

Discussion

Modify Move class and add Placement class

Now my Move class contains a list of letter tile placements and special tile placements. I add a Placement class to represent a letter placement or a special tile placement. Each placement has a tile and its location. Each player has its own Move. Placements in Move class will update in each turn. Move class is like a bridge between player and game. It holds temporary placements and check if letter tile placements are collinear. Then pass move as parameters to board, ask board to check validation associating with information in move class. This improves cohesion and lower coupling.

Do not place tiles on board in isValid method

Now in the isValid method, it only check validation using information from move class and from board. Do not place tiles on the board, so I do not need to undo placements if it's not valid. This makes isValid method makes sense.

Use abstract class instead of interface

I use an abstract class for special tiles instead of interface. It's better because abstract class can have some concrete methods and different special tiles do have some methods and some attributes in common. Then abstract class has abstract methods which are different between different special tile. It achieved information hiding and code reuse, which is better than interface.

Add game change listener and use observer pattern

In the GUI part, I use GameChangeListener class to achieve observer pattern. GamePanel class implements GameChangeListener class.