Diagramme UML (Global)

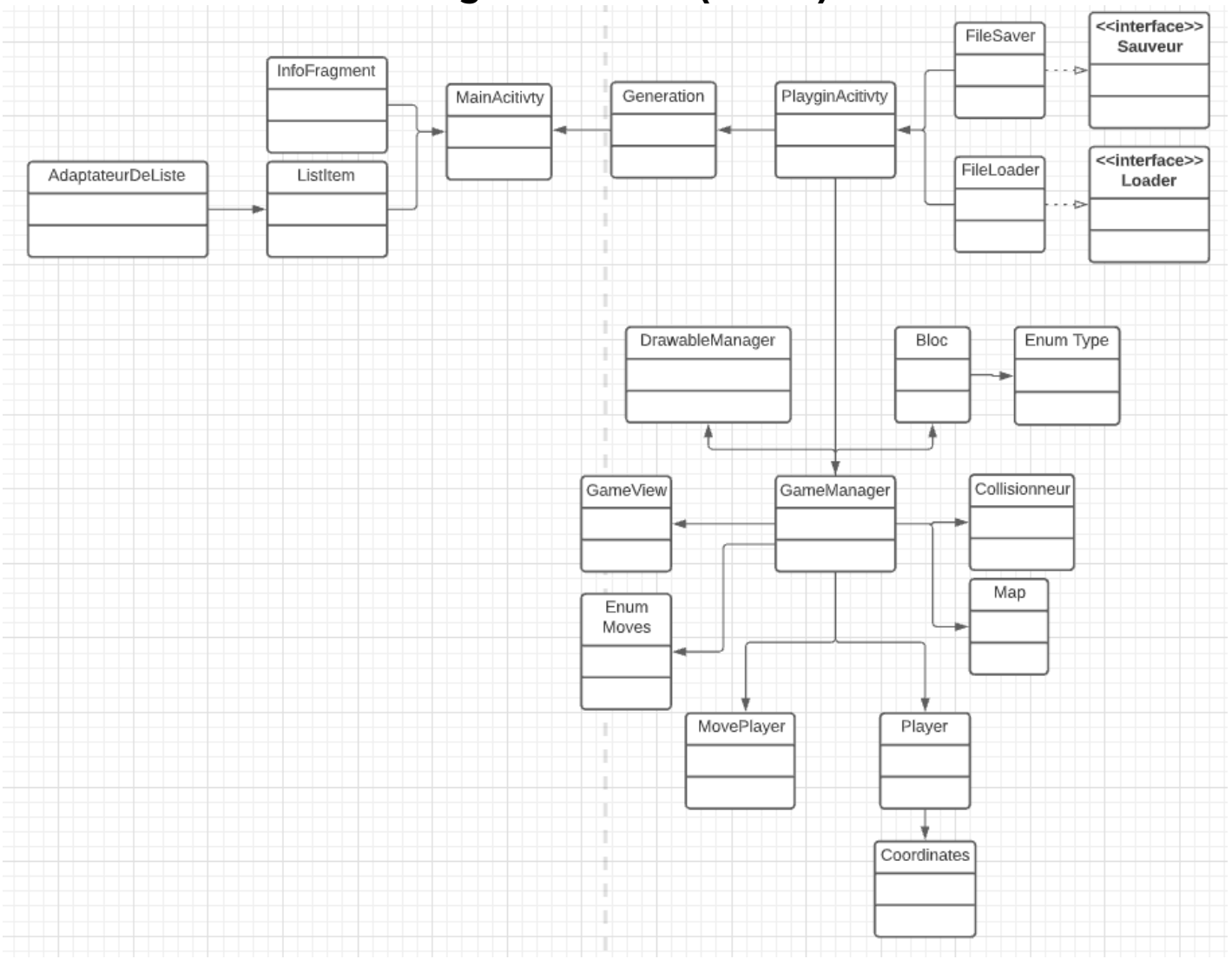


Diagramme UML (Partie Activité)

