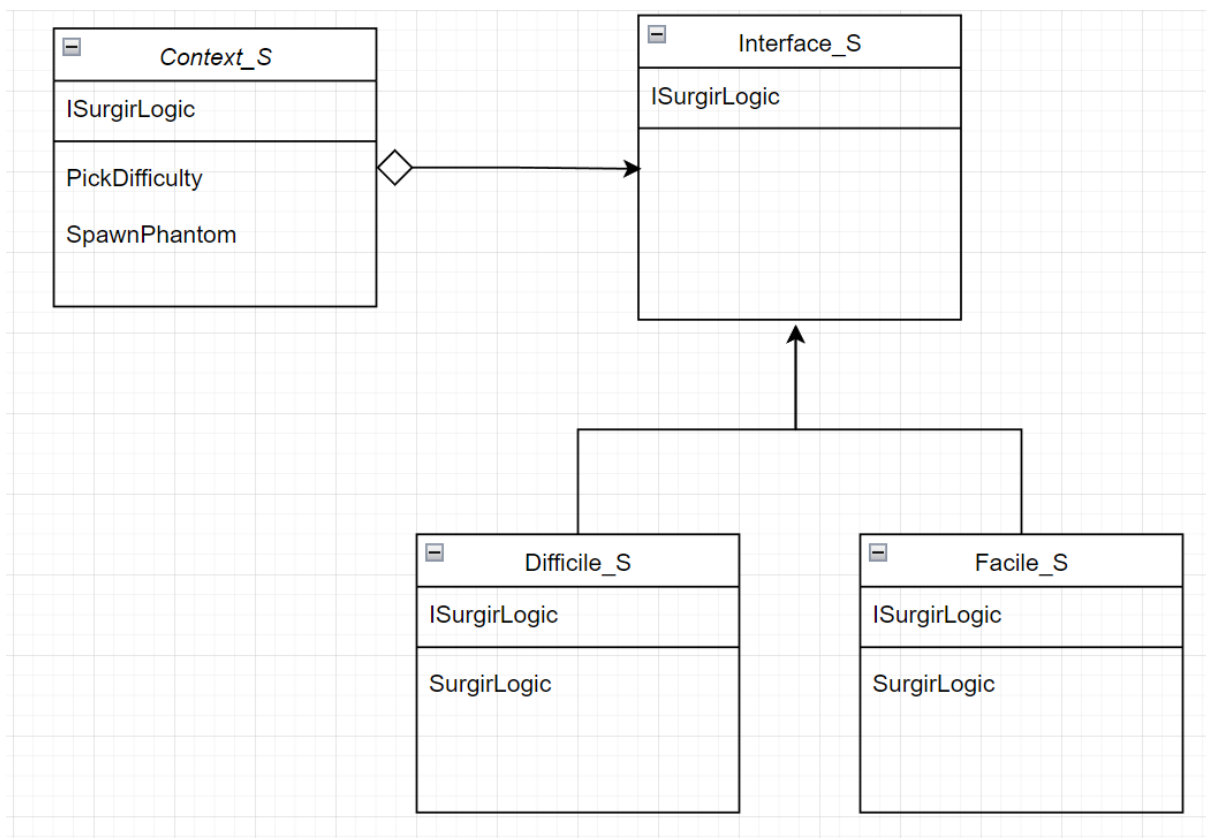


Rendu :

J'ai choisi d'implémenter le design pattern Stratégie pour choisir la difficulté du jeu, comme vue en cours, c'est pour moi le pattern idéal. En effet, quand vous commencer la partie, deux difficultés s'offre à vous, Stratégie permet donc sans surcharger le code de définir deux classes séparé qui sont définies dans un seul contexte.



Attendu :

Pour pouvoir prévoir la possibilité d'ajouter le support manette, le pattern Stratégie serait pour moi encore la meilleure solution pour les mêmes raisons énumérées précédemment.

Pour les IA, le design État convient parfaitement à la demande, c'était l'exemple utilisé pour nous apprendre ce design et à raison. En effet, c'est un patron de conception comportemental qui permet de modifier le comportement d'un objet lorsque son état interne change. L'objet donne l'impression qu'il change de classe.

Pour le niveau généré de façon procédurale, le design pattern du monteur pourrait être efficace, c'était une idée qui avait été suggéré à la fin de ma présentation sur ce design, car la génération procédurale aurait besoin de différentes variantes de level Design en utilisant des objets eux même construits par le même code de construction et c'est exactement ce que nous propose ce design.

Pour enregistrer la partie, le memento convient car c'est un patron de conception comportemental qui permet de sauvegarder et de rétablir l'état précédent d'un objet sans révéler les détails de son implémentation.