

DEV4 : BABA IS YOU

PARTICIPANTS :

53083, SAMUEL IRADUKUNDA

58141, GUILLAUME VILLERS

PROFESSEUR : QUENTIN HOUBEN

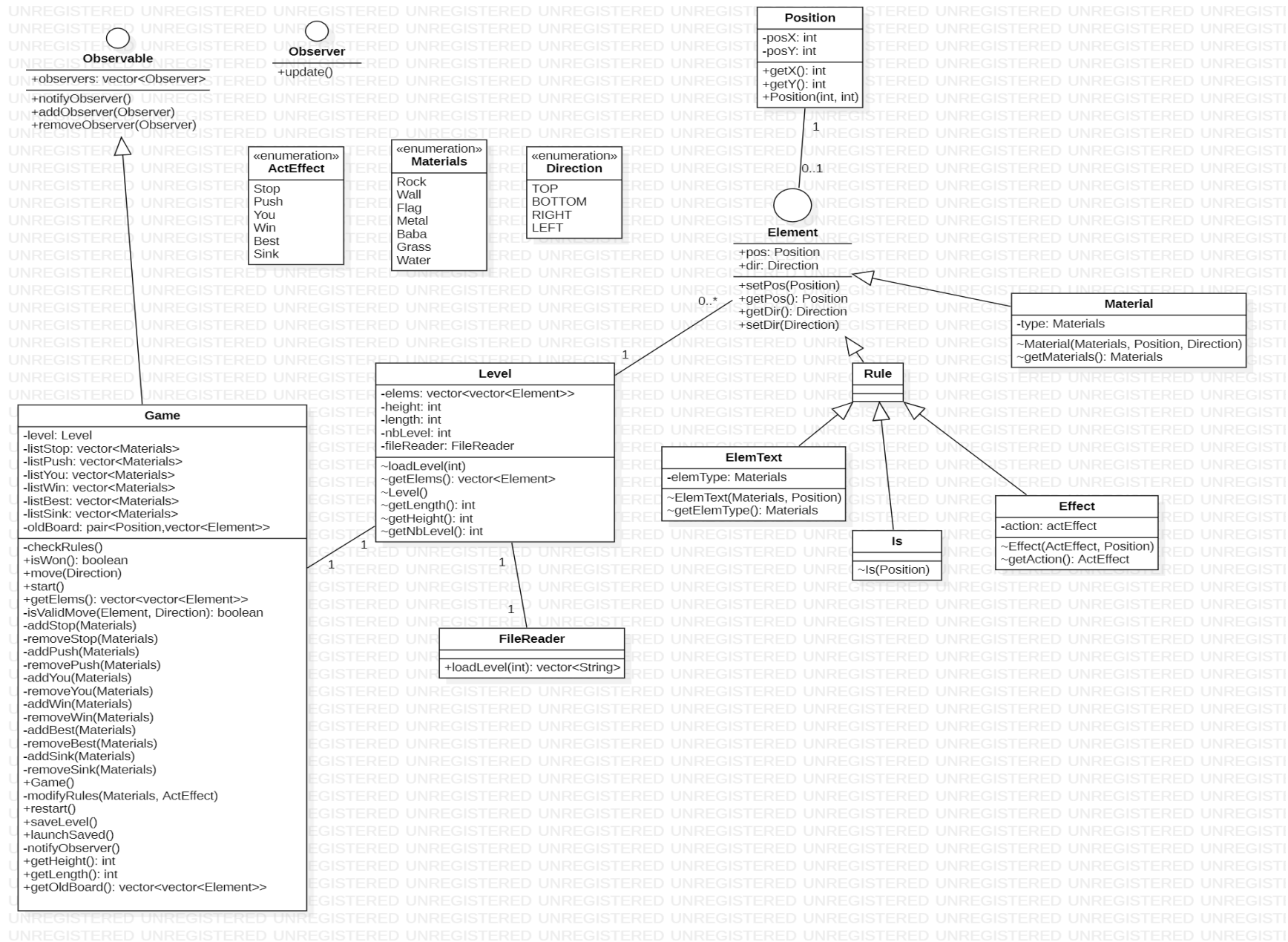
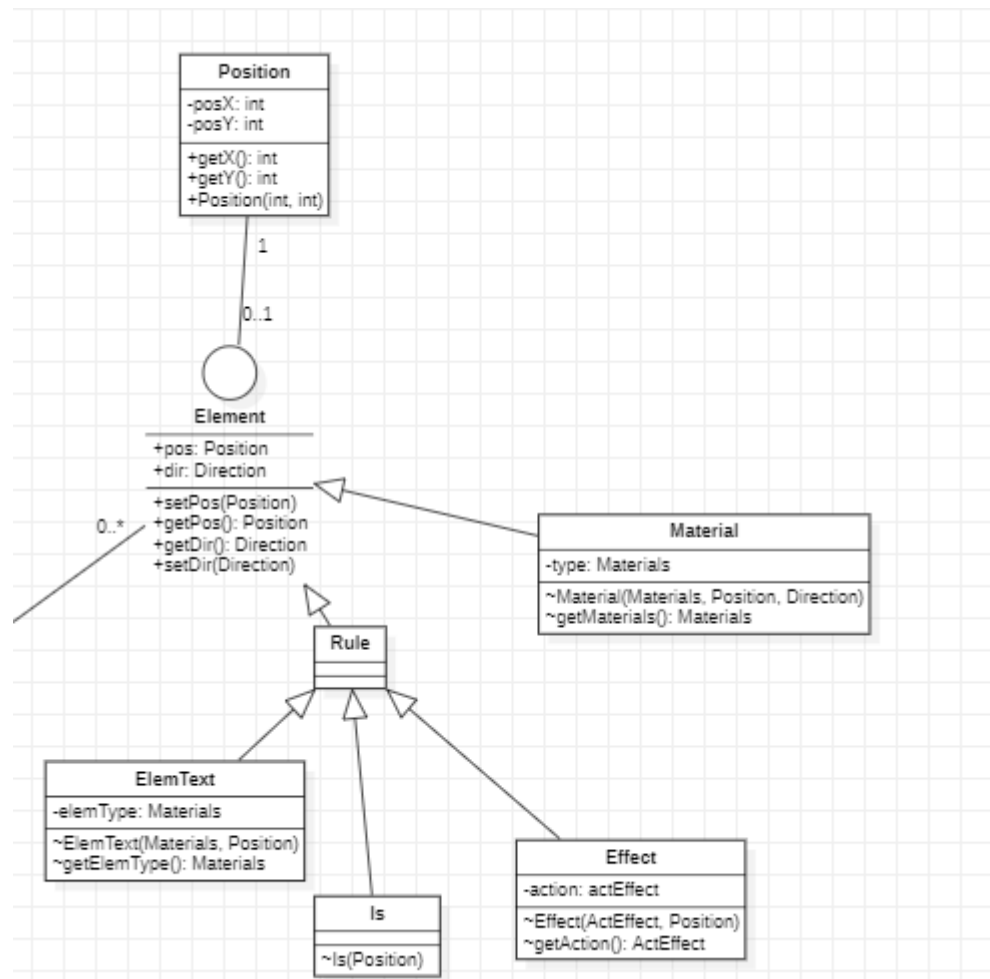