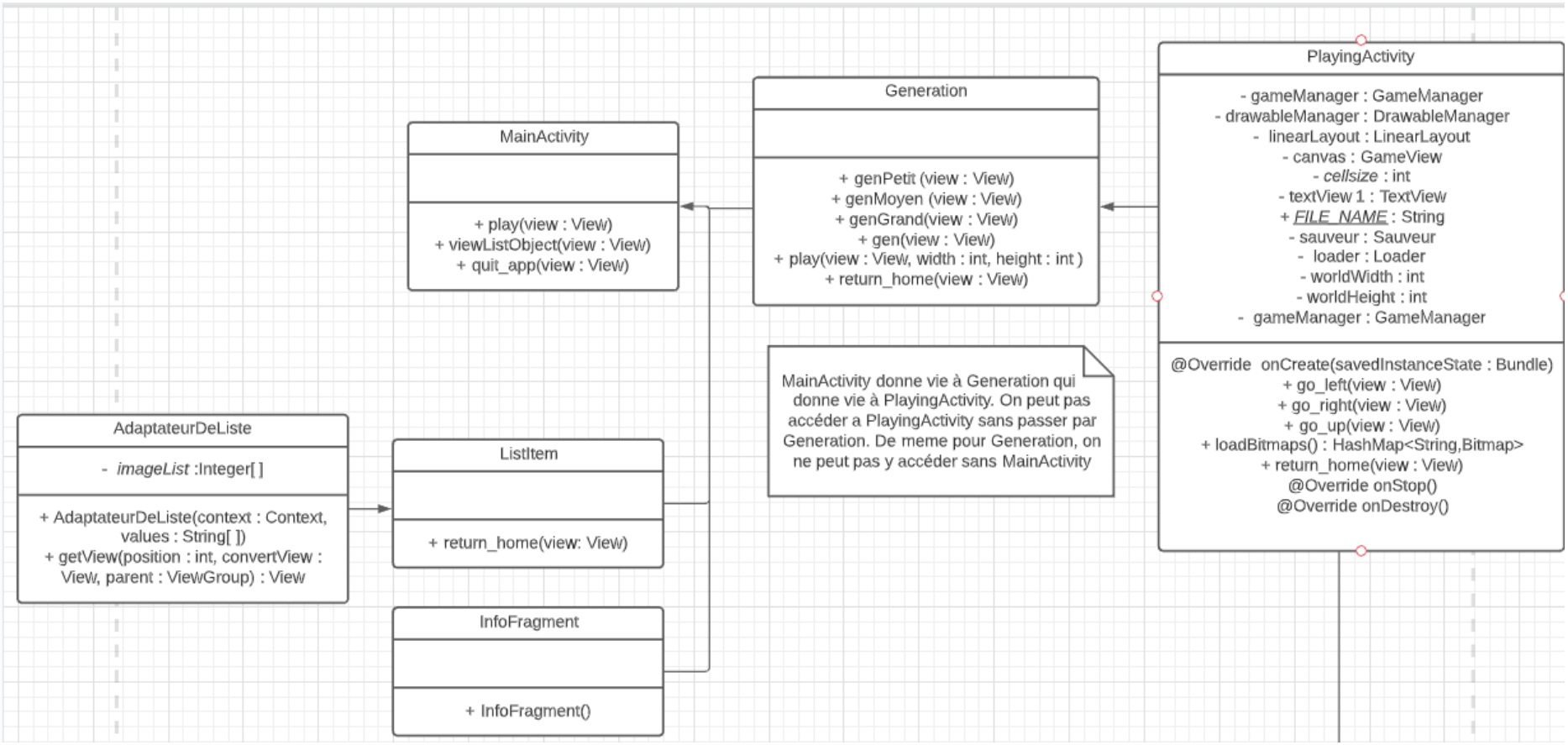


Diagramme UML (Partie Activité + Persistance)

