# Itération 1

## Diagramme de cas d'utilisation

Une image contenant texte, diagramme, ligne, Police

Description générée automatiquement

## Scénario

UC : Jouer au jeu de 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 affiche les tas ainsi que le nom du joueur qui doit jouer
12. Le joueur choisi dans quel tas il souhaite prendre un certain nombre d'allumette
13. Le système vérifie si la partie est gagné
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 incrément le classement entre les deux joueur

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

7. 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

10. 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

13. 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 11 du scénario nominal avec le même joueur

13. 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 11 du scénario nominal avec le même joueur

13. 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 11 du scénario nominal avec le même joueur

13. d La partie n'est pas fini

1. Retour au point 11 du scénario nominal avec l'autre joueur

16. 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 14 du scénario nominal

16. b L'utilisateur a choisi d’arrêter de jouer

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

## Diagrammes de Classes

classDiagram
    class main {

    }

    class Ihm {
        - Scanner scanner

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

    class ControleurJeuNim {
        - int numeroJoueurCourant
        - int nbParties
        
        + ControleurJeuNim(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 Tas {
        - int nombreBatonnet

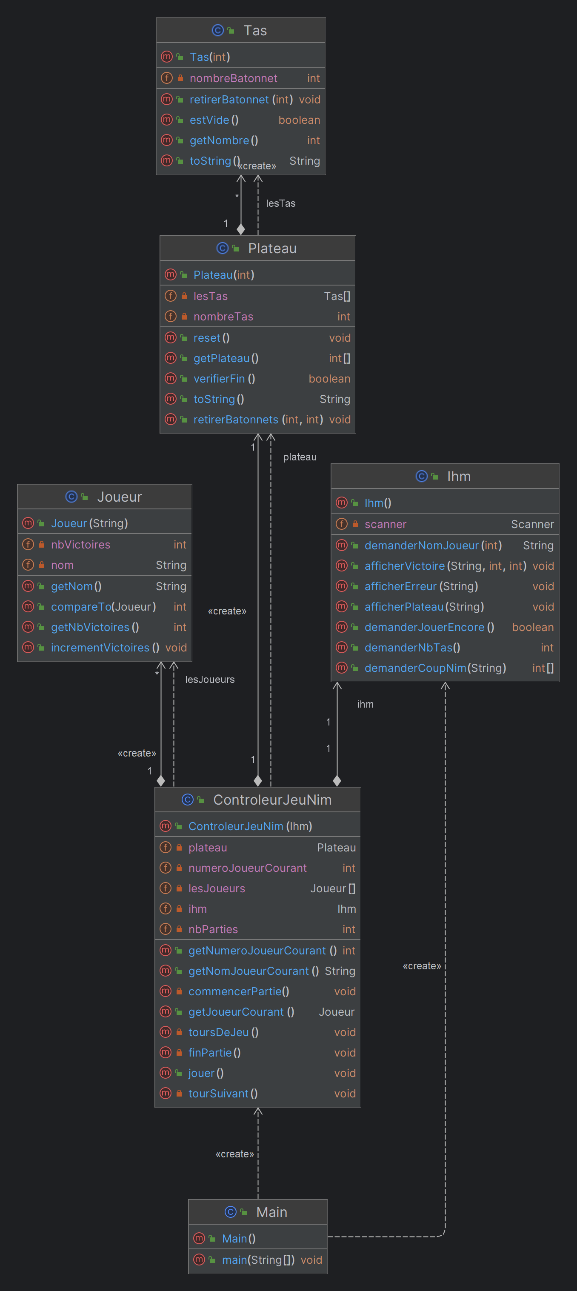
        + Tas (int nombreBatonnet)
        + retirerBatonnet(int n) void
        + getNombre() int
        + estVide() bool
        + toString() String
    }

    class Plateau {
        - int nombreTas
        
        + Plateau (int nombreTas)
        + reset() void
        + verifierFin() bool
        + getPlateau() int[]
        + retirerBatonnets(int m, int n) void
        + toString() String
    }

    Tas "1..n" --* "1" Plateau : lesTas
    ControleurJeuNim "1" <-- "1" Plateau
    ControleurJeuNim "1" <-- "1" Ihm
    ControleurJeuNim "1" <-- "2" Joueur : lesJoueurs
    ControleurJeuNim "1" --> "1" main
    Ihm "1" --> "1" main

