

# PLANNING POKER

## Présentation du projet

L'objectif de ce projet était de créer une application informatique conviviale qui reproduit le processus du Planning Poker tel qu'il est pratiqué traditionnellement, mais de manière numérique et accessible sur un ordinateur. Cette application offre plusieurs avantages. Les entreprises peuvent accéder à l'outil directement depuis leurs ordinateurs, éliminant ainsi la nécessité d'utiliser des cartes physiques ou des outils papier. L'application permet de centraliser les estimations de plusieurs membres de l'équipe, facilitant la collaboration et la prise de décision collective concernant les estimations de temps pour les tâches. L'outil peut être personnalisé pour répondre aux besoins spécifiques de l'équipe ou de l'entreprise, par exemple en permettant différents types de jeux de cartes ou en intégrant des fonctionnalités supplémentaires pour l'analyse des estimations. Le nombre de modes de jeu a été agrandi au maximum, il y en a 3, afin de pouvoir convenir au plus d'utilisateurs possible.

Pour réaliser ce projet, il est essentiel de concevoir une interface utilisateur intuitive, de garantir la sécurité des données et de développer des fonctionnalités de synchronisation en temps réel pour permettre une participation fluide et en temps réel des membres de l'équipe. C'est pour cela que nous lui avons donné une apparence ludique, avec comme thème "retro gaming" et "casino", avec des petites fonctionnalités amusantes, telles que : pouvoir changer le style des cartes, etc... Mais aussi des fonctionnalités utiles pour l'accessibilité, un des points les plus importants dans le développement de projets de nos jours. Cela se présente dans notre projet avec la possibilité de changer de langues, avec un choix de langues très grand et très varié. Il est également possible de sauvegarder la partie et de la reprendre plus tard en utilisant la carte Café.

# Design Pattern

## Décorateur

Afin de faire fonctionner notre projet, nous avons utilisé le Design Pattern du décorateur pour gérer les différents modes de jeu. Ainsi, nous avons une classe abstraite qui contient toutes les fonctions utiles au bon déroulement du jeu, classe qui est dérivée en Jeu concret qui contient les variables utiles au jeu et en Mode qui contient les fonctions pour le déroulement du jeu, ce Mode est la classe mère de tous les Modes que nous avons implémentés.

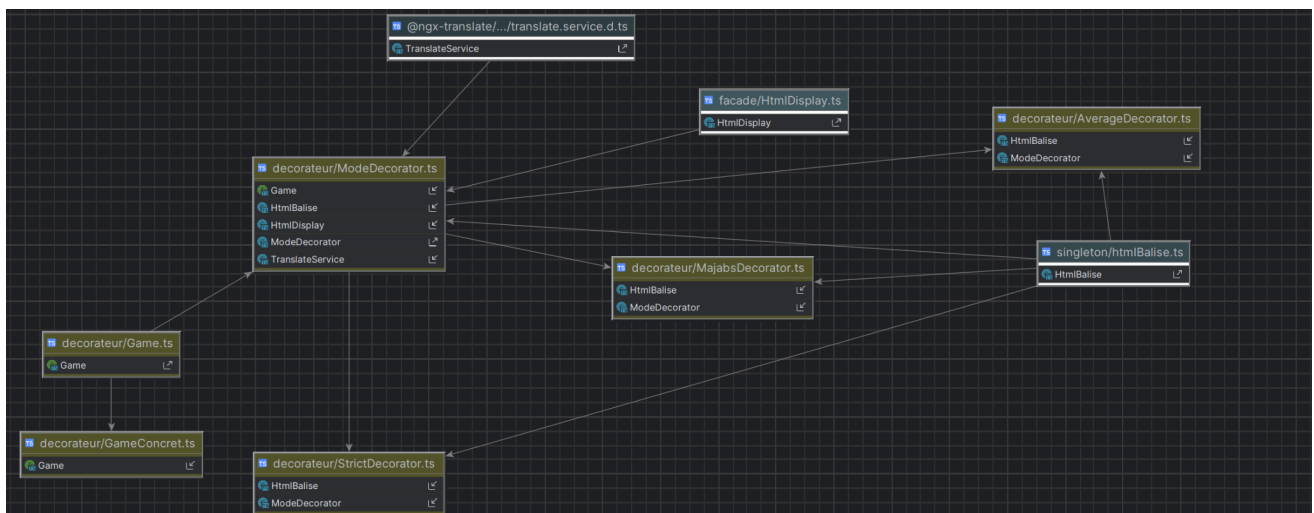
Pour simplifier l'implémentation de balise HTML via le TypeScript, nous avons utilisé le Design Pattern de la façade.

