

# Diagramme de cas d'utilisation

---

```
graph LR
    Joueur[🧑 Joueur] --- Sélectionner
    Sélectionner -. include .- Nim([Nim])
    Sélectionner -. include .- Puissance([Puissance 4])
```

## Scénario

---

### UC : Sélectionner

**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 à l'utilisateur à quel jeu il veut jouer
3. L'utilisateur répond à la question
4. UC: Nim

Extensions

4. a L'utilisateur à répondu puissance 4
  1. UC : Puissance 4
5. b L'utilisateur à répondu autre chose
  1. Le système affiche un message d'erreur
  2. Retour au point 2 du scénario nominal

### UC : Nim

**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 à l'utilisateur le nombre de tas qu'il souhaite
3. L'utilisateur répond à la question
4. Le système enregistre le nombre de tas
5. Le système demande le nom du joueur 1
6. Le joueur 1 répond
7. Le système enregistre le nom du joueur 1
8. Le système demande au joueur 2
9. Le joueur 2 répond
10. Le système enregistre le nom du joueur 2
11. Le système demande au joueur le nombre maximal de bâtonnet retirer par coup
12. Le joueur répond
13. Le système enregistre la réponse du joueur
14. Le système affiche les tas ainsi que le nom du joueur qui doit jouer
15. Le joueur choisi dans quel tas il souhaite prendre un certain nombre d'allumette
16. Le système vérifie si la partie est gagnée
17. Le système affiche le vainqueur et demande si l'utilisateur souhaite refaire une partie ("y" or "n")
18. L'utilisateur répond "y" ou "n"
19. Le système retourne au point 11 du scénario nominal et incrémente le classement entre les deux joueurs

## Extensions

4. a Le joueur répond autre chose ou un nombre inférieur à 1
  1. Le système affiche un message d'erreur "le nombre de tas doit être un entier supérieur à 1"
  2. Retour au point 2 du scénario nominal
5. a L'utilisateur répond à la question avec une chaîne vide
  1. Le système affiche un message d'erreur "Le nom ne peut pas être vide"
  2. Retour au point 5 du scénario nominal
6. a L'utilisateur répond à la question avec une chaîne vide
  1. Le système affiche un message d'erreur "Le nom ne peut pas être vide"
  2. Retour au point 8 du scénario nominal
7. a Le joueur répond autre chose ou un nombre inférieur à 0 (peut être supérieur à quelque chose à définir)
  1. Le système affiche un message d'erreur "le nombre de tas doit être un entier supérieur à 0"
  2. Retour au point 11 du scénario nominal
8. a Le joueur n'a pas entré une réponse au format  $m\ n$ 
  1. Le système affiche un message d'erreur "Le format de la réponse doit être  $n\ m$ "
  2. Retour au point 14 du scénario nominal avec le même joueur
9. b Le joueur a demandé un tas qui n'existe pas
  1. Le système affiche un message d'erreur "Tas inexistant"
  2. Retour au point 14 du scénario nominal avec le même joueur

10. c Le joueur veut prendre un nombre d'allumette supérieur au nombre contenue dans le tas

1. Le système affiche un message d'erreur "Pas assez d'allumette dans ce tas"
2. Retour au point 14 du scénario nominal avec le même joueur

11. d Le joueur veut prendre un nombre d'allumette supérieur à la limite

1. Le système affiche un message d'erreur "Le nombre max est ..."
2. Retour au point 14 du scénario nominal avec le même joueur

12. e La partie n'est pas fini

1. Retour au point 14 du scénario nominal avec l'autre joueur

13. a L'utilisateur a répondu autre chose que y ou n

1. Le système affiche un message d'erreur "Répondre avec "y" ou "n""
2. Retour au point 17 du scénario nominal

14. b L'utilisateur a choisi d'arrêter de jouer

1. Le système affiche le classement entre les deux joueur
2. Le système attend une action de l'utilisateur pour arrêter le programme

