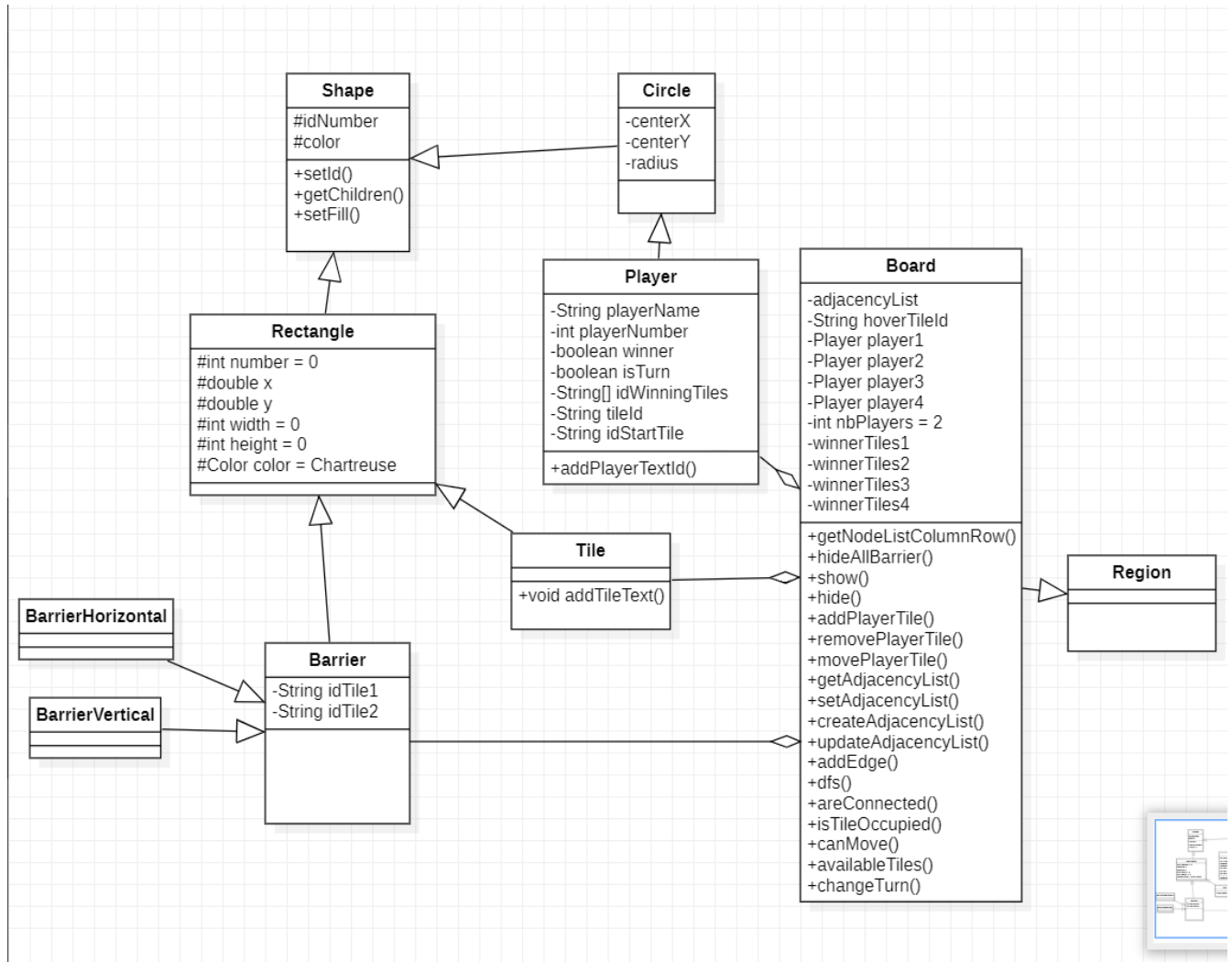


# CY-PATH

**Année 2022-2023**

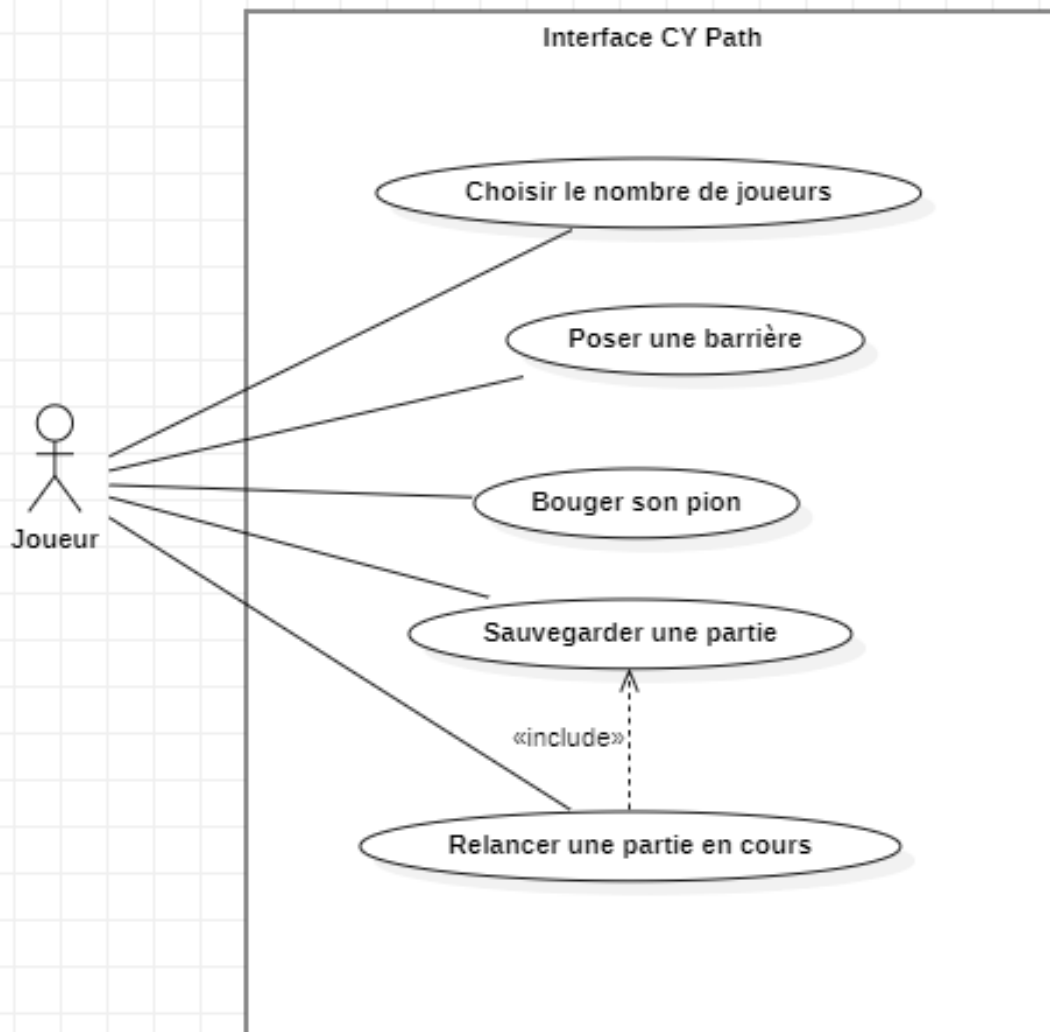
CARDENAS DONOVAN  
GUENNEAU ROMAIN  
JOHNSON ALEXANDRE  
MACHNIK ADRIEN  
RUELLAN BAPTISTE

