

Rapport du projet

Conception Agile - Application Planning Poker M1 Informatique

Binômes : Saad Soraya & Alpaslan Atasoy (groupe: TD004)

1. Introduction et Contexte

Ce projet consiste à développer une application de **Planning Poker**, utilisée en méthodologie Agile pour estimer collectivement la difficulté de fonctionnalités d'un backlog. L'objectif est de permettre à plusieurs joueurs de voter sur des tâches, de gérer les désaccords via plusieurs tours de vote, puis d'aboutir à une estimation finale enregistrée dans un fichier JSON.

L'application a été réalisée sous forme d'un site web fonctionnant **entièvement en local**, afin de rester simple à exécuter sur la machine de l'évaluateur et de se concentrer sur la logique du Planning Poker (gestion des tours, règles de validation, import/export JSON).

Le projet a été réalisé en binôme. L'organisation du travail s'est appuyée sur des séances de **pair programming** : une personne joue le rôle de "driver" (écrit le code) tandis que l'autre joue le rôle de "navigator" (relit, propose des améliorations, vérifie la cohérence avec les règles et les tests). Cette approche a permis de limiter les erreurs et de valider régulièrement le bon fonctionnement des fonctionnalités au fur et à mesure de l'avancement.

2. Choix techniques et architecture

L'application a été développée en **technologies web natives** (HTML, CSS et JavaScript), sans framework ni backend. Ce choix permet une exécution **entièvement locale**, simple à lancer sur la machine de l'évaluateur, tout en étant adaptée à la logique du Planning Poker.

Le projet est organisé de manière claire :

- `index.html` contient la structure des écrans,
- `style.css` gère l'apparence,
- `main.js` contient la logique de l'application,
- un fichier JSON peut être utilisé pour fournir un backlog de fonctionnalités.

L'application fonctionne comme une **single-page application simple** : un seul fichier HTML contient plusieurs écrans affichés ou masqués dynamiquement selon l'état de la partie. Le déroulement suit un flux linéaire (menu, configuration, vote, fin de partie), ce qui correspond naturellement au fonctionnement d'une session de Planning Poker.

La gestion de l'état repose sur un objet central (`appState`) qui stocke les informations essentielles (joueurs, mode de jeu, backlog, fonctionnalité en cours, votes). Cette approche facilite le passage entre les écrans, la gestion des tours de vote et la sauvegarde/reprise d'une partie via des fichiers JSON.

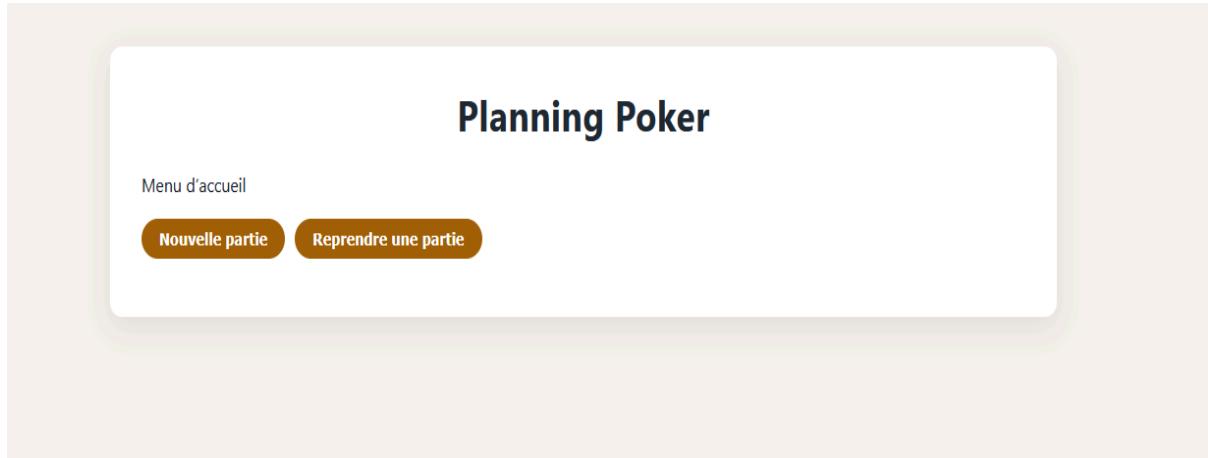
3. Manuel Utilisateur et Fonctionnalités

L'application fonctionne entièrement en local et se lance simplement en ouvrant le fichier `index.html` dans un navigateur web. Depuis le menu d'accueil, l'utilisateur peut démarrer une nouvelle partie ou reprendre une partie précédemment mise en pause à partir d'un fichier JSON.

Lors d'une nouvelle partie, les joueurs configurent la session en choisissant le nombre de participants, leurs pseudos, ainsi que le mode de jeu. Un backlog de fonctionnalités peut être importé au format JSON. Les fonctionnalités sont ensuite affichées une par une et les joueurs votent à l'aide des cartes de Planning Poker. En cas de désaccord, plusieurs tours de vote peuvent être réalisés jusqu'à la validation de l'estimation selon les règles choisies.

Si tous les joueurs sélectionnent la carte café, la partie est mise en pause et l'état courant est sauvegardé dans un fichier `pause.json`. À la fin de la partie, l'application affiche un récapitulatif des estimations et permet de télécharger un fichier JSON final contenant le backlog estimé.

Quelques captures d'écran expliquant le déroulement de la partie (backlog: constructions des rails d'un train “à titre d'exemple”)



Configuration de la partie

Nombre de joueurs :

Pseudo du joueur 1

Pseudo du joueur 2

Mode de jeu :

Backlog (JSON) :

Si aucun fichier n'est choisi, un backlog par défaut sera utilisé.

Démarrer la partie

Activer Windows
Accédez aux paramètres pour

Fonctionnalité à estimer

Conception des rails

Définir le tracé et les matériaux.

Statut : pending

[Passer au vote](#)

Mode : average — Joueurs : 1, 2

Vote Planning Poker

Joueur : 1

Fonctionnalité : Conception des rails

Tour : 1

[1](#) [2](#) [3](#) [5](#) [8](#) [13](#) [café](#)

[Valider le vote](#)

Fin de partie

Résumé des estimations

Conception des rails : 2

Fabrication des wagons : 2

[Télécharger le JSON final](#)

[Retour à l'accueil](#)