


Diagramme de cas d'utilisation

flowchart LR

Joueur[ Joueur] --- Puissance([Jouer au jeu de Puissance 4])

Scénario

UC : Jouer au Puissance 4

Périmètre : Le jeu

Niveau : But utilisateur

Acteur principal : Joueur

Pré-condition : Aucune

Post-conditions : Aucune

Scénario nominal

1. L'utilisateur lance le programme
2. Le système demande le nom du joueur 1
3. Le joueur 1 répond
4. Le système enregistre le nom du joueur 1
5. Le système demande au joueur 2
6. Le joueur 2 répond
7. Le système enregistre le nom du joueur 2
8. Le système affiche le plateau ainsi que le nom du joueur qui doit jouer
9. Le joueur choisi dans quelle colonne il souhaite mettre un jeton
10. Le système vérifie si la partie est gagnée
11. Le système affiche le vainqueur et demande si l'utilisateur souhaite refaire une partie ("y" or "n")
12. L'utilisateur répond "y" ou "n"
13. Le système retourne au point 8 du scénario nominal et incrémente le nombre de partie

Extensions

- 3.a L'utilisateur répond à la question avec une chaîne vide
 1. Retour au point 2 du scénario nominal
- 6.a L'utilisateur répond à la question avec une chaîne vide
 1. Retour au point 5 du scénario nominal
- 9.a L'utilisateur choisi un nombre n $n \notin \{x \in \mathbb{R}, \text{ tq } 0 < x < 7\}$
 1. Le système affiche un message d'erreur
 2. retour au point 8

- 10.a La partie n'est pas finie
 1. Le systeme change le joueur qui doit jouer
 2. retour au point 8 du scénario nominal
- 12.a l'utilisateur répond autre chose que 'y' ou 'n'
 1. le systeme affiche un message d'erreur
 2. retour au point 12 du scénario nominal
- 12.a l'utilisateur répond 'n'
 1. le systeme affiche le gagnant et le nombre de victoire du gagnant
 2. fin du programme
- 12.a.1.a il n'y a pas de gagnant
 1. le systeme affiche ex aequo et le nombre de victoire des joueurs (il est identique pour les deux)
 2. fin du programme

Modèle de Conception

Puissance 4 (Itération 2)

```
classDiagram
    class main {

    }

    class Ihm {
        - Scanner scanner

        + Ihm()
        + demanderNomJoueur(int numJoueur) String
        + demanderCoupPuissance4(String nomJoueur) int[]
        + demanderJouerEncore() bool
        + afficherPlateau(String plateauString) void
        + afficherVictoire(String nomJoueur, int nbVictoires, int nbParties) void
        + afficherErreur(String message) void
    }

    class ControleurPuissance4 {
        - int numeroJoueurCourant
        - int nbParties

        + ControleurPuissance4(Ihm ihm)
        + jouer () void
        - commencerPartie() void
        - toursDeJeu() void
        - tourSuivant() void
        - finPartie() void
    }
```

```
+ getNumeroJoueurCourant() int
+ getJoueurCourant() Joueur
+ getNomJoueurCourant() String
}

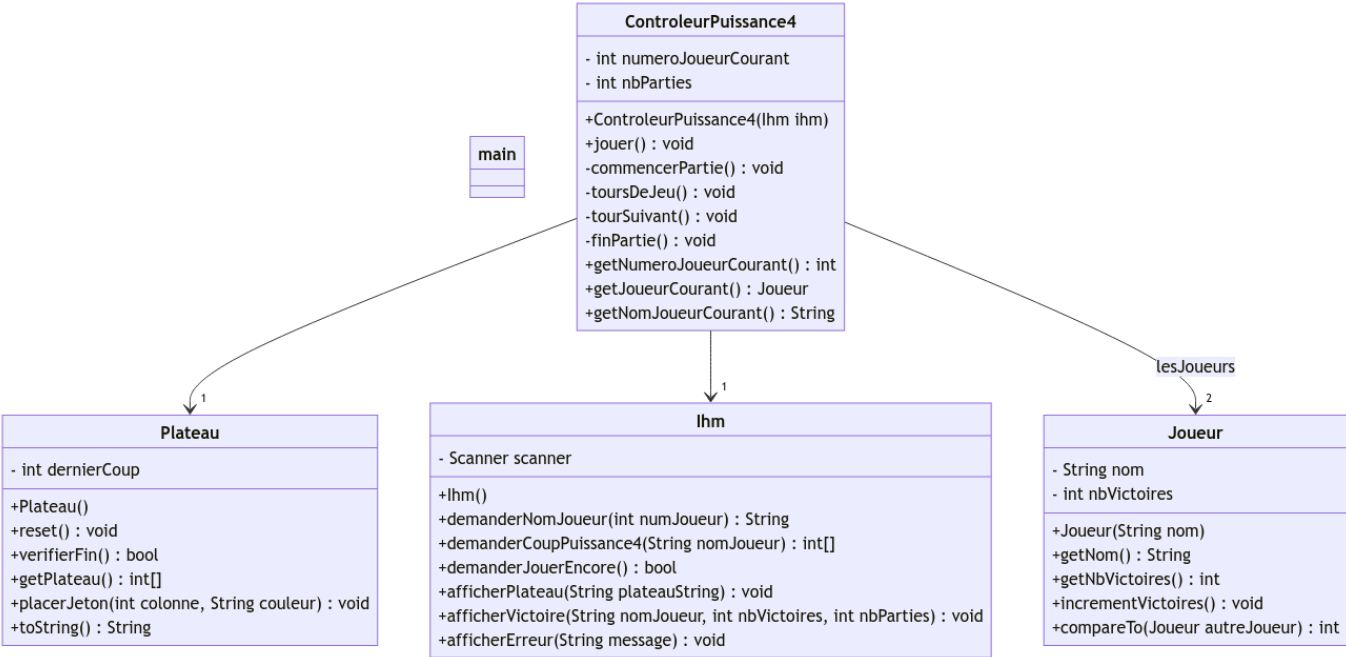
class Joueur {
  - String nom
  - int nbVictoires

  + Joueur(String nom)
  + getNom() String
  + getNbVictoires() int
  + incrementVictoires() void
  + compareTo(Joueur autreJoueur) int
}

class Plateau {
  - int dernierCoup

  + Plateau ()
  + reset() void
  + verifierFin() bool
  + getPlateau() int[]
  + placerJeton(int colonne, String couleur) void
  + toString() String
}
```

ControleurPuissance4 --> "1" Plateau
ControleurPuissance4 --> "1" Ihm
ControleurPuissance4 --> "2" Joueur : lesJoueurs



```
public static void main(String[] args) {
  Ihm ihm = new Ihm();
```

```
ControleurPuissance4 controleurPuissance4=new ControleurPuissance4(ihm);
controleurPuissance4.jouer();
}
```

Répartition des tâches

Itération 2

14/03/2024 (Publication du sujet)

Nathan, Dany, Agathe : review du sujet et brainstorming (check de la victoire etc)

16/03/2024

Nathan : Diagramme de classe, Cas d'utilisation

Dany : review du Diagramme

Utilisation de l'IA

Itération 2

- Nathan : Aucun Usage
- Dany : Aucun Usage
- Agathe : Aucun Usage