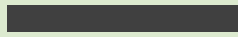


CONCEPTION AGILE



# Rapport Planning Poker



PRÉPARÉ PAR

Enzo Causse  
Said Aitahmad  
Mariyam Agjil



# Introduction

## 1. Contexte :

- Introduction à l'application de planning poker local.
- Description du mode de jeu où les joueurs utilisent le même ordinateur tour à tour.

## 2. Objectif de l'Application :

- Permettre aux équipes de planifier leurs projets de manière interactive à l'aide du planning poker.
- Faciliter le processus de vote pour estimer la complexité des fonctionnalités.



# Analyse des besoins

---

- **Identification des Utilisateurs :**

- **Les joueurs :** Personnes participant à la partie de planning poker.

- **Fonctionnalités Principales :**

- **Menu de Configuration :**

- Définir le nombre de joueurs.
- Entrer un pseudo pour chaque joueur.
- Choix des règles de planning poker (Moyenne, Médiane, Majorité absolue, Majorité relative).

- **Processus de Vote :**

- Chaque joueur vote pour l'estimation de la fonctionnalité.
- Validation ou non en fonction des règles sélectionnées.

- **Rejouer le Vote :**

- En cas de non-validation, possibilité de rejouer le vote pour la fonctionnalité en question.



# Analyse des besoins

- **Structure JSON pour les Paramètres :**
  - **Clefs :** Identifiants des paramètres.
  - **Valeurs :** Options choisies par les utilisateurs (nombre de joueurs, pseudos, règles de planning poker).
- **Validation Automatique :**
  - Mécanisme d'automatisation pour valider ou non une fonctionnalité en fonction des règles sélectionnées.
- **Enregistrement des Estimations :**
  - Sauvegarde des estimations finales dans un fichier JSON.
- **Interface Utilisateur (UI) Intuitive :**
  - Interface conviviale pour faciliter la configuration et le vote.
- **Adaptabilité aux Différents Scénarios :**
  - Flexibilité pour s'adapter aux variations du nombre de joueurs.



# Cas d'utilisation

## CAS D'UTILISATION : CONFIGURATION DE LA PARTIE

- **Acteurs Principaux :** Utilisateur
- **Description :** Le joueur souhaite configurer une nouvelle partie de planning poker.
- **Préconditions :** L'application est ouverte, et aucune partie n'est en cours.
- **Scénario Principal :**
  - Le joueur ouvre l'application.
  - Il sélectionne l'option "Nouvelle Partie".
  - Le menu de configuration s'affiche.
  - Le joueur entre le nombre de participants.
  - Il saisit un pseudo pour chaque joueur.
  - Il choisit la règle de planning poker parmi les options disponibles (Moyenne, Médiane, Majorité absolue, Majorité relative).
- **Extensions :**
  - Si le joueur souhaite annuler la configuration, il peut sélectionner "Annuler" à tout moment.
  - Si le joueur entre un nombre de participants invalide, l'application émet un avertissement.
- **Postconditions :** La configuration est enregistrée, et la partie est prête à commencer.



# Cas d'utilisation

## CAS D'UTILISATION: VOTE POUR UNE TÂCHE

- **Acteurs principaux:** Utilisateur, Système
- **Description :** Les joueurs votent pour estimer la complexité d'une tâche dans le cadre d'une partie de planning poker.
- **Préconditions :** Une partie de planning poker est configurée, et une tâche est prête à être estimée.
- **Scénario Principal :**
  - Le système affiche la tâche à estimer.
  - Chaque joueur attribue une estimation en fonction de la règle de planning poker sélectionnée.
  - Le système recueille les votes de chaque joueur.
  - Si la règle nécessite une discussion, les joueurs peuvent échanger des points de vue.
  - Le système enregistre les estimations.
- **Extensions :**
  - Si la règle sélectionnée nécessite une revote en cas de non-validation, le système propose de rejouer le vote.
  - Si les joueurs ne parviennent pas à un consensus après la discussion, une nouvelle itération de vote peut être envisagée.
- **Postconditions :**
  - Les estimations de la tâche sont enregistrées.
  - L'application passe à la tâche suivante si elle est configurée, sinon elle termine la partie.