## UC : Puissance 4

**Périmètre** : Le jeu

**Niveau** : But utilisateur

**Acteur principal** : Joueur

**Pré-condition** : Aucune

**Post-conditions** : Aucune

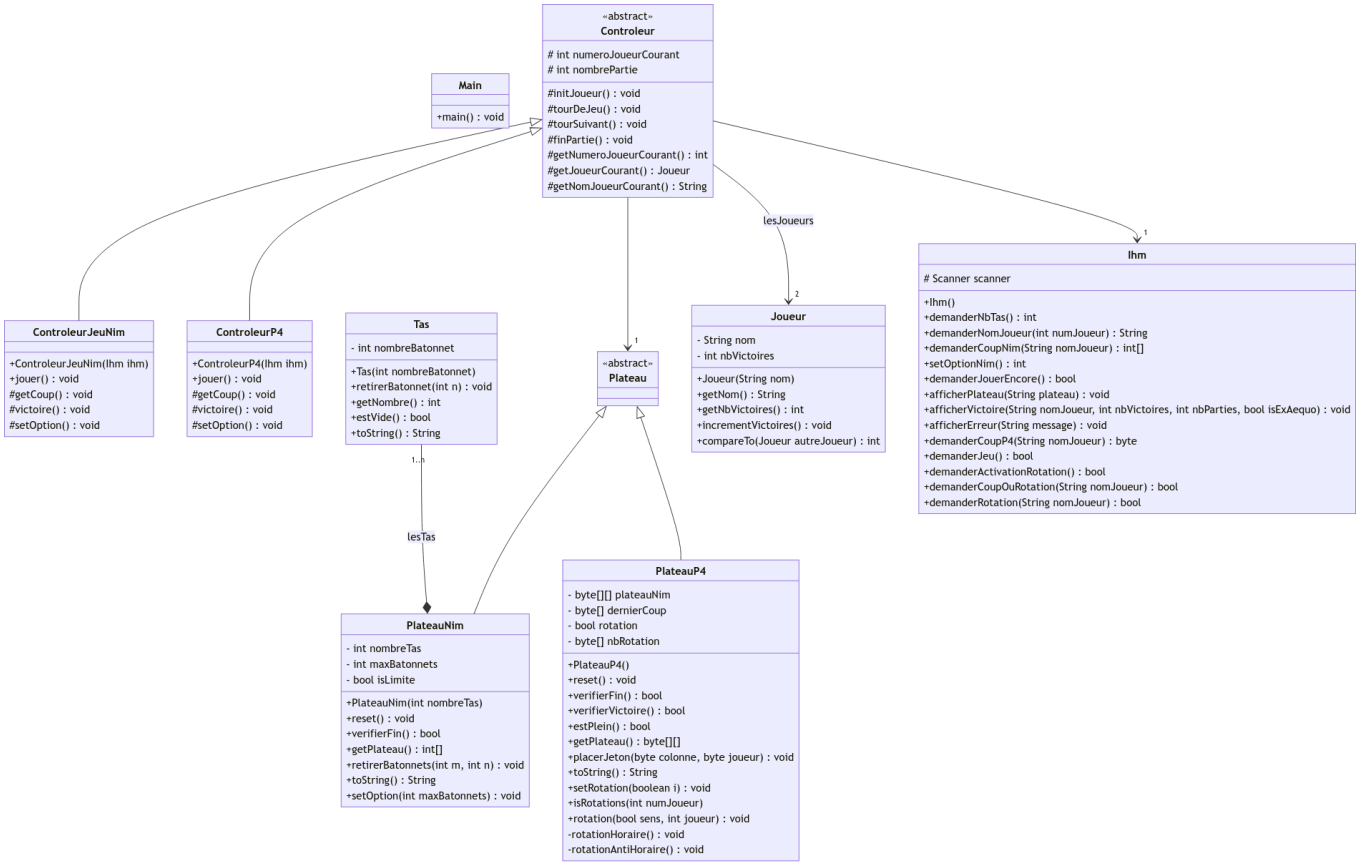
### Scénario nominal

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 le nom du joueur 2
6. Le joueur 2 répond
7. Le système enregistre le nom du joueur 2
8. Le système demande aux joueurs s'ils veulent activer la rotation
9. Le joueur répond
10. Le système enregistre la réponse du joueur
11. Le système affiche le plateau ainsi que le nom du joueur qui doit jouer
12. Le joueur choisi dans quelle colonne il souhaite mettre un jeton
13. Le système vérifie si la partie est gagnée
14. Le système affiche le vainqueur et demande si l'utilisateur souhaite refaire une partie ("y" or "n")
15. L'utilisateur répond "y" ou "n"
16. Le système retourne au point 8 du scénario nominal et incrémente le nombre de partie

