité et les éventuelles faiblesses de notre code, nous nous sommes concentré sur le fait que le jeu soit fonctionnel.

Bien que certains aspects n'aient pas été techniquement satisfaisants, la plupart des fonctionnalités demandées sont présentes et fonctionnelles. Cette première expérience nous permettra de nous familiariser avec les méthodes utilisées dans le monde professionnel et de développer notre travail d'équipe.