Diagramme de classe :



Pour notre jeu, on a décidé de représenter les différents éléments du plateau par une classe **Elément** de laquelle hériterait les différentes pièces. Donc nous avons comme 4 classes qui représentent les différents types de notre jeu. La classe **Matériau** pour les éléments comme les murs, la lave, l'herbe, le drapeau, ect (voir enum). Une classe **Rule** qui peut être soit un effet dans le jeu (**Effect**), le connecteur **IS** et enfin **ElemText** qui elle représente les textes des règles qui seront graphiquement affichés dans le jeu. Un élément aura toujours une position et un type (instance of).

«enumeration»
ActEffect

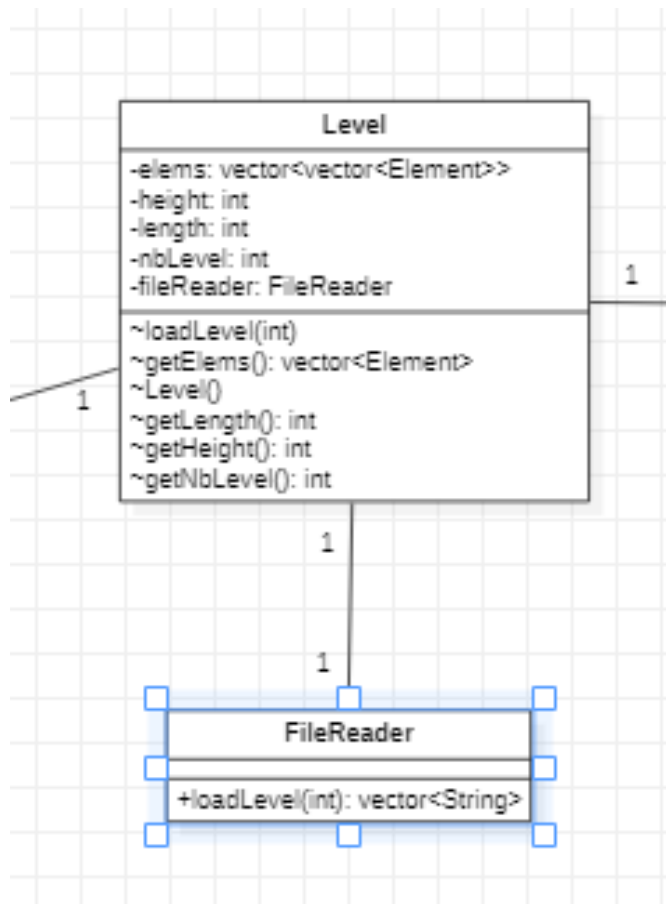
Stop
Push
You
Win
Best
Sink

«enumeration»
Materials

Rock
Wall
Flag
Metal
Baba
Grass
Water

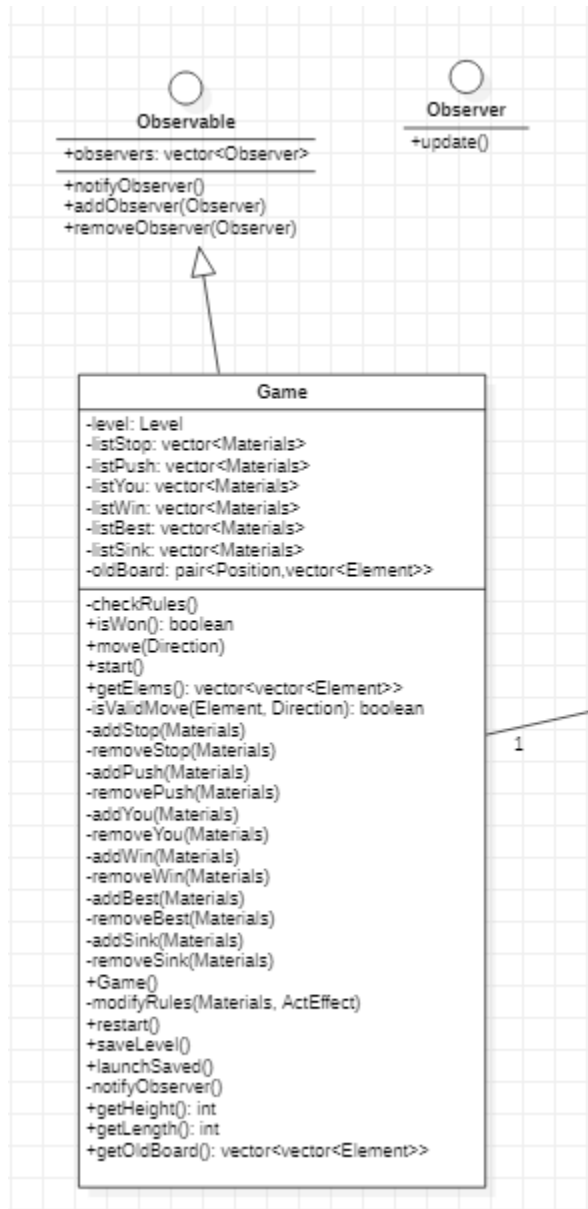
«enumeration»
Direction

TOP
BOTTOM
RIGHT
LEFT



Dans notre jeu, on a décidé de représenter les différents niveaux et les éléments qu'ils contiennent avec la classe **Level**. Cette dernière a comme attribut : un vecteur de vecteur qui contiendra les différents éléments pour une partie donnée. Elle a aussi comme attribut `height`, `length` pour nous permettre de déterminer la taille du plateau. Enfin on a un attribut `fileReader` qui est la classe, qui va récupérer les fichiers textes du level et les parser pour nous retourner un `vector string`. Via ce `string` récupérer, on va pouvoir créer les différents éléments du jeu au niveau de la méthode `loadLevel` et les stocker dans notre attribut « `elems` ».