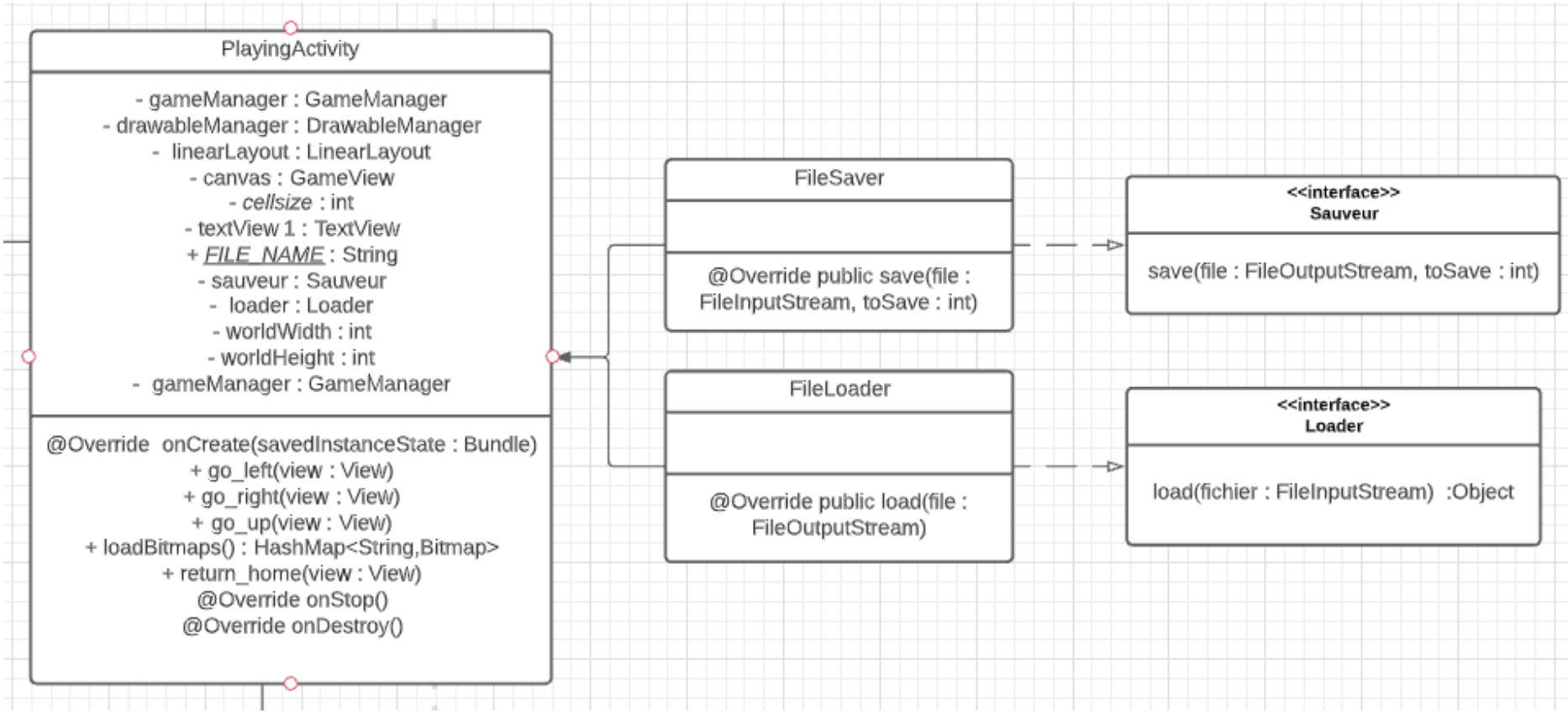


Diagramme UML (Liaison entre les managers et l'activité de jeu)