# Cas d'utilisation

## CAS D'UTILISATION : REJOUER LE VOTE POUR UNE TÂCHE

- **Acteurs Principaux :** Utilisateur, Système
- **Description :** Les joueurs décident de rejouer le vote pour une tâche déjà estimée, soit en raison d'un désaccord persistant, soit après une discussion approfondie.
- **Préconditions :** Une tâche a été précédemment estimée, mais le consensus n'a pas été atteint ou les joueurs souhaitent réévaluer l'estimation après une discussion.
- **Scénario Principal :**
  - Les joueurs décident de rejouer le vote pour la tâche spécifiée.
  - Le système réinitialise les estimations de la tâche.
  - Les joueurs revotent en prenant en compte les discussions antérieures ou tout nouvel élément d'information.
  - Le système recueille les nouveaux votes.
  - Si nécessaire, les joueurs peuvent engager une nouvelle discussion avant de voter à nouveau.
- **Extensions :**
  - Si les joueurs ne parviennent toujours pas à un consensus après le nouveau vote, une nouvelle itération de discussion et de vote peut être envisagée.
- **Postconditions :**
  - Les nouvelles estimations de la tâche sont enregistrées.
  - L'application peut passer à la tâche suivante si elle est configurée, sinon elle termine la partie.



# Objectif de l'application

L'application de planning poker locale vise à faciliter et à optimiser le processus de planification des projets au sein des équipes de développement. Les objectifs principaux de l'application sont les suivants :

- Offrir un environnement interactif et convivial pour la planification agile des projets.
- Permettre aux équipes de collaborer de manière efficace pour estimer la complexité des tâches à réaliser.
- Permettre aux utilisateurs de personnaliser les règles en fonction des besoins spécifiques de leur équipe.
- Soutenir les équipes locales qui utilisent un seul ordinateur, où chaque joueur prend le contrôle à tour de rôle.
- Sauvegarder les résultats des estimations de manière structurée dans un format JSON.
- Minimiser les frictions et simplifier les interactions pour une expérience utilisateur positive.
- Faciliter l'accès ultérieur aux données d'estimation pour une analyse rétrospective ou un suivi.





# Diagramme de classe

Les éléments à inclure :

- **Classe Principale:** Menu
- **Attributs:** Lancer, Options, Quitter
- **Classe:** Options
- **Attributs:** Temps du tour (en secondes), Mode de jeu
- **Relations:** Contient les sous-classes "Nom du joueur", "Nom de la tâche", et "Mode de jeu"

Sous-classe de Options: **Nom du joueur**

- **Attributs:** Nom, Ajouter, Supprimer

Sous-classe de Options: **Nom de la tâche**

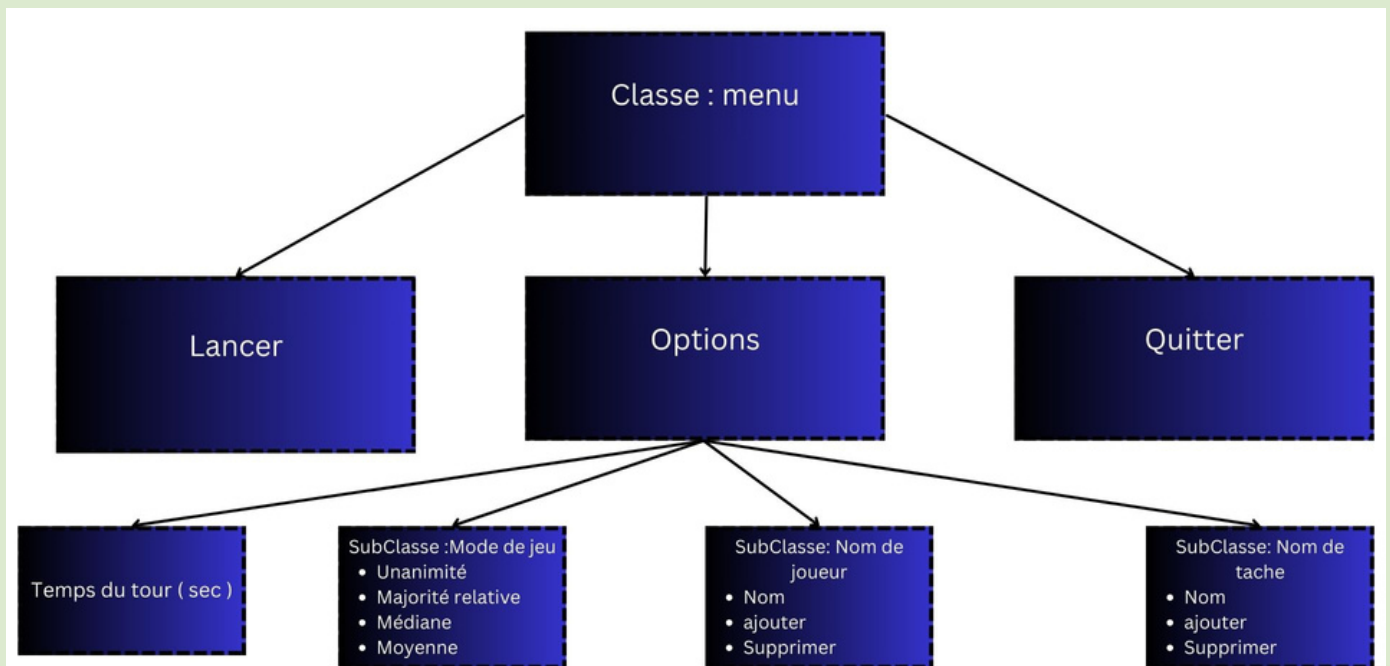
- **Attributs:** Nom, Ajouter, Supprimer

