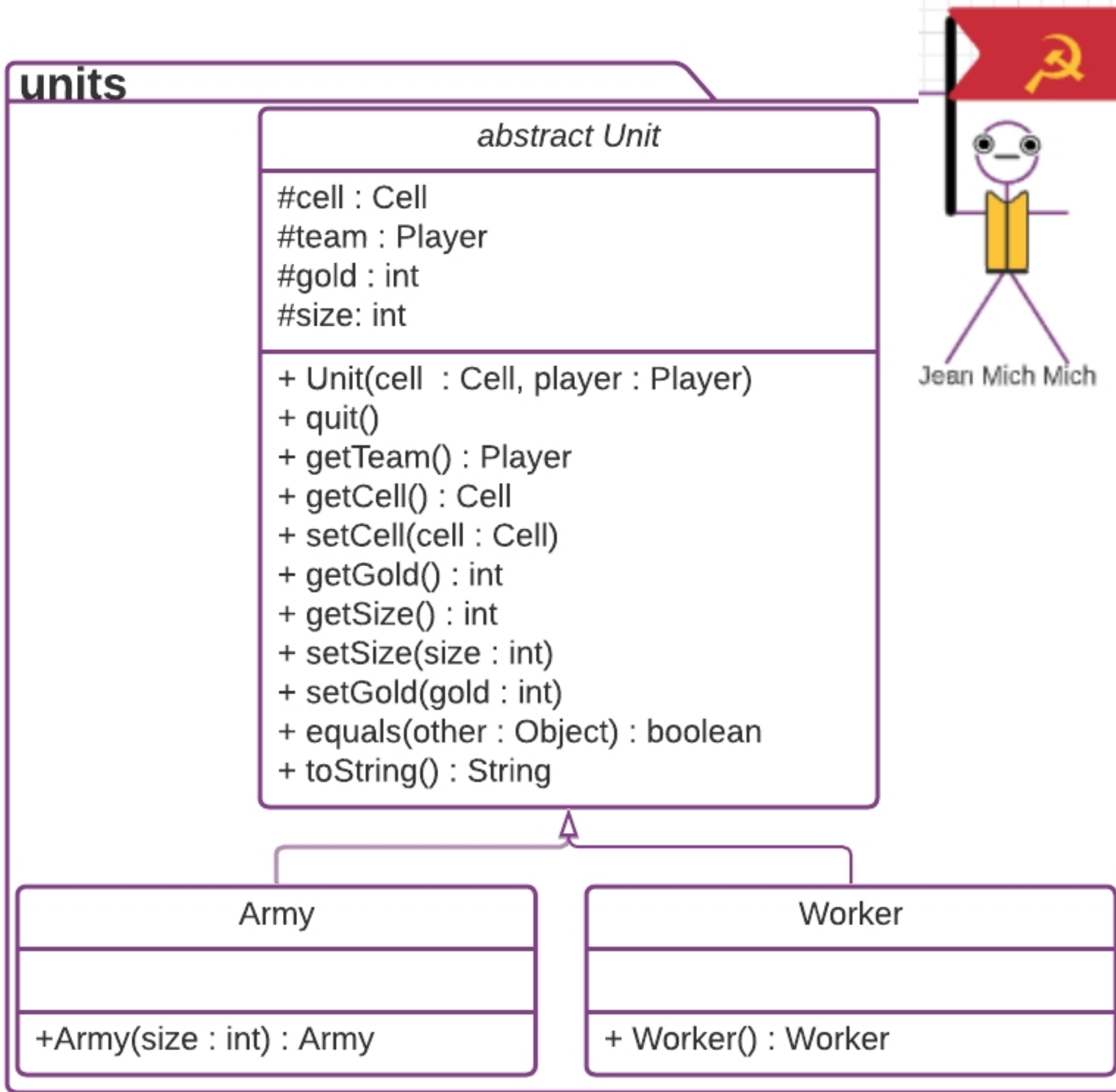
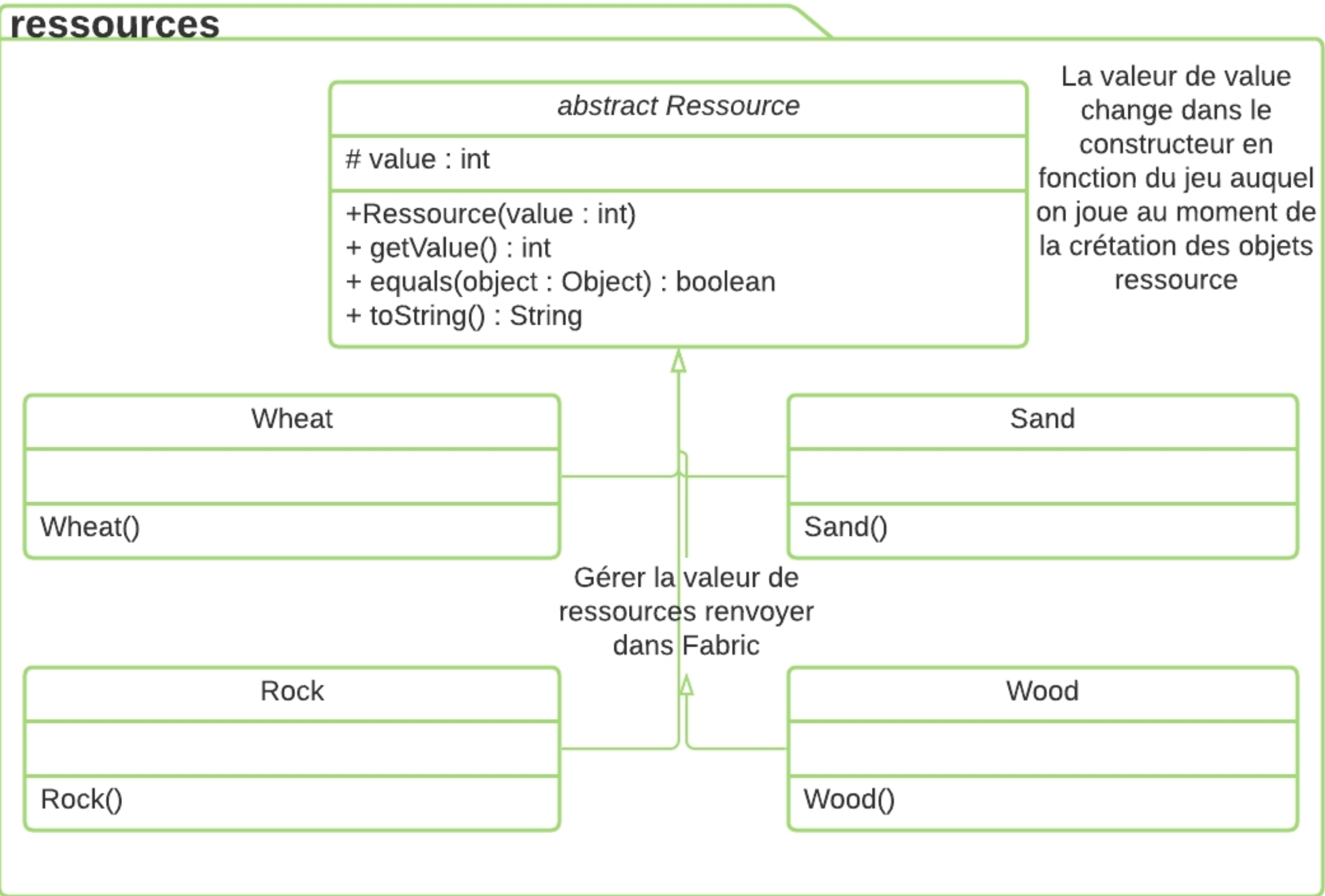
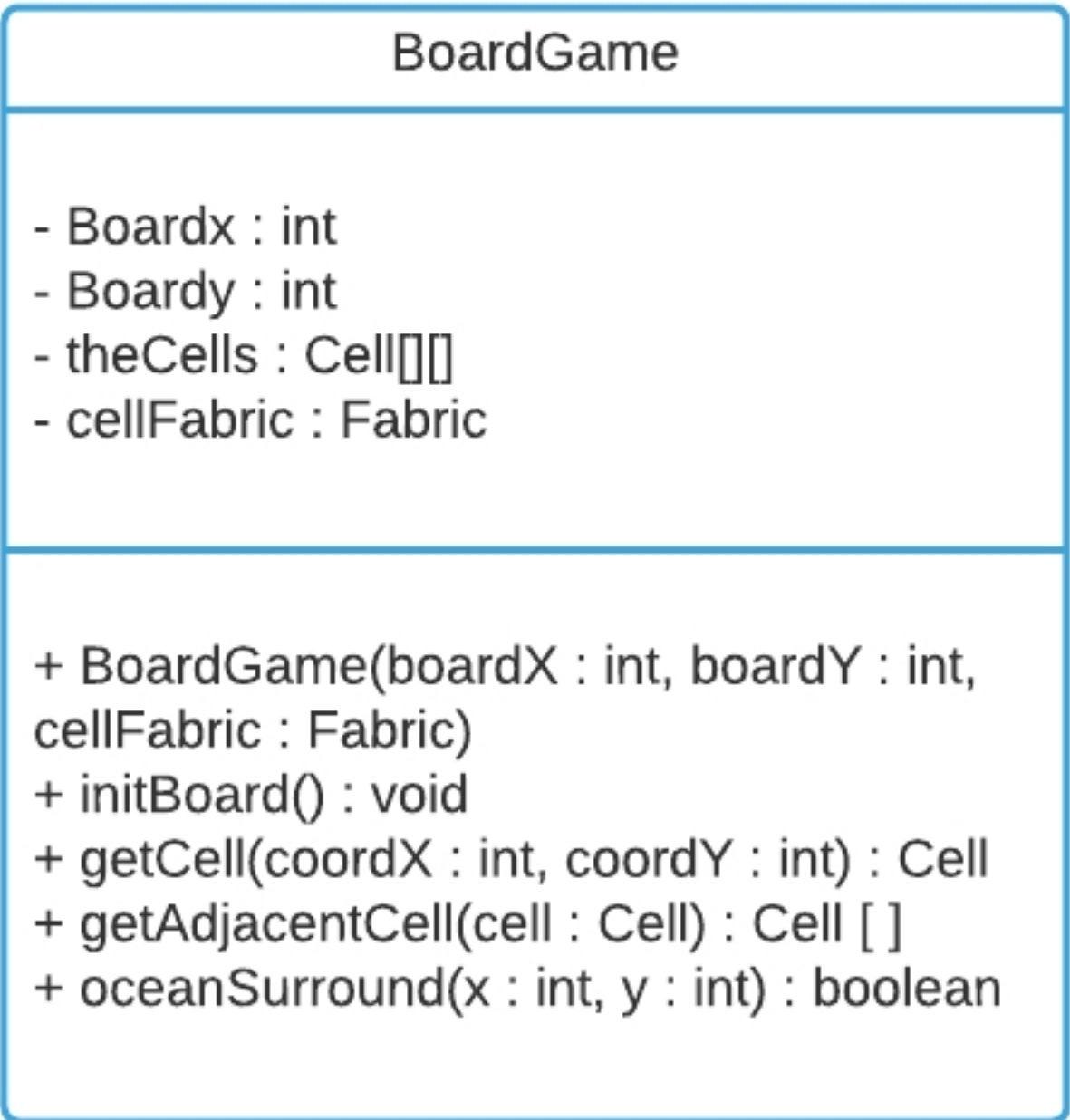


[Lien vers l'autre UML](#)

[Lien vers le sujet](#)



Dans game pour definir le coup du joueur :
this.player.strategie.chooseMove().execute(this.player)