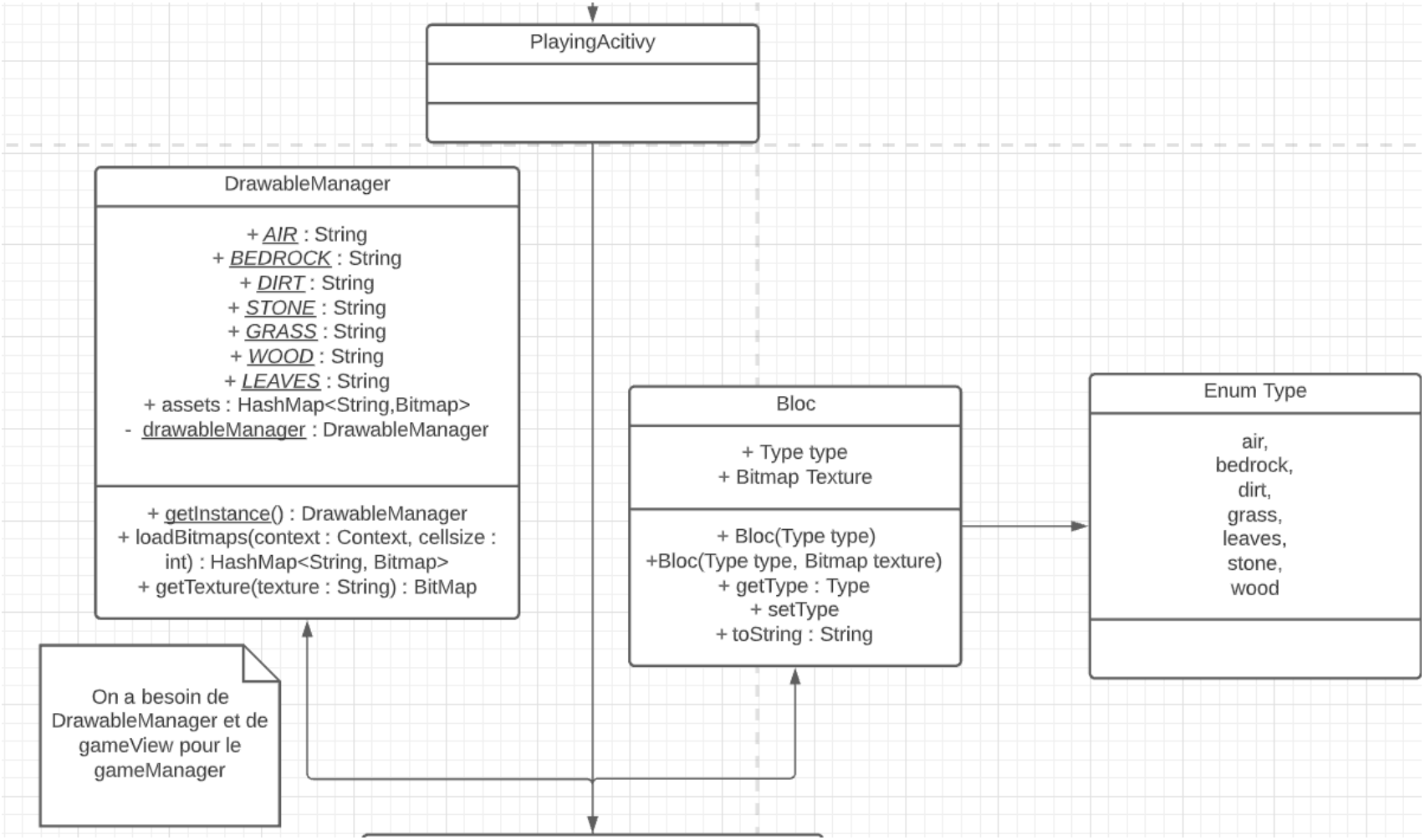


Diagramme UML (Managers)