Sous-classe de Options: **Mode de jeu**

- **Attributs:** Unanimité, Majorité relative, Médiane, Moyenne



# Diagramme de classe



# Architecture de l'application: Les interfaces

## ARCHITECTURE TECHNIQUE AVEC PYGAME ET PYTHON :

- **Interface Utilisateur (UI) avec Pygame :**

Nous avons utilisé la bibliothèque Pygame pour créer les interfaces graphiques de l'application car elle excelle dans la gestion des événements, l'affichage des éléments de l'interface utilisateur et l'interaction avec les utilisateurs.

- **Backend avec Python :**

Python a été sélectionné pour sa polyvalence dans la logique métier et la gestion des données. Les règles du planning poker et les mécanismes de vote ont été implémentés à l'aide de Python.

- **Gestion des Données avec JSON :**

Nous avons opté pour l'utilisation de JSON afin de stocker les données de configuration, le backlog des tâches, et les estimations dans des fichiers au format JSON. Cette approche s'appuie sur la facilité d'utilisation de la bibliothèque JSON de Python, tant pour la lecture que pour l'écriture des données.



# Architecture de l'application: Les interfaces

## AVANTAGES DE L'APPROCHE PYGAME ET PYTHON :

Les avantages de l'approche avec la bibliothèque Pygame et le langage Python sont :