Version Intellij

classDiagram
direction BT
class ControleurJeuNim {
  + ControleurJeuNim(Ihm) 
  - Plateau plateau
  - int numeroJoueurCourant
  - Joueur[] lesJoueurs
  - Ihm ihm
  - int nbParties
  + getNumeroJoueurCourant() int
  + getNomJoueurCourant() String
  - commencerPartie() void
  + getJoueurCourant() Joueur
  - toursDeJeu() void
  - finPartie() void
  + jouer() void
  - tourSuivant() void
}
class Ihm {
  + Ihm() 
  - Scanner scanner
  + demanderNomJoueur(int) String
  + afficherVictoire(String, int, int) void
  + afficherErreur(String) void
  + afficherPlateau(String) void
  + demanderJouerEncore() boolean
  + demanderNbTas() int
  + demanderCoupNim(String) int[]
}
class Joueur {
  + Joueur(String) 
  - int nbVictoires
  - String nom
  + getNom() String
  + compareTo(Joueur) int
  + getNbVictoires() int
  + incrementVictoires() void
}
class Main {
  + Main() 
  + main(String[]) void
}
class Plateau {
  + Plateau(int) 
  - Tas[] lesTas
  - int nombreTas
  + reset() void
  + getPlateau() int[]
  + verifierFin() boolean
  + toString() String
  + retirerBatonnets(int, int) void
}
class Tas {
  + Tas(int) 
  - int nombreBatonnet
  + retirerBatonnet(int) void
  + estVide() boolean
  + getNombre() int
  + toString() String
}

ControleurJeuNim "1" *--> "ihm 1" Ihm 
ControleurJeuNim "1" *--> "lesJoueurs *" Joueur 
ControleurJeuNim  ..>  Joueur : «create»
ControleurJeuNim  ..>  Plateau : «create»
ControleurJeuNim "1" *--> "plateau 1" Plateau 
Main  ..>  ControleurJeuNim : «create»
Main  ..>  Ihm : «create»
Plateau "1" *--> "lesTas *" Tas 
Plateau  ..>  Tas : «create»
 