## Façade

Ce Design Pattern nous permet d'implémenter facilement des balises HTML directement depuis le TypeScript. Ainsi, en une ligne, nous pouvons implémenter une balise Input avec des classes, un identifiant ou d'autre variable intégré et pour les insérer facilement dans le code, nous avons utilisé le Design Paterne Singleton.

## Singleton

Ce Design Pattern n'ayant qu'une seule implémentation dans tout le code nous permet de récupérer facilement divers emplacements HTML de la page afin d'y placer du code facilement.



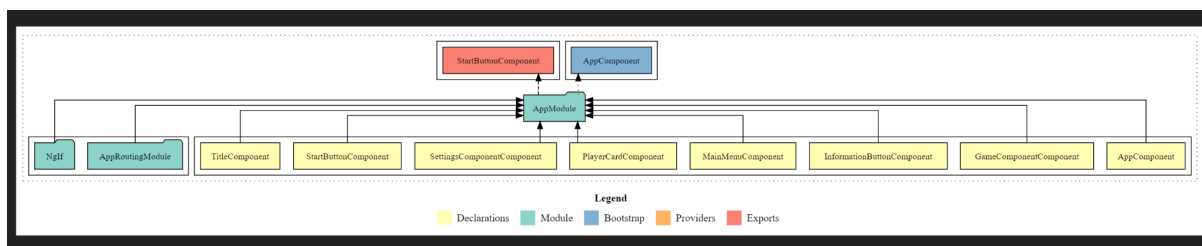
## Choix d'architecture

Nous avons choisi Angular, car ce framework est souvent choisi pour ses nombreuses fonctionnalités et son architecture robuste, offrant plusieurs avantages pour le développement de projets complexes tel un planning poker. En effet, son architecture, basée sur des composants, nous permet une organisation claire et modulaire de notre code. Les composants sont réutilisables, ce qui favorise la maintenabilité du projet tout au long du temps.

Angular est écrit en TypeScript, qui est un sur-ensemble de JavaScript. Le TypeScript ajoute une couche de typage statique au JavaScript, réduisant les erreurs potentielles pendant le développement et améliorant la lisibilité du code. Ce qui nous permet d'effectuer des tâches complexes plus simplement. Nous pouvons également noter qu'Angular offre un système de liaison de données bidirectionnelle, permettant une synchronisation automatique entre la vue et le modèle. Cela simplifie la gestion des données et améliore l'expérience utilisateur. En plus de proposer un modèle MVC simple et à la portée de tous.

Angular propose un ensemble complet d'outils pour faciliter le développement, y compris Angular CLI pour générer des composants, des services, des tests et pour gérer les dépendances, grâce au "npm install", qui permet d'installer toutes les dépendances nécessaires à la bonne version. Le framework intègre, également, des outils pour les tests unitaires et fonctionnels, favorisant une approche de développement axée sur les tests pour garantir la qualité du code.

Ces caractéristiques font d'Angular un choix solide pour les projets nécessitant une structure solide, une grande évolutivité et une architecture bien définie. C'est pourquoi nous l'avons choisi pour notre projet.



## Unit Test

Pour les tests unitaires, nous avons utilisé la bibliothèque Karma modifiée par Jasmine, qui nous permet de faire des tests sur Angular. Chaque Page est correctement testé dans le but de vérifier que l'ont peu importé et nous testons les différentes fonctions de notre jeu, principalement les Setters et Getters afin de vérifier leurs bons fonctionnements. Vous pouvez lancer les tests en lançant la commande ``ng test`` assurez-vous d'avoir Chrome d'installer avant de lancer la commande. Les outils Karma et Jasmine étaient déjà implémentés dans Angular, ce qui a rendu leurs implémentations faciles.

## Documentation

Pour la doc, nous utilisons "Compodoc", que nous pouvons utiliser en lançant la commande ``npm run compodoc`` qui vous génère la documentation et vous ouvrira un serveur sur lequel est présent toute la doc. Cette commande va automatiquement générer la documentation du projet, rassemblant des informations clés sur les différents composants, fonctions et structures utilisés dans le code. Une fois que la génération est terminée, un serveur local est lancé, hébergeant cette documentation. Cela signifie que vous pouvez accéder à toute la documentation générée en ouvrant votre navigateur et en visitant l'adresse spécifique à ce serveur. Cela offre une vue détaillée et organisée de l'architecture, des fonctionnalités et des détails techniques du projet, facilitant ainsi la compréhension et la collaboration pour toute l'équipe travaillant sur le Planning Poker.