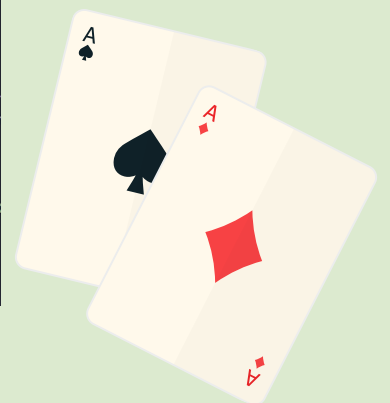
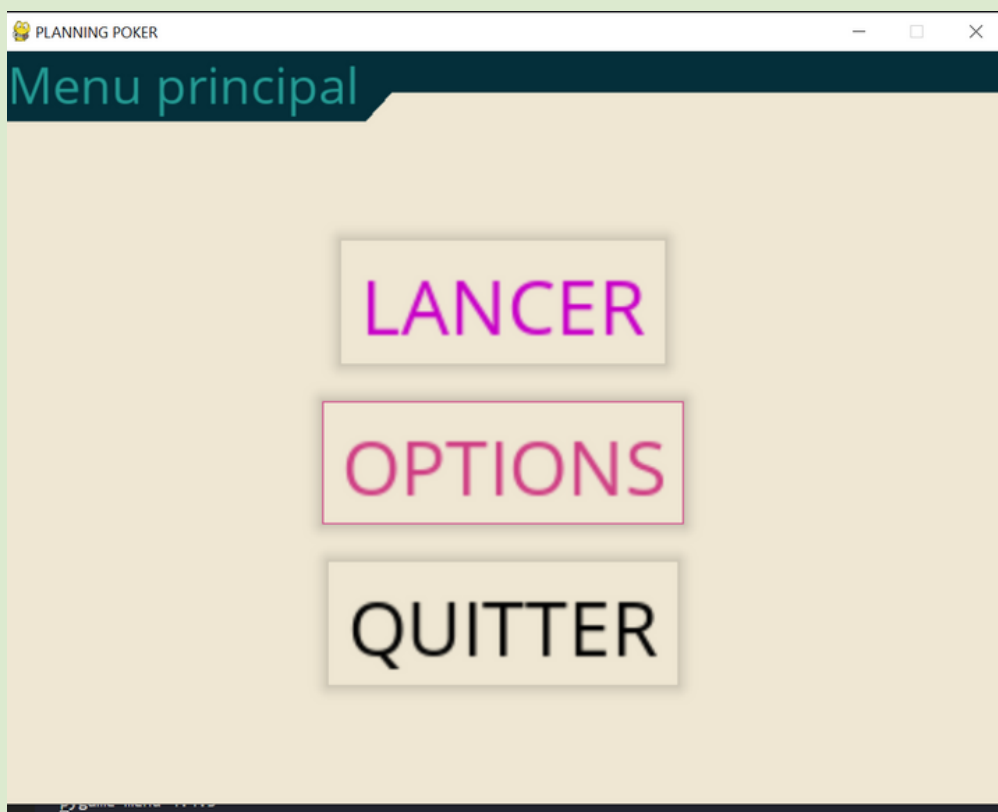
- Rapidité de Développement
- Facilité d'Utilisation de Python
- Adaptabilité et Évolutivité
- Stockage de Données Simple avec JSON
- Flexibilité de Pygame pour l'Interface Utilisateur



# Architecture de l'application: Les interfaces

## LES INTERFACES:

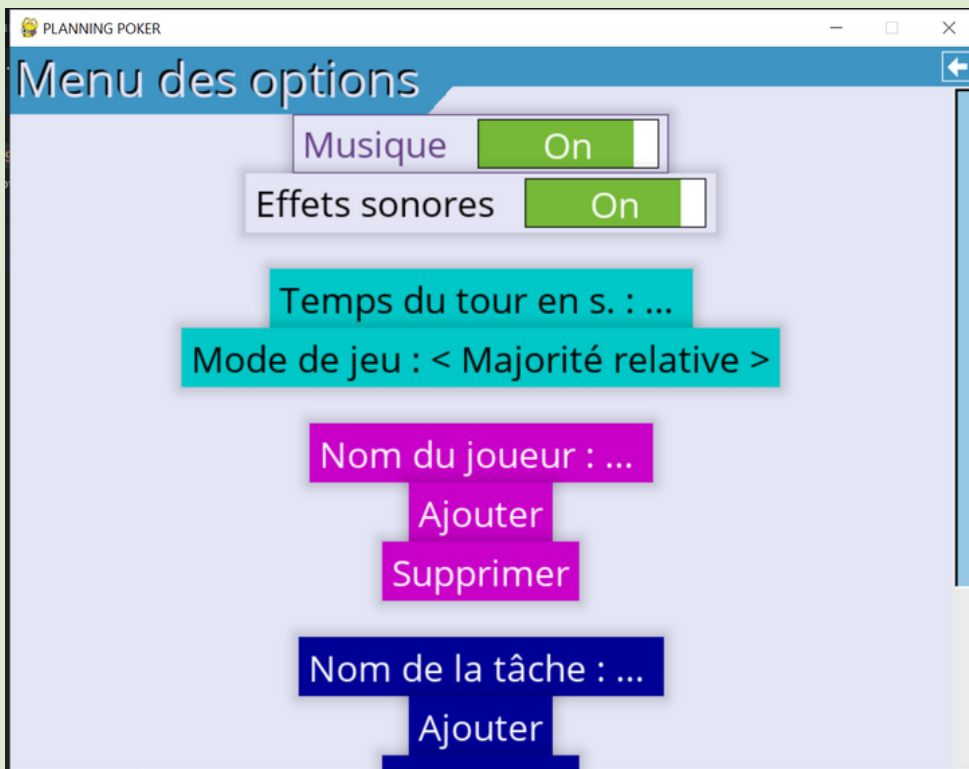
On a 3 boutons : Lancer, Options, Quitter.  
Premièrement, il faut configurer au moins une option.  
Ensuite, on lance le jeu, et si on souhaite quitter le jeu, on utilise le bouton Quitter.



