

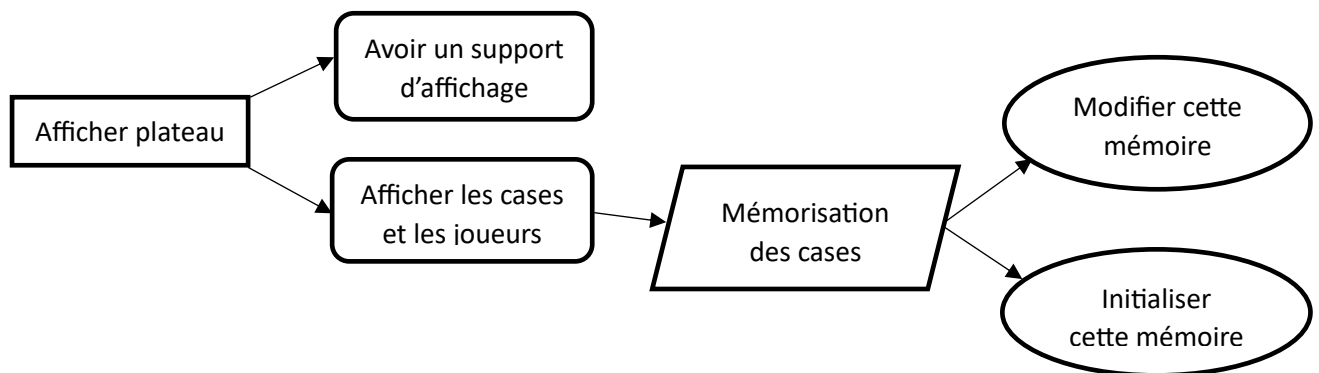
TOB – Projet long

Rapport personnel Itération 1

Frédéric Camail

Lors de cette itération mon travail était de répondre aux fonctionnalités liées au plateau. Pour cela j'ai construit la classe Plateau dont les réalisations on était simplifié mais qui interagie déjà complètement avec les autres classes afin qu'augmenté la complexité des fonctionnalités réalisées soit le plus simple possible. Pour rendre cela possible j'ai travaillé avec Emile pour que les méthodes et les types de nos classes s'accordent et Téo qui m'a présenté des outils plus complexes que ceux que j'utilisais mais plus efficace.

Voici les fonctionnalités auxquelles la classe répond avec les méthodes qui leurs correspondent et une explication succincte sur leurs implémentations.



Avoir un support d'affichage : Permet de séparer les cases entre elles. **Plateau** possède un attribut **plateau** qui est un **string buffer**. Dans la méthode **afficher**, il est initialisé avec des tirets qui sépareront les cases venant s'intercaler au milieu grâce à la fonction afficher de case.

Afficher les cases et les joueurs : Modifie le **plateau** afin d'y insérer les cases et les joueurs. Dans la méthode afficher de **plateau** on appelle **afficher** de la classe **CaseLibre** sur toutes les cases et on modifie le buffer pour placer tous les joueurs.

Mémorisation des cases : **Plateau** possède deux listes en attribut **CaseGraphique** et **CaseFonctionnelle** qui garde en mémoire ces cases avant de les afficher.

Modifier cette mémoire : **changerCaseGraphique** et **changerCaseFonctionnelle** utilisent simplement les méthodes de **ArrayList** pour modifier les listes.

Initialiser cette mémoire : Le constructeur de **Plateau** va créer le nombre souhaité de cases libres pour le plateau et va les placer dans les listes. Cela est possible car **CasesLibre** implémente les cases fonctionnelle et graphique.

Légende : **Classe** / **Méthode** / **Attribut** / **Type**