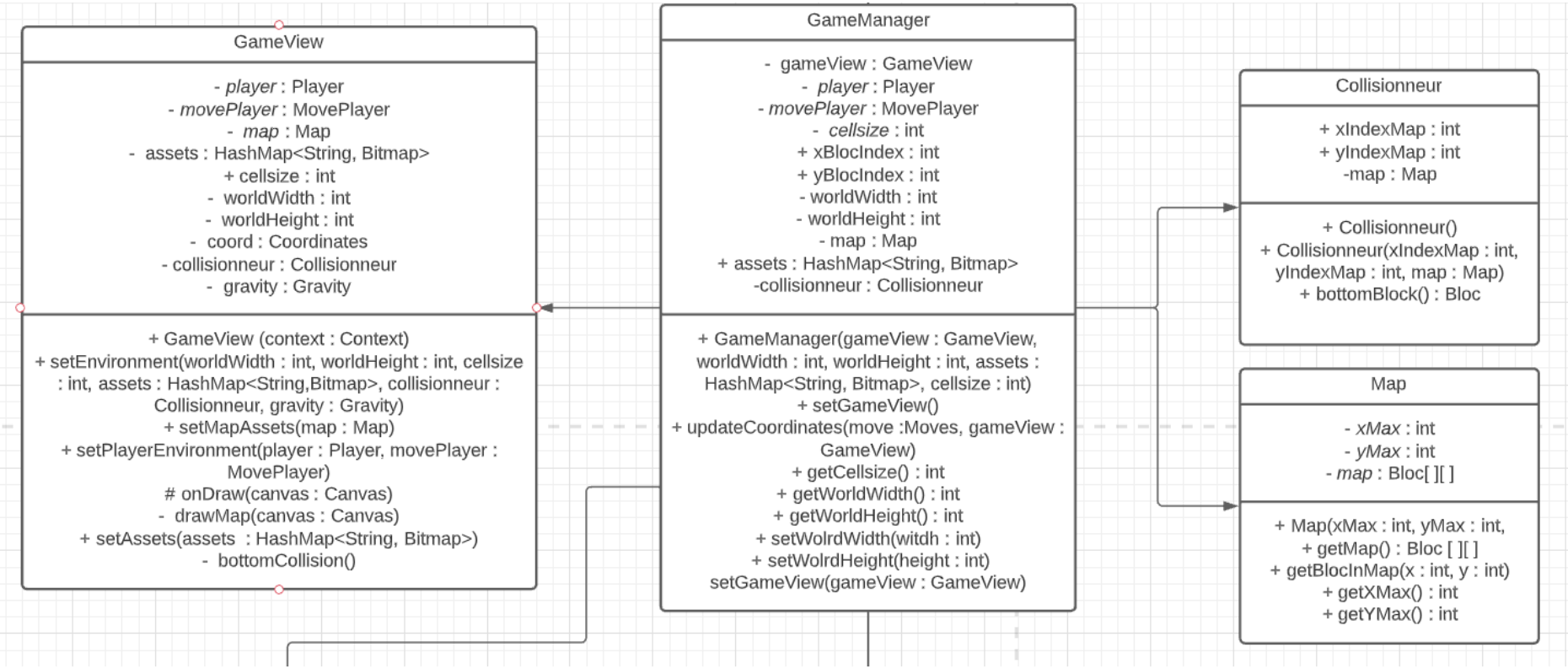
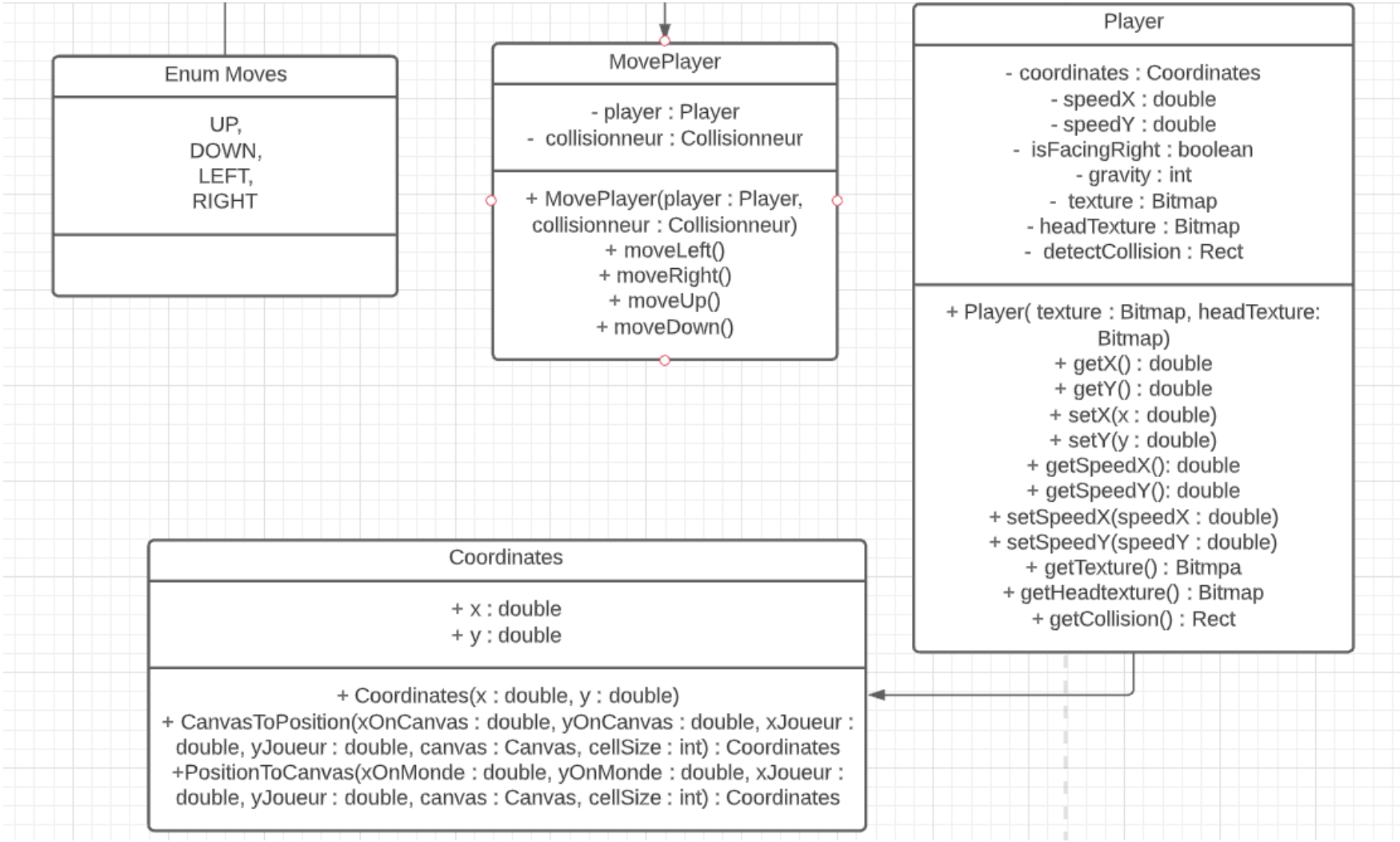


