Question

After reading this subsection, would the Observer Pattern be a good pattern to use when implementing a gameboard? Why or why not?

Answer

Observer Pattern would be a very suitable pattern for the implementation of the gameboard. We can separate the board into a GameBoard Class and a Display Class, with GameBoard being the subject and Display being the observer. The GameBoard will contain the actual implementation of the state of the game and the Display is responsible for printing out the gameboard. When the state of the GameBoard changes, it will notify the Display and change the display. This would be very useful because we separate the tasks of the gameboard, which improves coupling and satisfies the single responsibility principle.

Question

Suppose that we wanted to model SLC and Needles Hall more closely to Chance and Community Chest cards. Is there a suitable design pattern you could use? How would you use it?

Answer

We think that the Factory Method is a suitable design pattern to implement such features. We would build a Chance/CommunityChest Class and a Card Class, where the Chance/CommunityChest Class is the creator and the Card class is the product. When palyer step on a Chance or a Community Chest block, the class method creates a random Card and apply the effect of the Card to this player.

Question

Is the Decorator Pattern a good pattern to use when implementing Improvements? Why or why not?

Answer

Decorator Pattern is a good pattern to implement the improvements because the tuition fee for buildings are different and depend on the current number of improvements. By using a Decorator Pattern, we can determine the tuition at runtime. Moreover, since adding feactures directly to the Upgradable can be difficult and lead to errors, the Decorator Pattern would be a simpler and safer tool to use.