## Répartition des tâches

Publication du sujet

Dany : Premier jet du diagramme des UC et scénario

Nathan : Review et premier jet du diagramme de classe

Premier CM avec des questions

Dany : Review et amélioration du diagramme de classe

Agathe, Nathan, Dany : Review des diagrammes

Agathe : Finalisation du diagramme de Classes

Nathan : Développement du code

Nathan et Dany : Debug

Dany : Essaie de création d'une UI mais pas réussi

Début des vacances

Fin des vacances

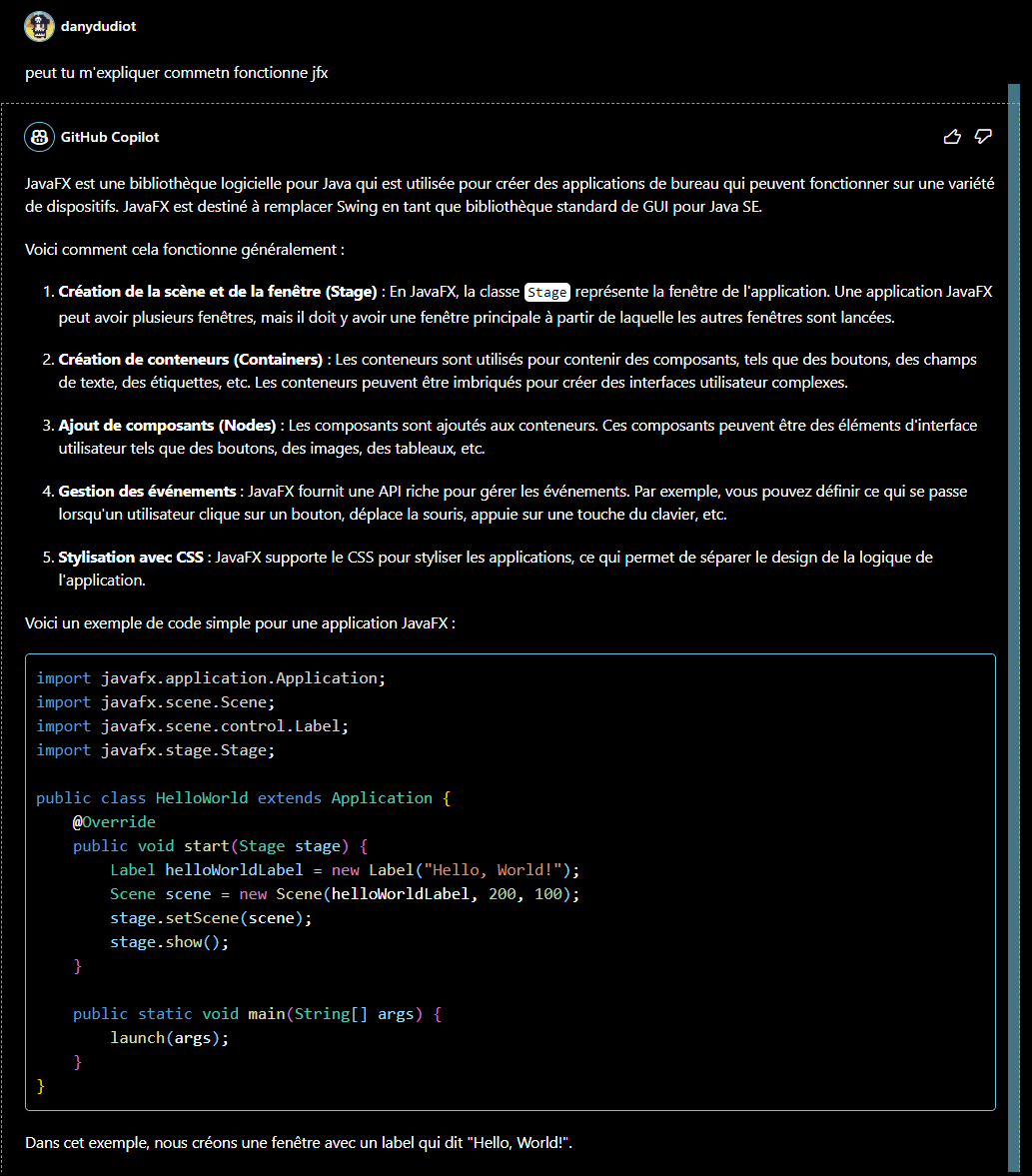
Dany : Petit fix avant le TP

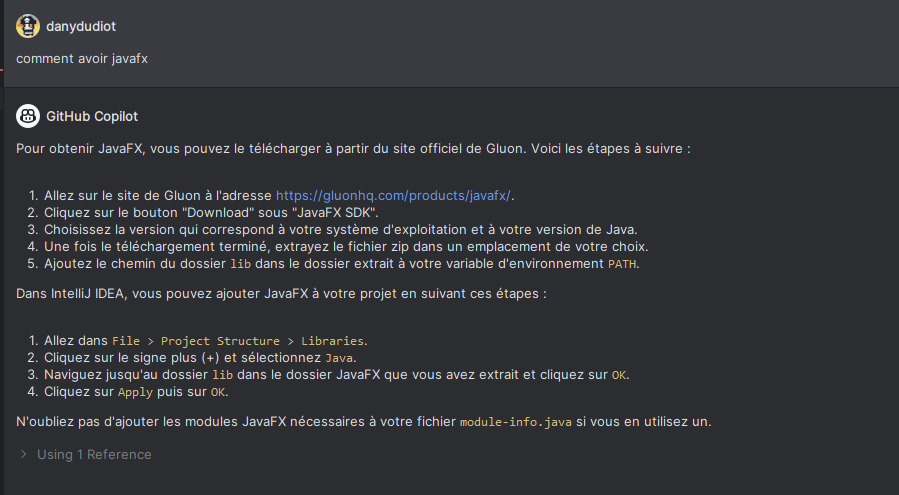
## Utilisation de l'IA

Nathan : Génération de Javadoc sur les fonctions simples (ex set-get, afficherErreur etc.) (<https://chat.openai.com/share/2db9f292-a089-4935-a584-acebe06836a9> )

Dany : usage de Github Copilot pour la découverte de JavaFX

Quelque exemple :

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Agathe : Aucun usage