# Architecture de l'application: Les interfaces

## LES INTERFACES:

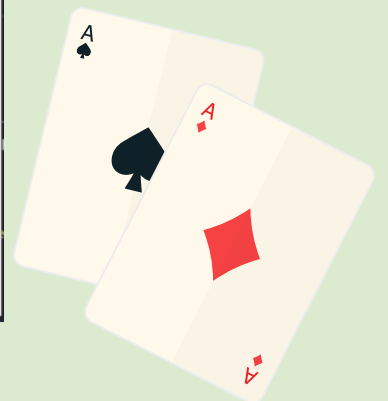
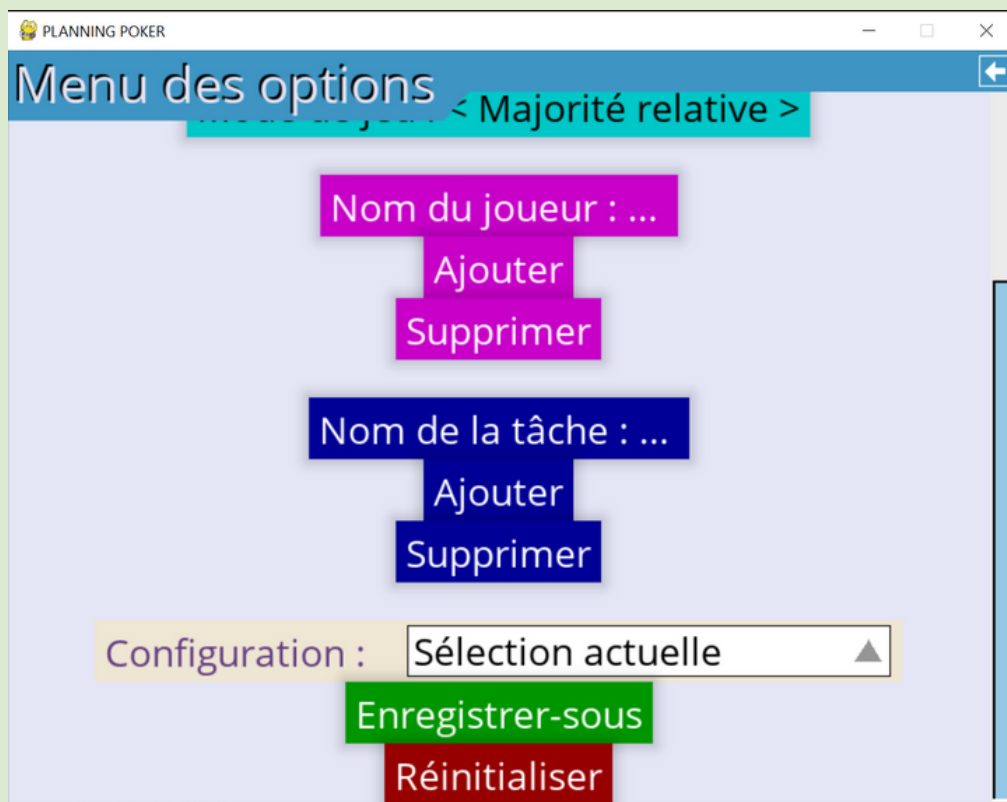
Pour débiter, il convient de saisir la durée du tour suivie du nom du joueur. Ensuite, on renseigne le nom de la tâche.