Notre méthode game est la facade via laquelle le controleur va pouvoir interagir via ses méthodes.



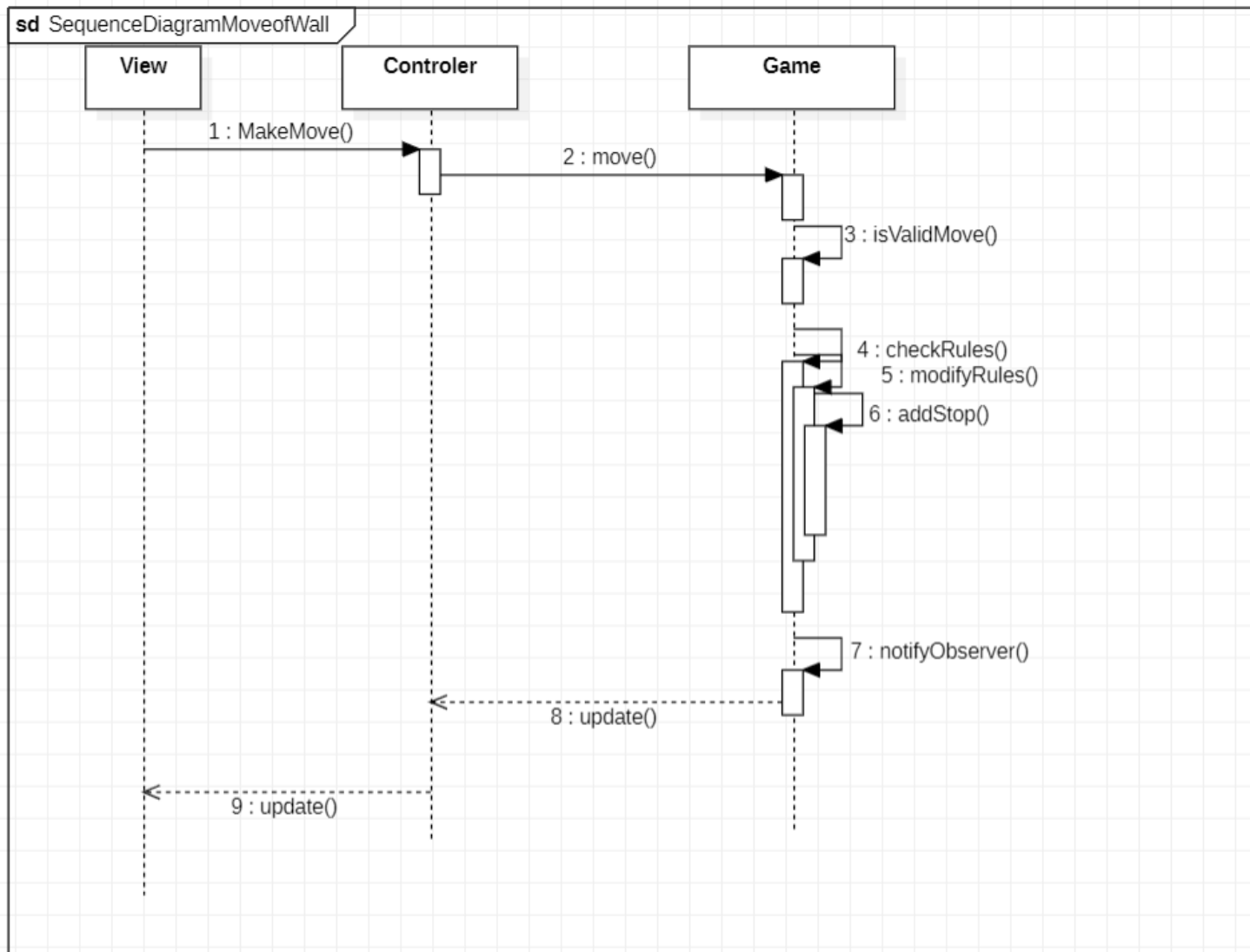


Diagramme de séquence

Ici on a un diagramme de séquence qui représente un mouvement dans le jeu, ce mouvement va déplacer un personnage qui va activer une règle, qui est la règle « Wall is stop ».