**EXPLICATION DU PROJET**

**Décomposition des Fichiers**

**Cellule.java**: Cette classe représente probablement une cellule individuelle du plateau de jeu. Elle pourrait contenir des informations sur l'état de la cellule (vide, occupée par un jeton), ses coordonnées, etc.

* **CelluleGraphi.java**: Cette classe semble être la représentation graphique d'une cellule. Elle pourrait gérer l'affichage de la cellule, sa couleur en fonction du joueur, etc.
* **Grille.java**: Cette classe représente la grille de jeu complète. Elle contient probablement un tableau de cellules et gère les opérations sur la grille, comme vérifier si une colonne est pleine, si un joueur a gagné, etc.
* **Jeton.java**: Cette classe représente un jeton de jeu. Elle pourrait contenir des informations sur la couleur du jeton, le joueur qui l'a posé, etc.
* **Partie.java**: Cette classe pourrait gérer le déroulement d'une partie, comme l'alternance des tours, la vérification des conditions de victoire, etc.
* **Player.java**: Cette classe représente un joueur. Elle pourrait contenir des informations sur le nom du joueur, sa couleur, etc.
* **que.java**: Ce fichier pourrait contenir du code spécifique à une question ou une fonctionnalité particulière du jeu, mais sans plus de contexte, il est difficile de le déterminer avec précision.

**Structure Probable du Projet**

En se basant sur ces noms de fichiers, on peut imaginer une structure de classes et de relations suivante :

* **Classe Jeu**:
  + Contient une instance de Grille.
  + Gère les joueurs.
  + Gère le déroulement d'une partie (alternance des tours, vérification des conditions de victoire, etc.).
  + Interagit avec l'interface graphique (Jeu.form).
* **Classe Grille**:
  + Contient un tableau de cellules.
  + Gère les opérations sur la grille (ajouter un jeton, vérifier si une colonne est pleine, vérifier les alignements, etc.).
* **Classe Cellule**:
  + Représente une cellule individuelle de la grille.
  + Contient des informations sur l'état de la cellule.
* **Classe Jeton**:
  + Représente un jeton de jeu.
* **Classe Player**:
  + Représente un joueur.

**MCD**

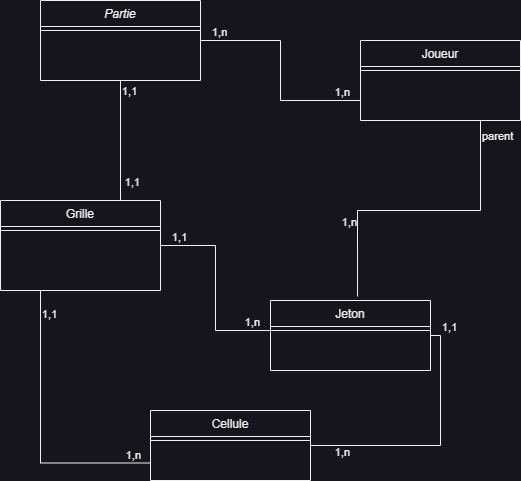
**Relations :**

* **Une partie est jouée par plusieurs joueurs.**
* **Une partie se déroule sur une grille.**
* **Une grille est composée de cellules.**
* **Une cellule contient un jeton.**
* **Un joueur peut avoir plusieurs jetons**
* **Une cellule contient un seul jeton**
* **La grille contient plusieurs cellules et donc plusieurs jetons**

**Optimisation du Projet**

Pour optimiser le projet, voici quelques idées simples :

* **Organisation** : Séparer les classes en groupes logiques comme les classes pour la grille, les joueurs et l'interface graphique.
* **Performance** : Optimiser les vérifications pour la victoire, par exemple, ne regarder que les zones concernées après un coup.
* **Interface** : Ajouter des couleurs pour différencier les joueurs et rendre le jeu plus joli.
* **Tests** : Vérifier que chaque partie du projet marche bien avec des tests simples.
* **Personnalisation** : Laisser les joueurs choisir leurs noms et couleurs avant de jouer.

****

**Build le projet dans Netbeans : src/superpuissance4/Jeu.java**