Diagramme UML (Player)



Description du diagramme UML

Premièrement nous avons notre MainActivity qui a besoin de InfoFragment pour afficher le mode du jeu (mode créatif). Par la suite, cette classe a besoin de l'activité ListItem afin d'afficher la liste des blocs. Pour les afficher nous avons besoin de notre classe AdaptateurDeListe. MainActivity permet d'afficher l'activité Generation. Generation permet à son tour d'afficher l'activité PlayingActivity. Generation et PlayingActivity ne peuvent pas être affichées si MainActivity n'existe pas.

Concernant la partie persistance, nous avons un FileSaveur et un FileLoader. Chacune de ces classes implémentent leur interface respective : Sauveur et Loader. Ces deux classes citées précédemment sont utilisées uniquement dans PlayingActivity.

PlayingActivity nécessite un GameManager qui va être la classe qui vient gérer l'ensemble de la partie de jeu.

Concernant GameManager :

GameManager est constitué d'un DrawableManager. Cette classe permet de dessiner sur l'écran la map.

Cette classe est constitué également d'un GameView qui vient créer Steve (le personnage principal.)

Nous avons une association de cette classe avec bloc qui est la classe permettant la création d'un bloc. Dans cette classe nous retrouvons également une énumération de types de blocs. C'est grâce à cela que l'on pourra créer un monde diversifié en blocs.

Pour poursuivre, nous avons la classe Collisionneur qui viendra gérer les collisions entre le joueur et les blocs de la Map.
En parlant de map, nous avons une classe spécifique à sa création qui s'appelle simplement Map.

GameManager a besoin d'un MovePlayer afin de déplacer le joueur lorsque les boutons dans le jeu seront cliqués.

Il a également besoin d'un Player (Steve). Ce player aura besoin, lui, de coordonnées. Pour cela on utilise la classe Coordinates.