# Architecture de l'application: Les interfaces

## LES INTERFACES:

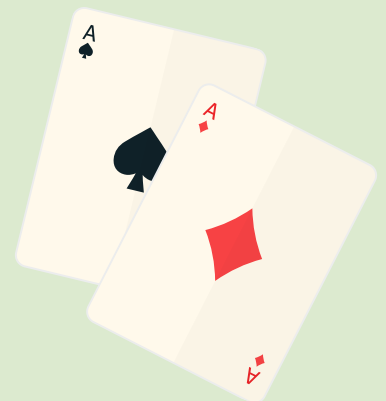
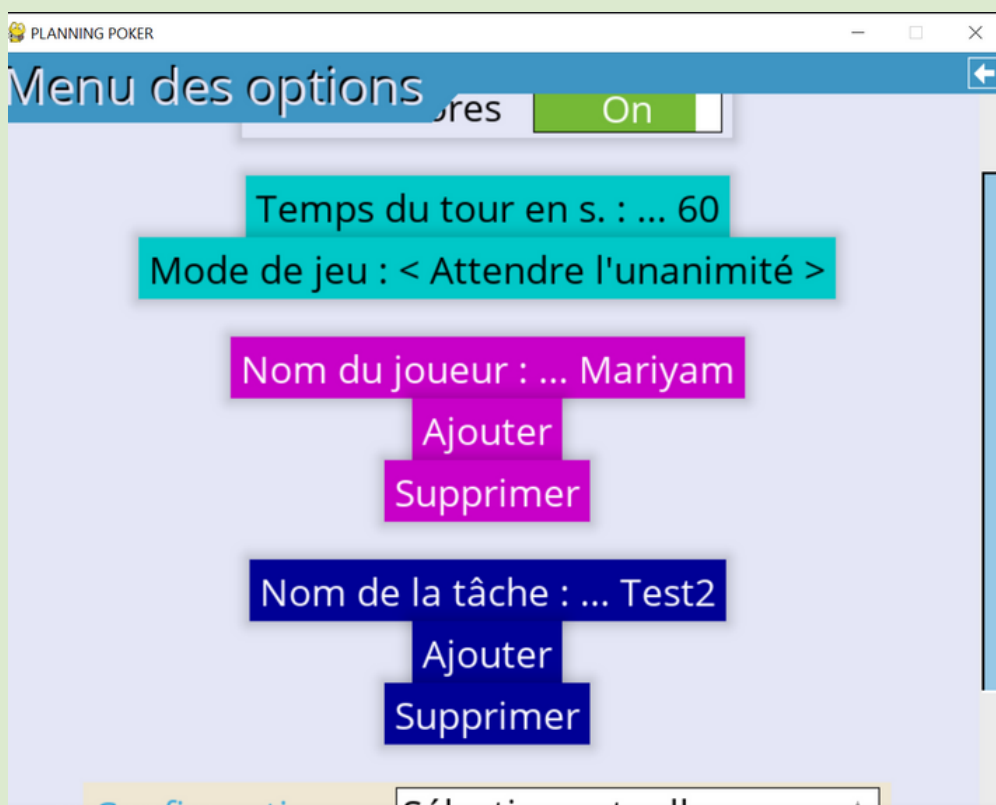
Puis on clique sur "Enregistrer sous". Alternativement, on peut réinitialiser en utilisant le bouton "Réinitialiser". Avec le bouton "Configuration", on a la possibilité de choisir la sélection actuelle ou une sélection déjà enregistrée.



# Architecture de l'application: Les interfaces

## LES INTERFACES:

Maintenant, nous effectuons un bref test. Nous avons sélectionné une durée de tour de 60 secondes. Pour le nom du joueur, nous avons saisi "Mariyam", et le nom de la tâche est "Test 2". Après avoir tout enregistré, nous sommes revenus en utilisant la flèche tout au fond à droite pour enfin cliquer sur "Lancer".

