

- rationale

First I design Board create Square. Because it is the board's responsibility to know the Square. Doing this I'm decreasing the coupling of the design because, for example, the GameSystem has no coupling with Square.

Second, I assign all the validation to the Board because Board has the knowing responsibility and doing responsibility of the Square and whenever GameSystem needs to validate sth. it must know the internal distribution of Board so it's very convenience to assign work in such a way. Besides, I increased the cohesion of the design because this is a proper way to assign works. Also, I make the the Board tell GameSystem about the result of every checks so that GameSystem can tell the player which mistake he made.

Then after the validation, apply the Move, which include SP and Char: apply Char first, then activate existing SP, last apply newly added SP. (One player can only add one SP in a turn). Last apply SP so that newly added SP won't affect the player's Move.

Use decorator pattern to Square.

For the minor:

(1) First I want to have an Rack Object, then I abandon this idea because rack is nothing but a ArrayList(Primitive Type) and can be easily described. So I then delete this object.

The same thing happened to the Dictionary Object. I delete it and add a private field in Board as a HashSet. Primitive type usually described as attribute.

(2) At first I want to cache the Move in Board. Then I abandon this idea because I don't think the Board has the responsibility to do so.