mais le truc c'est que dans certain cas il faudra aussi que le joueur entre d'autres paramametres en fonction du coup qu'il a choisit
Je vois 2 possibilités :

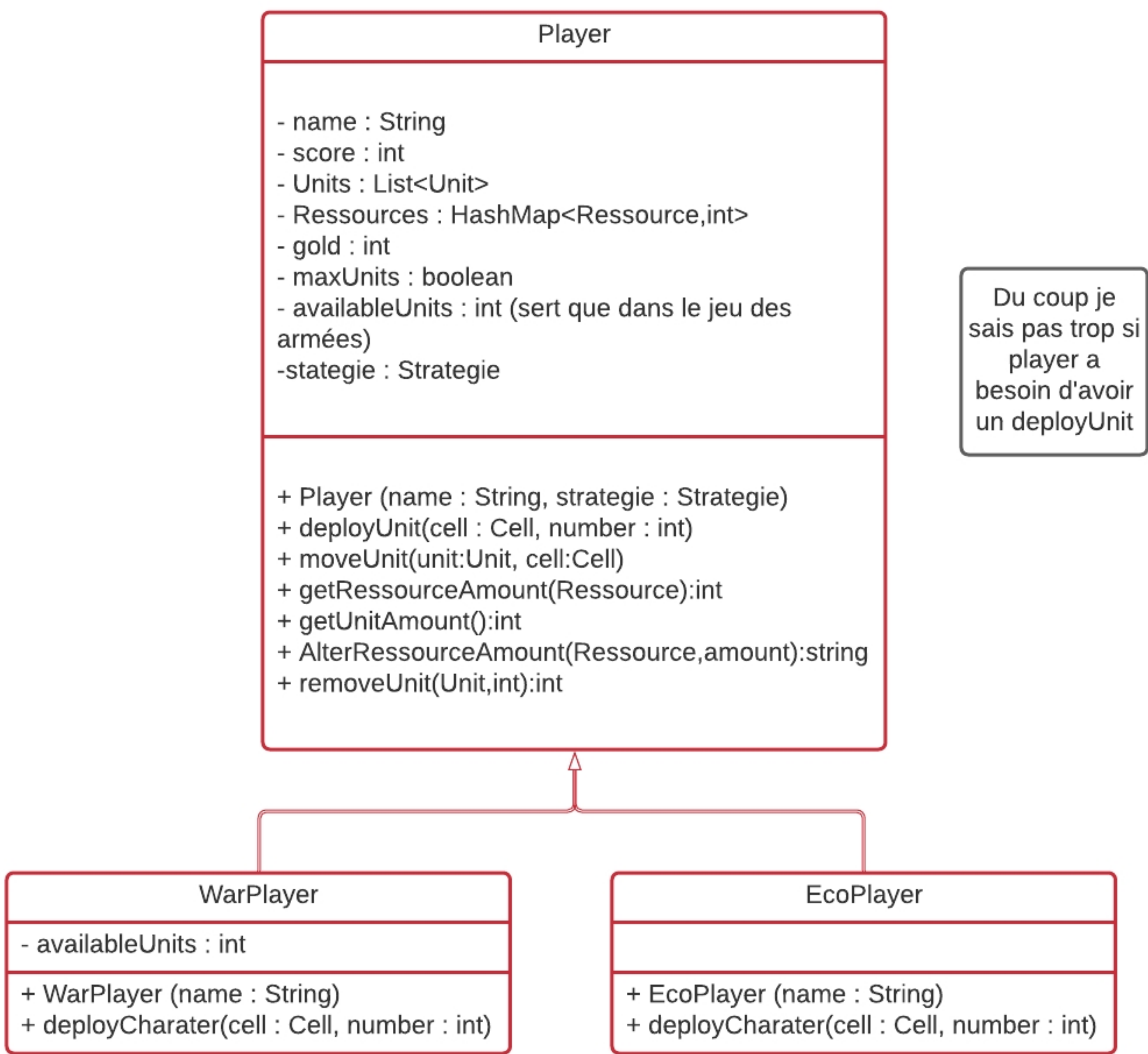
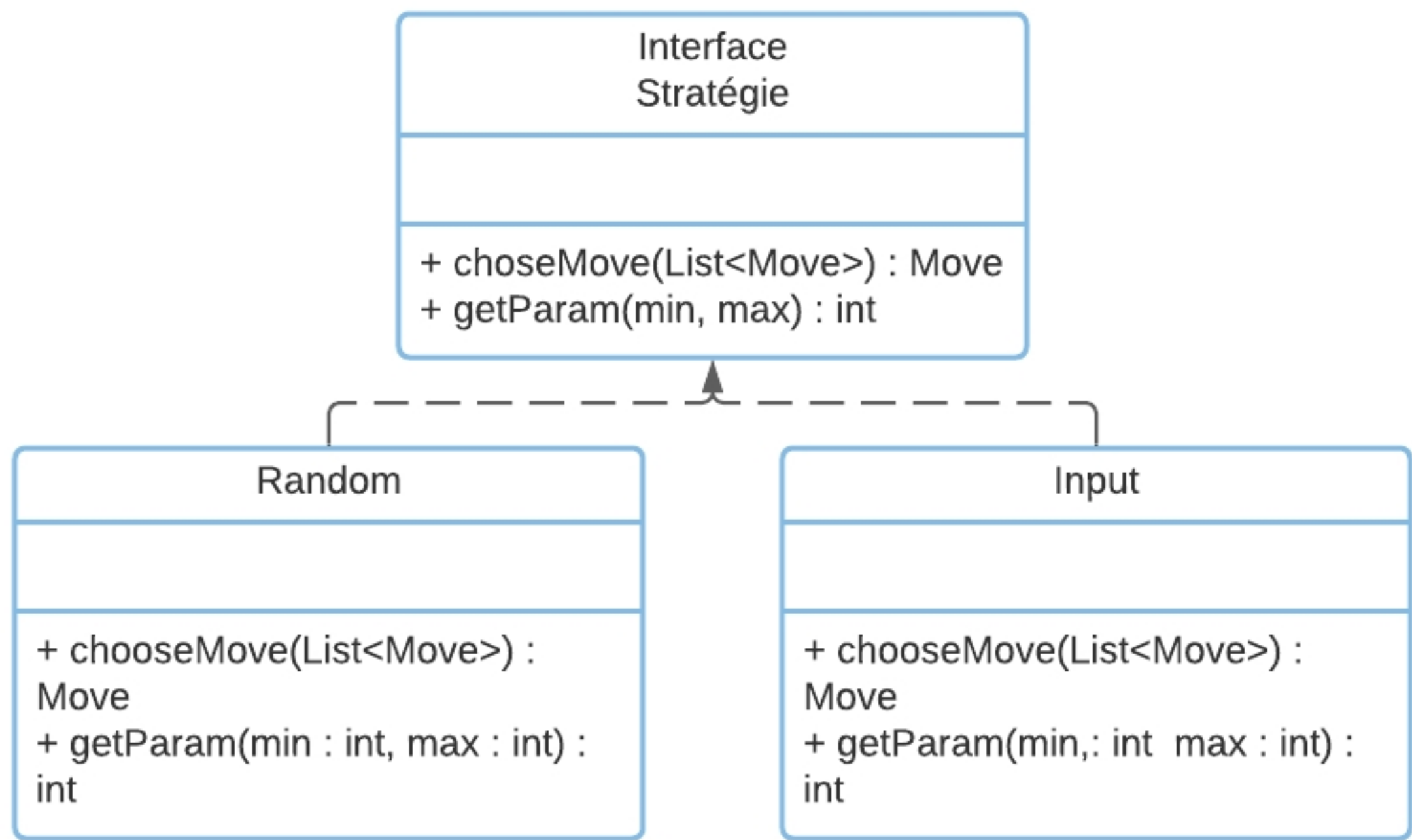
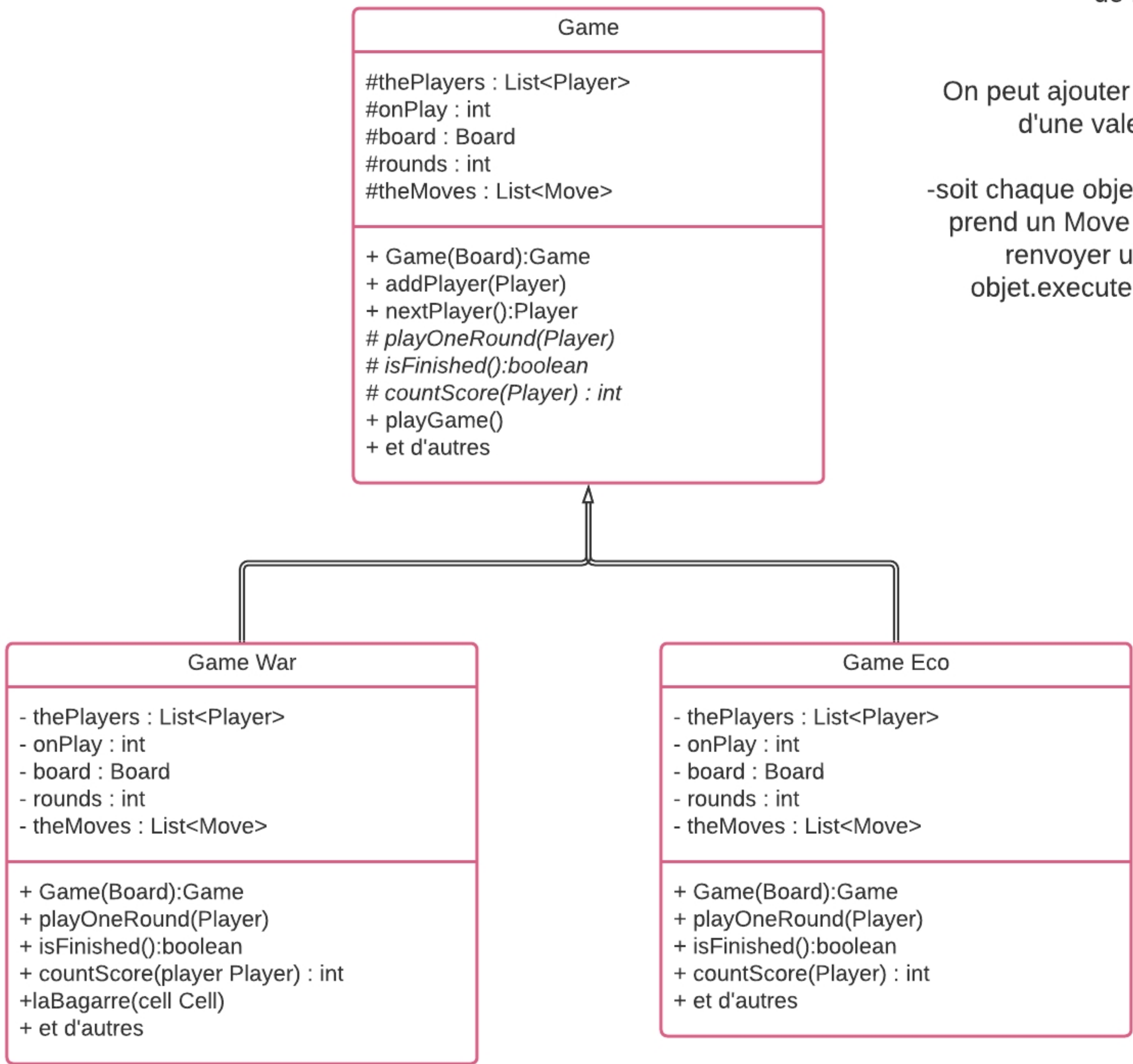