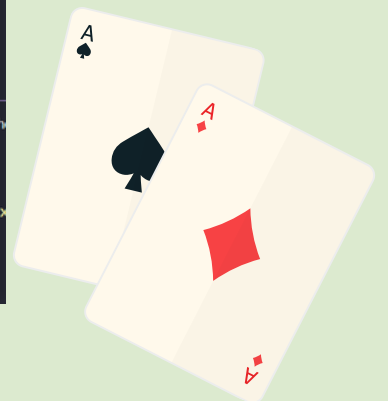
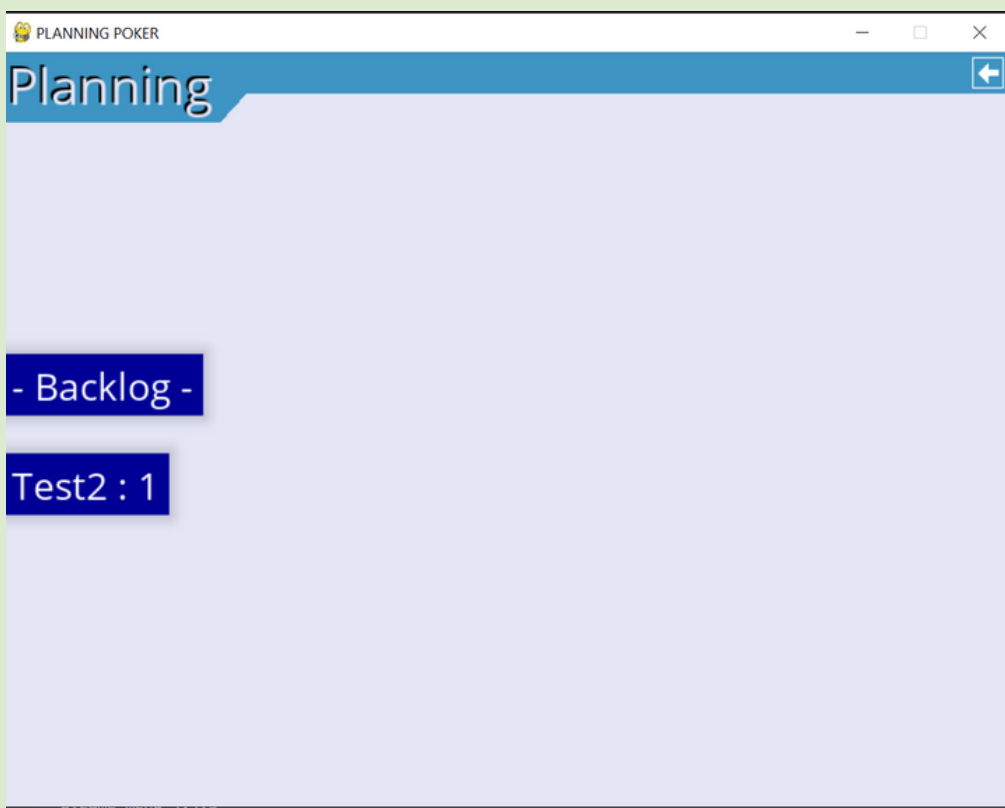




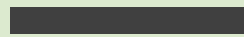
# Architecture de l'application: Les interfaces

## LES INTERFACES:

Et voici ce que nous visualisons : le nom de la tâche et le nombre de personnes qui ont voté. Dans ce cas, il n'y a qu'une seule personne, car il s'agit simplement d'un test pour vérifier le bon fonctionnement de l'application.

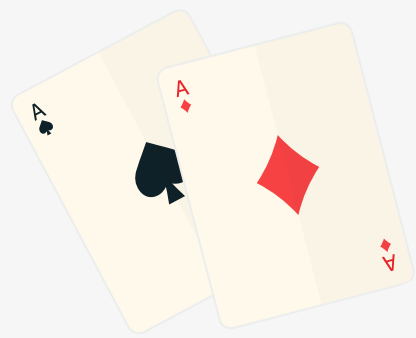


# Gestion de projet :



Pour le projet, il nous a fallu environ quatre semaines pour le mener à bien. Enzo Causse s'est principalement occupé de la création du prototype du jeu, Said Aitahmad a pris en charge le développement des interfaces, et Mariyam Agjil s'est concentrée sur la conception du projet ainsi que sur la rédaction du rapport. Il est important de souligner que nous avons mis en place un processus collaboratif, discutant et validant chaque étape tous les trois avant de passer à la suivante. Cette approche garantit que chacun de nous comprend et suit le travail accompli par les autres, favorisant ainsi une collaboration efficace.





# Conclusion

---

En résumé, le choix de Pygame et Python pour le développement de l'application de planning poker s'est avéré judicieux. Pygame a facilité la création d'interfaces graphiques interactives, tandis que Python a été utilisé de manière polyvalente pour la logique métier, la gestion des données, et l'interaction avec Pygame. L'utilisation de fichiers JSON pour le stockage des données offre une simplicité d'utilisation appréciable.

L'interface conviviale de l'application permet une configuration intuitive des tours, des joueurs et des tâches. Elle offre une expérience utilisateur complète en permettant le lancement du jeu, la configuration des options et la possibilité de quitter de manière simple et efficace.