## Extensions

- 4.a L'utilisateur répond à la question avec une chaîne vide
  1. Retour au point 2 du scénario nominal
- 7.a L'utilisateur répond à la question avec une chaîne vide
  1. Retour au point 5 du scénario nominal
- 10.a l'utilisateur répond autre chose que 'y' ou 'n'
  1. le système affiche un message d'erreur
  2. retour au point 8 du scénario nominal
- 13.a L'utilisateur choisit un numéro de colonne  $n \notin \{x \in \mathbb{N}, \text{ tq } 1 \leq x \leq 7\}$ 
  1. Le système affiche un message d'erreur
  2. retour au point 11 du scénario nominal
- 13.b L'utilisateur choisit un numéro de colonne valide, mais la colonne est pleine
  1. Le système affiche un message d'erreur
  2. retour au point 11 du scénario nominal
- 13.c L'utilisateur choisit de faire tourner le plateauNim
  1. Le système vérifie que l'option est activée
    1. Le système affiche un message d'erreur
    2. retour au point 11 du scénario nominal
  2. Le système fait tourner la grille
  3. retour au point 13 du scénario nominal
- 13.a La partie n'est pas finie
  1. Le système change le joueur qui doit jouer
  2. retour au point 11 du scénario nominal
- 16.a l'utilisateur répond autre chose que 'y' ou 'n'
  1. le système affiche un message d'erreur
  2. retour au point 14 du scénario nominal
- 16.b L'utilisateur répond 'n'
  1. le système affiche le gagnant et le nombre de victoire du gagnant
    - Il n'y a pas de gagnant
      1. le système affiche ex æquo et le nombre de victoire des joueurs (il est identique pour les deux)
      2. Retour au point 2 de l'extension 16.b
  2. fin du programme

# Modèle de Conception



## Répartition des tâches

### Itération 3

21/03/2024 (Publication du sujet)

Dany : Diagramme de classe, Cas d'utilisation, Fusion, Review


Nathan : Contrainte Nim, Fusion, Review

Agathe : Contrainte Puissance 4, Review


## Utilisation de l'IA

### Itération 3

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

 danydudiot

j'ai une classe qui herite d'une classe abstraite mais je ne peux pas implémenté toutes les méthodes définie abstraction comment faire sans que ma sous classe ne devienne abstract

 GitHub Copilot

Si vous avez une classe qui hérite d'une classe abstraite et que vous ne pouvez pas implémenter toutes les méthodes abstraites, vous avez plusieurs options :

1. Déclarer votre classe comme abstraite. Cela signifie que vous ne pourrez pas instancier cette classe, mais vous pourrez créer des sous-classes qui implémentent les méthodes manquantes.
2. Fournir une implémentation par défaut pour les méthodes abstraites que vous ne pouvez pas implémenter. Cette implémentation peut ne rien faire (c'est-à-dire être vide), lancer une exception, ou retourner une valeur par défaut.

Voici un exemple de la deuxième option en Java :


```
public abstract class ClasseAbstraite {
    public abstract void methode1();
    public abstract void methode2();
}

public class MaClasse extends ClasseAbstraite {
    @Override
    public void methode1() {
        // Implémentation de methode1
    }


    @Override
    public void methode2() {
        // Vous ne pouvez pas implémenter methode2, donc vous fournissez une implémentation par défaut
        throw new UnsupportedOperationException("Méthode non implémentée");
    }
}
```

Dans cet exemple, MaClasse hérite de ClasseAbstraite et implémente methode1. Pour methode2, qui ne peut pas être implémentée, une UnsupportedOperationException est lancée. Cela signifie que si methode2 est appelée sur une instance de MaClasse, une exception sera lancée.

> Using 4 References

 danydudiot

comment faire un regex qui est bon seulement avec "p4" ou "nim"

 GitHub Copilot

Pour créer une expression régulière (regex) qui correspond uniquement aux chaînes "p4" ou "nim", vous pouvez utiliser l'opérateur | qui signifie "ou". Voici comment vous pouvez le faire :

```
Pattern pattern = Pattern.compile("(p4|nim)$");
```

Dans cette expression régulière, ^ signifie le début de la chaîne, \$ signifie la fin de la chaîne, et (p4|nim) signifie "p4" ou "nim". Donc, cette expression régulière correspondra uniquement aux chaînes qui sont exactement "p4" ou "nim".

> Using 4 References