- soit si le execute a besoin de param il les demande lui meme (donc dans le execute y'a un truc genre input(param)) mais le problème c'est que le choix de ces param dépendent de la stratégie du joueur et ca risque d'être compliqué a faire

En fait c'est peut être possible :

si le execute() a player comme parametre :

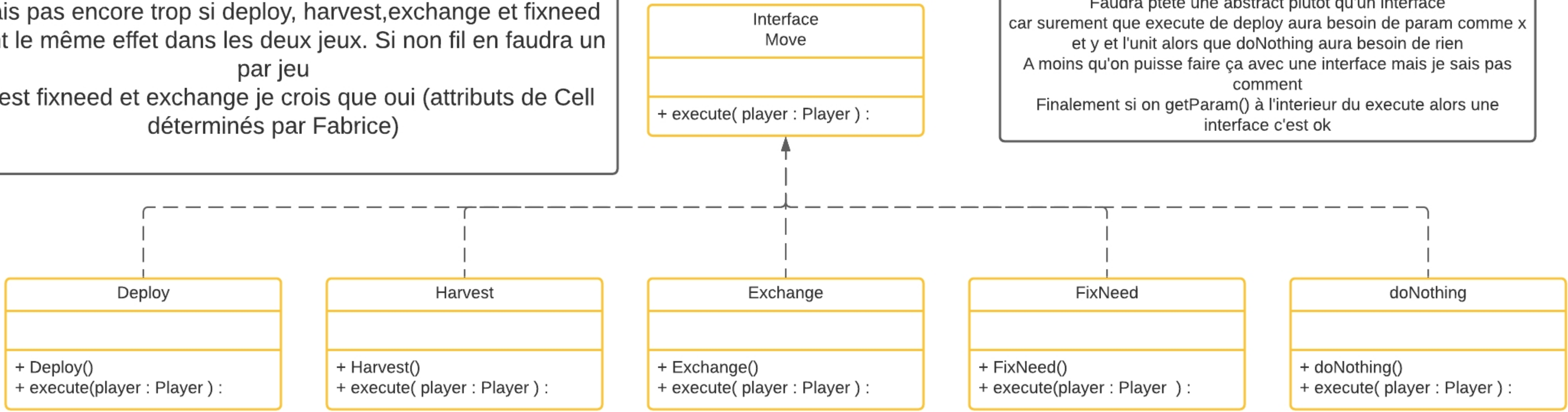
On peut ajouter une methode getParam() dans Strategie ,des que execute() a besoin d'une valeur donnée par le joueur il peut faire player.strategie.getParam()

-soit chaque objet Move a aussi une methode getParam() et dans strategie chooseMove prend un Move puis lance la fonction getParam de ce move mais donc strategie doit renvoyer un objet + un tableau de param qui devront être ensuite utilisé par objet.execute() et ça je sais pas si on peut spoiler non on peut pas on est en java



Du coup je
sais pas trop si
player a
besoin d'avoir
un deployUnit

Je sais pas encore trop si deploy, harvest,exchange et fixneed auront le même effet dans les deux jeux. Si non fil en faudra un par jeu
harvest fixneed et exchange je crois que oui (attributs de Cell déterminés par Fabrice)



Faudra ptete une abstract plutot qu'un interface car surement que execute de deploy aura besoin de param comme x et y et l'unit alors que doNothing aura besoin de rien
A moins qu'on puisse faire ça avec une interface mais je sais pas comment
Finalement si on getParam() à l'interieur du execute alors une interface c'est ok