## Diagramme de classes

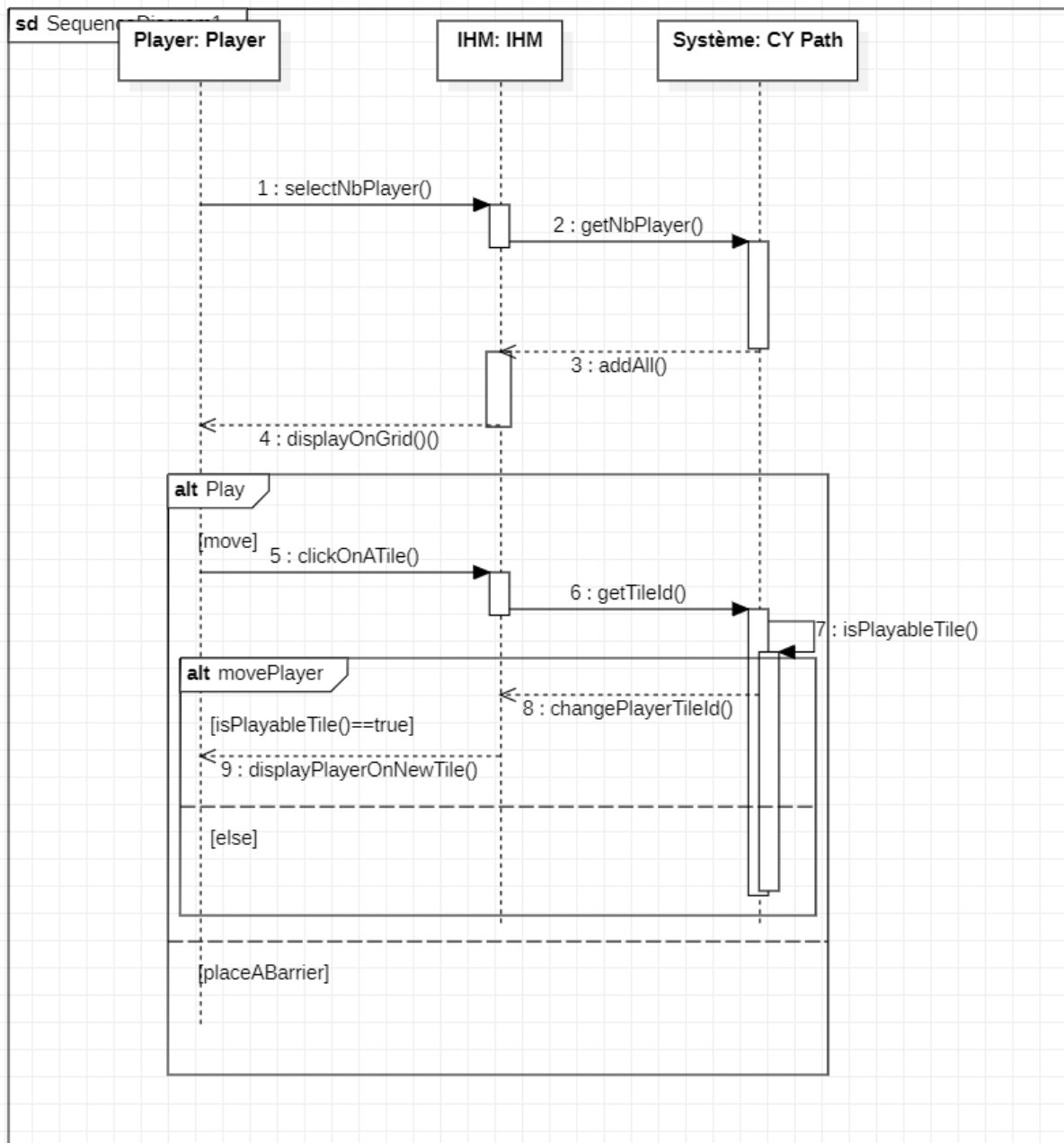


En étudiant le sujet, nous avons décidé de la création de plusieurs classes assez rapidement. La classe Barrier ou la classe Tile sont des idées qui sont venues très rapidement. Après quelques jours de travail sur le projet, l'idée d'une classe Board matérialisant le terrain de jeu et tous les éléments lui appartenant nous a paru la meilleure. Cette classe est notre classe principale, et elle hérite de toutes les autres que nous avons définies.

## Diagramme de cas d'utilisation



## Diagramme de séquence



# L'organisation

Nous avons décidé d'utiliser l'IDE Java Eclipse, la moitié du groupe travaillant déjà dessus.

La première étape a donc été d'expliquer à tous les membres du groupe comment avoir JavaFX et le faire fonctionner. Ensuite nous avons mis en place un Github et avons setup Github Desktop sur tous les ordinateurs afin de pouvoir échanger et travailler sur le projet efficacement. Concernant la répartition du travail, nous utilisons Discord afin de communiquer et avons surtout travaillé au jour par jour en implémentant les différentes fonctionnalités au fur et à mesure. Nous avons commencé par la création et l'affichage du terrain de jeu (Donovan) puis avons petit à petit rajouté les joueurs (Baptiste et Adrien), les règles (Alexandre et Romain) etc...

Concernant les difficultés rencontrées, nous avons au début eu du mal à faire en sorte que JavaFX fonctionne chez tout le monde. Ensuite la première problématique a été celle du choix du type de classe utilisé pour la modélisation de la grille de jeu. Nous étions partis sur un AnchorPane mais avons finalement utilisé un GridPane pour sa simplicité à ajouter de lignes ou colonnes suivant des contraintes de taille ainsi du fait que la surface réelle du Node affiché s'arrête net aux frontières des formes affichées. Cependant, comme nous communiquons et travaillons beaucoup en groupe (par partage d'écran sur Discord), lorsque nous rencontrons des difficultés, souvent un membre de l'équipe avait une idée et nous ne sommes pas souvent restés bloqués sur un problème.

L'un des problèmes de notre projet est la classe Board qui contient quasiment toutes les méthodes. Nous n'avons pas réussi à trouver une façon de rendre la classe moins grosse. Comme la classe contient autant de méthodes, il est parfois difficile de s'y retrouver et de modifier le code.

Il a fallu utiliser la bibliothèque Guava pour faire du formatage de texte afin d'enregistrer la liste d'adjacence dans un fichier .txt.

Importer Guava via Maven a été laborieux à prendre en main au début.

Comme amélioration nous aurions aimé faire un véritable menu au lancement du jeu, un écran de victoire ainsi qu'un drag and drop pour le positionnement des barrières.

13	14	15	16	17	18	19	20	21
Repos après les partiels et reflexion sur la forme du projet	Repos après les partiels et reflexion sur la forme du projet	Setup des IDE et Diagramme de Classe	Début du travail sur la création du terrain	Début du travail sur la création du terrain	Début du travail sur la création du terrain	Création des classes pour effectuer des actions sur les barrières	Ecriture du code qui effectue le DFS	Création de la liste d'adjacence des tuiles

22	23	24	25	26	27	28
Travail sur le code de déplacement des joueurs	Travail sur la condition de victoire	Travail sur le déroulement d'une partie et avancement des tours	Règles pour le positionnement des barrières	Début d'écriture du rapport et travail sur la fonction de sauvegarde	Rédaction du rapport	Finalisation du code (déplacement en diagonale)